Benchmarking Tape System Performance

Theodore Johnson
AT&T Labs - Research
180 Park Ave., Bldg. 103
Florham Park, NJ 07932
+1 973 360-8779, fax +1 973 360-8050
johnsont@research.att.com

Ethan L. Miller
Computer Science & Electrical Engineering Department
University of Maryland Baltimore County
1000 Hilltop Drive
Baltimore, MD 21250
+1 410 455-3972, fax +1 410 455-3969
elm@acm.org

Abstract

In spite of the rapid decrease in magnetic disk prices, tertiary storage (i.e., removable media in a robotic storage library) is becoming increasingly popular. The fact that so much data can be stored encourages applications that use ever more massive data sets. Application drivers include multimedia databases, data warehouses, scientific databases, data-intensive scientific research, and digital libraries and archives. The research community, has responded with investigations into systems integration, performance modeling, and performance optimization.

Tertiary storage systems present special challenges because of their unusual performance characteristics. Access latencies can range into minutes even on unloaded systems, but transfer rates can be very high. Tertiary storage is implemented with a wide array of technologies, each with its own performance quirks. However, little detailed performance information about tertiary storage devices has been published. As a result, mass storage system (MSS) implementers must rely on vendor-reported numbers or their own tests to select appropriate tertiary storage devices. Additionally, MSS designers must have detailed knowledge of the performance characteristics of their devices to optimally place files on media and perform other optimizations.

In this paper we present detailed measurements of several tape drives and describe the tests used to gather this data. The tape drives we measured include the DLT 4000, Ampex 310, IBM 3590, 4mm DAT, and the Sony DTF drive. This mixture of equipment includes high and low performance drives, serpentine and helical scan drives, and cartridge and cassette tapes. This data is suitable for system performance modeling or system performance optimization studies. By measuring and modeling a variety of devices in a single study, we are able to characterize a wide range of tertiary storage devices. In addition, we hope that our simple benchmarks will become more widely used to gauge tape performance and identify potential performance bottlenecks.

This paper appeared in the 6th Goddard Conference on Mass Storage Systems and Technologies, held jointly with the 15th IEEE Mass Storage Systems Symposium, College Park, MD, March 1998

1. Introduction

A tertiary storage system typically refers to a data storage system that uses drives that accept removable media, a storage rack for the removable media, and a robot arm to transfer media between the storage rack and the drives. The media can be disks (usually optical disks) or tapes, though in this paper we concentrate on tape-based tertiary storage. Tertiary storage is used for massive data storage because the amortized per-byte storage cost is usually two orders of magnitude less than on-line storage (e.g., see [1]). Tertiary storage has other benefits, including the removability of the media and fewer moving parts. However, access time to a file stored on tertiary storage can range into the minutes.

In this paper, we measure a variety of devices used in tertiary storage systems and present performance characterizations of these devices. The contribution of this work is the scope and detail of our measurements and the benchmarks used to generate them. We measure many aspects of tape access, including mount time, seek time, transfer rates, rewind time, and unload time. The devices we measure include high, medium, and low performance tape drives. We include measurements that have not previously been published (to our knowledge) but are vital to efficient tertiary storage system implementations, such as short seek times. This information can be used to guide the design of better tertiary storage systems by showing strengths and weaknesses of tape technologies as well as those of specific tape systems.

2. Background

Tertiary storage is often viewed as a necessary evil by file system designers and users — the massive storage capacity of a robotic storage library incorporating high-volume tape drives is needed to store massive data sets, but management of and access to tape-resident data can be painful. Tertiary storage is often used in "write-once, read-never" applications such as storage of large data sets (in which data is rarely reused), backup, and archiving. Today, however, it is becoming more common to use tertiary storage to store active data that is still useful, but is not used sufficiently frequently to warrant the cost of purchasing additional secondary storage. Fortunately, much work has been done on hierarchical storage management systems (HSMs) to simplify access to tertiary storage data.

2.1. Hierarchical Storage Systems

Hierarchical storage systems extend secondary storage (disk-based) file systems by adding tertiary storage as an "overflow area." In a typical implementation, HSMs use secondary storage as a cache for the set of files that reside on tertiary storage. If an application (including a shell tool) opens a file that is not in the cache, the file is brought in from tertiary storage. The user only notices a delay in opening the file. HSMs that have been studied previously include Unitree, HPSS [2], AMASS, and ADSM [3]; there are several other systems that are in production use. An alternative approach is to build a log-structured file system on top of tertiary storage, with secondary storage being treated as a cache for log segments [4,5].

Computer data analysis is transforming scientific research, and it is also forcing the creation of systems that can store terabytes or even petabytes of data. The very large data sets

that can be collected have created enormous data storage and retrieval problems. For example NASA's EOSDIS [6], which supports research into climate change, will collect and archive on the order of ten petabytes of data. Many other scientific projects, such as high-energy physics [7,8] also have very large data storage requirements. More recently, though, newer applications such as data warehouses [9], scientific databases [6,10], multimedia [11], and digital libraries [12] are driving the creation of very large databases that integrate tertiary storage into a database system.

2.2. Tertiary Storage Modeling and Optimization

The building of very large scale scientific archives and the efforts at integrating tertiary storage into database systems have motivated considerable recent research activity. In this section we summarize tertiary storage modeling and optimization work. The planning and integration problems of building large tertiary storage installations have motivated recent work in the performance modeling of tertiary storage systems. Pentakalos et. al. [15] present an analytical model of a scientific computing system that incorporates tertiary storage, and Johnson [14] presents a detailed queuing model of a robotic storage library. Menasce, Pentakalos, and Yesha [15] give an analytical model of tertiary storage as a network attached storage device, and Nemoto, Kitsuregawa, and Takagi [16] make a simulation study of data migration.

The above cited research all shares the characteristic of depending on a model of the behavior of tertiary storage devices (robot arms, tape drives, etc.) to either model or to optimize performance. However, the performance of tertiary storage devices is not well understood. As a result, the otherwise high-quality work discussed above use limited, inaccurate, or incorrect models of tertiary storage devices. Fortunately these deficiencies can be remedied by the use of accurate device models that rely on performance measurements gathered from a variety of tertiary storage devices.

