Issues in the Architecture of a Document Archiver using Optical Disk Technology

S. Christodoulakis *
Computer Systems Research Institute
University of Toronto
*Part of this work was done while the author was visiting the Computer Research Institute of Crete

ABSTRACT

We present issues related to the architecture of a document archiver using optical disk technology. In particular we examine the problems of data placement in the optical disk, storage hierarchies, data duplication and version control. Simulation results and analytical results are presented. These results are used to analyze the effect of various design decisions on the performance of such a system.

1. Introduction

In today's offices there is a need for systems which efficiently store and retrieve multimedia documents which contain information in the form of attributes, text, graphics, bitmaps and voice. Such systems will have to archive a large number of documents for a long period of time *(multimedia document archivers)*. The users of these systems will be able to access multimedia documents by specifying some information on the document's content. For such systems to be succesful access times for multimedia documents should be small. In this paper we describe some architectural issues related to the design of a document archiver and we analyze their impact on the performance of the system.

Existing secondary storage technology presents several limitations for document archiving. The most successful technology, the magnetic disk, has far too small storage capacity for such an application. Documents with bitmaps even if they are compressed they may easily require storage capacity of the order of hundreds of kilobytes, and there will be many such documents in an office of a large organization where information is archived for several years. A new emerging secondary storage technology, the optical disk, appears to be the most promising technology for document archiving due to the large storage capacity which it provides, the small cost per bit of stored information, the capability of random access, and the much better archival life [Fujitani 84], [Izawa 84]. Table 1 gives some comparative figures for optical disks and magnetic disks.

Table 1: Characteristics of Magnetic and Optical Disks

	User Capacity Mbytes	Access Time ms	Data Rate Mbits/s	Cost/Bit in cents	Archival Life in years
Magnetic Disk (IBM 3340)	70	35	7.0	10^{-4}	2-3
Optical Disk (Philips DOR)	1000	100-225	2.0	10^{-8}	>10
Jukebox (Philips)	128000	20secs if disk exchange required	2.0	10^{-9}	>10

Note1: Jukebox is a collection of 64 optical disks. It comes with one or two drives. The above figures are approximate and vary within members of the same family.

Note2: Prototypes and commercially available systems with much higher transfer rates (10 to 50 Mbits/s) exist [Kenney et al.79], and some experimental systems which use multiple light beams advocate 100 to 320 Mbits per second. On the other hand seek times are slower than seek times of magnetic disks [Fujitani 84].

The optical disk is organized in much the same way as the magnetic disk for reasons of compatibility ([Fujitani 84], [Maier 82], [Kenney et al 79], [Philips 84]). A laser beam from a fixed laser is used to read and write in the top of a metal film. Optics which is mounted on a sled which is moved from track to track by a linear motor is used to position the beam in the appropriate track. The major difference with the magnetic disk is that the recording into the optical disk is by ablating holes in the metal film. Another very significant difference from the performance point of view is that in some designs the optics can be used to move the beam to a nearby track in one revolution (from 1 to 50ms) without moving the entire optical assembly [Bell 83]. This is done using light deflectors with small inertia. We will call a *span* the set of tracks forwards from a given location of the optical disk head from which information can be read or written without a mechanical move of the disk head.

Current optical disk technology has the limitations of the write-once storage. However, this is not a problem for our particular application environment (document archival) since we do not expect a large number of updates. In some cases the write once restriction may even be desirable. The optical disk technology provides particularly attractive features for a document archiver. These are huge storage capacity, cheap storage per bit and long archival life. The random access of the optical disk is much slower than the random access of the magnetic disk. Although the random access times for optical disks will improve in the future, the inertia of the optical assembly system will limit the improvement (as in the case of the magnetic disks). Another significant problem is the increased contention in these devices which may result in increased responce times for the users due to queueing. The increased contention will be the result of the much higher storage capacity provided by a single device and the probable centralization of these devices within the organization. It is therefore important that we investigate methods for improving system performance. On the other hand the very high transfer rates possible and the span access capability are important performance differences of the optical with the magnetic disk and they should be exploited. For example very high transfer rates make sequential access methods like signature files much more desirable.

In this paper we present issues related to the architecture of a document archiving system using optical disk technology. We describe the issues and analyze some decisions which affect the performance of the system. In section 2 we describe document states; in section 3 we describe the function of the archiver, in section 4 we analyze the problem of document placement in the optical disk; in section

5 we discuss the results of a simulation; in section 6 we describe the problem of storage hierarchies and document migration in the magnetic disk;in section 7 we describe document copy in the on-line optical disk from an off-line disk; in section 8 we discuss version support, finally in section 9 we present a summary and current research.

2. Document States

In this section we describe the life of a document in a multimedia information system. Such a multimedia information system (MINOS) is under implementation

([Christodoulakis et al. 84], [Christodoulakis et al. 85]).

The progress of a document through the system is described by the document states. Document states are shown in Diagram 1.

Documents may arrive from another workstation or they may be created using an editor formator capability.

