Tertiary Storage: An Evaluation of New Applications

by

Ann Louise Chervenak

B.S. (University of Southern California) 1987
M.S. (University of California at Berkeley) 1990

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in Charge

Professor Randy H. Katz, Chair
Professor David A. Patterson
Professor Kyriakos Komvopoulos

1994

The dissertation of Ann Louise Chervenak is approved:

Chair Date

Date

Date

University of California at Berkeley

1994

Tertiary Storage:
An Evaluation of New Applications

by

Ann Louise Chervenak

Abstract

Tertiary Storage: An Evaluation of New Applications

by

Ann Louise Chervenak

Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Randy H. Katz, Chair

This thesis focuses on an often-neglected area of computer system design: tertiary storage. In the last decade, several advances in tertiary storage have made it of increasing interest, including increased tape capacities, less expensive tape drives and optical disk drives, and the proliferation of robots for loading tertiary devices automatically. Concurrently, faster processor speeds have enabled a growing number of applications that would benefit from fast access to massive storage. We evaluate the usefulness of current tertiary storage systems for some of these new applications.

First, we describe the design and performance of tertiary storage products. Next, we evaluate the technique of data striping in tape arrays. We find that tape striping improves the performance of sequential workloads. However, striped tape systems perform poorly for applications in which there are several non-sequential, concurrent requests active in the tape library because of contention for a small number of tape drives.

We characterize two new workloads: video-on-demand servers and digital libraries. For the former, we evaluate design alternatives for providing storage in a movies-on-demand

system. First, we study disk farms in which one movie is stored per disk. This is a simple scheme, but it wastes substantial disk bandwidth, since disks holding less popular movies are under-utilized; also, good performance requires that movies be replicated to reflect the user request pattern. Next, we examine disk farms in which movies are striped across disks, and find that striped video servers offer close to full utilization of the disks by achieving better load balancing. Finally, we evaluate the use of storage hierarchies for video service that include a tertiary library along with a disk farm. Unfortunately, we show that the performance of neither magnetic tape libraries nor optical disk jukeboxes as part of a storage hierarchy is adequate to service the predicted distribution of movie accesses.

Throughout the dissertation, we identify several desirable changes in tertiary storage systems. To support new applications with higher concurrencies, tertiary libraries should be redesigned with a higher ratio of drives to media, higher bandwidth per drive and faster access times.

Professor Randy H. Katz, Chair Date

To my parents,

Loretta and Norris Chervenak,

for their constant love and support.

Contents

List of Figures

List of Tables

Acknowledgements

It has been my privilege at Berkeley to work closely with two fine systems professors, Randy Katz and David Patterson. Randy has been an excellent advisor. His rapid flow of ideas and wry humor make him an exhilarating person to work with. Randy combines a tremendous breadth of knowledge with impressive technical depth; discussions with him invariably reveal new perspectives on problems and fresh research directions. From the first draft of my masters report to the fourth revision of my dissertation, Randy has been a careful, constructive, tough critic of my written work. Always generous in encouraging his students to attend conferences and make presentations, Randy has helped me dramatically improve my speaking skills. Throughout my graduate career, Randy has been a great source of advice regarding all aspects of graduate school, publishing papers, the job search and the career of a young professor. Over the last two years, the experiences he has shared from his time in Washington at ARPA have given me insight into the workings of government funding agencies and great respect for Randy's dedication to public service.

In the last two years, Dave Patterson assumed the role of my local advisor, and I am very grateful to him for the tremendous help he has given me. He has offered valuable technical input and encouraged me to pursue several interesting new research directions. Dave has also offered advice and support on many topics besides my dissertation research, including interviewing, effective presentations, dissertation organization and writing, the goals of a first-year professor and coping with the travails of life. Warm, funny, and generous with his time and attention, Dave communicates enthusiasm and joy in being a professor and advisor. It is very satisfying and validating for a graduate student to have an advisor who takes such evident pleasure in sharing new ideas and results. I have greatly enjoyed working with him.

I am also grateful to the third reader on my dissertation, Professor Kyriakos Komvopoulos, who offered valuable input on tribology and reliability issues for tape systems early in my research in those areas.

My last year as a graduate student has been particularly difficult, and I am deeply grateful to many people for their love, friendship and support.

God blessed me with wonderful parents. My mother has a truly unselfish heart. She has always given me unconditional love and support and is my most tireless cheerleader and my staunchest defender. She also has a keen intelligence, a deep understanding of human nature, and a compassion that allows her genuinely to forgive other people and always seek to find good in them. My father is a warm, kind, intelligent, honorable, and unfailingly honest man. He has always been quietly confident of my abilities, and over the years he has patiently offered counsel on everything from calculus to finances to relationships. I treasure his confidence in me. Together, my parents have taught me the importance of family, loyalty, honesty, compassion and forgiveness. I love them with all my heart.

I am equally blessed with wonderful siblings, my sisters, Mary and Virginia, my brother, Tim, and my brother-in-law, John. My nephew, Jack, is the most brilliant and beautiful baby imaginable. All of them have given me tireless and generous support this year, as well as the joy of spending time with them. I am especially grateful to Mary for helping me to remain sane during crazy times.

Because I have a large, warm extended family, I always feel that the world is populated with kind, loving people. I deeply miss my grandmother, a remarkable woman, but her influence is felt in the presence of her children, grandchildren and great-grandchildren. Among many wonderful aunts, uncles and cousins, I treasure my aunts, Connie and Rosalie, and my cousin Connie and her family, who have always been so loving and supportive of me.

There are several friends on whom I have leaned particularly hard in the last year, and I am very grateful to them for their support. Kim Keeton has been a generous and loyal friend, on several occasions dropping everything in her busy life to help me and always being a patient and compassionate listener. Shellie Sakamoto and John Takamoto have offered the incomparable comfort of longstanding friendship as well as the pleasure of their company and their cats to cheer me up. Janet and Peter Chen have given me many prayers, loyal friendship and invaluable spiritual counsel. John Hartman and Ken Shirriff have been so good to me, offering humor, support, and the male perspective while welcoming me into the fun, eclectic community centered at the Hillegass house. Judi Franz, Sunita Sarawagi, Hilary Kaplan, Bret Fausett, Valerie Taylor, Mary Baker and Margo Seltzer have all given me warm friendship over the years as well as much-needed support at essential moments in the last few months.

Perhaps my most valuable professional experience at Berkeley was my participation in the RAID project. I was fortunate to work with many bright systems students who became good friends. My former officemates, Peter Chen and Ed Lee, were a great pleasure to work, talk, argue and spend time with. Other good friends and valued colleagues on the project were Garth Gibson, Srini Seshan, and Ethan Miller. Additional participants who played a big part in the success of RAID were Rich Drewes, Rob Pfile, Rob Quiros and Mani Varadarajan. In addition, the overlap of the RAID, Sprite and Postgres projects introduced me to a collection of wonderful people, including John Hartman, Ken Shirriff, Mary Baker, Margo Seltzer, and Mendel Rosenblum. Besides their participation in the XPRS project and retreats, John Ousterhout was a valuable critic of my masters' project, and Mike Stonebraker was a helpful member of my qualifying exam committee. Tom Anderson offered very helpful criticism of my interview talk and gave a lot of great advice about job-hunting. Over the years, I also had a wide collection of officemates who made coming to work fun and from whom I learned a great deal. They include David Wood, Mario Silva, Tzi-cker Chiueh, Joel Fine, Remzi Arpachi, Elan Amir and Hari Balakrishnan.

Besides my advisors, the person at Berkeley from whom I learned most is Ken Lutz, the engineer responsible for making RAID work. He taught me not only how to design and

debug hardware better, but also methods for approaching problems and management skills. He generously offered advice for dealing with fellow students, professors, companies and bureaucrats. Ken approaches his work with savviness, skill, straightforward honesty and slightly cynical good humor. I can offer no better advice to young systems students than to learn all they can from Ken.

Theresa Lessard-Smith and Bob Miller, our grant administrator and assistant, play an important role in the lives of all systems students at Berkeley. They make our lives easier in so many ways and have become valued friends to me. Terry is a warm, sweet person, and I have greatly enjoyed discussing gardens and life with her. Bob, with his slightly sarcastic wit, always makes me laugh.

Being at Berkeley has been a wonderful experience for me, and I will miss this place and all the people who have been a part of it.

Chapter 1

Introduction

Just as a biologist studying animal locomotion in a jungle might ignore a lumbering elephant in favor of a speeding cheetah, so do computer systems designers largely ignore the tape and optical disk systems that store masses of digital information to concentrate on making fast processors run faster and small disks grow smaller. In this dissertation, we focus on one of the most neglected areas of computer system design, tertiary storage.

Tertiary storage includes magnetic tape and optical disk devices, as well as more exotic technologies like optical tape and holographic storage. Named because it is traditionally the third level in a computer system storage hierarchy from which data are fetched, tertiary storage is inexpensive but slow. Figure 1.1 shows a typical storage system hierarchy. At the top of the hierarchy is *primary storage*: random access memory (RAM) used for caches and main memory. Typical RAM technologies are static (SRAM) and dynamic (DRAM). Below RAM is solid state memory and then magnetic disk devices, commonly called *secondary storage*. At the bottom of the hierarchy are *tertiary storage* devices: magnetic tape and optical disk. In going from the top of the hierarchy to the bottom, average access times increase dramatically from tens of nanoseconds for DRAM to tens of milliseconds for magnetic disks, tens of seconds for optical disk jukeboxes and several minutes for tape

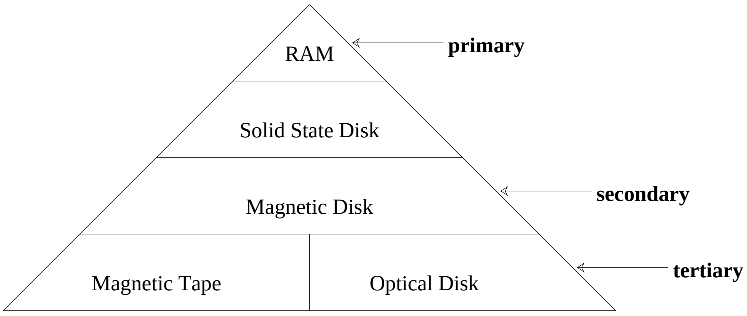


Figure 1.1: *The storage hierarchy shows a traditional classification of devices for computer data storage.*

libraries. Descending the pyramid also results in dramatically decreased cost per megabyte of storage. The size of the pyramid blocks suggests that computer systems will generally contain small amounts of more expensive technologies like RAM and larger amounts of less expensive storage like tape.

Magnetic tape devices have long been an important component of storage systems, but access to data on tape is notoriously slow. Loading a tape has historically required human intervention; accessing information stored in an archive might take hours or days. As a result, data were typically sent to tertiary storage only if data sets were too large to be stored on disk or if the data were unlikely to be accessed again. Thus, the main applications that use tertiary storage have long been backups of file systems, archives of large databases, and staging operations that move large scientific data sets onto and off of disks for supercomputer computation. The resulting workload to the tertiary storage system consists mostly of sequential write operations for backup and archival applications; for staging data onto disk systems, the tertiary system also performs large read operations [22], [29], [39], [49], [72], [110], [114], [113], [63], [64], [75], [51], [70], [70], [89], [16], [23]. There is generally a single process reading or writing tertiary storage at any time; since the tertiary storage system has long been off-line, the fast response times required for interactive operation are not assumed by the applications.

In the last decade, several advances in tertiary storage technology have made it of increasing interest to computer system designers. First, increases in bits-per-inch and tracks-per-inch densities have increased tape capacity dramatically. Second, a variety of inexpensive tape drives has become available. Next, magnetic tape and optical disk media are inexpensive compared to magnetic disk; typical tape cost is \$.005 per megabyte compared to \$.50 per megabyte for magnetic disk. The low cost of storage makes it economical to build massive tertiary storage systems. Fourth, optical disk technology, and in particular CD-ROM technology, have emerged as a popular, convenient, and inexpensive way to distribute information. Fifth, a large number of robotic devices for handling both magnetic tape and optical disk allow access to tertiary storage without human intervention, making response times more predictable. Robots make it possible to consider the tertiary storage system as being nearly on-line. Finally, increases in computer processing speed have enabled a growing number of applications, ranging from climate modeling and geographic information systems to digital libraries and multimedia servers, that would benefit from fast access to a massive storage system.

In this dissertation, we evaluate how well tertiary storage systems succeed in supporting some of these new applications. These applications will have very different workloads from traditional backup and archival applications. We demonstrate that digital libraries and video servers, for example, will have a high concurrency of active requests, will have fairly strict response time requirements, and will not necessarily make sequential requests for data. Tertiary storage systems cannot be expected to replace magnetic disks; rather, we investigate whether tape libraries and optical disk jukeboxes perform well enough that they can become an effective part of a storage hierarchy servicing these new applications. In applications like video service and digital libraries, we assume that a magnetic disk system would serve as a cache for the most popular data, while the tertiary storage system would service infrequent accesses to less popular data.

Unfortunately, we find that despite the advantage of inexpensive mass storage, the promise of current tertiary storage systems is not fulfilled because of their poor performance. Bandwidth of tape drive and optical disk drives is quite low, and access times are on average minutes and tens of seconds, respectively, compared to milliseconds for accesses to magnetic disk. In addition, tape libraries and optical disk jukeboxes are designed for traditional, archival applications, with a large number of tapes or platters and a handful of drives; this configuration does not support the high concurrency and fast access required by the new applications. After showing the inadequacy of current tertiary systems, we evaluate several feasible technology improvements. We show that increased drive bandwidth, lower latency for tape operations, and a higher proportion of drives to media within robots are essential to supporting workloads with high concurrency and strict response time constraints.

1.1 Contributions

The contributions of this dissertation are as follows:

- We present the first extensive workload analysis of large-scale tertiary storage systems used for traditional backup and archival workloads as well as new kinds of multimedia applications. We describe the basic operation of tertiary devices, including the performance of existing products on a variety of workloads.

- We find that the technique of data striping, which was used so successfully in disk arrays to increase the bandwidth and reduce the latency of large accesses, is only effective in tape libraries for a limited class of applications. Striping is effective for workloads that have mainly sequential accesses or low concurrency. For a greater number of outstanding requests to more randomly distributed data, striping the tape library is detrimental to performance. This poor performance is caused by increased contention for a limited number of tape drives in a typical library.

- We characterize two new workloads: video-on-demand service and digital libraries.

- In a movies-on-demand video server application, we predict that accesses to movies will be highly localized, potentially making the video server a good candidate for a storage hierarchy in which tertiary storage services accesses to the least popular movies. Unfortunately, we find that tape libraries and optical disk jukeboxes don't perform adequately as part of a storage hierarchy to service the predicted workload. The small number of tape or optical disk drives in a typical tertiary library and their low bandwidth make tertiary storage systems unable to service more than a small number of concurrent video streams.
- We evaluate the use of disk farms for movies-on-demand video service. We show that disk systems are much more effective at supporting a large number of video streams than are storage hierarchies that include tape or optical disk. Among disk systems, we show the advantage of using striping to use bandwidth effectively, thus supporting more video streams with the same I/O hardware.
- We present a list of desired improvements in magnetic tape and optical disk systems.

1.2 Methodology

The methodology embodied in this dissertation first uses measurements of real devices to derive models of their behavior. Then we incorporate these models into performance simulators (Chapters 3, 7). We also propose workload models for various applications that are used to drive simulations (Chapters 4, 5, 6). Finally, we use simulation results to analyze design tradeoffs for different storage system configurations and to make recommendations intended to influence the design of future tertiary storage systems (Chapters 4, 5, 7).

1.3 Thesis Outline

This thesis is composed of eight chapters. Chapter 2 describes the technology of magnetic tape and optical disk devices. It surveys existing drive and robot products and discusses tradeoffs in the design of tertiary systems. Chapter 3 describes the event-driven tertiary storage simulator used to generate many of the results in this thesis. The chapter also describes our measurements of tape drive, tape robot and optical disk devices, and presents the performance models used in our simulations. The fourth chapter is a general study of tape library and optical jukebox performance. It presents simulation results for a variety of workloads including traditional archival applications and more demanding multimedia applications. Chapter 5 presents a study on the usefulness of striping in tape libraries. It shows that striping is very effective in increasing the bandwidth of sequential workloads, but that when accesses are more random in nature or the concurrency of the workload is high, striping performs poorly.

Next, Chapter 6 presents an initial characterization of two emerging applications: video-on-demand and digital libraries. We characterize typical access sizes, locality patterns, response time constraints, and loads. The seventh chapter evaluates storage systems to support video-on-demand service. We compare storage systems composed entirely of disks to storage hierarchies that include magnetic disk and tertiary storage. For disks, we compare non-striped and striped systems and find that striped video servers support many more viewers. We also find that storage hierarchies perform poorly for this application because of the long access times and low bandwidth of tertiary libraries. The dissertation ends with concluding remarks and a bibliography.

Chapter 2

Tertiary Storage Technologies

2.1 Introduction

In the last chapter, we described a typical computer system storage hierarchy. In this chapter, we describe the workings of tertiary storage, the lowest level of the hierarchy. In Section 2.2, we discuss magnetic tape drive technology, including tradeoffs in the design of tape systems. In Section 2.3, we do the same for optical disk technology. We give examples of both magnetic tape and optical disk drive and robot products and describe technology trends. We finish the chapter with a discussion of two more exotic tertiary storage technologies: holographic storage and optical tape.

2.2 Magnetic Tape Technology

2.2.1 The Tape Drive

Magnetic tape drives [4], [34], [38], [40], [43], [48], [58], [59], [66], [96], [97], [107], [118] store data as small magnetized regions on a tape composed of magnetic material deposited on a thin, flexible substrate such as plastic. Magnetic tapes are written by inducing a current proportional to an input signal in the coil of an inductive write head;

this current induces a magnetic field in the tape below the write head that magnetizes a small tape region. After data have been written, a read head can detect magnetic flux on the tape; the rate of change of this flux results in a voltage on the read head that is used to deduce the original input signal [59] [96]. A tape drive mechanism consists of reels, motors, gears, brakes and belts that thread the tape, control tape reeling, guide the tape to maintain its position relative to the heads, control the acceleration and steady-state velocity of the tape, and control tape tension [66].

The substrate for magnetic tapes is a poly-ethylene terepthalate (PET) base film from 12 to 36 microns thick [13]. There is a tradeoff between reliability and capacity in choosing the thickness of the substrate. A thicker substrate is more durable and able to withstand high accelerations and start and stop operations without excessive distortions [66]. A thinner substrate allows a single reel to store more tape, with greater resulting capacity.

On top of the substrate is the magnetic layer that stores data. Ideally, this magnetic layer should be very smooth to allow the minimum spacing, or air bearing, between the read/write head and the tape; the smaller this air bearing, the greater the density of magnetic information that can be stored on the tape. (Typically, the separation between the head and moving tape is approximately 0.2 microns [66].) There are two types of magnetic layers: Magnetic Particle (MP) and Metal Evaporated (ME). Historically, most tape systems have used magnetic particle tapes in which the PET substrate is covered with a particulate magnetic coating that is 2 to 5 microns thick. The coating contains magnetic particles such as CrO_2 or Fe_2O_3, polymeric binders and lubricants. In the last five years, metal evaporated tape systems have overcome earlier difficulties with corrosion and tribology (the head/tape interface) and now offer superior recording qualities in many tape products. Metal evaporated tapes have a thin film of metal (usually Co-Ni) evaporated onto the PET substrate in a layer about 100 nanometers thick. Because the thin

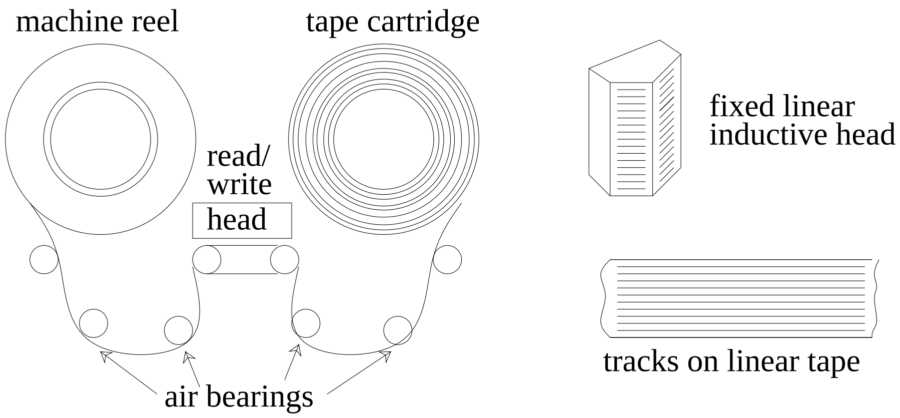


Figure 2.1: *A Linear Recording Magnetic Tape Drive*

film magnetic layer is much thinner than a particulate magnetic coating, thin film tapes hold greater capacity. In addition, thin film media have recording characteristics superior to those of particulate media, including higher effective magnetization, lower error rates, greater smoothness, highly isolated asperities, and better signal-to-noise ratios [59], [82], [13].

Besides the plastic substrate and the magnetic coating, most magnetic tapes have a back coating for protection against static. Electrostatic attraction can lure and embed dust particles from the atmosphere into the tape, causing read and write errors.

Figures 2.1 and 2.2 show the two most common tape formats and drive mechanisms for magnetic recording of computer data: linear and helical-scan.

In linear recording magnetic tape drives, data are written in tracks parallel to the edges of the tape. Many linear recording devices use stationary, multi-track heads to read or write the entire width of the tape simultaneously. Figure 2.1 shows a simplified drawing of a recording head for a linear tape drive. The recording head is made up of a coil and a gapped magnetic structure that intensifies and localizes the magnetic field during writing [96]. The read and write elements are of different size, orientation and composition; the read elements are magnetoresistive, while the write elements are inductive. Before writing the tape, a separate erase head erases the tape with a magnetic field. The read/write head

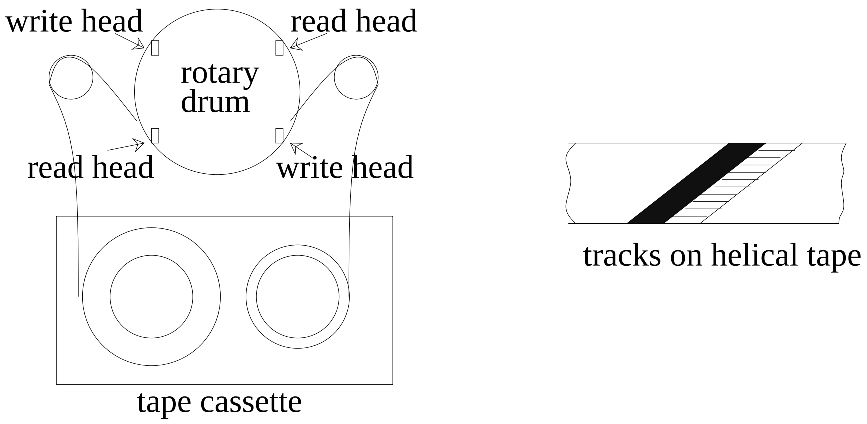


Figure 2.2: *A Helical-Scan Magnetic Tape Drive*

is narrower than the erase head, so when new data are written, gaps or “write guard bands” are left between adjacent tracks to minimize cross talk. Further, the read elements on the head are narrower than the write elements, creating a “read guard band” to further reduce the risk of reading erroneous information. Data are read immediately after being written to check for errors and rewritten if errors occur. In older linear tape drives, the write guard bands or separations between adjacent tracks are large; as a result, the areal density of these drives is relatively low compared to helical scan tapes. More recent linear tape drives have reduced or eliminated the separation between tracks and greatly increased the areal density. Another type of linear recording drive is a serpentine drive that reads or writes a few tracks at a time down the entire length of the tape; then tape motion switches directions, and the heads themselves are shifted to read or write a different set of tracks.

Figure 2.2 shows a helical scan tape drive. Helical scan tape drives achieve high areal densities and, potentially, high data rates [10] [74] [100] [108]. Read and write heads are situated on a drum that rotates at high speed (around 2000 r.p.m.). Tracks are written at a small angle (10 to 20 degrees) with respect to the direction of tape movement onto a slow-moving (around 1/2-inch per second) magnetic tape. On some drives, a stationary erase head erases old data before writes occur. In addition, helical scan drives usually contain a *servo* head that can read servo information written at the edges of the tracks.

The presence of the servo head makes it possible for the heads to be positioned precisely, so that gaps between adjacent tracks can be small. If two sets of read/write heads are set at opposite angles, then two tracks can be written simultaneously. These adjacent tracks may overlap slightly, writing a herringbone pattern as shown in Figure 2.2. Using different write angles on adjacent tracks minimizes crosstalk [108]. Because adjacent-track gaps are small or nonexistent in helical scan drives, areal density on these drives is high. As for linear tape drives, in helical-scan drives, data are read immediately after being written; the drive continues to write the data in subsequent locations until the write completes without errors.

There is substantial controversy over the relative merits of linear and helical scan technologies. Stationary head, linear drives are thought to have a less abrasive head-to-tape interface than helical scan systems, resulting in longer-lived heads and tapes that tolerate more recording and play passes. Helical, rotating-head systems boast a relatively slow normal tape speed that allows substantial speedup during fast search and rewind operations; for example, the 4mm DAT drives have a fast positioning operation that is 200 times faster than the normal tape speed. By contrast, the tape typically moves quickly in a linear drive, since the tape speed along with the track density and number of heads determines the data rate of the tape drive. In addition, Bhushan argues that helical systems potentially have higher data rates, since it should be easier to speed up the rotating drum in a helical tape drive than to quickly move heavy rolls of tape in a linear drive [13].

Even among helical scan drives, there is controversy over which design is superior. For example, the "wrap angle," or amount of tape that is wrapped around the rotating drum that contains the recording heads, is 90 degrees for a 4mm DAT drive and 220 degrees for an 8mm drive. Advocates of 4mm technology claim the shorter wrap angle is less stressful, reducing wear and improving access times [74], while 8mm proponents claim their longer wrap angle is actually less stressful, providing better tape guidance [10].

Several features of magnetic tape drives are notable. Most tapes are append-only, although a few update-in-place magnetic tape drives do exist [108]. When data are written on any portion of an append-only tape, whatever had been written beyond the new write point is irretrievable. There are several reasons for the append-only nature of tapes. First, tape mechanisms are not well-suited for the large numbers of forward and reverse positioning operations that are typical of update-in-place devices like magnetic disk drives. The mechanical stop and start operations are time-consuming and cause wear on the tape, so they are avoided. Instead, applications strive to use tape drives in a *streaming* mode of operation in which data are sent at a constant rate to or from the tape head. Second, update-in-place is difficult because the high rate of errors on the media make it difficult to predict exactly where data will lie on the tape. When a tape drive writes data, it immediately reads back what it has written to check for correctness. When there is an error, the tape drive re-writes the data at a later position on the tape, repeating the read-after-write checking until the data are written without errors. Unlike magnetic disks, where data are mapped into particular sectors at predictable locations on a disk platter, it is impossible to know before a write operation exactly where data will physically reside on a magnetic tape.

There have been a few update-in-place magnetic tape drives, notably the recent Data/DAT format for 4mm tape drives [108]. This format has not found favor compared to its more traditional, append-only competing format, DST. Update-in-place tape drives work by dividing the tape into zones or sectors, leaving wide buffers of empty tape between sectors to allow data to be rewritten if write errors occur. These buffer zones must also be large to protect good data from the tape drive's large erase head. Because of these large buffer zones, update-in-place tapes have lower storage capacity than append-only tapes.

Many tape drives require that tape cartridges be rewound before they can be ejected from a tape drive. There are several reasons for this requirement. Some tapes are stored on a single tape reel and must be rewound before they can be physically removed. Even for double-reel cassettes, rewind is often required because there may be servo information, used for precise positioning, at the start of the tape that must be read on insertion. Also, because inserting and ejecting a tape can cause considerable wear, some manufacturers prefer that those operations occur on a strip of tape where no data are stored. Some tape drives include load/eject "zones" scattered throughout the tape; these zones contain servo information to allow tapes to be inserted at those points without rewinding to the start of tape. Finally, keeping tapes in streaming operation is essential to achieving good performance. In order to minimize tape and head wear, tape drives will release tension on the tape if there is a lapse of a certain length of time after an operation; if the delay goes on even longer, the drum in a helical scan tape drive will stop spinning. Subsequent requests will have to wait for the drum to spin up and for tape tension to be reapplied. To avoid these delays that reduce throughput, the drive should operate in streaming mode.

2.2.2 Representative Products

There are a number of magnetic tape products available that provide a variety of tape capacities and speeds for a range of prices. Table 2.1 compares several tape drives based on cost, cartridge capacity and data transfer rate [1], [10], [62], [74], [119], [77], [100], [108], [115], [93], [11]. At the low end, an inexpensive 4mm DAT drive stores several gigabytes of data but has a low transfer rate of 366 kilobytes/second. At the high end, the very expensive Sony D-1 drive stores about 25 times the capacity stored by the DAT drive and transfers data at approximately 100 times the transfer rate of the DAT drive.

2.2.3 Tape Robots

To provide higher bandwidth and capacity than can be supplied by a single device, several companies have built automated library systems. These libraries hold tens, hundreds

Tape Drive	Price ($)	Media Cost ($/MB)	Capacity (GBytes)	Data Transfer Rate	Drive Type
Sony SDT-4000	$ 1,695	$0.015	4	183-366 KB/sec	4mm helical
Exabyte EXB8500	$ 2,315	$0.008	5	500 KB/sec	8mm helical
Storage Tek 4220	$ 19,000	$0.013	0.2-0.8	712 KB/sec	1/2-inch linear
Metrum 2150	$ 32,900	$0.0013	18	2 MB/sec	1/2-inch helical
Ampex D-2	$100,000	$0.0018	25	15 MB/sec	19mm helical
Sony D-1	$270,000	$0.0016	90	32 MB/sec	19mm helical

Table 2.1: *Price, cartridge capacity and data transfer rates of a variety of magnetic tape products. Prices from January, 1994 issue of* **SunWorld** *magazine.*

or thousands of cartridges that can be loaded by robot arms into a collection of magnetic tape drives.

Table 2.2 shows a classification of some of the robots available for handling magnetic tape cartridges automatically [62], [52]. Table 2.3 describes examples of each type of robot. Large libraries generally contain many cartridges, several drives and one or two robot arms for handling cartridges. The cartridges are often arranged in a rectangular array. Other "large library" configurations include a hexagonal "silo" with cartridges and drives along the walls, and a library consisting of several cylindrical columns holding cartridges that rotate to position them. Usually these large libraries are quite expensive ($500,000 or more), but they often have lower cost per megabyte than less expensive robotic devices. One disadvantage of large tape libraries is the low ratio of drives and robot arms to cartridges. In a heavily-loaded system, there may be contention for both arms and drives.

Carousel devices are moderately priced (around $40,000) and hold around 50 cartridges. The carousel rotates to position the cartridge over a drive, and a robot arm pushes the cartridge into the drive. In most cases, there are one or two drives per carousel.

Finally, the least expensive robotic device ($10,000 or less) is a stacker, which holds approximately 10 cartridges in a magazine and loads a single drive. The magazine may move vertically or horizontally to position a tape in front of the drive slot, or the stacker

Type	No. Cartridges	No. Drives	No. Robot Arms	Cost
Large Library	10 to 1000	several	one or two	high
Carousel	around 50	one or two	one (carousel)	moderate
Stacker	around 10	one	one (magazine or arm)	low

Table 2.2: *Classification of tape robots.*

Tape Library	Drives	Tapes	Total Capacity	Price ($)	Tape Drive Technology	Robot Type
ADIC DAT Autochanger 1200C	1	12	24 GB	$ 8,900	4mm helical	stacker
Exabyte EXB10i	1	10	50 GB	$ 13,000	8mm helical	stacker
Spectralogic STL800 Carousel	2	40	200 GB	$ 31,465	8mm helical	carousel
Exabyte EXB120	4	116	580 GB	$ 100,000	8mm helical	library
Metrum RSS600	5	600	10.8 TB	$ 265,900	1/2-in helical	library
Ampex DST600	4	256	6.25 TB	$1,000,000	19mm helical	library

Table 2.3: *Comparison of several tape robots. Prices from January, 1994 issue of* **SunWorld** *magazine.*

may have a robot arm that moves across the magazine to pick a cartridge. A storage system composed of stackers would have the highest ratio of robot arms and drives to cartridges.

Robot access times are fairly short compared to tape positioning operations like search and rewind. A tape switch operation, which replaces a tape loaded in a tape drive with a new tape from a shelf, involves the use of the robot arm. In the next chapter, we show that a tape switch operation may take several minutes. The tape switch may first require rewinding the currently-loaded tape. Next, that tape must be physically ejected from the tape drive. The robot arm moves to unload the old tape and load a new one. Then the tape drive physically loads the new tape, including reading servo information at the start of the tape. The tape drive performs a forward search operation to position the tape. Finally the tape drive performs the read or write data transfer operation. The robot contribution to the tape switch time is between 5 and 40 seconds on typical robots.

Drive	Corrected Bit Error Rate
DAT	10^{-15}
Exabyte EXB8500	10^{-13}
1/4”	10^{-14}
Metrum 1/2”	10^{-13}
19mm D-2	10^{-12}

Table 2.4: *Compares the error rates per bits read or written for a variety of tape drives.*

2.2.4 Reliability of Tape Drives and Media

Magnetic tape systems face difficult reliability challenges [9]. Tape wear, head wear and rates of errors uncorrectable by error correction codes (ECC) may make errors more frequent in large tape systems than in disk arrays. It is an open research question how much error correction will be required to make tape arrays adequately reliable.

Tape Media Reliability

The rate of raw errors (i.e., errors before any error correction has been performed) is quite high on magnetic tape media: approximately one error in every 10^5 bits read. Most of these errors are caused by “dropouts,” in which the signal being sensed by the tape head drops below a readable value. Dropouts are most commonly caused by protrusions on the tape surface that temporarily increase the separation between the head and the tape, causing a drop in signal intensity [66]. There are several ways that debris can become embedded in the tape and cause dropouts. When the tape is originally sliced, loose pieces of tape substrate and coating are left at the edges of the tape; these may become embedded in the tape surface. Second, as the tape passes through the drive mechanism, it can become charged and attract particles in the atmosphere. Another way debris accumulate on the tape is through the wear that occurs when the tape drive starts and stops. At very low velocities, the separation between the head and tape surface normally caused by air flow is not maintained. The head contacts the tape coating, producing a fine, dry powder. Pushed by the head, this powder can accumulate and cause dropouts. Dropouts are also caused by inhomogeneities in the tape's magnetic coating.

Because of the high raw bit error rates on all magnetic tape devices, all drives incorporate large amounts of error-correction code. However, some errors will occur that the ECC cannot correct. The rate of such errors is called the corrected or uncorrectable bit error rate. Table 2.4 shows that uncorrectable bit error rates for the different tape technologies vary from one error in every 10^{12} bits read to one in every 10^{15} bits read. Taking the Exabyte 8mm drive as an example, suppose a small tape library contains eight Exabyte drives that each transfer data at 500 KBytes/sec. Assuming that the drives operated on just a 10% duty cycle, an uncorrectable bit error will occur on average every 36.2 days.

It is notable that one of the most expensive tape drives, the Ampex DST800, has one of the highest error rates, while the inexpensive DAT drive has the lowest rate. The DAT drive achieves this low error rate by including 3 layers of Reed-Solomon error correction. The extensive encoding required to perform this error correction consumes a great deal of capacity. For example, the first two ECC layers consume 30% of the bits available for data storage; depending on the amount of encoding performed for the third ECC level, another 4% to 18% of tape capacity may be consumed [108]. Thus, there is a tradeoff between data reliability and tape capacity available for data storage. Incorporating the third ECC level is also expensive, requiring more complex electronics and larger buffers. DAT manufacturers recommend the use of the third level of ECC only for sensitive data.

Magnetic tapes that are frequently read or written eventually wear out. In a traditional archival system, where data are written and probably never read again, this is not a serious concern. However, in applications like digital libraries, there is no limit on the number of times a tape may be read. Tapes last on average several hundred passes [13], [47], [60]. However, they wear out sooner if a particular segment of the tape is accessed repeatedly. A Hitachi study of DAT tape drives showed that the raw error rate (before error correction) after 2500 passes to a single segment of a tape was over one error in ten ECC blocks read [31]. Tapes written by linear recording drives do not suffer so quickly from tape wear-out as tapes written by helical scan drives because the interface with the head is less abrasive, but wear is still a concern. In large tape libraries, controllers will be required to monitor the number of passes to a tape cartridge and replace it before wearout occurs.

In an interactive library application, wear due to stops and starts on the tape is likely to be severe, since accesses to the library will not be sequential. Severe wear is manifested by large portions of the magnetic binding material flaking away from the tape backing. Such problems make large sections of a tape unreadable.

Another set of reliability concerns involves the long-term storage of data on tape. Over time, the metal pigments in tape are subject to corrosion; this problem is eliminated to a large extent by the use of appropriate binders. The tape also undergoes mechanical changes including tape shrinkage, creasing of the edges, peeling of the magnetic layer, and deterioration of surface smoothness [47]. Back coating transfer can also occur, in which the magnetic coating and the back coating from adjacent tape layers are pressed together; when shrinkage occurs during storage, the roughness of the back coating can transfer onto the magnetic layer and cause a deterioration in tape smoothness [47]. Many manufacturers recommend rewinding tapes every 6 months to avoid such problems. Finally, the proliferation of incompatible recording formats threatens the future accessibility of archival data [60].

Tape Head Wear

Tape heads undergo considerable wear in all tape systems. In helical scan systems, they last for a few thousand hours of actual contact between the head and medium. Linear tapes are thought to produce less wear because the interface between the tape and the head is less abrasive. Some tape wear is necessary in order to keep the heads in optimum condition [67]. Tape wear helps remove particles from the head that may have been transferred

Repair Type	%
Replace heads	44
Tape mechanism (reel motors, tape tension, etc.)	21
Card failure	17
Other (firmware, power supply, etc.)	14
No defect found	4

Table 2.5: *1991 repair statistics for Exabyte 8mm drives. (Source: Megatape)*

there from the tape surface or the atmosphere, or that came from the tape coating under conditions of friction or extremely high or low humidity. All tape drive manufacturers recommend periodic use of a cleaning cartridge to remove debris from the tape head. Extensive wear occurs when new tapes are used, since new tapes tend to have a lot of surface debris that is removed by the heads during the first few passes. One way to extend the life of drive heads is to use burnished tapes, from which much of the surface debris has been removed.

