# WARP: Warp Around Data Placement Technique for Serpentine Tapes

**Ali Esmail Dashti**
Computer Engineering Department
College of Engineering and Petroleum
Kuwait University
P.O. Box 5969, Safat 13060, Kuwait
dashti@eng.kuniv.edu.kw
tel +965-481-1188 ext. 7438
fax +965-481-7451

**Abstract**

Due to the information explosion we are witnessing, a growing number of applications store, maintain, and retrieve large volumes of data, where the data is required to be available online or near-online [10, 3, 8]. These data repositories are implemented using hierarchical storage structures (HSS). One of the components of HSS is tertiary storage, which provides a cost-effective storage for the vast amount of data manipulated by these applications. To bridge the access-gap between the tertiary storage and the secondary storage, we propose a data placement technique, namely *WARP*: Wrap ARound data Placement. *WARP* may reduce the access time by 1 order of magnitude, depending on the tape device specifications and object sizes. An important feature of *WARP* is that it optimizes access-time independently of the retrieval order. We have implemented this technique on an IBM 3590 tape drive and have observed up to 5 times improvement in access-time as compared to other data placement techniques. Moreover, we report on the use of *WARP* with multiple load/unload position tapes (of the future).

## 1 Introduction

Due to the information explosion we are witnessing, a growing number of applications store, maintain, and retrieve large volumes of data, where the data is required to be available online or near-online [10, 3, 8]. These data repositories are implemented using hierarchical storage structures (HSS). One of the components of HSS is tertiary storage, which provides a cost-effective storage for the vast amount of data manipulated by these applications. However, it is crucial that the 3-4 orders of magnitude difference in access time between the tertiary storage and the secondary storage be bridged to allow online or near-online access to the tertiary resident data. This wide access-gap is mainly due to: the sequential nature of the most popular tertiary technologies (i.e., tapes) and the low number of drives per media in tertiary storage juke boxes. In this paper, we report on a data placement technique specifically designed for the serpentine tape technology, namely: *Wrap ARound data Placement* (*WARP*). We focus on tape technology because they provide the most cost-effective storage for very large databases, and more specifically on serpentine tapes because they are increasingly the technology of choice for mid-range and high-end systems. *WARP* may reduce the access time by 1 order of magnitude, depending on the tape device specifications and object sizes. An important feature of *WARP* is that it optimizes access-time independently of the retrieval order. This is achieved by exploiting the serpentine tape technology characteristics as opposed to the application characteristics. As part of this study, we also report on using multiple load/unload position tape devices with *WARP* data placement techniques.

We have implemented this technique on an IBM 3590 tape drive and have observed up to 5 times improvement in access-time as compared to other data placement techniques. Moreover, with *WARP*, the variance between the best, the worst, and the average access-times is very small, allowing for a priori prediction of the access-time behavior by time sensitive applications (e.g., real-time). When using multiple load/unload position tapes, we have shown that the access time to be a function of the track switch time ($T_{Track}$) and the rewind time ($T_{Rewind}$) for *WARP* and traditional data placement techniques respectively. This observation leads us to believe that *WARP* will outperform traditional data placement techniques by one order of magnitude, even with multiple load/unload position tapes.

The remainder of this paper is organized as follows. In Section 2, we present a short description of tape storage devices. In Section 3, we describe the general design of *WARP* data placement technique and two other data placement techniques, and we report a snapshot of the performance results. In Section 4, we describe the use of *WARP* with multiple load/unload position tapes. Finally, in Section 5, we summarize the results and conclude the discussion.

## 2 Tape Storage

Tape juke boxes (or silos) consist of a number of tape cartridges, tape drives, and robot arms. The data is stored on reels of tapes housed in cartridges (or cassettes). There are a large number of different tape formats that are distinguished according to the following criteria: 1) tape track orientation, 2) tape width, and 3) cartridge size [6]. The current tape technologies primarily use one of the two tape track orientations: *helical-scan*, or *serpentine* [6, 9]. With helical-scan recording, the data is recorded in diagonal stripes using a rotating drum (that contains the read/write heads) and a slow moving motor (that runs the tape over the drum). The axis of the drum is slightly slanted and, hence, the tracks formed by the passage of the heads across the tape are a series of diagonal stripes [6, 9].

This technology is similar to the VCR technology. Examples of helical-scan tape drives are: Exabyte Mammoth 8mm drives and Sony AIT 4mm drives.