Arriving documents are documents which have been sent from another workstation and they have not been looked yet by the user. At some point in time they may be looked by the user using a *document browsing* capability. A user can only look at documents for which he has *authorization*. When the user finishes examining an arriving document he may archive it or he may edit it, or annotate it and then archive it. Since arriving documents are not in a stable state (in the sense that they will change state after they have been looked by the user, and this change of state may mean further editing or deletion) these documents are stored in a magnetic disk. This allows for fast document access as well as in-place update which again results in space savings and write time savings.

Documents in the *editing* state are edited, created or annotated. Document creation may require synthesis of information from other documents. Cut and paste techniques can be employed ([Christodoulakis et al. 84], [Christodoulakis et al. 85]). Since at a given point in time a user only edits a limited number of documents, access to documents in this state is done by name rather than by content. Version support, as well as recovery functions to guarantee that long work is not lost, are also necessary in this state. Documents in editing state may be frequently accessed and modified. Thus they should also reside in a magnetic disk storage unit.

Archived documents are documents which are in a stable state in the sense that the user who produced them or received them from the communication lines or inserted them in the system using a fascimile capability feels that the documents will remain unchanged for the forseable future. At this point he explicitly sends the documents to the archival subsystem. A unique identification number is assigned to any document inserted in the archiver. Document modifications (editing, annotation) are not allowed within the archiver. If at some point of time the user feels that there is a need for modifications of a document he has to extract the document into the editing subsystem. If he wants to replace the old document with the new one he gives to the system the identification number of the old document so that the system deletes it. He can then insert the new document in the archiver. Alternatively he may want to indicate that the new document is a version of the old document, but he may want to allow both documents to coexist in the archiver. He can do that by establishing a link between the two documents.

Documents in the archived state require different system support than documents in the arriving or editing state. Documents may remain in the archived state many years before they are discarded from the system. Due to their large number and age the users will not remember their name (unique

identification number). Thus content addressibility is required for these documents. Moreover, since the documents are in a stable state, they are appropriate to be stored in an optical disk.

In the remainder of this report we will concentrate on the archival subsystem. More details on the editing subsystem appear in [Christodoulakis et al. 85].

3. Document Archiver

The document archiver is the subsystem responsible for storage and retrieval of archived documents. We assume from now on that the archived documents are stored in an optical disk device. In this section we summarize the major functions of the document archiver. In the following sections performance issues related to the archiver will be discussed.

The document archiver supports the following functions:

1. Internal representation and presentation
2. Content addressibility
3. Access methods for content addressibility
4. Browsing Retrieval Interface
5. Information extraction capability
6. Version support
7. Authorization

The archiver should support an internal representation which reduces the storage requirements and the retrieval costs and at the same time it makes sure that the documents maintain their original appearance. This is done by a mapping software which reconstructs the documents from their internal representation when the documents are presented to the user [Christodoulakis et al. 84].

Content addressibility in multimedia documents is achieved by allowing the user to ask queries on the content of documents (e.g. text, attributes, images) as well as on some presentation aspects. Since users may not be able to specify precisely which document they want the retrieval interface supports a browsing capability with which the user can browse through documents or within a document until he finds the desired information. Once he finds this information he can extract it so that he may use it for document preparation of new multimedia documents. Extraction of parts of multimedia documents (pictures, graphs, text sections...) and document formation is done outside the document archiver [Christodoulakis et al. 85].

The access method that the archiver uses is based on signatures of the document contents (attributes, text, images) [Christodoulakis and Faloutsos 84]. Signatures have the advantage that they require a small storage overhead (up to 10% of file size for text signatures) and they allow very easy insertion of new documents. With the advent of cheaper main memories in the future, signatures of on-line documents may be kept main memory resident. With current technology it is more likely that signatures of the on-line optical disk will be kept in the optical disk itself or in a collection of magnetic disks.

When signatures are kept in an optical disk they present an additional advantage: insertion of new documents is very easy since the signature of the new document is just appended at the end of the signature file. In contrast insertions of new documents when a tree structure is used to provide content addressibility may be difficult due to the fact that updates in the index structure may have to propagate upwards which is difficult for write once storage devices. The very high data transfer rates possible with optical disks are also helpful because the signature files are accessed sequentially.

4. Document Placement in the Optical Disk

We could allocate documents in the optical disk sequentially as they come in the archiving subsystem. We call this placement algorithm *Algorithm 0*. With each document we would associate some authorization information in the form of a capability vector. A compressed form of this vector could be appended at the beginning of the signature of every document so that additional false drops coming

from the fact that the user has no authorization for some of the qualifying documents are avoided. The advantage of this method comes from its simplicity. It does not require any directory maintenance but a simple pointer to the next available location in the optical disk. There are no problems of disk fragmentation. Finally since the documents are allocated sequentially as they come and their signatures are appended at the end of the signature file in the same order, there is no need for a directory for the signatures, and signatures of documents never have to be moved to make space for more signatures.

