Tertiary Storage: Current Status and Future Trends *

S. Prabhakar D. Agrawal A. El Abbadi A. Singh
Department of Computer Science
University of California
Santa Barbara, CA 93106

Abstract

This report summarizes current state of the art in tertiary storage systems. We begin with a comprehensive discussion of magnetic tape and optical storage technologies. This is followed by a classification of commercial products based on their performance characteristics. Our analysis of product data indicates that in contrast to disk technology, tertiary storage products have significant variablility in terms of data transfer rates as well as other performance figures. We then summarize efforts in the areas of operating systems, databases and advanced applications to integrate tertiary storage. We point out that different assumptions about the underlying technology result in entirely different algorithms and system design. We conclude the report with a speculation of future trends.

1 Introduction

With the recent improvements in network and processor speeds, several data intensive applications have become much more feasible than ever before. Examples of such applications include digital libraries, large scale scientific modeling and other multimedia applications. These applications are characterized by very large computational and storage requirements. Applications such as the Earth Observing System Data and Information System (EOSDIS) have estimated data generation rates of well over a Terabyte a day. In the present commercial setting and most likely in the near future, the only practical solution for storing such enormous amounts of data is tertiary storage. Although tertiary storage, in particular magnetic tapes, has been used solely for archiving or backup purposes, the exploding storage requirements and the high cost of secondary storage are forcing computer architects and designers to re-eavaluate the role of tertiary storage. In this paper we present some of the more recent research activities that study tertiary storage with this renewed interest.

There have been three major directions in which research on tertiary storage has been pursued. The first is research on tertiary storage systems from the operating system point of view. Due to the high seek and switch times for tertiary devices, reordering the I/O requests can have a major impact on performance. Research work on I/O scheduling for tertiary systems is reported. When data is stored in tertiary memory, it is necessary to extend the convenient and consistent access abstraction of the file system for secondary memory to operate with tertiary storage. Several efforts made to develop file systems for tertiary storage are

*Work supported by a research grant from NSF/ARPA/NASA IRI9411330.

discussed. In order to improve the bandwidth of secondary storage, striping has been used to harness the transfer rate of several disks for a single access. A natural question that arises is whether striping could also help alleviate the transfer rate problems of tertiary storage. Studies focusing on this question are summarized in this paper. The second is the investigation of issues involved in the integration of tertiary storage systems directly into database management systems (DBMS). Traditional database management systems are unaware of tertiary storage devices and do not optimize for them. This can lead to significant performance degradation if data resides on tertiary storage. Traditional databases exploit the physical properties of secondary storage, e.g. data layout to optimize data accesses. Hence it is necessary to reevaluate optimization techniques for DBMS when tertiary storage holds data. The third area of research has developed out of existing applications that are compelled to use tertiary storage due to their large data storage needs. These include scientific applications that manipulate terabytes of multidimensional array data and digital libraries. We summarize the initial experience from all these efforts in this paper.

The rest of the paper is organized as follows. Section 2 discusses the major tertiary storage technologies and the characteristics of tertiary devices. Section 3 describes current research work. Section 4 concludes the paper.

2 Tertiary Devices - current technology

In this section we present the various tertiary technologies that are prominent in the marketplace today. Subsection 2.1 describes the technologies used in present day commercial products and Subsection 2.2 gives performance and cost figures for these technologies.

2.1 Current Technologies

The most common tertiary storage devices are magnetic tapes, optical disks and magneto-optical disks. Tapes have been the traditional devices for archiving large amounts of data whereas optical devices such as CD-ROM and WORM are more recent tertiary devices. Two other technologies which hold a promise for the future are *Holographic Storage* and *Optical Tapes*. Both technologies offer very high storage capacities (100GB and 1TB respectively) but have only recently appeared on the market and have therefore not yet gained prominence. Consequently, there is very little information about their use or research based on these technologies, hence they are not discussed further - more details can be found in [Che94, HS96a].

2.1.1 Magnetic Tapes

Tapes offer the highest storage densities of all current storage media, however they are sequential access devices and most allow append-only updates. This is the reason why tapes have been historically used as an archival medium. New storage requirements however, necessitate the extension of the role of tapes in the storage hierarchy. Tapes are available in two forms: cartridge and cassette.

A cartridge contains only one reel on which the complete tape is wound when the cartridge is not loaded, whereas cassettes have two reels and the tape is spooled from one to the other as data is accessed. Cartridges always need to be rewound before they are unloaded from a drive since there is only one reel. Cassettes also need to be rewound because location data and other statistics about the tape are written at the beginning of the tape where they can be read when the cassette is loaded into a drive. However, newer models have multiple load/eject zones where the tape can be ejected.

The raw or uncorrected bit error rate for tapes is quite high - on the order of 1 in 100,000. Consequently, in order to make tapes more reliable, manufacturers make extensive use of error correcting codes. The error rates after error correction are between 1 in 10^{12} and 1 in 10^{15}. These error correcting codes consume a large portion of the tape space resulting in lower storage capacity. Most tape drives also offer on the fly data compression and decompression which increases the storage capacity. The amount of compression achieved depends on the data and typically is a factor ranging from 1 to 5.

Tapes can be classified according to the format in which data is recorded. There are three methods of data recording on tapes: *Longitudinal, Helical Scan* and *Transverse* .