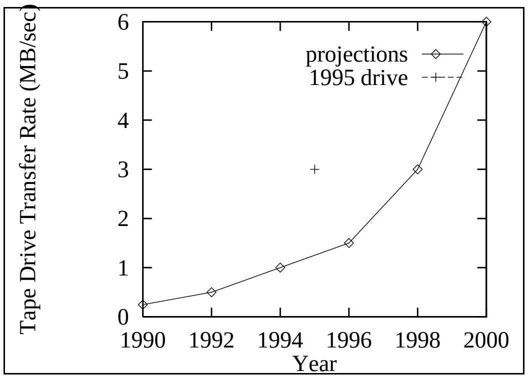
Eventually, the head wear becomes extreme. Tape library controllers will need to schedule both cleaning and replacement of the heads to assure adequate reliability. This may require keeping statistics on how many hours particular drives have run.

Mechanical Reliability

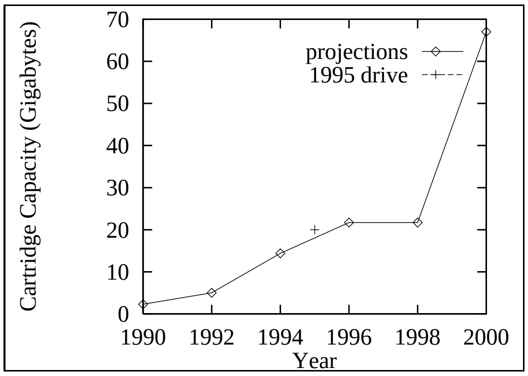
Head failure is the main cause of tape drive failure; however, the drive may also have other mechanical or electrical failures. Figure 2.5 shows repair statistics for Megatape, an OEM of Exabyte 8mm drives. 44% of the time, tape drive failures were due to failed heads. 21% of the time, some other component of the tape drive mechanism failed. 17% of the time a failure with the electronics was to blame, while 14% of the time other components such as power supplies caused the failures.

2.2.5 Trends in Magnetic Tape Drives

Magnetic tapes follow the same technology curves [57], [50], [73] as magnetic disks since the magnetic material, whether deposited on a hard disk or a flexible tape, is much the



2.3a: *Predictions for Bandwidth Improvements.*



2.3b: *Predictions for Capacity Improvements.*

Figure 2.3: *Predictions made in 1990 for bandwidth and capacity improvements for Exabyte 8mm tape drives in this decade. Improvements required to reach these goals include increased track density, decreased track width and pitch, reduced tape thickness and increased rotor speed. Source: Harry Hinz, Exabyte Corporation. The crosses in each graph show the bandwidth and capacity, respectively, of the drive to be introduced in 1995, which exceeds the 1990 projections.*

same. Currently, magnetic disk capacity is increasing at a rate of over 50% per year [79], and magnetic tapes should increase in capacity at a similar rate. In 1990, Hinz [37] predicted the growth shown in Figure 2.3 for tape capacity and data transfer rate for 8mm tape drives in this decade. The figure shows both capacity and throughput doubling approximately every two years, reaching 67 GBytes per tape and 6 MBytes/sec by the end of the decade. The generation of Exabyte tape drives being introduced in 1995 will exceed Hinz's predictions for that year; each cartridge will hold 20 gigabytes of storage and transfer data at 3 megabytes per second vs. Hinz' prediction of 14.4 gigabytes and 1 megabyte per second for 1994 drives. The data points for the new drive are marked by crosses in Figure 2.3.

Besides data transfer rate, other components of tape drive access time are improving. Several tape drive manufacturers are reducing rewind and search times by implementing periodic zones on the tapes where eject and load operations operations are allowed, rather than requiring that a tape must always be ejected and loaded at the start of the tape [69]. Mechanical operations like load, eject and robot grab and insert will be substantially faster

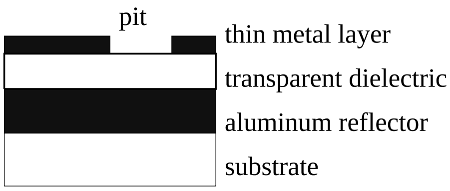


Figure 2.4: *Optical disk structure*

in the next generation of tape drives.

2.3 Optical Disk Technology

2.3.1 Optical Disks

In optical recording, a non-contact optical head uses a laser beam to store information on the disk surface by creating "pits" in the surface material [7], [120], [98], [83]. Optical disks can only be written once and are often called WORM (write once, read many) devices. Figure 2.4 shows a typical optical disk structure; a substrate (often plastic) is covered with an aluminum reflective layer, a transparent dielectric, and finally a thin layer of metal. During writing, in response to an electrical input signal, a highly-focused laser beam can melt a small region of the metal layer, opening a hole or "pit." Later, during reading, a lower-intensity, unmodulated laser beam is reflected off the surface of the disk. A photodetector interprets information stored on the disk by detecting differences in reflectivity between pits and the surface of the thin metal layer, called the "land." Data are encoded and stored as alternating regions of pits and land. The encoding may be a simple as a pit representing a zero and the land a one. In CD-ROM disks, a more complicated encoding scheme minimizes the number of consecutive zero or one bits to minimize intersymbol interference on the disk surface [83].

On optical disks, data may be stored in a single spiral track as in a CD-ROM or as a series of concentric circular tracks. Figure 2.5 shows a simplified diagram of an optical

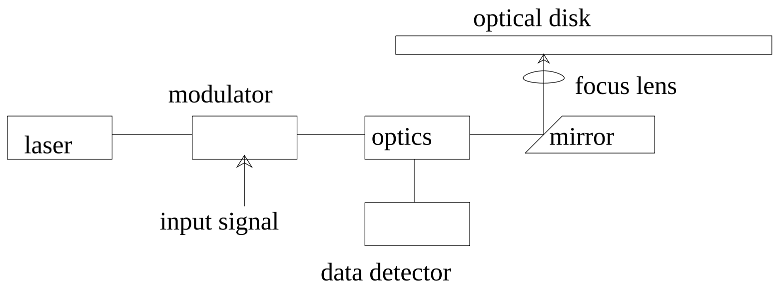


Figure 2.5: *Typical components of an optical disk*

disk. The optical disk uses several servos, one for tracking, one for focusing and one to control the rotation speed so that the data rate is constant. As the read/write head moves from the center to the outer edge of the disk, the rate of rotation decreases to maintain a constant linear velocity.

Defects in disk material may have several causes. Dust particles may accumulate over time on the disk surface; this effect can be minimized using overcoats to protect the disk surface. Unevenness in evaporated coatings may also cause defects. Finally, intrinsic defects in the substrate disks can be minimized by pre-coating the substrate with a thin layer of plastic.

2.3.2 Magneto-optical Recording

Magneto-optical disks use lasers in a process called thermomagnetic recording to store and read information on magnetic material [68], [33]. An optical recording head that has no direct contact with the recording medium contains a diode laser. During a write operation, the laser heats a spot on the magnetic material above a critical temperature and then allows the spot to cool while a magnetic field is applied perpendicular to the disk surface. The direction of the magnetic field determines the direction of magnetization in the recording; one direction is used to record a “zero” value, and the opposite, a “one” value. Figure 2.6 shows a simplified diagram of the write mechanism. The magnetic field is generated using a bias coil. The laser beam is focused through a lens and polarizer; a beam

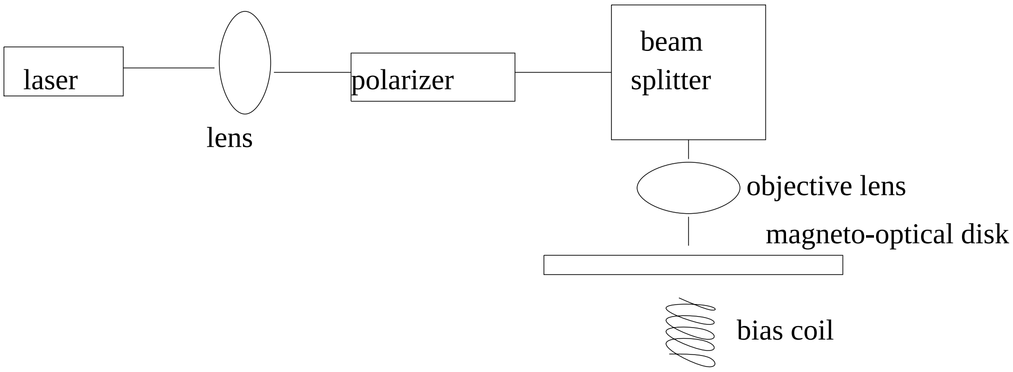


Figure 2.6: *Simplified diagram of magneto-optical disk.*

splitter separates components of the beam, using the main beam for writing and reading and the other beams for tracking. Because bits are stored on the disk surface in areas of up and down magnetization, the disk can be re-written by repeating the write procedure after a separate erase phase.

Reading the bits off the disk is accomplished by reflecting a linearly-polarized light beam off the magnetic material. An optical principle known as the Kerr effect states that linearly polarized light reflected from a vertically-magnetized material will undergo a rotation in the plane of polarization of the light. The direction of the rotation depends on the direction of magnetization of the magnetic material; by detecting the change in rotation of the light, the magnetization of the material and hence the values of the bits stored are deduced. The read process uses a low-power laser beam that does not heat the magnetic material to its critical temperature, so data stored on the disk are not corrupted during the read process.

The magneto-optical disk is a molded plastic disk onto which a magneto-optical magnetic film is deposited using sputtering or evaporation techniques. The magneto-optic film is approximately 0.1 microns thick and is covered with a protective coating. The film is usually an alloy of rare earth and transition metals, such as gadolinium terbium cobalt (GdTbCo) or gadolinium terbium iron (GdTbFe). The principle method of degradation of

Optical Disk Drive	Price ($)	Capacity (megabytes)	Data Transfer Rate (Read)	Drive Type
Sony CDU-541 (CD-ROM)	$ 420	680	150 KB/sec	optical
Sony CDU-561 (CD-ROM)	$ 460	680	300 KB/sec	optical
Hitachi internal CD-ROM	$ 785	682	153.6 KB/sec	optical
Panasonic LF-3600	$1795	128	0.61 MB/sec	magneto-optical
Pioneer DE-UH 7101	$2400	645	635 KB/sec	magneto-optical
Panasonic LF-5014	$2995	940	0.625 MB/sec	optical
Maxoptix 3-1300	$3485	1330	3 MB/sec	magneto-optical
Hewlett-Packard C1617C	$3900	650	1 MB/sec	magneto-optical

Table 2.6: *Price, capacity and data transfer characteristics of a variety of optical and magneto-optical disk devices. The Panasonic drive is 3.5 inches in diameter; all the other drives are 5.25-inch disks. Prices from January, 1994 issue of* **SunWorld** *magazine and June, 1992 issue of* **SunExpert**.

the material is oxidation, which can be minimized using over- and undercoats of protective material; these coatings also have the advantage of enhancing the signal-to-noise ratio of the data [68]. Another reliability problem with the medium is the existence and growth of microscopic defects. It is estimated that the raw bit error rates (before error correction) are about one in 10,000 bits. Lifetime of the material is estimated to be at least 8 years.

2.3.3 Representative Optical and Magneto-optical Drive Products

Table 2.6 compares a number of optical and magneto-optical disk drives [65], [62]. The range of capacity and performance is narrower for optical disks than for magnetic tape drives. The low-end CD-ROM drives are priced at about 10% the cost of a high-end drive, have similar capacity per platter and data rates that are lower but on the same order of magnitude as the high-end drives. The table shows examples of normal and double-speed CD-ROM drives; there are also triple- and quad-speed drives available.

Media costs for optical disks vary widely. For example, a 650 megabyte magneto-optical platter for the Hewlett-Packard C1617C drive costs approximately $185.00 [112], or $0.28 per megabyte. Since CDROMs are read-only media, their cost may be determined largely by the cost of their contents; for the purposes of comparison, we use the cost of

Optical Jukebox	Drives	Platters	Total Capacity	Cost ($)	Drive Type
Pioneer	1	6	4 GB	$ 1,845	optical
Ricoh	1	5	3.25 GB	$ 8,990	optical
Hewlett-Packard 20LT	1	16	20 GB	$ 9,495	magneto-optical
Hewlett-Packard 40LT	1	32	41.6 GB	$ 22,500	magneto-optical
Plasmon FR50JM	2	50	50 GB	$ 36,500	magneto-optical
Hewlett-Packard 120LT	4	88	114 GB	$ 52,500	magneto-optical
Hewlett-Packard 200LT	4	144	187 GB	$ 67,700	magneto-optical
Plasmon FR495JM	4	495	495 GB	$129,500	magneto-optical

Table 2.7: *Comparison of several optical and magneto-optical robot devices. All the optical disk drives used in these robots are 5.25-inches in diameter. Prices from January, 1994 issue of* **SunWorld** *magazine and June, 1992 issue of* **SunExpert**.

manufacture, about $2000 for 1000 platters containing 680 megabytes each [5], or $0.003 per megabyte.

2.3.4 Optical Disk Jukeboxes

Table 2.7 compares a number of robotic devices or jukeboxes that automatically load optical or magneto-optical platters into disk drives [65], [62], [25], [105]. The jukeboxes range from inexpensive autochangers for CD-ROMs to large jukeboxes with hundreds of platters. Like tape robots, large optical disk jukeboxes have a large number of platters and a small number of disk drives. Usually a single robot arm loads all the drives, and robot operations take several seconds.

2.3.5 Trends in Optical Disk Technology

In predicting the future of optical disk drives, it is most important to compare their progress with that of magnetic disks. Magnetic disks are increasing in capacity and dropping in price per megabyte of storage at dramatic rates, currently over 50% per year [79]. Optical disks have not kept up with these dramatic improvements in magnetic disk technology. As a result, the cost per megabyte of magnetic disk drives is now comparable to the price per megabyte of many optical disk jukeboxes, approximately 50 cents per megabyte. Given a choice between an optical disk jukebox involving a robot arm and average response time of several seconds and a collection of magnetic disks with millisecond response times for the same price, the choice is highly likely to be in favor of the magnetic disk system. Thus, today optical disks stand in a precarious position compared to magnetic disks.

CD-ROM devices appear to be a possible exception because of their popularity and convenience as an inexpensive means of distributing software and data. In addition, CD-ROM technology will be driven by the high quantities associated with consumer devices. Therefore, they should drop in price faster than other optical disk technologies and may successfully compete with magnetic disks in the future.

2.4 Other Tertiary Storage Technologies

There are a number of tertiary storage technologies still in the laboratory or in a small number of products that may someday compete with magnetic tape and optical disk. In this section we discuss two: holographic storage and optical tape.

2.4.1 Holographic Storage

Perhaps the highest-density storage medium will be the holographic store, which stores data in a pattern of electronic charges in a crystallite material [88]. Parish [78] describes one such product. The storage material itself is a photorefractive crystalline structure of strontium barium niobate doped with cerium. Each crystal is divided into a series of two-dimensional "pages" or patterns of light. A collection of pages in one crystallite is called a "stack." Figure 2.7 shows a simplified diagram of a holographic store.

Data are written in the holostore by first creating a pattern of dark and light spots corresponding to the digital data. This pattern is superimposed onto the laser beam. As this data pattern is sent through the crystallite array, an interference grating is written in

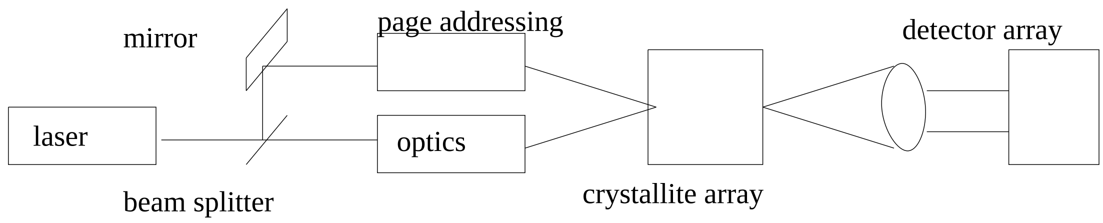


Figure 2.7: *Simplified diagram of holographic store.*

the photorefractive material. That interference grating is converted to a stored electronic charge pattern that modifies the crystallite, storing an image of the bit pattern from the data beam.

Data are read from the holostore by shining a reference beam through the crystallite at a particular angle corresponding to the address of a page of data. The image of the bit pattern written to the page is reconstructed by focusing the bit pattern onto a detector array that reproduces the original digital signals.

Parish projects the following achievable targets: one megabit per page, 100 pages per stack or crystallite and 1000 stacks per module of size 10x10x0.5 centimeters. A module of that size would have capacity over 100 gigabytes. Page read time is predicted as 100 nanoseconds, page write time 10 microseconds, and sustained transfer rate would be 1 terabyte per second. The projected cost of the storage is less than twice the cost per bit of magnetic disk.

2.4.2 Optical Tape

Optical tape technology [106] is similar to that used in optical disks except that the optical coatings are deposited onto flexible material. On top of a polyester base film like that used in magnetic tapes, ICI Imagedata, one manufacturer of optical tape, deposits a proprietary smoothing layer, a metallic reflective layer, an infrared-absorbing dye-polymer layer that is tuned to the wavelength of the read/write laser, and finally a protective overcoat [91], [90]. A backing coat is also applied to the reverse side of the tape.

During the write operation, an 830 nanometer laser with a power of approximately 10 milliwatts is used to melt instantaneously small sections of the dye-polymer layer. Surface tension forms a physical pit with well-defined edges and a depth of about half the layer thickness; the overcoat remains above the pit as a "protective dome." While the unwritten tape has a reflectivity of about 50%, the pit has lower reflectivity. A low-power laser beam is used during reading to detect changes in reflectivity and determine the value of the original data. Because data are stored as physical pits in the tape, optical tape can only be written once.

Reliability studies on the media suggest lifetimes of over 15 years. The medium does not degrade with exposure to UV radiation and is resistant to hydrolysis. Because the optical head has a non-contact interface to the medium, there is no head wear. Tapes can undergo tens of thousands of read passes before unacceptable media wear occurs.

Two companies, Creo [103] [104] and LaserTape [15], have announced products using this optical tape medium. Creo ships a tape drive that uses an optical tape reel that is 12 inches in diameter, 35 millimeters wide and holds 1 terabyte of data.

2.5 Summary

In this chapter, we have given a brief overview of the workings of magnetic tape and optical disk drives and robots. We surveyed representative products and discussed technology trends. Magnetic tape offers the highest capacity for the lowest cost but suffers from the longest access times. Optical disk has shorter access times than magnetic tape but is more expensive. Trends show that magnetic tape systems should continue to have dramatic increases in both capacity and transfer rate. Optical disk systems face increasing competition from magnetic disks, which now match optical disk libraries in cost per megabyte. Finally, we described two emerging tertiary storage systems, holographic storage and optical tape, that may someday compete with magnetic tape and optical disk.

Tape or Platter	Capacity per Unit	Media Cost	Units for 1 Terabyte	Cost for 1 Terabyte
8mm helical tape	5 GBytes	\$0.008	200	\$8,389
D-2 helical tape	25 GBytes	\$0.0018	41	\$1,887
3480 linear tape	800 MBytes	\$0.013	1310	\$13,631
HP magneto-optical disk	650 MBytes	\$0.28	1613	\$293,601
CDROM optical disk	680 MBytes	\$0.003	1542	\$3,146

Table 2.8: *Comparison of media costs for storing a terabyte in different tertiary storage technologies.*

We end this chapter by summarizing the cost and performance characteristics of various technologies. Table 2.8 compares the media cost to store 1 terabyte of data. The high capacity D-2 helical tape has the lowest storage cost. The low manufacturing costs of CD-ROM make it a close second. The highest cost by far is the magneto-optical disk, which is an order of magnitude more expensive than any other technology.

Table 2.9 shows the total system cost for purchasing various robots to store 1 terabyte of data and shows the maximum bandwidth of the resulting systems. Note that these bandwidth numbers are normalized for comparison purposes, so that in the case of the Ampex robot, since we use only a small fraction of the total capacity of the robot, we likewise attribute only a portion of the total bandwidth of one robot to the system. Using this normalization, we see that among the magnetic tape systems, the cost of storing a terabyte is approximately the same in each case; the largest library is slightly less expensive than the others. The normalized bandwidth of the systems is highest for the stacker tape system, reflecting its high ratio of tapes to tape drives. The next best bandwidth belongs to the Ampex library, reflecting its high per-drive bandwidth. For optical disk systems, the system cost for purchasing the robots is substantially higher than for the magnetic tape systems; however, the aggregate bandwidth of the systems is substantially higher as well.

After having introduced tertiary storage technologies, we now turn our attention to applications. Early research on tertiary storage systems focused largely on file migration strategies for file systems and massive storage systems. Smith [101], [102] evaluated the

Robot	Robot Capacity	Robots to Store 1 Terabyte	Cost to Store 1 Terabyte	Max. Bandwidth of Collection of Robots
EXB10i Tape Stacker	50 GBytes	20.5	$180,359	10.25 MB/sec
EXB120 Tape Library	580 GBytes	1.77	$177,000	3.54 MB/sec
Ampex Tape Library	6400 GBytes	0.16	$160,000	9.6 MB/sec
Pioneer CD Changer	4 GBytes	256	$472,320	76.8 MB/sec
HP120LT M.O. Jukebox	120 GBytes	8.5	$446,250	54.4 MB/sec

Table 2.9: *Compares system cost to store 1 terabyte of data and the resulting maximum bandwidth for a collection of robots. Prices from January, 1994 issue of* **SunWorld** *magazine and June, 1992 issue of* **SunExpert***.*

effect of various file migration algorithms on the file system at the Stanford Linear Accelerator, and concluded that the best algorithm migrates files with the largest space-time product, a quotient of their last reference time and file size. Lawrie [54] reached similar conclusions for a much different file system, the Illinois Cyper 175. More recently, Miller perfromed extensive evaluation of a modern massive storage system at the National Center for Atmospheric Research [71]. He concluded that file migration policies should be designed to optimize subsequent read operations. He also found that although files have increased in size and number in the NCAR file system compared to earlier systems, file reference patterns have changed little; the likelihood that a file will be re-referenced drops sharply a few days after file creation.

In this dissertation, we examine not only traditional applications like massive archives but also new ones, such as digital libraries and video servers, in which tertiary storage may play a role. In the next chapter, we present the tools and metrics that will be used in our evaluations.

Chapter 3

Simulation, Measurements, Models and Metrics for Evaluating Tertiary Storage Systems

3.1 Introduction

This chapter describes the event-based tertiary storage simulator that is used in this thesis to explore options for configuring tertiary storage systems. Next, we present detailed performance models of tape drive, optical disk and robot behavior used by the simulator, along with the measurements of actual devices from which the simulation models were derived. Finally, we describe metrics we use throughout this dissertation to evaluate the performance of tertiary storage systems.

3.2 Tertiary Storage Simulation

We wrote a closed, event-based simulator that uses models of tape drive, optical disk drive and robot behavior derived from measurement of real devices. A *closed simulator* is one in which the number of simultaneous requests is set at runtime and is maintained throughout the simulation; a new request is issued immediately after an existing request completes service. Closed systems have the characteristic that for a constant request size and concurrency, response time and throughput have an inverse relationship; if either decreases, the other increases. An *open simulator* is one in which requests arrive according to an event arrival distribution and leave the simulation after they are serviced. We chose a closed simulation for our simulator because it allows us greater control over the system load.

The simulator is event-based. A queue of relevant events is maintained and processed in time order. Table 3.1 lists the events that can be scheduled by the simulator. When an existing request completes, a new *arrival* event is issued and placed on the event queue. When an arrival event is processed, the corresponding request is mapped onto particular physical tapes or optical disk platters according to the prescribed data layout. Individual tape or platter access requests are placed on queues for the robot arms. Each time a robot arm completes a load operation, an *armfree* event is scheduled on the event queue. When an armfree event is processed, the first request is removed from the robot arm queue and the appropriate tape or disk platter is loaded by the robot arm into a free device. If no device is free, the arm must wait for a *devicefree* event. A devicefree event is added to the event queue after a tape drive or optical disk drive completes an I/O operation. In a tape library, before one tape can be removed from a tape drive and another inserted, the existing tape must be rewound and ejected. In the simulator, a *rewindejectdone* event signals the completion of the rewind and eject operations. When the rewind/eject completion event is processed, a robot arm that has been waiting for the tape or disk platter to be ejected is allowed to proceed with a tape or disk switch operation.

There are several input files used to run simulations. First, a *cost* file specifies the total cost of the collection of robots to be simulated. A *workload* file specifies request size and starting position distributions, the percentage of requests that will be writes vs. reads and

Event	Description
arrival	request enters the system
device free	drive finishes processing a request
arm free	robot arm finishes moving an object
rewind and eject done	drive finishes rewinding and ejecting a loaded tape or platter

Table 3.1: *Events in the tertiary storage array simulator that are handled by the event queue.*

sequential vs. randomly distributed, the concurrency of the simulation and the response time constraint, if any. *Device* and *robot* files contain parameters for simulating specific tape drives, optical disk drives and robots. Finally, a *striping* file specifies the striping configuration for the storage array, if applicable. The results generated by a simulation run include the mean and standard deviation of the response time and system bandwidth.

There are two main variations of the simulator code. The first allows the user to set a particular workload concurrency or number of outstanding requests and observe the resulting mean response time and bandwidth; this result is obtained after a single simulator run. This scheme, used in Chapter 5, allows us to observe how well the simulated storage system performs for a specified load. The other option allows the user to set a response time constraint that requires that 95% of all requests finish within the specified response time; for this mode of operation, the user supplies a guess at what might be the final concurrency. This guess is used to minimize the number of simulation runs performed. Starting from the user's guess, the simulator performs a series of simulation runs, increasing or lowering the concurrency of the workload at each iteration until it identifies the maximum load at which the response time constraint can be met. This second variant of the simulator allows us to evaluate the maximum capabilities of a particular storage system configuration for the specified response time constraint; it is used extensively in Chapter 7.

To improve the reliability of the results, the simulator never uses results from the startup period of simulator operation. Statistics are only gathered after a threshold number of requests has been processed. This threshold is typically 1000 requests.

Simulator results are averaged over at least as many I/O operations as specified by a minimum records parameter. Currently this parameter is set at 250 batches of 20 requests each, or a minimum of 5000 I/O operations. Each simulation run continues until the variance of the response time on the operations has stabilized to within a 95% confidence interval. Typical simulation runs take 10 to 30 seconds.

3.3 Simulation Models and Device Measurements

Most previous device modeling work dealt with models of disk behavior [30], [95], [55], [41]. Next, we present our models for tape drive, optical disk and robot behavior along with the measurements upon which those models are based. For tape drives, we characterize load and eject times, rewind and forward search behavior and transfer rates. For optical disk drives, we discuss platter load and unload times, seek and rotation operations and transfer rates. For robots, we characterize arm movement time as well as the time required to move objects between drives and library shelves.

3.3.1 Tape Drive Model

The following parameters must be incorporated into any model that purports to characterize tape drive behavior accurately.

1. Load Time

   This is the time to load a tape into a tape drive. This includes time to wrap a tape around tape reels and read servo and initialization information.

2. Eject Time

   The eject time is the time required to unwrap a tape from the reels and push it out the door of the tape mechanism.

3. Rewind Time: Startup Overhead and Rewind Rate

   All tape drives include a rewind operation that is considerably faster than its normal data transfer rate. Most tapes must be rewound before they can be ejected. Because eject and load operations strain the tape, drive manufacturers leave a strip of unused tape at the beginning of the tape to bear the brunt of these operations. Measurements show that rewind time is characterized by a constant startup overhead to accelerate the tape drive mechanism followed by a constant rewind data rate.

4. Search Time: Startup Overhead and Search Rate

   Tape drives generally include a fast search mechanism for advancing to a particular spot on the tape at a rate much faster than the normal data transfer rate. Small index fields are written on the tape that can be used for positioning information during a fast search. As for the rewind operation, search behavior can be characterized as linear after a constant startup overhead.

5. Transfer Rates

   The transfer rate is the sustained rate at which a drive can transfer data. To maintain the highest possible data rate, the tape drive must be kept in a streaming mode of operation with a steady flow of data transfer.

3.3.2 Tape Drive Measurements

Table 3.2 shows our measurements for the tape drive components of access time for three drives: an Exabyte EXB8500 8mm drive, a WangDAT DAT drive and a Metrum 1/2" drive. All are helical scan magnetic tape drives.

The first two operations measured are mechanical: loading a tape into a drive and ejecting a tape from a drive. In each case, variance between measurements was low. For example, the Exabyte drive had a mean eject time of 16.5 seconds with a variance of 1.01.

Operation	4mm DAT	8mm Exabyte	0.5" Metrum
Mean drive load time (sec)	16	65.8	28.3
Mean drive eject time (sec)	17.3	16.5	3.8
Constant rewind startup time (sec)	15.5	23	15
Rewind rate (MB/sec)	23.1	42.0	350
Constant search startup time (sec)	8	12.5	28
Search rate (MB/sec)	23.7	36.2	115
Read transfer rate (MB/sec)	0.17	0.47	1.2
Write transfer rate (MB/sec)	0.17	0.48	1.2

Table 3.2: *Measurements of 4mm, 8mm and 0.5" helical scan magnetic tape drives.*

The part of the tape load time from insertion into the drive until the ready light turned on had a mean of 35.41 seconds with a variance of 0.08. (The entire load operation is longer, since it entails reading additional servo information.) For the DAT drive, the mean eject time was 17.25 seconds with a variance of 0.93, and the mean load time was 15.95 seconds with a variance of 0.47. The load and eject operations are quite slow; part of the reason is the fairly complex mechanical manipulation of the tape in a helical scan system. On each of the three devices, the combination of an eject and a load operation, which are required during a tape switch, takes at least 30 seconds.

Figures 3.1 and 3.2 show measured rewind and search behavior for the Exabyte EXB8500 drive; these measurements were made for tapes written entirely with 10 megabyte files along with 48 kilobytes filemarks specifying the start of each file. The graphs show one set of measurements; the tests were run several times, and little variance was exhibited. We observe that, after a constant startup time for accelerating the tape mechanism, rewind and search times scale linearly with the number of bytes passed over. Table 3.2 shows the startup time and linear rewind and search rates for each of the drives.

Finally, Table 3.2 shows the sustained read and write rates to a user process measured for each of the drives. In each case, the read and write bandwidth obtained are close to specifications, but the drive can easily perform much worse than this optimum if it is not kept streaming.

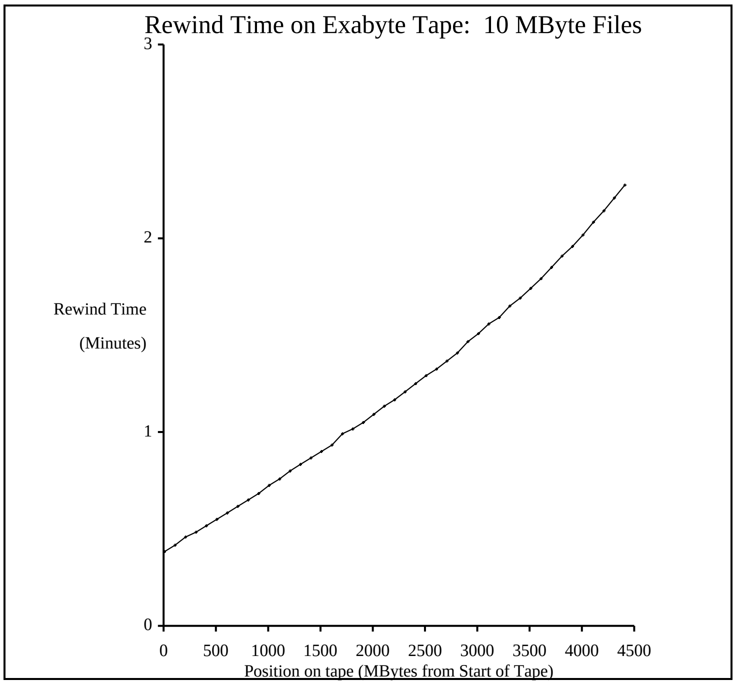


Figure 3.1: *Measured rewind behavior for Exabyte EXB8500 drive. Tape written entirely with 10 MByte files.*

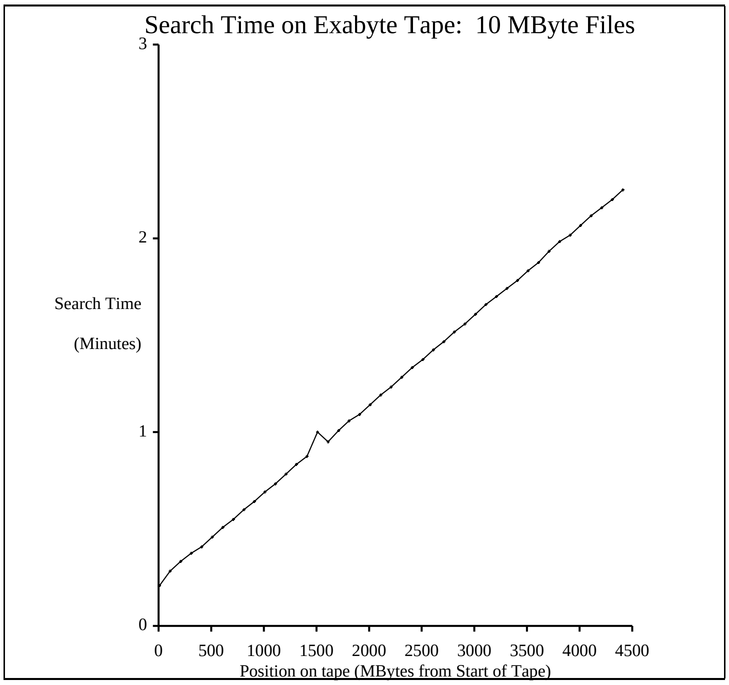


Figure 3.2: Measured search behavior for Exabyte EXB8500 drive. Tape written entirely with 10 MByte files.

Device	1/3 tape volume (GBytes)	Time (sec)
4mm DAT	.400	25
8mm EXB8500	1.5	54
Metrum VLDS	5	70

Table 3.3: *Average seek times for each 4mm DAT, 8mm EXB8500 and Metrum VLDS drives, where the average seek is defined as being the time to search over 1/3 the volume of the tape.*

Table 3.3 shows an important access time parameter, the "average seek" time, or the time to search over 1/3 of the volume of the tape. Such an average seek time may correspond poorly to actual workloads, but we use it as a basis for comparison between products.

3.3.3 Optical Disk Model

The model we use for optical disks is similar to that used for the magnetic tape drive. It differs mainly in the positioning operation, which is now involves a disk seek and rotation operation, rather than the acceleration of the tape mechanism followed by a constant rate of search or rewind.

1. Load Time

   This is the time to load a platter into an optical disk drive. It includes the time to spin up the platter.

2. Eject Time

   The eject time is the time required to eject an optical platter from a disk drive. It includes the time required to spin down the platter.

3. Positioning

   Positioning operations on an optical disk drive include a seek and rotation operation. Unfortunately, we have found little information on the seek behavior of optical disks.

Block size	Data Rate
8K	214 kilobytes/sec
16K	321 kilobytes/sec
256K	544 kilobytes/sec
1024K	542 kilobytes/sec
16384K	378 kilobytes/sec

Table 3.4: *Data rates for different request sizes for an HP magneto-optical disk drive. Measurements were made by Sunita Sarawagi using the Unix* **dd** *command with various block sizes.*

We are attempting to profile this seek behavior. Initially, we have used the average seek time specified by the manufacturer in our simulations.

4. Transfer Rates

   The transfer rate is the rate at which data can be read or written from optical disks. Some optical disks are read-only. For those that are re-writable, the write operation typically involves a separate erase phase, so the write transfer rate of optical disks is often half the read rate.

3.3.4 Optical Disk Measurements

We measured one Hewlett-Packard magneto-optical disk drive; Table 3.4 shows transfer rate measurements made by Sunita Sarawagi. These data rates were measured using the Unix **dd** command and are very sensitive to block size.

3.3.5 Robot Model

There are three parameters that quantify the main activities of the robot.

1. Robot Arm Movement Time

   This is the time required for a robot arm to move from one position to another. Our measurements show little variability in this timing whether the robot arm moves across the length of the robot or to a new position very near its current one. The maximum variance we observed was about half a second, which is only a small fraction of average switch time, especially for tape libraries. Therefore, we model arm movement time as a constant for our simulations.

2. Robot Load Item from Shelf (Pick)

   This is the time required for the robot arm to grab a tape or optical platter off a shelf and insert it into the appropriate drive. The robot load operation is often called a pick operation.

3. Robot Unload Item To Shelf (Place)

   This is the time required for the robot arm to remove an ejected tape or platter from a drive and deposit it on one of the robot's shelves. The robot unload operation is often called a place operation.