With serpentine recording, the data is placed in longitudinal tracks down the length of the tape [6, 9]. The data is read by running the tape past a stationary recording assembly that contains the read/write head(s). If the number of read/write heads is smaller than the number of tracks (as is the case with most serpentine tapes) then at each pass only a portion of the tracks can be read, and multiple passes are required to read the tape in its entirety. (Note: large objects may occupy a number of tracks requiring multiple passes.) To optimize the read/write signal strength, the tape drive recording-assembly is moved slightly to establish exact track position for each tape pass. Examples of serpentine tape drives are: IBM 3590 and Quantum DLT7000.

To bridge the access-gap between tape storage and disk storage, the following solutions have been proposed: 1) to overlap disk and tape operations, 2) to apply intelligent tape scheduling, 3) to reduce the miss ratio at the magnetic disk level, and/or 4) to use intelligent data placement on tape. We focus on data placement on tapes to reduce the access-gap, where the data placement techniques can be classified into the following schemes: prediction-based and retrieval-based. The former predicts the future access pattern by analyzing the past references, the characteristics of the data, and/or the association between different data objects. The latter analyses the characteristics of the data retrieval to match them with those of the tape devices. These techniques can be used to optimize data placement on a single tape cartridge (i.e., intra-media optimization) to reduce the reposition time, $T_{Reposition}$, portion of the access time, $T_{Access}$, and/or optimize data placement on all tape cartridges within a juke box (i.e., inter-media optimization) to reduce the switch time, $T_{Switch}$, portion of $T_{Access}$[1]. The optimization can be for space, time, or a combination of the two. In this paper, we focus on design of a data placement for serpentine tapes, using intra-media retrieval-based data placement techniques.

### 2.1 Tape Model

The tape drive mechanism (including the load/unload mechanism and the recording assembly) is modeled with the following parameters:

- Number of read/write heads in the recording assembly, $N_{r/w}$.
- Tape load/unload position, $Pos$: the tape position when loading and unloading a tape cartridge from the tape drive, e.g.,
  - beginning of the tape, $Pos$ =*beginning* (e.g., Quantum DLT and IBM 3590 drives), or
  - middle of the tape, $Pos$ =*midpoint* (e.g., IBM Magstar MP drives).

[1]Note: $T_{Access} = T_{Switch} + T_{Reposition}$.

- Maximum rewind time, $T_{Rewind}$: it is the time required to rewind from one end of the tape to the other end (e.g., $T_{Rewind}$ for IBM 3590 tape drives is 64 seconds).

- Track switching time, $T_{Track}$: it is the time required to align the recording assembly from one track (or set of tracks) to another track (or set of tracks), with some minor adjustments to locate the appropriate tape block location. Note: helical-scan tapes do not incur a track switch time because there is only one (set of) track(s).

- Tape search time, $T_{Search}(i, j)$: it is the time required to search from tape location $i$ to location $j$ (i.e., long search). In our analysis, we assume that this time is a linear function of $T_{Rewind}$ and the number of blocks on each track, $T_{Search}(i, j) = \frac{T_{Rewind}}{N_S} \times \mid i - j \mid$. This would generate a saw-tooth like search time from different locations. However, as shown in [4, 5, 7, **?**, 2], the saw-tooth function contains a number of discontinuities. These discontinuities are due to the specific characteristics of tape devices are due to the different transport mechanisms used in locating tape blocks at different locations and distances.

A track group ($TG$) is the set of tracks that are read/written in parallel with the $N_{r/w}$ heads. These parameters characterize the general behavior of tape drives. We use them to illustrate the data placement techniques on serpentine tape devices. We verify the actual characteristics of the data placement techniques on an IBM 3590 tape device by measuring the total search times, track switching times, and rewind time (if necessary). Moreover, these parameters can be modified if necessary to better fit a specific serpentine tape device without affecting our proposed data placement techniques.

