Choosing the Best Storage System for Video Service

Ann L. Chervenak
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
annc@cc.gatech.edu

David A. Patterson
Randy H. Katz
Computer Science Division
University of California
Berkeley, California 94720
patterson@cs.berkeley.edu
randy@cs.berkeley.edu

ABSTRACT

In this paper, we evaluate three storage systems for movies-on-demand video servers. We show that striped disk farms, in which portions of movies are interleaved among disks, perform best. They achieve close to full disk utilization, good load balancing, and the lowest cost per video stream. Disk farms that store one movie per disk have lower performance, since disks holding less popular movies are under-utilized; however, these systems achieve relatively good performance by replicating movies according to their popularity in the request pattern. Finally, we evaluate the use of storage hierarchies to support movies-on-demand video service. We show that their performance is so poor that their effective cost per video stream is two orders of magnitude higher than that of magnetic disk systems. We conclude that for the foreseeable future, most video servers will store movies entirely on disk.

INTRODUCTION

This paper explores storage system alternatives for movies-on-demand video servers. We examine both magnetic disk farms and storage hierarchies composed of disk farms plus tape libraries. We can use a variety of metrics to evaluate storage architectures. The video server customer detects system availability and response time as well as video stream quality. For example, if the video server misses playback deadlines for a number of video packets, the user may detect diminished presentation quality. The service provider is concerned with the total number of streams a video server can support, the average cost per stream, the maximum response time to the customer and the cost of storing video. The storage system designer has additional concerns such as load balancing and resource utilization. We assert that the most important storage system metric is the average cost per video stream. This metric directly contributes both the cost to the customer and the profit to the service provider. (Other contributions include the cost of content and of network delivery.) We show that the cost per stream is more important than the cost of storing video. Other important requirements, such as response time and presentation quality, are conservatively satisfied in our simulations.

In this paper, we focus only on the storage system component of the video server, and assume the existence of processing and network hardware to deliver the required bandwidth to the home; likely network technologies include ASDL-2, hybrid fiber coax and fiber-to-the-home [2], [12]. We evaluate three storage systems for video service. We find that striped disk farms, in which moves are interleaved among disks, provide the lowest cost per video stream. They do so by achieving high disk utilization and good load balancing. Disk systems that store one movie per disk perform worse than striped systems. Although the one-movie-per-disk scheme is simple, it wastes substantial disk bandwidth, since disks holding less popular movies are under-utilized. The scheme achieves fairly good performance if movies are replicated to reflect the user request pattern. Finally, we evaluate the use of storage hierarchies that include a tape library along with a disk farm. At first glance, storage hierarchies seem attractive because they offer much lower cost per megabyte of storage than do magnetic disks. Unfortunately, we show that the performance of magnetic tape libraries is inadequate to service the predicted distribution of movie accesses. The average cost per video stream for a tape system is two orders of magnitude higher than for disk farms. We conclude that for the foreseeable future, most video servers will store movies entirely on disk.

CHARACTERIZING THE WORKLOAD

Before we can discuss storage system design for a video server, we must characterize the workload the server will support. Unfortunately, there is little reliable information available on the workload for this application. Many companies are building prototypes to gather such data but are reluctant to share the information because of the fiercely competitive nature of the video-on-demand industry. Also, it is expected that access patterns observed in prototype systems will be dramatically affected by prices and marketing practices. Thus, it is very difficult to predict realistic access patterns. In the absence of workload measurements and reliable predictions, storage systems designers need to make a "best-guess" estimate of the workload.

Such a workload characterization is inevitably controversial. To minimize this controversy, we restrict ourselves to the relatively straightforward movies-on-demand application. In earlier work, we presented evidence to support our workload characterization [7]. This included predicted sizes for video objects and the throughput required per video stream for typical compression schemes; an estimate of locality of requests based on video store rental patterns; and typical user population sizes and response time constraints based on a survey of existing and proposed video servers.

Table 1 summarizes the workload that will be used in the simulation studies in this paper. We assume an MPEG-II compression scheme with a data rate of 3 megabits per second per video stream. Based on this data rate, object sizes range from 675 megabytes for a 30-minute television program to over 2 gigabytes for a 100-minute movie. In our simulations, we assume all objects are 100-minute movies.

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 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.

Further, we assume that all objects are requested in their entirety and played sequentially. In reality, some percentage of the time users will deviate from sequential viewing to perform VCR-like operations such as pause, rewind and fast-forward. There is little data available on the frequency of these operations. For simplicity, we ignore them in this study.