3.3.6 Robot Measurements

The robot contribution to the request access time is between 5 and 50 seconds when an operation includes a switch of a tape or optical disk platter into a drive. Table 3.5 shows measurements of grab time for an Exabyte EXB120 robot, a simple rectangular array of 116 tapes and four tape drives. The layout of tapes and tape drives in the EXB120 is shown in Figure 3.3. Figure 3.4 shows a graph of measurements of robot arm movement, where the robot arm moves between a tape position and a tape drive without grabbing any tapes. These measurements were made by Wayne Chen. The arm movement time is fairly constant, between 1.1 and 1.7 seconds. The mean of the arm move times is 1.35 seconds with a variance of 0.013. These variances, which depend on the distance the robot arm travels, are small, especially compared to the average overall access time of several minutes in a typical tape library or tens of seconds in an optical jukebox. Because of the small contribution of robot arm movement to total access time, we model robot arm movement

Time to "pick" tape from drive	19.2 sec
Time to "place" tape into drive	21.4 sec

Table 3.5: *Measured times for robot to grab a tape from a drive and push a tape into a drive for the EXB-120 robot system.*

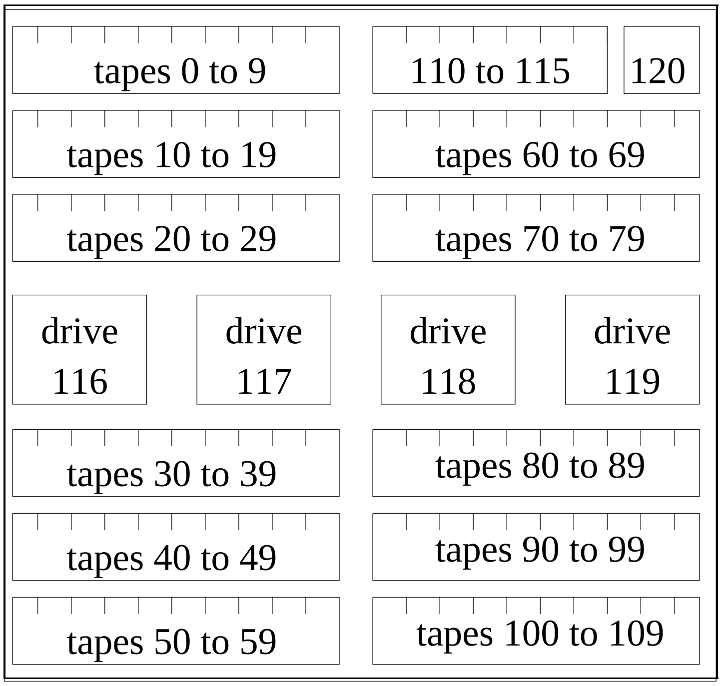


Figure 3.3: *Shows the layout of magnetic tapes and tape drives in the Exabyte EXB120 robot. There are 116 tapes, labeled 0 through 115, and four tape drives, labeled 116 through 119. Slot 120 is a port for adding or removing tapes from the library.*

time as a constant. For the EXB120 robot, the value used in simulations is 1.4 seconds.

Figure 3.5 shows measurements, again made by Wayne Chen, of the time required for the robot to load a tape into a tape drive, including the arm movement time and the "place" time, or time to insert the tape in the drive. Assuming an arm movement time of about 1.4 seconds, the graph indicates that place time is between 21 and 22 seconds.

With these measurements of the EXB120 robot and earlier measurements of the EXB8500 tape drive, we can estimate the average time required to do an access that requires a tape switch operation. We model tape switch time as the sum of the times to rewind the old tape to an ejectable position, the eject operation, the time for the robot to shelve the

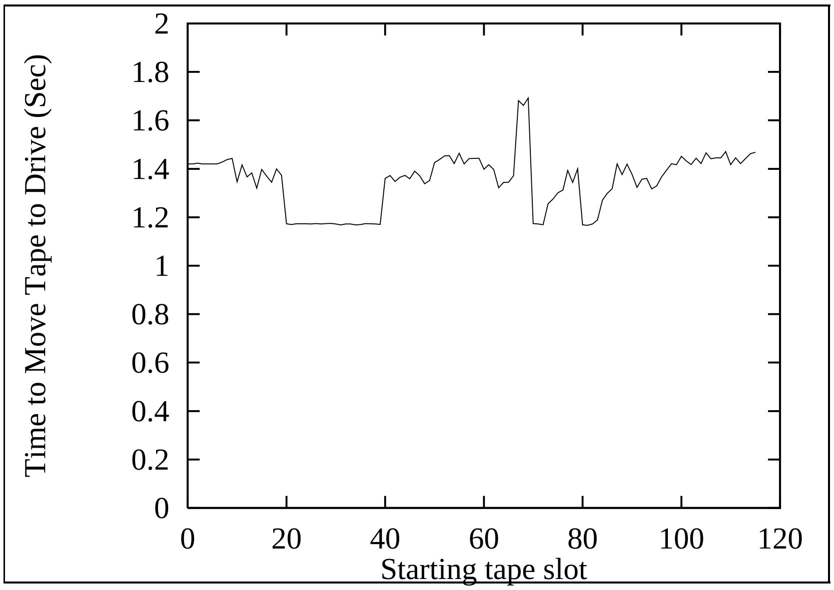


Figure 3.4: *Shows the time to move the robot arm between any tape slot and one of the tape drives. Times are between 1.1 and 1.7 seconds. Measurements made by Wayne Chen.*

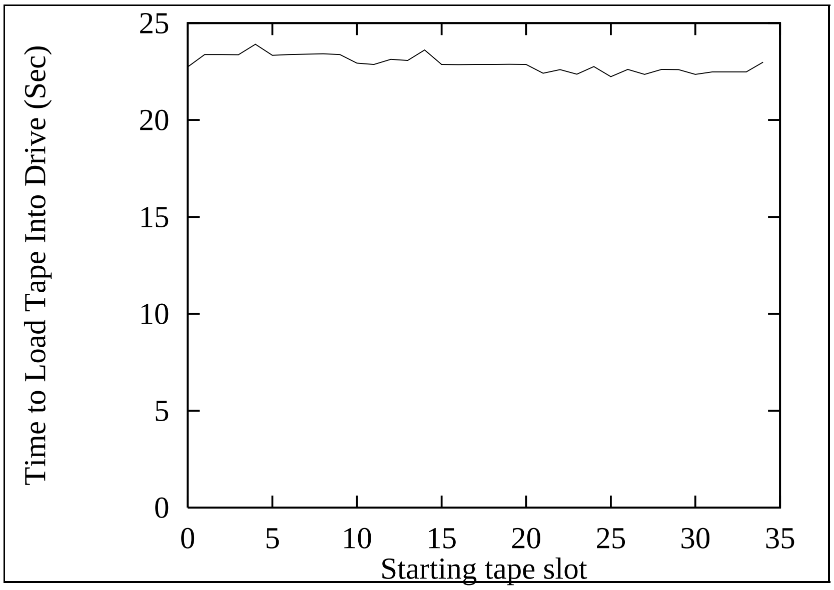


Figure 3.5: *Shows the robot time to load a tape from the given position on a tape shelf into the tape drive. This time includes the robot pick time as well as the robot movement time. The drive load time is not included. Measurements made by Wayne Chen.*

Operation	Time (sec)
Rewind time (1/2 tape)	75
Eject time	17
Robot unload	21
Robot load	22
Device load	65
Search (1/2 tape)	84
Total	284

Table 3.6: *Components of tape switch time for Exabyte EXB120 Robot.*

Operation	Time
Load empty disk drive	8 seconds
Platter switch (remove old, load new)	13 seconds
Time to perform read after threshold	8 seconds

Table 3.7: *Measurements of HP100 magneto-optical disk jukebox.*

old tape and grab a new one, the drive load operation, and the fast search operation to the new data transfer position. Table 3.6 shows that the tape switch time for the EXB-120 robot (not including data transfer) takes four minutes when we assume an average (1/2 length of tape) rewind and search operation. Even more expensive, high-bandwidth drives (D-1 and D-2) and robots with faster robot arms and drive mechanics may take up to a minute for a tape switch.

We also measured the performance of an HP100 optical disk jukebox. Table 3.7 shows the measurements for that robot. Loading an empty disk drive takes about 8 seconds, while a platter switch operation takes approximately 13 seconds. The last number in the table indicates that the jukebox automatically unloads platters that have not been accessed within a certain threshold length of time. So, when the jukebox is idle, its drives will be quickly emptied. This helps to minimize the time to load a new platter but can be harmful to performance if the probability of later re-accessing a platter is high.

3.3.7 Summary of Tape and Optical Disk Models

The access time for a tape operation that includes a tape switch operation is defined as follows:

accessTime =
*rewindStartupTime + (numberOfBytesToRewind * rewindDataRate)*
+ driveTimeToEjectTape
+ robotTimeToRemoveOldTape + robotArmMovementTimeToPutTapeOnShelf
+ robotArmMovementTimeToRemoveTapeFromShelf + robotTimeToLoadNewTape
*+ driveTimeToLoadTape + searchStartupTime + (numberOfBytesToSearch * searchDataRate)*
*+ (numberOfBytesToTransfer * driveTransferRate)*

For an optical disk operation, the access time is defined as follows:

accessTime =
platterSpinDownAndEjectTime
+ robotTimeToRemoveOldPlatter + robotArmMovementTimeToPutPlatterOnShelf
+ robotArmMovementTimeToPutPlatterOnShelf + robotTimeToLoadNewPlatter
+ platterInsertAndSpinUpTime
+ averageSeekAndRotationTime
*+ (numberOfBytesToTransfer * driveTransferRate)*

3.4 Evaluating Storage System Performance

Several metrics help to characterize storage system performance and effectiveness. They can be divided into four categories: latency, throughput, capacity and cost.

Latency characterizes the average response time of a storage system for a particular workload. For some applications, such as an incremental backup of a file system, the important metric may be the time until a particular I/O operation completes. For applications like video service, we also care about the time required to start transferring the first byte of data. Thereafter, we are also concerned with whether the storage system is capable of meeting delivery guarantees for the data stream; for example, many video stream standards require that approximately 30 frames of video be delivered per second. If video frames are delayed, overall stream quality suffers.

Throughput characterizes the net amount of work that can be done by the system. There are two main throughput metrics: bandwidth and concurrency. For sequential, archival or backup workloads, the important throughput metric is the bandwidth that can be delivered to or from the storage system. For more interactive workloads like video servers or digital libraries, the important throughput metric is the number of processes or video streams that can be supported concurrently.

Capacity is the amount of data that can be stored in the tertiary storage system. Important considerations are total capacity of the system, capacity per tape or optical platter, and capacity per square foot of machine room space.

Finally, *cost* is an important metric. In our evaluation of systems, we look at both the *storage cost*, or cost per megabyte of data stored, and the *access cost*, or the cost per process or stream accessing the storage system.

Cost is also the basis we have chosen to compare different system configurations. Other bases for comparison are systems that have the same storage capacity and systems that have the same throughput. All these bases for comparison are problematic. Tertiary storage devices have widely different storage capacities and performance; moreover, tape libraries and jukeboxes have a wide range of ratios of media to drives. For example, a 120-gigabyte HP120 optical disk jukebox with two drives that each transfer at 1.6 megabytes/second (on reads) has a list price of about $50,000. A 600-gigabyte Exabyte EXB120 tape library with four tape drives that each transfer at 0.5 megabytes/second has a list price of about $100,000. To compare systems with approximately the same capacity, five HP120s would be compared to a single EXB120. To compare systems with approximately the same bandwidth, the ratio of HP120s to EXB120s would be 5 to 8. To compare systems with the same list price, one would compare two HP120 jukeboxes to a single EXB120.

We choose system cost as our basis of comparison because there is no easy way to reconcile the vast differences in capacity and performance among the different storage system options. This choice also presents difficulties, however. It is often hard to understand the relationship between cost and price for a particular technology. The difference may be very small in a commodity product such as a disk drive and very large in a high-performance, specialized tape drive used for a small number of supercomputing sites. For simplicity, we generally use list price as the metric for comparison, and we frequently assume list price discounts that we specify and that would be available to universities or those who buy in large quantities.

3.5 Summary

In this chapter we have described the tertiary storage array simulator used in the remainder of this thesis. We have described the models used for the simulations and presented the drive and robot measurements on which these models are based. Finally, we have described the metrics that will be used in this dissertation for evaluating system performance and for comparing performance between systems. In the chapters that follow, we use our simulator to evaluate the performance of a variety of tertiary storage systems.

Chapter 4

Workload-Based Performance Evaluation of Tertiary Storage Systems

In this chapter, we briefly characterize the performance of five tertiary storage robots on three workloads. The first workload is a sequential workload that is representative of backup and archival applications; performance on this workload will be measured by the total bandwidth the hardware can support. The second workload represents movies-on-demand accesses in a multimedia database, and the third represents accesses to a digital library. For both these latter workloads, performance is measured as the number of concurrent accesses that can be supported within a specified response time constraint.

In the next section, we describe the simulation parameters for the five robots. In Section 4.2, we characterize the three workloads. In Section 4.3, we evaluate the performance of the robots for each workload.

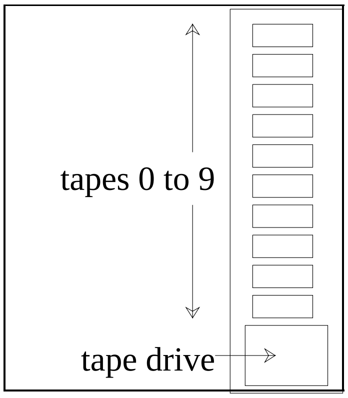


Figure 4.1: Picture of EXB10i tape stacker, containing ten tapes and one tape drive.

4.1 Hardware Simulation Parameters

Based on the model and parameters presented in Chapter 3, we simulate the performance of three tape libraries and two optical disk robots. The tape libraries are the Exabyte EXB10i stacker, the Exabyte EXB120 medium-sized library, and a high-performance library based loosely on the Ampex DST600 library. The optical systems are the Hewlett Packard HP120 magneto-optical disk jukebox and the Pioneer CDROM autochanger.

4.1.1 EXB10i

The EXB10i stacker, illustrated in Figure 4.1, contains 10 tapes and a single EXB8500 tape drive. The parameters characterizing the tape drive are shown in Table 4.1. Robot parameters are shown in Table 4.2. The stacker is the smallest of the tape robots. It contains just one tape drive, so its bandwidth is limited. Among all the robots, the stacker offers the least expensive tape bandwidth, since it has the highest ratio of tape drives to tapes. Conversely, the per-megabyte cost of storage in a stacker is the highest of all tape libraries.

Tape Eject Time	16.5 sec
Tape Load Time	65.8 sec
Rewind Startup Time	23.0 sec
Rewind Rate (after startup)	42 MB/sec
Forward Search Startup Time	12.5 sec
Forward Search Rate (after startup)	36.2 MB/sec
Read Transfer Rate	470 KB/sec
Write Transfer Rate	470 KB/sec
Tape Capacity	5 GBytes

Table 4.1: *Simulation Parameters for EXB8500 tape drive.*

Robot movement time	2.0 sec
Robot pick (grab) time	20.0 sec
Robot place (put) time	20.0 sec
Number of tape drives	1
Number of robot arms	1
Number of tapes	10
List price	$8798

Table 4.2: *Simulation parameters for EXB10i stacker, which includes one EXB8500 tape drive.*

4.1.2 EXB120

The second library we simulate is the medium-sized EXB120 tape library. The library contains 116 tapes and four EXB8500 tape drives. The parameters describing the drive performance were already described in Table 4.1. Simulation parameters for the EXB120 robot are shown in Table 4.3.

Robot movement time	1.4 sec
Robot pick (grab) time	21.4 sec
Robot place (put) time	19.5 sec
Number of tape drives	4
Number of robot arms	1
Number of tapes	116
List price	$100000

Table 4.3: *Simulation parameters for EXB120 tape library, which includes four EXB8500 tape drives.*

Tape Eject Time	5.0 sec
Tape Load Time	5.0 sec
Rewind Startup Time	5.0 sec
Rewind Rate (after startup)	750 MB/sec
Forward Search Startup Time	5.0 sec
Forward Search Rate (after startup)	750 MB/sec
Read Transfer Rate	15 MB/sec
Write Transfer Rate	15 MB/sec
Tape Capacity	25 GBytes

Table 4.4: *Simulation Parameters for high performance tape drive.*

Robot movement time	2.0 sec
Robot pick (grab) time	3.0 sec
Robot place (put) time	3.0 sec
Number of tape drives	4
Number of robot arms	1
Number of tapes	256
List price	$1,000,000

Table 4.5: *Simulation parameters for high performance library.*

4.1.3 High Performance Library

The third robot we evaluate is a high-performance robot. The parameters for this robot are based loosely on the Ampex DST600 robot. The robot contains 256 tapes and four tape drives. The performance of the tape drives is characterized according to the parameters shown in Table 4.4. (These parameters are extrapolated from drive specifications; we did not measure the robot.) The tape drives are much higher-performance than the EXB8500; data transfer rate is 15 megabytes per second. Robot parameters are listed in Table 4.5. Although the robot is expensive (approximately $1 million), it offers the lowest per-megabyte cost of any of the libraries. Also, because the bandwidth of the drives is high, sequential applications that require high bandwidth should be well-suited to this library.

Average Seek Time	0.07 sec
Read Transfer Rate	1.6 MB/sec
Write Transfer Rate	0.8 MB/sec
Platter Capacity	1.3 GBytes

Table 4.6: *Simulation Parameters used to simulate C1617T magneto-optical disk drive.*

Platter unload time	2.3 sec
Platter load time	1.4 sec
Number of optical drives	4
Number of robot arms	1
Number of platters	88
List price	$52500

Table 4.7: *Simulation parameters for HP120 magneto-optical disk jukebox.*

4.1.4 HP120

Next, we show simulation parameters for the Hewlett-Packard HP120 magneto-optical disk jukebox. Parameters for the C1617T optical drive and for the jukebox are shown in Tables 4.6 and 4.7. These parameters are deduced form the product literature; they were not measured.

4.1.5 CD Changer

Tables 4.8 and 4.9 show the parameters used to simulate the Pioneer CDROM drive and the CDROM autochanger. Again, these simulation parameters are deduced from product literature; they were not measured.

Average Seek Time	350 msec
Read Transfer Rate	307 KB/sec
Write Transfer Rate	None (read only)
Platter Capacity	600 MBytes

Table 4.8: *Simulation Parameters used to simulate CDROM optical disk drive.*

Platter unload time	2.5 sec
Platter load time	2.5 sec
Number of optical drives	1
Number of robot arms	1
Number of platters	6
List price	$1200

Table 4.9: *Simulation parameters for CD autochanger.*

4.2 Workload Characterization

In this section, we describe the three workloads used in this chapter: a sequential workload, a video server workload, and a digital library workload. The sequential workload is representative of various traditional tertiary storage applications including incremental and full backups of files systems and storage of archival data sets. The video server and digital library workloads represent future applications in which tertiary storage may play a role; these workloads are described in detail in Chapter 6.

4.2.1 The Sequential Workload

The sequential workload is characteristic of a variety of backup and archival applications. The parameters used to simulate this workload are shown in Table 4.10. We run these simulations at a fixed concurrency of one outstanding request. The initial placement of a request uses a uniform request distribution. Thereafter, the workload issues requests sequentially. The workload is exclusively a write workload, imitating the majority of backup and archival operations. To mimic the performance of a range of operations ranging from incremental backups to large archive operations, the mean request size of operations ranges from 10 megabytes to 10 gigabytes. In the simulation results that follow, we show the maximum bandwidth that can be achieved in a particular tape library or optical disk jukebox for this workload. Such an evaluation assumes that a single process constantly sends data to the tertiary store; thus, it represents the upper limit on bandwidth of such operations.

Workload concurrency	1
Request placement distribution	uniform
Percentage sequential accesses	100%
Percentage write accesses	100%
Response time goal	10,000 seconds
Request size distribution	Exponential
Request size means for simulation runs	10 megabytes
	100 megabytes
	1 gigabyte
	10 gigabytes

Table 4.10: *Shows the parameters characterizing the sequential workload.*

Request placement distribution	Zipf
Percentage sequential accesses	0%
Percentage write accesses	0%
Response time goal	60 seconds
	300 seconds
	600 seconds
	3600 seconds
Request size distribution	Constant
Request size means for simulation runs	2.2 GB

Table 4.11: *Shows the parameters characterizing the video server workload.*

4.2.2 Video Server Workload

In the second workload, we simulate accesses to a movies-on-demand video server. This workload is described in detail in Chapter 6 and summarized in Table 4.11. Movies are picked according to a highly localized distribution called the Zipf's Law distribution. We run the simulations to determine the number of concurrent accesses that can be sustained by each library for a given response time constraint, which varies between simulation runs. All requests read movies in their entirety; there are no write operations and no sequential operations in the workload. All movies are assumed to be 100 minutes in length, and based on assumptions about the compression scheme (listed in Chapter 6), consume 2.2 gigabytes of data storage.

Request placement distribution	Zipf
Percentage sequential accesses	0%
Percentage write accesses	0%
Response time goal	60 seconds
	300 seconds
	600 seconds
	3600 seconds
Request size distribution	Exponential
Request size means for simulation runs	4 MB

Table 4.12: *Shows the parameters characterizing the digital library workload.*

4.2.3 Digital Library Workload

Last, we describe a digital library workload, also discussed in more detail in Chapter 6. Table 4.12 summarizes the workload parameters. We simulate the pattern of requests to the digital library assuming that most users request journal articles. Requests vary in size according to an exponential distribution with a mean of 4 megabytes, which is our estimate of the average length of a journal article stored as a bitmapped image. Requests to the database are highly-localized, made according to the Zipf's Law distribution. Again, response time constraints vary from one minute to one hour; the workload is read-only.

4.3 Performance

4.3.1 Sequential Workload

Figures 4.2, 4.3, 4.4 and 4.5 show the performance of the three tape libraries and the magneto-optical jukebox on the sequential workload for various request sizes. The CDROM autochanger is a read-only robot, so it cannot execute this write-only sequential workload.

Figure 4.2 shows that performance for the EXB10i is constant at approximately the bandwidth of a single drive, regardless of the request size. This is not surprising, since the stacker contains only one drive.

Figure 4.3 shows the performance of the EXB120 library. When request sizes reach 10 gigabytes, the bandwidth doubles compared to smaller request sizes. This is because a single tape in the EXB120 holds approximately 5 gigabytes of storage. Requests larger than 5 gigabytes will span two tapes. Since there are four tape drives in the library, the two tapes can be accessed in parallel, doubling the aggregate bandwidth. In Chapter 5, we show that the technique of data striping is particularly effective in increasing the performance of sequential workloads by using the available tape drives in parallel.

Figure 4.4 shows the performance of the high performance library on the workload. The high native performance of the tape drives gives better aggregate bandwidth. The higher bandwidth shown for the largest accesses suggests that some of the ten gigabyte accesses are spanning two tapes, allowing some of the accesses to be performed by two tape drives in parallel.

Finally, Figure 4.5 shows the performance of the HP120 optical disk jukebox. The write bandwidth of each optical disk drive is approximately 0.6 megabytes per second, and each platter holds 1.3 gigabytes of data. As accesses reach 1 gigabyte in size, many requests span two disk platters, approximately doubling the aggregate bandwidth. As requests grow to 10 gigabytes in size, all four optical disk drives are operating in parallel to achieve the maximum of 2.4 megabytes per second of write throughput.

4.3.2 Video Server Workload

Table 4.13 shows the performance of the five storage robots on the workload that mimics requests to a movies-on-demand video server. The table shows the number of concurrent accesses that can be sustained by the robot given a specified limit on the average time to deliver the first byte of data.

Both the EXB120 and EXB10i robots use EXB8500 tape drives, which have quite long access times. As a result, for a response time constraint of one minute, neither robot

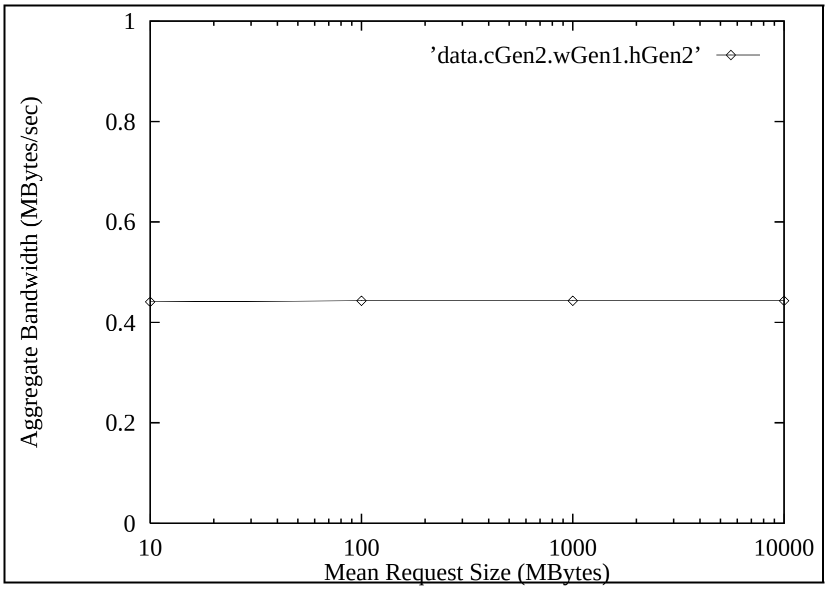


Figure 4.2: Performance of EXB10i stacker on sequential workload.

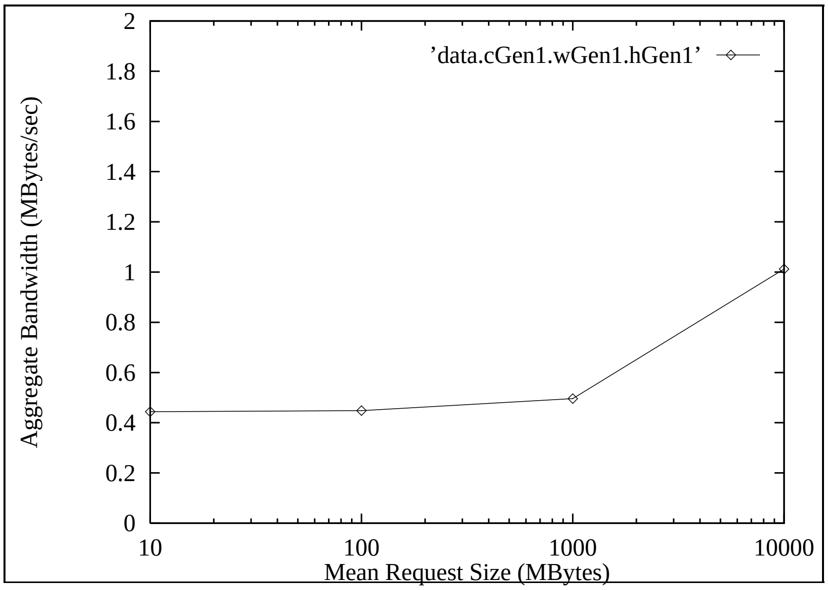


Figure 4.3: Performance of EXB120 library on sequential workload.

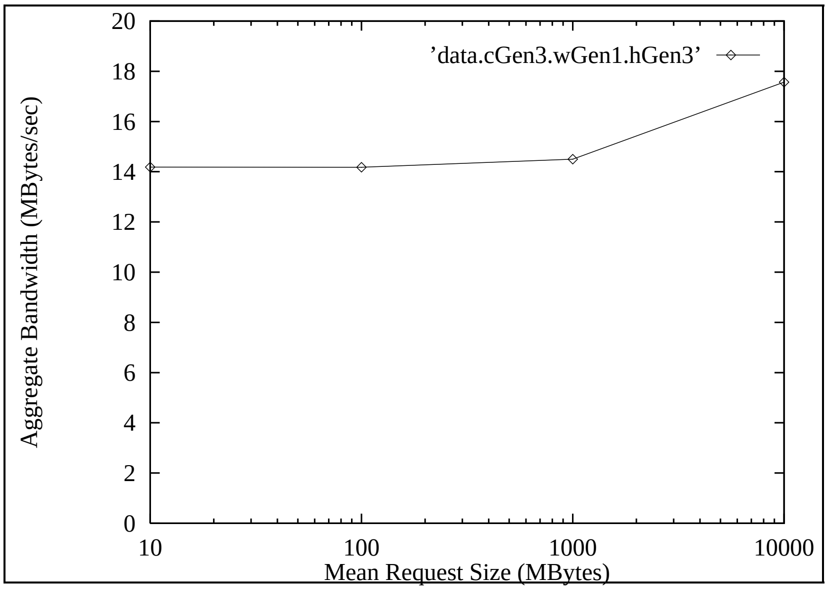


Figure 4.4: Performance of high performance library on sequential workload.

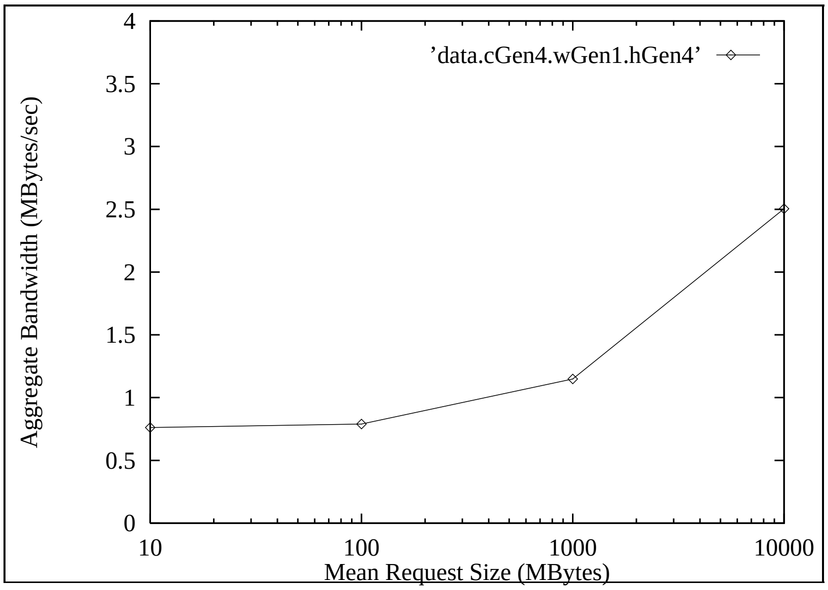


Figure 4.5: Performance of magneto-optical jukebox on sequential workload.

Robot	Response Time Limit (seconds) 60	300	600	3600
EXB10i	0	1	1	1
EXB120	0	1	1	4
High Perf Library	1	6	10	36
HP120 MO Jukebox	1	2	3	10
CD Autochanger	1*	1*	1*	2*

Table 4.13: *Shows the number of concurrent streams that can be maintained in each tertiary storage robot for the video server workload given the specified limit on the response time to first byte of data. The * by the CD Autochanger results indicates that although the robot can satisfy the response time constraint of the workload, the bandwidth of the CDROM drive is insufficient to satisfy the demands of the 3 megabits/second video stream. The workload assumed for the video-on-demand application is discussed in detail in the last two chapters of this dissertation.*

can sustain any concurrent accesses. As the response time limit increases, the EXB10i stacker is able to support a single access; when the response time limit reaches one hour, the EXB120 can sustain up to four concurrent accesses. Both are limited by the number of tape drives in the robot.

With a much higher data transfer rate, the high performance library is capable of supporting considerably more video accesses. While only one request can be sustained with a response time limit of one minute, with a response time limit of one hour, 36 concurrent accesses are possible.

For the HP120 magneto-optical disk jukebox, the concurrency supported is limited by the number of optical drives in the jukebox. At a response time limit of one minute, a single stream is supported; when the response time limit reaches an hour, ten streams can be sustained. This concurrency is determined by the total system bandwidth. Since each movie will span two 1.3 gigabyte platters, two drives can service a movie in parallel in 704 seconds; thus, in an hour, four magneto-optical drives can service 10 movies.

Finally, the CDROM autochanger could sustain up to two accesses with a response time limit of an hour. However, as explained in Chapter 6, we assume that each video stream

Robot	Response Time Limit (seconds)			
	60	300	600	3600
EXB10i	0	1	2	22
EXB120	0	4	9	68
High Perf Library	4	23	45	331
HP120 MO Jukebox	27	137	357	4103
CD Autochanger	2	15	32	228

Table 4.14: *Shows the number of concurrent streams that can be maintained in each tertiary storage robot for the digital library workload given the specified limit on the response time to first byte of data.*

requires approximately 3 megabits per second of sustained data transfer. The data rate of the CDROM drive is slightly too low to satisfy this constraint.

4.3.3 Digital Library Workload

Table 4.14 shows the performance of the robots on the digital library workload. In this case, accesses are much smaller, with a mean of 4 megabytes. Requests to the data are highly localized according to the Zipf's Law distribution.

As for the video server workload, the access times of the EXB8500 tape drive are slow enough that for a response time of 60 seconds, neither the EXB10i or the EXB120 can sustain any concurrent accesses. However, for longer response times, both robots sustain considerably more concurrent accesses than for the video server workload. The reason for this is the much shorter requests for this workload. The data transfer time is reduced, and because accesses to the data are highly localized, relatively few tape switch operations are required. Thus, for a response time constraint of one hour, the EXB10i supports 22 concurrent accesses and the EXB120 supports 68 accesses. The other tape library and the optical disk jukeboxes show similar performance increases. Because of the much smaller request sizes, each robot is able to sustain a relatively high number of concurrent accesses.

4.4 Summary

In this chapter, we have briefly characterized the performance of five robots on three workloads. For a sequential workload, we found that the aggregate bandwidth achieved for a variety of request sizes is generally limited by the bandwidth of a single drive; if several drives can perform an access in parallel, the aggregate bandwidth increases considerably. For a movies-on-demand video server workload, we showed than none of the robots is capable of sustaining more than a few concurrent accesses for response times of one minute or five minutes; the small number of drives and the long access times result in poor performance of the library except when response time limits approach an hour. Finally, for a workload characteristic of accesses to a digital library, the robots perform better; with a highly-localized workload, each robot can sustain a small or moderate number of concurrent accesses with a response time limit of a few minutes and a more impressive number of requests for a response time limit of an hour.

The native performance of these tertiary storage libraries is fairly disappointing. The bandwidth of individual drives and the small number of drives in a typical tertiary storage system limit their usefulness for many applications. In the next chapter, we evaluate data striping as a technique to improve performance. In Chapters 6 and 7, we evaluate tertiary storage systems as components of a storage hierarchy to support digital library and video server applications.

Chapter 5

Tape Striping

5.1 Introduction

In the last chapter, we evaluated the performance of various tertiary storage devices on three workloads. Unfortunately, the performance of tape libraries was poor on two of those workloads because there are relatively few tape drives in a typical library, the bandwidth of individual tape drives is low and response times for tape accesses are long. Since individual tape drives perform poorly, in this chapter we investigate using a collection of tape drives in parallel in a scheme called *data striping* to improve tape library performance.

Data striping is a technique for interleaving or striping data from individual files across several storage devices [92], [46], [56], [30]. Since these devices can access the individual stripe partitions of the file in parallel, a striped storage system can provide greater throughput to the file and reduce the response time of large accesses. Striping has been used very successfully in arrays of magnetic disks.

In this chapter, we explore striping in arrays of magnetic tapes and tape drives. In earlier chapters, we discussed the performance of various tape drives and robots and showed that many inexpensive tape drives have low bandwidth. Thus, tape systems appear to be good candidates for the increased bandwidth that can be achieved using striping.

In the results that follow, we show that for a limited class of workloads and array configurations, tape striping can be very effective. Specifically, tape striping works well for sequential workloads and workloads in which there are few concurrent accesses. Unfortunately, at higher loads when accesses are more randomly distributed, the performance of striped tape arrays is poor, largely due to contention for the tape drives.

We begin this chapter with a brief overview of striping in magnetic disk arrays and tape libraries. Section 5.3 describes a number of configuration and performance issues in striped tape systems. Section 5.4 presents simulation results for typical tape robots running sequential and randomly-distributed workloads. We include simulations that show how striped tape performance changes with improvements to tape drives and robots; these improvements include libraries with more tape drives and with faster tape drives and robots. The improvement most beneficial to striped tape performance is a larger number of tape drives in the system.

5.2 Data Striping in Disk Arrays and Tape Libraries

5.2.1 Striping in Disk Arrays

In striped arrays of magnetic disks [92], [46], [56], [30], a single file is striped or interleaved across several disks as shown in Figure 5.1. The unit of data interleaving or striping among the disks is known as the *stripe unit.* The collection of disks over which data are interleaved is the *stripe* or *stripe group.* Because a striped file can be accessed by several disks in parallel, the sustained bandwidth to the file is greater than in non-striped systems, where accesses to the file are restricted to a single disk. As a result, latency is reduced for large accesses that have long periods of data transfer.

Striping or data interleaving in a disk array may be done in small or large blocks [30]. The smallest unit of data that can be written to any disk is a sector (512 bytes for

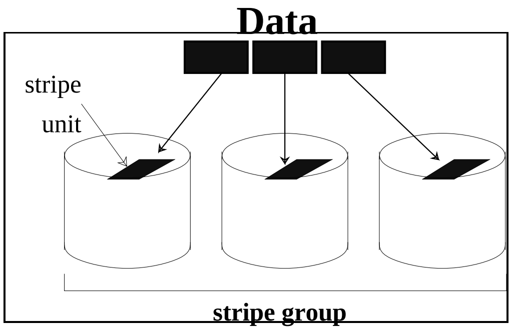


Figure 5.1: *Illustrates data striping or interleaving in a disk array. The unit of data interleaving is known as the stripe unit. The collection of disks over which data are interleaved is called the stripe or stripe group.*

most disks), although logically, data might be bit- or byte-interleaved across the disk stripe. Since most file systems access data on disk in units of 4 or 8 KBytes, a system using small block interleaving will involve all the disks in a stripe in every access. This makes logical synchronization of the disks straightforward but does not allow independent accesses to the disks in the stripe.

Data interleaving may also be done in larger increments. The size of the interleave or stripe unit might be chosen to optimize sustained bandwidth (as done by Chen and Patterson [20]) or to minimize response time. In large block interleaved systems, concurrent independent accesses within a stripe may occur if individual accesses are small enough that they don't involve all the disks in the stripe. This potential parallelism is an advantage of large block interleaving over small block interleaving. This advantage may be offset, however, by increased latency penalties; drives acting independently will become unsynchronized, and subsequent large accesses involving several drives will have to wait for the completion of the slowest device.

5.2.2 Disk Array Reliability

Failures are a concern in disk arrays, since failures are more frequent in systems with many components. For example, assuming independent failures, 100 disks collectively

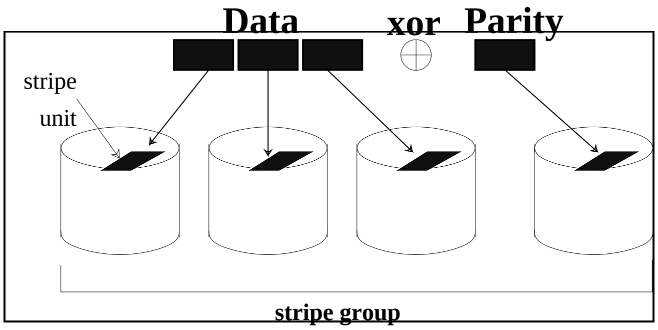


Figure 5.2: *Illustrates data striping with single-bit parity; parity is computed over the stripe units in a stripe group using an exclusive-or operation.*

have only 1/100th the reliability of a single disk [19]. In large storage arrays, potential failures include transient media errors, media wear, head failure, other mechanical problems with the device, breakdown of disk controllers, and failed power supplies or cables [94]. There is error correction information embedded in each disk track that enables the disk to volunteer the information that it has an error. To correct disk errors, the disk array maintains additional "parallel" error correction information across the disks. Although it is not necessary to perform striping to include such redundancy information [32], it is convenient to calculate error correction codes over a stripe. Figure 5.2 illustrates striping with single-bit parity for redundancy.

