Figure 1: Relationship of Sequoia RBigFootS and STC ACS in terms of latency and capacity

For systems costing $250,000 to $500,000. Bigfoot relies on several latency lowering techniques: this graph assume 80% to 85% of accesses go to disk cache, 10% to 15% write to a logging tape system, and 1% to 5% are random tape reads. Figure 2: Typical Storage Hierarchy, Circa, 1995.

Conventional disks have been replaced by disk arrays, a method of obtaining much higher I/O bandwidth by striping data across multiple disks. A new level of storage emerges between on -line disk and off-line tape. It provides very high capacity, but at access times measured in seconds due to the mechanical robot. Hence the names Rrobo-lineS or Rnear-line.S For robo-line storage, optical disk jukeboxes have faster access times than tape-oriented systems, but lower capacity. We expect the storage hierarchy to span local and wide area networks.

Vendor	Formfactor (inches)	Drives	Platters	Capacity (per platter)	Data Transfer (KBytes/second)	Access Time (second)
HP 6300 Model 20GB/A	5.25S	2	32	600 MB	340 - 680	7.0 (pick) 4.0 (load) 2.4 (spin-up) 0.1 (seek)
Kodak 6800 Automated Disk Library	14S	1 - 3	50 - 150	6800 MB	1000	6.5 (disk change) 0.1 - 0.7 (seek)

Table 1. Relevant Metrics for Alternative Optical Disk Jukeboxes.

Technology	Capacity (MB)	Media Cost (Removable)	Density (Mb/in2)	Data Xfer (KB/s)	Access Time (1/3 full seek)	Xchg Time (unload/load)	Media Limits (Reads)	(Writes)
Conventional Tape								
Reel-to-Reel (1/2S)	140	$5	0.11	549	minutes	minutes	0	0
Cartridge (1/4S)	150	$15	1.25	92	minutes	minutes	0	0
IBM 3480 (1/2S)	200	$5	0.87	3,000	15 secs	10 secs	0	0
Helical Scan Tape								
VHS (1/2S)	15,000	$29	?	4,000	45 secs	6 secs	500	500
Video (8mm)	4,600	$14	70.56	492	45 secs	100 secs	500	500
DAT (4mm)	1,300	$8	114.07	183	20 secs	55 secs	500	500
DD-2 (19mm)	25,000	$140	46.00	15,000	15 secs	5 secs	1,000	1,000

Optical Tape								
CREO (35mm)	1,000,000	$5000?	224.00	3,000	30 secs	minutes	0	1
LaserTape (1/2S)	50,000	$35	224.00	3,000	15 secs	10 secs	0	1
Magnetic Disk (fixed media)								
Seagate Elite (5.25S)	1,200	n.a.	63.01	3,000	18 ms	n.a.	0	0
IBM 3390 (10.5S)	3,800	n.a.	62.44	4,250	20 ms	n.a.	0	0
Optical Disk								
Sony MO (5.25S)	640	$100	453.54	500	100 ms	8 secs	0	0
Kodak (14S)	3,200	$500	296.33	1,000	100Us ms	7 secs	0	1

Table 2. Relevant Metrics for Alternative Storage Technologies.

Latency of robo-line storage is determined by Exchange Time (Xchg Time: the time to unload the removable media and then load a new unit and get it ready to read) plus the time for an average seek (time to seek 1/3 of the full storage on the media.) Helical scan tapes wear out after 1000 to 2000 accesses.

Operation	Machine	SPECMarks	Network	Raw B/W (Mb/s)	Latency (ms)
Sprite RPC	Sun-3	2	Ethernet	10	2.5
Sprite RPC	DS5000	18	Ethernet	10	0.8
Sprite RPC	Sun-3	2	Ultranet	1000	3.5
UW RPC	DS5000	18	Ethernet	10	0.34
UW RPC	DS5000	18	FDDI	100	0.38
Nectar	Sun-4	9	Nectar	100	0.45

Table 3. Bandwidth and Latency of Some Workstations and Networks.

Note that the higher bandwidths of the FDDI and UltraNetwork do not yield lower network latencies. While faster processor will reduce latency, it does not do so at the same rate as its processing power increases. Latency overhead is expected to dominate network overhead.

	Capacity	Latency	Bandwidth	Reliability
Massive Storage Laboratory	High	High	Medium	High
Managing Storage Latency	Medium	High	Low	--
Compression	High	Medium	High	--

Interleaving Multiple Components	--	Low	High	--
Redundant Components	--	--	Medium	High

Table 4. Impact of our four approaches on each of the four classic storage metrics. For Capacity: Use Tape

For Writes: Use Tape

For Repeated Reads: Use Optical Disk

For High Write Bandwidth: Use Striped Tape Readers/Writers (See Section 3.7)

For High Read Bandwidth: Use Striped Optical Disk Readers (See Section 3.7)

Table 5. Responsibilities in a hybrid optical disk/helical scan tape massive storage system. Figure 3: Improved Transfer Rate via Data Compression

Peripheral manufacturers are embedding lossless compression hardware into individual I/O devices, such as tape drives. A system level approach can support compression/decompression along any of the interfaces of the system, such as the local area network (LAN), local device, wide area network (WAN), and remote device. In addition, the system level approach can leverage information about the types of files to apply more effective data type specific lossy compression strategies.

Robo-line Storage: Low Latency, High Capacity Storage Systems over Geographically Distributed Networks

Randy H. Katz, Thomas E. Anderson, John K. Ousterhout, David A. Patterson

Sequoia 2000 Technical Report 91/3

Computer Science Division

Electrical Engineering and Computer Science Department

University of California, Berkeley

Berkeley, CA 94720

Abstract: Rapid advances in high performance computing are making possible more complete and accurate computer-based modeling of complex physical phenomena, such as weather front interactions, dynamics of chemical reactions, numerical aerodynamic analysis of airframes, and ocean-land-atmosphere interactions. Many of these Rgrand challengeS applications are as demanding of the underlying storage system, in terms of their capacity and bandwidth requirements, as they are on the computational power of the processor. A global view of the EarthUs ocean chlorophyll and land vegetation requires over 2 terabytes of raw satellite image data [ISTP91]!

In this paper, we describe our planned research program in high capacity, high bandwidth storage systems. The project has four overall goals. First, we will examine new methods for high capacity storage systems, made possible by low cost, small formfactor magnetic and optical tape systems. Second, access to the storage system will be low latency and high bandwidth. To achieve this, we must interleave data transfer at all levels of the storage system, including devices, controllers, servers, and communications links. Latency will be reduced by extensive caching throughout the storage hierarchy. Third, we will provide effective man- agement of a storage hierarchy, extending the techniques already developed by Ousterhout for his Log Structured File System. Finally, we will construct a protototype high capacity file server, suitable for use on the National Research and Education Network (NREN). Such research must be a cornerstone of any coher- ent program in high performance computing and communications.