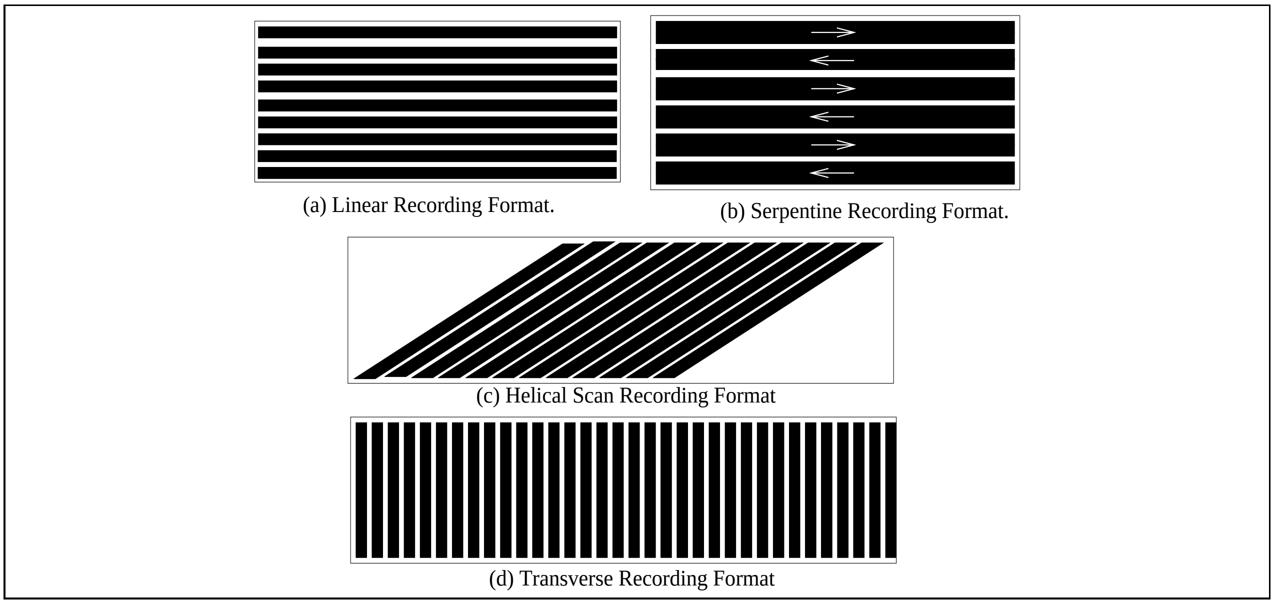

Figure 1: Tape Recording Formats

In longitudinal recording, also known as *Linear* recording, a number of heads positioned perpendicular to the length of the tape write data in tracks that run parallel to the edge of the tape, as shown in Figure 1(a). Each head writes one bit at a given position. The number of heads is usually small, such as the 9-track tapes which have nine heads that store a byte and a parity bit. A popular variation of linear tape is the *Serpentine* tape which records a set of tracks in one direction and at the end of the tape it reverses direction, shifts the heads and writes an adjacent set of tracks in the opposite direction. This process is repeated until a large number of loops of the tape have been made, as shown by the arrows in Figure 1(b). The last block on a serpentine tape is usually written on the same end of the tape as the first which results in a very small rewind time when the complete tape is written as compared to other recording techniques where the last record is written at the opposite end of the tape than where the first block is written. Typical values of the rewind times are 2 secs versus about a minute.

Helical scan tapes use the same method of data recording as Video Cassette Recorders (VCR), where the data is written at an angle of about 10 to 20 degrees to the direction of the tape as shown in Figure 1(c). There are a number of read/write heads that are placed on a cylinder which rotates at an angle to the direction of the tape. The tape is in contact with a part of this cylinder's circumference and moves at a slow speed as compared to the fast spinning head. Some index or location information may be recorded on the edge of the tape which can be read by a different head to allow precise position resulting in higher storage densities than linear tapes. Since the tape moves at a slow rate during normal play, rewind and forward operations can be done at 50 to 75 times the play rate by speeding up the tape and using the index information.

Due to the greater amount of contact between the head and tape in helical scan systems, there is greater wear of the tape and head as compared to linear or transverse recording techniques. Typical helical scan tapes can withstand about 1,500 passes over the head before the likelihood of significant data loss become high, whereas typical serpentine tapes can withstand 500,000 or 1,000,000 passes before similar loss of reliability. It therefore becomes important to replace helical scan tapes more frequently than the others. The advantage of the helical scan method of recording is higher storage density than the linear or serpentine methods.

Transverse tapes record data perpendicular to the length of the tape (see Figure 1(d)). The write head moves across the width of the tape and the tape advances slowly between each move. This technology offers high data density but due to the unsteady movement of the tape the transfer rate is not high. Therefore this type of recording is used for applications which need to store large amounts of data at low rates. It is used only in a small number of applications such as recording low rate sensor data.

2.1.2 Optical and Magneto-Optic Disks

Optical disks are similar to magnetic disks except that the recording medium is not magnetic. Data is read by reflecting a laser beam off the surface of the disk and detecting *pits* that have been burnt in during the recording phase. Optical disks can be classified into three categories: Read-only, Write-Once-Read-Many (WORM) and Read/Write. Read-only disks such as CD-ROMs are manufactured with written data which cannot be modified. WORM devices can be written only once and Read/Write disks can be written many times. The write operation on optical disks is usually slow since data can only be rewritten one track at a time. Therefore in order to modify part of a track the whole track must be read into a buffer, then the track is erased and finally the modified track data is written. The data density and transfer rate of an optical disk may be constant (Constant Linear Velocity or CLV disks) or may vary with the distance from the center (Constant Angular Velocity or CAV disks). As compared to magnetic disks, optical disks are slower, have less capacity and are more expensive. Their advantages over magnetic disks are that they are removable and they are not as susceptible to head crashes. Like magnetic disks, optical disks are random access devices with similar latencies.

Magneto-optic disks use both magnetic and optical technologies. The data is stored using the magnetization direction of a material as in regular magnetic disk technology, however the act of reading or writing is done through a laser beam instead of a magnetic read/write head. Magneto-optic disks are faster and less expensive than read/write optical disks and have been more successful commercially.

2.1.3 Robotic Devices

Many systems that use tertiary storage devices need to incorporate large numbers of tapes or disks to achieve the desired capacities. In order to provide automated access to these media, robotic changers are often employed to load and unload tapes or disks from the drives. Depending on the number of units in the system, various different mechanisms are available for automatic loading and unloading of tertiary devices.

For a small number of tapes (around 10) a stacker is most cost effective. A stacker usually moves a rack of tapes vertically or horizontally to bring the desired tape to a single drive. For larger numbers of tapes (up to about 100) a carousel is used. Carousels have one or two drives and a single robot arm which loads the tapes onto these drives. For larger numbers of tapes (up to around 1000) tape libraries that have several drives and one or two robot arms are available. For removable disks, jukeboxes that can hold several hundred disks are used.