A parallel error correction code (ECC) for storage arrays is chosen based on its ability to protect the data against likely errors and based on its impact on the performance and capacity of the array [30]. Performance of write operations is affected by the addition of ECC, since extra redundancy calculations and extra write operations to store the error correction information must be performed. Also, the choice of ECC will affect performance when data are reconstructed after a disk failure. The ECC chosen also affects the amount of useful data storage on the array, since redundancy information is stored in place of other data.

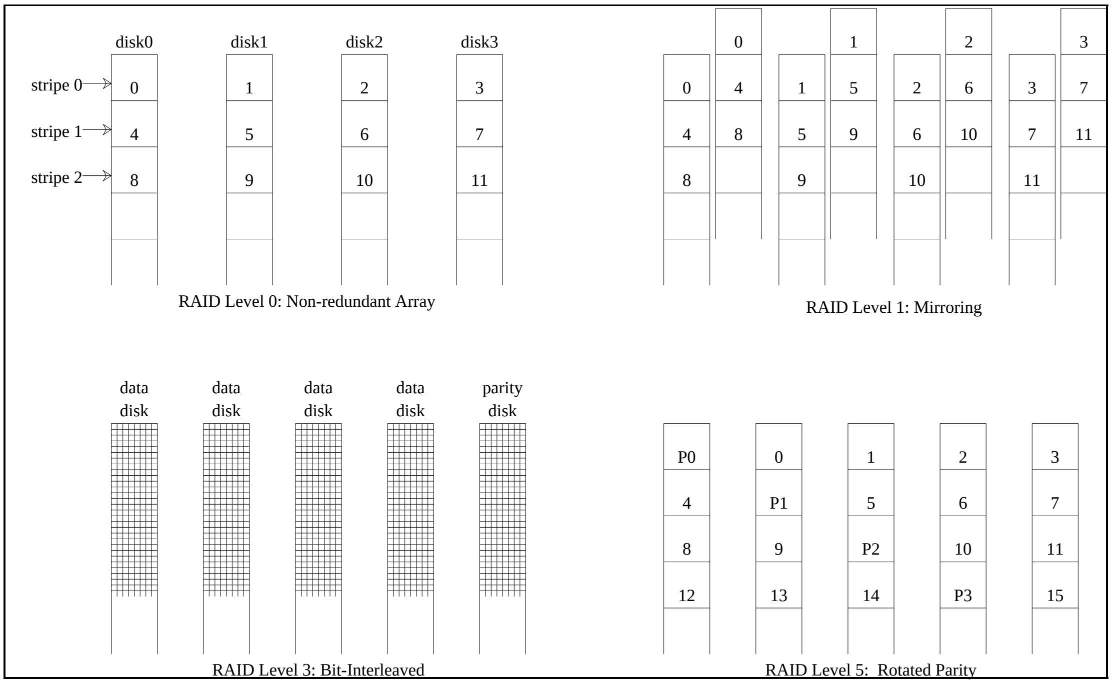


Figure 5.3: *Illustrates the disk array taxonomy. In RAID Level 5 picture, blocks labeled Pn are parity blocks.*

5.2.3 RAID Taxonomy

The RAID group at U. C. Berkeley developed a taxonomy for RAID systems to describe the different options for data interleaving and redundancy [19]. Figure 5.3 shows several RAID classifications or "levels" [18].

RAID Level 0 is a non-redundant collection of disks. RAID Level 1 uses mirroring; whenever data are written to a disk, a copy of the data is written to a redundant disk. Mirrored systems contain twice as many disks as non-redundant disk arrays. RAID Level 2 decreases the number of disks required to provide reliable storage by calculating a Hamming code that makes it possible to identify a failed disk and recover the lost data. For example, one scheme protects four disks using a Hamming code that requires three additional disks to implement. RAID Level 3 takes advantage of the fact that disk controllers can identify failed disks; therefore, no error detection is necessary, and a single parity disk can provide protection from a single disk failure. RAID Level 3 assumes that data are bit-interleaved within the disk array. RAID Level 4 uses large block interleaving and single disk parity. Unfortunately, storing all parity data on a single disk may result in that disk becoming a performance bottleneck. Therefore, RAID Level 5 uses large block interleaving and distributes or rotates the parity blocks uniformly over all the disks in the array. Finally, RAID Level 6 provides protection from a second disk failure by incorporating a P + Q Reed-Solomon encoding that requires two redundant disks.

5.2.4 Tape Striping

Tape striping spreads the data from a single file across several tapes. The technique should reduce the data transfer time of a large access by using several tape drives in parallel to increase the aggregate throughput. However, striping does not change the time required to remove a tape cartridge and insert a different tape, called the tape switch time. We saw in Chapter 3 that this time may be quite long; for example, several minutes on average in an EXB120 library. Striped systems are likely to require more of these time-consuming tape switches, since files will be spread across several tapes; each tape must be loaded into a tape drive before the data can be accessed. Since most tape libraries have a relatively small number of tape drives, contention for those resources may result in long delays in servicing the requests. Thus, despite the benefits of greater aggregate throughput for each request, striped systems may perform poorly if they spend most of their time switching tapes into and out of drives.

Tape striping, usually without redundancy, is already being used for backups, archival operations, and for capturing large scientific data sets [24], [84]. Existing systems use striped tape to increase the throughput for archival applications. The workload for these applications usually consists of a single process writing large amounts of data to a collection of tape drives operating in parallel. The benefits of tape striping for such applications are confirmed by our results in Section 5.4.2.

In this chapter, we also examine the performance of tape striping for an application that has two characteristics that distinguish it from the archival application. First, several processes concurrently request access to large amounts of data (hundreds of megabytes or more). Second, the accesses are randomly distributed through the tape array rather than sequential. This is an important workload since it will likely reflect the access pattern of future multimedia databases. Unfortunately, we find that existing tape libraries perform poorly for this application. In Section 5.4.3, we show that performance of the striped tape system on this workload is limited by contention for the small number of tape drives in the library. We show that several tape library characteristics must change to support this application. Most importantly, a higher ratio of tape drives to media is required. Next in importance are higher transfer rates on tape drives, followed by faster search and rewind operations and mechanical operations such as load and eject.

5.3 Tape Striping Issues

In this section, we discuss configuration and performance issues for striped tape systems. First, we discuss the alternatives of striping data within a robot or among several robots. Next, we examine the optimum stripe width and the choice of redundancy scheme. We discuss synchronization and buffer space requirements. Finally, we conclude with a discussion on how tape striping will be affected as tape drives and robots improve.

5.3.1 Configuring a Tape Array

In Chapter 2, we discussed the wide variety of tape drive and robot technologies that may be used in configuring a storage system. The tape drives range from inexpensive, low-bandwidth drives to much more expensive and higher performance tape drives. Tape striping has obvious potential benefits for increasing the throughput of inexpensive, low-

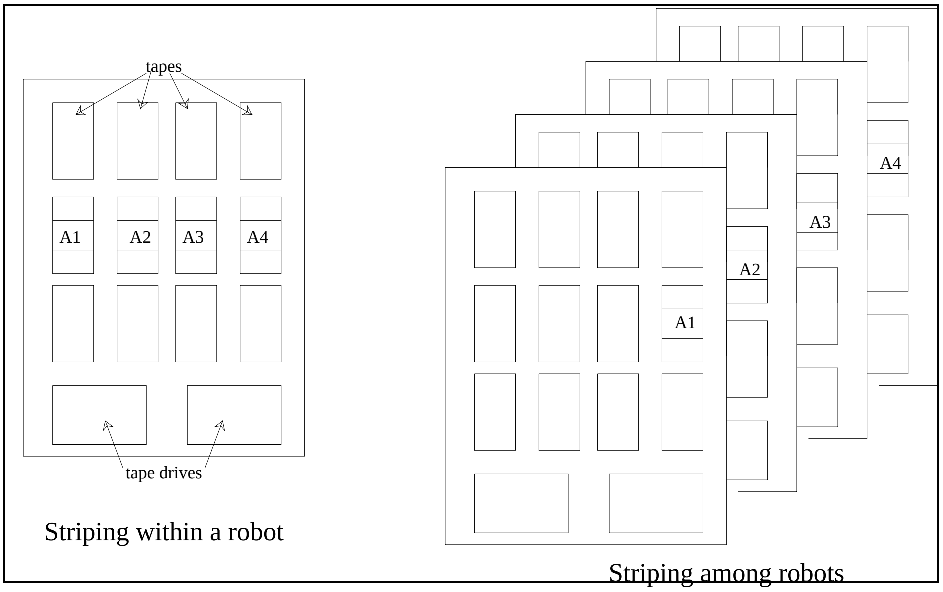


Figure 5.4: *Examples of the two options for striping in tape libraries. The left figure shows the blocks of file A striped within a single large tape robot. On the right, the blocks of the file are striped among several physically separate robots.*

performance drives. For many applications, the bandwidth of the more expensive drives will be adequate; however, striping is still a useful technique for high performance tape drives to satisfy the demands of the most bandwidth-intensive applications.

A large storage system may be composed of a single large robot or some combination of smaller robots. The robots characterized in Chapter 2 range from inexpensive stackers with a few tapes to large libraries with hundreds or thousands of tapes. Large libraries take up the least square footage of machine room space for a given capacity, while stackers take up the most. Large libraries have the lowest ratio of tape drives to tapes, however, while stackers have the highest. Large libraries have the lowest cost per megabyte of storage.

There are two main options for striping in storage systems composed of a collection of tape robots. Data may be striped within an individual robot or across several robots. Figure 5.4 illustrates the alternatives.

The most straightforward application of tape striping is within a large robotic library. Most such libraries have a large number of tapes (hundreds or thousands) and a relatively small number of tape drives (typically between 2 and 16). In most libraries, all tapes are loaded into tape drives by a single robot arm. Tape striping within a large library would require the robot arm to load all the tapes involved in a striped access in sequence and may result in contention for the robot arm. Contention for the tape drives is also likely. Because striped accesses span several tapes, more tapes must be loaded per typical access for a striped system than for a conventional library configuration. As a result, more tapes compete for access to a limited number of tape drives. When the concurrency of the workload is high, contention results, as described in Section 5.4. One of the advantages of striping within a single large library is that it will be convenient to keep track of tapes that are logically connected in a stripe when they are in a single physical enclosure.

Striping across independent robotic libraries has several potential advantages over striping within a single large robot. First, each library has its own robot arm and a unique set of tape drives into which tapes from that library are loaded. Thus, there is less likelihood of contention for a single robot arm in this configuration. If the striping is performed across small robots such as stackers, these robots will have a better ratio of tapes to tape drives than the large libraries, so there is less likelihood of contention for tape drives in such a system. One of the disadvantages of striping between physically separate robots is the complexity of keeping track of tapes that are logically grouped into a stripe set, yet are stored in physically separate libraries. This administrative problem is alleviated if tapes never leave the library or if there is a standard procedure for moving stripe sets between the library and the shelf.

5.3.2 Stripe Width

In the parlance of disk arrays, the *stripe width* is the number of devices across which data from a single file are interleaved. In tape arrays, the stripe width is the number of tapes across which a single file is striped. The choice of stripe width is likely to have a large impact on the performance of striped tape systems because there are many tapes in a library or collection of robots, but relatively few tape drives. When several requests are being serviced simultaneously, the tapes being accessed compete for the available tape drives. If the stripe width is chosen such that too many tape drives are involved in each access, there may be long delays in servicing requests, especially for workloads where there are several outstanding accesses.

An obvious rule-of-thumb is that the stripe width should not exceed the number of tape drives in the storage system; otherwise, multiple tape switches will be required from some of the drives just to service a single request. If it is desirable to support multiple users accessing the storage system simultaneously, then the stripe width should ideally be chosen to support the expected workload. For example, if a library contains sixteen tape drives and system load could be expected to average several simultaneous requests, the system designer might choose a stripe width of four to ensure that enough tape drives are available to handle several concurrent requests.

5.3.3 RAID Levels in Tape Striping

In Section 5.2.3 we discussed the RAID taxonomy for different data interleaving and redundancy schemes. The two most commonly-used options are RAID Level 3, or bit-interleaved data, and RAID Level 5, where data are interleaved in larger blocks and parity is rotated among the disks or tapes.

The use of RAID Level 5, or large block interleaving, would have the advantage of potentially requiring fewer tape switch operations than a RAID Level 3 scheme. Relatively small accesses to the striped tape array could often be handled by loading a single tape; in a RAID Level 3 scheme, by contrast, an entire tape stripe must be loaded to satisfy even a small request. Large accesses to a RAID Level 5 tape array would still span several tapes, so they would still receive the throughput advantages of being accessed by several tape drives in parallel.

In this chapter, we assume the use of RAID Level 3 striping because using RAID Level 5 striping may require update-in-place operations to maintain parity. As discussed in Chapter 2, with few exceptions, magnetic tape drives are append-only devices and do not perform update-in-place. The need for update-in-place operations arises because in a RAID Level 5 system, tapes are accessed relatively independently. Assume that tapes 1, 2, 3 and 4 make up a stripe. If a write operation smaller than the stripe unit is performed at the start of tape 1, the corresponding parity is written to the appropriate tape, say tape 3. If a second, even smaller write operation is performed at the start of tape 2, then new parity will be computed and written to tape 3. However, the access to tape 2 is smaller than the previous access, so the parity information on tape 3 need only be partially re-written. Because of the append-only nature of the tape drive, however, the remaining parity will be lost when the new parity is written, unless steps are taken to buffer the remaining parity and rewrite it along with the new parity.

If update-in-place operations could be avoided, a RAID Level 5 tape library would have the advantage of requiring fewer tape switches than RAID level 3. The need for update-in-place would be eliminated in a read-only storage system. Perhaps the simplest solution for a read/write storage system would be to limit tape accesses to some integral multiple of stripe units to avoid partial overwrite of parity. Another possibility is to handle the parity for the tape library differently than the data, for example, storing the parity for a collection of tapes on a device such as a magnetic disk that would be better-suited to update-in-place operations. However, given the capacity of typical tapes and libraries, it may be prohibitively expensive to store the parity on magnetic disk.

For the remainder of this chapter, we assume the use of RAID Level 3 striping.

5.3.4 Synchronization Issues and Buffer Space Requirements

Synchronization of drives in a striped array is necessary since components of the striped data transfer must be merged before delivery to the process that requested the transfer. In disk arrays, synchronization is fairly straightforward. Each disk involved in a striped access performs a seek operation that may take a maximum of about 20 milliseconds on a typical disk. Each disk also must perform a rotation to the correct data transfer position. Many disk drives are capable of spindle synchronization, however, so that they are positioned at the same point on a disk cylinder as the other disks in the array; spindle synchronization virtually eliminates variance in the time required to perform a rotation. In general, the disks will not be out of synchronization by more than the time required for a maximum seek operation on the disk. To reassemble the contents of a data stripe, the disk array controller collects the components of the transfer in buffer memory; after all have arrived, the reconstructed data are passed to the requesting process.

Synchronization in tape arrays is more complicated. First, unlike in disk arrays where there is a separate disk head for every piece of storage media, in a tape array, there are a limited number of tape drives and robot arms. In retrieving the components of a striped access, long delays are experienced if no free drive is available to read or write some component of the access or if a single robot arm must sequentially load several tape drives. Second, low-level synchronization like the spindle synchronization used in disk arrays is impossible in tape arrays. Because of the high rate of write errors, as explained in Chapter 2, data are sometimes rewritten several times before they can be read back correctly; as a result, it is impossible to predict exactly where data will reside on a tape, and therefore impossible to do synchronization based on such predictions. (One argument for incorporating parallel error correction in a tape array is that this read-after-write checking might be eliminated; any resulting errors could be corrected using parallel ECC, and the difficulty of synchronization would be reduced. Since inexpensive, commodity tape drives are likely to incorporate error correction and read-after-write checking, it may be desirable to allow a user to turn off such checking when using parallel ECC. Determining the reliability of such a scheme is left for future work.)

Finally, for systems using large block or RAID Level 5 interleaving, there is an additional synchronization problem. Requests smaller than the interleave factor can be satisfied by accessing a single tape. The tape library may process several such requests independently, loading and accessing unrelated tapes. Later, a large request requiring access to an entire stripe width of tapes may see widely different latencies for the tape accesses; each tape requested might be on the shelf, loaded in a tape drive, or being read or written to service a previous request.

To mask delays in accessing tapes that contain the components of a striped access, the system stores the components in buffer memory as they are retrieved. Because tape delays are so long and tape accesses are generally large, the amount of buffer space required to perform synchronization may be quite large. In addition, buffer space requirements increase with the load or concurrency of outstanding requests in the system and with the size of the interleave unit.

5.3.5 Future Devices

In Chapter 2, we discussed trends in magnetic tape technology. We showed that tape capacity and drive transfer rate follow trends in magnetic disk devices and are steadily increasing. Other components of tape drive access time are also improving, including rewind and fast forward times and robot access times.

Improvements in tape drive *throughput* may make tape striping unnecessary for some workloads whose bandwidth requirements can be satisfied by individual tape drives, as shown in Section 5.4. However, data striping will always be a useful technique for getting more throughput out of a storage system by accessing several drives in parallel.

Improvements in tape drive *access time* will make striping more attractive, since they will reduce the heavy penalty currently paid for every tape switch operation.

5.4 Performance of Striped Tape Systems

While striped tape systems can improve throughput to individual requests, the aggregate performance of striped tape arrays may be much worse than that of *standard* (non-striped) tape arrays for certain workloads. The reason is that striped systems perform more time-consuming tape switch operations than standard systems.

In this chapter we show simulation performance for striped tape systems handling two workloads. In Section 5.4.2, we show that striping offers tremendous benefits for a sequential workload with a single process sending data to the tape library. This workload is typical of backup and archival applications. We show that the benefits of striping increase linearly with the number of tape drives over which data are striped.

In the remainder of the chapter, we focus on a workload in which several processes may be requesting data simultaneously, and requests are large (usually hundreds of megabytes) but distributed randomly through the array. The workload applied here is a generalization of the workload described in Chapter 6 and is meant to approximate how data will be accessed in future multimedia databases. We find that the striped performance of typical tape libraries on this workload is inadequate because of contention for the small number of tape drives in the array. We simulate various improvements to the tape library and evaluate their impact on striped performance; these improvements include adding drives to the robot and improving the speed of the tape drives and robots.

5.4.1 Simulations

The simulation results presented in this chapter use the event-driven simulator described in Chapter 3. In all these simulations, the request size is kept constant during individual simulation runs. For Section 5.4.2, the requests are sequential and a single process sends output to the I/O system at a time; all requests are write operations. This workload reflects that of traditional backup and archival applications. For the remaining sections of the chapter, we simulate a workload that we believe will reflect future multimedia databases. In particular, requests in this workload are randomly distributed and are mostly read operations, and there may be several outstanding requests active at any time. The workload used in this chapter is a generalization of the workloads used in Chapters 4 and 6; it uses a uniform distribution of requests rather than the highly-localized Zipf distribution. We simulate 75% read operations and 25% write operations. For all the simulations except those in Section 5.4.6, A RAID Level 0 (no redundancy) striping scheme is used for these simulations for the striped tape arrays. For Section 5.4.6, we use a RAID Level 3 [80] striping configuration with single bit parity; the reliability information maintained in this last set of simulation results is largely irrelevant to the performance study presented in this chapter.

The simulation results in the following sections compare striped and non-striped performance for two tape arrays: the EXB120 and the high performance library described in Chapter 4. Files are striped over groups of three data tapes plus one parity tape. Recall that the EXB120 library holds 116 tapes and four EXB8500 tape drives, and the high performance library holds 600 tapes and four tape drives. In Table 5.1, we summarize the simulations performed in the remainder of this chapter.

5.4.2 Sequential Request Performance

Tape striping works very well for important sequential workloads such as backup and archival applications. Figure 5.5 shows the performance of an Exabyte library that is striped with various *stripe widths*, or numbers of tapes across which data are interleaved. Although Exabyte libraries actually contain 4 tape drives, we simulate a library with up to

Request Distribution	Hardware Description	Simulation Varies...	Result
Sequential	EXB120, 32 drives	Stripe width 1 to 32 tapes	BW
Sequential	EXB120, 8 drives	Mean request size	BW
Random	EXB120, 4 and 16 drives	Mean request size, concurrency of 1,4 and striped vs. standard	RT
Random	High Performance library, 4 and 16 drives	Mean request size, concurrency of 1, 4 and striped vs. standard	RT
Random	EXB120 with 4 drives	Mean request size, concurrency of 1, 4, striped vs. standard speeding up tape drive, robot and both by factor of 10	RT
Random	EXB120 with 16 drives	Mean request size, concurrency of 1, 4, striped vs. standard speeding up tape drive and robot by factors of 1.5, 2 and 10	RT
Random	EXB120 with 16 drives	Mean request size, concurrency of 1, 4, striped vs. standard speeding up tape drive only by factors of 1.5, 2 and 10	RT
Random	EXB120 with 16 drives	Mean request size, concurrency of 1, 4, striped vs. standard speeding up robot only by factors of 1.5, 2 and 10	RT
Random	Hypothetical robot with 24 drives	Mean request size, concurrency of 1, 4, striped vs. standard	RT
Random	Hypothetical robot with 40 drives	Mean request size, concurrency of 1, 4, striped vs. standard	RT
Random	EXB120 with 16 drives	Drive transfer rate, concurrency of 1, 4, striped vs. standard	RT RT
Random	EXB120 with 16 drives	Robot pick/place time, concurrency of 1, 4, striped vs. standard	RT RT
Random	EXB120 with 16 drives	Drive transfer rate, concurrency of 1, 4, striped vs. standard	RT RT

Table 5.1: Summary of simulations in the chapter; for the results column, abbreviation BW signifies that the result of the simulation is the aggregate bandwidth of a library, and RT signifies the mean response time of accesses.

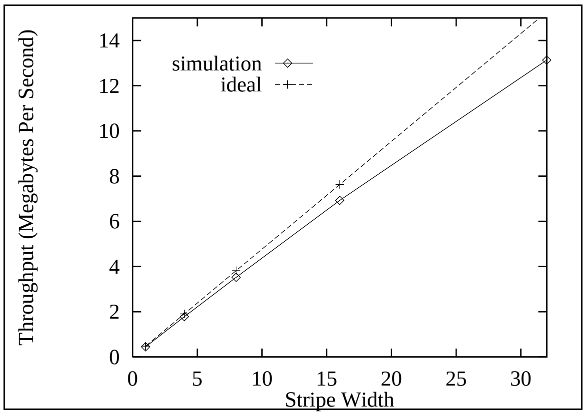


Figure 5.5: *Sequential Performance for an EXB120 Library with 32 tape drives. Shows that as data are striped over a greater number of tape drives, the aggregate bandwidth for the sequential workload increases.*

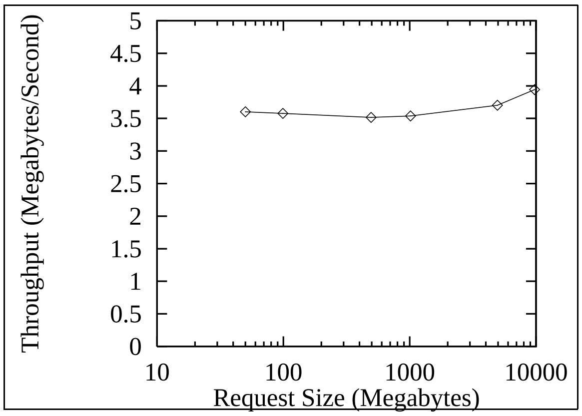


Figure 5.6: *Graph shows sequential performance for a variety of request sizes; EXB120 library with 8 drives.*

32 drives. The workload is made up entirely of sequential write operations. Request sizes are chosen according to an exponential distribution with a mean of 100 megabytes. A single process makes requests to the library; as soon as one request finishes, a new request begins. The graph shows that the aggregate bandwidth of the tape library increases linearly with the stripe width, since more drives can service each request in parallel. Thus, for a sequential workload, striping is very effective, and the wider the stripe, the better the performance.

In Figure 5.6, we show striped performance for different mean request sizes. The simulations are of an Exabyte library with 8 tape drives when requests vary in size from 50 megabytes to 10 gigabytes. The graph shows that, for large requests, the request size has little effect on sequential throughput. Data rate is determined by the number of drives over which data are striped.

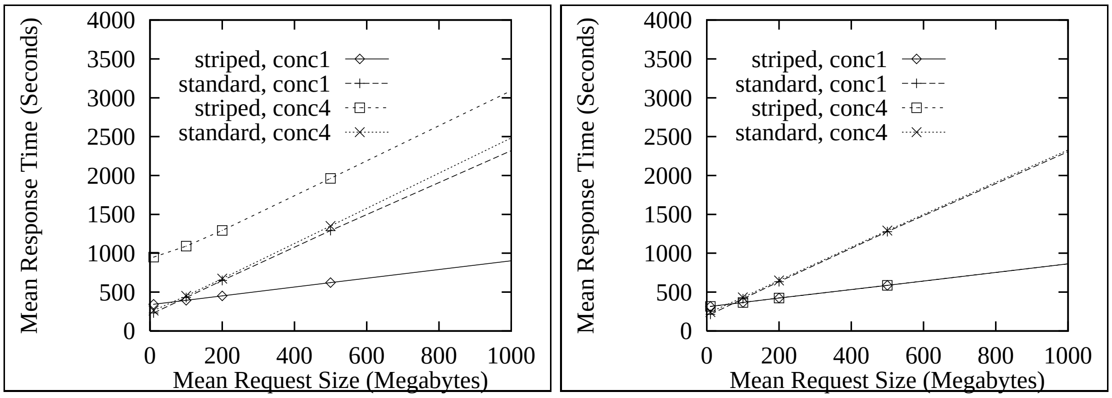
Both figures show the clear advantage of data striping in a sequential workload. Such a workload benefits dramatically from the greater throughput offered by several drives operating in parallel.

5.4.3 Performance for Random Workload with High Concurrency

Tape striping is less effective for workloads where there is a high concurrency of randomly distributed requests. In the simulations presented in this section, we assume that requests are uniformly distributed in the tape array. We also assume that several requests may be active at a time and that 75% of the operations are reads. We measure average response times, and find that striping is advantageous for a single outstanding request. However, with several outstanding operations, there is contention for the few tape drives in a tape library. When we alleviate this contention by adding tape drives, striped performance improves.

Figure 5.7(a) shows simulation results for an EXB120 robot. It shows striped and standard performance for a single active requests (concurrency = 1) and for four active requests. The vertical axis of the graph shows the mean response time, so lower values are desirable. The simulations use a request size that varies between 10 megabytes and 1 gigabyte. Request size is kept constant during a single simulation run. When there is a single outstanding request, striping is very effective at increasing throughput and reducing response time. However, when four requests are active, the response time of the striped array is much worse than the standard array because there is contention for the four tape drives in the system.

To understand why, consider our striped configuration. We are using a RAID level



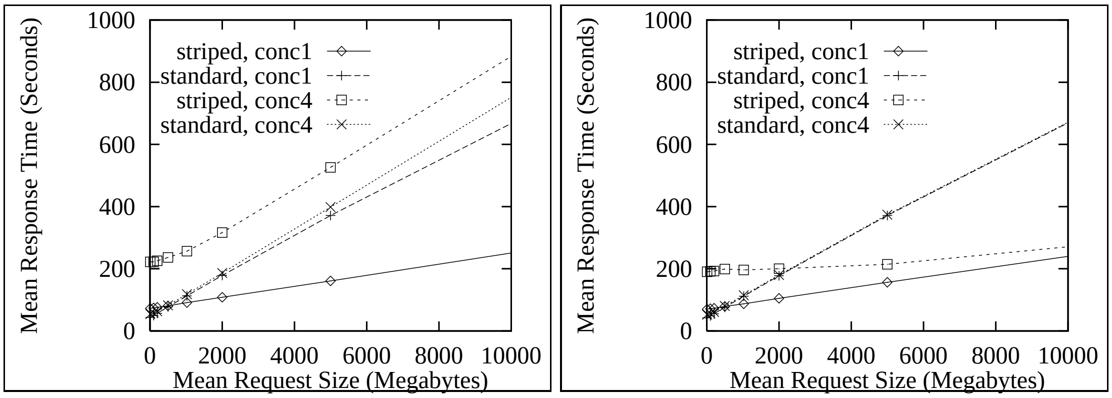
5.7a: *Response time (seconds) vs. request size for EXB120 with four tape drives.*

5.7b: *Response time (seconds) vs. request size for EXB120 with sixteen tape drives.*

Figure 5.7: *Performance of EXB120 robots with four and sixteen tape drives. (a) At concurrency of one, striping lowers response time compared to non-striped accesses, but at a concurrency of four, the response time of the striped system is worse than the non-striped. This is caused by contention for the four tape drives. (b) Increasing number of tape drives to sixteen alleviates contention for four outstanding requests; response time of striped system much lower than non-striped for request sizes greater than 200 MBytes.*

0 striping scheme with four tapes in each stripe group. This means that every striped read operation spans four tapes. Four active requests involve sixteen tapes at a time. Each of these tapes must be loaded to access data, but there are only four tape drives. The result is contention for the tape drives.

This dearth of tape drives is the primary factor limiting the usefulness of current tape robots for applications that must support a high concurrency of relatively random requests. Tape robots contain a massive amount of data on inexpensive tapes, but only have a few tape drives available for accessing this information. Figure 5.7b shows that contention for tape drives is alleviated when more drives are added to the robot. The striping configuration remains the same, with three data tapes and one parity tape, but now we simulate sixteen tape drives in the robot. With contention reduced, striping is effective even with four simultaneous requests, increasing system throughput and reducing response times. There is overhead associated with striping, since more cartridges must be



5.8a: *Response time (seconds) vs. request size for high performance library with four tape drives.*

5.8b: *Response time (seconds) vs. request size for high performance library with sixteen tape drives.*

Figure 5.8: *High performance library with four and sixteen tape drives. Shows striped and standard tape array performance at a concurrency of one and four outstanding requests. (a) Striped performance is better than non-striped performance at concurrency = 1 for requests over size 500 MBytes; at a concurrency of four, striped performance is worse than non-striped. (b) Increasing the number of tape drives to sixteen makes striping even at higher concurrencies attractive for large request sizes (over 2 GBytes in size).*

loaded into drives. In most workloads there is a threshold request size at which striping becomes effective. Below this threshold, tape switch time dominates, so standard systems that do fewer tape switches perform better than striped arrays. Above the threshold, data transfer time dominates and striping is advantageous. In Figure 5.7b, the point at which striping becomes effective is for requests larger than approximately 50 megabytes.

Figure 5.8 shows another example of contention for tape drives, this time for the high performance tape library described in Chapter 4. Figure 5.8a shows simulated performance of the high performance library with four tape drives. With one request active, striped response time beats standard response time for requests larger than 500 MBytes. However, at higher concurrencies, striped performance suffers, even for requests over 1 gigabyte in size. As in the EXB120, contention for the small number of tape drives causes this poor performance.

In Figure 5.8b, we show that increasing the number of tape drives alleviates this contention. However, the crossover point at which striping becomes effective at a concurrency of four active requests is approximately 2 gigabytes, a very large request size by most standards. Because this tape drive delivers high bandwidth (15 megabytes per second), the bandwidth requirements of many applications can be satisfied without striping. Since striped systems perform more tape switches and the native performance of the drives is high, only very large requests benefit from tape striping. Another problem with striping in this high performance library is the high cost of tape drives (over $100,000). This cost may make adding enough tape drives to alleviate contention prohibitively expensive. Because of this and because requests of size 2 gigabytes are uncommon except in some scientific computing applications, it is unlikely that striping would be chosen for this tape array for a candidate workload that includes concurrent, randomly distributed requests. (Tape striping will still perform well on sequential workloads.)

Because adding tape drives to an expensive, high performance library is infeasible, for the rest of this chapter, we focus on the less expensive Exabyte EXB120 robot.

5.4.4 Improving Tape Drives and Robots

In this section, we evaluate improvements to the EXB8500 8mm tape drive and the EXB120 robot. We explore how speedups to these components affect the performance of both striped and standard (or non-striped) tape libraries. Table 5.2 shows the set of parameters used to simulate these improvements. We simulate tape drives and robots that are 50% faster, twice as fast, and ten times as fast as the original devices. For each simulation, we vary the request size between 10 MBytes and 1 GByte, keeping the request size constant for individual simulation runs.

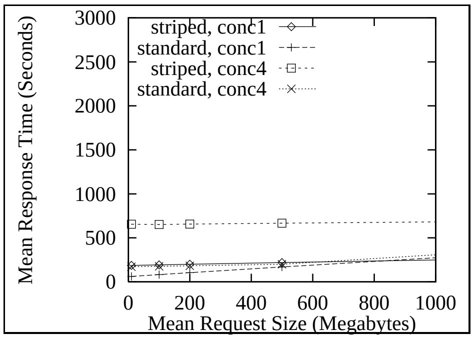
For the EXB120 robot with four tape drives, contention for tape drives limits performance regardless of whether tape drives, robots or both are improved. Figure 5.7a showed the performance of a standard EXB120 robot with EXB8500 tape drives. Figure

Speedup:	Original	50%	2X	10X
EXB8500				
Transfer Rate (KB/sec)	470	705	940	4700
Search Rate (MB/sec)	36.2	54.3	72.4	362.0
Search Startup (sec)	12.5	8.3	6.25	1.25
Rewind Rate (MB/sec)	42.0	63.0	84.0	420.0
Rewind Startup (sec)	23.0	15.3	11.5	2.3
Load Time (sec)	65.5	43.3	32.5	6.5
Eject Time (sec)	16.5	11.0	8.25	1.65
EXB120				
Robot Arm Load (sec)	21.4	14.3	10.7	2.1
Robot Arm Unload (sec)	19.5	13.0	9.8	2.0

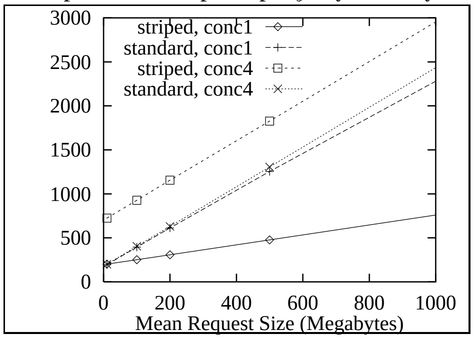
Table 5.2: *Simulation parameters for tape drives and robots at various speedup factors.*

5.9 shows how the mean response time changes when tape drives, robots, and both are ten times as fast as currently available devices. In each case, although the mean response time is reduced, striped performance at a concurrency of four is consistently worse than non-striped performance. The contention caused by the small number of tape drives is not alleviated simply by speeding up the tape drives or robots. More tape drives are needed in this system for striping to be effective for loads other than a single outstanding request.

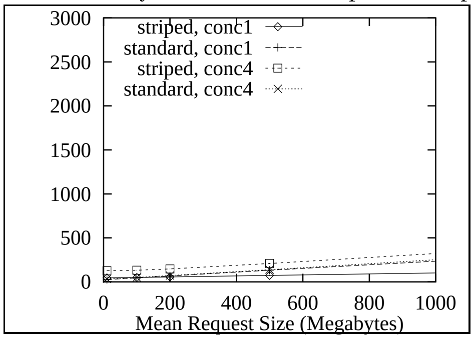
Consequently, for the rest of this section, we consider only EXB120 robots with 16 tape drives. Figure 5.10 shows the effect of speeding up both tape drives and robots. Improvements in tape drive and robot speeds dramatically reduce response times. Figure 5.10c shows an order of magnitude improvement in response when the tape drives and robot are sped up by a factor of ten; performance before the speedup was shown in Figure 5.7b. In Figure 5.10c, the request size threshold at which striping is effective is about 200 megabytes for a concurrency of 1 and 500 megabytes for a concurrency of 4. Note that as tape drives and robots speed up, going from Figure 5.10a through 5.10c, the crossover threshold increases. Both striping and increased drive bandwidth reduce data transfer time. As the native bandwidth of a drive increases, the need for striping at a particular request size decreases.



5.9a: *Response time (sec) when tape drive is sped up by a factor of ten but robot speed is unchanged.*



5.9b: *Response time (sec) for ten times faster robot but tape drive speed unchanged.*



5.9c: *Response time (seconds) when both tape drive and robot are sped up by a factor of ten.*

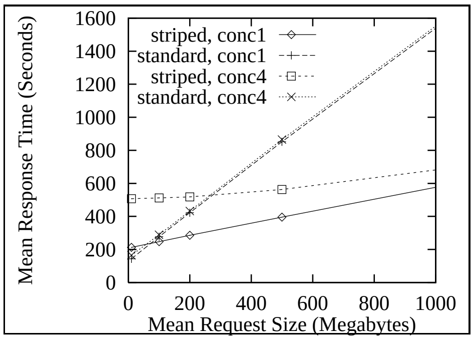
Figure 5.9: *Improvements in library performance for EXB120 robot with four EXB8500 tape drives. In each case, mean response time improves, but striped performance remains inferior to non-striped performance at higher concurrencies. The small number of tape drives generates contention that cannot be resolved by speeding up the tape drives or robots.*

Response time (seconds)	**Concurrency = 1**		**Concurrency = 4**	
	standard	striped	standard	striped
100 Megabyte requests				
EXB120 with 16 drives	415	367	432	750
Same, faster drives and robots	44	46	47	109
1 Gigabyte requests				
EXB120 with 16 drives	2355	876	2377	1020
Same, faster drives and robots	238	97	240	130

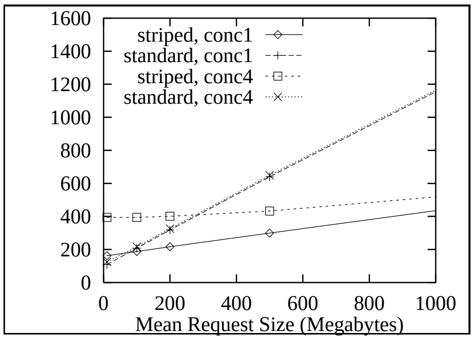
Table 5.3: *Comparison of response times before and after ten times speedup for tape drives and robots.*

In Table 5.3, we compare response times for requests of size 100 megabytes and 1 gigabyte before and after the ten times speedup. We choose these requests sizes because, as we saw in Chapter 2, the overhead of a tape switch operation is high; it is likely to be difficult in an EXB120 to get much benefit from striping at request sizes much lower than 100 megabytes, since access time will be dominated by tape switch time. The response time of a 1 gigabyte request, which is dominated by data transfer time, should be greatly reduced by striping.