Based on video store rental patterns, we assume that accesses to movies in a video server will be highly localized, with a small number of movies receiving most of the accesses. We use Zipf's Law [27] to characterize this locality. Zipf's Law states that the probability of choosing the nth most popular of M movies is

$$C/n,$$

where

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

For example, using this distribution, the 5th most popular movie is requested one fifth as often as the most popular movie. We have shown [7] that one example of movie rental history approximately matches a Zipf's Law distribution. This highly-localized distribution is a plausible choice to characterize accesses to a video server in the absence of much 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.

Locality influences system design by suggesting how effective caching strategies can be in the storage server. A high degree of locality would suggest that caching or hierarchical storage layouts will be effective for video servers. This locality may also make it economically rewarding for companies to supply only a small number (perhaps 50 to 200) of the most popular movies, servicing the head of the request distribution and ignoring the tail.

Our study focuses specifically on storage systems that could economically support a larger number of video objects. In all our simulations, we assume that each video server stores up to 1000 unique movie titles.

Typical video-on-demand systems will have fairly strict response time constraints. When possible in our simulations, we demand that the storage system hardware begin delivery of video within 60 seconds of a user request. The server must also maintain strict performance constraints to guarantee adequate presentation quality; the compression standard demands delivery of a frame as often as every 1/30th of a second. In reality, occasionally these "deadlines" for frame delivery can be missed without noticeable degradation of video stream quality. Our simulations are conservative, however, and assume a striping scheme that misses no deadlines unless disks fail.

Finally, the peak number of active users in the system is determined by the capabilities of the hardware. This is the main result of our simulations.

STORAGE SYSTEM ALTERNATIVES

There are two practical possibilities for building a storage system for video service, illustrated in Figure 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 [19]) 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 to the standard I/O interfaces of hosts, usually without maintaining parity information and often without data striping. In this paper, we focus only on disk farms, whose pricing corresponds to the cost of commodity disks.

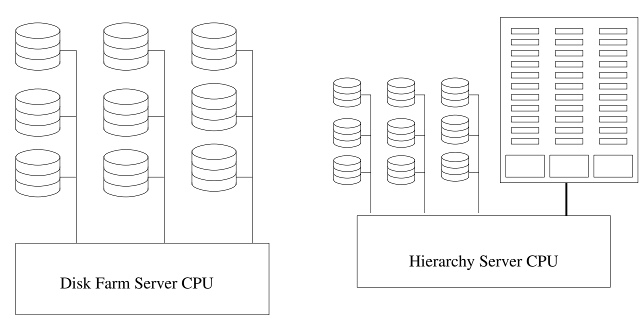

Figure 1 Shows the possibilities for building a video server: entirely disk-based, or a storage hierarchy of disk and tertiary storage.

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.

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, or interleaves, data blocks of movies across a collection of disks.

One Movie Per Disk

Figure 2 shows a 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 each disk.

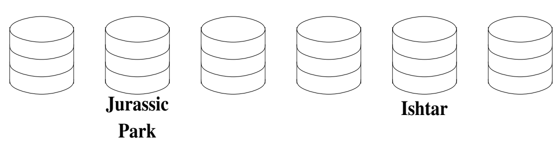

Figure 2 Shows disk-based video server configuration with one movie per disk.

One problem that may occur with a one-movie-per-disk scheme arises if some movies are more popular than others. Such popular movies will create "hot spots" or bottlenecks in the disk system. For example, in Figure 2 , 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 1995 is 2 to 3 megabytes per second. If each video stream consumes 3 megabits or 0.375 megabytes per second, then a single disk can support approximately five independent viewers watching the movie concurrently. (This calculation assumes disk bandwidth of 2.5 megabytes per second and an average seek and rotation operation lasting about 30 milliseconds for each viewer.) 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 reschedule 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 this paper, we present simulation results for several variations of the one-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.

Striping Movies Among Disks

Figure 3 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.

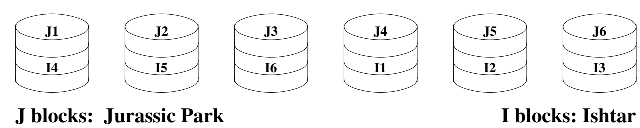

Figure 3 Shows disk-based video server configuration with movies striped across several disks.