While some work has been done to measure, model, and classify the performance of tertiary storage devices, broad-based, comprehensive studies have not appeared, though comprehensive models of secondary storage (i.e, disk drives) have been published [17]. Many works on systems incorporating tertiary storage include benchmarking studies [18,19,20], while other studies have focused on an aspect of a particular device. Ford and Christodoulakis [21] model optical continuous linear velocity disks to determine optimal data placement. Hillyer and Silberschatz [22] give a detailed model of seek times in a DLT 4000 tape drive, to support a tape seek algorithm [23], and van Meter [24] is researching appropriate delay estimation models for tertiary storage. The Mass Storage Testing Laboratory (MSTL) is developing benchmarks for HSM systems [25]; however, these benchmarks test the performance of a software and hardware system, while our benchmarks are only concerned with hardware performance. Additional performance measurement studies include [26] and [27].

3. Taxonomy

The technology used to implement a tape drive influences the performance that the user will obtain from the drive. In this section, we discuss the technologies used to build common tape drives. Table 1 shows the list of tape drive features relevant to the material in this

paper. For a deeper discussion of these matters, we refer the reader to the many papers in

Tape drive attribute	Possible values
Data track layout	Helical scan, linear (serpentine)
Tape package	Cartridge, cassette, cassette with landing zones, “scramble bin” (tape loop)
Directory	None, at beginning or end of tape, calibration tracks, embedded microchip
Data compression	Yes, no
Block size	Fixed, variable
Partitioning	Yes, no, not important

Table 1. Characteristics of tape drives.

other proceedings of the IEEE Mass Storage System Symposium and the NASA Goddard Conference on Mass Storage Systems and Technologies as well as other storage system conferences and journals.

A fundamental characteristic of a tape drive is the layout of data on the tape. To achieve a high density, the tape drive must use as much of the available surface area as possible, and a tape is typically much wider than the data tracks. A helical scan tape writes data tracks diagonally across the tape surface, and packs the diagonal tracks tightly together (e.g., as in a VHS video cassette). A linear tape lays multiple sets of data tracks across the tape. Typically, the data tracks alternate in direction, hence the name “serpentine” (e.g., an audio cassette with autoreverse).

The tape package can be a cartridge (containing 1 reel) or a cassette (containing 2 reels). The tape in a cartridge must be extracted from the cartridge before the tape mount can complete. In addition, the tape cartridge must be rewound before it is unmounted. A cassette can be removed from the tape drive without being rewound. However, the tape in a cartridge must be positioned at a special zone (a “landing zone”) to ensure that data is not exposed to contaminants. If the tape drive does not support landing zones, the cartridge must be rewound.

The geometry of a tape makes defining the position of a particular block more difficult than for disk drives. Modern data storage tapes typically embed some kind of directory to expedite data seeks; this directory is implemented in hardware and is separate from any user-created directory. These directories can be written at the beginning of the tape (or at other special tape positions), in special directory tracks, or in silicon storage devices mounted on the tape package. A precise directory can permit high-speed seeks. In addition, the requirement to read a directory area can increase the mount time, and the requirement to write a directory area can increase the unmount time.

Many tape drives use hardware data compression to increase their capacity and to improve their data rates. However, compressed data is variable sized. Since the location of a block can vary widely, fast seeks can be more difficult to implement. Similarly, a variable size record length increases the flexibility of a tape drive, but can lead to increased seek times.

Some tape drives allow the user to partition the tape into distinct regions. The Ampex tape drive that we tested allows partitioning, while the others do not. Partitioning simplifies some data management functions, and does not have a significant effect on performance. Some serpentine tape drives that support partitioning can improve seek times within a partition; however, we were unable to gain access to such a device for this paper.

Other factors that can affect performance are the tape transport implementation and the use of caching. Helical scan tape drives need to wrap the tape around the read/write head. Performing a high-speed seek requires that the tape be moved away from the head to prevent excessive wear, resulting in a large delay in starting the seek. Linear tapes use a simpler transport and do not suffer from this problem. Because the data rate from the host may not be constant, many tapes use data caches to allow the drive to remain in streaming mode even if the host machine suffers occasional delays in submitting read or write requests. Additionally, this buffer can be used to store read-ahead blocks; many tape drives read a few blocks after the current location even if they are not explicitly requested (yet). Some drives will return this pre-fetched data after short block seeks.

There are many other considerations involved in tape drive technology, especially those of reliability and longevity, that we do not address in this paper. Another important consideration is cost. Some of the drives we measure in this paper can have an order of magnitude better performance than another drive, but they typically cost an order of magnitude more money as well.

4. Benchmark Methodology

Our interest is to measure and develop performance models for the following access characteristics listed below. Taken together, they summarize the end-to-end performance of a tertiary storage device.

- **Mount time**: This is the time from when the robot arm has placed the tape into the drive to the time when the tape is "ready" (i.e., the special file for the drive can be opened and operations performed without incurring I/O errors).
- **Seek time**: This is the time from when a seek command is issued to the time when the seeked-to data block can be read into memory (the seek system call might return before the read operation can be initiated). We measure three particular types of seeks:
  a. **Long seek from beginning of tape**: We measure the time to seek to an arbitrary location in the tape.
  b. **Long seek from the middle of the tape**: We measure the time to seek from one arbitrary location on the tape to another arbitrary location. Since this requires $O(B^2)$ measurements (where B is the number of tape blocks), we pick representative locations in the middle of the tape.
  c. **Short seek from the middle of the tape**: A seek is expensive to initiate on most tapes. The behavior of a seek for a short distance can be very different from that for a long seek.
- **Transfer rate**: This is the rate (Mbytes / second) at which the tape drive will service read or write requests. This rate can be influenced by the compressibility of the data, the record size, and by the time between successive requests to for tape reads (writes).

- **Unmount time**: This is the time from the request to when the tape can be extracted from the drive by the robot arm.

While we tried to make our measurements as consistent as possible from platform to platform, we needed to take special measures for some of the devices. We tested the devices on a wide variety of platforms, each with its own local environment. In all cases the tape drive is attached to a SCSI bus. Also, some devices have special characteristics such as compression, seek location hints, partitioning, etc. Lastly, we had access to some devices for only a limited time.

5. Tape Systems Tested

We tested a wide variety of tape systems ranging from low to high performance (and cost) and with a wide range of characteristics listed in Table 1. In this section, we discuss the basic characteristics of each drive.

