An Approach to Cost-Effective Terabyte Memory Systems

Randy H. Katz, David A. Patterson, Ann Chervenak-Drapeau, Joel Fine, Ethan Miller

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

Abstract

Cost effective terabyte memory systems are now becoming possible. New methods for high capacity storage systems have been made possible by low cost, small formfactor magnetic and optical tape systems. To achieve low latency and high bandwidth access to the storage system, we must interleave data transfer at all levels of the storage system, including devices, controllers, servers, and communications links. Latency can be reduced by extensive caching throughout the storage hierarchy. In addition, we must provide effective management of a storage hierarchy, extending the techniques already developed by Ousterhout for his Log Structured File System. We are incorporating these ideas into a prototype high capacity file server.

1. Introduction

The past decade has witnessed stunning increases in the computational power available for a broad spectrum of applications and users. The explosion in computing power seems likely to continue for many more years. Yet processing power alone does not make a computing system. Every increase in CPU power must be accompanied by an increase in the capacity and bandwidth of its storage system. 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.

Our previous investigations have focussed on I/O performance. We now believe 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. Current storage systems are woefully inadequate for storing and accessing information on a large scale. 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.

Our research group is actively participating in Sequoia 2000, a project attacking the grand challenge of global change. 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.

In this paper, we describe our approach for developing 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 "Bigfoot," 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.

2. Underlying Technologies

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 "pickers" 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 "robo-line" 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 oper-

0-8186-2655-0/92 $3.00 © 1992 IEEE

ator.

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 400 MBytes and providing transfer rates of 6 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 19" rack can hold up to four tape readers and over one hundred 8mm cartridges, thus providing a storage capacity of 500 GBs [Exabyte 90] 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 3.5" 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.

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 (100 GBytes capacity, 6 MBytes per second transfer rate). A 3480 tape reader that is "retrofitted" 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.

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:

1. **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.
2. **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.
3. **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 interconnections.
4. **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 approaches can potentially improve several of the storage system metrics. Table 1 shows the impact of each approach on the classic goals for storage systems.

3.2. Low Latency via Caching on the Local Server

Abstraction is our first strategy for latency hiding. 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. For Thematic Mapper satellite images, we might supply only the 3 most important spectral bands, reduce the 7000 x 6000 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, 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 we 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 messages. By keeping a subset of the information on the file server—such as the date, sender, subject line, and selected index terms from the body of the message—and 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 paper 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 user's 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 takes 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 the user 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. The user wins when requesting a file that he has requested previously, and it appears nearly instantaneously depending on 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.

3.3. Low Latency via Additional Storage Levels

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 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.

	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 1. 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
For High Read Bandwidth:	Use Striped Optical Disk Readers

Table 2. Responsibilities in a hybrid optical disk/helical scan tape massive storage system.

One interesting latency lowering technique uses 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 causing 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 5.25" 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 storage system may offer performance-capacity-price characteristics that cannot be matched by homogenous tertiary storage system. Table 2 illustrates the individual roles of the components in the hybrid organization.

3.4. 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 bandwidth 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 video or image compression. 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 "file" guesses the type of file contents by examining the its first 512 bytes.

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 application's 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 variable 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 "on the fly." 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 errors. One possibility is to improve the error correction capabilities of data stored in the storage system, perhaps 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 are developing an architecture in which to support system-level compression and decompression 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.

3.5. Interleaving Multiple Components

3.5.1. 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. A tertiary storage systems typically has 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.

We have been using the EXB-120 Cartridge Handling System to experiment with the viability of striped tape as a method for improving the transfer bandwidth of helical scan tape. Each tape transport has its own embedded controller that supports the SCSI command set. An individual 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 several 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.

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 files. Most likely a hybrid approach to striping will be necessary, where very large files are striped and small ones are not, or perhaps small files can be clustered into large groups which can then be striped.

Tape arrays face the same kinds of reliability challenges as disk arrays. The tape drive mechanisms, due to their complex electromechanical nature, are even less reliable than magnetic disks. Spreading data across multiple transports reduces the reliability even further. In addition, tape wear, and the eventual loss of the ability to read previously written data, is an important consideration. To guarantee that data in the array is kept available, a scheme comparable to parity striping for disk arrays is attractive. Tapes are organized into stripe groups of N+1 tapes, with parity redundancy computed horizontally bitwise across N tapes and stored on the N+1st tape. The striping software must keep track of which tapes form a stripe group. If a tape reader fails, the contents of tapes in that stripe position can still be read by inverting the parity calculation. Writes can continue as before, even though one tape cannot be accessed, but its contents will need to be reconstructed after the tape reader is repaired.