1. Introduction

The past decade has witnessed stunning increases in the computational power available for a broad spectrum of applications and users. Timeshared machines serving dozens of users with a performance rating of 1 SPECmark have been replaced by workstations that dedicate 50 SPECmarks or more to a single user. In the supercomputing arena, $10-million machines capable of 200 megaFLOPS have given way to machines in the same price range capable of almost 12,000 megaFLOPS. Furthermore, the explosion in computing power seems likely to continue for many more years. By the mid- to late -1990s we will see workstations containing multiple 500-SPECmark CPUs and supercomputers offering performance in the teraFLOPS range. The increases in CPU power will revolutionize not only Computer Science, but many other fields that use computers, such as Physics, Chemistry, Earth Sciences, and Economics.

Processing power alone does not make a computing system, however. Every increase in CPU power must be accompanied by an increase in the capacity and bandwidth of its storage system in order to provide the additional information that will be manipulated by faster CPUs. There is already some evidence that new systems are unbalanced in their storage capacities relative to their processing power. For example, the Intel Touchstone machine is 500% faster than the Cray-YMP, yet it has only 4% to 8% as much storage on-line (secondary plus tertiary) as typical Cray-YMP supercomputers centers.

Over the past four years, our group has pursued a successful research program to develop new high- performance I/O architectures to match the ever higher performance processors. Our major research achievements are RAID, Redundant Arrays of Inexpensive Disks, and Log Structured File Systems. The latter overcomes the one disadvantage of RAIDs and lays the groundwork for very wide application of RAID technology. The Berkeley RAID project probably represents the single largest coordinated hardware-software research program in I/O architectures.

So far, we have focussed on I/O performance, and in particular on bandwidth. We took this approach because we saw the raw performance of CPUs increasing much faster than the raw performance of I/O devices, and we were afraid that slow I/O devices would become a performance bottleneck. We now believe, however, that capacity is as important an issue as bandwidth. Very fast computers not only need to access information quickly, but they need to incorporate an ever-increasing amount of information into their calculations. Unfortunately, current storage systems are woefully inadequate for storing and accessing information on the scale that is needed for future computing systems. Only a few large-scale storage systems exist, such as the Storage Technology Automated Cartridge System (ACS); they are very expensive, they don’t use the most capacious and cost effective storage technology, they are not widely available, and they are somewhat restrictive in their system integration. New developments in storage technology, such as helical-scan tapes and optical disks and tape, offer the prospect of storage systems containing tens or hundreds of terabytes. This represents a thousand-fold increase in storage capacity relative to most of today’s systems. We believe that such massive increases in on-line storage will be at least as revolutionary as the increases in computing power.

Storage capacity is probably even more important for applications outside of Computer Science than for applications in the field. [ISTP91] describes several RGrand ChallengeS scientific applications that will drive high-performance computing and communications in the coming decade, and many of these challenges are limited as much by storage capacity as by computational power. One of the most interesting grand challenges involves the analysis of satellite data to understand global effects such as weather front interactions and ocean-land-atmosphere interactions. The data requirements for such studies are truly massive: for example, a global view of the Earth’s ocean chlorophyll and land vegetation requires over two terabytes of raw satellite image data [ISTP91]. Over the next ten years the Earth Observation System will provide enormous amounts of new data for earth scientists (approximately one terabyte per day when it is fully functional). Efficient access to this massive information base is an enabling technology for research in global change and the earth sciences.

Our research group is actively participating in a project, called Sequoia 2000, which attacks the grand challenge of global change. It involves global change researchers at U.C. Santa Barbara, U.C. Los Angeles, and the Scripps Institute, plus computer scientists from Berkeley and other U.C. campuses. The Sequoia 2000 project brings together computer scientists and global change researchers to develop new storage systems and make them available over high-speed networks to scientists studying satellite data. Sequoia 2000 provides us with the exciting opportunity to work closely with demanding users in one of the most important research areas of this decade. Digital Equipment Corporation and the University of California have provided core funding for the Sequoia 2000 project in the form of workstations and telecommunications resources to interconnect the participating institutions.

In this paper, we describe a program of research to complement the Sequoia 2000 project. Our goal is to develop system architectures for secondary and tertiary storage systems and high-speed networks that will allow storage systems with 100-1000 terabytes total capacity to become practical and widely used by the mid- to late-1990s. We will test our ideas by building a prototype named RBigfoot,S that will store 10-100 terabytes (depending on the maturation of emerging storage technologies) for Sequoia global change researchers. We will evaluate our ideas and the Bigfoot prototype using Sequoia applications as benchmarks. Bigfoot will provide a thousand-fold increase in on-line storage capacity over today’s disk-based systems, and it will provide a 10-100x improvement in capacity/cost over today’s tertiary storage systems. Figure 1 demonstrates the area of investigation of our research program.

Our research program extends the RAID work in three directions: tertiary storage, network integration, and application support. The first new dimension is the inclusion of tertiary storage, such as robotically-controlled libraries of helical-scan tapes or optical disks. Such systems are necessary to provide multi-terabyte capacity at low cost, but they have serious performance limitations both in terms of latency and bandwidth. For example, the latency of a single access can be measured in tens to hundreds of seconds; this is so high that it can make demand fetches intolerable. One of the themes of our work is a multi-faceted attack on the latency of tertiary storage to make it as Ron-lineS as possible. In addition, the transfer speed of tertiary storage is almost two orders of magnitude worse than that of disk arrays. Another of our research themes is to use compression and interleaving at a number of levels to amplify the capacity and bandwidth of storage systems.

The second new dimension to our work here is techniques for integrating storage systems into a networked environment. This includes a number of research thrusts, such as striping data across servers and across networks in order to increase performance and using compression to amplify the limited bandwidth of long-haul networks.

The third way in which the research extends and improves upon the RAID work is that it couples us tightly to the global change scientists of Sequoia 2000. This application coupling provides us with a source of measurement data to guide our designs. It also allows us to focus our efforts on problems whose solution will provide the greatest benefit. At present the global change researchers have no viable alternative for their storage requirements, so we are guaranteed that our research results will be used and evaluated.

The rest of this paper is organized as follows. In the next section, we describe the trends in the under- lying storage and network technologies that are making possible our vision of very large storage systems on a network. Section 3 describes our technical approach in detail, focussing on our strategies for library management, reduced latency, increased bandwidth, and geographic distribution. Section 4 compares our approach with other related research efforts. Our summary and conclusions are given in Section 6.