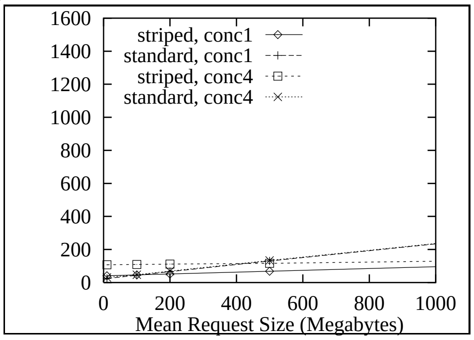
The response times shown in Table 5.3 are reduced by an order of magnitude after the ten times speedup of tape drives and robots. For the 100 megabyte request, comparing standard and striped response times in a single row of the table shows relatively little benefit from striping either before or after the speedup, and there is actually a penalty for striping at a concurrency of 4. As expected, striping doesn't offer much of a throughput benefit for this workload because the data transfer portion of the access is not long compared to the tape switch time. At higher concurrencies, the extra tape switches hurt striped performance. For the 1 gigabyte requests shown in Table 5.3, striping helps response time significantly both before and after the ten times speedup. Before the speedup, striping cuts response by more than half, reducing response time by more than 20 minutes on average. After the ten-fold speedup, striping still cuts response time by about 50%. However, the faster devices have greatly reduced response times; the resulting improvement in response time is only about



5.10a: *Response time (sec) for EXB120 with 16 drives when drives and robot are sped up by 50%.*



5.10b: *Response time (sec) for EXB120 with 16 drives when drive and robot speeds double.*



5.10c: *Response time (sec) for EXB120 with 16 drives when drive and robot speeds increase by factor of 10.*

Figure 5.10: *Array performance for EXB120 library with sixteen EXB8500 tape drives, when both tape drives and robots improve in performance. Striped performance is consistently better than non-striped performance when drive and robot performance scale at similar rates.*

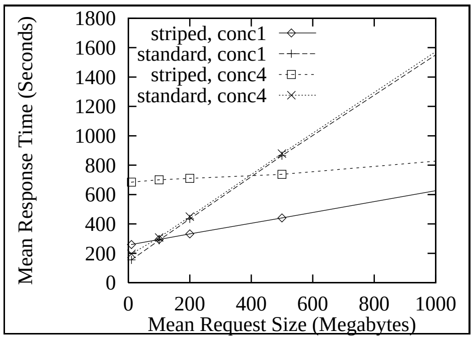
Response time (seconds)	**Concurrency = 1**		**Concurrency = 4**	
	standard	striped	standard	striped
100 Megabyte requests				
EXB120 with 16 drives	415	367	432	750
Same, faster drives, unchanged robots	76	174	157	**587**
1 Gigabyte requests				
EXB120 with 16 drives	2355	876	2377	1020
Same, faster drives, unchanged robots	273	228	288	**619**

Table 5.4: *Comparison of response times before and after ten times speedup for tape drives; robot speed is unchanged.*

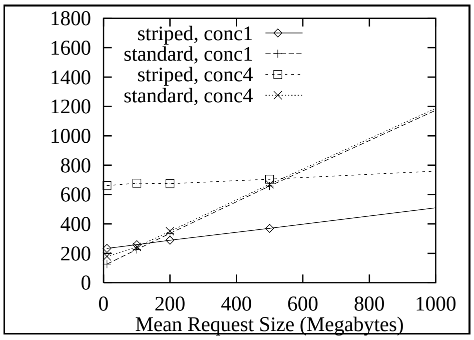
2 minutes, compared to 20 minutes before the speedup. This bolsters our contention that the benefits of striping, and thus the incentive to do striping, will decrease as tape drives and robots get faster.

Next, we show that improving the speed of the tape drive is more important to response time than speeding up the robot; however, if the robot arm is ignored, eventually it does become a performance bottleneck. Figure 5.11 shows the performance of the EXB120 robot with 16 tape drives when the tape drive speed increases by factors of 50%, 200% and 1000% but the robot speed is unchanged. In Figures 5.11a through 5.11c, as the tape drive speed increases, overall response times decrease dramatically. This suggests that most of the performance gains seen in Figure 5.10 from speeding up both tape drives and robots were actually due to the improved tape drive speed. However, in Figure 5.11c, striped performance at both a concurrency of 1 and 4 is worse than standard performance. The reason for this is that overall performance is limited by contention for the robot arm.

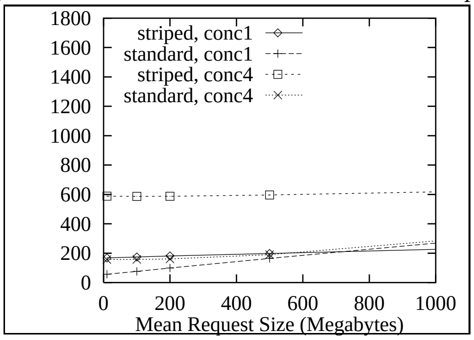
Table 5.4 confirms that contention for the robot arm limits performance. Like Table 5.3, it presents response times for striped and standard tape arrays for request sizes of 100 megabytes and 1 gigabyte. It shows performance before and after a ten-times speedup in the tape drive; robot arm speed is unchanged. After the tape drive speedup, striped performance is worse than standard performance except for 1 gigabyte requests with a concurrency of 1. The reason for the poor striped performance is contention for the robot



5.11a: *Response time (sec) for EXB120 with 16 drives when drive is 50% faster; robot unchanged.*



5.11b: *Response time (sec) for EXB120 with 16 drives when drive speed doubles; robot unchanged.*



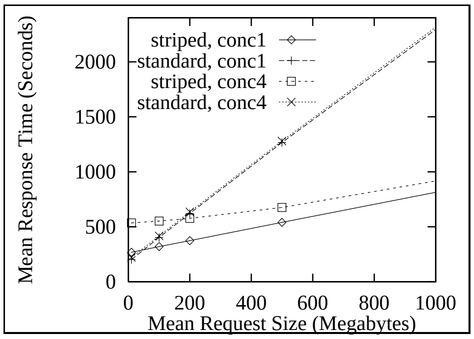
5.11c: *Response time (sec) for EXB120 with 16 drives when drive speed increases by factor of 10; robot unchanged.*

Figure 5.11: *Array performance for EXB120 library with sixteen EXB8500 tape drives, when tape drives improve in performance. When tape drive speedup reaches ten times normal EXB8500 performance, striped performance becomes worse than non-striped performance. For the striped system, performance is limited by the robot arm, which must perform more context switches than the non-striped system.*

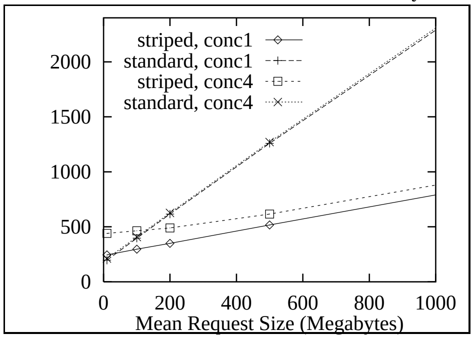
arm. Because the tape drive is fast, the robot arm component of the tape switch operation becomes more important. Since striped arrays perform many more tape switches, their performance suffers compared to standard arrays. The workload of 1 gigabyte requests with a concurrency of 1 is an exception because the single active request does not require so many tape switches, and unlike the 100 megabyte requests, the 1 gigabyte requests are large enough that data transfer time is still relatively significant.

To illustrate this robot arm contention, we show two numbers in bold print in Table 5.4. For both 100 megabyte and 1 gigabyte requests at a concurrency of four, the response time averages over 500 seconds. In each case, this is much longer than the corresponding response time for a standard tape library. The reason in each case is that performance is limited by the large number of robot operations that must be performed sequentially. Recall that each of four outstanding requests uses four tapes, so there are sixteen tapes in active use. Each of these must be loaded by the robot arm, an operation that takes approximately 40 seconds. As a result, mean response time is around 600 seconds for both the 100 megabyte and 1 gigabyte requests. Since a striped tape library performs more tape switch operations than a non-striped or standard tape library, the robot arm is used more heavily in the striped system. The robot arm eventually becomes a performance bottleneck. This suggests that speeding up the tape drive alone is not enough. Eventually an Amdahl's Law effect is evident with performance becoming limited by the component in the system that has not improved.

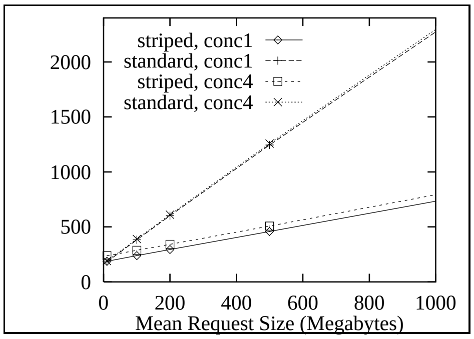
Finally, Figure 5.12 shows that speeding up only the robot arm without changing the speed of the tape drive has little effect on response time for a standard library but significantly reduces striped response times. Table 5.5 shows response times for 100 megabyte and 1 gigabyte requests. The faster robot is most beneficial for striped accesses at a concurrency of four. This is because such workloads require sixteen tapes to be active at any given time, necessitating many tape switch operations. Making the robot faster



5.12a: *Response time (sec) for EXB120 with 16 drives with 50% faster robot; drive speed unchanged.*



5.12b: *Response time (sec) for EXB120 with 16 drives when robot speed doubles; drive speed unchanged.*



5.12c: *Response time (sec) for EXB120 with 16 drives when robot speed increases by factor of 10; drive speed unchanged.*

Figure 5.12: *Array performance of EXB120 library with sixteen EXB8500 tape drives with faster robot. Improving robot speed has little effect on response time. It does reduce the performance gap between workloads of different concurrencies in a striped system.*

Response time (seconds)	**Concurrency = 1**		**Concurrency = 4**	
	standard	striped	standard	striped
100 Megabyte requests				
EXB120 with 16 drives	415	367	432	750
Same, unchanged drives, faster robots	383	239	390	288
1 Gigabyte requests				
EXB120 with 16 drives	2355	876	2377	1020
Same, unchanged drives, faster robots	2321	746	2343	806

Table 5.5: *Comparison of response times before and after ten times speedup for robots; tape drive speed is unchanged.*

reduces tape switch times and has a relatively large impact on performance. By contrast, a standard tape library performs fewer tape switch operations, and its performance is not much affected by improvements in robot speed.

We have shown that for tape libraries to perform better on workloads with a high concurrency of random requests, they need more tape drives and faster tape drives and robots. Most important is that the tape library contain enough tape drives to handle the expected concurrency of the workload. Next in importance is the speed of the tape drive. Finally, robot speed cannot be ignored because it eventually limits overall performance, especially for striped tape arrays. We also showed that as tape drives speed up, striping will be less effective.

5.4.5 Hypothetical Robots

Finally, we evaluate the effect of a higher ratio of tape drives to tapes by simulating two hypothetical tape libraries, and we find that striped performance benefits from this greater ratio in an unexpected way. We simulate libraries containing 24 and 40 EXB8500 tape drives. Each robot is identical to the EXB120 robot except we simulate a faster robot arm that can perform a pick or place operation in 5 seconds rather than 20. Each library contains 116 tapes, giving us a tape-to-tape-drive ratio of 5-to-1 in the 24-drive robot and 3-to-1 in the 40-drive robot. Figures 5.13 and 5.14 show the performance of the two robots.

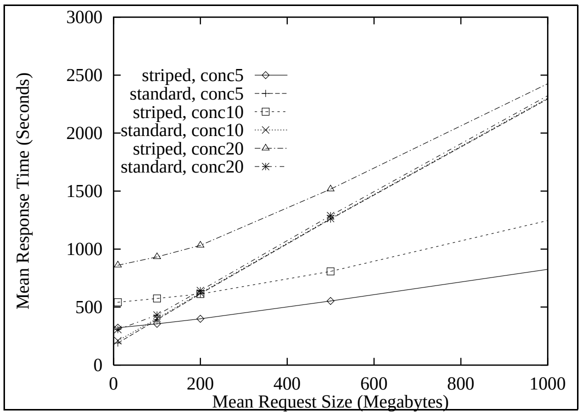


Figure 5.13: *Performance of hypothetical robot with 24 drives; striped performance superior to non-striped for concurrency up to 10.*

For the 24-drive robot, we see in Figure 5.13 that striped performance at a concurrency of 10 is better than nonstriped for request sizes larger than 200 megabytes, but at a concurrency of 20, striped performance is always worse than non-striped. Similarly, in Figure 5.14, 20 concurrent requests benefit from striping if request sizes are over 350 megabytes. In both cases, striping is advantageous for a higher concurrency than would be expected strictly based on the number of tape drives in the system. We would expect that striping would be beneficial in the 24-drive system for up to 6 concurrent accesses, and in the 40-drive system for up to 10 accesses. With a higher ratio of tape drives to tapes, a larger proportion of the total tapes are always loaded into the tape drives, making it more likely that they can be re-accessed before they must be switched out. For example, with 40 tape drives, about a third of all tapes are loaded into tape drives at any time. If tape drives are scheduled intelligently, so that no tape is switched out if a new access is pending, the tape library supports a higher concurrency of requests.

5.4.6 Improving individual properties of tape drives and robots

In this section, we examine the effect on striped and standard tape array performance when we vary four tape drive and robot parameters. All these simulations use

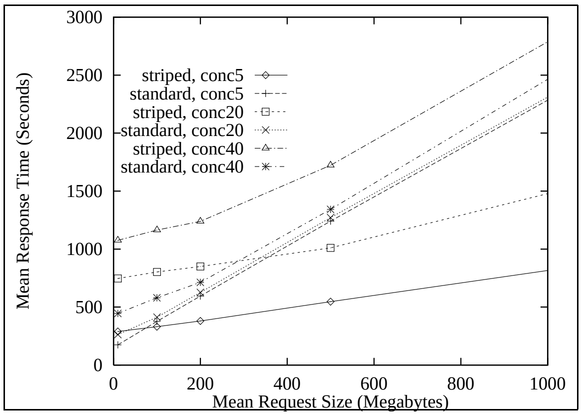


Figure 5.14: *Performance of hypothetical robot with 40 drives; striped performance superior to non-striped for concurrency up to 20.*

an EXB120 robot with 16 tape drives. We vary a single simulation parameter at a time, keeping the others constant at the measured values described in Chapter 4. Request size is held constant at 200 MBytes. We simulate concurrencies of 1 and 4 outstanding requests.

First, we study the effect of tape drive bandwidth; we simulated drives with improved transfer rates and found that standard systems benefit more from the higher transfer rate than striped systems. Figure 5.15 shows that striped performance at a concurrency of four experiences contention, while the non-striped system does not. In the non-striped system, the transfer rate eventually becomes high enough that data transfer time is negligible; the response time of an access corresponds to the tape switch time (284 seconds for an EXB120 robot). By contrast, in the striped case, the 200 MByte access is split into several tape accesses, and the system experiences contention for the tape drives.

Next, we evaluate the impact of robot grab (pick) and insert (place) operations. Figure 5.16 shows that a slow robot with long pick and place times hurts striped performance more dramatically than standard performance. This is because more tape switches are required in a striped system, and an increase in pick/place time slows every tape switch operation.

Third, we investigate the benefits of fast search and rewind positioning rates on the

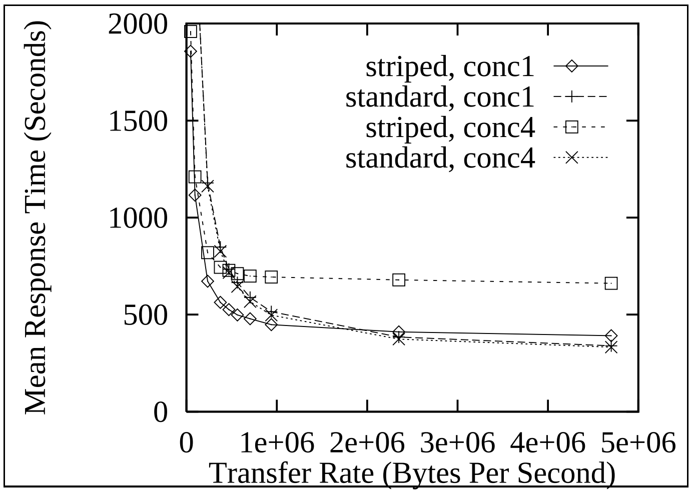


Figure 5.15: *Response time vs. Tape Drive transfer rate. Comparison of striped and non-striped performance when varying tape drive transfer rate. Performance of the non-striped system improves more than the striped system because the striped system experiences more contention.*

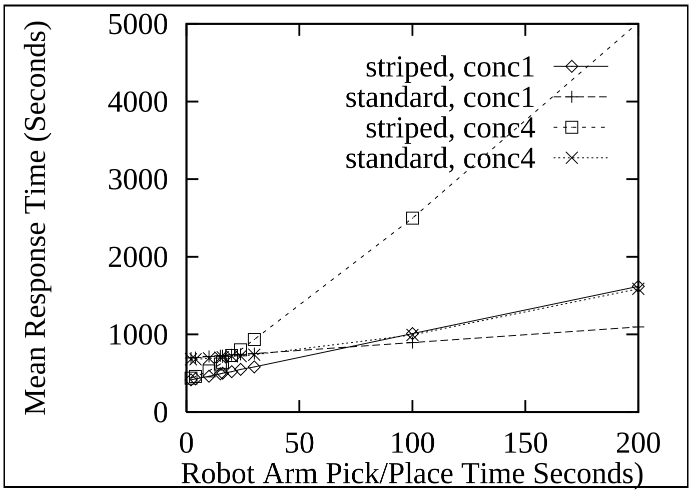


Figure 5.16: *Response time vs. Robot arm speed. Comparison of striped and non-striped performance when varying robot arm speed. Response time decreases as robot arm gets faster; striped performance is more dramatically affected, since every access involves several tape switches.*

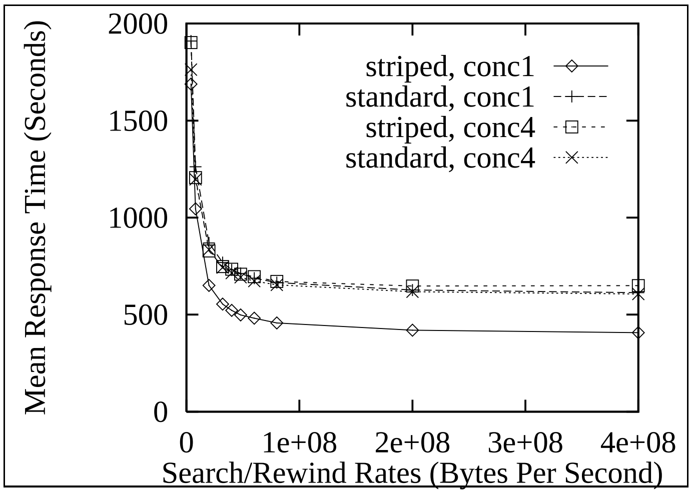


Figure 5.17: *Response time vs. Search/rewind rate. Comparison of striped and non-striped performance when varying tape drive search and rewind rates. After search and rewind rates exceed 100 MBytes/sec, performance levels off, indicating that some other parameter limits performance.*

drive. Figure 5.17 shows that higher search and rewind rates on the tape drives reduce the penalty for tape switches. As a result, a faster robot helps striped response times more than standard, since striped systems perform more tape switches. (This is most clearly seen at a concurrency of one.) In these simulations, we use a single value for the search and rewind rates for simplicity, although in reality, the rewind rate is slightly faster. We leave the search/rewind startup overheads at their original values. Most of the benefits of increasing the search/rewind rates are achieved by the time those rates reach 100 MBytes/sec; after this, performance is limited by other parameters.

A final simulation result, not shown in a graph, is that increasing the capacity contained in a single tape increases the response time for both striped and standard arrays. Since search and rewind rates are constant, positioning operations for longer, higher-capacity tapes are more time-consuming. To minimize search and rewind times, shorter tapes with lower capacities are ideal. Unfortunately, this conflicts with the need for high capacity in a massive storage system.

In this section, we evaluated improvements to individual device characteristics;

in the last section, we examined overall improvements in tape drives and robots. From these simulations, we can make several recommendations about improving the performance of tape libraries. Most importantly, tape libraries need a larger number of tape drives. A higher ratio of tape drives to tapes would make it possible for these systems to support more concurrent accesses. We also conclude that it is more important to improve tape drive speed than robot arm speed. For individual tape drives, the most effective performance improvement comes from increasing drive transfer rate. This improvement proportionally helps non-striped systems more than striped libraries because striped libraries perform many more tape switch operations, and tape switch time is not affected by improved transfer rate. Other desirable changes for tape drives are faster search and rewind operations. Robot arm speed, while less of a bottleneck than tape drive speed, cannot be ignored; eventually, as tape drive performance improves, a slow robot limits overall performance. Poor robot arm speed harms striped performance more than non-striped since striped libraries use the robot arm more frequently to perform tape switch operations.

5.5 Summary

In this chapter, we have used the simulator described in Chapter 3 to compare the performance of striped and non-striped systems and to predict the performance of large tape libraries with a variety of improvements.

Tape striping performs well for important sequential workloads like backup and archiving but poorly for randomly-distributed workloads with a high concurrency of requests. This is because striping creates contention for the small number of tape drives in a typical tape library. For both mid-range and high performance tape libraries, increasing the number of tape drives in the system alleviates this contention and makes striping more attractive. This strongly suggests that the success of striping in large tape libraries will depend on including an adequate number of tape drives to handle the expected workload.

We next examined the effect on performance of speeding up tape drives and robots by certain factors. We discovered that speeding up the drive without speeding up the robot arm eventually hurt striped tape array performance compared to standard performance, since the striped system uses the robot arm more heavily. Speeding up only the robot arm helps striped performance considerably but has little effect on non-striped response times. Scaling both tape drive and robot performance at comparable rates keeps striped performance superior even for tape drives and robots faster than current devices by a factor of ten.

Finally, we examined the effects of changing particular tape drive and robot characteristics. Increasing the tape drive's transfer rate doesn't benefit striped performance as much as non-striped because of contention for the tape drives. Slower robot arms have a more harmful effect on striped than non-striped systems, since the former generate more tape switches and utilize the robot arm more heavily. Search and rewind rate improvements reduce the tape switch penalty that is so harmful to striped performance. Increasing tape capacity causes an increase in average response time because of the extra time required to do search and rewind operations.

It should be noted that although the performance benefits of striping are limited to sequential and low concurrency applications, there are also reliability benefits to be gained from striping. Incorporating a parallel error correction code across the tapes in a library offers protection against uncorrectable tape errors and may eliminate the need for read-after-write checking that wastes significant time and tape capacity. Quantifying the reliability benefits of striping in a tape library is left for a future study.

In Chapter 4, we showed that the performance of individual drives in a tertiary library is inadequate for supporting multimedia workloads. In this chapter, we demonstrated that data striping is not an effective technique for improving tertiary library performance on similar applications. In the next two chapters, we evaluate the use of storage hierarchies including tertiary storage for multimedia applications. Chapter 6 predicts video server and digital library workloads, and Chapter 7 evaluates the performance of disk farms and storage hierarchies on a movies-on-demand video server workload.

Chapter 6

Toward a Workload Characterization for Video Servers and Digital Libraries

6.1 Introduction

In this chapter, we characterize the workloads of two emerging multimedia applications. *Video-on-demand* service will allow customers to request that certain movies or television shows be played on their television sets immediately, giving individuals direct access to the materials stored in a *video server*. *Digital Libraries* will allow users to browse through large multimedia databases that include text from many sources, still images, video clips, and perhaps interactive access to applications such as word processors, simulation packages, drawing programs and games. Both video-on-demand and digital library services are expected to become commonplace in the next decade.

Information servers that support these applications will provide access to massive amounts of storage. For example, each movie stored using a typical MPEG-II compression format consumes over 2 gigabytes of storage. The Mead Data Central database contains over 2 terabytes of plain text information [117]; this figure would increase dramatically if the database also included bit-map images of pages, still images and video clips. To design effective storage systems for video server and digital library applications, computer architects need accurate models of the application workloads.

Unfortunately, due to the fiercely competitive nature of these emerging industries [121], we have found it difficult to obtain traces of existing systems or predictions of future workloads, and this lack of information is likely to continue for several years. In the absence of measured workload specifications, storage systems designers need to make a "best-guess" estimate of the workloads for these applications. In this chapter, we present our predictions for these workloads, together with the evidence we have gathered to support those predictions. In Section 6.2, we describe the model we use to characterize the application workloads. In Section 6.3, we describe our model of accesses to a video server, and in Section 6.4, we characterize the workload for a digital library system.

6.2 A Workload Model

In this section, we define several parameters of the workload model that will characterize the video-on-demand and digital library workloads. Table 6.1 lists the parameters: the number of objects managed by the storage system, the size of those objects, the request size distribution, the locality of accesses to the storage system, the number of users active in the system at peak loads, the response time goals of the system, and the throughput required by each user of the system. In this chapter, we concentrate on the workload at the application level rather than at the file system or device level; depending on the storage system design, the logical request patterns we identify might be translated by the file system and device controllers into a variety of different patterns of access to physical devices.

An *object* in a video server system might be a movie, a television show, or a taped

Workload characteristics
Object size
Number of objects
Block Size Distribution
Locality of accesses
Peak Number of Active Users
Response Time Requirement
Throughput per Active User

Table 6.1: *Summary of workload characteristics for estimates of video server and electronic library workloads.*

lecture. In an electronic library system, an object might be a journal article, a book, a video clip or an image. *Object size* is determined by the nature of the application, by the text or image format, and by the compression scheme implemented. The *number of objects* is determined by the needs of the application and the cost and capacity of the storage components.

The *request size distribution* characterizes user requests to the storage system. For example, in a video-on-demand system, users typically request that they view an entire movie sequentially, except for occasional VCR-like operations such as pause and rewind. In a digital library, request sizes will vary based on the types of searches performed and the information retrieved. Storage system designers use request size information to determine the optimal layout and retrieval sizes of data on physical storage devices.

We must also characterize the *locality* of requests to the storage server. Locality is a measure of the skew or unevenness of the distribution of accesses to a storage system. A highly-localized distribution suggests that some data objects are more popular and more frequently accessed than others. Locality is frequently described as either *temporal* (objects accessed by a storage server are likely to be accessed again soon) or *spatial* (when an object is accessed, objects stored near it on physical devices are likely to be accessed soon.) Since we are concerned here with the application-level workload and not with data layout on particular storage devices, we focus on temporal locality.

Locality influences system design by suggesting how effective caching strategies can be in the storage server. Especially in video servers, there is likely to be a high degree of locality, with a large percentage of requests going to a small number of movies. This suggests that caching or hierarchical storage layouts will be effective for video servers. It is likely that digital libraries will also exhibit locality, especially in accesses to the index files of the database [42]. Mead Data Central reports infrequent re-use of data, however, making caching ineffective [117].

The expected *peak number of active users* greatly influences storage system design; it determines a minimum level of resources that must be provided to service requests. Many information servers implement an admission control scheme, which restricts the entry of new users into the system if certain response time constraints cannot be met. A related workload characteristic is the system's *response time guarantee* or goal. In an digital library system, the response time goal might be quite short, for example, at most ten seconds to complete a search operation. A strict admission control policy might not be necessary in such a system, however, since increased load will result in slower search operations, but will not change the quality of the search results. By contrast, a video-on-demand server requires less interactive operation, so the response time constraint might be more flexible, ranging from a few seconds to a few minutes to start a movie. Admission control in a video server is likely to be much more strictly enforced, however, since each user must receive a guaranteed level of bandwidth after the performance of a movie has begun.

Finally, an important workload characteristic is the amount of *throughput* or *bandwidth* that will be delivered to each user. This value is determined by the nature of the application, the hardware delivering the data, and the compression scheme chosen for the implementation.

6.3 Video-on-Demand Servers

Video-on-demand service promises huge profits to the companies whose implementations are successful. United States consumers spent approximately $29 billion in 1992 on movie theater tickets, video rentals and cable television [3]. Because of the potential profits, companies are often protective of their experience with video server prototypes and their predictions regarding future workloads. Several prototype systems exist, including video-on-demand services available in hotels from companies like Spectradyne and cable television-based prototypes in several cities. Little information is available regarding the workloads observed in these prototype systems; moreover, many of these prototypes serve small user communities whose usage patterns are unlikely to apply to larger populations.

In this section, we describe several video-on-demand prototype systems. We then characterize the video-on-demand workload at the application level.

6.3.1 Background

Chang et al. [17] describe the components of a video-on-demand system, from a video server through a distribution network to the *set-top box.* The set-top box displays video to a television set or computer monitor in a home or office. *Local video servers* are located near customers in telephone, cable television or other network switching offices. *Remote video servers* are reached over a backbone network that links geographically-distributed regions.

In our work, we focus on the design of video servers and, in particular, on their storage systems. Video may be stored in the server on RAM, on magnetic disk, on optical or magneto-optical disk, and on magnetic tape. In response to requests from customers to see particular videos, the server is responsible for determining whether there are enough resources to support the new customer (admission control), fetching the data from storage (from either local devices or at remote servers over the network), setting up network connections to guarantee adequate stream bandwidth to the customer, packaging the data (perhaps including data encryption), and sending the data over the network to the customer's set-top box. The server may also be responsible for supporting VCR-like operations such as pause and rewind that are initiated by the customer.

In this section, we describe several video-on-demand prototype systems. First, we describe four systems that exclusively use magnetic disk to store video data: the IBM Almaden Shark/Tiger Shark projects, the Silicon Graphics/Time Warner prototype, the Starlight Networks video server, and the Microsoft Tiger system. Second, we survey three proposed systems that would include a storage hierarchy including both magnetic disks and magnetic tape to store and deliver video. These systems are proposed by Bell Communications Research (Bellcore), the University of California at Berkeley, and IBM Yorktown.

IBM's Almaden Research Center has developed two video server prototypes called Shark and Tiger Shark [35], [36]. Shark was used for a video-on-demand trial in Alexandria, VA with Bell Atlantic, and is being used in a similar trial in Hong Kong with Hong Kong Telecom. The latter system sustains 150 simultaneous MPEG-1 streams over ADSL, a communication scheme that allows transfer rates of 1.5 Mbits/second over existing twisted-pair telephone lines. The system includes a 16 kilobyte back channel used for interactive control. The video server holds approximately 100 movies and serves 300 to 400 customers. The set-top box, built by IBM, bundles the decompression and frame buffer hardware. The second prototype is named Tiger Shark because the server uses data striping or interleaving (described in Chapter 5) to lay out data on disk. Tiger Shark is designed to scale to a large size using very wide striping. Placement of the interleave units on the disk is random to maximize load balancing among the disks. The system will include a large amount of parallelism for a high degree of fault tolerance; because of the current high cost of RAID (redundant arrays of inexpensive disks [80], [116]), the system will use mirroring or duplication of data on disks to recover from disk or other component failures.

Silicon Graphics, Inc., is building a video-on-demand prototype system that will be implemented in Orlando, Florida, with Time Warner Inc.[76]. The SGI video server system is actually a collection of some number of smaller video servers, each containing a multiprocessor, a medium-sized array of disks, and network hardware. The collection of video servers will contain 600 gigabytes of magnetic disk storage and serve 4000 viewers [8]. It is expected that some large fraction (perhaps up to half) of the homes might be watching movies during peak periods. Each video server will hold approximately 250 movies, including several copies of popular movies to satisfy high demand. The response time constraints are about ten seconds to deliver a movie, and much faster (at most 250 milliseconds) to give the user a response during interactive communication used to select movies and agree to payment.

Tobagi [111] describes the disk-based video-on-demand server from Starlight Networks, Inc. This 486-based server supports at least 20 simultaneous users by striping data over a minimum configuration of three disks. Tobagi describes the design tradeoffs between latency requirements, the number of clients supported, the size of accesses to the disk for each client, and the amount of buffer memory required in the system. In this server, clients are serviced in a round-robin fashion, with enough data being read for each stream to satisfy the bandwidth constraints of that stream during each cycle. Tobagi notes that amount of storage accessed per client per cycle increases, the number of users that can be supported increases. This is because disks are used more efficiently, with each seek operation amortized over a longer data transfer operation. However, both the memory required for buffering and the latency to deliver video increase as the disk access size grows. Memory requirements increase because larger amounts of data must be buffered for each user during each cycle. Latency increases because each round-robin cycle to service a collection of users becomes longer as the period of data transfer increases; new arrivals have to wait longer to be serviced.

Microsoft has developed the Tiger Media server that uses off-the-shelf, commodity-priced computers and disks to provide low-cost video service [87]. Software manages video streams and fault-tolerance. Movies are "load-balanced" (striped) across disks. Access time for the system is less than one second, and the servers support all virtual VCR functions. The system is highly fault-tolerant, containing no single point of failure, and performs automatic replacement of failed components with hot spares. The system contains a collection of disk servers, called Cubs, managed by a controller machine, called Tiger, that send data through ATM switches to video redistributors (cable companies, phone companies); in turn, the redistributors deliver video to a set-top box or computer in a home or office. A 3-Cub system can support up to 26 video streams of 4 megabits per second, even if one of the Cubs fails. The manufacturers of Tiger systems are Compaq and Intel, and the systems will be used in partnerships in the United States with TCI Cablevision, in Canada with Rogers Cable and in Japan with NTT.

One last disk-based video server has been announced by Oracle and NCube. The proposed system will use a high-end NCube multiprocessor to service 10,000 active peak streams. We have received few technical details of this system.

Bell Communications Research has built a video server prototype that uses a storage hierarchy [45]. The video server is a workstation with an array of nine disks and a magnetic tape stacker (a small robotic device). The video server sends data to the equivalent of a telephone company central office. The data rate from the server to the central office is 150 Mbits/second, or enough to support 100 MPEG-1 streams at 1.5 Mbits/second. For each stream being supported on the network, there is a line card that contains most of the intelligence in the system. The line card buffers data from a video server and sends it across the network at the appropriate data rates to a set-top box. The line card contains two processors, one to handle communications with the server and one to handle communications with the customer. The card also contains 8 megabytes of RAM, or enough to hold about 40 seconds of MPEG-I data, which is used for speed-matching between the server and customer and which greatly simplifies synchronization. The prototype includes three set-top boxes: one built by Bellcore, a personal computer, and a Phillips CD Interactive device. The connection to the home is assumed to be ADSL running on copper wires at 1.5 megabits/sec. The disk array uses RAID level 3 with data striped in 5-second intervals. The reverse channel is a 9600 baud RS232 line. One of the key goals of the prototype is to put intelligence in the line cards, allowing video servers to be relatively simple and inexpensive. Although the video server prototype is hierarchical, they report that the performance of the tape stacker portion of the hierarchy is inadequate.

In 1991, Sincoskie from Bellcore proposed a hierarchical video server [99]. Sincoskie describes a video database that would be located or closely connected to a local central switching office, supporting approximately 10,000 customers. The proposed system would have asynchronous transfer mode (ATM)-based multi-cast connections. The video server storage architecture that Sincoskie describes consists of three levels: a main-memory buffer for each user; a fairly large magnetic disk farm, called “copier memory”, that serves as a cache for the most popular movies; and finally, the “library”, a magnetic tape or optical disk jukebox that holds less actively requested movies. If a customer requests a movie that no one else is watching, and the system does not predict that the movie is likely to be requested again soon, a connection is established directly between the tape (or optical disk) drive and the viewer; unpopular movies need never be cached on magnetic disk. In this proposed video server, movies are always read from the tertiary storage library in their entirety. This is sensible because of the severe penalty paid for switching tapes; reading the entire movie minimizes the number of switch operations.

Federighi and Rowe [28] describe the design and initial implementation of a distributed hierarchical storage manager for video-on-demand. Rather than simply servicing movies-on-demand, this system is intended to support complex queries to a vast repository of video information. The scheme envisions all movies stored on centralized, tertiary archives; many videos will also be cached locally on disk-based servers to reduce response times and guarantee real-time delivery once playback begins. To accommodate the long delays of tertiary storage, the system allows users to supply hints that enable the staging of data with a high likelihood of use to local disk servers. The system provides a single name space for all video objects and plans to offer software-based striping of movies across disk servers. The distributed video server allows users to publish video in a variety of fomats, encodings and fidelities and to set priorities for video requests.

Finally, a group at IBM T. J. Watson Research Center has done work on hierarchical storage systems [109], [61]. This group comes to some conclusions similar to those described in Chapter 7. Studying the disk portion of the storage hierarchy, they conclude that to support many simultaneous viewers, striping of the data among a collection of disks is necessary. They propose algorithms for staging data from the tape archive into the disk system, including starting the playing of the movie to the viewer as soon as the first blocks are retrieved from tape and discarding movie blocks as soon as they have been sent to the viewer. For tape libraries, they conclude that the number of tape drives is critical to performance. They propose a cost model to evaluate storage components for video-on-demand and find that disk-based video servers today cost about \$200 per stream, optical-disk based approximately \$500 per stream, and tape library-based up to \$50,000 per stream. They suggest that for a large number of viewers (perhaps 100,000), storing a movie in solid-state disk is economical, about \$100 per stream. (In Chapter 7, we use a different methodology and arrive at similar stream costs: approximately \$240 for magnetic disks, \$11,000 for one optical disk system and \$20,000 for magnetic tape systems.)

Compression Scheme	Data Rate (Mbits/sec)	Image Dimensions (pixels)
MPEG-I	1.5	320x240
MPEG-II	1 to 8	640x480
JPEG	8	640x480

Table 6.2: *Data rates and image dimensions for various compression schemes.*

6.3.2 Application-Level Workload Characterization

To characterize the video server workload at the application level, we must specify the size and number of video objects, the size of user requests and the locality of the request distribution, the number of users active in the system at peak periods, and the response time goals and throughput guarantees made to each user.