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 its transfer capabilities to showing whichever movies have been requested. Because movies are spread over a collection of disks, the system has better load balancing. In this paper, we assume that movie intervals are distributed across the disk farm in a round-robin fashion. Several commercial video server products and prototypes also stripe data, including systems from IBM Almaden [10], Microsoft [21] and Starlight Networks [25]. Unlike our simple round-robin layout of successive movie increments among the disks, some of these commercial systems use a hashing function to distribute movie blocks; this scheme is thought to provide better disk load balancing and is designed to have a low probability of missing playback deadlines for video frames.

The main challenge for striped layout of movies in a video-on-demand server is somewhat more complex scheduling than 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 a 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 below. Once movie playback has begun, the system must guarantee that there will rarely or never be 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, as in our simulations.

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 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. 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. Another consideration is that disk array performance is seriously degraded during reconstruction operations.

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

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.

Disk Array Simulations

The first set of disk simulations is for a disk server that places one movie on each disk. For these simulations (and for the tape simulations discussed at the end of this paper, we use a simple event-driven simulator. 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. More details on this simulator are given in the discussion on tape array simulation.

We also simulate a striped disk farm in which movies are interleaved across collections of disks. Controlling a striped disk video server is largely a scheduling problem. To simulate the striped disk farm, we maintain 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 a convenient number, since it is the minimum length of time over which MPEG-II compression can be performed. Earlier studies showed that striping on a very fine grain is not advantageous, since it involves several disks in even the smallest video block retrieval [11]. Conversely, striping in larger blocks increases the buffer space requirements in the video server [25]. Based on these tradeoffs, we chose the relatively small interleave factor of a half-second of video time. A more exhaustive study of optimal interleave factors would be desirable.

One-Movie-Per-Disk Performance

Figure 4 shows the performance of a video server disk farm that places one movie on each disk. The graph shows 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 on each line from left to right show the number of active streams supported when the total number of movies in the system is 10, 100 and 1000, respectively. Assuming 1995 costs of 50 cents per megabyte for disks, the total disk system cost including replicates is shown on the horizontal axis of the graph. The vertical axis shows the number of active (simultaneous) streams that can be supported while maintaining the constraint that 95% of all requests must begin video playback within a specified response time limit of 60 seconds. 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 video frames are met.

Requests to the video server are highly localized according to the Zipf's Law distribution. The graph confirms the intuitive conclusion that replicating movies to satisfy this highly-localized access pattern provides the least expensive cost per video stream.

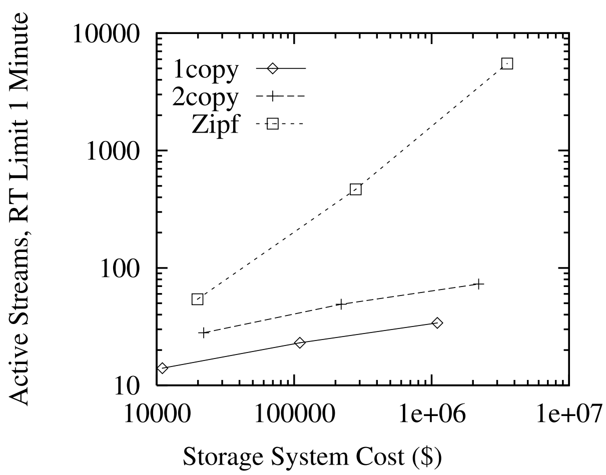

Figure 4 One-movie-per-disk Performance, Response time limit 60 seconds: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must complete within 60 seconds. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.

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. 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 4 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 performance.

Finally, the highest line in Figure 4 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 2 summarizes the results for the three different configurations. For the most important cost/performance metric for video servers, *cost per stream*, Table 2 shows that a replication pattern reflecting the Zipf's Law distribution of requests 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.

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

Table 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

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 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.

Figure 5 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 movie, two copies, and movies replicated according to the Zipf's Law distribution, respectively. The response time limit for the simulation is 60 seconds.