2. Technical Rationale

2.1. Introduction

In this section, we describe the detailed technology trends in storage systems hardware and software, applications, and compression that lead us to believe that significant new results can be achieved in distrib- uted storage systems. These trends provide the technical underpinings of the research discussed in the next section.

The rest of this section is organized as follows. In Section 2.2, we review the trends in storage systems technology. We concentrate on how the storage hierarchy has evolved over the past decade, and predict how we see it continuing to evolve through the end of this decade. The key technical developments, such as disk arrays, robo-line storage, tape technology, and storage systems on network-based interconnect, are also de- scribed. Section 2.3 describes the developments in software for managing the storage hiearchy, in particular, the Mass Storage System Reference Model. Section 2.4 reviews the needs of scientific applications, the most likely clients for kinds of massive storage systems described in this proposal. Compression technolo- gy, a key element of our approach for managing image data across geographically distributed sites, is dis- cussed in Section 2.5.

2.2. State of the Art in Storage Systems Technology

2.2.1. Evolution of Storage Hierarchies

The storage hierarchy is traditionally modeled as a pyramid, with a small amount of expensive, fast storage at the pinnacle and larger capacity, lower cost, and lower performance storage as we move towards the base. In general, there are orders of magnitude differences in capacity, access time, and cost among the layers of the hierarchy.

A typical minicomputer of 1980 had a very simple memory hierarchy: perhaps 4 - 8 MBytes of main memory, 100 MBytes of magnetic disk, and essentially unlimited magnetic tape. The primary storage, at least as seen from the viewpoint of the I/O system, was a small file cache allocated in the main memory. This held data likely to be accessed in the near future. Secondary storage was universally provided by magnetic disk, with transfer rates in the 1 MByte/second range and access times of approximately 50 ms. Magnetic tape provided the tertiary storage off-line, primarily for archive/back-up. Nine track, 6250 BPI tapes, able to hold about 140 MBytes, were the dominant technology.

By 1990, the environment shifted to workstation/server computing, and with it, a more complex and distributed storage hierarchy. A primary performance limitation is the added latency of network accesses to obtain data from a remote server. With the advances in semiconductor technology, it is not unusual to find todayUs client workstations with 4 - 8 MBytes (of 32 MBytes) dedicated for use as a client cache. Most of the serverUs memory, perhaps 128 MBytes, is also dedicated to a cache function. Much of the active portion of the file system, especially the file system Rmeta-data,S can be held in the serverUs fast semiconductor memory, thus avoiding the latency of a disk access in servicing user requests.

The success of client/server computing in the workstation environment suggests that a similar ap- proach can work in the high performance arena given sufficiently fast interconnect between the client pro- cessor and the storage server. Figure 2 depicts one possible scenario for the storage hierarchy of 1995. Three major technical innovations shape the organization: disk arrays, storage subsystems based on optical disk or automated tape libraries, and high bandwidth wide area networks. To understand why these will become pervasive components of the future storage hierarchy, it is necessary to review the progress in the underlying storage and network technologies.

2.2.2. Disk Arrays, RRobo-LineS Storage Systems, and Network Integration

Because of the rapidly decreasing formfactor of magnetic disks, it is becoming attractive to replace a small number of large disk drives with very many small drives. This is called a disk array. The resulting secondary storage system can have much higher capacity since small format drives traditionally obtain the highest areal densities. And since the performance of both large and small disk drives is limited by mechanical delays, it is no surprise that performance can be dramatically improved if the data to be accessed is spread across many disk actuators. Disk arrays provide a method of organizing many disk drives to appear logically as a small number of very reliable drives of high capacity and high bandwidth [Katz 89].

Unlike conventional archival tape, which is meant to be written once and (hopefully) never read, robo- line storage systems (also called near-line storage systems) are designed to provide a very large storage capacity that can be frequently read and potentially rewritten. This is accomplished through the use of very densely recorded media, such as optical disk or high capacity tape, in conjunction with robotic RpickersS to stage media between shelves and readers. Access times are measured in milliseconds to seconds for data that has already been loaded in a reader to tens of seconds for data on the shelf. The name Rrobo-lineS suggests a compromise in both latency and cost between directly connected on-line storage and off-line storage which requires the assistance of a human operator.

Optically recorded disks have long been thought to be ideal for filling this level of the storage hierar- chy [Ranade 90]. They combine improved storage capacity (2 GBytes per platter surface originally to over 6 GBytes per side today) with access times that are approximately a factor of ten slower than conventional magnetic disks (several hundred milliseconds). The first generation of optical disks were written once, but could be read many times, leading to the term RWORMS to describe the technology. The write once nature of optical storage actually makes it better suited for an archival medium than robo-line storage, since it is impossible to accidently overwrite data once it has been written. A problem has been its relatively slow transfer rate, 100K P 200K bytes per second. Newer generations of optical drives now exceed one megabyte per second transfers.

Magneto-optical technologies, based on a combination of optical and magnetic recording techniques, have recently led to the availability of erasable optical drives. Read transfer rates are comparable to those of conventional magnetic disks. Access times are still slower than a magnetic disk due to the more massive read/write mechanism holding the laser optics. The write transfer rate is worse in optical disk systems because (1) the disk surface must first be erased before new data can be recorded, and (2) the written data must be reread to verify that it was written correctly to the disk surface. Thus, a write operation could require three disk revolutions before it completes.

Nevertheless, as the formfactor and price of optical drives continue to decrease, optical disk libraries will become more pervasive. Table 1 describes the system parameters of two alternative optical disk juke- box systems, the Hewlett-Packard Series 6300 Model 20GB/A Optical Disk Library System [Hewlett-Pack- ard 89] and the Kodak Optical Disk System 6800 Automated Disk Library [Kodak 90]. Despite an order of magnitude difference in the price of these two systems, the expected access time only varies between 7 and 14 seconds!

2.2.3. Tape Technology

The success of automated tape libraries has demonstrated that tape can also be used to implement a robo-line storage system. The most pervasive magnetic tape technology available today is based on the IBM

3480 half-inch tape cartridge, storing 200 MBytes and providing transfer rates of 3 MBytes per second. However, there has been an enormous increase in tape capacity, driven primarily by helical scan recording methods. The technology is based on the same tape transport mechanisms developed for video cassette recorders in the VHS and 8 mm tape formats and the newer 4 mm digital audio tape (DAT) systems.

Each of these systems provide a very high storage capacity in a small easy -to-handle cartridge. The small formfactor makes them particularly attractive as the basis for automated data libraries, because of the simpler robots that can be used. Tape systems from Exabyte, based on the 8 mm video tape format, can store 5 GBytes and transfer at approximately 500 KBytes per second. A tape library system based on a 19S rack can hold up to four tape readers and over one hundred 8mm cartridges, thus providing a storage capacity of 500 GBs [Exabyte90] for less than $40,000 (OEM pricing).