2.2 Characteristics of Tertiary Storage Devices

Unlike magnetic disks whose parameters (such as seek times, rotational delays, head switch times etc.) have the same relative importance in determining behaviour, tertiary devices are greatly varied. For some technologies the seek time is the dominating factor, for others it is the switch time. Therefore it is important to understand the characteristics of the different technologies. The discussion in this paper is based on 1996 data available on the world wide web and recent research work of other researchers in this field.

2.2.1 Magnetic Tapes Systems

There is a large number of different tape standards that are commercially available today. Although it is not possible to discuss all of these, some of the more prominent ones are described below. Magnetic tape systems can be divided into low-end, mid-range and high-end products according to the capacity and transfer rates that they support.

On the low-end, the 4mm (Digital Audio Tape or DAT) and 8mm helical scan tapes are the most common formats. Typical 4mm cartridge capacities are 1.3, 2.0 and 2.3 GBytes with tape lengths of 60, 90 and 120 meters respectively. Each cartridge costs about $6 - $7. An example of a commercial 4mm drive is the Exabyte EXB-4200 which has a sustained transfer rate of about 250 KB/s. The search rate is 47MB/s and the rewind speed of the drive is 164 ips (inches per second). Two high capacity 8mm tapes are the Exatape160mXLe and Exatape AME with capacities of 7GB and 20GB respectively. The latest 8mm drive from Exabyte, the Mammoth, has a sustained transfer rate of 3MB/s, a search rate of 188MB/s and a rewind speed of 63 ips.

The half-inch serpentine tape is the most popular recording format in the mid-range technology. Quantum manufactures a series of tapes with capacities ranging from 2.6GB to 35GB. The Quantum DLTtape IV has a capacity of 35GB recorded on 208 serpentine tracks. These tapes have a much higher tape life of approximately 500,000 or 1,000,000 passes as compared to the helical scan formats. The DLT7000 tape drive which uses the 35GB tapes, can deliver a transfer rate of 5MB/s sustained and 12MB/s burst rates.

On the high-end are tapes and drives that are used in large high performance libraries such as the SONY DIR-1000 library. This drive uses the 19mm helical scan tapes which can store 12GB, 40GB or 100GB of data. The transfer rate of this drive can be as high as 32MB/s. Another example is the Ampex DST310 drive which can accept 25GB to 165GB 19mm helical scan cartridges and deliver a transfer rate of 15MB/s. The drive has a high seek rate of 800MB/s and also supports multiple load/eject zones which reduces the rewind time during cartridge unload.

Overall, we see that the transfer rates of magnetic tape devices vary from about 250 KB/s to 32 MB/s. This represents about two orders of magnitude, whereas the transfer rates for magnetic disks lie in the 1 MB/s to 15MB/s range. We note that tape drives not only have a greater variation in transfer rates, but the range is significantly larger both on the high end and the low end. Research efforts should therefore consider the case where the disk is faster as well as the case where the tape is faster [ML95].

The storage capacities of tapes also show a similar distribution when compared to the storage capacities of magnetic disks. Tapes vary in capacity from as low as 1.3 GB to 165 GB. Magnetic disks on the other hand have capacities of about 1GB to 10GB. The main advantage of tapes over disks is the cost of storage. The cost per megabyte of tape cartridges is around \$0.002, and for magnetic disks is around \$0.2. Note that the cost of magnetic disk storage inherently includes the cost of a drive because the magnetic disks are not removable media. Tapes on the other hand are removable media and therefore can be purchased independently of the drives. Tapes also offer the highest storage density among all the different storage media. The storage density per unit volume for tapes is an order of magnitude lower than that of disks. The low storage cost and high density are the prime factors behind the popularity of magnetic tape drives at present and in the foreseeable future.

The major disadvantage of tapes is that they are sequential access devices whereas disks (magnetic, optical or magneto-optical) are random access devices. The typical seek times for tapes range from tens of seconds to minutes, as compared to a few tens of milliseconds for magnetic disks and a few hundreds of milliseconds for optical and magneto-optical disks. This low seek speed of tapes renders them highly unsuitable for random access workloads. Although tape storage is cheap, the transfer rate of tape storage is usually not better than magnetic disks especially if cost is taken into account. Low cost drives for tapes cost a few thousand dollars, mid-range drives are at a few tens of thousands of dollars and high end drives (those that have transfer rates higher than disk rates) can cost well over \$100,000. Therefore, if we consider the drive cost per MB/s transferred, we see that tape drives are one or two orders of magnitude more expensive. Consequently, tape systems usually have a small number of drives handling a large amount of data and the aggregate transfer rates are not high.

2.2.2 Optical and Magneto Optical Disks

SONY is a leading manufacturer of optical and magneto optical products. The present standard for CDs and CD-ROMs is a 600MB storage and a transfer rates of 153.6 and 176.4 KB/s. However drives are available with transfer rates that are multiples of 2, 4, 6 and 8 of the 153.6 KB/s rate. CD-ROM drive costs range from about \$100 to \$350. The average access time for CD-ROMs is around 160-190ms, which is about an order of magnitude slower than magnetic disks. Recordable CD-ROMs are now also available, allowing one time recording. The recording rate is usually lower than the read rate by a factor of 2.

Common sizes for optical disks are 5.25 inches and 12 inches. The 5.25 inch disks have capacities ranging from 600MB to 1.3 GB. The data transfer rates are around 1.1 MB/s to 2.4 MB/s depending on the distance from the center of the disk and the disk format (bytes per sector and sectors per track). The 12 inch disks from SONY have capacities ranging from 3.2 MB to 15GB and transfer rates of around 1.3 - 2.7 MB/s for reads and 0.6 - 1.3 MB/s for writes. The average seek times for CAV (Constant Angular Velocity) disks is lower than that for CLV (Constant Linear Velocity) disks. For example, the SONY WDD531 drive has average seek times of 190ms and 800ms for CAV and CLV disks respectively.