To retrieve an object from a loaded cartridge, two operations are performed: 1) tape and recording assembly are repositioned to the appropriate location(s), 2) data is transfered from the tape. The first operation may consist of: a tape search time, a number of track switches, and/or a rewind at the end of the retrieval (if the tape is to be ejected). The total time spent in tape and recording assembly repositioning (i.e., $T_{Search}$ and $T_{Track}$) is referred to as repositioning time, $T_{Reposition}$. We account for the fact that some retrieval behaviors require a rewind to $Pos$ after reading an object (to prepare the tape for ejection), by adding this time to $T_{Reposition}$. The total access time of an object, $T_{Access}$, is the sum of $T_{Switch}$ and $T_{Reposition}$ ($T_{Access} = T_{Switch} + T_{Reposition}$). From our evaluation of the IBM 3590 tape device, we have observed the following important facts: 1)$T_{Search}$ values range from 1 second to 64 seconds, and 2) $T_{Track}$ values range from 2 seconds to 3.5 seconds. These observations lead us to believe that exchanging $T_{Track}$ in place of $T_{Search}$ may improve the access time by up to one order of magnitude.

## 3 *WARP*: Wrap ARound data Placement

*WARP* data placement is designed specifically for serpentine tapes. Its objective is to eliminate $T_{Search}$ entirely by: 1) placing the object on even number of tracks, and 2) adjusting the amount of data on each track group ($TG$), such that every two tracks (that have opposite direction) hold *almost* the same amount of data. These constraints will result in minimizing the distance between the head of the object (i.e., the first block) and its tail (i.e., the last block). If the first block is placed on the load/unload position, $Pos$, then the tape will be at $Pos$ at the end of each object retrieval. This is desirable since most drives require the tape to be positioned at $Pos$ before unloading it. Moreover, by having the first and last blocks of all objects in $Pos$ then very little reposition time is incurred when retrieving multiple objects from the same tape cartridge. In essence, the observed tape length is optimal for each object when *WARP* is utilized, i.e. the object is wrapped around prior to reaching the physical end of the tape. The *WARP* data placement technique might require an extra track due to the added restrictions on the start position of the object head. *WARP* minimizes the search time in a tape cartridge, and hence, improves the initial latency and effective bandwidth of tape drives; it trades space for time (i.e., it optimizes for time) and may improve the $T_{Reposition}$ by 1-order of magnitude, depending on the tape device specifications and object sizes; and, it optimizes data placement on a serpentine tape regardless of the retrieval order, and hence it is superior to prediction-based data placement techniques, on serpentine tapes.

We are not aware of any work on intra-media retrieval-based data placement optimization to reduce access time. This lack of attention is mainly attributed to the assumption that with tapes, positioning time between two logical locations is proportional to their sequential distance. Hence, the only data placement optimization can be based on placing the blocks that are predicted to be retrieved together on sequential locations that are close to each other. First, this assumption is not true for serpentine tapes. That is, although the reading of serpentine tapes is sequential, its read/write head follows a zig-zag movement sweeping the tape back and forth many times. For example, two locations might be far from each other in sequential distance, but close physically (by performing a track-switch operation). Second, before ejecting a tape, the tape needs to be rewound to a load/unload position. Hence, a placement technique that leaves the head close to load/unload position after each object retrieval has merit. *WARP* exploits these characteristics of serpentine tapes to place the objects intelligently on a single tape in order to reduce the $T_{Reposition}$ portion of $T_{Access}$ (independent of the retrieval order of the objects).

We compare the performance and the cost effectiveness of *WARP* with two traditional intra-media retrieval-based data placement techniques: 1) contiguous data placement with track sharing (*CP w/ sharing*), and 2) contiguous data placement without track sharing (*CP w/o sharing*). *CP w/ sharing* traditionally has been the data placement of choice for tape resident data because of its high data packing characteristics (i.e., it optimizes for storage space), and *CP w/o sharing* is a modified version of the *CP w/ sharing* to improve $T_{Reposition}$ for certain types of retrieval behaviors.

Without loss of generality and to simplify the discussion, we assume a single read/write head per recording assembly, $N_{r/w} = 1$, and we assume an object of size $O$ to be placed on a tape consists of $n$ equi-sized blocks ($b_0$, $b_1$, ..., $b_{n-1}$), where each object block size is equal to a tape block size. To analyze the different data placement techniques, we assume two modes of retrieval: Single Object Retrieval (SOR), and Multiple Object Retrieval (MOR). With SOR, one object is read from a single tape, hence, we expect the reposition time of each object to consist of a reposition time from $Pos$ to the beginning of the object, a number of track switches, and a rewind time from the end of the object to $Pos$. With MOR, multiple objects are read from a single tape, hence, We expect the reposition time to consist of a reposition time from the end of the previously read object to the beginning of the requested object, and a number of track switches. We refer interested readers to [1, 2] for a complete analysis of the different data placement techniques.