DAT tape provides smaller capacity and bandwidth than 8 mm, but enjoys other advantages [Tan 89]. Low cost tape readers in the 3.5S formfactor, the size of a personal computer floppy disk drive, are readily available. This makes possible the construction of tape libraries with a higher ratio of tape readers to tape media, increasing the aggregate bandwidth to the robo-line storage system. In addition, the DAT tape formats support subindex fields which can be searched at speed two hundred times greater than the normal read/write speed. A given file can be found on a DAT tape in an average search time of only 20 seconds. Due to competitive pressures, 8mm tape systems have recently incorporated a similar fast search capability.

Helical scan techniques are not limited to consumer applications, but have also been applied for certain instrument recording applications, such as satellite telemetry, which require high capacity and bandwidth. The Ampex TeraStore Data D-2 (DD-2) cassette recorder features a 15 MByte/second transfer rate from tape cassettes that can hold from 25 GBytes (small cassette) to 165 GBytes (large cassette). An associated tape robot can hold up to 6.4 TBytes of data (256 small cassettes) in 21 square feet (306 GBytes/sq. ft.). Because of their relatively low production volumes, such systems are very expensive.

A new recording technology that appears promising is optical tape. The recording medium is called digital paper, a material constructed from an optically sensitive layer that has been coated onto a substrate similar to magnetic tape. The basic recording technique is similar to write once optical disk storage. One 12 inch diameter by 2400 foot reel holds 1 TB of data, can be read or written at the rate of 3 MBytes per second, and can be accessed in a remarkable average time of 28 seconds. Two companies are developing tape readers for digital paper: CREO Corporation [Spencer 88] and LaserTape Corporation. CREO makes use of a 12 inch tape reels and a unique laser scanner array to read and write multiple tracks 32-bits at a time. The system sells for over $200,000. LaserTape places digital paper in a conventional 3480 tape cartridge (50 GBytes capacity, 3 MBytes per second transfer rate). A 3480 tape reader that is RretrofittedS costs approximately $25,000 compared to the $15,000 for the older model. Existing tape library robotics for the 3480 cartridge formfactor can be adapted to LaserTape without changes.

Table 2 summarizes the relevant metrics of the alternative storage technologies. The metrics displayed are the capacity, removable media cost, areal density (in millions of bits per square inch), data transfer rate (KBytes per second for sustained transfers), average positioning times, exchange times of removable media (time to unload an old tape and load a new one), and maximum number of reads and writes the media can sustain. Exchange times and access times are especially important for evaluating robo-line storage media. The table shows that while helical scan media are very inexpensive, especially for consumer markets such as VHS, video, or DAT, the helical scan heads wear out the tapes in 500 to 1000 accesses. The long switch and access times plus limited accesses before the tape must be replaced decrease the attraction of helical scan for robo-line storage. The price per megabyte of optical disk media, on the other hand, is 30 to 80 times worse than helical scan tapes. Overall, optical tape appears to be the most promising technology in terms of capacity, latency, and bandwidth, but the optical tape technology is the least mature of all the technologies in Table 2.

The key technical challenges are how to build a storage system with the low-cost capacity of tape with the access times of optical disk. We intend to overcome the low bandwidth of robo-line storage systems by leveraging interleaving and compression techniques. And we will reduce the latency to robo-line storage system by constructing a storage hierarchy that migrates infrequently accessed data to the tape while keeping frequently accessed data on disk.

2.2.4. Networks

The 1980Us were the Ethernet decade. Ten-megabit Ethernet became widely available in the early 1980Us, and interface chips quickly became cheap enough to include with all engineering workstations. Ethernet, as the dominant technology for local-area networks, spurred the development of network file systems and a variety of other network software, and played a large role in the emergence of high performance engineering workstations.

By the end of the 1980Us it was clear that 10 Mbits/second was not enough bandwidth to support the high-speed RISC workstations of the 1990Us. In the last few years a number of high-speed networking technologies have appeared, including the FDDI standard (100 Mbits/s), UltraNet (100 MByte/second), and a number of research efforts such as CMUUs Nectar (100 Mbit/second) [Arnould 89] and DECUs Autonet (100 Mbit/second per link with higher aggregate network bandwidth) [Schroeder 90]. Our rule of thumb, based on experience with file systems at U. C. Berkeley for example, is that a local area network connecting 50- 100 workstations should have 1 Mbit/second of bandwidth for each SPECmark of computing power in a single workstation. Thus todayUs state-of-the-art workstations with 20 -50 SPECmarks can easily overload Ethernet. Given that workstations with 100-500 SPECmarks will be available within a few years, it seems likely that FDDI will be a short-lived standard and will be superceded very soon by a networking technol- ogy in the 1 Gbit range.

We see two overall trends in networking technology. First, bandwidths in the gigabit range seem cer- tain to become widely available in the next 5-10 years. At this speed, network bandwidth will be comparable to the memory bus bandwidth of the machines attached to the network, so there becomes very little performance distinction (at least in terms of bandwidth) between a device attached to an individual machine and a device accessed over the network.

The second overall trend is that network latency is not improving as rapidly as either network band- width or processor speed; in many cases, networks with higher bandwidth actually have worse latency than Ethernet. We measured the bandwidth and latency of a number of different networks (see Table 3). The commercially available UltraNetwork has much higher raw bandwidth than Ethernet, but the round trip latency for RPC is actually 30% worse. Also, note that a DECStation 5000 shows only a 3-fold reduction in network latency relative to a SUN-3, even though it has 10 times the processing power. Researchers at the University of Washington measured latencies 10% worse for FDDI than for Ethernet [Levy 91] even on a small FDDI network; a larger FDDI network will have higher latency due to the additional links packets must traverse. CMUUs Nectar project has made an aggressive attack on both latency and bandwidth; it improved bandwidth by nearly a factor of 10 relative to other systems, but improved latency by only about a factor of 2.

The lack of improvement in network latency presents potential problems for future networked sys- tems. Even in scientific environments with many large files, there are also many small files; typically, large files account for most of the bytes transferred, but most accesses are to small files. Furthermore, the increas- es in network bandwidth make more and more transfers latency dominated. For example, the median file size in an engineering environment is about 5 KBytes. On a 1 GBit/second network, such a file requires 0.04 milliseconds of transmission time. Given the latencies from Table 3, the cost of reading or writing such a file will be totally dominated by network latency. Even a 100-KByte file requires only 0.8 milliseconds of transmission time; more than one-third of the time to transfer such a file will be due to network latency.