5.1. 4mm DAT

The 4mm DAT drive is a low-cost, low-performance drive, in common use for backup and data transfer. It uses helical scan recording, and comes in a cartridge. However, the cartridge does not have landing zones, so the tape must be rewound before unmounting. We measured a data transfer rate of 0.325 Mbytes per second, independent of block size. We were able to measure an average mount time of 50 seconds with a standard deviation of 0.0, and an unmount time of 21 seconds with a standard deviation of 1.3.

5.2. DLT 4000

The DLT 4000 is a moderate-cost, medium performance tape in common use for backup and archiving. The DLT 4000 is a serpentine tape that is packaged in a cartridge. The drive supports automatic data compression; however, all of our experiments were carried out with tape compression turned off. Although the DLT 4000 supports variable block sizes, the SCSI interface limited the maximum writable block size to 64 Kbytes. We found that the transfer rate did not depend strongly on the block size. For block sizes between 16 and 60 Kbytes, the transfer rate is 1.27 Mbytes per second, and the transfer rate declined for larger and smaller blocks.

The software environment that we used to measure the DLT 4000 did not allow us to measure the mount and dismount times directly because the volume management software would load a requested tape, and return when the tape is mounted. If all drives were full, the volume management software would first unload a drive and return the tape to the shelf. To get around this problem, we submitted requests to load tapes when the drives were either all full or all empty. By subtracting the estimate of 9 seconds (from the Storagetek 9710 specifications) to perform a tape fetch using the robot arm, we found that mounting a DLT 4000 tape requires 40 seconds and unmounting a DLT 4000 tape requires 21 seconds. These values are in line with the DLT 4000 performance specifications.

5.3. Ampex DST 310

The Ampex DST310 is a high performance helical scan tape drive that uses a tape cassette. The tape can be formatted with landing zones, eliminating the need to rewind before an unload. The tape also can be formatted with multiple partitions, but they are intended for data management — keeping files together, allowing a partial tape erase, etc. An unusual feature of the Ampex is the availability of "high-speed positioning hints". These hints are returned from the `get_pos` query and can be used in subsequent seek commands. The logical block size is 8 Kbytes, and all data transfers must be made in multiples of 8 Kbytes.

While the recommended transfer size for the Ampex is 4 Mbytes, we tested the read and write transfer rates with a variety of transfer sizes, and found that for transfer sizes of 1 Mbyte and larger, we achieved a throughput of about 14.2 Mbytes/sec. However, the transfer rate declined rapidly for transfer sizes smaller than 1 Mbyte. The average mount time is 10.1 seconds, with a standard deviation of 0.63. The unmount time is more complex. If an unmount is requested without rewinding the tape, the tape is moved to a system zone and then unmounted (this is done to protect the tape), so variance in rewind time becomes variance in unmount time.

5.4. Sony DTF

The Sony DTF is a high-performance tape drive. It uses a helical scan data layout and is packaged in a cassette. The cartridge can be unmounted without rewinding, but only when the current position is near the end of tape. In our system, however, the default command to eject the tape always rewound the tape completely. The Sony DTF supports variable size blocks and compression. We measured an average of 51 seconds to mount a tape, with a standard deviation of 0.0, and an average of 17.8 seconds to unmount a rewound tape, with a standard deviation of 1.2. We achieved a transfer rate of 12 Mbytes/sec with a block size of 512 Kbytes.

5.5. IBM 3590

The IBM 3590 is a high performance tape drive. It uses a serpentine data layout and is packaged in a cartridge. The drive supports variable size blocks and compression. We measured a transfer rate of 8.9 Mbytes/second using a block size of 512 Kbytes and a SGI host; this rate was the same for reads and writes.

6. Benchmark Results

In this section, we discuss the more interesting benchmark results gathered on the tape drives listed in Section 5. While transfer rates were relatively uninteresting (except for their relationships to the values quoted by the manufacturers), both seek times and mount/unmount times provided interesting comparisons. In particular, we focus on both long and short seeks and show that, often, reading can be faster than seeking to a nearby position.

6.1. Mount and Unmount

As the discussion in Section 5 indicates, mount and unmount times are nearly deterministic when the tape is rewound before unmount. However, the Ampex drive is different from the other drives we measured because we were able to unload the tape without rewinding it. If an unmount is requested without rewinding the tape, the tape is moved to a system zone and then unmounted (this is done to protect the tape). Variance in rewind time then becomes variance in unmount time. We tested the unmount time by seeking to a random location on the tape and then unmounting, as shown in Figure 1. The effect of the system zones can be seen in the sets of two parallel lines, offset by about 6 seconds that appear in the data. The average unmount time is 12.24 seconds with a standard deviation of 3.1 seconds.

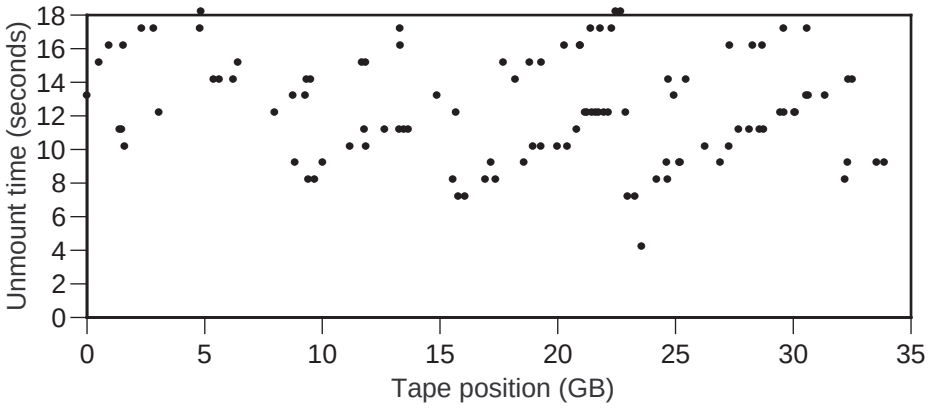

Figure 1. Unmount times for the Ampex 310.

If an Ampex tape is unmounted without being rewound, the first seek time increases (as is shown in Section 6.2). Because the seek and rewind times on the Ampex are so fast, rewinding a tape before unmounting reduces access times on average. We ran an experiment of repeatedly mounting a tape, seeking to a random location, reading 1 block of data, then unmounting returning the tape. We collected 60 data points for the case of rewinding before unmounting, and 60 data points for the case of unmounting without a rewind. If we rewound the tape before unmounting, then a fetch/return cycle takes 71 seconds with a standard deviation of 13. If we unmounted the tape without a rewind, the fetch/return cycle takes 85 seconds with a standard deviation of 30. A difference of means test indicates a significant difference between the two quantities.