Magneto Optical disks come in two sizes - 3.5 and 5.25 inches. The single sided 3.5 inch disks have capacities of about 120 MB to 230MB, and the double sided 5.35 inch disks have capacities of about 650MB to 2.6GB. The data transfer rates of magneto optical drives varies from 512 KB/s to 1.68MB/s for writes and 1 to 3.37 MB/s for reads. The average seek time ranges from about 28ms to 40ms.

Both Optical and Magneto Optical disks have greater reliability and storage life than magnetic disks. Since these technologies use lasers, the distance of the head from the recording medium can be larger than that for magnetic disks which use magnetic fields to read and write. Consequently, the chances of a head failure are reduced. The shelf life of data recorded on magnetic disks is 5 to 10 years compared to about 30 years for most optical media.

As compared to magnetic disks, both types of optical disks have lower transfer rates, particularly for write operations. The average seek time for optical disks is 10 to 100 times larger than magnetic disks, and for magneto optical disks it is about 2 times larger. In summary, magneto optical disks are superior to optical disks in terms of transfer rate and access latency. They are also cheaper than optical disks. Magneto optical disks are in turn slower than magnetic disks. The main advantage of magneto optical disks is that they are removable media and have a lower storage cost than magnetic disks.

2.2.3 Automatic libraries

Tape and disk libraries can be classified in terms of the number of media that they manage. Tape stackers usually handle up to 10 tapes, carousels and jukeboxes handle between 10 and about one hundred media and libraries handle several hundreds to thousands or even tens of thousands of media units.

An example of a stacker is the Exabyte EXB10h for 8mm helical scan tapes. It manages up to 10 cartridges for a capacity of 70GB and a single drive with a transfer rate of 500KB/s. The SONY OSL-2001 is an example of a jukebox. It can store up to 20 optical disks, each with a capacity of 1.3GB, giving a total of 26GB of storage. It has one or two disk drives each with a transfer rate of 2.4MB/s. The average disk exchange time is specified to be less than 5 seconds. The Exabyte EXB-480 is a small library that can hold a maximum of 80 8mm cartridges, each with a capacity of 20GB, yielding a total of 1600GB of storage. It can house up to 4 Exabyte Mammoth drives, each with a transfer rate of 24 MB/s. Finally, an example of a large library is the SONY DMS 700 system which can store up to 30 TB of data with 3 DIR-1000 series drives that can deliver as much as 32 MB/s each.

Robot access times vary from a few seconds to about 20 seconds for most libraries. Cartridge switch times, which include the time to rewind the currently loaded cartridge, eject the cartridge, replace it on shelf, pick up the new cartridge, load it into the drive and to setup the drive for reading, for libraries vary from a few seconds to about a minute. This time is the major source of latency for random access to tertiary devices. In the case of tapes, the seek time is also a significant factor.

It is important to note that the relative values of the switch time and the seek time vary greatly for the different types of devices. For optical and magneto-optical disks, the seek time is on the order of tens or hundreds of milliseconds whereas the switch time is a few seconds. For these devices, the switch time dominates the seek time. For tapes both times are of the same order of magnitude and it is not clear which, if any, dominates. For example, the DLT4700 mini library has a load/unload cycle of 29 seconds and the average seek time is about 68 seconds. In this case the seek time is more dominant.

2.2.4 Analysis

Tapes have the lowest storage cost for any storage medium today. They are two orders of magnitude cheaper than magnetic disks and almost an order of magnitude cheaper than optical disks. Optical and magneto optical disks also have a lower storage cost than magnetic disks. One factor for this cost advantage of tertiary storage is that the media is removable. This implies that one can purchase storage media independent of the drives, whereas magnetic disks (hard disks) are usually purchased together with the drives. Due to this factor and the wrapping of a large surface area of the tape onto a spool, tapes also have the highest storage per unit volume. These two factors make tapes the medium of choice when large amounts of data are to be stored.

While the removable media result in cheaper storage, one disadvantage is cartridge switching. Because cartridge switching takes place at mechanical speeds, switch times are of the order of several seconds or higher. This results in two or three orders of magnitude worse performance as compared to magnetic disks with latencies of the order of tens of milliseconds. Thus in order to obtain acceptable performance from tertiary devices it is important to reduce media switches.

Another drawback of magnetic tapes is that they are sequential access devices with large seek times. For many tapes the seek time is much larger than the cartridge switch time. The long seek and switch times make tapes highly unsuitable for random access workloads. Tapes also suffer from rapid media and head wear out due to the greater amount of contact between the tape and other parts of the drive mechanism that ensure the correct transport of the tape and the heads. This wear out needs to be monitored and appropriate replacement of media is crucial to the reliability of tape systems.

3 Current Research Directions

3.1 Operating Systems and Tertiary Storage

With the changing role of tertiary storage devices from being used solely for backup and archival purposes to holding "on-line" or "active" data, the need for better operating system support for these devices is imminent. The immediate need is for a file system abstraction, similar to that provided for magnetic disks.

3.1.1 I/O Scheduling

Due to the high cost of media switches and the high latency of tertiary devices, it would appear that reordering of I/O requests for tertiary devices could result in performance gains. This idea has been exploited for specific access patterns in almost all research activities on tertiary storage including the work done with DBMSs and other applications. Silberschatz and Hillyer have studied reordering in the general context for batched I/O requests for single DLT serpentine tapes [HS96c]. They have studied the problem of scheduling a set of I/O requests for a single DLT tape. The effectiveness of various strategies has been evaluated using a model of the DLT tapes that they have developed in [HS96b]. Simulation results indicate that for a small number of requests the best strategy is brute force, i.e. evaluate all combinations and choose the best, for larger sets, up to about 1500 requests, the best policy is to use a greedy algorithm for the asymmetric traveling salesman problem discussed in [LLKS85] called *LOSS* which has quadratic running time, and for even larger requests it is best to scan the whole tape sequentially and service the requests in the order in which they are placed on the tape.