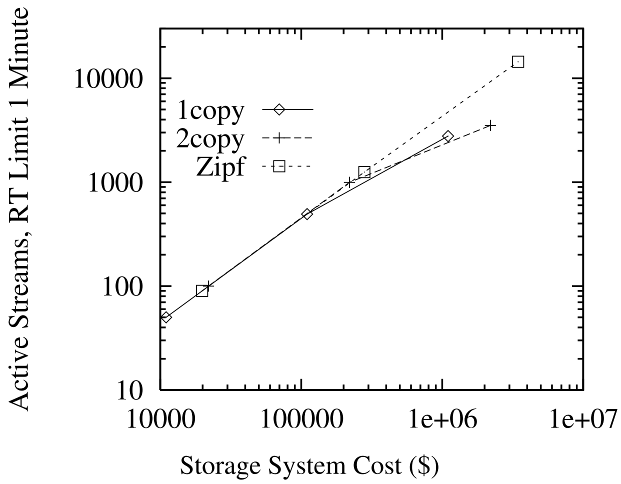

Figure 5 Striped disk performance, Response time limit 60 seconds: Shows number of streams for each layout scheme (1-copy, 2-copy and Zipf) that can be sustained when 95% of the requests must begin video playback within 60 seconds. The three data points on each line represent systems with 10, 100 and 1000 total movies, respectively.

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 time and achieve high disk utilization. The middle line in Figure 5 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 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.

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 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.

Compared to the one-movie-per-disk scheme, the striped video server supports many more streams, as shown in Table 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. Substantial disk bandwidth is wasted in the one-movie-per-disk scheme because accesses to the least popular movies are unlikely to fully utilize the disks on which the movies reside. The striped video server offers higher disk utilization and lower cost per stream. The latter is shown in Table 5.

	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 4 Comparison of number of streams supported by one-movie-per-disk and striped video servers for systems with 10, 100 and 1000 total movies. Ratio is (striped streams to 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 5 Comparison of cost per stream for one-movie-per-disk and striped video servers for systems with 10, 100 and 1000 total movies. Ratio is (cost of striped stream to cost of one-per-disk stream).

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, 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 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 show that more and faster tape drives would improve tape library performance in a video server storage hierarchy.

Storage Hierarchy Simulations

For the simulation results presented in this section, we use an event-based, closed simulator based on measurements of tape drives and tape robots [7]. The simulator maintains a queue of relevant events and processes them in time order. Events include an arrival of a new request, after which the request is mapped onto physical devices; events that indicate when a robot arm or a tape drive goes free; and an event to indicate when a rewind and eject operation completes. There are several input files used to run simulations. 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 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 and robots. The results generated by a simulation run include the mean and standard deviation of the response time and bandwidth.

The user may set a response time constraint that requires that 95% of all requests finish within the specified response time. The simulator then determines the maximum load it can sustain and still meet the response time limit. 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. Simulator results are averaged over at least as many I/O operations as specified by a minimum records parameter. Each simulation run continues until the variance of the response time on the operations has stabilized to within a 95% confidence interval.

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 serviced by the disk farm and the proportion that will “fall through” to be serviced by the tertiary storage portion of the hierarchy. Table 6 summarizes the simulation parameters used in this section, and Table 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 to help evaluate the effectiveness of particular tertiary storage libraries for video service.

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

Table 6 Simulation parameters for the example used in the remainder of this paper.

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 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.

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. We also do not simulate tape striping; previous work [7] [8] demonstrated that striping is not advantageous for this workload.

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.

Performance of the Exabyte EXB120 Library

Figure 6 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 tape libraries have very long access times [7].) 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.

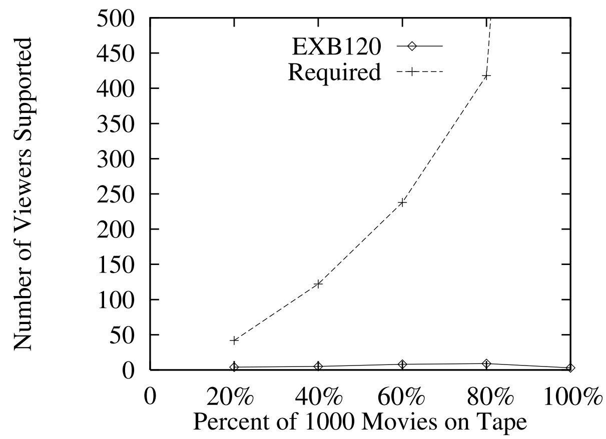

Figure 6 Figure shows performance of EXB120 tape libraries with four drives. Number of streams supported is low compared to the requirements of the application when 1000 movies and 2000 total streams are processed by the server.

The top line in Figure 6 shows the values from the rightmost column of Table 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 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 low performance is the result of contention for the few tape drives in the robot. One tape drive has only enough bandwidth (about 0.5 megabytes per second) to service a single concurrent stream. The performance of the EXB120 tape library suggests a grave imbalance between the number of movies that can be stored (200) and the number of streams that can be serviced (4). An EXB120 library costs approximately $100,000, yielding a per-stream cost for the library of $25,000, two orders of magnitude higher than for magnetic disks.