6.2. Seek and Rewind Times

Because tapes are sequential media, they have large seek times. Overcoming seek time delays is a major focus of system optimization research. However, seek times on tapes can exhibit unexpected behavior. In this section we measure and model three types of seeks: the first seek after a mount (usually, but not always, seek from beginning of tape), a seek from the middle of the tape, and a short seek.

6.2.1. First Seek From Mount

For most tapes, "first seek from mount" is equivalent to "seek from beginning of tape" because the tape can only be unmounted (and thus mounted) when it has been rewound. This is true for all of the tapes in our study except for the Ampex; however, we will also report seek from beginning of tape for it.

The seek time from beginning of tape (BOT) for all of the tape devices we measured is shown in Figure 2. These charts also show the rewind time. As expected, seek times track rewind times well. For the helical scan tapes, both seek and rewind are linear in the distance the tape must travel.

The serpentine tapes, however, exhibit more complex behavior. Because the IBM 3590 and DLT 4000 have many pairs of tracks running in opposite directions, the seek & rewind times for a single pair of tracks characterize the seek & rewind behavior for an entire tape. The IBM 3590 has 4 pairs of tracks, and the DLT 4000 has 32 pairs. Figure 3 shows a detail of the DLT 4000's seek & rewind behavior, equivalent to one peak & valley (corresponding to a single forward and reverse track from the graph in Figure 2. This graph is similar to that of the IBM 3590, shown next to it for comparison; both show a piecewise linear seek time function along with the linear rewind function. Both serpentine tape drives use the two-dimensional topology of the tape as the primary mechanism for implementing a high-speed search. The fast seek speed is only 1.5 times and 2 times faster than the read speed for the DLT 4000 and the 3590, respectively. To attain high-speed search, the drives store the location of particular blocks in the tape's directory. To implement a distant seek, the tape moves to the last known block position that occurs before the desired block using the high-speed movement and then reads the tape until the desired block is located.

The Ampex 310 can be unmounted without a rewind. When the tape is mounted again, it is positioned at the middle of the tape and therefore is closer to the desired first seek position. In Section 5.3, we also measured the time to perform the first seek after a mount. The block positions for the first seek were randomly selected. We recorded the block position at unload and the block position for the first seek, as well as the seek time. However, we did not find any correlation between the distance between the block positions and the seek time. Instead, the seek time seems to be correlated with the seek block position. Figure 4 shows the result of the experiment. The time to seek to a block if no rewind is done before a mount is considerably larger (i.e., 25 seconds larger) than the time to perform a seek on a tape that has been rewound. Since most rewinds take less than 25 seconds (64% of the tape), we found that it is faster on average to rewind tapes after use.

6.2.2. Seek Times from Mid Tape

For the helical scan tapes, seek times between distant blocks in the middle of the tape fit well to a linear function. Figure 5, which shows seek times to and from a fixed block position on a 4mm DAT drive, is representative of helical scan drives.

However, seek times on a serpentine tape follow a more complex topology. The starting position divides the tape into four regions: before the starting point on a same-direction track, after the starting point on a same-direction track, before the starting point on a

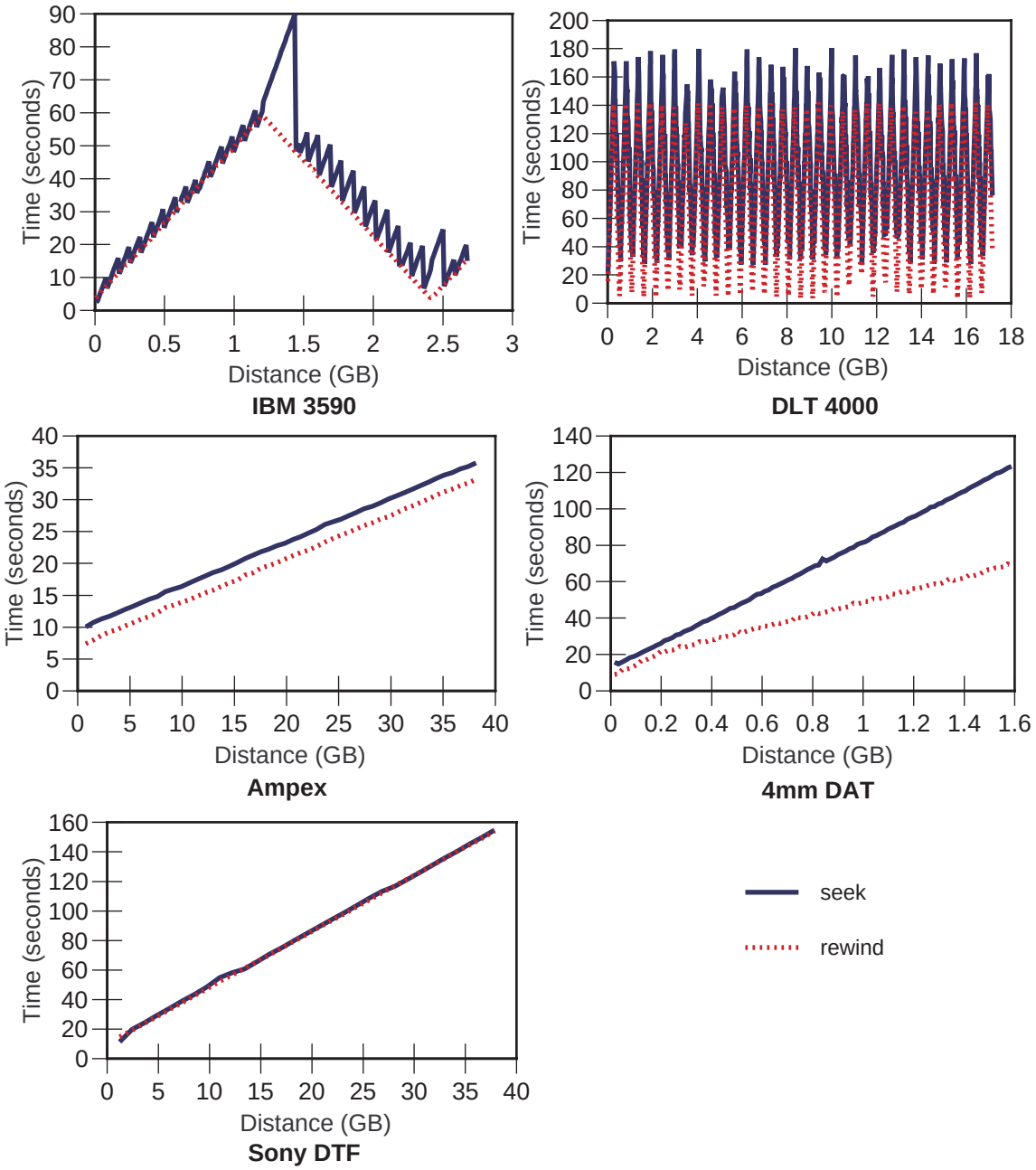

Figure 2. Seek and rewind times for various tape drives.