Much of the workload characterization depends on the compression scheme chosen. Table 6.2 shows the data rates for three compression schemes, MPEG-I, MPEG-II, and JPEG. The data rates correspond to the *throughput requirements* on a per-viewer basis. Throughput requirements vary from 1 to 8 megabits per second. Image sizes for each compression standard are also indicated. The 640x480 pixel image is a full-size image, while the 320x240 pixel image is a quarter-size image. To extrapolate these numbers to HDTV systems, assume an image size of 1024x1024 pixels; this corresponds to 3.4 times as many pixels as for the full-size image, so data rates for HDTV can be calculated by multiplying the corresponding values by a factor of 3.4.

Table 6.3 shows the *sizes* of typical video server objects using the three compression schemes. Objects include a 30-minute television show, a 50-minute class lecture and a 100-minute movie. The objects are very large, ranging from a few hundred megabytes for a television show using MPEG-I compression to 6 gigabytes for a JPEG movie. *Request sizes* to the storage server will approximately correspond to the object sizes, since we assume that in a video-on-demand server, viewers usually request that videos be shown sequentially and in their entirety. Some small percentage of the time, users will deviate from this request

Object Sizes (Megabytes)			
Object	MPEG-I 1.5 Mbits/sec	MPEG-II 3 Mbits/sec	JPEG 8 Mbits/sec
30-minute TV show	338 MBytes	675	1800
50-minute class lecture	563	1125	3000
100-minute movie	1125	2250	6000

Table 6.3: *Object sizes for typical video server objects for the three compression schemes.*

pattern by performing VCR-like operations such as pause and rewind. There is little data available on the frequency of these operations. For simplicity, we assume that 10% of viewers perform pause operations and 5% of viewers perform positioning operations like search and rewind. These assumptions should be revisited as traces of user behavior become available.

Some information on the *locality* of accesses to the video server can be inferred from video store rental patterns. This information is derived from magazines for the video rental industry and from discussions with video rental store owners. One store owner reports about 60% of rentals to movies in the "new-release" category, which includes movies released in the last four months. Table 6.4 shows a ranking of the 40 most popular video rentals for the week ending September 25, 1993, with statistics on rentals for each title for an average video store. The popularity rankings are from *Billboard Magazine* (September 25, 1993 issue), based on a national sample of retail store rental reports for the week ending September 25, 1993. Data in other columns are taken from *Video Store Magazine*, October 3-9, 1993, ranked by number of rentals per copy in average video store for the week ending September 26, 1993. Blank spaces in the table indicate entries in the Billboard chart that did not appear in the Video Store magazine chart. These may occur because the videos did not rank in the top 50 based on rentals per copy, or because the videos are older than the 120 day limit imposed by the Video Store charts. (The rental statistics do not strictly correspond to the Billboard popularity rankings.)

From Table 6.4, we see that approximately 25% of these most popular movies rent about 20 times per week per video store. The next 25% rent about ten times a week, and the

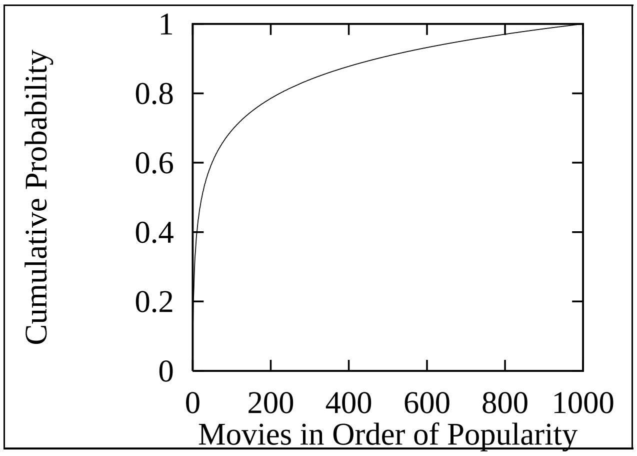


Figure 6.1: *Zipf's Law distribution with 1000 movies.*

remaining 50% about five times a week. This suggests that even within such a small number of movies, there is a high degree of locality. Of course, it is tricky to extrapolate from video store rental statistics, which are affected by the number of copies on hand, to video-on-demand server access patterns, where a much larger number of simultaneous accesses to popular movies is possible. The locality patterns are instructive, however, and video store owners have strong economic incentives to purchase a sufficient number of copies.

Since video rental statistics suggest a highly localized distribution of accesses, we consider some typical request pattern distributions with high locality. Among these are the 80/20 and 90/10 rules, in which 80 (90) percent of accesses go to the most popular 20 (10) percent of the data. For the video-on-demand workload, we choose instead to use the Zipf's Law distribution to characterize the locality of accesses. Figure 6.1 shows the cumulative probability distribution of the Zipf's Law distribution for a server containing 1000 movies. The Zipf's Law distribution stipulates that the probability of choosing the nth most popular of M movies is

$$C/n,$$

where

$$C = 1/(1 + 1/2 + 1/3 + ... + 1/M).$$

Popularity Ranking and Movie Title	Days in Release	Copies Per Store	Average Rentals Per Week Per Store	Average Rentals Per Copy
1. Falling Down	47	5.4	18.9	3.51
2. Scent of a Woman	61	6.0	17.4	2.88
3. Groundhog Day	33	6.6	24.4	3.70
4. The Bodyguard	75	7.5	12.0	1.60
5. Unforgiven	82	6.8	11.1	1.64
6. Sommersby	54	5.4	12.4	2.29
7. A Few Good Men	89	7.0	10.6	1.52
8. Benny and Joon	47	3.4	11.7	3.44
9. Homeward Bound				
10. Home Alone 2				
11. Point of No Return	26	5.3	25.5	4.77
12. The Crying Game	82	4.3	8.2	1.90
13. Mad Dog and Glory	40	2.6	10.3	3.94
14. Alive	26	6.4	22.7	3.55
15. Untamed Heart	54	2.7	7.4	2.71
16. Nowhere to Run	61	3.8	9.7	2.53
17. Sniper	54	3.5	9.3	2.66
18. The Vanishing	47	3.1	8.5	2.72
19. The Temp	33	2.3	9.9	4.20
20. The Bad Lieutenant	40	1.2	5.7	4.57
21. Bram Stoker's Dracula	96	4.7	5.6	1.18
22. Forever Young	110	4.6	5.1	1.11
23. Boiling Point	33	3.1	12.4	4.05
24. Army of Darkness	54	2.0	5.3	2.66
25. Amos and Andrew	68	2.4	5.2	2.20
26. Leap of Faith	89	3.0	5.2	1.73
27. Malcolm X	68	2.6	3.2	1.21
28. This Boy's Life	26	1.6	5.1	3.15
29. Body of Evidence	103	3.5	7.2	2.05
30. Damage	103	1.3	3.0	2.35
31. Jennifer 8	96	2.3	5.4	2.32
32. Eden 2				
33. Hear No Evil	47	1.4	4.4	3.08
34. Wild Palms				
35. Lorenzo's Oil	75	2.5	4.5	1.78
36. Broadway Bound				
37. Passion Fish	68	1.4	3.4	2.42
38. The Dist. Gentleman				
39. The Lover	103	0.6	2.4	3.65
40. A River Runs Thr. It				

Table 6.4: *Video rental statistics,* Billboard *9/25/93 and and* Video Store *10/3-9/93.*

For example, using this distribution, the 5th most popular movie is requested one fifth as often as the most popular movie; as seen in Figure 6.1, this is a highly-localized distribution. Zipf's Law is a plausible choice to characterize accesses to a video server in the absence of other empirical information because the distribution has been used successfully to model similar access patterns, such as the distribution of materials checked out of a traditional, "paper" library.

Matching the data in Table 6.4 to a Zipf's Law curve is difficult, since accesses to the most popular videos are limited by the number of copies on-hand. However, a Zipf's Law distribution for 2000 movies, which would be a typical number for a video store, and 950 total weekly rentals approximately matches the data in Table 6.4 except for accesses to the four most popular movies. In the average video store, accesses to those most popular movies are limited to about 25 total rentals by the number of copies on hand. However, the Zipf distribution estimates that, with an unlimited number of copies on hand, there will be 116 rentals of the most popular movie, 58 rentals of the second most popular movie, 39 of the third, and 29 of the fourth. Thus, video rental data for a single week indicates that a Zipf distribution with a truncated head makes a fairly good match to video rental patterns; more detailed studies of video store rental patterns would be useful.

The choice of the Zipf's Law distribution may affect the *number of video objects* stored in the database. Because the distribution is highly localized, with a large percentage of accesses going to a small percentage of the data, many companies building video servers plan to service only the most heavily-accessed portion of the distribution. Such a decision would allow the most popular movies to be stored on magnetic disk and deliver the maximum revenue for the capital investment in the storage system. Many video server systems will provide between 50 and 250 movies on magnetic disk to service this "head" of the request distribution while ignoring the remainder, or "tail", of the distribution. Other systems may incorporate magnetic tape and magnetic disk in a storage hierarchy to provide access to a

Video-on-demand system	Population size
Starlight Networks	20
Local Switching Office	2,000
Cable Television Subscriber Areas	4,000 to 100,000

Table 6.5: *Approximate sizes for various populations served by proposed and existing video servers.*

larger number of movies, thereby servicing the tail of the distribution.

Table 6.5 shows several population sizes for video server systems. The Starlight Networks server supports a small user population, about 20 simultaneous users in the minimum configuration. A telephone local switching office may serve approximately 2000 customers. The SGI/Time Warner prototype will support a population of 4000 customers. Cable television systems may be much larger, with up to 100,000 customers. We assume that at peak periods, approximately 20% of the population will be active in the video server system.

Finally, we assume that typical video-on-demand systems will have fairly strict response time goals, delivering video within 10 seconds of its being requested. For systems that incorporate a storage hierarchy, requests that fall through the magnetic disk systems to the tertiary storage system may have much longer response times, for example, several minutes for accesses to magnetic tape systems. In the next chapter, we examine the performance of different storage configurations for this workload.

6.4 Digital Libraries

Digital Libraries allow users to browse through large multimedia databases that include text from many sources, still images, video clips, and perhaps interactive access to applications like word processors and simulation packages. In this section, we survey existing digital library systems and propose a workload characterization.

Overall Melvyl Statistics	
System Sessions	132,250
System response time	
Find operations	2.04 seconds
Display operations	0.4 seconds
Total number of Find Operations	380,420
Total number of Display Operations	3,726,000
Maximum active users in 2 minute intervals	423
Find commands per hour at peak load	5251
Display commands per hour at peak load	81599

Table 6.6: *Usage statistics for Melvyl bibliographic database.*

6.4.1 Existing Systems

In universities, one type of digital library is commonplace: on-line bibliographic catalogs. Users search the bibliographic databases to identify interesting documents. Unfortunately, few university library systems currently offer the ability to retrieve the full text of chosen documents. This is primarily because of copyright concerns from publishers, who fear financial losses due to illegal proliferation of texts in electronic form.

At the University of California, the bibliographic database is called Melvyl [53]. Melvyl is accessed heavily by the user community, with approximately 130,000 user sessions per week performing approximately 400,000 search operations and 4 million display operations. Melvyl consists of several smaller databases, including databases for magazines and newspapers, computer articles, medical and life sciences journals, and psychology journals. Table 6.6 shows weekly statistics for the Melvyl databases [42]. The peak number of users during one two-minute interval was 423. The peak number of FIND commands in an hour was 5251. Table 6.7 shows response time statistics for Melvyl. From that table, we see that 90% of all to Melvyl requests finish in 2.2 seconds on average, and that 95% finish in 5.5 seconds. For FIND commands, which do database searches, the 90% response time is 5.4 seconds, and the 95% response time is 9.8 seconds.

Unfortunately, statistics on bibliographic databases don't reveal much about how

Response time (seconds)	90%	95%
Average	2.2	5.5
Prime Time (10am-6pm)	2.4	5.9
Peak load (2-4pm)	2.4	6.1
Find commands	5.4	9.8
Display commands	0.9	1.2

Table 6.7: *Response time statistics for Melvyl bibliographic database. Column labeled 90% indicates time by which 90% of all access have been completed for each category. Column labeled 95% indicates time by which 95% of all accesses are complete.*

full-text databases would be used. Users would still use bibliographic indexes to search for relevant articles, but would also be able to retrieve all or part of documents. The COMP database component of the Melvyl database is an example of a small full-text, electronic library prototype being implemented in the university. COMP contains 200 journals and magazines related to computers, and for several months, full text has been available for retrieval on some of these articles. Librarians report that about 20% of DISPLAY operations request full text display [27]. However, this statistic may change dramatically as more texts become available.

The World-Wide Web can be considered a distributed digital library [12]. The World-Wide Web is a collection of repositories of information located all over the world. Contributors to and users of the web use a common data model and hypertext-based interface, allowing users on the internet network to browse through data stored at thousands of sites. Usage patterns on World-Wide Web sites should provide valuable information on how people use digital libraries.

One interesting full-text application is the Electronic Reserve Book Room at San Diego State University [14]. The university was faced with dwindling library budgets coupled with increased demand by professors and students for use of books and articles held "in reserve", that is, checked out only for short periods to allow a large number of students access to the materials. They responded by putting reserve materials on-line. Bit-mapped images of the materials are stored in a 10-gigabyte WORM optical disk jukebox. Students request documents at computers on the library's local network and are charged for printing costs. Royalty payments are automatically calculated and sent to publishers. To avoid unauthorized proliferation of texts, the network on which the texts reside is isolated from the rest of the campus computer network. Paper copies of reserve materials are available for those students who prefer not to pay for reserve materials. In such a system, locality will be determined by the number of people taking particular classes, and most students will request that articles are printed in their entirety.

Several commercial databases offer full text retrieval of documents. One example is Mead Data Central, which offers legal, medical and newspaper articles from over 2000 sources to clients [117]. Although traces of user behavior are not available, some conclusions about the workload can be drawn from this system. The entire database consumes two terabytes of magnetic disk storage. Processing is distributed over several IBM-370 type mainframes, with each mainframe controlling accesses to a set of disks and peak I/O rates of about 1000 I/Os per second per mainframe. Currently, the system services about 250,000 user requests per day, where a typical search request generates 500-600 disk I/O operations. At peak periods, the number of active users is about 2000.

Mead Data Central reports that the level of temporal locality in system references is very low, with average recurrence rates for accesses to index files about four or five times an hour, and for data files in the range of hours or days. Given the large volume of data and the very high rate of requests, the probability that data in memory will be re-used before it is overwritten is very low. Therefore, no caching is implemented.

Mead doesn't make response time guarantees to customers, but their systems are implemented with a response time goal of ten seconds. Average search time is eight seconds. They characterize users as belonging to two classes, knowledge workers and researchers. The two types of users access the system in dramatically different ways. Knowledge workers search for specific information needed to do their jobs. An example is a newspaper reporter who needs to find the most recent articles dealing with a particular event or person. Knowledge workers tend to access the electronic library only occasionally and for short periods of time, to retrieve relatively small search results. Researchers, on the other hand, are paid to find information. An example would be a librarian in a biotechnology company who needs to do exhaustive searches of patent applications, medical, chemistry and biology journals, and newspapers to track the research and try to deduce the business plans of competitors. Researchers tend to use the electronic library systems for long periods and retrieve much larger search results. Modeling the pattern of accesses to electronic libraries will require an understanding of both types of users.

6.4.2 Application-Level Workload Characterization

Scaling the workload

The digital library workload that we derive should be scalable over a wide range of user populations. We argue that three parameters, the number of objects, locality and number of users, should scale to justify a claim of increased performance. Other workload parameters are unchanged.

The most basic measure of scaled or increased performance is that more accesses per hour are delivered by the information server given a particular response time guarantee. We believe that the number of users and the number of objects in the digital library must scale to justify a claim of a greater number of accesses per hour. Our rationale is analogous to the arguments given for the TPC-B database benchmark [2] relating to Automatic Teller Machine (ATM) transactions. The TPC-B benchmark specifies that manufacturers cannot claim that they support a higher rate of transactions per second without scaling the number of users and the total amount of account information. These scaling requirements are logical because more transactions per second in a bank ATM system don't imply that customers are suddenly performing more transactions; rather, it suggests that there are more customers, and hence more account files and higher storage requirements.

Likewise, we don't believe that an increase in accesses per hour delivered by an information server will correspond to increased activity by individual users, but to an increase in the number of users. Although a larger number of users won't strictly require more capacity as it would in an ATM system, it is logical to assume that a larger user population will make it economically feasible to store more data and will demand a greater diversity of material. As a result, we stipulate that the number of objects in the database must also increase as performance scales. Based on this assumption about larger databases, locality patterns may also change as digital libraries scale.

We did not require similar scaling of the number of objects in a video server to justify claims of increased performance. Although the same logic holds (more users make it affordable to store more material), given the highly-localized usage patterns, scaling the number of objects is not necessary as the number of viewers increases; many companies will choose to serve only the most localized portion of the Zipf's Law distribution. By contrast, digital libraries have lower locality in their request patterns, and users are more likely to demand a wider variety of material.

Several parameters are unaffected as performance scales. The size of an object (video clip, book, article) depends on the application and the compression scheme used, and is unaffected by the service rate of the information server. Request sizes are also unchanged; they are determined by the optimal access patterns for the hardware used to implement the server and by the application. Response time goals or requirements likewise remain constant as the number of users supported by the server increases, since they depend only on the nature of the application. Finally, the throughput required per user depends on the application and on such implementation choices as the compression scheme and image format, but does not change as performance scales.

Location	Volumes in Library 30 June 1992	Serials Received 30 June 1992
Berkeley	7,854,630	88,374
Davis	2,519,048	51,122
Irvine	1,500,867	17,642
Los Angeles	6,247,320	94,156
Riverside	1,561,662	12,951
San Diego	2,188,722	24,388
San Francisco	700,389	4778
Santa Barbara	2,074,813	24,400
Santa Cruz	1,037,099	10,183
Total	25,739,962	327,994

Table 6.8: *Statistics on volumes in each library and number of serial publications currently received by each of the nine campuses of the University of California.*

The Workload

To characterize the digital library workload, we attempt to specify the size and number of library objects, the size and locality distributions of user requests, the peak number of active users, and the response time goals and throughput guarantees made to each user.

Tables 6.8 and 6.9 use data from libraries of the University of California campuses and branch libraries of U. C. Berkeley to suggest a scaling relationship between the *number of objects* in a digital library and the user populations served by those libraries. In particular, Table 6.9 shows a ratio of volumes to users that ranges from 30 to 183, and a ratio of serials to users ranging from 0.6 to 2.3.

Table 6.10 estimates object sizes for news and journal articles and books for pages stored as ASCII data and as bit map images. We assume 2 pages for a newspaper article, 20 pages for a journal article and 200 pages for a book. We assume that ASCII pages consume 4 kilobytes and bitmap page images consume 200 kilobytes.

It is likely that users will retrieve bibliographic data, portions of articles, portions of books including chapters, and entire articles or books. One of the open research questions

Library Location	Number of Volumes 30 June 1992	Serials Received 30 June 1992	Approximate Number Faculty and Students	Ratio Volumes to Users	Ratio Serials to Users
Main Building	3,943,916	33,333	32,000	123	1.04
Biosciences	365,660	6871	3000	121	2.29
Music	146,239	1372	800	183	1.72
Chemistry	63,241	769	1200	52.7	0.64
Optometry	9830	191	320	30.7	0.60

Table 6.9: *Statistics for a few branch libraries of the University of California at Berkeley. Includes number of volumes, number of serials currently received, and approximate number of faculty and students for individual departments. For the main library, we include the number of faculty and students for the entire campus.*

Object Type	Number Pages	ASCII Storage (kilobytes)	Image Storage (kilobytes)
News article	2	8	400
Journal article	20	80	4000
Book	200	800	40000

Table 6.10: *Typical sizes for ASCII and bit-map page images.*

in this area is the distribution of request sizes.

Table 6.11 shows usage characteristics for two of the databases of the Melvyl system, IAC and INSPEC. The IAC database includes bibliographic data for three components: 1500 magazines (MAGS), five major U.S. newspapers (NEWS), and computer-related magazine articles (COMP). For the last several months, IAC has also included some full text for articles in the COMP database. INSPEC contains a bibliographic index for 4000 physics, electronics and computing journals.

There should be a high degree of *locality* in accessing commonly-used index and bibliographic data files. Table 6.12 gives statistics on I/O operations and file sizes for the 14 files that make up the IAC database. The table shows evidence that there is high locality of accesses in this database. In the IAC database of the Melvyl system, 50% of all accesses are to index files that occupy around 35% of data space. About 30% of all accesses are to the bibliographic data file, which consumes about 25% of the database. Most of the remaining

Combined IAC and INSPEC databases	
Index files	8214 Megabytes
Data files	18254 Megabytes
Temporary storage	368 Megabytes
Weekly statistics: IAC	
Number of Sessions	14,000
Response time (FIND)	1.5 seconds
Response time (DISPLAY)	0.2 seconds
Number FIND operations	47,000
Number DISPLAY operations	500,000
Weekly statistics: INSPEC	
Number of Sessions	2700
Response time (FIND)	2.3 seconds
Response time (DISPLAY)	0.2 seconds
Number FIND operations	7500
Number DISPLAY operations	75,000

Table 6.11: *Statistics for IAC and INSPEC databases: weekly averages over 10 weeks: July 12 to September 3, 1993.*

accesses are to database keyword files. Table 6.12 shows the database files numbered 60 through 73. File 60 holds some pointers, but is not used in search processing. Files 61-63 are index files for journal (MAGS), newspaper (NEWS), and computer (COMP) articles, respectively. Files 64-66 are keyword pool files for the three types of articles, used to enable faster keyword searches. File 67 contains the bibliographic data, with pairs of authors/titles for various articles. Files 68-72 are keyterm files, used to perform exact keyword searches for authors, subjects, and journal titles. Finally, File 73 contains the full text when available for the articles in the COMP database. Access patterns show high locality, with 36.9% of all accesses going to magazine index (file 61), which occupies 10.9% of total blocks, and 31.7% of accesses to the bibliographic data file (file 67), which occupies 22.8% of total blocks.

The locality shown in Table 6.12 suggests that in a digital library of this size, caching the index data should improve performance considerably. (A movies-on-demand video server should receive a similar benefit from caching index data, since a relatively small number of movies are added each year.)

File Type	File Number	Number of Index Reads	Number of Data Reads	Percent Total Reads	Number Index Blocks	Number Data Blocks	Percent Total Blocks
Pointers	60	0	0	0.0%	3	150	0.0%
Index	61	134540	213384	36.9%	170354	87407	10.9%
Index	62	35076	52284	9.0%	327096	161475	20.7%
Index	63	10059	15734	2.7%	88675	40505	5.5%
Keyword Pool	64	46399	46399	8.0%	100811	74501	7.4%
Keyword Pool	65	14529	14529	2.5%	128971	97242	9.6%
Keyword Pool	66	4391	4391	0.8%	96321	66375	6.9%
Biblio. Data	67	78107	183308	31.7%	7775	529982	22.8%
Keyterms	68	9169	9169	1.6%	15901	6530	1.0%
Keyterms	69	24689	24689	4.3%	39003	16390	2.4%
Keyterms	70	9922	9922	1.7%	60415	24462	3.6%
Keyterms	71	1669	1669	0.3%	17028	7110	1.0%
Keyterms	72	1303	1303	0.2%	195	47	0.0%
Full Text	73	370	1171	0.2%	819	188981	8.0%

Table 6.12: *Shows statistics for the files of the IAC database, which actually contains the MAGS (magazine), NEWS (newspapers), and COMP (computer-related articles) databases. For each of the files that make up the IAC database, shows number of reads to the index and data portions of the file and percent of total read accesses, and shows number of blocks making up the index and data portions of the file and percent of total capacity. Database block sizes are 2544 bytes for index files and 5064 bytes for data files.*

Object size	2.2 gigabytes for average movie
Number of objects	50 to 10,000
Request Size Distribution	Request entire movie
Locality of accesses	Zipf distribution
Peak Number of Active Users	20% of user population
Response Time Requirement	10 seconds to few minutes
Throughput per Active User	3 megabits per second

Table 6.13: *Shows workload characteristics for a typical movies-on-demand video server application.*

In the Mead Data Central database [117], the frequency of re-use of data is much lower, with recurrence rates for accesses to index files about four or five times an hour and for data files in the range of hours or days. Part of the reason for this rare re-use of data is the vast amount of material in the database and the diversity of accesses to the database. In such a database, caching index material will be less effective.

For both the Melvyl and Mead Data Central digital libraries, response time goals are less than ten seconds.

Finally, the *throughput required per user* can be specified by human reading rates, which have been estimated at 0.030 to 0.375 kilobytes/second for reading text, and 125 kilobytes/second for image recognition [81].

6.5 Summary

We have presented candidate workloads for two emerging applications: video-on-demand servers and digital libraries. The workload model used in this chapter defines object sizes, number of objects, the request size distribution, locality of accesses, the peak number of active users, the response time requirements, and the throughput required per active user.

Accurate workload information for these new applications is difficult to obtain because of strong competition between companies developing such systems. In the absence

Object size	4 megabytes (average journal article)
Number of objects	approximately 100 books, 2 serials per user
Request Size Distribution	unknown
Locality of accesses	Zipf distribution
Peak Number of Active Users	hundreds or thousands
Response Time Requirement	10 seconds
Throughput per Active User	up to 125 kilobytes/second

Table 6.14: *Shows workload characteristics for a typical digital library application.*

of traces or even predictions of these workloads, we have used the available evidence to make a first attempt at characterizing the workloads. Tables 6.13 and 6.14 summarize the characteristics of the two workloads. In the next chapter, we evaluate storage system options for the first of these workloads, movies-on-demand video service.

Chapter 7

Storage Systems for Video Service

7.1 Introduction

In this chapter, we discuss the design of storage systems to support video service. We focus exclusively on the movies-on-demand application. In the last chapter, we described the expected workload for this application, summarized in Table 7.1. We assume an MPEG-II compression standard with a data rate of approximately 3 megabits/second. The objects (movies) consume over 2 gigabytes of storage, on average, and will most likely be requested in their entirety. For each stream, strict performance constraints must be maintained to guarantee adequate presentation quality; the compression standard demands delivery of up to 30 frames per second. Finally, we assume that requests to the video server are highly localized, with most of the requests going to a small number of the most popular movies. We use the Zipf's Law distribution to model this request pattern.

There are two practical possibilities for building a storage system for video service, illustrated in Figure 7.1. The first is to build the server entirely out of magnetic disks. Disk-based systems fall into two categories: *disk arrays* and *disk farms*. Disk arrays (RAIDs [80]) include specialized software and/or hardware controllers that stripe data among disks and maintain parity information. Disk farms are simple collections of disks generally attached

Throughput per stream	3 megabits/second
Frame rate	30 per second
Typical movie length	100 minutes
Typical movie capacity	2.2 gigabytes
Locality of accesses	Zipf's Law Distribution

Table 7.1: *Summary of workload characteristics for video service when the compression scheme is MPEG-II with a target data rate of 3 megabits/second per stream.*

to the standard I/O interfaces of hosts, usually without maintaining parity information and often without data striping. In this chapter, we focus only on disk farms.

The second possibility for a video server storage system is a storage hierarchy. A two-level storage hierarchy combines a small amount of faster, more expensive storage like magnetic disk with a larger amount of less expensive storage such as magnetic tape or optical disk. A storage hierarchy can take advantage of locality of accesses in the workload by storing the most frequently-accessed videos on the faster, more expensive storage and the less popular objects on slower, less expensive storage. Objects migrate between levels of the hierarchy based on growing or waning popularity. The ideal of a storage hierarchy is to give all accesses approximately the performance of the fastest level of the hierarchy at approximately the cost of the least expensive level.

Figure 7.2 shows approximate cost and access time characteristics of various storage devices. The fastest access is from disk farms and disk arrays, where data can be retrieved in milliseconds. Disk farm prices are based on the cost of commodity disks (currently about 50 cents per megabyte and dropping approximately 60% per year [79]), while disk array prices, which are based on disk controller costs rather than commodity disk prices, average about $2 per megabyte of storage for hardware RAID controllers and $1 per megabyte for software RAID implementations [116]. Next in speed of access are optical disk jukeboxes designed for computer storage that can access video data in approximately 10 seconds and store data for about 50 cents per megabyte. Finally, tape libraries take on average several minutes to access data at a cost of approximately 10 cents per megabyte.

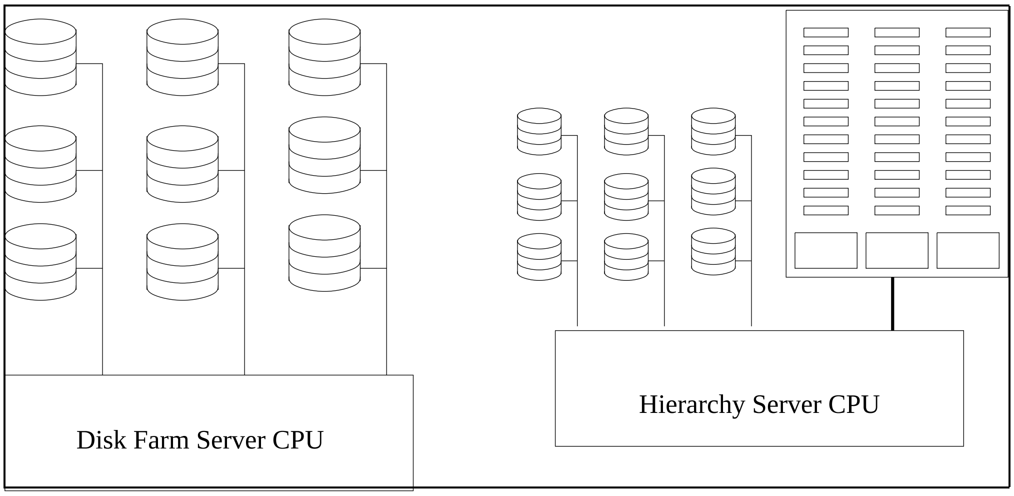


Figure 7.1: *Shows the possibilities for building a video server: entirely disk-based, or a storage hierarchy of disk and tertiary storage.*

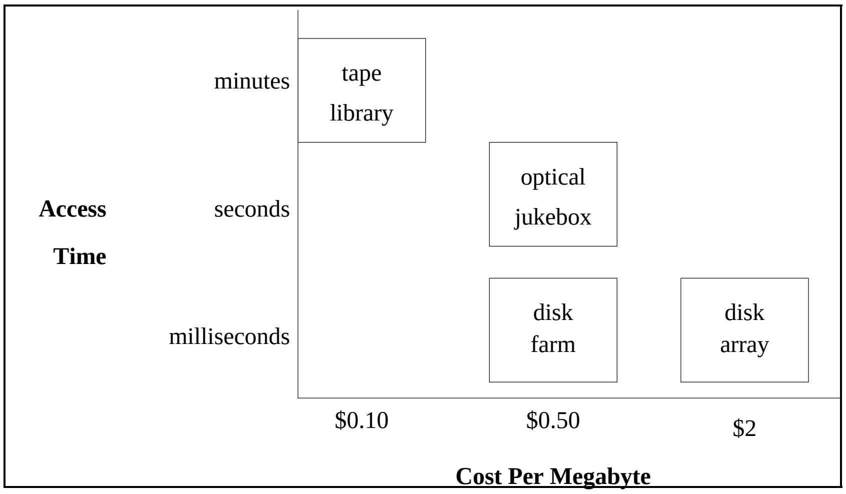


Figure 7.2: *Possible components of a storage hierarchy, with approximate cost and access times for 1994.*

In this chapter, we first explore disk storage systems for video service and then examine the use of storage hierarchies. We find that disk systems perform best on the most important cost/performance metric for video servers: the cost per video stream. In Section 7.2, we compare striped and non-striped disk farm performance and show that a striped system allows us to support more viewers with a lower cost-per-stream. In Section 7.3, we show that, although storage hierarchies offer inexpensive storage capacity, their cost per video stream is high compared to that of disk systems. As a result, we predict that the current generation of tertiary storage systems will not be a useful part of most video servers. Finally, we evaluate improvements in tape libraries and optical jukeboxes that would make them perform better in a video server storage hierarchy.

7.2 Disk Systems for Video Service

In this section, we explore the use of disk farms to supply data for video service. We will examine two possibilities for storing video data in disk farms. The first is the straightforward mechanism of putting a single movie on each disk, and the second is a scheme that stripes movies across a collection of disks. Data striping was described in Chapter 5.

7.2.1 One Movie Per Disk

Figure 7.3 shows video server disk farm in which one movie is stored on each disk. Given the transfer rates and compression scheme that we are assuming, a movie consumes about 2.25 gigabytes. The current generation of disks has approximately the same capacity, making the one-movie-per-disk scheme simple to implement. As disk capacities increase dramatically over the next few years, it is likely that this scheme will place several movies on the same disk.

One problem that may occur with a one-movie-per-disk scheme arises if some

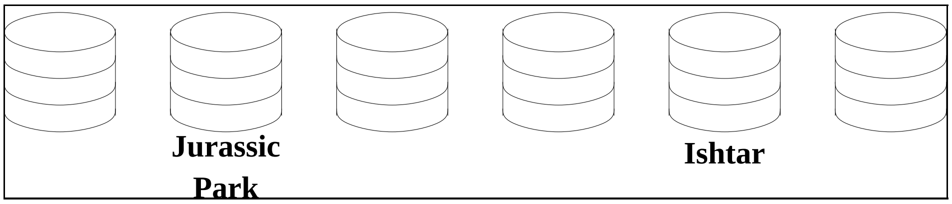


Figure 7.3: *Shows disk-based video server configuration with one movie per disk.*

movies are more popular than others. Such popular movies will create hot spots or bottlenecks in the disk system. For example, in Figure 7.3, we might expect that *Jurassic Park* would be more popular than *Ishtar*. A single disk holding Jurassic Park can support some small number of users. For example, typical disk bandwidth in 1994 is approximately 2 megabytes per second. If each video stream consumes 3 megabits or 0.375 megabytes per second, then a single disk can support about five independent viewers watching the movie concurrently. If more than five viewers at a time want to see Jurassic Park, then the movie must be replicated onto other disks. At the same time, if Ishtar is less popular, there may at times be fewer than five viewers watching Ishtar. During such periods, some of the capacity and the bandwidth of the disk holding Ishtar is essentially wasted.

There are several advantages to the one-movie-per-disk scheme. The first is that VCR-like operations, such as fast-forward and rewind, should be easy to support in this environment since the disk is naturally a random-access medium and it is easy to schedule seeks to the appropriate disk sectors. The second advantage of the one-movie-per-disk scheme is the simplicity of handling failed disks. When a single disk fails, that copy of a movie becomes unavailable. While this may disrupt a small number of streams (at most five, in the example above), it will be relatively simple to re-schedule those streams on other disks holding the same movie or to import a new copy of the movie onto a spare disk. Below, we show that both VCR-like operations and disk failures are somewhat more difficult problems for a striped disk scheme.

In Section 7.2.3, we present simulation results for several variations of the one-

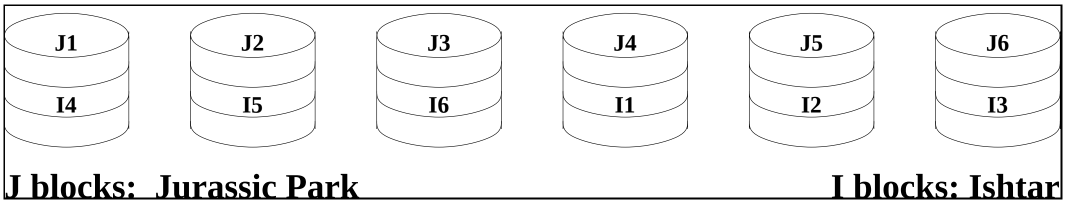


Figure 7.4: *Shows disk-based video server configuration with movies striped across several disks.*

movie-per-disk scheme. We explore different replication strategies for movies and show that the one-movie-per-disk scheme is only cost-effective when movies are replicated to reflect the locality of the access pattern.

7.2.2 Striping Movies Among Disks

Figure 7.4 shows a scheme for laying out movies in a disk farm in which portions of movies are striped across disks. For each movie, subsequent small increments of video time are placed on different disks. During any time interval, each disk can service some small number of requests to different movies (say five, given the example discussed in the last section). In the next time interval, each active video stream moves to a different disk for service.

The advantage of this scheme over the one-movie-per-disk scheme is that in the striped configuration, unwatched movies consume disk capacity but do not waste disk bandwidth. Instead, every disk can devote all its transfer capabilities to showing whichever movies are most in demand. Because movies are spread over a collection of disks, the system has better load balancing. In this chapter, we assume that movie intervals are distributed across the disk farm in a round-robin fashion.

The main challenge for striped layout of movies in a video-on-demand server is somewhat more complex scheduling that was required in a one-movie-per-disk configuration. Scheduling is more complicated for normal requests, for VCR-like operations, and during disk failures. During normal operation, when a viewer wants to see a movie, playback of the movie must wait until there is an open scheduling interval on the disk holding the first segment of the movie. If the system is highly-loaded, this wait could be substantial. To minimize the wait times, several copies of movies may be maintained, as discussed in Section 7.2.3. Once movie playback has begun, the system must guarantee that there will be no further scheduling difficulties that might result in missing a deadline for playing back a video frame. This can be achieved straightforwardly if all streams move in a round-robin fashion among the disks.

If the viewer disrupts normal playback to do a VCR-like operation such as pause, fast-forward or rewind, these operations must also be scheduled during available intervals of disk operation. For example, if the viewer pauses, no further disk data transfer is required until the viewer resumes watching the movie; at that time, however, the stream must be resumed in a timely manner with the same playback quality guarantees as before. If the viewer is performing a fast-forward operation where either the frames are played back at a higher rate or some subset of the frames is played back, the appropriate frames must be fetched from a collection of disks during free scheduling intervals on those disks. In either case, if the disk system is heavily-loaded, scheduling these operations may be difficult.

Finally, if a disk fails in the striped scheme, the effect is potentially more damaging than in the one-movie-per-disk scenario. Now a failed disk will contain portions of many movies, threatening the playback of perhaps a large number of video streams. If there are other copies of the same movies in the array, the streams can be serviced by the disks holding duplicates of the missing movie blocks. The contents of the failed disk may be reconstructed, either by restoring them from tape or by using redundancy information, if it is maintained, to rebuild the missing data. The latter rebuild operation involves read operations to retrieve existing data, redundancy calculations to deduce the missing data, and write operations to write the restored data; these disk read and write operations would introduce further scheduling complexity.