There are however some possible disadvantages associated with the above organization. The first is that the method does not exploit any clustering of documents in the optical disk. This may result in expensive seeks. We investigate this aspect in more detail later on. The second is that authorization information will have to be associated with every document and kept with the signature of the document in main memory. In addition a copy of the authorization information will have to be kept in the magnetic disk so that it is not lost in the case of a system breakdown. The reason that this information is not stored with every document in the optical disk is that the capability vector may have to change at some point in time (addition of a new user say).

Another disadvantage of the method is that it requires searching of the signatures of all documents in the optical disk in order to answer a given query. This requires more CPU time and more main memory to store all the signatures and the capability vectors. If enough memory does not exist then user queries will force the system to retrieve a portion of the signature file from the magnetic or the optical disk with additional CPU and IO overhead. Assuming that main memory becomes very cheap and that we associate enough main memory with the document archiver to store all the signatures and the authorization information as well as that we use some hardware for the search of signatures, the only remaining disadvantage may come from the longer seeks of the optical disk head for locating qualifying documents.

A variation of Algorithm 0 which may be used to achieve better clustering is the following: small size documents are stored as they come at the beginning of the optical disk. Large documents are stored together at the last tracks of the optical disk. The size separating small and large documents is a parameter of the system. We call this placement algorithm *Algorithm 01* The motivation behind this placement algorithm is that in order for the optical disk head to bypass a larger document which does not qualify a seek will have to be done. Thus by moving large documents to another place in the optical disk we expect to find several documents within a track and thus reduce the average seek cost. This intuitive explanation can be described more formally using properties of Schur concave functions [Christodoulakis 84b]. This variation however suffers also from the same disadvantages of the previous method (although it is expected to improve clustering).

A third approach is to divide the optical disk into a number of areas which correspond to some general files in which the documents can be categorized. With each file we could associate some authorization information. The separation into general files will be environment dependent and thus it will have to be decided within the organization. The advantages of the approach is that it exploits the clustering of information in the optical disk in order to avoid expensive seeks and that it does not replicate authorization information with every document. Finally since the user requests refer to a particular file the signature search is restricted. In addition, even if the signatures of all the documents in the optical disk do not fit in main memory some user requests will be completely examined using only signatures which exist in main memory. With some smart utilization of main memory (e.g. selection of the signature file to be replaced) it will be the case that many user requests will be completely satisfied without any secondary storage access. This indicates that the main memory that we have to associate with the server for storing signatures is less than the main memory that we need with the previous approach, which may lead to a less expensive system. If special purpose hardware for search is associated with fixed sizes of main memory this indicates further reductions in the system cost.

The disadvantage of the approach is additional complication. The optical disk space is now structured according to a directory which resides in main memory or a magnetic disk (since updates to the directory may be needed). To avoid fragmentation of the optical disk some algorithm should be used

to decide what happens when the contiguous space which has been allocated to a file in the optical disk is exhausted. There are several algorithms which can be used for this purpose (Algorithms 1 thru 8 below).

We would like to examine the effect of document placement in the optical disk on the system performance. We will ignore the signature search time and we will concentrate on the retrieval time from the optical disk. The problem presents similarities with the problems of compaction and fragmentation in file systems ([Teorey and Fry 82], [Claybrook 83]). There are however a number of differences between the two environments:

1. Environment Characteristics

There are no physical deletions in the optical disk. Thus there is no problem of utilizing the optical disk space again. There are no insertions in a particular order (or within a text file) which may force the rewriting of the file (as in an editor environment). In general file systems it is difficult to parametrize realistically those factors. The fact that these restrictions hold in our environment makes modelling and performance evaluation easier.

2. File Characteristics

In general file systems files may exhibit drastically different characteristics (e.g. files with high update to query ratio or vice versa, text files versus data files, and so on). Performance of clustering algorithms will be sensitive to file characteristics and thus it will be environment dependent. Again the document archiving environment is more uniform and easier to parametrize.

3. Query Characteristics

The ways in which files are accessed by the users are many (e.g. sequential scan versus random access, file dependencies like joins, ...). Our environment is a server type of environment using the same access method for all files. The signature access method allows for accessing the documents in the optical disk in sequence so that the access mechanism is not moved forth and back.

4. Performance Characteristics

The optical disk characteristics are different than the characteristics of magnetic disks. Random access time is considerably slower than magnetic disks, while the transfer rate can be very fast (10 to 50 Megabits per second or faster). This should have an effect on the performance of clustering algorithms. In addition some optical disk designs allow that data in a number of nearby tracks are accessed without a move of the access mechanism [Maier 82]. This should also affect the performance in the case that documents are clustered into files.

We have decided to examine the performance of a number of placement algorithms in this environment. In the remaining of this section we present these algorithms.

The first two algorithms allocate space for documents as they come without separating them in files.

Algorithm 0: Allocate space for documents as they come. This algorithm was described at the beginning of this section.

Algorithm 01: Allocate space for documents as they come. Keep small documents at the one end and large documents at the other end of the optical disk. This variation was also described at the beginning of this section.

All the other algorithms partition the optical disk space into a number of files. Each file is given originally a number of bytes according to what the user predicted based on his knowledge about the files. Let B_i, $i=1, \ldots M$ be the number of bytes that a file has been allocated originally. Some of these files may not expand as fast as the designer has predicted and some others may expand faster. When the space which was originally allocated to a file is exhausted, then an algorithm has to be used for deciding where to place an incoming document.