To analyze the performance and cost effectiveness of these data placement techniques, we implemented them using an IBM 3590 tape drive, housed in a 3494 tape library. The tape drive has a 9 MB/sec sustained data rate, and the tape cartridges have a 10 GB capacity. We evaluated these techniques under different object sizes and request behaviors. We have observed that *WARP* consistently incurs a shorter $T_{Reposition}$ as compared to the other techniques, under all test conditions. For example, when retrieving a single 2 GB object, we have observed that the average reposition times are as follows: 1) $11$ seconds (using *WARP*), 2) $58$ seconds (using *CP w/ sharing*), and 3) $29$ seconds (using *CP w/o sharing*), and the worst access times are as follows: 1) $13$ seconds (using *WARP*), 2) $93$ seconds (using *CP w/ sharing*), and 3) $30$ seconds (using *CP w/o sharing*). However, this improvement in $T_{Reposition}$ comes at an increased storage cost. In this case, *WARP* wastes approximately $600$ MB of storage per $2$ GB object (*CP w/o sharing* also wastes $600$ MB per $2$ GB object).

To evaluate the effectiveness of *WARP*, We consider two metrics for evaluation: cost per MB ($Cost_{MB}$) and initial latency ($T_{Latency}$). We show that *WARP* consistently has a lower $T_{Latency}$, i.e., $49\% - 69\%$ reduction in $T_{Latency}$, even though it increases the number of tape switches ($T_{Switch}$). However, $Cost_{MB}$ of *WARP* is sometimes higher and sometimes lower than the other techniques, depending on the application requirements (i.e., database size and tape bandwidth requirement). *WARP* for the most part is more expensive for small objects (i.e., objects 500 MB or smaller), while it is either cheaper or similar in price for large objects (i.e., objects 1.5 GB or larger). For medium sized objects (i.e., objects between 500MB and 1.5 GB), the difference between $Cost_{MB}$ for the different data placement techniques vary depending on the application requirements.

## 4 Multiple Load/Unload Position Tapes

One of the main limitations of the current serpentine tape technology is the long access times. To improve the access time, manufactures may increase the number of load/unload positions along the tape, which might improve the reposition time for SOR retrievals. MOR retrievals cannot take advantage of the multiple load/unload positions, because they do not require tape cartridge switching. Currently, all serpentine tapes have either $Pos = beginning$ or $Pos = middle$. This is due to: 1) the characteristics of the type of reels used in the tape cartridges, and 2) the servo mechanism used by these tape drives. For example, with IBM 3590 and DLT 7000 tape cartridges, there is only one reel to hold the tape. When the tape cartridge is loaded onto a tape drive the tape lead is pulled into the drive and attached to the tape drive reel. The advantage of this type of cartridge design is the reduced tape cartridge footprint. Assuming that the tape cartridge design does not constrain the number of load/unload positions, we assume the following possible alternatives:

1. $Pos = beginning$,
2. $Pos = middle$,
3. $Pos_1$, $Pos_2$, $Pos_3$, ..., $Pos_{N+1}$, and
4. $Pos = anywhere$.

The first two alternative load/unload positions were discussed in Chapter 3. For alternative 3, we assume that the tape servo mechanism can eject a tape at pre-specified locations on the tape, e.g., $Pos_1$, $Pos_2$, ..., $Pos_{N+1}$. These locations can be either identified as simply load/unload positions or they can be associated with partition boundaries, i.e., the beginning and end of partitions. For alternative 4, we assume that the tape can load/unload at any location along the tape. This might be possible if there is enough information along the servo track and/or the tape drive maintains the location information in some type of on-tape cartridge memory (i.e., smart tape cartridges).

For alternative 3, we assume that there are a number of load/unload positions (i.e., $N+1$ load/unload positions), where these positions are equally spaced along the tape. With *CP w/ sharing*, the reposition time will depend on the number of load/unload positions:

$$T_{Reposition}(CPw/sharing) = \sum_{j=1}^{N+1} g(Pos_j) \sum_{i=1}^{N} (\frac{1}{N}(f(Pos_j, Part_i) + \frac{3}{4})\frac{T_{Rewind}}{N}). \quad (1)$$