In the following section, we demonstrate that striped disk farms are successful in supporting many video streams.

7.2.3 Performance of Disk-Based Video Servers

In this section, we present simulation results that compare the performance of the one-movie-per-disk and striped disk schemes for video service. We simulate only typical movie requests to the video server; the performance impact of disk failures and of VCR-type operations, such as pause and rewind, are subjects for future work.

One-Movie-Per-Disk Performance

The first set of results uses a simple, event-driven, one-movie-per-disk disk farm simulator that has many of the same features as the tertiary storage array simulator described in Chapter 3. For a set of disk parameters including bandwidth and average seek and rotation times, we determine the number of streams each disk can support. Then for a given pattern of replication of movies (for example, two copies of each movie), we determine the performance of the disk farm for a specified load. We perform closed simulations, keeping the load or number of outstanding video requests constant. Results for the simulations include the average wait time for a given simulation, the peak load actually achieved by the disk farm and the average cost per stream based on a disk cost assumption of 50 cents per megabyte.

Figures 7.5, 7.6, and 7.7 show the performance of a video server disk farm that places one movie on each disk. The graphs each show three lines representing different disk configurations: a single copy of each movie, two copies of each movie, and a configuration that bases the number of copies of a movie on the Zipf's Law distribution, with each movie residing on at least one disk. The data points in each graph show the number of active streams supported when the total number of movies in the system is 10, 100 and 1000;

using our assumption of 50 cents per megabyte for the cost of a disk, the total disk system cost including replicates is shown on the horizontal axis of the graph. The three graphs differ in the response time limit imposed on the simulation. In each case, the y axis shows the number of active (simultaneous) streams that can be supported while maintaining the constraint that 95% of all requests must finish within a specified response time limit. After a stream begins play, it continues to be serviced by the disks in a round-robin fashion, so all its subsequent deadlines for playing frames are met. Figure 7.5 shows the shortest (and most commercially viable) response time limit of the three, that of one minute. For comparison purposes, the other graphs show response time limits of 5 minutes and 100 minutes, the latter being the length of the movies simulated. These longer response time limits are less likely to be commercially successful, since they belie the term "video-on-demand." Customer are unlikely to accept a video service that takes longer to deliver a movie than it takes them to drive to a conventional video store and rent a movie.

In each simulation, requests to the video server are highly localized according to the Zipf's Law distribution. The graphs confirm the intuitive conclusion that replicating movies to satisfy this highly-localized access pattern provides the least expensive cost per video stream.

In Figure 7.5, the lowest line in the graph shows performance when the video server holds only a single copy of each movie. The concurrencies achieved are quite low, approximately 10, 20 and 30 streams active for 10, 100 and 1000 total movies, respectively. With only a single copy of the more popular movies, and a maximum of 5 streams supported per disk, requests to the most popular movies suffer long delays and limit the overall number of streams that can be serviced within the one-minute response time constraint.

The middle line in Figure 7.5 shows that the number of active streams approximately doubles as two copies of each movie are supported. This represents a doubling of the bandwidth available for servicing the most popular movies, which again limit overall

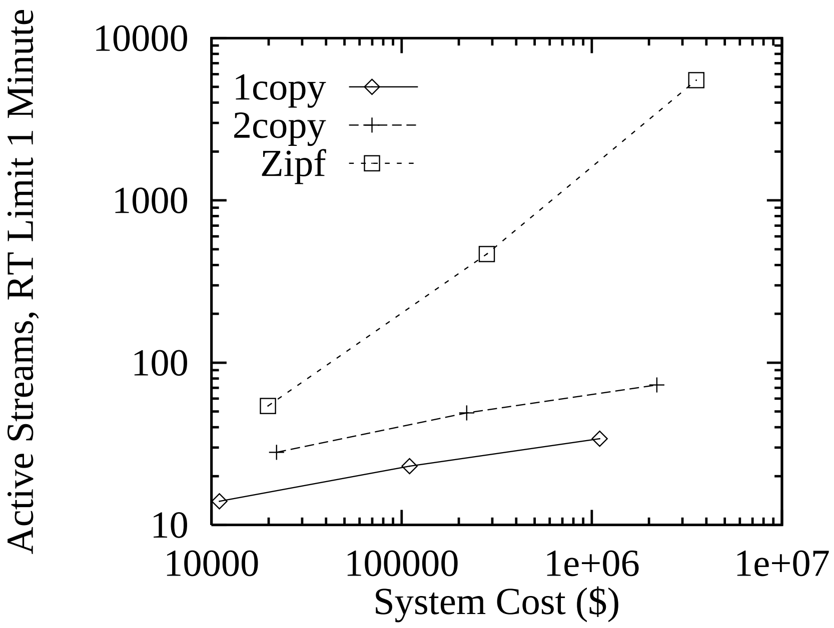


Figure 7.5: *One-movie-per-disk Performance, Response time limit 60 seconds: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must complete within 60 seconds. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.*

performance.

Finally, the highest line in Figure 7.5 shows the performance of the disk farm configured so that movies are replicated according to their popularity in the request pattern. The graph shows an improvement of one to two orders of magnitude in the number of streams that can be supported by the disk farm. Table 7.2 summarizes the results for the three different configurations. For cost per stream, the most important cost/performance metric for video servers, Table 7.2 shows that a replication pattern based on the Zipf's Law distribution and reflecting the request pattern to the video server is far superior to the 1-copy or 2-copy replication schemes. In a one-movie-per-disk scheme, replicating movies to reflect user access patterns, and most likely also dynamically copying movies to reflect changing viewing patterns, will be imperative for providing inexpensive video streams.

Figures 7.6 and 7.7 show video server performance for response time limits of 5 minutes and 100 minutes, respectively. The performance for a response time of 5 minutes

Layout Scheme	Total Movies	Total Disks	Active Streams	Cost Per Stream	Total Cost
1 copy	10	10	14	$786	$11,000
2 copy	10	20	28	$786	$22,000
Zipf	10	18	54	$367	$19,800
1 copy	100	100	23	$4,783	$110,000
2 copy	100	200	49	$4,490	$220,000
Zipf	100	255	467	$601	$280,500
1 copy	1000	1000	34	$32,353	$1,100,000
2 copy	1000	2000	73	$30,137	$2,200,000
Zipf	1000	3221	5513	$643	$3,543,100

Table 7.2: *For a response time limit of 60 seconds, compares the 1-copy, 2-copy and Zipf layout schemes for 10, 100 and 1000 movies. The Zipf layout is able to sustain many more concurrent streams, resulting in a much lower average cost per stream.*

is virtually identical to the 1-minute response time. This is not surprising, since streams take 100 minutes to be serviced, and free slots for playing new movies do not become available until old streams complete. For the 100-minute response time limit, the concurrency supported goes up dramatically, for example from about 5000 streams to 10,000 streams for a video server containing 1000 movies replicated according to the Zipf's Law distribution. Again, this is not surprising since the long response time limit allows many streams to complete and new streams to begin. Despite the higher concurrency level, however, the 100-minute response time limit is probably commercially unacceptable, since it fails to meet most definitions of "video-on-demand."

Striped Disk Farm Performance

In this section, we show that a striped disk farm supports a much larger number of concurrent streams than the one-movie-per-disk video server. For simplicity, all the simulation results that follow assume that each movie is striped among all the disks in the disk farm. In practice, disk farms may be partitioned into smaller groups of disks, with a subset of the movies striped among the disks of each partition.

There would be two advantages to this partitioning. First, our striped simulation

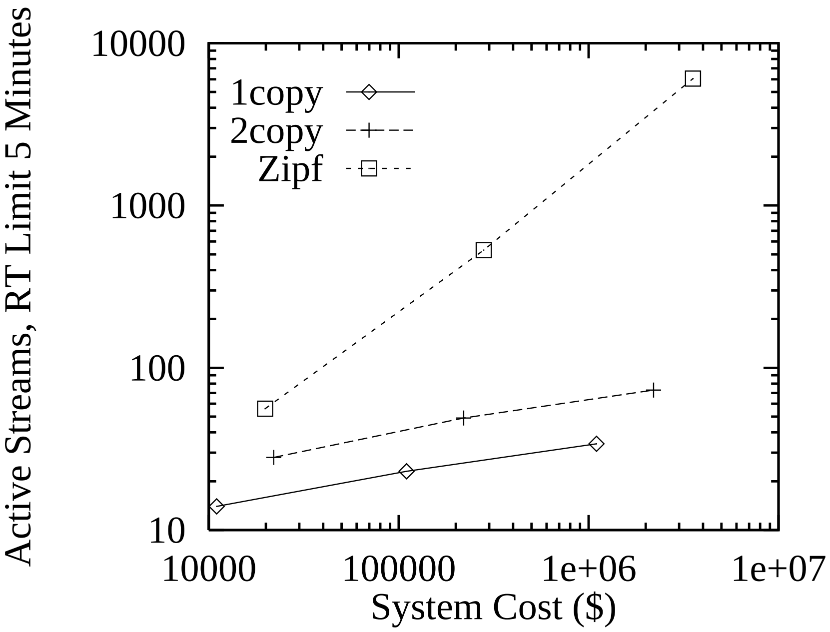


Figure 7.6: *One-movie-per-disk Performance, Response time limit 5 minutes: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must complete within 5 minutes. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.*

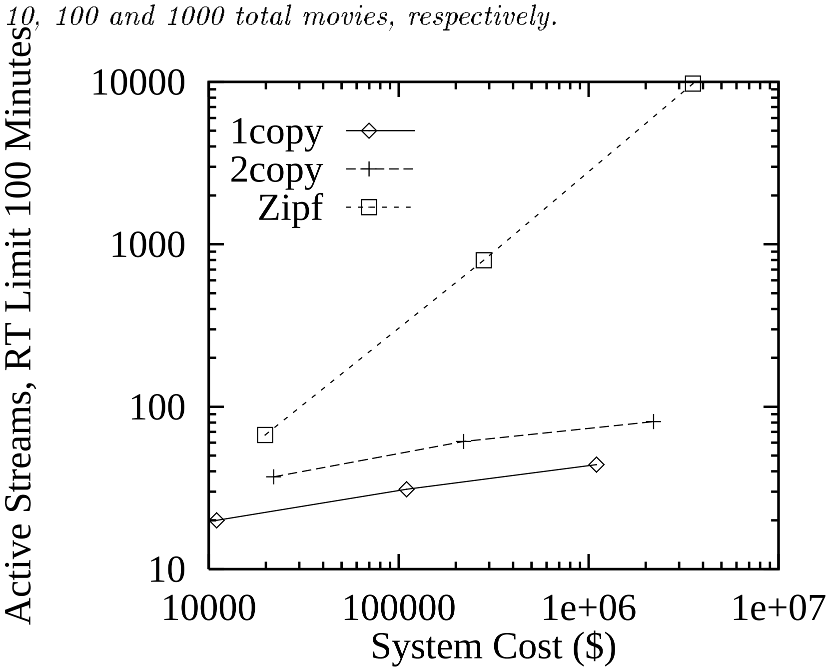


Figure 7.7: *One-movie-per-disk Performance, Response time limit 100 minutes: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must complete within 100 minutes, the length assumed for movies in this simulation. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.*

results will show that for a small disk array (10 or 100 disks) and a response time constraint of 60 seconds, a disk farm can be fully utilized with only a single copy of each movie on the disk farm. However, with a larger disk array (we simulate 1000 movies), if the system is highly utilized, there may be a long wait before a request for a popular movie can be serviced; minimizing this wait time may require replication of movies within the disk farm. Partitioning of the disk system may reduce the need for replication.

The second advantage to partitioning the disk farm is that such a system should be less vulnerable to disk failures. If all movies are striped across each disk, then any disk failure threatens all active streams, since a portion of each movie becomes inaccessible. Limiting movies to a subset of the disks limits the number of streams vulnerable to a disk failure. The optimal partitioning of a disk farm is left for a future study.

Since controlling a striped disk video server is largely a scheduling problem, the simulation results that follow use a short piece of C code that maintains a scoreboard of active video streams. Each disk can play several movie intervals during an interval of real time; these movie intervals are called "slots" and are scheduled at each real time interval for each disk. Before play begins for a new movie request, the simulator checks the disk (or one of a collection of disks, if the movie is replicated) on which the first interval of the movie is stored. If there is a free slot available on that disk during the next time interval, the new stream is scheduled; otherwise, the request is placed on a waiting list for the disk. When the disk has a free interval for playing a new stream, it takes a request off its queue. Once a movie stream has begun, it traverses the disks in a round-robin fashion, being serviced by each disk in turn until the full length of play (100 minutes) has elapsed. We assume all movies are striped on half-second intervals; this is the minimum length of time over which MPEG-II compression can be performed.

Figure 7.8 shows the performance of a striped disk farm containing 10, 100 and 1000 movies. The three lines in the graph show the disk farm with a single copy of each

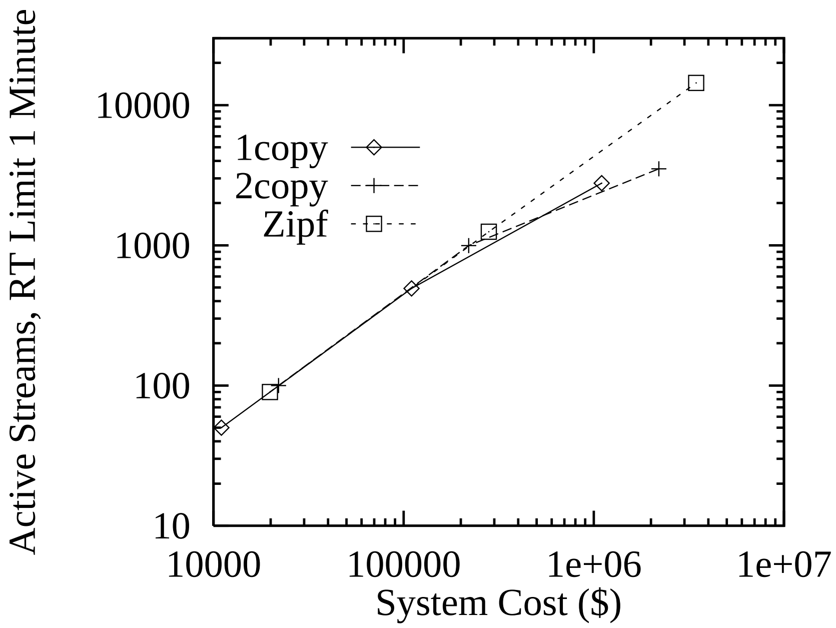


Figure 7.8: *Striped disk performance, Response time limit 60 seconds: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must complete within 1 minute. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.*

movie, two copies, and movies replicated according to the Zipf's Law distribution, respectively. The response time limit for the simulation is 60 seconds.

For disk farms containing 10 and 100 movies, the number of active streams fall on a straight line regardless of the replication pattern of the disks; in each case, the disk farm is approximately fully utilized, sustaining about 5 streams for every disk. For 1000 movies, the disk farm is no longer fully utilized when there is only a single copy of each movie. Rather than 5000 streams, the disk farm supports 2774 streams. The number of concurrent streams is limited because movies are striped in half-second intervals; with 1000 disks, a highly-loaded system might wait up to 500 seconds to find a free slot to play a movie, greatly exceeding the response time limit of 60 seconds. In the 100-movie disk farm, the maximum wait for an available free slot was 50 seconds, which is within the 60-second response time limit. Therefore, in larger video servers, we may need to replicate movies and scatter their starting positions throughout the disk farm to reduce the average wait

Layout Scheme	Total Movies	Total Disks	Active Streams	Cost Per Stream ($)
1 copy	10	10	50	220
2 copy	10	20	100	220
Zipf	10	18	90	220
1 copy	100	100	492	224
2 copy	100	200	997	221
Zipf	100	255	1248	225
1 copy	1000	1000	2774	397
2 copy	1000	2000	3515	626
Zipf	1000	3221	14415	240

Table 7.3: *For a response time limit of 60 seconds, compares the performance of a striped video server with 1-copy, 2-copy and Zipf layout schemes for 10, 100 and 1000 movies.*

time and achieve high disk utilization. The middle line in Figure 7.8 shows a video server with 1000 total movies where each movie is duplicated. When there are two copies of each movie, there is more disk bandwidth available to schedule movie streams, but the maximum wait time for a free slot is the same as before, since two copies of the movie will reside 1000 disks apart. Therefore, duplicating the movies doesn't help much in increasing the number of concurrent streams. The top line in the graph shows the performance of a disk server that replicates movies according to the Zipf's Law distribution of requests. In this case replication helps dramatically, since there are now many copies of the most popular movies scattered throughout the array, reducing response times at heavy loads. The Zipf replication pattern achieves close to full utilization of the disk farm. We argued earlier that smaller partitions of disk farms have reliability advantages as well as reducing the need for replication.

Table 7.3 shows the cost-per-stream of the different striped configurations. For disk farms of size 10 and 100 movies, the cost per stream does not vary with the replication pattern, since all the disks are fully utilized. With 1000 total movies, replication to reflect the access pattern is required to provide inexpensive streams.

Compared to the one-movie-per-disk scheme, the striped video server supports

	10 movies			100 movies			1000 movies		
	1-Per	Striped	Ratio	1-Per	Striped	Ratio	1-Per	Striped	Ratio
1 copy	14	50	3.6	23	492	21.4	34	2774	81.6
2 copies	28	100	3.6	49	997	20.3	73	3515	48.2
Zipf	54	90	1.7	467	1248	2.7	5513	14415	2.61

Table 7.4: *Comparison of number of streams supported by one-movie-per-disk and striped video servers. Ratio is (striped streams : one-per-disk streams).*

	10 movies			100 movies			1000 movies		
	1-Per	Striped	Ratio	1-Per	Striped	Ratio	1-Per	Striped	Ratio
1 copy	$786	$220	0.28	$4783	$224	0.047	$32353	$397	0.012
2 copy	$786	$220	0.28	$4490	$221	0.049	$30137	$626	0.021
Zipf	$367	$220	0.60	$601	$225	0.374	$652	$240	0.378

Table 7.5: *Comparison of cost per stream for one-movie-per-disk and striped video servers.*

many more streams, as shown in Table 7.4. This is true even when the movies are replicated in the non-striped system to reflect the access pattern; for a pattern of duplication corresponding to the Zipf distribution, the striped disk farm supports a factor of 1.7 to 2.7 more streams than the one-movie-per-disk configuration. Because accesses to the least popular movies are unlikely to fully utilize the disks on which these movies reside, substantial disk bandwidth is wasted in the one-movie-per-disk scheme. The striped video server offers higher disk utilization and lower cost per stream. The latter is shown in Table 7.5.

7.3 Storage Hierarchies for Video Service

In the last section, we showed that a video server composed of magnetic disks uses the relatively inexpensive bandwidth of disks to provide low-cost video streams. But, although disk bandwidth can be considered inexpensive, the cost of storage for a movie on disk is high compared to the cost of storage in a tape library. For video servers designed to hold massive amounts of information, a less expensive alternative to a huge disk farm is a storage hierarchy containing both magnetic disks and tertiary storage. A storage hierarchy should be well-suited for use in a video server because we expect accesses to the server to be highly localized; this locality would allow the most popular objects to reside on the fastest, most expensive level of the hierarchy, with less popular objects migrating to less expensive levels.

In this section, we explore the use of tertiary storage within a storage hierarchy for video service. For a Zipf's Law distribution of movie requests, as described in Chapter 6, we conclude that a hierarchy that includes tertiary storage will not perform well enough to service the workload. Despite the low cost of storing movies on tertiary storage, the bandwidth of the current generation of tape libraries and optical disk jukeboxes is so low, the average response time for requests is so long and the average cost per video stream is so high that hierarchy performance is inadequate. We evaluate various changes in tertiary storage systems that would make them more attractive for video service; these include a larger number of drives in tape libraries and optical disk jukeboxes and bandwidth and response time improvements for tape drives. In Section 7.3.4, we also look at different patterns of locality in movie request distributions and demonstrate that the current generation of tertiary storage products will be adequate to service extremely highly-localized access patterns.

7.3.1 Storage Hierarchy Simulations

In the simulation results that follow, we assume a video server storage hierarchy that contains 1000 movies. Some portion of these movies is stored on disk and the remainder in tertiary storage. For these simulations, we assume a simple partitioning of the system and do not simulate migration of movies between levels of the storage hierarchy. For example, if 40% of the movies are stored on disk and the rest on tape, we assume that the movies stored on disk are the most popular and frequently-requested movies. If we assume a standard Zipf's Law distribution of requests to movies and assume that there are 2000 video streams active in the server at a given time, we can predict the proportion of the streams that will be

Parameter	Value
Movies	1000
Simultaneous Streams	2000
Request Distribution	Zipf's Law

Table 7.6: *Simulation parameters for the example used in the remainder of this chapter.*

Movies in Tertiary Store (1000 Total)	Streams Serviced by Tertiary Store (2000 Total)
200	42
400	122
600	238
800	418
1000	2000

Table 7.7: *Shows the number of requests serviced in disk and tertiary storage portions of the hierarchy in a video server containing 1000 movies when 2000 streams are active.*

serviced by the disk farm and the proportion that will "fall through" to be serviced by the tertiary storage portion of the hierarchy. Table 7.6 summarizes the simulation parameters used in this section, and Table 7.7 shows the number of requests that fall through to the tertiary storage level of the hierarchy. We use the right column of Table 7.7 to help evaluate the effectiveness of particular tertiary storage libraries for video service.

We simulate only the tertiary storage portion of the hierarchy and ignore requests to movies on disk. The tertiary storage system therefore sees the tail of the Zipf's Law distribution, and except when all movies are stored on tape, this distribution is flatter and has less locality than the overall movie request distribution. Because we assume that the distribution is flatter than the overall distribution, we do not simulate replication in the tertiary store.

The simulation results presented below use the simulator described in Chapter 3. All are closed simulations, with the load or number of concurrent processes kept constant throughout a simulation run. A simulation run continues until the mean response time of a request reaches an equilibrium represented by a 95% confidence interval. The metric used for evaluation is the number of concurrent streams that can be sustained by the tertiary storage system while requiring that 95% of all requests start displaying video within a specified response time constraint. Requests are issued to the video server according to a Zipf's Law distribution for 1000 movies; the requests not serviced by the disk farm fall through to the tertiary storage system.

7.3.2 Tape Libraries in the Hierarchy

We evaluate two tape libraries for their performance in a video server storage hierarchy: the relatively low-performance, inexpensive Exabyte EXB120 with four tape drives, 116 tapes and a total transfer rate of 2 megabytes per second, and a high-performance library modeled on the Ampex DST600 with four tape drives, 256 tapes and aggregate bandwidth of 60 megabytes per second. These libraries are described in Chapter 4.

Performance of the Exabyte EXB120 Library

Figure 7.9 shows the performance of the Exabyte EXB120 in a video server storage hierarchy. The vertical axis shows the number of streams serviced by the tape library when 95% of requests must be serviced within an hour. (We chose such a long response time constraint because, as discussed in Chapters 3 and 4, tape libraries have very long access times.) The horizontal axis shows the percentage of the total of 1000 movies that are stored on tape, with the remainder of the movies stored on disk. A single EXB120 library holds approximately 200 movies, or about 20% of the total; as a result, increasing values on the horizontal axis represent the simulation of additional tape libraries.

The top line in the graph shows the values from the rightmost column of Table 7.7, the requests that *should* be serviced by the tape library; we assume that the remainder are serviced at the magnetic disk level of the storage hierarchy. The bottom line in the graph shows the requests *actually* serviced, which are far fewer than required by the application. In

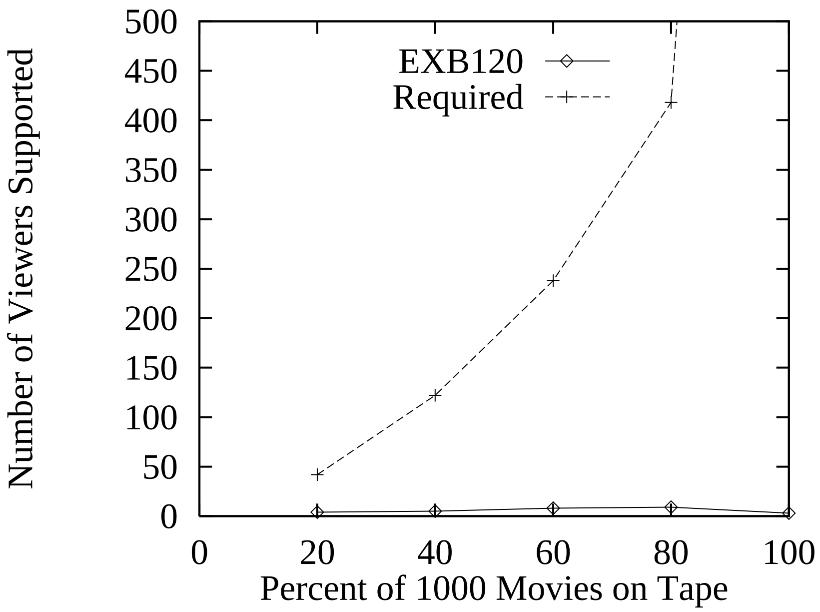


Figure 7.9: *Performance of Exabyte EXB120 is low compared to the requirements of the application when 1000 movies and 2000 streams are processed by the server.*

fact, a single EXB120 library only sustains 4 concurrent streams given the 1-hour response time constraint. (By contrast, a single disk can support more simultaneous streams.) This suggests a grave imbalance between the number of movies that can be stored in the EXB120 (200) and the number of streams that can be serviced (4). If the library stores 200 movies, table 7.7 indicates that the library should support 42 concurrent streams. By contrast, using similar calculations, if only 4 streams are supported by the library, this limits the number of movies that can practically be stored and serviced to approximately 15. This number is far too small to make a storage hierarchy useful. We define the *practical storage cost* for a tertiary storage system as the cost per megabyte of storage for the number of movies that can be adequately serviced by the library. Thus, for an EXB120 that costs approximately \$100,000, the practical storage cost (based on 15 movies that can be serviced) is \$3 per megabyte. By contrast, if all 200 movies that could be stored in an EXB120 could be adequately serviced, the practical storage cost would drop to \$.22 per megabyte. Disk storage costs are approximately \$.50 per megabyte. Because four streams can be serviced concurrently, the per-stream cost for the library is $25,000, two orders of magnitude higher than for magnetic disks.

The precipitous drop in performance for the simulation when the number of movies in the tertiary store goes from 80% to 100% requires explanation. This drop in performance occurs because the tertiary storage system is being asked to handle all requests for movies, even the most highly-localized part of the distribution. The tape library is unable to service this demand and the response times increase dramatically.

EXB120 with Additional Tape Drives

The downfall of the EXB120 library is a lack of sufficient bandwidth for servicing streams and contention for the small number of tape drives. Figure 7.10 shows an EXB120 simulated with 20 tape drives rather than 4. The performance improves considerably, although it is still well below that required for the application. A single EXB120 now has a total aggregate bandwidth of 10 megabytes per second and can service 21 streams within the specified response time limit. The higher bandwidth of the 20 tape drives increases the number of movies that can be practically serviced by a single EXB120 from 15 to 119. The cost per stream (assuming a cost of $3000 for each additional tape drive) drops to $7048 and the practical cost of storing movies drops to $.55 per megabyte.

Faster EXB120 with Additional Tape Drives

Next, we simulate not only twenty drives in each EXB120 robot, but also tape drives and robot arms that have increased in speed by a factor of ten. Figure 7.11 shows the performance of this configuration. For up to 60% of the total storage in the tape library, this improved tape system is adequate to provide the required performance.

A single EXB120 now supports 96 video streams, or enough to service up to 333 movies on tape. Since this exceeds the number of movies an EXB120 can hold, the practical

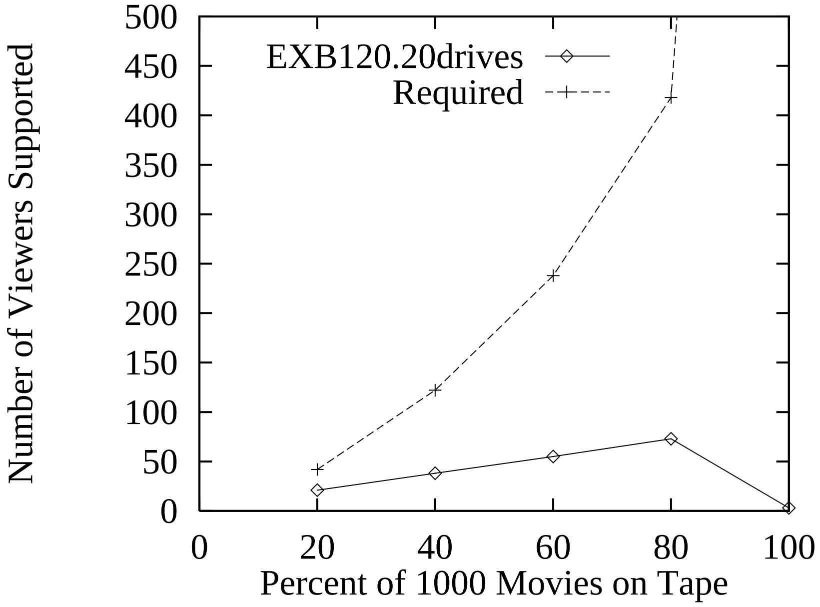


Figure 7.10: *Shows the number of concurrent streams supported by the EXB120 robot with 20 tape drives compared to the number required by the application. The bandwidth of the additional tape drives increases the number of concurrent streams compared to the 4-drive configuration, but the concurrency achieved is still below what is required.*

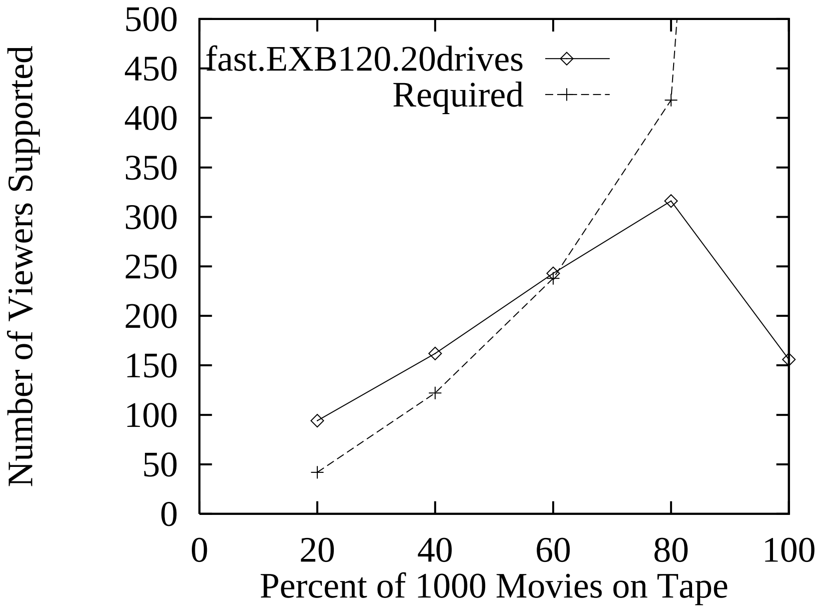


Figure 7.11: *Shows the performance of and EXB120 robot with 20 drives where tape drives and robot are sped up by a factor of ten. The number of concurrent streams exceeds the requirements for up to 60% of the movies stored in the tape library.*

storage cost (assuming each improved drive still costs approximately $3000) is that of 200 movies, or $0.22 per megabyte. The cost per stream falls to $1541. This is still considerably higher than the cost per stream of magnetic disk.

From these simulations, we conclude that to service a standard Zipf distribution, movies-on-demand workload, the EXB120 tape library requires more tape drives and faster tape drives and robot arms, even to satisfy the questionable service time guarantee of one hour.

High Performance Tape Library

In this section, we run similar performance simulations for a high performance library that contains four tape drives that each transfer at 15 megabytes per second. Figure 7.12 shows the resulting performance, where the top line in the graph again represents the required performance and the bottom line in the graph shows the actual performance. Since the high performance library, modeled on the Ampex DST600 library, holds 256 tapes that each hold 25 gigabytes of storage, all 1000 movies can fit in a single library. As a result, moving along the horizontal axis in the graph does not represent an increase in the number of robots or tape drives. A single robot costs about $1,000,000 with four tape drives.

Figure 7.12 shows that a single library supports up to 51 concurrent streams, resulting in a per-stream cost of approximately $20,000. 51 streams are adequate to service approximately 220 movies stored on tape; this corresponds to a practical storage cost of $2 per megabyte.

Like the EXB120 tape library, this higher-performance library has too few readers to service all the movies it can store. Unfortunately, the high performance tape drives are so expensive (approximately $150,000 each) that adding tape drives to balance the system is likely to be prohibitively expensive.

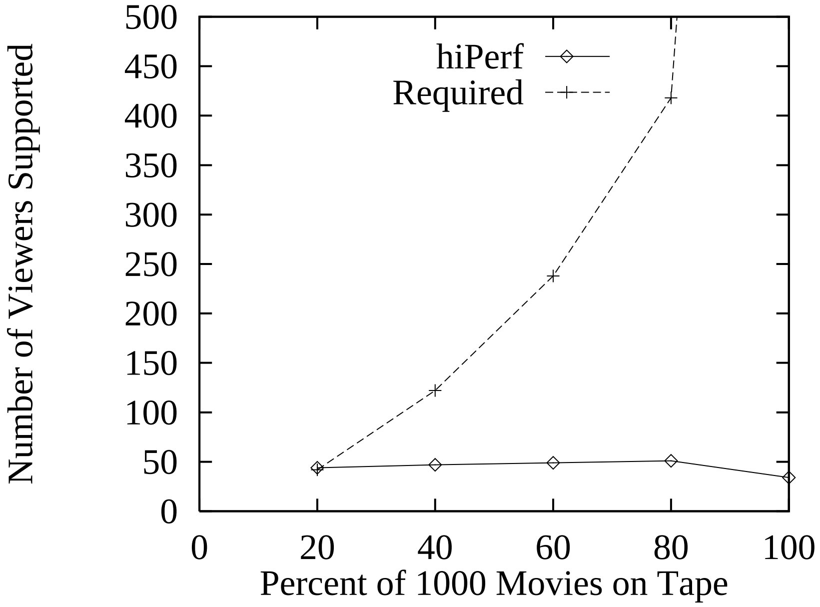


Figure 7.12: *Shows the performance of the high performance library compared to the required performance for the application. Although the high performance library has drives that deliver 15 megabytes per second, the total system bandwidth is not adequate to service all the movies that can be stored in the array.*

Hypothetical Tape Robots

Figure 7.13 shows the performance of the two hypothetical tape libraries described in Section 5.4.5 on the video server application. Again, the graph line labeled "24 drives" shows a tape library similar to the EXB120 robot, except with 24 tape drives and robot operations about four times faster than the EXB120 robot, for a tape-to-drive ratio of 5-to-1. The line labeled "40 drives" shows performance for the same library with 40 drives, a 3-to-1 ratio of tapes to drives. In both cases, additional drives increase the number of streams supported, but the libraries still supply far fewer streams than are required by the application. Faster tape drives are required.

7.3.3 Optical Disk and Storage Hierarchies for VOD

In this section, we explore the possibility of using optical disk rather than magnetic tape libraries to support video-on-demand. We use the Hewlett-Packard HP120 optical disk

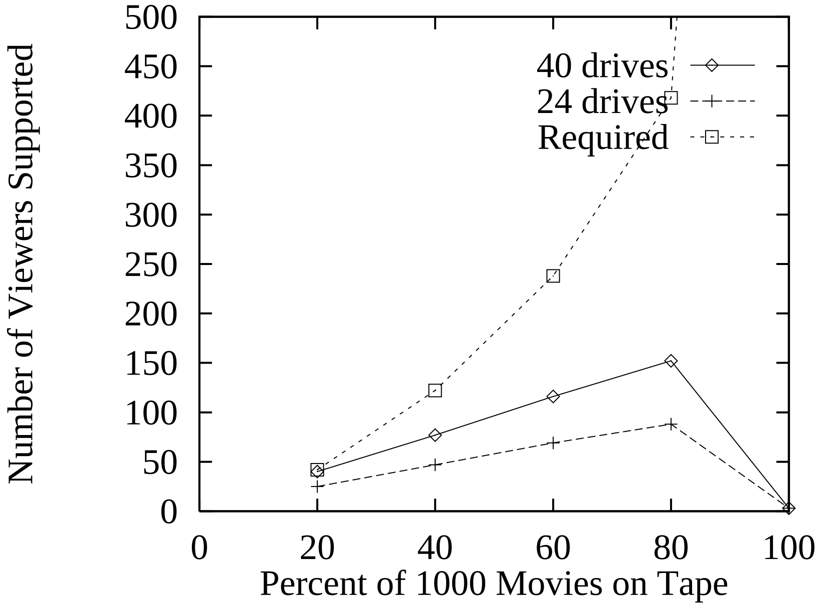


Figure 7.13: *Shows the performance hypothetical tape robots with 20 and 40 tape drives.*

jukebox in these simulations.

Hewlett-Packard HP120

The HP120 holds approximately 120 gigabytes of total storage on 88 rewritable magneto-optical disk platters holding 1.3 gigabytes of storage each. The system contains four optical disk drives that each transfer at 1.6 megabytes per second on reads and 0.8 megabytes per second on writes. List price for the system is $52,500. The price for optical disk drives is $4100.

The same simulations are run for the HP120 as were run for the tape libraries in the last section. Some portion, shown on the horizontal axis of the graphed results, of the movies is stored on optical disk; the remainder on magnetic disk. Movie requests are made to the video server according to the Zipf's Law distribution. The simulations have a response time constraint of one hour.