The technique of reordering I/O requests for tertiary devices can be used not only within single tapes to minimize search times, but also at the automatic library level to minimize switch and search delays. Due to the very high cost of media switches, it is expected that reordering will result in significant gains. Other optimizations that the operating system can provide for tertiary storage include prefetching, caching and migration of data between secondary and tertiary storage. It is desirable that the user have knowledge about tertiary resident data and also some control over the movement of data. This control could be in terms of providing the operating system information about likely future requests to aid in prefetching and migration. Due to the high latencies of access, giving the user some control over the placement of data could result in performance gains. Users could then attempt to improve performance by techniques such as clustering data that is accessed together. On the other hand, data reorganization based upon observed access patterns could be automatically handled by the operating system, in a manner that is transparent to the user.

Myllymaki and Livny [ML96] have also investigated the benefits of executing tape and disk I/O in parallel. The authors show that any application that requires access to both disk and tape resident data concurrently, can benefit from overlapping the disk and tape I/O. Two buffering schemes are proposed for holding the tape resident data in a disk buffer. In one scheme the available buffer is divided into two equal parts which are used to alternately hold data from the tape and to perform the computation with the tape resident data. Due to the smaller buffer size more cycles are needed and this increases the I/O traffic, therefore a suggested modification is to use the same buffer. As data is used up by the computation, the freed space is used to buffer the data from tape.

3.1.2 File Systems for Tertiary Storage

One of the earliest efforts in this direction resulted in the production of the DataTree and UniTree commercial hierarchical file and storage management systems [McC90]. DataTree is MVS-based and UniTree is UNIX-based and both are the commercial versions of file systems developed at the Los Alamos National Laboratory and the Lawrence Livermore National Laboratory, respectively. The Unitree Mass Storage System, exports a Unix-like file system interface for secondary and tertiary storage. Access to the data is through the FTP and NFS protocols at the file level granularity. UniTree manages both secondary and tertiary storage.

Analyses of file system traffic indicate that most read requests are satisfied by the cache and as a result disk traffic tends to be write dominated. Moreover these writes are randomly distributed across the disks. In order to improve write performance, Log Structured File System (LFS) [OD89, RO92] was developed where the writes are committed to disk in such a way that new data blocks are written as a contiguous log rather than random locations on the disk. Modified data are written to the end of the log, invalidating the older copy, and data are read from the most recent location. Due to the sequential nature of access to tertiary systems, a log structured file systems appears to be a natural fit for providing efficient access.

Kohl et al [KSS93] extended LFS to tertiary storage and developed the HighLight tertiary storage file management system. Unlike UniTree, HighLight provides the capability of migrating parts of files and metadata about files to tertiary storage. Highlight exports a file system that incorporates both secondary and tertiary storage. The tertiary storage is used as a spill over area for secondary storage. User accesses are satisfied from secondary storage and if the data is not present, it is fetched from tertiary onto secondary storage. Garbage collection in HighLight is done only on the segments stored in secondary storage and although garbage collection of tertiary resident segments is planned, it has not been implemented yet.

In contrast, a log structured tertiary file system has been proposed by Ford and Myllymaki in [FM96] which manages data on tertiary storage alone. Segment buffers are used to hold data that is being written to tertiary storage and when these become full they are copied to a tertiary medium that is loaded on the *write drive*. Data is read by loading the medium that holds the data onto an available drive and reading it directly. Garbage collection is done by selecting some media to be cleaned using information that is maintained in a single file, loading those media onto a drive and copying the live data to the write drive and freeing up all the segments on the medium being cleaned. Data is accessible at the block level, several of which make up a segment. Simulation results show that this file system can achieve write performance similar to that of secondary storage and equal or better read performance.

Although some tape systems allow update-in-place at the expense of storage capacity, most allow append-only updates. Consequently, when data is stored on a tape, updates to the data would necessitate copying of large amounts of data if regular file systems were used. Log structured file systems however do not have this limitation since data updates never go to the same location as the old data, and data is always written in a sequential fashion on the current write media resulting in efficient use of the tape. This append-only limitation of tapes is also a strong reason for using log structured file systems.

3.1.3 Data Striping

Due to the high latency of access to tertiary devices, significant efforts have been made to improve tertiary storage performance. When the structure of the data is well understood such as in the case of the relational DBMS, or when the data is structured as arrays, as in some scientific applications, access to the data can be reordered to improve tertiary storage performance. However, in the general case, there is no common structure of the data nor is there sufficient knowledge of the access patterns. In such general settings, other methods have been proposed to improve the performance of tertiary devices. The idea of striping data across several disks and thereby overlapping the transfer of several disks has been extensively studied for RAID [PGK88] architectures. Could a similar technique be employed to improve the low transfer rate of tertiary devices? Drapeau and Katz have studied striping in the context of large tape libraries in the presence of concurrent random I/O [DK93].

They have found that although striping is beneficial for applications whose access patterns consist mainly of large sequential accesses where only one request is handled at a time, it is not necessarily beneficial for concurrent random access patterns. The reason for this poor performance is mainly the small number of readers available in these systems which leads to contention for the readers. A single access to striped data requires mounting all the media across which the data is striped as opposed to mounting only one medium for accessing non-striped data. In general, the request sizes must be large in order that striping outperforms non-striped layouts. The effect on improvements of various characteristics of reader, robots and tape drives was also investigated. With striping contention for readers continues even with largely improved robots and readers, resulting in poor performance for striping with small numbers of readers. The evaluations have been done through event based simulation of large tape libraries.

A similar analytical study of striping in libraries has been conducted by Golubchik and Muntz [GM95]. The model developed is a more general open system model with multiple sizes of requests within a single run and various stripe widths. The stripe width is the number of media units across which a file is striped. In the Katz and Drapeau work the model was a closed system with fixed size requests and fixed stripe width for all runs. A queueing model is developed for the library and workload described. Due to the complexity of the model, simulation is used to evaluate various striping policies. Their findings indicate that determining an optimal stripe width for every request size is highly dependent on the workload and that striping all request sizes across the same number of tapes is a good heuristic for minimizing average response time.