The following algorithms can be used to decide where a document is placed when the space which was originally allocated to the file where the document belongs is exhausted.

Algorithm 1: Find the nearest location in the optical disk which is empty and allocate the document there. (Look forwards and backwards).

- The disadvantage of the algorithm is that there may not be many empty positions at a time and the result will be that the documents will be spread out. In addition the performance of requests to the other files may deteriorate.

Algorithm 2: Find the file in the optical disk with the most available space and allocate the new document there.

- The disadvantage is that this place may be too far and the result may be long seeks.

Algorithm 3: Same as Algorithm 1 but split the space into two sections proportional to the rate of expansion of the two files.

Algorithm 4: Same as Algorithm 2 but split the space into two sections proportional to the rate of expansion of the two files.

Algorithm 5: Find the file in the optical disk with the minimum of the ratios $\frac{U_i}{B_i}$, where U_i is the number of bytes of file i which were used so far. Thus this minimum indicates the file which is not expanding as fast as it was predicted at the beginning.

Algorithm 6: Same as 5 but the space is divided among the two files.

Algorithm 7. Find the nearest file in the optical disk which has a value of $\frac{U_i}{B_i}$ smaller than the $\frac{1}{M}\sum_{i=1}^{M}\frac{U_i}{B_i}$. In other words the first file which is expanding slower than expected. Insert the document there.

Algorithm 8 Same as Algorithm 7 but the space of the file found is allocated proportionally to the rate of expansion of the two files.

The performance of these algorithms was examined using a simulation. The results of the simulation are described in the next section.

5. Simulation Results

In this section we describe the simulation used to examine the performance of the placement algorithms and we analyze the results. A document which is stored in the archiver has a length of B_D bytes The length B_D of documents may vary drastically from document to document. Documents without any bitmaps have a small length in general (somewhere between 500 bytes and a few Kbytes). Documents with bitmaps even after compression may be very long (from tens to hundreds or even thousands of Kbytes). We have used document lengths which are selected from two different distributions for short and long documents. The selection is done with probability P_S from the short documents and P_L for the long documents.

Documents are inserted in the system with frequencies p_i, $i=1, M$, and they are requested in user queries with frequencies q_i, $i=1, M$. A proportion of the documents in a given file qualify in a user query. Let N_i be the number of documents in file i. A number n_i of qualifying documents are randomly selected from the documents of the file i.

The cost model for retrieval is similar to the one described in [Maier 82] and it is based on the characteristics of the Philips optical disk [Philips 84]. An analogous model for evaluating performance of magnetic disks in a dedicated environment is presented in [Teorey and Fry 82]. As in this model the cost of moving the access mechanism depends on the distance traveled. The major difference is that if the optical mechanism is located in a given track, it can still read data from the T-1 tracks that follow by readjusting the optics, where T is the number of tracks in a span. The cost ss (for short seek) in

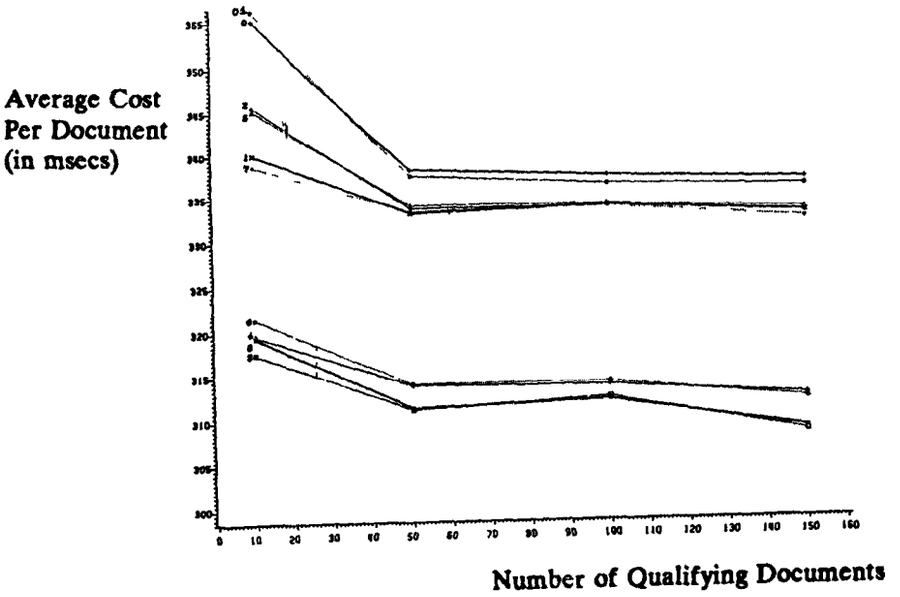

Figure 1: Performance of various clustering algorithms.