2.3. Mass Storage System Reference Model

Supercomputer users have long had to deal with the problem that high performance machines do not come with scalable I/O systems. As a result, each of the major supercomputer centers has been forced to develop its own mass storage system, typically a network-based storage organization in which files are staged from the back-end storage server, usually a robo-line subsystem, to the front -end supercomputer.

The Mass Storage System (MSS) Reference Model was developed by the managers of these super- computer centers, to promote more interoperability among mass storage systems and to influence vendors to build such systems to a Rstandard.S The purpose of the reference model is to provide a framework within which standard interfaces can be defined. They begin with the fundamental underlying premise that the stor- age system will be distributed over multiple machines potentially running different operating systems.

The MSS Reference Model defines six elements of the mass storage system: Name Server, Bitfile Cli- ent, Bitfile Server, Storage Server, Physical Volume Repository, and Bitfile Mover. Bit files are the modelUs terminology for uninterpreted bit data streams. There are different ways to assign these elements to underlying hardware. For example, the Name Server and Bitfile Server may run on a single Mass Storage control processor, or they may run on independent communicating machines.

An applicationUs request for I/O service begins with a conversation with the Name Server. The name service maps a user-readable file name into an internally recognized bitfile ID. The clientUs requests for data are then sent to the Bitfile Server, identifying the desired files through their IDs. The Bitfile Server maps these into requests to the Storage Server. It handles the logical aspects of file storage and retrieval, such as directories and descriptor tables. The Storage Server handles the physical aspects of file storage, and manages the physical data volumes. It may request the Physical Data Repository to mount volumes if they are off-line. For example, one storage server may be specialized for tape handling while another manages disk. The Bitfile Mover is responsible for moving data between the Storage Server and the client, usually over a network. It provides the components and protocols for high-speed data transfer.

The MSS Reference Model has been incorporated into at least one commercial product: the Unitree* File Management System sold by General Atomics, Inc. This is a UNIX -based hierarchical storage man- agement system, based on software originally developed at the Lawrence Livermore National Laboratory. This, and other similar systems, are described in Section 4.2.

2.4. Applications Needs

[ISTP91] describes several RGrand ChallengeS scientific applications that will drive high performance computing and communications in the coming decade. As faster, more powerful processors become available, scientists will use them to compute over ever larger data sets with even finer time steps. These computations will place enormous demands, both for capacity and performance, on the underlying storage systems.

We highlight a particular applications areas of considerable interest to the Earth System Scientists with whom we are participating on the Sequoia 2000 project: the numerical simulation and remote sensing-based analysis of large scale land-ocean-atmosphere interactions. Much of what these scientists need to do is sim- ilar to defense applications: remote sensing, high definition systems, simulation of physical phenomena, weather prediction, and scientific visualization.

Earth System Science researchers need access to geophysical and biological information, as well as raw data from spaceborne instruments or in situ sensors. The researchers must be able to collate and cross-correlate data sets about the Earth by processing the data from the satellite and aircraft observatories and other selected sources. For example, Prof. Dozier of U.C.S.B. routinely examines images from the Thematic Mapper satellite to determine the water content of the snow cover in the mountains. Each image requires 300 MB, and he is interested in both the long term history and the most recent broadcasts. An additional application is in models of the dynamics, physics, and chemistry of climatic subsystems which are accom- plished through coupled General Circulation Models (GCMs). These can generate huge data sets of output that represent grids of variables denoting atmospheric, oceanic, and land surface conditions. The models need to be analyzed and validated by comparison with values generated by other models, as well as with those values actually measured by sensors in the field.

To get a feeling for the kinds of data sets involved in evaluating global change over the last century, consider the following. The archives of the National Meteorological Center (NMC) contain daily weather analyses from 1946, and collectively measure 2 x 1011 bits (20 GBytes). A second data set contains the weather observations from the logs of over 72 million ships between the years 1854 and 1979. In addition, high resolution satellite data collected since 1968 brings the total data volume to 1 x 1013 bits (1 TByte). As of late 1988, the mass storage system at NCAR in Boulder, Colorado stored almost 300,000 bit files and managed 50,000 3480 tape cartridges, for a total of 55 TBytes of data.

[Halem89] describes some of the computational challenges to be addressed by Earth System Science researchers in this decade. NASAUs Earth Observing System (EOS) is expected to archive 1 to 10 petabits (100 to 1000 TBytes) per year for the 20 year lifetime of the program starting in the mid-1990s! This is on the order of 1 TByte per day.

A key limitation for Earth System Science researchers is the lack of a storage system that meets their need for capacity and remote access. Much of the critical data for the analysis of long term climate changes are inaccessible to the average researcher. By working closely with Earth System Science research community, it becomes feasible for us to think of filling a multi-terabyte tertiary store.

2.5. Compression

Compression has long been of interest in computer systems as a method for increasing the capacity of the storage system. Algorithms can be characterized as being lossless or lossy. Lossless algorithms lose no information during the compression and decompression process. Lossy algorithms achieve higher compression ratios by losing information, and thus are only of use for image or video data where such losses can be tolerated.

One of the most extensively used lossless compression algorithms is the Lempel-Ziv method. A one pass algorithm, it builds a translation table on the fly, recoding variable length data strings with shorter length codewords built up in the translation table. For typical ASCII text, Lempel-Ziv yields compression ratios in the range of 2 to 3 (i.e., the compressed file is 1/2 to 1/3 the size of the uncompression file). The algorithm is fast enough to be implemented as software utilities on many systems. Industry standard deriv- atives of the Lempel-Ziv method are now being embedded in a variety of tape drive systems, primarily to increase the storage capacity of the medium.

For image data in which some loss of resolution can be tolerated, the alternative lossy compression strategies are very attractive. Two emerging industry standards are the JPEG (still image) and MPEG (video image) methods. We describe the JPEG (Joint Photographic Expert Group) method first.

JPEG begins by dividing an image into 8 x 8 pixel blocks. RGB values are then converted to an alternative colorspace representation based on luminance, chrominance, and intensity (UVY). This representation has better compression properties. At this point, an optional subsampling information losing step can be inserted. Basically, alternative rows and columns of pixels are deleted from the image.

The next step is to apply the forward discrete cosine transform (DCT), mapping the pixels into a frequency representation. Since most of the pictureUs information content is captured by the lower frequencies (upper lefthand corner of the 8x8 matrix), many of the higher frequency entries contain small values or zero.

The next step, quantization, introduces most of the information loss. The matrix entries are rescaled by dividing the 8 x 8 transform matrix by an 8 x 8 quantization matrix and then rounding. The quantization matrix is the major RparameterS to the compression algorithm that controls the level of loss in the image. The effect of the quantization step is to introduce zeros into the higher frequency entries of the matrix.