We therefore see that in order for striping to be effective in a concurrent environment, it is necessary to have an adequate number of readers. At present however, the high cost of readers limits the availability of readers. Improvements in performance of the various components alone will also not help striping unless the cost of readers decreases. Due to the high latency of access of tertiary systems, it is difficult to synchronize all the drives needed for a single striped access. A slow load on any of the drives can delay a striped data access. This implies that striping would only be effective for large accesses where the gain of parallel transfer outweighs such synchronization delays. Due to the great differences between tertiary and secondary technology, it seems unlikely that striping techniques used for secondary storage will be effective for tertiary storage without serious rethinking - a simple port of the ideas will most likely not prove effective.

3.2 Tertiary Storage in Database Management Systems

Commercial database systems are optimized for performance with primary and secondary memory. Tertiary memory, if included in the system serves only as a backup medium or slow store from which data is first moved onto secondary storage and then the DBMS operates on it as though it were secondary resident data. Commercial DBMSs are not directly aware of the presence of tertiary memory and do not attempt to optimize the access for this type of storage. On the other hand, significant work has been done on performance improvement for secondary storage systems. These enhancements are tailored to the characteristics of magnetic disks. Tertiary devices differ significantly from magnetic disks, in particular, they are not random access devices and media switch times are several orders of magnitude higher. Due to these differences, optimizations that work well for magnetic disks can result in disastrous performance on tertiary storage.

Recent improvements in processor and network speeds have made applications such as digital libraries, movies-on-demand and other data intensive applications much more feasible. Most of these applications have very large data storage requirements. As we move towards information and database systems comprising of terabytes or even petabytes of data, it is only a matter of time that tertiary storage will need to be integrated into the database storage hierarchy. Below we describe preliminary work in this area.

3.2.1 Join Operations

Research on integration of tertiary storage into DBMSs has focused on relational database management systems. To the best of our knowledge, at present the only existing DBMS that incorporates tertiary memory is POSTGRES [SRH90]. Tertiary storage is a first class storage entity in POSTGRES and tuples can reside on either secondary or tertiary storage. One of the most common and expensive operations of relational database systems is the *join* operation. Optimization schemes for joins of relations stored on disks have been shown to be inefficient when either one or both the relations involved in the join are stored on tape [SS93]. Work on the optimization of join operations for tape resident relations has been done using two different approaches.

Sarawagi and Stonebraker have investigated 2-way joins of relations which are both tape resident [SS93]. They assume that only one reader is available to the DBMS to read both relations and that the amount of disk space available for the join operation is small compared to the size of these relations. Consequently the drive will have to load the two media holding the relations alternatively in order to compute the join. Thus reducing the number of media switches has been a goal of the study. In the absence of indices on the join attributes, the authors suggest computing the join one block at a time rather than a tuple at a time in order to reduce the number of times the outer relation is fetched. In the presence of indices, the authors propose a two stage computation. In the first stage, the tuples that will be needed are determined using the indices. In the second stage, these tuples are fetched in an efficient manner by reordering the fetch operations based on the placement of these tuples. These techniques result in about two orders of magnitude savings in the number of switches and fetches.

Myllymaki and Livny, on the other hand, have studied join operations where one relation resides on secondary storage and the other on tertiary storage [ML95]. They have studied the performance and disk space requirements of various algorithms as the amount of main memory available to the join algorithm is varied. The medium holding the relation on tertiary storage is assumed to be mounted on the drive and is not unmounted during the join operation except when the relation spans multiple volumes. In this case the data on one volume is read completely before the next is mounted and since the time to read the entire volume is much larger than the time to mount a volume, the switch time has been ignored in this study. The benefits of executing disk and tape I/O in parallel have also been investigated. The authors [ML95] have shown that contrary to the conventional choice of the smaller relation to be the outer relation, when the larger relation is stored on a slower device, it is more efficient to select the the larger relation to be the outer relation. The authors observe that the operations of disk and tape access can be overlapped to reduce the total execution time. In order to study the memory requirements and performance of overlapping, memory and disk buffering are employed. It is found that if very little memory is available, no buffering should be used. With more memory, buffering is beneficial and if there is enough memory, then memory buffering outperforms disk buffering. The study also investigates the situation where the tape transfer rate is greater than the disk rate. In such a situation, both disk and memory buffering are harmful.

3.2.2 Architecture Issues

Sarawagi and Stonebraker have also investigated the architecture of database management systems that incorporate tertiary devices directly [Sar95a, Sar95b]. Traditionally, DBMSs have several independent components - the query executor, the disk cache manager and tertiary memory device scheduler. Queries are executed in a first come first serve manner - this mode of operation can lead to poor performance when data are fetched from tertiary storage. The authors argue in favor of a central *Scheduler* that has knowledge of the currently pending queries, the contents (and semantics) of the disk cache and the state of the tertiary memory. Using this knowledge the scheduler determines which data to read from tape onto disk, which data items to flush from the disk to tertiary memory if the disk is full and which queries to schedule next for execution based on the current contents of the disk. Data is exchanged between the disk cache and tertiary memory in units called fragments which are stored contiguously on a single medium. Queries on relations are broken into subqueries on the fragments of the relations. The subqueries operate on fragments that are in the disk cache and therefore the standard optimization techniques for relations on disk can be used. Due to the qualitative variations in the characteristics of different types of tertiary storage devices and the applications that use these systems, it is suggested that the policies for deciding which fragments to fetch and evict should take as input some of the device characteristics as well the workload characteristics and disk cache size. Finding an optimal solution to the fragment fetching problem is NP-complete and therefore several heuristics are presented [Sar95b].

For eviction, fragmented LRU and weighted-LRU type of policies are not effective because reasonable prediction of future use of fragments is possible from knowledge of the pending subqueries, and a more appropriate eviction policy is described. Simulations of the new architecture and policies indicate that the modifications are able to achieve most of the maximum possible improvements over the baseline policy of FCFS for fetching fragments and LRU for evicting them. These ideas have been incorporated into the

POSTGRES [Ols92] database management system and the performance of the new architecture and policies on the SEQUOIA benchmark [SFGM93] has been significantly better than the old architecture.