reverse-direction track, and after the starting point on a reverse-direction track. Figure 6 shows seek times from a tape block 73% into a reverse track of a DLT 4000 tape to other nearby track positions. The peaks of the seek time curve are offset from the track ends. After examining a number of these seek time curves, we found that the sizes of the offsets are reasonably stable. However, it is important to note that the seek times are non-linear, and that they indeed fall into the different categories mentioned above. This difference must be taken into account when laying out files that will potentially be accessed together.

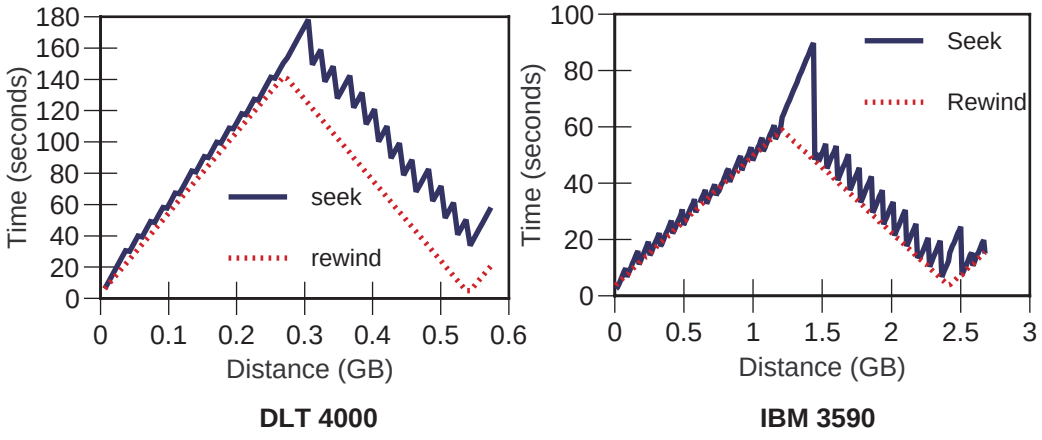

Figure 3. Detail of seek times for linear tape drives DLT 4000 and IBM 3590.

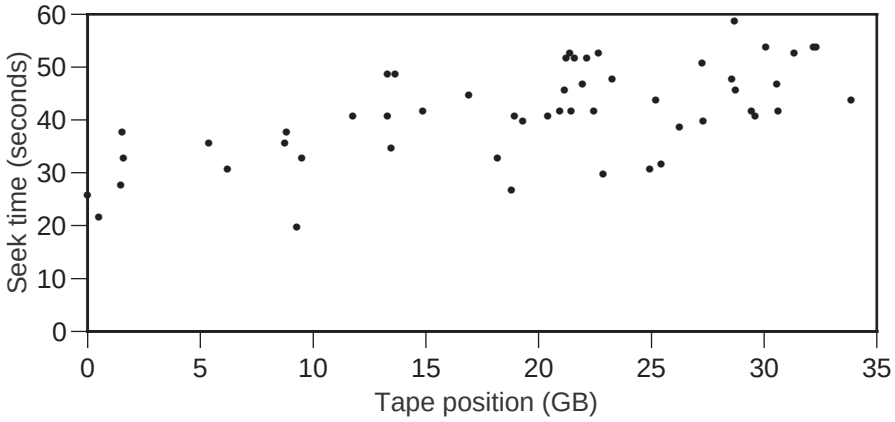

Figure 4. Seek times for Ampex 310 for tapes ejected without rewinding

6.2.3. Short Seeks

The increasing interest in building using tertiary storage in a more active role in both traditional mass storage systems and newer applications such as databases points to the need to investigate the performance of short seeks on tape. Data placement, indexing, and sizing algorithms have been developed with the assumption that seek time is directly proportional to seek distance. Often this is not the case. Furthermore, the seek time function over a short distance can be significantly different from the seek time function over a long distance. Because a tape seek often incurs a substantial delay, an important characterization is the distance at which it is faster to seek to the desired block position than it is to read in the unnecessary blocks.

We ran a set of experiments where we would repeatedly read in K blocks, and another set of experiments where we should seek past K-1 blocks and read in one block. The results of our short seek tests on all of the tape drives are shown in Figure 7. As the graphs show, serpentine drives differ markedly from helical scan drives in short seeks. This difference comes from the way that helical scan tapes must wrap the tape around the tape heads after

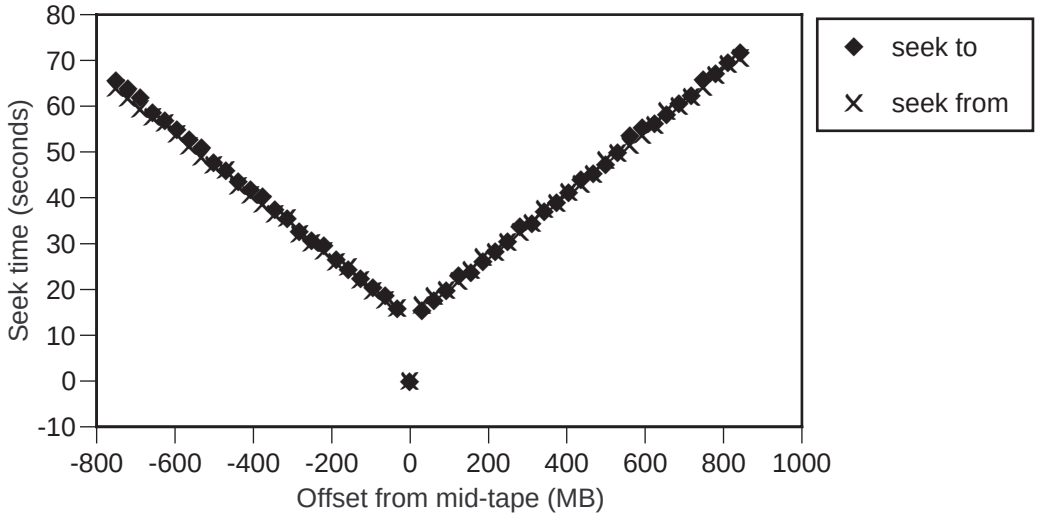

Figure 5. Mid-tape seeks on an 4mm DAT.