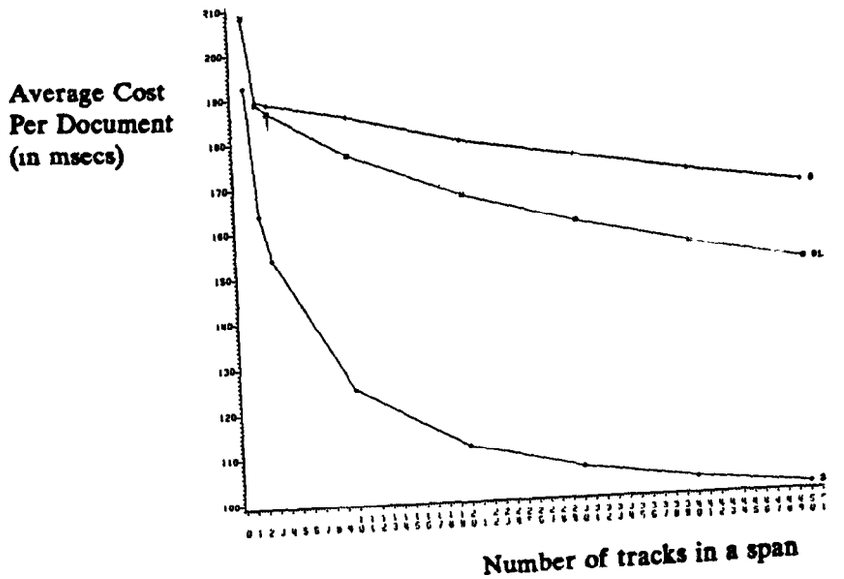

Figure 2: Effect of the number of blocks per space on performance.

this case is equal to the rotational delay. We assume that the pointers to qualifying documents are always sorted so that the access mechanism moves only in one direction.

Figures 1, 2, and 3 show results of the simulation. In figure 1 the performance of the various algorithms is compared. The simulation was run in the following way. Originally the optical disk space was divided among 20 files. The document insertion rates however were non-uniform and independent on the sizes allocated so that when the optical disk would be nearly full many documents of a certain file would have to be positioned into the space which was originally allocated to other files. We started measurements of the average cost per document retrieved (in ms) only after the optical disk was full of documents. Qualifying documents were randomly spread over the documents of a file.

All these conditions are unfavorable to the algorithms which separate the space of the optical disk into files. One reason is that in this environment a large percentage of the queries on a file will be submitted to the system while the optical disk is not full. In this case the documents of a file may all be in a single area. Thus the effect of spreading the documents of a file into other locations will not affect as much the average cost of queries. A second reason is that the person who will originally separate the optical disk space into files will have some idea on what the insertion rate is going to be in each of the files so that he will allocate space proportional to this rate. This suggests that the independence assumption is also unfavorable to algorithms that separate the disk space into files. Finally a third reason is that qualifying documents may not be uniformly spread over the documents of a file. The uniformity assumption may also lead in pessimistic estimates [Christodoulakis 84b].

The reason which we have run the simulation in such a way was that we were mainly interested to compare the performance of the algorithms that select a new placement for the incoming documents.

The results in figure 1 seem to show a clear distinction of the performance of the algorithms into three different classes. The first class contains the two algorithms which do not separate the disk space into files (Algorithms 0 and 01). The second class contains the algorithms which place a document in an empty location but they do not reserve more space (Algorithms 1,2,5, and 7). Finally the algorithms with the best performance are those that, when the file space is filled up, they split the space that has been allocated to another file (Algorithms 3,4,6,8). From the last set of algorithms the algorithms 3 and 8 present the best performance. Algorithm 3 is simpler to implement. Algoritms 0, 01, and 3 were used in the subsequent simulations.

Figure 2 shows the results of a simulation which was performed in order to examine the effect on placement algorithms of the capability to read more than one track at a time without moving the access mechanism (span access). This time the simulation was run so that concurrent insertions and queries were taking place. Only statistics on queries were kept. As expected the performance improves for larger values of the number T of blocks per span because of clustering. The improvement is better for the algorithms that separate the space into files because qualifying documents are located near each other. It may therefore be concluded from this experiment that placement algorithms which separate the documents of the archiver into files becomes more desirable when optical disks with larger number of tracks per span are used.

6. Storage Hierarchies-Migration in the Magnetic Disk

In this section we describe some storage hierarchy and file migration problems which appear in the document archiver environment.

Even if the write-only limitation of the optical disk did not exist, it is unlikely that the optical disk would completely replace the magnetic disk especially in systems where high performance is essential [Fujitani 84]. The reason is that the random access time of 100 to 500 milliseconds of the optical disk is too slow. In many high performance environments the two types of storage will coexist and complement each other. In addition, in our environment and with currently available technology, the existence of magnetic disks is essential since they will store documents in the editing or arriving state. There is therefore a four level storage hierarchy in this environment: main memory, magnetic disk, on-

line optical disk, off-line optical disk (jukebox).

The main memory will mainly be used for storing signatures of documents. If the main memory cannot store all the signatures of documents that exist in the optical disk then replacement algorithms have to be considered. The problem is similar to the problems examined in database environments. If signatures are clustered and the size of a cluster is fixed, then simple LRU algorithms will do. If signatures correspond to files in the optical disk then replacement algorithms will have to consider the signature length in the replacement as well as the time from the last reference.