The final step is a Huffman encoding, which can apply run-length encoding to the streams of 0Us. This significantly reduces the number of bits needed to encode the image. By choosing the degree of subsampling and quantization, it is possible to vary the quality of the image. Lossless compression will typically yield a compression ratio of approximately 2:1, RexcellentS image quality is approximately 8:1, RhighS quality is 15:1, RgoodS quality is 20:1, and RfairS quality is 40:1. Depending on the sharpness and contrast in the image, much higher compression ratios can be achieved.

JPEG compression/decompression algorithms have recently become available as software utilities, and while these are acceptable for occasional image manipulation, they are not yet fast enough for 30 frame per second video decompression. Several companies, including CCube, Storm Technologies/Micron, and LSI Logic have produced special purpose hardware to accelerate JPEG compression/decompression.

The MPEG (Moving Picture Experts Group) method builds on top of the JPEG standard to apply compression to video image streams. MPEG adds new techniques to reduce temporal redundancy, while using JPEG methods to reduce spatial redundancy.

MPEG organizes a video stream into three different kinds of pictures: intrapictures (I), predicted pic- tures (P), and bidirectional pictures (B). An intrapicture carries enough information to be decompressed without reference to proceeding or following frames. It can be the target of a random access, but obtains only a moderate degree of compression. Predicted pictures are computed as a difference from a previous picture in the sequence. This obtains better compression for the forward translation of localized motion. Bi- directional pictures are computed in terms of differences between previous and future pictures. This obtains better compression for uncovered regions and exhibits better error recovery behavior. To decompress bidirectional pictures, the algorithm needs access to pictures before it and after it in the video sequence.

Compression ratios in the range of 50 to 100:1 are claimed for MPEG. The first hardware accelerator chip sets for MPEG, from CCube, have recently been demonstrated.

3. Technical Approach

3.1. Overview

The classic goals of a storage system are greater capacity, lower latency, greater bandwidth, and high reliability. We have four main approaches to towards these goals (see Table 4):

1. Massive Storage Laboratory: Massive storage is currently facing a revolution in new technologies, and there is little information on the performance/reliability trade-offs of these new options. We will create a laboratory containing several different storage systems, evaluate the tradeoffs between the dif- ferent technologies, and develop support software that can be used on multiple storage systems. The purpose of the lab is to give us head-start in evaluating the emerging technologies and in developing the software for them, as well as to give us a forum to influence future offerings from the industry.

2. Managing Storage Latency: While these emerging technologies make tremendous improvements in capacity, they do this at tremendous cost to latency. By taking a systems approach to the problem, we hope to overcome some of the latency problems. Ideas include avoiding tertiary accesses via caching and abstracting, lowering media latency by revising controller software, and designing hybrid massive storage

systems.

3. Compression: Data compression acts as a multiplier both for the capacity of a storage system and the bandwidth of interconnect. Traditional approaches to compression have focussed on isolated portions of a system, such as a single link or storage device. In this project we will take a system-level approach to compression: more data will stay compressed longer, and different compression algorithms will be used, depending on the nature of the data. By doing this we hope to increase the benefits provided by compression.

4. Interleaving Across Multiple Components: Interleaving provides a method to scale I/O bandwidth. If the transfer rate of a single disk is insufficient for the application, in the RAID project we spread the data over multiple disks and perform the transfers in parallel. This same basic argument applies to all aspects of the I/O system: it is possible to increase bandwidth by interleaving across multiple tapes, multiple robo-line storage subsystems, multiple file servers, and even multiple network interconnec- tions.

5. Redundant Components: The opportunity to replace large disks by small ones in RAID enabled both the interleaving to get higher bandwidth and the redundant drives to give higher reliability. The same approach applies here: multiple components in a massive storage system allow higher bandwidth via interleaving and higher reliability by providing more components than the absolute minimum. The extra components allow failures to be detected, corrected, and then avoided until repaired.

Each of these five approaches can potentially improve several of the storage system metrics. Table 4 shows the impact of each approach on the classic goals for storage systems.

3.2. Massive Storage Laboratory

One difficulty with massive storage is that the newest and most exciting technologies are being offered by essentially start-up companies. Many candidate technologies are coming to market for the first time in 1991 or 1992, and each technology is usually associated with a single company. So unlike the RAID project we have no body of knowledge on performance, reliability, or even the reality of some of the options that appear in trade magazines.

One solution is to wait two to three years for the dust to settle, and then work with the winners. This approach has several drawbacks:

% Time to General Acceptance: By waiting three years before starting with the technology, we delay its inclusion into computing systems; in particular, we delay the creation of the software that can put the technology to work, which is typically the obstacle to the success of any new computer technology.

% Influencing the Next Generation: By cooperating with these emerging companies and reporting the results of our experiments, we can be in a position influence the next generation of these products to be more effective in high performance, scalable computer systems.

% False Starts: With the wide variety of options, several groups will likely pick ones that prove to be white elephants. Sour initial experiences can result in rejection of an entire industry. Early evaluation by an unbiased research group can avoid wasting the time and money of many people making false starts, as well as avoid initial negative impressions that are difficult to overcome to gain acceptance of a new technology.

We believe that we will more quickly reach the goal of terabytes of storage being readily available by acting immediately. Hence our plan is to get several initial examples of this emerging technology and put them to work:

% Small robot and reader for 3480 cartridges using Digital Paper; each cartridge has 50 GB storage capac- ity in a write-once medium and the drives transfer at 3 MB/S bandwidth;

% Small robot and reader for 19 mm helical scan tapes (DD2), each with 25 GB capacity and 15 MB/S bandwidth;

% Small robot and reader for 1/2 inch helical scan tapes (T-120), each with 15 GB capacity and 4 MB/S bandwidth;

% Small robot and reader for 8 mm helical scan tapes, such as the EXB -120

% Stackers containing 4mm or 8mm helical scan tapes (DAT or video), each with 1 to 5 GB capacity and 0.5 MB/S bandwidth;

% Low cost robot for Read/Write optical disks, each with 0.65 GB capacity and 1 MB/S read bandwidth.

Hands-on experience with these systems will allow us to understand the strengths and the weaknesses of each option. Moreover, the software we create must work with several technologies because the RwinnersS may be application dependent, and our experience is that software is difficult to port unless it is developed in a heterogeneous environment.

As users of these new technologies we hope to be able answer questions about their reliability, based both on our experience and in working with the manufacturers to answer our questions. We found that get- ting reliability numbers on individual disks was nearly impossible when we asked as researchers, but that customers had a leverage with the company and could get the data. We believe this will be especially true of this emerging massive storage industry, as companies need to address the customerUs problems to sur- vive. Moreover, we can run experiments ourselves to verify the answers we get from these start-up compa- nies. And one clear goal in our interactions with companies is to influence the next version of the technology, particularly in making it work well in high performance, scalable computer systems.