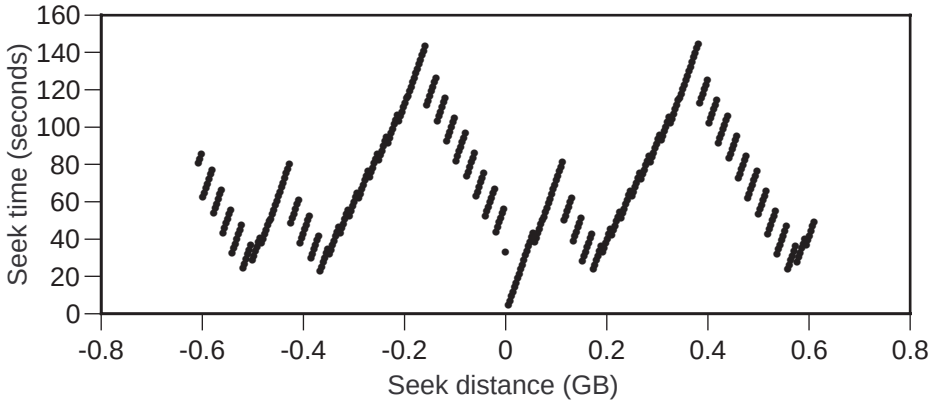

Figure 6. Mid-tape seeks on a DLT 4000, starting from 1 GB into the tape.

a high-speed seek. This wrapping process takes time, so it is often faster to simply read the intervening data than it is to perform a seek on helical scan drives.

For serpentine tapes, on the other hand, it takes almost no time to switch from high-speed tape transport to reading. As a result, seek and read times are close together. Additionally, the serpentine tape drives that we tested were sufficiently smart to pick the fastest seek method (high-speed scan or read) when a seek was requested.

7. Implications for Mass Storage System Designers

Our experiments provide several major benefits for mass storage system designers. First, they provide accurate (and unbiased) performance measurements of several modern tape drives, as reported in Section 6. Second, they highlight the performance differences between modern, high-performance tape drives using the competing technologies of helical scan and serpentine recording; in this section, we detail the implications these differences have for mass storage systems. Third, the benchmarks we used to gather the

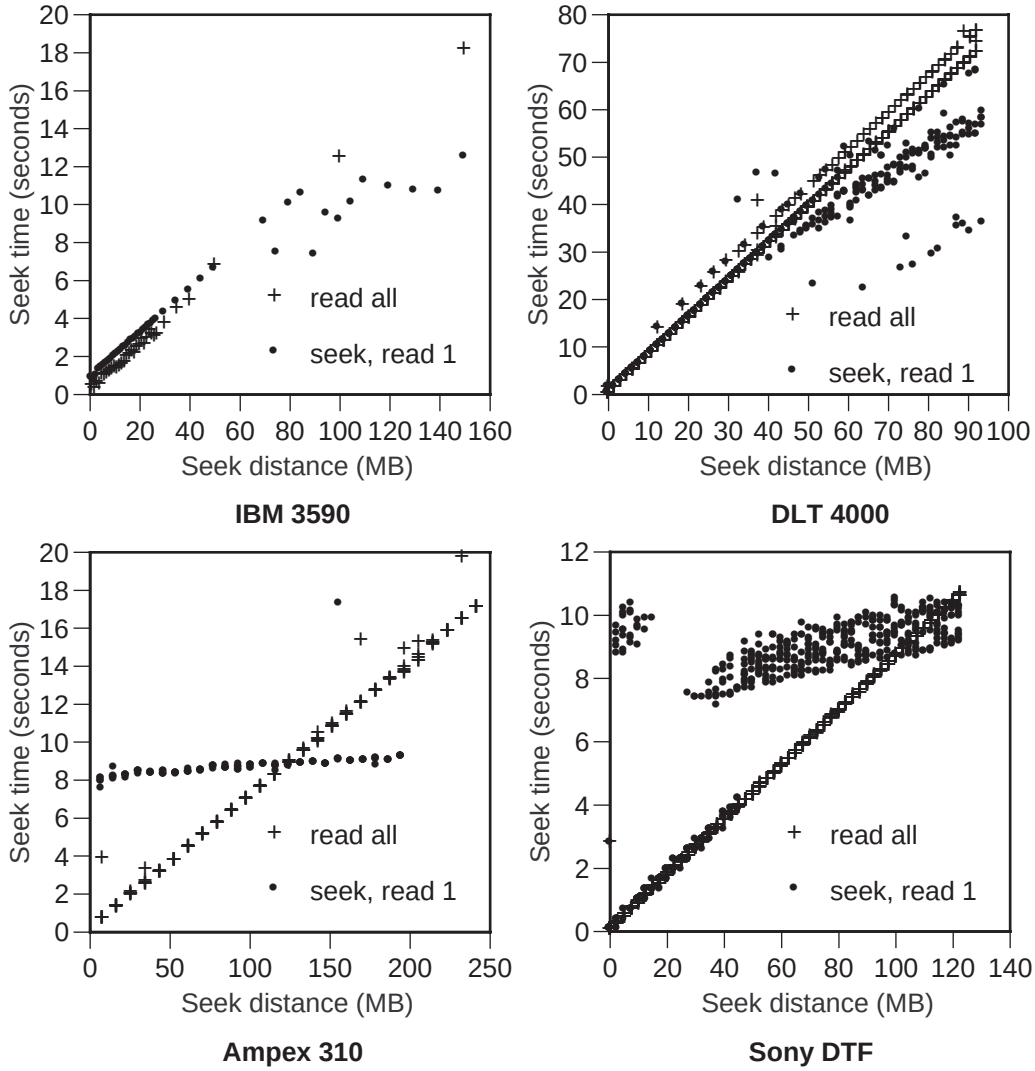

Figure 7. Short seek times for various tape drives

measurements can be run on newer tape drives, enabling others to do the benchmarking we reported in this paper.