3.2.3 Future Directions

Although these studies provide a good starting point for the integration of tertiary storage and database management systems, there are lots of areas that need to be explored. The above studies on joins have focused on 2-way joins. Efficient join algorithms need to be studied for joins of more than two relations. The studies have also assumed that the tape drives were used solely by the DBMS. In general, however, the drives may be shared and a loaded tape may be ejected for use by some other user. This would cause additional rewind and search operations which would affect performance adversely. More detailed studies investigating the influence of rewind and search should be conducted since the search time for tapes can be higher than the switch time. Future research efforts should also investigate the possibility of combining non relational database systems such as object oriented database systems with tertiary storage. In general more detailed studies need to be done in order to incorporate tertiary storage into DBMS environments.

3.3 Application Oriented Research

Applications with very large data storage requirements need to use tertiary storage to hold active data. Prime examples of such applications are digital libraries and scientific applications that generate terabytes of data at a regular rate, such as the EOSDIS and the sequoia projects. Researchers working on such applications have focused on performance improvements tailored for the applications.

3.3.1 Large Multidimensional data

Scientific data is usually structured as large multidimensional arrays. The traditional method of storing these linearly, such as row-major or column-major schemes. This can result in very poor performance. Since data are accessed in small multidimensional blocks which require retrieval from widely separated locations in storage resulting in large seeks and media switches. The reason why data are usually not available in one location on the storage media is that traditionally data is stored in the order of generation. Two independent studies by Sarawagi and Stonebraker [SS94] and Chen et. al [CDK+95], have suggested that such data should be stored in a manner that facilitates retrieval for specific access patterns.

Sarawagi and Stonebraker [SS94] have proposed array storage schemes that optimize retrieval for a usage pattern that is either supplied by an end user or obtained from sampling a DBMS. The large multidimensional array is broken down into block sized *chunks* (multidimensional rectangular regions) such that the average number of blocks fetched for the given access pattern is minimized. Although an analytical solution to this problem is difficult, the authors suggest techniques to reduce the number of possibilities that need to be evaluated when conducting an exhaustive search. The chunks are then stored on optical disk in an order that achieves minimal average seek distance. A satisfactory condition for achieving this is also developed. In the presence of various classes of access patterns, the authors recommend keeping multiple copies of the same array optimized for different classes rather than having only one copy which optimizes the average for all classes. When data are stored on tertiary storage, it is desirable to reduce the number of media switches incurred. A partitioning of the array into chunks of the size of a medium and minimizing the number of platter switches for the access patterns is proposed for tertiary storage. However, for tapes, minimizing the seek time is more crucial than minimizing the number of switches. Experimental results indicate that these methods are very effective in reducing the access times for large arrays. Chunking is found to be useful even if the access patterns are not known.

The work done by Chen et al [CDK+95] focuses on the storage techniques for multidimensional data produced by scientific applications on tertiary media, more specifically tapes. Traditionally such data is stored in the order in which it is produced by the applications, whereas the authors propose storage based upon the access patterns and storage characteristics of the storage devices. Modifications to mass storage system interface protocols were made to control the placement of data as desired. The access types studied are queries that request data for certain ranges of spatial and temporal dimensions such as all temperature data for a certain region over a given month. The different queries types are given relative weights to indicate the importance of optimizing their performance. Since the assignment of weights has a major impact on the performance of the system and is a complex task, an interactive workbench has been developed which allows end users to observe the performance impact of assigned weights and to thereby choose by experimentation the weights that produce the more desirable results.

The method used to determine the data layout is as follows: The variables are first grouped into sets such that no variables in different groups are accessed together. Next, each variable in a group is partitioned into basic units, where a basic unit is the largest chunk of data that is either accessed totally or not at all by any query. Next, a linear order of these basic units is obtained which minimizes the sum of the distance between the first and last basic units needed by each query. This problem is shown to be NP-complete and some heuristics are presented to solve it in reasonable time. In order to reduce the number of files in the system, neighboring basic units are then combined into files which will be the unit of retrieval from the tertiary system. Finally, neighboring files are stored onto tape volumes such that a given file is stored on exactly one volume so as to minimize the average query response time using the hardware characteristics such as mount and seek times. Experimental and simulation results show that the new layout can result in reductions of almost two orders of magnitude. However, due to conflicting requirements of different query types, certain queries may actually perform worse. Such degradation can be reduced by data duplication.

4 Concluding Remarks

The recent trends in network and processor technology have made data intensive applications such as digital libraries and video-on-demand highly feasible. The storage requirements of these applications cannot be met by secondary storage due to the high cost and low storage density. Tertiary storage has traditionally been relegated to the role of storing archival or backup data. However, with its low cost and high storage density, tertiary storage, in particular magnetic tape technology, is the only reasonable solution to the large scale storage requirements. Tape technology however, is sequential in nature and is therefore ill suited for applications that require random access. Optical and magneto-optical disks do not suffer from this problem but current optical disks have low transfer rates. Even though it is highly likely that in the future, optical technology will overcome these limitations and become the technology of choice for tertiary storage, the current need for large scale storage can only be filled by magnetic tapes. Hence in the near future, tapes will be used for storing data that is accessed randomly.

The use of tapes in random access applications results in poor performance. Research efforts to overcome this limitation have been made but the gap is still large. Due to the variability of the various characteristics of tertiary systems, it is important that solutions take into account the parameters of the system for which the solution is designed. Thus in contrast to solutions for magnetic disks, it is not obvious that optimizations that work for one type of tertiary technology will also work for others. Much work needs to be done to overcome the problems associated with integrating tape storage.

Digital libraries [SF95] are one of the most exciting new applications. Although modern traditional libraries have electronic on-line catalogs for search purposes, it is only recently that the display and delivery of images, audio, video and other such data formats have become feasible. Due to the large volume of data, digital libraries will need to use tertiary storage. At present there is almost no reported work on the use of tertiary storage in digital libraries. Part of the reason is that it is only recently that such libraries are being built. In their earlier stages most systems do not support the large holdings that need tertiary storage. However, as these systems evolve tertiary storage will inevitably be incorporated. A major obstacle for research in digital libraries is the non-availability of user access patterns. Although one could attempt to extrapolate from existing on-line catalog usage patterns, it is not clear how valid such estimates would be because it is not known what uses such a facility will be put to. Presumably, given the power of digital tools, these libraries will serve much more productive purposes than traditional libraries.