To simulate a library with better performance and less contention for tape drives, we show in Figure 7 the performance of an EXB120 robot that contains twenty tape drives (instead of four) and in which the tape drives and robot operate ten times as fast as the current generation of hardware. 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 "improved" EXB120 now supports 96 video streams within the response time limit of one hour. Assuming that tape drives cost $3000, this would reduce the per-stream cost to $1541, which is still considerably higher than the cost per stream for 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.

The precipitous drop in performance for the simulations 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.

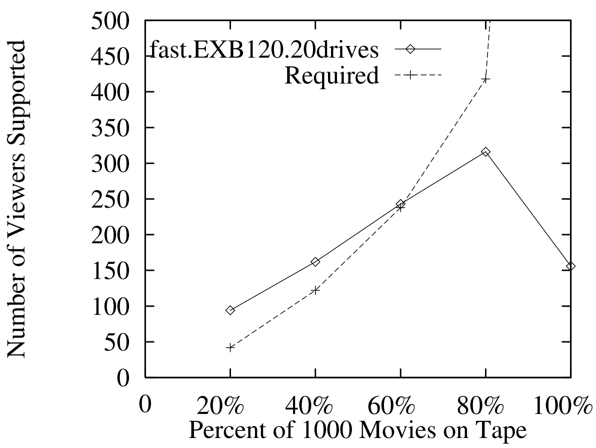

Figure 7 Figure shows the performance when the tape library contains twenty tape drives and drives and robot are faster by a factor of ten. The number of concurrent streams exceeds the requirements for up to 60% of the movies stored in the tape library.

High Performance Tape Library

In this section, we show simulation results for a high performance library that contains four tape drives that each transfer at 15 megabytes per second. Figure 8 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 approximately $1,000,000 with four tape drives.

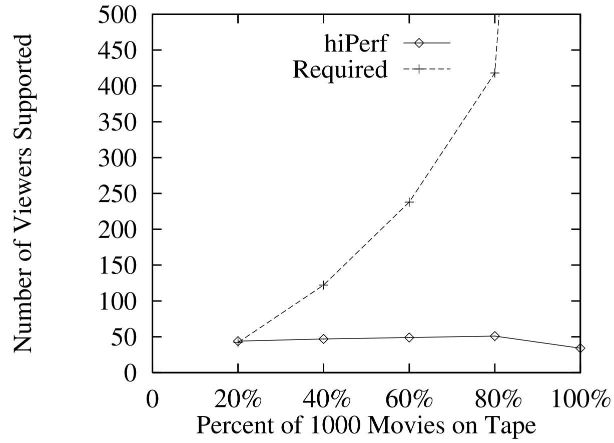

Figure 8 Shows the performance of the high performance library compared to the required performance for the application. Although the high performance library has drives that deliver 15 megabytes per second, the total system bandwidth is not adequate to service all the movies that can be stored in the array.

Figure 8 shows that a single library supports up to 51 concurrent streams, resulting in a per-stream cost of approximately $20,000. Like the EXB120 tape library, this higher-performance library experiences contention for a small number of tape drives. 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.

Summary of Hierarchical Video Server Costs

From a marketing perspective, video service providers would like to claim the availability of a vast number of video titles, even if the chance of the more obscure titles being accessed is very small. In this section we briefly characterize how rare accesses to the tape library must be for a hierarchical storage scheme to be affordable.

In all the simulation results presented in this paper, we assumed a standard Zipf distribution of accesses. In [7], we examined more highly-localized variations of the Zipf distribution. For such distributions, the disk level of the storage hierarchy would capture a larger proportion of the requests, and the resulting demands on the tertiary storage system would be reduced. Under such circumstances, current tape library products may be able to supply the required number of simultaneous streams at an acceptable overall cost per stream. However, accesses that fall through to the tape system will still have to tolerate long response times.

We have shown that the per-stream cost of supplying video from both low-cost and high-performance tape robots is currently in the range of $20,000. By contrast, stream costs on magnetic disks are much lower, approximately $200. Suppose 99% of all requests are serviced by the disk system and 1% fall through to the tape system. The resulting storage cost per stream is $398, double the cost of a system composed entirely of disks. This is likely to be unacceptably expensive. By comparison, if 99.9% of all requests are serviced by the disk system, then the cost per stream drops to approximately $220, or just 10% more per stream than a system composed entirely of disk. This cost is more likely to be acceptable.