Figure 7.14 shows the performance of a collection of HP120 optical disk jukeboxes. The vertical axis shows the number of concurrent streams with a response time limit of one

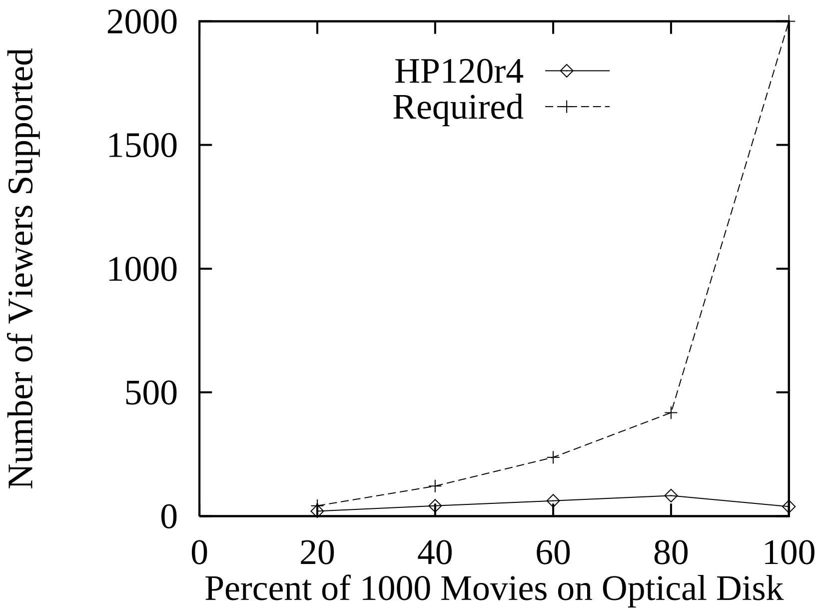


Figure 7.14: *Shows the performance of the HP120 optical disk jukebox on the video server application. This performance does not meet that required by the application, shown on the top line in the graph.*

hour, and the horizontal axis shows the percentage of movies in tertiary storage. As in the graphs from the last section, the top line in the graph shows the requirements of the tertiary storage system: the number of accesses that fall through the magnetic disk level of the hierarchy and must be serviced by the tertiary store. The lower line shows the number of streams actually serviced by the jukebox. As before, the actual performance of the jukebox is considerably lower than is required. When 20% or 200 of the movies are stored on optical disk, the resulting collection of four HP120 robots services about 19 concurrent streams. This corresponds to a per-stream cost of $11,052. The 19 streams supported by the hardware are enough to service 111 movies, resulting in a practical storage cost for the optical disk jukebox of $.84 per megabyte.

Again, performance is limited by system bandwidth. Figure 7.15 shows the performance of the HP120 with 20 optical disk drives rather than 4. The additional bandwidth alleviates the mismatch between the number of movies that can be stored and the number of

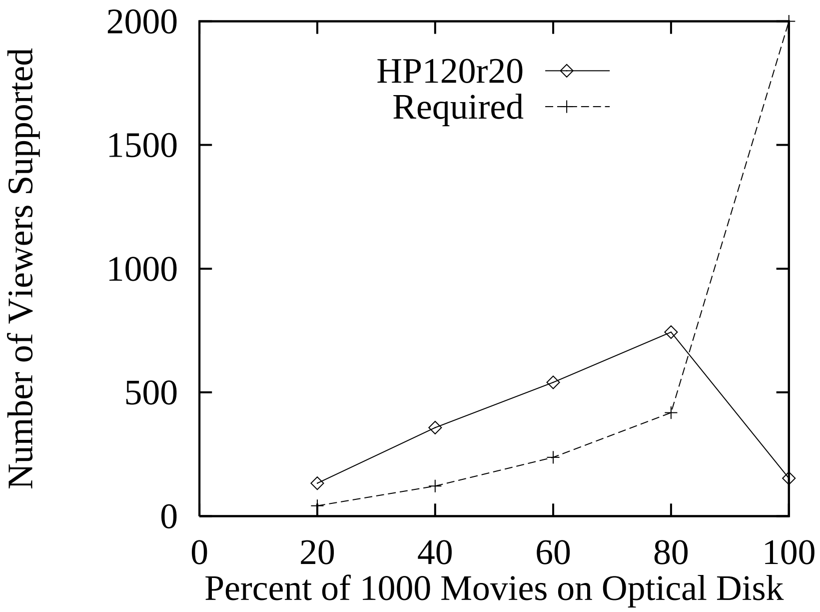


Figure 7.15: *Shows the performance of the HP120 optical disk jukebox with 20 optical disk drives instead of 4. Now performance does exceed the application requirements.*

video streams that can be serviced. Now a collection of four HP120s that stores 200 movies can support 127 streams, three times the number required by the application. Assuming that additional disk drives cost $4100 each, this corresponds to a practical storage cost of $0.26 per megabyte and a cost of $929 per video stream.

7.3.4 Different Locality Patterns

We have shown that current tertiary storage technology is inadequate to support a standard Zipf Distribution of requests. We chose to model movie requests using the Zipf distribution based on a small amount of video store rental history and because the distribution has been used successfully to model human behavior in similar applications, for example, in the way materials are checked out of a conventional library. Our choice is based on very little practical experience. In fact, no one developing video servers has very much knowledge about how a large population of viewers will request movies; many video server trials are underway to gather just such information. It is possible that the real distribution of accesses to a video server will be substantially different from our predictions, and may indeed be even more highly-localized than the standard Zipf distribution. A drastically different locality distribution might change our results dramatically. For example, if requests are much more highly-localized than we predict, then a disk cache would capture a much larger proportion of the requests, and the resulting demands on the tertiary storage system would be reduced. Under such circumstances, current tertiary storage products may perform adequately to service the request distribution, although customers will still have to tolerate long response times and high cost per video stream.

Knuth identifies a parameterizable variation of the Zipf distribution. The parameter θ, where θ is a small positive number, is used to vary the locality of the distribution. The probability that the n^{th} of N movies will be requested is p_n, where:

$$p_n = c/n^{(1+\theta)}$$

for

$$c = 1/\sum_{i=1}^{N}(1/i^{(1+\theta)})$$

Figure 7.16 shows the parameterizable Zipf distribution for four different values of θ. Again, we use an example where there are 1000 total movies and determine the number of streams that would fall through to the tertiary storage system out of a total of 2000 active streams. The standard Zipf distribution corresponds to a value of $\theta = 0$.

As θ increases, the distribution becomes more highly-localized, with a larger and larger proportion of accesses going to a smaller and smaller number of movies. For example, Table 7.8 gives a measure of the locality of the distribution by indicating that 80% of movie requests go to some number of the most popular movies. The standard Zipf distribution, where $\theta = 0$, is commonly called an 80/20 distribution. In other words, 80% of the accesses go to approximately the most popular 20% of movies. By contrast, when $\theta = 1.0$, 80% of

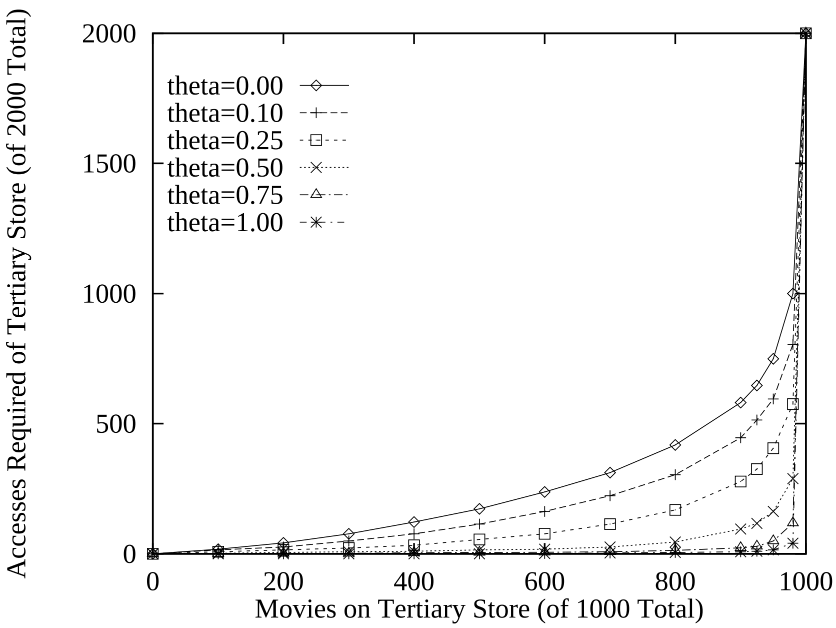


Figure 7.16: *Shows locality patterns for a parameterizable Zipf distribution for different values of the parameter, theta.*

θ	number of popular movies that get 80% of requests
0.00	224
0.25	52
0.50	12
0.75	5
1.00	3

Table 7.8: *Indicates the locality of the distribution for a given value of θ; shows the number of the most popular movies that receive approximately 80% of all movie requests.*

requests go to the top 3 movies, or the most popular 0.3%.

As is evident in the Figure 7.16, the more localized distributions put fewer demands on the tertiary storage portion of the hierarchy. For example, for $\theta = 0.75$, if 200 movies are on tape and 800 on disk, only 5 streams on average need be serviced by the tertiary store. Since the EXB120 can service 4 streams with a response time constraint of one hour, and the high performance tape library and the optical disk jukebox can service considerably more than five streams, such a distribution is obviously better matched to the capabilities of current tertiary system than a standard Zipf distribution. The performance of the EXB120 achieved in Figure 7.9 is satisfactory for a workload with a value of $\theta = 0.75$ for up to 60% of movies on tape, and with a value of $\theta = 0.80$ for up to 80% of movies on tape.

7.3.5 Summary of Storage Hierarchies

We have examined the use of storage hierarchies for video-on-demand servers where the tertiary portion of the hierarchy is a tape library or an optical disk jukebox. In both cases, we found that the current generation of library devices is inadequate to support a standard Zipf distribution of movie requests because the small number of drives in each system limits the concurrency that can be achieved. In addition, in the case of tape libraries, we found that we not only need more tape drives but also faster tape drives and robots for the library to support the application. Even with these improvements, in every case the cost per video stream for tertiary storage is significantly higher than for magnetic disks. Finally, we demonstrated that the current generation of tertiary storage products is adequate to support a more highly-localized distribution of movie requests. To the extent that real movie request patterns correspond to these more highly localized distributions, it will be feasible to use tertiary storage to reduce overall cost of storing a large number of movies.

7.4 Future Work

There are many unanswered questions about the design of storage systems for video service. We mention three.

One open research question is the effect of including VCR-like operations such as search, rewind and pause. In a disk farm, these operations will present additional complexity, especially for a striped video server. For example, a fast forward operation on a striped disk system will require reading intervals from the disks in a different order than the strict round-robin ordering of normal play, requiring the scheduler to find free intervals on disks holding the necessary movie blocks. Because tape systems are sequential rather than random access in nature, it is unlikely that a tape system would provide VCR-like operations.

For striped disk farms, another open research question is the proper interleave factor for striping. In our studies, we have used an interleave unit of half a second; this is a logical number since it is the minimum amount of movie time over which MPEG-II encoding is performed. Advantages to interleaving in larger units are that more data are fetched per disk seek and rotation operations, resulting in more efficient use of disk bandwidth. The disadvantage to larger disk units is that more buffer space is required to store the larger increments of data until they are performed for the viewer. (This buffering may be performed at the video server or in the set-top box at the customer's home; to minimize the cost of the set-top box, most manufacturers plan to include only a small amount of buffer memory.)

A study of optimum partitioning of the disk farm is needed. We have already discussed the advantages of partitioning the disk farm to bound wait times for a free scheduling slot and to limit the vulnerability of streams to disk failures.

7.5 Summary

In this chapter, we have examined the design of storage systems to support video-on-demand service. For entirely disk-based systems, we compared two schemes: one movie stored on each disk and movies striped across a collection of disks. For the former, we found that popular movies creating hot spots on particular disks that can limit overall system performance. Replication of popular movies to reflect the pattern of movie requests is essential to cost-effective performance. For the latter scheme, we showed that striped video servers use disk bandwidth much more effectively to support a greater number of streams than the one-movie-per-disk configuration. We showed that for small disk farms, replication in a striped system is unnecessary; the entire disk farm can be fully utilized. For larger striped disk farms, replication is necessary to avoid long delays in starting video streams. We argued that VCR-like operations and recovery from failed disks will be somewhat more complicated to schedule in a striped disk video server.

We also evaluated the use of tertiary storage as part of a video server storage hierarchy. The motivation for using a hierarchy is the lower cost per megabyte of storage for magnetic tape libraries and optical disk jukeboxes compared to magnetic disks. Unfortunately, we showed that the performance of the current generation of tape libraries and optical disk jukeboxes is so poor that they can support very few concurrent streams. If we assume a standard Zipf distribution of movie requests, then the performance of the current generation of tertiary storage libraries is inadequate to service the workload. We evaluated various improvements to tape libraries and optical disk jukeboxes that would improve their performance, including more drives for either type of robot and faster drives and robots for tape libraries. Although these improved robots do support more streams and lower the cost per megabyte of storage, in all cases, the cost per video stream for tertiary storage is substantially higher than the cost per stream of disk systems.

Finally, we questioned our own assumption of the Zipf's Law distribution of movie requests. If the access patterns of future video servers are in fact more highly-localized than indicated by the Zipf's Law distribution, then tertiary storage systems may be incorporated into storage hierarchies to lower the costs of storing large amounts of video material. Customers of such systems, however, would still have to tolerate the high cost per stream and long response times characteristic of current tertiary technologies when accesses were serviced by the tertiary level of the hierarchy. Applications that could tolerate both high stream costs and long response times are likely to represent a small fraction of the total commerce in video-on-demand. For the vast remainder of applications, and especially if our assumptions about the locality of movie request patterns are correct, we predict that for the foreseeable future, most commercial video servers will be composed entirely of magnetic disks.

Chapter 8

Conclusions

This thesis has focused on an area of computer systems that has long been neglected: tertiary storage. We began by describing the technologies, the tradeoffs in tertiary system design, and their traditional applications. We then evaluated their usefulness in new configurations, like striping, and in new applications, such as digital libraries and video servers.

We evaluated the technique of data striping in tape arrays. Data striping interleaves data from individual files across a collection of tapes so that they may be accessed by several tape drives in parallel. The technique has been used quite successfully in disk arrays, and we found that striping is also effective in tape arrays for improving the performance of sequential workloads or for workloads in which a single request is active. However, striped tape systems perform poorly for applications in which there are several non-sequential, concurrent requests active in the tape library. This poor performance is caused by contention for the small number of tape drives in a typical tape library. Striping exacerbates this contention because striped accesses are spread across several tapes, each of which must be loaded into a tape drive. Our evaluation showed the need for a higher ratio of tape drives to tapes in a library. Changes in tape drive performance would also affect the need for and effectiveness of striping. Higher bandwidth tape drives reduce the need for striping; conversely, faster tape switch times reduce the penalties for striping and make it more desirable.

Next, we characterized two new workloads: video-on-demand servers and digital libraries. For video service, we identified typical compression schemes and described the expected object sizes and performance guarantees determined by the choice of compression scheme. To predict customer request patterns to a video server, we looked at video store rental histories, which support our hypothesis that requests will be highly localized. We proposed a Zipf's Law distribution to model the pattern of movie requests. Next, we looked at a range of video server population sizes, from a classroom to a cable company with 100,000 subscribers. We predicted a response time goal of approximately 10 seconds for most video servers. In a movies-on-demand server, we predict that most customers will request movies in their entirety; a small percentage will perform VCR-like operations such as pause and rewind.

For the digital library workload, we argued that to justify claims of increased performance, the size of material stored in a digital library should scale. More accesses to the digital library imply that more users are making requests; a larger user population will likely demand a greater diversity of material. We looked at a number of university libraries to deduce a relationship between the amount of material in a library and the population using that library. To get information on locality in the access pattern of a digital library, we looked at usage of a bibliographic database and found that database index files are accessed heavily. Response time requirements of several seconds are typical in digital libraries. Unfortunately, few traces of user access patterns to a full text library are available, so request distributions remain a subject for future research.

We evaluated two alternatives for providing storage in a movies-on-demand video server: disk farms and storage hierarchies. In our study of disk farms, we examined two design choices: storing one movie on each disk and striping movies over a group of disks. Storing one movie on each disk is a straightforward scheme that results in simple scheduling. However, the scheme wastes substantial disk bandwidth, since there is a high degree of locality in the movie request pattern, and the disks holding less popular movies will be under-utilized. We showed that cost-effective performance in the one-movie-per-disk scheme depends on replication of movies to reflect the user request pattern.

Striping movies within a video server disk farm results in much better performance, since it achieves greater load balancing and allows all disk bandwidth to be devoted to whatever movies are in highest demand. As a result, a striped video server supports many more customers than a non-striped system. However, scheduling a striped video server is somewhat complicated compared to a one-movie-per-disk scheme. Replication of movies in a striped video server is not necessary to achieve full utilization of a relatively small (say, 100 disks) disk farm; for larger disk farms, replication may be necessary to avoid long service delays.

Last, we looked at the use of storage hierarchies that include magnetic tape or optical disk along with a disk farm to provide storage for a video server. A storage hierarchy appears to be advantageous compared to storing all movies entirely on disk, since despite the relatively inexpensive bandwidth of disks, the cost to store data on disk is much higher than on tapes or optical disk platters. A storage hierarchy would allow disks to service the most localized portion of the request distribution, while access to less popular movies would be serviced by the tertiary storage system. Unfortunately, we showed that the performance of neither magnetic tape libraries nor optical disk jukeboxes is adequate to service the tail of a standard Zipf's Law distribution of movie accesses. Tape libraries need more and faster tape drives and optical jukeboxes need more disk drives for their performance to be adequate to service this workload. Finally, we questioned our own assumption that the request distribution would correspond to a standard Zipf distribution. There is very little data available on how customers will actually request movies. If, in fact, accesses are more highly localized than the standard Zipf distribution, then current tertiary storage systems may perform adequately.

Throughout this dissertation, we have identified several desirable changes in tertiary storage systems:

- Tertiary storage libraries are unbalanced systems, typically holding vast numbers of tapes or optical platters but only a few drives for accessing them. These systems have very low storage costs, making them ideal for backup and archival applications. To service new applications, these systems must be redesigned with a higher ratio of drives to media so they can support more total bandwidth and a higher concurrency of accesses.

- Higher bandwidth is required both from inexpensive tape drives and from optical disk drives. In the video-on-demand application, for example, the inexpensive cost of storing data on tape was not nearly as important as the inexpensive bandwidth provided by disks, since bandwidth determines the number of video streams that can be supported. Tertiary storage systems will not replace disks, but they must perform better to comprise a useful part of a storage hierarchy.

- Faster access times in tertiary libraries would dramatically improve their usefulness. As long as tape switch operations take several minutes, tapes will be inadequate for many classes of applications. Any improvements in mechanical operations such as load and eject, or any policies such as periodic eject zones that enable faster search and rewind times, are highly desirable.

- Finally, although not currently a bottleneck in most systems, the performance of robots should not be ignored. In any application that requires frequent tape switch operations, sequential operations by the robot arm threaten to become a performance bottleneck if other components of the storage system are improved.

Bibliography

[1] The Future Today: DAT Technology Computer Peripherals. Background paper, Archive Corporation, 1650 Sunflower Avenue, Costa Mesa, CA 92626, 1989.

[2] TPC Benchmark B Standard Specification. Technical report, Transaction Processing Performance Council, August 1990.

[3] Blockbuster Entertainment 1992 Annual Report. Blockbuster Entertainment Corporation, Fort Lauderdale, Florida, 1992.

[4] Tasuya Adachi, Dazuo Arai, Kinji Kawamoto, Hideshi Taki, and Kazuhiro Murase. A Fast Random Accessing Scheme for R-DAT. *IEEE Transactions on Consumer Electronics*, CE-33(3), August 1987.

[5] alt.cd-rom newsgroup. Frequently asked questions about CD-ROMs. Available by ftp from ftp.apple.com, March 1994.

[6] David P. Anderson and George Homsy. A Continuous Media I/O Server and Its Synchronization Mechanism. *IEEE Computer*, 24(10):51–57, October 1991.

[7] Robert A. Bartolini. Optical Recording: High-Density Information Storage and Retrieval. In *Proceedings of the IEEE*, pages 516–524, June 1982.

[8] Jim Barton. SGI, February 1994. Personal communication.

[9] Richard A. Baugh, David J. Bromley, and Bruce F. Spenner. Extremely Low Error Rate Digital Recording with a Helical Scan Recorder. *IEEE Transactions on Magnetics*, 22(5), September 1986.

[10] Kelly J. Beavers. Helical Scan Recording Technology: 8mm Evolves. *SunTech Journal*, September/October 1990.

[11] John Berbert, Ben Kobler, P. C. Hariharan, Jen-Jacques Bedet, and Alan M. Dwyer. Magnetic media. Review of magnetic media and recording systems from NASA Earth Observing System Data and Information System (EOSDIS) group at the NASA Goddard Space Flight Center, Greenbelt, MD, and Hughes STX Corporation, Greenbelt, MD., August 1993.

[12] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, and Bernd Pollermann. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1(2), Spring 1992.

[13] Bharat Bhushan. *Tribology and Mechanics of Magnetic Storage Devices*. Springer-Verlag, New York, 1990.

[14] Don L. Bosseau. Anatomy of a Small Step Forward: The Electronic Library Reserve Book Room at San Diego State University. *The Journal of Academic Librarianship*, 18(5):366–368, January 1993.

[15] Peter C. Boulay. The LaserTape 'DOTS': Next Digital Paper Product. Technical Report IDC Washington Publication No. W1220, IDC Washington, Inc., July 1990.

[16] Wiliam M. Callicott. Data management in NOAA. In *Goddard Conference on Mass Storage Systems and Technologies*. NASA/Goddard Space Flight Center, September 1992.

[17] Yee-Hsiang Chang, David Coggins, Daniel Pitt, David Skellern, Manu Thapar, and Chandra Venkatraman. An Open-Systems Approach to Video On Demand. *IEEE Communications Magazine*, May 1994.

[18] Peter M. Chen, Garth A. Gibson, Randy H. Katz, and David A. Patterson. An evaluation of redundant arrays of disks using an amdahl 5890. In *Proceedings SIGMETRICS*, pages 74–85, May 1990.

[19] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. RAID: High-Performance, Reliable Secondary Storage. Technical Report UCB/CSD 93/778, University of California at Berkeley, November 1993.

[20] Peter M. Chen and David A. Patterson. Maximizing performance in a striped disk array. In *Proceedings International Symposium on Computer Architecture*, May 1990.

[21] E. I. Cohen, G. M. King, and J. T. Brady. Storage Hierarchies. *IBM Systems Journal*, 28(1), 1989.

[22] Sam Coleman and Steve Miller, editors. *Mass Storage System Reference Model: Version 4*. IEEE Technical Committee on Mass Storage Systems and Technology, May 1990.

[23] Bill Collins. High-Performance Data Systems. In *Digest of Papers*. Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[24] Penelope Constanta-Fanourakis, Ken Kaczar, Gene Oleynik, Don Petravick, Margaret Votava, Vicky White, George Hockney, Steve Bracker, and Jussara M. De Miranda. Exabyte Helical Scan Devices at Fermilab. *IEEE Transactions on Nuclear Science*, 36(5), October 1989.

[25] Daniel R. Dauner, Raymond C. Sherman, Michael L. Chreistensen, Jennifer L. Methlie, and Jr. Leslie G. Christie. Mechanical Design of an Optical Disk Autochanger. *Hewlett-Packard Journal*, December 1990.

[26] Ann L. Drapeau and Randy H. Katz. Striped Tape Arrays. In *Digest of Papers.* Twelfth IEEE Symposium on Mass Storage Systems, April 1993.

[27] Laine Farley. Usage patterns in the COMP database, October 1993. Personal communication.

[28] Craig Federighi and Lawrence A. Rowe. A Distributed Hierarchical Storage Manager for a Video-on-Demand System. *IS&T/SPIE Symposium on Elec. Imaging Science and Technology*, February 1994.

[29] Joy Foglesong, George Richmond, Loellyn Cassell, Carole Hogan, John Kordas, and Michael Nemanic. The livermore distributed storage system: Implementation and experiences. In *Digest of Papers.* Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[30] Garth Alan Gibson. *Redundant Disk Arrays: Reliable, Parallel Secondary Storage.* PhD thesis, U. C. Berkeley, April 1991. Technical Report No. UCB/CSD 91/613.

[31] H. Goto, A. Asada, H. Chiba, T. Sampei, T. Noguchi, and M. Arakawa. A New Concept of Data/DAT System. *IEEE Transactions on Consumer Electronics*, 35(3), August 1989.

[32] Jim Gray, Bob Horst, and Mark Walker. Parity striping of disc arrays: Low-cost reliable storage with acceptable throughput. In *Proceedings Very Large Data Bases*, pages 148–161, 1990.

[33] Peter Hansen and Heinrich Heitmann. Media for Erasable Magnetooptic Recording. *IEEE Transactions on Magnetics*, 25(6), November 1989.

[34] J. P. Harris, W. B. Phillips, J. F. Wells, and W. D. Winger. Innovations in the Design of Magnetic Tape Subsystems. *IBM Journal of Research and Development*, 25(5), September 1981.

[35] Roger Haskin. The Shark Continuous Media File Server. In *Proceedings IEEE COMPCON*, pages 12–15, 1993.

[36] Roger Haskin. Shark and Tiger Shark video-on-demand systems, IBM Almaden Research Center, April 1994. Personal communication.

[37] Harry C. Hinz. Magnetic Tape Technology in the 1990s. In *Digest of Papers.* Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[38] Akihiko Hitomi and Tetsuya Taki. Servo Technology of R-DAT. *IEEE Transactions on Consumer Electronics*, CE-32(3), August 1986.

[39] Carole Hogan, Loellyn Cassell, Joy Foglesong, John Kordas, Michael Nemanic, and George Richmond. The livermore distributed storage system: Requirements and overview. In *Digest of Papers.* Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[40] Fukuzo Itoh, Haruo Shiba, Masashi Hayama, and Takateru Satoh. Magnetic Tape and Cartridge of R-DAT. *IEEE Transactions on Consumer Electronics*, CE-32(3), August 1986.

[41] Raj Jain. *The Art of Computer System Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling.* John Wiley and Sons, Inc., New York, 1991.

[42] Fathilah Kamaluddin. Melvyl Statistics, University of California, Office of the President, November 1993. Personal communication.

[43] Masao Kawagishi and Takashi Niwa. Characterstics and Performance of Metal Particle Tape in a Broadcast Video Recorder. Video Recorder Division, Matsushita Electric Industrial Company Ltd., Kadoma City, Osaka, Japan.

[44] Kimberly Keeton, Ann L. Drapeau, David A. Patterson, and Randy H. Katz. Storage Alternatives for Video Service. In *Digest of Papers.* Thirteenth IEEE Symposium on Mass Storage Systems, June 1994.

[45] Rajesh Khandelwal. Bell Communications Research Video Server Prototype, May 1994. Personal communication.

[46] M. Y. Kim. Synchronized disk interleaving. *IEEE Transactions on Computers*, C-35:978–988, November 1986.

[47] Toshiyuki Kitahara. On the Long Term Storage of Metal Tapes. Fuji Photo Film Co. Magnetic Recording Lab; FIAT/IFTA Technical Commission in Hilversum (The Netherlands), June 14th 1988.

[48] T. Kitamoto, M. Nakamura, and S. Takayama. Magnetic Recording Tapes for Video Recording. Fuji Photo Film Co., Ltd.; SMPTE Private Committee Document.

[49] Ben Kobler and John Berbert. NASA earth observing system data information system (EOSDIS). In *Digest of Papers.* Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[50] Mark H. Kryder. Data Storage in 2000–Trends in Data Storage Technologies. *IEEE Transactions on Magnetics*, 25(6), November 1989.

[51] Thomas W. Lanzatella and Paul G. Rutherford. Storage management issues for cray research. In *Digest of Papers.* Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[52] David D. Larson, James R. Young, Thomas J. Studebaker, and Cynthia L. Kraybill. StorageTek 4400 Automated Cartridge System. In *Digest of Papers*. Eighth IEEE Symposium on Mass Storage Systems, May 1987.

[53] Ray Reed Larson. *Workload Characteristics and Computer System Utilization in Online Library Catalogs*. PhD thesis, University of California, Berkeley, March 1986.

[54] Duncan H. Lawrie, J. M. Randal, and Richard R. Barton. Experiments with Automatic File Migration. *Computer*, pages 45–55, July 1982.

[55] Edward K. Lee. *Performance Modeling and Analysis of Disk Arrays*. PhD thesis, University of California at Berkeley, September 1993.

[56] M. Livny, S. Khoshafian, and H. Boral. Multi-disk management algorithms. In *Proceedings SIGMETRICS*, pages 69–77, May 1987.

[57] S. B. Luitjens. Magnetic Recording Trends: Media Developments and Future (Video) Recording Systems. *IEEE Transactions on Magnetics*, 26(1), January 1990.

[58] John C. Mallinson. Tutorial Review of Magnetic Recording. *Proceedings of the IEEE*, 64(2), February 1976.

[59] John C. Mallinson. Achievements in Rotary Head Magnetic Recording. *Proceedings of the IEEE*, 78(6):1004–1016, June 1990.

[60] John C. Mallinson. Magnetic Tape Recording: Archival Considerations. In *Digest of Papers*. Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[61] William H. Tetzlaff Martin G. Kienzle, Dinkar Sitaram. Using a Storage Hierarchy in Movie-on-Demand Servers. *submitted to ACM Multimedia*, 1994.

[62] Michael McCarthy and Barry Green. Buyers Guide Part 2: Mass Storage OEMs. *SunWorld*, pages 69–90, January 1994.

[63] Fred McClain. DataTree and uniTree: Software for file and storage management. In *Digest of Papers*. Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[64] Fred W. McClain. Mass storage at the san diego supercomputer center. In *Digest of Papers*. Eighth IEEE Symposium on Mass Storage Systems, May 1987.

[65] Maureen McKeon. A Sampling of Optical Disk Drives. *SunExpert*, pages 63–71, June 1992.

[66] C. Denis Mee and Eric D. Daniel, editors. *Magnetic Recording, Volume II: Computer Data Storage*. McGraw-Hill, New York, 1988.

[67] C. Denis Mee and Eric D. Daniel, editors. *Magnetic Recording, Volume III: Video, Audio, and Instrumentation Recording*. McGraw-Hill, New York, 1988.

[68] W. H. Meiklejohn. Magnetooptics: A Thermomagnetic Recording Technology. In *Proceedings of the IEEE*, pages 581–592, November 1986.

[69] Metrum Information Storage, Denver, Colorado. RSS-600 Rotary Storage System product literature.

[70] Ethan L. Miller and Randy H. Katz. Input/output behavior of supercomputing applications. In *Proceedings of Supercomputing '91*, pages 567–576, November 1991.

[71] Ethan L. Miller and Randy H. Katz. An analysis of file migration in a Unix supercomputing environment. In *USENIX—Winter 1993*, January 1993.

[72] Stephen W. Miller. MSS requirements for data acquisition systems. In *Digest of Papers*. Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[73] Fred Moore. Long-range Storage Perspectives. Technical report, Storage Technology Corporation, 2270 South 88th Street, Louisville, Colorado, 80028-4315, 1991.

[74] Eugene C. Nagel. Digital Storage Technology: DAT Evolves. *SunTech Journal*, September/October 1990.

[75] Marc Nelson, David L. Kitts, John H. Merrill, and Gene Harano. The NCAR mass storage system. In *Digest of Papers*. Eighth IEEE Symposium on Mass Storage Systems, May 1987.

[76] Mike Nelson. SGI, October 1993. Personal communication.

[77] Staff of Digital Review Labs. DAT Takes Backup to the Future. *Digital Review*, June 18, 1990.

[78] Tom Parish. Crystal Clear Storage. *Byte*, pages 283–288, November 1990.

[79] David A. Patterson. Trends in magnetic disk technologies., August 1994. Personal communication.

[80] David A. Patterson, Garth Gibson, and Randy H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings ACM SIGMOD*, pages 109–116, June 1988.

[81] David A. Patterson and John. L. Hennessy. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.

[82] Klaus J. Peter and Dennis E. Speliotis. Recording and Wear Characteristics of 4 and 8 MM Helical Scan Tapes. Media Logic Inc., 310 South St., Plainville, MA 02762, Sept. 28 1992.

[83] Andy Poggio. CD-ROM Technical Summary: From Plastic Pits to 'Fantasia'. Available by ftp from ftp.apple.com, March 1988.

[84] Pyramid Technology Corporation, Mountain View, CA. FastTRAC product literatrue, 1992.

[85] P. Venkat Rangan and Harrick M. Vin. Designing File systems for Digital Video and Audio. In *Proceedings of the 13th ACM Symposium on Operating System Principles*, 1991.

[86] P. Venkat Rangan, Harrick M. Vin, and Srinivas Ramanathan. Designing an On-Demand Multimedia Service. *IEEE Communications Magazine*, 30(7):56–64, July 1992.

[87] Rick Rashid. Microsoft's tiger media server. Workshop Record, First Network of Workstations Workshop, San Jose, California, October 4, 1994.

[88] Steve Redfield and Jerry Willenbring. Holostore Technology for Higher Levels of Memory Hierarchy. In *Digest of Papers*. Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[89] B. Ross and J. Richards. Volume management by the book: The NASstore volume manager. In *Digest of Papers*. Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[90] Andrew Ruddick and Joe Duffy. ICI's Optical Tape Offers Flexible Alternative to Rigid Optical Media. *Optical Memory News*, February 1991.

[91] Andrew J. Ruddick. ICI Optical Data Stroage Tape – An Archival Mass Storage Media. In *Goddard Conference on Mass Storage Systems and Technologies*. NASA/Goddard Space Flight Center, September 1992.

[92] K. Salem and H. Garcia-Molina. Disk striping. In *Proceedings IEEE Data Engineering*, pages 336–342, February 1986.

[93] Gerry Schadegg. Optimizing Digital 8mm Drive Performance. In *Goddard Conference on Mass Storage Systems and Technologies*. NASA/Goddard Space Flight Center, September 1992.

[94] Martin Schulze, Garth Gibson, Randy H. Katz, and David A. Patterson. How reliable is a RAID? In *Proceedings IEEE COMPCON*, pages 118–123, Spring 1989.

[95] Margo I. Seltzer, Peter M. Chen, and John K. Ousterhout. Disk Scheduling Revisited. In *Proceedings of the Winter 1990 USENIX Technical Conference*, January 1990.

[96] Michael P. Sharrock. Particulate Magnetic Recording Media: A Review. *IEEE Transactions on Magnetics*, 25(6), November 1989.

[97] Paul H. Siegel. Recording Codes for Digital Magnetic Storage. *IEEE Transactions on Magnetics*, 21(5), September 1985.

[98] Glenn T. Sincerbox and James M. Zavislan, editors. *Selected Papers on Optical Storage*. SPIE–The International Society for Optical Engineering, Bellingham, Washington, 1992.

[99] W. D. Sincoskie. System Architecture for a Large Scale Video On Demand Service. *Computer Networks and ISDN Systems*, 22:155–162, 1991.

[100] Steven F. Small. 8mm and DAT Formats Compete for Tape Technology of Choice. *Computer Technology Review*, Spring 1990.

[101] Alan Jay Smith. Analysis of Long Term File Refernce Patterns for Application to File Migration Algorithms. *IEEE Transactions on Software Engineering*, 7(4):403–417, July 1981.

[102] Alan Jay Smith. Long Term File Migration: Development and Evaluation of Algorithms. *Communications of the ACM*, 24(8):521–532, August 1981.

[103] Ken Spencer. Terabyte Optical Tape Recorder. In *Digest of Papers*. Ninth IEEE Symposium on Mass Storage Systems, October 1988.

[104] Ken Spencer. The 60-second Terabyte. *Canadian Research*, June 1992.

[105] Donald J. Stavely, Mark E. Wanger, and Kraig A. Proehl. A Rewritable Optical Disk Library System for Direct Access Secondary Storage. *Hewlett-Packard Journal*, December 1990.

[106] Andrew J. G. Strandjord, Steven P. Webb, Donald J. Perettie, and Robert A. Cipriano. Flexible Storage Medium for Write-Once Optical Tape. In *Goddard Conference on Mass Storage Systems and Technologies*. NASA/Goddard Space Flight Center, September 1992.

[107] Hiroshi Sugaya. Recent Advances in Video Tape Recording. *IEEE Transactions on Magnetics*, 14(5), September 1978.

[108] Eng Tan and Bert Vermeulen. Digital Audio Tape for Data Storage. *IEEE Spectrum*, October 1989.

[109] William Tetzlaff, Martin Kienzle, and Dinkar Sitaram. A Methodology for Evaluating Storage Systems in Distributed and Hierarchical Video Servers. In *Proceedings IEEE COMPCON*, February 1994.

[110] Erich Thanhardt and Gene Harano. File migration in the NCAR mass storage system. In *Digest of Papers*. Ninth IEEE Symposium on Mass Storage Systems, October 1988.

[111] Fouad A. Tobagi, Joseph Pang, Randall Baird, and Mark Gang. Streaming RAID(TM) – A Disk Array management System for Video Files. In *Proceedings of ACM Multimedia 93*, August 1993.

[112] Michael Jay Tucker. Optical Disks: The Fabulous Closet of Fibber McGee. *SunExpert*, pages 52–62, June 1992.

[113] David Tweten. Hiding Mass Storage Under UNIX: NASA's MSS-II Architecture. In *Digest of Papers*. Tenth IEEE Symposium on Mass Storage Systems, May 1990.

[114] David Tweten and Alan Poston. Distributed NAStore as the next step. In *Digest of Papers.* Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[115] Bert Vermeulen. Helical Scan and DAT–a Revolution in Computer Tape Technology. In *Systems Design and Networks Conference (SDNC)*, pages 79–86, May 1989.

[116] Dick Wilmot. Editor, Independent RAID Report, November 1994. Personal communication.

[117] Dave Withers. Mead Data Central, September 1993. Personal communication.

[118] Roger W. Wood. Magnetic Recording Systems. *Proceedings of the IEEE*, 74(11), November 1986.

[119] Tracy Wood. D-1 Through DAT. In *Digest of Papers.* Ninth IEEE Symposium on Mass Storage Systems, October 1988.

[120] Ichiro Yamada, Minoru Saito, Akinori Watanabe, and Kiyoshi Itao. Automated Optical Mass Storage Systems with 3-Beam Magneto-Optical Disk Drives. In *Digest of Papers.* Eleventh IEEE Symposium on Mass Storage Systems, October 1991.

[121] Stephen Kreider Yoder and G. Pascal Zachary. Vague New World; Digital Media Business Takes Form as Battle of Complex Alliances; Partnerships Across Industries Coalesce in Chaotic Race to Establish a Market; A Pattern Akin to 'Keiretsu'. *Wall Street Journal*, July 14, 1993.