Another application that is an appropriate candidate for tertiary storage is video-on-demand. This application has two characteristics - once the retrieval of a movie is initiated, then the sequence of data access for that request is completely determined, and there is a stron timing requirement for data retrieval. Preliminary studies [FR94, CPK95] indicate that although the cost effectiveness of tertiary memory is very attractive for video-on-demand systems, the current performance of tape drives and tape libraries is inadequate. In spite of these limitations, new playback techniques based on playback linking can be employed to succesfully use optical storage for video-on-demand [MSNN96]. Furthermore, it has been shown that if popularity of videos is non uniform, then combination of disk and tape storage becomes viable for video-on-demand applications [SHT96, KDST95].

References

[CDK+95] L. T. Chen, R. Drach, M. Keating, S. Louise, D. Rotem, and A. Shoshani. Efficient organization and access of multi-dimensional datasets on tertiary storage systems. In *Information Systems*, pages 155–83. Elsevier Science, 1995.

[Che94] A. L. Chervanak. *Tertiary Storage: An Evaluation of New Applications*. PhD thesis, Univ. of California, Berkeley, 1994.

[CPK95] A. L. Chervenak, D. A. Patterson, and R. H. Katz. Storage systems for movies-on-demand video servers. In *Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems*, pages 246–56, Los Alamitos, CA, Sep. 1995. IEEE Computer Society Press.

[DK93] A. L. Drapeau and R. H. Katz. Striping in large tape libraries. In *Proc. of Supercomputing*, pages 378–387, Portland, Oregon, 1993. ACM.

[FM96] D. A. Ford and J. Myllymaki. A log-structured organization for tertiary storage. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 20–7, New Orleans, Louisiana, 1996.

[FR94] C. Federighi and L.A. Rowe. A distributed hierarchical storage manager for a video-on-demand system. In *Proceedings of SPIE - The International Society for Optical Engineering*, pages 185–97, San Jose, CA, Feb 1994.

[GM95] L. Golubchik and R. Muntz. Analysis of striping techniques in robotic storage libraries. In *Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems*, pages 225–38, Monterey, CA, 1995.

[HS96a] B. Hillyer and A. Silberschatz. Storage technology: Status, issues and oppurtunities. Unpublished Manuscript, January 1996.

[HS96b] B. K. Hillyer and A. Silberschatz. On the modeling and performance characteristics of a serpentine tape. In *SIGMETRICS*, pages 170–9, Canada, 1996.

[HS96c] B. K. Hillyer and A. Silberschatz. Random I/O scheduling in online tertary storage. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Canada, 1996.

[KDST95] M.G. Kienzle, A. Dan, D. Sitaram, and W. Tetzlaff. Using tertiary storage in video-on-demand servers. In *Digest of Papers. COMPCON '95*, pages 225–33, 1995.

[KSS93] J. Kohl, M. Stonebraker, and C. Staelin. HighLight: a file system for tertiary storage. In *Proceedings of the Twelfth IEEE Symposium on Mass Storage Systems*, pages 157–61, Monetery, CA, 1993.

[LLKS85] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem*. Wiley, Chichester, 1985.

[McC90] F. McClain. DataTree and UniTree: Software for file and storage management. In *IEEE*, pages 126–8, 1990.

[ML95] J. Myllymaki and M. Livny. Disk-tape joins: Synchronizing disk and tape access. In *Joint International Conference on Measurement and Modeling of ComputerSystems. SIGMETRICS '95/PERFORMANCE '95*, pages 279–90, Ottawa, Canada, 1995.

[ML96] J. Myllymaki and M. Livny. Efficient buffering for concurrent disk and tape I/O. In *To appear in Proceedings of Performance '96*, 1996.

[MSNN96] T. Mori, H. Suzuki, K. Nishimura, and H. Nakano. Playback techniques for a video-on-demand system using an optical mass storage system. *Japanese Journal of Applied Physics*, 35(1B):495–9, January 1996.

[OD89] J. K. Ousterhout and F. Douglis. Beating the I/O bottleneck: A case for log-structured file systems. In *Operating Systems Review*, volume 23, pages 11–27, 1989.

[Ols92] M. A. Olsen. Extending the POSTGRES database system to manage tertiary storage. Master's thesis, Univ. of California, Berkeley, 1992.

[PGK88] D. A. Patterson, G. Gibson, and R. H. Katz. A case for redundant arrays of inexpensive disks (raid). In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 109–116, Chicago, 1988.

[RO92] M. Rosenblum and J. K. Ousterhout. The design and implementation of a log-structured file system. In *ACM transactions on Computer Systems*, volume 10, pages 26–52, 1992.

[Sar95a] S. Sarawagi. Database systems for efficient access to tertiary memory. In *Proceedings of the Fourteenth IEEE Symposium on Mass Storage Systems*, pages 120–6, Monterey, California, 1995. IEEE Comput. Soc. Press.

[Sar95b] S. Sarawagi. Query processing in tertiary memory databases. In *Proc. of the 21st Int. Conf. on Very Large Data Bases*, pages 585–596, San Francisco, California, 1995. Morgan Kaufmann.

[SF95] T. R. Smith and J. Frew. Alexandria digital library. *Communications of the ACM*, 38(4):61–62, April 1995.

[SFGM93] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith. The sequoia 2000 storage benchmark. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 2–11, 1993.

[SHT96] G.C. Shueng-Han and F.A. Tobagi. Hierarchical storage systems for on-demand video servers. In *Proceedings of the SPIE - The International Society for Optical Engineering*, volume 2604, pages 103–20, 1996.

[SRH90] M. Stonebraker, L. A. Rowe, and M. Hirohama. The implementation of POSTGRES. In *IEEE Trans. on Knowledge and Data Eng.*, volume 2, pages 125–142, March 1990.

[SS93] S. Sarawagi and M. Stonebraker. Single query opimization for tertiary memory. Technical Report s2k-94-45, Computer Science Div. U.C. Berkeley, December 1993.

[SS94] S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. In *IEEE Int. Conf. on Data Engineering*, 1994.