In Equation 1, we assume that the probability of accessing an object is uniformly distributed along the tape. In the first summation, $g(Pos_j)$ is the probability of loading a tape at any of the $j$ positions (when previously it was ejected after completing an object transfer):

$$g(Pos_j) = \begin{cases} \frac{1}{2N}; \text{ when } j = 1 \\ \frac{1}{N}; \text{ when } 1 < j < N+1 \\ \frac{1}{2N}; \text{ when } j = N+1 \end{cases} \quad (2)$$

Since we assume that object tails are uniformly distributed along the tape, then the probability of loading a tape at any of the positions between $1$ and $N+1$ is $\frac{1}{N}$. For the first ($Pos_1$) and last ($Pos_{N+1}$) positions, the probability is reduced in half, i.e., $\frac{1}{2N}$. This is due to the fact that at the first ($Pos_1$) and the last ($Pos_{N+1}$) positions there are discontinuities, i.e., at $Pos_1$ there is no tape before beginning of tape (BOT) and at $Pos_{N+1}$ there is no tape after end of tape (EOT).

In Equation 1, the second summation is the sum of all possible expected $T_{Reposition}$'s for a position $j$. In this case, we assume that there are $N$ virtual partitions, $1$ through $N$. Since object *heads* are uniformly distributed along the tape, therefore, the probability of finding the *head* of an object in any of the $i$ partitions is $\frac{1}{N}$. The distance, in number of partitions, traveled from the expected load/unload position ($Pos_j$) to any particulate partition $i$ is governed by function $f(Pos_j, Part_i)$:

$$f(Pos_j, Part_i) = \begin{cases} 0; \text{ when } j = i \\ j - i + 1; \text{ when } j > i \\ i - j; \text{ when } j < i \end{cases} \quad (3)$$

After repositioning to the appropriate partition, on average, it is necessary to reposition the tape read/write *head* $\frac{1}{2}$ a partition length, to locate the object *head* in the partition. Upon the completion of an object transfer, it is necessary to locate the closest load/unload portion to eject the tape. On average, the distance between the tail and the closest load/unload position is $\frac{1}{4}$ a partition length, because there are two load/unload positions on each side of every partition. This yields a total of $\frac{3}{4}$ of a partition length reposition to locate the object *head* and to locate the closest unload position from the tail. To convert the partition length into time, we assume that the search time in the partitions is linearly proportional to $T_{Rewind}$[2].

As $N$ increases, the reposition time approaches $\frac{1}{3}$ of a $T_{Rewind}$, i.e., $T_{Reposition} = \frac{1}{3}T_{Rewind}$. This is similar to MOR access retrieval, because the load/unload position is not restricted and is uniformly distributed along the tape to match the uniform distribution of the object tails and, hence, the tape is ready to eject immediately after object transfer. When the tape is loaded, since the load position is uniformly distributed along the tape, on average, the distance between the *head* of the referenced object and the load position is $\frac{1}{3}$ of a tape. Therefore, for alternative 4, the average $T_{Reposition}$ of *CP w/sharing* is approximately $\frac{1}{3}T_{Rewind}$.

With *CP w/o sharing*, the objective is to fix the location of the *head* so that it is possible to start the object transfer immediately after loading the tape cartridge. With *WARP*, there is also a restriction on the location of the tail of the object, such that the tape cartridge is ready for ejection immediately after reading the object. For alternative 3 and 4, it is possible to have multiple load/unload positions, and hence, it may be possible to have multiple locations for the object *heads* and tails. This will result in a higher $T_{Reposition}$, for both placement techniques. For example, if there are two load/unload positions then, on average, it is necessary to travel $\frac{1}{2}$ the distance between the two load/unload positions, due to the uniform distribution of the accesses to the two load/unload positions. It is more advantageous to use a single load/unload position with these data placement techniques (specially *WARP*) to minimize the $T_{Reposition}$. However, these two techniques waste tape storage space, see [1, 2]. It was shown that these two techniques waste on average $\frac{1}{2}$ track when the load/unload position is in the middle of the tape, i.e., $Pos = middle$.

[2]For large $N$ this might not be true, due toe the acceleration and deceleration characteristic of short searches.

With multiple load/unload positions, it might be possible to use a position that minimizes the wasted space. This can be achieved by shifting the position from the middle of the tape to either end of the tape, i.e., to $BOT$ or $EOT$. This will result in two different virtual partition sizes. The smaller partition may be used for smaller objects (and hence, less space is wasted per smaller objects), and the larger partition is used for larger objects. On average, less than one track is wasted for small objects and more than one track is wasted for large objects. However, this method might reduce the percentage of the wasted space, and in turn, it might reduce the overall storage cost of *CP w/o sharing* and *WARP*.