7.1. Helical Scan vs. Serpentine Tape Drives

Our benchmarks uncovered several important performance differences between helical scan and serpentine tape drives that have implications for those building mass storage systems. While these differences are intuitively obvious, few (if any) mass storage systems take them into account.

7.1.1. Data Layout on Tape

The first major difference between the two types of tape recording is in the arrangement of data tracks. While maximum seek times for helical scan and serpentine drives are comparable, the seek profile from beginning of tape is not. Helical scan systems always require

longer to reach block N + X than they do to seek to block X from the start of the tape. However, serpentine tapes can often reach block N+X more quickly because of the arrangement of data on the many forward and reverse tracks.

As a result, designers of mass storage systems must take these differences into account in two ways. First, seek planning should consider track arrangement. It is not always faster on a serpentine tape to read sparsely stored data from a tape in "standard" order because this may involve many back-and-forth traversals of the tape. Instead, an algorithm such as the SCAN or CSCAN disk seek algorithms [28] should be used. Hillyer and Silberschatz have done some introductory work on applying such scheduling to tape systems [23], but there is still much to be done.

We also believe that data placement algorithms should take track arrangement into account. Large files should be placed at the physical start of track because the seek time to them will be considerably shorter. Even if this is done at the cost of wasting small amounts of space, the savings in seek time and thus response time to requests will balance the relatively small cost of purchasing additional tape media.

It is important to note that these layout optimizations apply only to serpentine tape; however, they may serve to make serpentine tape more attractive to mass storage system users by reducing the average seek time to the start of the data.

7.1.2. Seeking vs. Reading

Most current mass storage systems issue seek requests whenever they need to advance to new location on a tape. As the experiments in Section 6.2.3 show, however, this method results in significantly longer seek times than simply reading the intervening data.

This problem can be addressed in two ways. Systems using current tape drives should use the profile data presented in this paper (or other data gathered in similar ways) to compute whether seeking or reading to the desired location is faster. This phenomenon occurs not just for short seeks of a megabyte or two, but over seek distances of 100 MB or more; thus, storage systems that do not take this seek profile into account will have worse response time than those that do.

However, the optimal solution to this problem is for tape manufacturers to incorporate this knowledge into their tape systems. The seek profiles are relatively simple, allowing the tape drive itself to compute whether "fast seek" or reading is a faster way to reach the destination. Even if this is done, though, mass storage system designers must know what the seek profile is to optimally lay out their data.

7.2. Tape Benchmarks

The benchmarks we used to gather the data in this paper will also be very useful for those designing mass storage systems and those who build tape systems. Our benchmarks provide an extensive profile of tape drive performance for a wide variety of tape drives, better enabling storage system designers to optimize access.

7.2.1. Performance Quirks

Our benchmarks can uncover performance "hiccups" in a tape system, as they did for one of the tape drives covered in this paper. The poor seek performance near the start of the reverse track on the IBM 3590 is the result of a bug in the tape's microcode, which has since been fixed by IBM (though our tape drive was not updated before the tests were run). Nevertheless, the IBM 3590 still performs relatively poorly on seeks to the start of a reverse track even with the microcode fix applied. To work around this problem, we would suggest not starting files in this area and instead placing "dummy" data there. This optimization would reduce seek time to all files on the tape at relatively little overhead in space.

Other performance quirks can similarly be worked around if mass storage system designers know of their existence. Both the IBM 3590 and the DLT 4000 have a list of locations to which they can "fast seek." Mass storage systems that know the locations in the list could optimize file placement to minimize the seek time to most files on tape, not just the few that are placed near the physical end of a tape.

While helical scan tape drives do not share either of the above performance issues, they have different issues, as was discussed in Section 7.1. Knowledge of the crossover point where seeking becomes faster than reading will allow MSS designers to optimize access to random locations on tape, thus improving performance.

It is important to realize that many features implemented to improve performance may not actually do so. For example, mid-tape unmounting on the Ampex 310 is (presumably) intended to provide a method of optimizing performance by reducing rewind time; however, our experiments show that mid-tape unmounting is actually slower than rewinding before each unmount when the tape is formatted with the default number of system zones. Also, fast unmounts followed by slow mounts may be desirable in some environments such as real-time recording. Overall, however, performance "enhancements" should be benchmarked to ensure that they do indeed improve performance.

7.2.2. Benchmarking Issues

Our experiments also uncovered difficulties in gathering the performance data necessary for this study. Most of these were related to the software interface between the tape system and higher-level software. It is important for those benchmarking tape systems to understand the potential traps of gathering performance data.

The most significant problem we experienced was that some software "lied" about when a tape was actually ready to execute the next command. In many cases, opens, seeks and rewinds returned before the tape was positioned properly; as a result, the ensuing read would seem longer than it should be. Designers benchmarking other tape systems should be careful of this problem.

Another problem we experienced was that it was impossible to use a single program to conduct all of the benchmarks because of the differences in the interfaces between the many tape systems. One workaround for this problem was to use Perl rather than C for some of the benchmarks; while this made coding easier, we still had to modify the code to handle different tape systems on different platforms.

Discovering the "true" tape block size was also a difficult task. Some tape drivers accepted tape "blocks" larger than a certain amount, but chopped the blocks into smaller sizes for writing to tape. If the benchmark is trying to track changes in performance as tape block size changes, it is important to make sure that the device driver is actually writing the data in the desired block size.

8. Conclusions

We took detailed measurements from five common tertiary storage tape drives, spanning the range of low to high performance. The drives included helical scan and serpentine data layouts, and cassette and cartridge tape packages. Based on the benchmarks that we ran on the tapes, we made suggestions for improving the performance of any system that uses tapes as an active data storage medium — traditional mass storage systems, scientific databases, and archival storage systems.

Our future work will include more detailed measurements of tape system parameters as well as analytic models for the time necessary to perform various tape operations such as seeking and reading. While the measurements in this paper provide a good foundation for mass storage system designers, we hope that simple formulas for seek time and other parameters will make optimization of data layout and access an easier task.

We also plan to take performance measurements of additional aspects of tertiary storage devices. Of particular interest are measurements of very large robotic storage libraries in which the variance in the time to fetch a tape can be large, and serpentine tape drives that support partitioned tapes.