Other Storage Hierarchies

In previous work, we have demonstrated that storage hierarchies that use optical disk jukeboxes for the lower level of the hierarchy experience contention for disk drives similar to the contention for tape drives seen in magnetic tape libraries [7], [6]. As with tape libraries, this contention is caused by too few optical disk drives. These unbalanced systems have long average response times and high cost per video stream.

Another hierarchy option is the use of RAM for storing the most popular movies, which would have the highest cost of storing video but also the highest bandwidth. Finally, we have examined only two-level storage hierarchies, but it may be appropriate to include more levels in a storage hierarchy. For example, a system may include two levels of magnetic disk, with more expensive, higher bandwidth disks at a higher level and slower, less expensive commodity disks at a lower level.

SUMMARY

We have compared the performance of three storage systems to support video 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 create hot spots that limit overall 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 achieve greater disk utilization and load balancing and support a larger 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 compared to magnetic disks. Unfortunately, we showed that the performance of the current generation of tape libraries is so poor that they can sustain very few concurrent streams. As a result, the cost per video stream for tertiary storage is substantially higher than that of disk systems. Because of the high cost per stream and long response times of the tertiary storage portion of the storage hierarchies, we predict that for the foreseeable future, most commercial video servers will be composed entirely of magnetic disks.

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. For striped disk farms, another issue is the best interleave factor for striping. Finally, we discussed the advantages of partitioning the striped disk farm to bound wait times for a free scheduling slot and to limit the vulnerability of streams to disk failures; a study of optimum partitioning of the disk farm is needed.

REFERENCES

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

[2] G. Keith Cambron and Fred Kemmerer. "Bandwidth Management in a Hybrid Fiber Coax Network." In *Proceedings of the 1995 2nd International Workshop on Community Networking*, June 20-22 1995.

[3] Ed Chang and Avideh Zakhor. "Scalable Video Data Placement on Parallel Disk Arrays." *IS&T/SPIE Symposium on Elec. Imaging Science and Technology*, 2185: Image and Video Databases II, February 1994.

[4] 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.

[5] Peter M. Chen, Edward K. Lee, Garth A. Gibson, Randy H. Katz, and David A. Patterson. "RAID: High-Performance, Reliable, Secondary Storage." *ACM Computing Surveys*, 26(2), June 1994.

[6] Ann L. Chervenak, David A. Patterson, and Randy H. Katz. "Storage Systems for Movies-on-Demand Video Servers." In *Digest of Papers*. Fourteenth IEEE Symposium on Mass Storage Systems, November 1995.

[7] Ann Louise Chervenak. *Tertiary Storage: An Evaluation of New Applications*. PhD thesis, U. C. Berkeley, December 1994. University of California at Berkeley Technical Report UDB/CSD 94/847, December, 1994.

[8] Ann L. Drapeau and Randy H. Katz. "Striping in large tape libraries." *Supercomputing '93*, November 1993.

[9] 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.

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

[11] 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.

[12] Tim Kwok. “A Vision for Residential Broadband Services - ATM-to-the-Home.” In *Proceedings of the 1995 2nd International Workshop on Community Networking*, June 20-22 1995.

[13] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, F.W. Reeve, D.H. Schelleng, and D. Venkatesh. “A Digital On-Demand Video Service Supporting Content-Based Queries.” In *Proceedings of ACM Multimedia 93*, August 1993.

[14] P. Lougher and D. Shepherd. “Design and Implementation of a Continuous Media Storage Server.” In *Proceedings of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video*, 1992.

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

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

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

[18] Eitetsu Oomoto and Katsumi Tanaka. “OVID: Design and Implementation of a Video-Object Database System.” *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629–643, August 1993.

[19] 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.

[20] 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.

[21] Rick Rashid. “Microsoft’s Tiger Media Server.” Workshop Record, First Network of Workstations Workshop, San Jose, California, October 4, 1994.

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

[23] Ralf Steinmetz. “Analyzing the Multimedia Operating System.” *IEEE Multimedia*, Spring 1995.

[24] 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.

[25] 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.

[26] Harrick M. Vin and P. Venkat Rangan. “Designing a Multiuser HDTV Storage Server.” *IEEE Journal on Selected Areas in Communications*, 11(1):153–164, January 1993.

[27] G.K. Zipf. *Human Behavior and Principle of Least Effort: an Introduction to Human Ecology*. Addison Wesley, Cambridge, Massachusetts, 1949.