This experience with these first system lays the foundation to construct a high capacity massive stor- age system that has low latency, scalable bandwidth, and high reliability.

3.3. Low Latency via Caching on the Local File Server

One of the first latency hiding techniques we call abstraction. It lowers latency by completely avoiding the access to tertiary storage. Our assumption is that many file accesses are for browsing rather than reading the file in full. To permit fast browsing without tertiary storage accesses, we will keep small abstracts on disk. An abstract is a highly-compressed version of a file, where the exact information kept in the abstract is chosen based on the type of the file. For an image, the abstract might be a sampled reduction of the image. In DozierUs case we might supply only the 3 most important spectral bands, reduce the grid to 1000 x 850 via averaging, and then compress. Just as we intend to use different compression algorithms depending on the type of data in a file (see Section 3.6 below), we can invoke different abstraction programs depending on that file type. Abstraction could reduce the size of the image by 7/3 x 7000/1000 x 6000/850 x 8 or 920. Hence Dozier could maintain almost 1000 abstracts on his local file server for the cost of one full image. His latency is limited only by the speed of the local area network, the speed of decompression, and the RAID on his file server.

Another application of abstraction is with text files, such as browsing through electronic mail messag- es. By keeping a subset of the information on the file serverQsuch as the date, sender, subject line, and selected index terms from the body of the messageQand then compressing that data, keeping a fraction of the message on the disk may serve to avoid many accesses to tertiary storage. For example, a message on the outline of this proposal contained 2011 characters. The author, subject line, date, and a few words from the body took 54 characters. If this data were compressed 3:1, then this abstract would represent less than 1% of the original message. In this case the userUs response time would be limited only by the latency of the local area network and the file server.

A second latency lowering technique is anticipatory fetching. Anticipatory fetching hides latency by reading the data in advance of its demand. This can take several flavors: reading all the files in a directory versus a single file on a request, reading all the files in a makefile versus sequential requests, passive learning of the probabilities of access to other files given past history, and user hints. As an example in this last category, assume a window pops up at 5PM to ask if Dozier will be in tomorrow, and if so what geographic areas and time periods might he be interested in. He would answer this question by browsing through the abstracts of the images he is likely to work on next. His estimate of usage is sent to the massive storage system which merges this request with all other users. Data is then shipped overnight to local file servers using the most economical means of data transfer. The data is on his file server when he comes in the next morning, available as fast as his machine can read the data over the network.

The third technique is file caching. Based on the use of files at a site, the data is kept on disk, in main memory of the local file server, or in the main memory of the workstation. Dozier wins when requesting a file that he has requested previously, and it appears nearly instantaneously depending on the its location in the hierarchy.

These ideas bring new challenges for the file system. While there exist well known techniques with two-level storage hierarchies, managing three-levels of storage offers new opportunities for the systems designer. In particular, we must balance the storage requirements and latency-hiding benefits of file caching with those of abstraction and anticipatory fetching. These issues can only be addressed by evaluating the patterns of access of real users, modeling the benefits of each option, and subjecting proposed policies to actual use. Sequoia 2000 provides us the framework to see how well these ideas will work.

3.4. Low Latency Helical Scan Tape Readers

As mentioned above, one obstacle to making inexpensive storage useful is the long load/unload times. While not a problem for archival applications, where data is written once and almost never read, it directly conflicts with the repeated unpredictable reads needs of robo-line storage. Preliminary investigations suggest that these long times are a function of the anticipated application rather than the technology. For example, some of this long load/unload time is spent checking to see if the tape has been written before. Another obstacle is rewinding the tape. Rewinding is used to set the read head over a dataless leader of the tape. Since tape wear and stretching are most likely to occur while the tape and heads are coming up to speed, the leader acts to reduce the chance of damaging data.

To turn archival storage into robo-line storage we will modify the controller firmware, ideally in co- operation with a controller manufacturer, to improve these characteristics. For example, a computer system knows if the tape is written or not in advance, thereby skipping the time for seeing is the tape is written. A second idea is to add multiple RleadersS interspersed throughout the tape, lowering the storage capacity by, say, 10% but avoiding the need of a full rewind before a tape is unloaded. Another benefit of multiple leaders is that the head would be closer to the middle of the tape then the beginning, thereby cutting the average time to find a file. Multiple leaders could potentially halve the long load/unload and seek time of 8mm helical scan tapes.

The difficulty of these proposals is that so little is known about helical scan media beyond the few companies with helical scan products that there well may be technical roadblocks to such suggestions. Again the massive storage laboratory (Section 3.2 above) should address these issues. Our intent is to try to enhance these characteristics in part to improve the prototype and in part to make the needs of robo-line known to the tape controller industry.

3.5. Low Latency via Another Level of the Storage Hierarchy

Helical scan tapes are cheaper, rewritable, and on a faster curve of density improvement than optical disks, but they have long load/unload and seek times (see Table 2 above) and limitations as to the number of reads or writes before the tapes must be replaced. Optical disks, on the other hand, have low load times and have no limits to the number of reads but they are more expensive, their density is improving slowly, and they write slowly. Hence the strengths and weaknesses of tapes and optical disks do not overlap. Another interesting fact is that many of the global change applications write much more data than is ever read.

One latency lowering technique that we will pursue is to use an optical disk jukebox as a cache on the helical scan tape libraries. The tapes would be used in a log structured file system, which is optimized for writes. (A log structured file system simply appends writes to the end of a sequential media, with reads caus- ing seeks that access data in proper order to get the most recent version of a file.) Hence long latency of the tape load/unload and seek would normally only occur when a tape is full. The first time the data is read the long latency of the helical scan tapes must be overcome, and the data is transferred to the magnetic disk of the file servers. Then a copy of the data will be placed on an optical disk in the jukebox. If the data is reread the request will be satisfied by the jukebox in a second.

One promising jukebox comes from a new company that promises 7 second access to any of 700 5.25S disks for about $100,000. When complemented by the low cost helical scan tapes, this combination may offer low latency for reads and writes with high capacity. When multiple optical and tape readers are striped together, this combination may also offer high read bandwidth and high write bandwidth. This hybrid stor- age system may offer performance-capacity-price characteristics that cannot be matched by homogenous tertiary storage system. Table 5 below illustrates the individual roles of the components in the hybrid orga- nization.

Once again these ideas can only be validated by interactions with real users, such as the Sequoia 2000 community. The threshold of when to make an optical copy will be determined by the relative costs of the media, write speeds, and patterns of use by the global change scientists.