Acknowledgments

We would like to thank all of the people who gave us access to the tape drives and helped get the experiments set up. They include Jim Ohler and Jim Finlayson at the Department of Defense, P. C. Hariharan, Jeanne Behnke, and Joel Williams at the Mass Storage Testing Laboratory at NASA Goddard, and Gary Sagendorf at the Infolab at AT&T Labs - Research. We are also grateful for the feedback and guidance provided by the program committee, particularly our paper shepherd.

References

[1] J. Myllymaki and M. Linvy, "Disk-Tape Joins: Synchronizing Disk and Tape Access," *Proceedings of the 1995 ACM SIGMETRICS Conference*, 1995.

[2] D. Teaff, D. Watson and B. Coyne, "The Architecture of the High Performance Storage System (HPSS)," *NASA Goddard Conference on Mass Storage Systems and Technologies*, 1995, pages 45-74.

[3] L-F. Cabrera, R. Rees and W. Hineman, "Applying Database Technology in the ADSM Mass Storage System," *Proceedings of the 21st Very Large Data Base Conference*, 1995, pages 597 - 605.

[4] D. A. Ford and J. Myllymaki, "A Log-Structured Organization for Tertiary Storage," *International Conference on Data Engineering*, 1996.

[5] J. Kohl, M. Stonebraker, and C. Staelin, "HighLight: A File System for Tertiary Storage," *Proceedings of the 12th IEEE Symposium on Mass Storage Systems*, 1993, pages 157 - 161.

[6] B. Kobler, J. Berbert, P. Caulk and P C Hariharan, "Architecture and Design of Storage and Data Management for theNASA Earth Observing System Data and Information System (EOSDIS)," *Proceedings of the 14th IEEE Mass Storage Systems Symposium*, 1995, pages 65 - 78.

[7] L. Lueking, "Managing and Serving a Multiterabyte Data set at the Fermilab D0 Experiment," *Proceedings of the 14th IEEE Mass Storage Systems Symposium*, 1995, pages 200 - 208.

[8] J. D. Shiers, "Data Management at CERN: Current Status and Future Trends," *Proceedings of the 14th IEEE Mass Storage Systems Symposium*, 1995, pages 174 - 181.

[9] D. Schneider, "The Ins and Outs (and Everything Inbetween) of Data Warehousing," *Proceedings of the 23rd International Conference on Very Large Data Bases*, 1997, pages 1-32.

[10] M. Stonebraker, "Sequoia 2000: A Next-Generation Information System for the Study of Global Change," *Proceedings of the 13th IEEE Mass Storage Systems Symposium*, 1994, pages 47 - 53.

[11] P. Triantafillou and T. Papadakis, "On-Demand Data Elevation in a Hierarchical Multimedia Storage Server," *Proceedings of the 23rd Very Large Database Conference*, 1997, pages 226 - 235.

[12] R. A. Coyne and H. Hulen, "Toward a Digital Library Strategy for a National Information Infrastructure," *Proceedings of the 3rd NASA Goddard Conference on Mass Storage Systems and Technologies*, 1993, pages 15 - 18.

[13] D. Menasce and O. Pentakalos and Y. Yesha, "An Analytical Model of Hierarchical Mass Storage Systemswith Network Attached Storage Devices," *Proceedings of the 1996 ACM SIGMETRICS Conference*, 1996, pages 180 - 189.

[14] T. Johnson, "An Analytical Performance Model of Robotic Storage Libraries," *Performance '96*, 1996, pages 231 - 252.

[15] O. Pentakalos, D. Menasce, M. Halem and Y. Yesha, "Analytical Performance Modeling of Mass Storage Systems," *Proceedings of the 14th IEEE Mass Storage Systems Symposium*, 1995.

[16] T. Nemoto, M. Kitsuregawa and M. Takagi, "Simulation Studies of the Cassette Migration Activities in a Scalable Tape Archiver," *Proceedings of the 5th International Conference on Database Systems for Advanced Applications*, 1997.

[17] C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *IEEE Computer*, March 1994, pages 17 - 28.

[18] A. Drapeau and R. H. Katz, "Striped Tape Arrays," *Proceedings of the 12th IEEE Mass Storage Systems Symposium*, 1993, pages 257 - 265.

[19] J. Myllymaki and M. Linvy, "Efficient Buffering for Concurrent Disk and Tape I/O," *Performance Evaluation*, volume 27, pages 453 - 471.

[20] L. Golubchik and R. R. Muntz and R. W. Watson, "Analysis of Striping Techniques in Robotic Storage Libraries," *Proceedings of the 14th IEEE Mass Storage Systems Symposium*, 1995, pages 225 - 238.

[21] D. Ford and S. Christodoulakis, "Optimal Placement of High-Probability Randomly Retrieved Blocks on CLV Optical Disks," ACM Transactions on Information Systems, **9**(1), 1991.

[22] B. Hillyer and A. Silberschatz, On the Modeling and Performance Characteristics of a Serpentine Tape Drive," *Proceedings of the 1996 ACM SIGMETRICS Conference*, 1996, pages 170 - 179.

[23] B. Hillyer and A. Silberschatz, "Random I/O Scheduling in Online Tertiary Storage Systems," *Proceedings of the 1996 ACM SIGMOD Conference*, 1996, pages 195 - 204.

[24] R. van Meter, "SLEDs: Storage Latency Estimation Descriptors," *Proceedings of the NASA Goddard Conference on Mass Storage Systems and Technologies*, March, 1998.

[25] R. Venkataraman, J. Williams, D. Michaud, P C Hariharan, B. Kobler, J. Behnke, and B. Peavey, "The Mass Storage Testing Laboratory at GSFC," *Proceedings of the NASA Goddard Conference on Mass Storage Systems and Technologies*, March, 1998.

[26] G. Hull and S. Ranade, "Performance Measurements and Operational Characteristics of the Storagetek ACS 4400 Tape Library with the Cray Y-MP EL," *Proceedings of the NASA Goddard Conference on Mass Storage Systems and Technologies*, 1993, pages 229 - 240.

[27] D. Therrien and Y. L. Cheung, "Using Magnetic Tape Technology for Data Migration," *proceedings of the NASA Goddard Conference on Mass Storage Systems and Technologies*, 1993, pages 241 - 256.

[28] R. Geist and S. Daniel, "A Continuum of Disk Scheduling Algorithms," ACM *Transactions on Computer Systems*, **5**(1), 1987, pages 77-92.