Due to the faster access time of the magnetic disk it is desirable that some frequently accessed documents reside on the magnetic disk, thus forming a two level storage hierarchy with the on-line optical disk. Appropriate fetch and replacement algorithms will have to be selected. The problem presents many similarities to the problems in cache memories and in file migration [Smith 81]. In buffering data base blocks simple algorithms like LRU were found to work as well as any other algorithms when they work for fixed size blocks. Smith found however which in the file migration problem, algorithms that migrate files with the largest value of the product of the size times the interreference interval work better. Although our environment is not an editor environment like Smith's, these algorithms should also perform well. In [Smith 81b] ways to calculate the length of the interreference interval are presented.

Most commercial systems that use file migration techniques fetch files from the mass storage on reference or upon explicit user request. In addition new files are always placed first in the disk. This is reasonable for an editor environment where once the file has been fetched an activity period for modifications is expected. It is also reasonable as a buffering strategy for many data base files due to the locality of references in programs. This is not the case with our environment. The decision on where to first put a file, or if a referenced file in the optical disk should be moved in the magnetic disk replacing some other documents should be based on the values of the space times expected interreference interval for the document and the documents in the magnetic disk.

To calculate the expected interreference interval for documents some parameters should be kept and updated (at least periodically) with every document of the on-line optical disk. These parameters can be kept with the signatures of the documents if all signatures are main memory resident. They are updated immediadely when the signature of the document is found to qualify. When all signatures are not main memory resident this approach becomes more expensive because updates may imply block writes

An alternative is to keep main memory resident pairs of id numbers and associated parameters (including length) for every document. It seems however that this approach requires much main memory and CPU time. A second alternative is to keep current this information only for these documents which reside in the magnetic disk. These parameters could be directly used to indicate which documents are to be replaced from the magnetic disk. The decision of if a (referenced or incoming) document should be placed in the magnetic disk can be based on the size of the document and the *expected* interreference interval for the age of the document. All the documents in the archiver which have the same age are assumed to have the same interreference interval. The age of the document is recorded at the time that the document is inserted in the archiver Note that this structure can also be used to indicate if a document is located in the magnetic disk and where. This information is difficult to be stored with the signatures if they are not continuously main memory resident because it requires updates.

Documents in the magnetic disk which do not have a copy in the optical disk yet should be copied automatically when the server is idle. The copying can be facilitated by batching all writes of documents that refer to the same file. Thus the cost per document write will be reduced since some seeks will be avoided. Documents are selected for replacement if they have already a copy in the optical disk. The documents of the file with the maximum sum of the product of size times interreference interval are first copied in the optical disk.

7. Document Copy to the Online Optical Disk

A second storage hierarchy problem appears when the optical disk is full and it has to be moved into an off-line storage facility like the Philips jukebox. Should some portions of the optical disk be copied into the new on-line optical disk to speed up retrieval? Note that the problem is important since access times for the off-line optical disks are very slow. (For example for the Philips jukebox access time for an off-line optical disk is of the order of 20 seconds while access times for the on-line optical disks are between 100 and 500 milliseconds). Note also that the problem is not a file migration problem. Once some data is copied into the on-line optical disk it stays there for ever due to the write once property of the disk. They do not migrate forth and back.

The first question which we face is what is the unit of data that is copied into the on-line optical disk: files or documents. The advantage that the copy of files has is that the system will only have to look at the on-line data for all the documents of the file. In addition keeping statistics on the use requires only one location per file. It seems however that transfer of files to on-line storage may be far from optimal since the ages and projected uses of the documents of a given file may differ considerably. In addition files may be too large. This would imply that the disk would fill up faster and that a new copying process would have to take place. Thus we have decided for a scheme which copies only certain documents from a file.

Again there are two ways to select documents which are copied into the on-line disk: based on the expected use of the document or based on the date of insertion of the document in the archiver. The reason for considering date as the criterion is that it provides a heuristic method for selecting these documents which will probably be requested more frequently in the future (since in general the use of documents decreases with time) and second that the date of insertion could be exploited by the system to avoid the retrieval of documents from the off-line storage. For example the system could ask the user if he wants to see any documents before a given date. This would be profitable if the user had an idea about possible dates. However this method is not transparent to the user. In addition the archiver will be used for storing a large amount of information which a particular user may never see. In this environment it is less possible that the user will have an idea about dates.

The final method makes automatic selection of the documents to be copied to the on-line storage based on the projected document use and the document characteristics.