Tape data redundancy can also be exploited for the lagging tape problem described above, at least for reads. Rather than waiting for a slow tape to catch up, its contents can be reconstructed from the other tapes in its stripe group.

3.5.2. Interleaved Robots

For the purposes of reliability, it is important that storage robots have a degree of redundancy. If a robot breaks, it is disastrous if the storage system becomes unavailable. Further, in the striped tape organizations described above, the robo-line storage system could become unavailable should one of the tape drive mechanisms fail. Given these observations, it is attractive to interleave across multiple robots as well as readers. A "stripe" is formed from individual

tapes/readers controlled by different robots. This "orthogonal" organization protects against both reader and robot failures.

3.5.3. Interleaved Networks

The theme throughout this section is to increase bandwidth by striping across multiple objects. We do this at the level of devices (disks and tapes) and media handling robots, as well as servers. If one component cannot provide enough bandwidth, we simply add additional components in parallel and interleave accesses across them.

We need not stop at the level of interleaving across servers. If the network bandwidth is the bottleneck, then it should be possible to stripe accesses across multiple network interfaces as well.

An interesting issue in distributed control is how to get the servers participating in a common interleaved transfer to use different pathways to the client. This requires a different approach than the standard network routing literature. The existing algorithms are optimized for gradual load balancing and congestion avoidance. In the environment given above, the algorithms must make rapid routing decisions, geared towards maximizing the network bandwidth available to related transfers.

3.6. Redundant Components

The sparkle of thousand-fold increase in capacity dims when confronted with the possibility that data may be lost, or that data may be unavailable for long periods due to hardware failures. An unreliable storage system is a useless storage system; hence we must address reliability and availability.

We need to understand the failures of the components to understand how to create a highly available system, and we have no body of knowledge on reliability to guide us. For example, all of the following are plausible solutions, depending on the needs of the users and the weaknesses of the technology:

- Users must never lose data but can live with occasional unavailability, and the tape media is determined to be the weak link of the reliability chain. One solution is to simply make copies using the least expensive media and save the copies as off-line storage.
- Readers are the weak link in the chain, so every robot must have multiple readers.
- Robots are the weak link in the chain. One solution is to use a stacker as the building block. Each stacker has a reader, simple robot, and a limited set of tapes. The stackers are used in a parity scheme, much like disks are used in RAID.
- The error recovery scheme we invent to overcome whatever weakness are inherent in the technology is also sufficient to handle normal tape read/write errors. The ECC mechanism for tapes is simplified to offer higher tape write bandwidth (no read after write) and greater tape capacity (reduce the 25% of storage dedicated to ECC).

4. Summary and Conclusions

The goal of our research project is to understand how to structure a geographical distributed mass storage system. We are building a low latency, high capacity, network-attached mass storage system as part of our commitment to the Sequoia 2000 Earth Scientists.

Our research approach is founded on developing new techniques for managing latency, integrating compression, leveraging interleaving, providing redundancy within the storage system. Some of the latency management strategies we have discussed include access hints and caching, reduced load times on tape media, and mixing optical disk and tape within the same hierarchy.

Our methods for increasing bandwidth include compression and interleaving. With the advent of new compression hardware, we believe that it will provide an important new element of storage technology. The question is how effectively compression can be applied to scientific data sets. Interleaving has proved effective in disk array organizations. We think that the concept can be generalized to other media, I/O controllers, file servers, and network connections to scale up the bandwidth of the I/O system.

Reliability remains a major unknown for tertiary storage devices. We need a better understanding of the failure mechanisms of the various storage technologies. Tape media and read/write heads suffer from much more pronounced wear-out than comparable disk systems. By combining parallel error correction with an interleaved approach, we can obtain storage systems with much higher levels of availability.

5. References

[Exabyte 90] Exabyte Corporation, "EXB-120 Cartridge Handling Subsystem Product Specification," Part No. 510300-002, 1990.

[Spencer 88] Spencer, K., "The 60-Second Terabyte," *Canadian Research Magazine*, (June 1988).

[Tan 89] Tan, E., B. Vermeulen, "Digital Audio Tape for Data Storage," *IEEE Spectrum*, V. 26, N. 10, (October 1989), pp. 34 - 38.