## 5 Conclusion

A trend in serpentine tape technology indicates an increase in the number of tracks per cartridge. Consequently, the read/write heads require to sweep the tape a number of times in forward and backward directions in order to read a large object. We have designed and developed a new data placement technique for serpentine tapes, termed *WARP*, that takes advantage of this zig-zag sweeping characteristic to reduce the access time to tape resident data. As a result, the access gap between disk storage and serpentine tapes is reduced significantly to allow for on-line or near-online access to tape resident data, within HSS. Moreover, given the expected sub-second track switch time, $T_{Track}$, in the near future, this novel data placement technique is expected to observe an even better performance.

We investigated the performance and cost effectiveness of *WARP* and two traditional data placement techniques, in the context of CM servers using tape systems with IBM 3590 tape drives. We observed $49\%$-$69\%$ ($1.96$-$3.23$ times) reduction (improvement) in $T_{Latency}$, when using *WARP*. Moreover, the variance between the best, the worst, and the average $T_{Reposition}$ (and in turn $T_{Latency}$) is very small, allowing for a priori prediction of $T_{Latency}$ behavior for multimedia applications. However, $Cost_{MB}$ of *WARP* is sometimes higher and sometimes lower than that of *CP w/ sharing* (but always equal to or better than *CP w/o sharing*, depending on the application requirements (i.e., database size and tape bandwidth requirement). *WARP* for the most part is more expensive for small objects (i.e., objects 500 MB or smaller), while it is either cheaper or similar in price for large objects (i.e., objects 1.5 GB or larger). For medium sized objects (i.e., objects between 500MB and 1.5 GB), the difference between $Cost_{MB}$ for the different data placement techniques vary depending on the application requirements. Using the two metrics, a hierarchical storage manager or a file system may judicially decide the appropriate placement technique, given the object size and application requirements.

We also extended our study of the data placement techniques to consider multiple load/unload position tapes. With multiple load/unload positions, we have shown that $T_{Reposition}$ of *CP w/ sharing* will be a function of $T_{Rewind}$, whereas $T_{Reposition}$ of *WARP* is a function of $T_{Track}$, and hence, *WARP* is expected to out perform *CP w/ sharing* by an order of magnitude.

**Acknowledgments**

This research was supported in part by the Kuwait University under grant EE 01/00, and NIH grant 1P20MH52194-01A1 (USCBP).

**References**

[1] A. E. Dashti and C. Shahabi. Data placement techniques for serpentine tapes. In *Proceedings of the New Trends in Multimedia Systems in the Software Technology Track of the Thirty-Third Hawaii International Conference on System Sciences (HICSS-33)*, 2000.

[2] A. E. Dashti, C. Shahabi, and R. Zimmermann. Cost analysis of serpentine tape data placement techniques in support of continuous media display. In *Proceedings of the 10th International Conference on Computing and Information (ICCI '2000)*, 2000.

[3] G. Gibson, J. S. Vitter, and J. Wilkes. Report on the Working Group on Storage I/O for Large-Scale Computing. *ACM Computing Surveys*, 28(4), December 1996.

[4] B. Hillyer and A. Silberschatz. On the Modeling and Performance Characteristics of a Serpentine Tape Drive. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 170–179, May 1996.

[5] B. Hillyer and A. Silberschatz. Random I/O Scheduling in Online Tertiary Storage Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 195–204, June 1996.

[6] B. Hillyer and A. Silberschatz. *Storage Technology: Status, Issues, and Opportunities*. http://www.bell-labs.com/user/hillyer/, June 1996.

[7] B. Hillyer and A. Silberschatz. Scheduling Non-Contiguous Tape Retrievals. In *Proceedings of the Joint NASA/IEEE Mass Storage Conference*, pages 113–124, March 1998.

[8] D. Malon and E. May. Critical Database Technologies for High Energy Physics. In *Proceedings of the Very Large Databases (VLDB) Conference*, pages 580–584, 1997.

[9] P. McGowan and J. Hickey, editors. *Quantum DLT Handbook*. Quantum Corp., 1998.

[10] A. Silberschatz, M. Stonebraker, and J. Ullmen. Database Systems: Achievements and Opportunities. *NSF Workshop on the Future of Database Systems Research*, May 1995.