A selection algorithm is used to do the selection as described below. Only documents from active files (files which still accept insertions) are considered by the selection algorithm. The documents selected by the selection algorithm are copied to the on-line optical disk. The signature of the whole file (for all active files) remains on-line (in the magnetic disk or main memory). Within the signature the part which corresponds to the on-line and to the off-line documents of the same file are separated. The search of the signature of a file starts from the part which corresponds to on-line documents. With a careful selection of the documents which are copied to on-line optical disk the off-line part of the file will not have to be accessed. We believe that this approach has some advantages over the two other approaches that we described before. In the following of this section we describe this approach in more detail

To decide which documents are to be copied to the on-line optical disk we examine what happens when the on-line optical disk is filled up and has to be moved off-line. Consider a document i. The cost of not copying the document to the on-line disk is the very expensive accesses to the off-line storage for all the remaining queries which result in the retrieval of the document. In estimating the cost of copying back a document in the on-line optical disk we will ignore the additional storage cost (although the cost equations can be easily modified to take this cost into account). The reason is that optical disk storage is very cheap (4 or 5 orders of magnitude cheaper than the magnetic disk). The cost of copying the document i in the on-line optical disk is that the on-line disk will be filled up earlier and thus all the documents of the optical disk will have to be moved off-line with the result of more off-line block accesses for all the other documents.

The above reasoning is approximate for two reasons. The first reason is that it does not consider the fact that when the on-line disk is moved off-line some other documents will be moved again to the new on-line optical disk (if the file is still active). The second reason is that it considers a particular document independently on all other documents. This would be correct if queries were asking for a single document from the file. However, when more than one document from the off-line part of a file qualify the cost of accessing the off-line storage is shared among the qualifying documents since the selection of the particular off-line disk from the jukebox, its move into the drive, and the warming up is done only once. A more detailed analysis of the problem would have to consider the probability of accessing the off-line part of the file for a given selectivity S. This probability however changes as new documents are added to the on-line part of the file and it is difficult to calculate reliably. Our model has assumed that the probability that more than one document of the off-line part of a file qualify in the same query is zero (which would be true if a good selection of the documents to be copied was done). This assumption of the model will tend to result in copying more documents into the on-line optical disk than necessary.

Parameter Description

CNC = cost to not copy the document to the on-line optical disk.

CC = cost to copy the document to the on-line optical disk.

RM_i = remaining references (user queries) for document i at the time of the decision. This number can be calculated for the interreference time probability distribution from document i.

$size_i$ = size in bytes of document i.

$ocost(x)$ = cost of accessing x bytes of data from the off-line optical disk.

$lcost(x)$ = cost of accessing x bytes of data from the on-line optical disk.

$\bar{l}$ = average number of bytes per document.

q = average number of queries per second to the data of the on-line disk when it is near full.

p = average number of insertions of new documents per second.

The two costs can be expressed as

$$CNC = RM_i * ocost(size_i)$$

$$CC = RM_i * lcost(size_i) + \frac{size_i}{\bar{l} * p} q * ocost(\bar{l})$$

and the decision to copy a document in the on-line disk is taken if $CC - CNC < 0$. Or

$$RM_i * lcost(size_i) + \frac{size_i}{\bar{l} * p} q * ocost(\bar{l}) - RM_i * ocost(size_i) < 0$$

or

$$\frac{RM_i}{\frac{size_i}{\bar{l}*p} q} * \frac{(lcost(size_i) - ocost(size_i))}{ocost(\bar{l})} + 1 < 0$$

This equation indicates that it is more desirable to copy a document in the on-line storage as the number of remaining requests to the document increases, and as the difference of the costs of accessing a number of bytes from off-line storage minus the cost of accessing a set of bytes from on-line storage increases. The profitability of copying a document to on-line optical disk decreases as the ratio of the size of the document to file average document size increases and as the ratio of the queries to insertions increases.

There is an interesting explanation of the ratio $\frac{RM_i}{\frac{size_i}{\bar{l}*p}q}$. Since $\frac{size_i}{\bar{l} * p}$ is the average time required to fill up the space occupied by document i the product $\frac{size_i}{\bar{l} * p} q$ is the average number of queries on the documents of the on-line optical disk that will be executed until the space occupied by file i is covered by new documents.

Thus the ratio $\frac{RM_i}{\frac{size_i}{\bar{l}*p}q}$ indicates the ratio of the total remaining requests to the file i divided by the average requests that will be executed until the space occupied by file i is covered by incoming documents.

The above arguments apply to a two drive architecture where the one drive is used for the on-line optical disk and the other for one of the off-line optical disks. A three drive architecture may be desirable in such a system. An optical disk in this architecture is moved off-line when both on-line optical disks are full. At replacement time a decision is taken if certain data is going to be copied. The same equations still apply with the understanding that RM_i refer to the remaining queries asking for document i when its optical disk is to be moved off-line, and q is the query frequency on the documents of the optical disk which is to be moved off-line.

The analysis presented above is useful because it gives an intuition on which are the documents which should be copied back into the on-line optical disk. From the practical point of view a limitation of the method is that some parameters should be kept and updated (at least periodically) with every document of the archiver so that RM_i can be computed. A tradeoff is to keep this information current only for documents which are frequently accessed and therefore they reside in the magnetic disk as was described in the previous section.