3.6. Compression

3.6.1. Basic Concept of Compression

Compression in storage systems has traditionally been used to increase the capacity of the system. However, compression can also be used to amplify the bandwidths of communication channels. Many peripheral manufacturers are placing lossless compression hardware into their embedded device controllers. This has the effect of increasing tape capacity while maintaining the interchangeability of the tape media, at least among drives of the same manufacturer. (An industry standard algorithm for lossless compression has been proposed, and is being adopted by most peripheral manufacturers). Conventional unencoded data enters and exits the tape drive, and there is no change to the driver software on the host.

Unfortunately this architecture does little to improve the overall transfer bandwidth of the I/O system. >From a system viewpoint, it is advantageous to keep the data compressed until it is actually delivered to the application. Compressed data can be exploited to increase I/O system, memory system, and network band- width as well as storage device bandwidth.

Further, at a system level, knowledge of the file type can be used to choose among a variety of different compression algorithms, some of which may be most effective for text while others are better suited for vid- eo or image compression (see Figure 3). Even without application hints, it is often enough to examine the beginning of the file to heuristically determine its type. For example, the UNIX command RfileS guesses the type of file contents by examining the its first 512 bytes.

3.6.2. Technical Challenges of Compression

There are many challenges associated with embedding support for compression within the system. The first is whether it is necessary to include hardware accelerators for decompression. To some extent, this depends on the kind of data and compression algorithm, as well as the applicationUs tolerance for latency. Decompression of ASCII text files probably do not require hardware support while 30 frame per second video playback is impossible without it.

We expect the server to have full capabilities for compression and decompression. But the second challenge is how to support a heterogeneous environment in which some clients have special purpose hardware for compression/decompression while others do not. The file system must have knowledge of where the compression or decompression is to be done, based on the capabilities of the clients.

The industry standard compression algorithms, such as JPEG, offer an interesting possibility for vari- able resolution playback. Data can be placed in the storage system in lossless, encoded form. Depending on the available bandwidth available in the I/O path, the image can be played back at full resolution or at degraded resolution, by performing the quantization step and Huffman encoding Ron the fly.S We expect to explore the dynamic interplay between available bandwidth, bandwidth/resolution guarantees, and variable playback resolution.

Another interesting interaction exists between compression and error correction in the storage system. A bit error within a compressed file renders unreadable the portion of the file after the point of the error. Error correction schemes must be integrated with compression strategies to minimize the impact of bit er- rors. One possibility is to improve the error correction capabilities of data stored in the storage system, per- haps by using parallel error correct (see the discussion of interleaved tape and disk in the next section). Another is to partition the compressed data into units with which error correction is associated, rather than spreading the correction across the entire file.

We propose to develop an architecture in which to support system-level compression and decompres- sion in an internetworked storage environment. Both hardware and software strategies will be examined, supported at the level of I/O controllers, file server software, and hardware/software in the clients.

3.7. Interleaving Multiple Components

3.7.1. Interleaved Tape

Interleaving is a technique that increases bandwidth by using multiple storage or communication elements operating in parallel. Interleaving usually introduces slight additional latency and makes less efficient use of system resources than non-interleaved approaches, so it is does not make sense unless bandwidth is the performance bottleneck. The tertiary storage systems we are considering have relatively low bandwidth in comparison to the disk arrays used for secondary storage, which makes interleaving attractive. However, they also have relatively high latency, which lessens the benefit of interleaving. In this research we will apply to tertiary storage the same kinds of striping techniques that we have pioneered for disk storage and evaluate the architectural issues, benefits, and costs associated with interleaving.

Rather than construct a tape handling system of our own design, we will make use of commercially available robotics. Figure 4 shows the elements of one such system, an Exabyte EXB-120 Cartridge Han- dling System. We describe this system for the purposes of illustration; we are not yet ready to commit to a particular tertiary storage technology. The standard 19S rack can be configured with four 8mm tape readers, up to 116 x 5 GByte tape cartridges, and a tape handling robot that can pick any cartridge and deliver it to any tape reader in under 20 seconds. The robot mechanism and each tape reader are on independent SCSI (Small Computer System Interface) interfaces, making it possible to control the robot while independently reading from or writing to any of the tape drives. Exabyte is in the process of developing a 5.25S halfheight tape reader, which will make it possible to place eight tape drives into the subsystem. The tape handler also has an integrated bar code reader, making it possible to inventory the contents of library with ease.

Using a cartridge handling system such as the EXB-120, we propose to demonstrate the viability of striped tape as a method for improving the transfer bandwidth of helical scan tape. The concept is shown in Figure 5. Each tape transport has its own embedded controller that supports the SCSI command set. An in- dividual Exabyte tape drive has the potential to sustain 512 KBytes/second. By striping data from the same file across multiple drives, we can get a multiplicative speed-up in the transfer rate to a single file.

Actually achieving this speed-up will be difficult, however. There are a number of aspects that make striping tapes more difficult than striping disks. First, it is more difficult to synchronize the actions of sev- eral tape drives than several disks. With disks it is possible to synchronize the rotations of the drives so that they truly operate in lock-step. With tapes, no such synchronization is possible. Furthermore, the error correction techniques used by the Exabyte system add variability to the speed of each drive. The Exabyte system performs a verifying read after each write, and if errors are detected (even correctable ones) a new copy of the data is written to tape after the original copy. This leaves two copies of the data on tape, which slows down both the write and later reads.

This variation in drive speed makes it more difficult to achieve the full benefits of interleaving, since it could result in one or more drives sitting idle while slower drives catch up. If a drive stays idle too long, the tape system will unload the tape in order to reduce wear, and this results in a delay of several seconds to reload the tape before the next operation. The result is even greater variation in drive speed. We believe that the skew problems can be solved with adequate buffering, but we expect that experimentation will be required to determine the right sizes for buffers and the right algorithms for staging data into and out of them. An important aspect of our research will be to quantify the effect of tape skew in striped tape systems by constructing a software striping driver for a multiple tape drive system and measuring its performance. We have at our disposal several SCSI-bus analyzers that can be used to monitor activities on the interfaces between the host bus adapters (SCSI controllers designed to couple the host to the SCSI bus) and the tape drives. In addition, Exabyte will provide us with software that will allow us to monitor error correction events within the tape drives themselves.

The long load times for tape systems also work against interleaving. For example, once the robot arm has placed one tape in a drive, it takes about 5-10 seconds for it to load a second tape in a second drive. This means that the first drive will be able to start transferring about 5-10 seconds before the second drive, and during this 5-10 seconds it will be able to transfer 2.5-5 MBytes of data (for the Exabyte system). This implies that there is no benefit to striping within a single tape robot for files that are smaller than 2.5-5 MBytes. Furthermore, the long load times for tape, 100 seconds or more for the Exabyte system, result in very poor drive utilization if only a few MBytes of data are transferred per drive. This also argues for striping only very large