8. Version Support and Duplication Control

The main reason for versions in a document archiver environment will be for putting some private comments at the top of a document (annotation) or to show that a document has been produced from another with (minor) modifications. This suggests a version support method based on a tree organization of versions. The users will be able to traverse the tree up and down if they are allowed to do that based on their authorization. An example where a version may not be visible to all users is when a user puts his own annotation to a document and he does not want the other users to see the annotation.

A unique document identifier is associated with each document. This unique identifier can be the address of the document within the optical disk for documents which have already a copy in the optical disk. This will allow faster access to documents. Users may use this identifier in order to indicate to the system to which document a new document is a version. Such an implementation would store information about versions separately in tree structures and use the document identifier to locate in which tree a particular document exists. (This requires another tree structure with the identifiers of all documents which belong to one of the version trees.) Since this information requires updates it is kept in the magnetic disk. We expect that the version tree structures will be very shallow.

An analysis of the system performance in the presence of version support mechanisms has to examine two questions: How much storage overhead is involved, and how efficiently the system answers questions on most recent versions. The simplest implementation is to store the whole document for each new version. Using this method user queries on recent versions will be answered very efficiently since the whole document is in one place on the optical disk. However, the method may require excessive storage waste since some documents may be very long (several Megabytes) and the differences of the two versions minimal (a few lines of annotation say). On the other hand if the new version of a document only stores the changes, an extra seek is required to access the data from the original

document.

We note here that the problem of controlling the amount of duplication of data is not only related to version control. In the document archiver environment documents may be synthesized from some other documents which existed in the document archiver and the user may not link them as versions. In this case (which is not typical of traditional data base applications) a large amount of data duplication may take place, especially if bit maps of pictures are duplicated. The two problems, version control and duplication control however result in the same performance problem which is analyzed below.

Previous approaches to the problem of version control have used differential files [Severance and Lohman 77], [Rochkind 75], negative differential files [Katz and Lohman 84] and shadow pages [Lorie 77] (although not explicitly proposed for version control). Our approach is different. We will allow duplication for relatively small documents or parts of documents since optical disk storage is much cheaper than magnetic disk storage while random access is more expensive. The decision on what to duplicate depends on system parameters. If we draw the analogue of records with documents, our approach is different in that the granularity of duplication is different (in the case of records the whole record stays or the whole record is duplicated), and in that the decision on what to duplicate depends on usage characteristics (instead of fixed size pages). Note that a document may start from the middle of a track and occupy several tracks in the optical disk. In the following we will assume that every piece of information within the archiver has a unique address in the optical disk associated with it.

We now examine the tradeoffs involved in the decision of what information to duplicate. Since the cost of storage in the optical disk is very small (several orders of magnitude less than the storage cost in the magnetic disk) we will omit it in the following. The cost equations that we give below however can be modified to accommodate it. Suppose that we do not copy a piece of information from a parent version to a son version. The extra cost involved is that all the times that the son version is going to be accessed it will require an extra seek to find this information from the parent version. If this information is copied in the son version the result will be that some space in the optical disk will not be available for storing incoming documents. As a result the optical disk will have to be moved off-line faster than if the duplication had not taken place. Thus some queries that would normally be served from an on-line optical disk will be served by an off-line one. Table 2 describes the system parameters.

Table 2: Parameter Description

CND = cost to not duplicate the part of the document with the new version.

CD = cost to duplicate the part of the document with the new version.

$\bar{T}$ = total average number of requests expected for any document coming into the archiver.

d = number of bytes to be duplicated (common in the two versions).

$ocost(x)$ = cost of accessing x bytes of data from the off-line optical disk.

$lcost(x)$ = cost of accessing x bytes of data from the on-line optical disk.

$\bar{l}$ = average number of bytes per document.

q = average number of queries per second to the data of the on-line disk when it is near full.

p = average number of insertions of new documents per second.

Thus $CD = \frac{d}{\bar{l} * p} * q * ocost(\bar{l})$ where $\frac{d}{\bar{l}*p}$ is the expected number of seconds that the optical disk will be filled up faster than if duplication would not take place, and $\frac{d}{\bar{l}*p}*q$ is the expected number of queries over all documents of the archiver in this time interval. During this time all queries would have to be served from the off-line storage. We also have that

$$CND = \bar{T} * lcost(d)$$

And the document is to be duplicated if $\frac{d}{\bar{l} * p} * q * ocost(\bar{l}) - \bar{T} * lcost(d) < 0$, or if

$$\frac{\bar{l} * p * \bar{T}}{d * q} \frac{lcost(\bar{l})}{ocost(d)} > 1$$

Thus the profitability of duplication increases as the total expected number of requests to a document $\bar{T}$ is greater and as the ratio p/q of insertions to queries is greater, and it decreases as the ratio $\frac{d}{\bar{l}}$ of the number of bytes to be copied to the average document length increases and the ratio $\frac{ocost(d)}{lcost(\bar{l})}$ of the off-line to on-line access increases.

As an example, for an environment where $\frac{p}{q} = \frac{1}{2}$, $\frac{ocost(x)}{lcost(x)} = 40$ and $\bar{T} = 20$ the decision to duplicate becomes $\frac{\bar{l}}{d} * \frac{1}{2} * 20 * \frac{1}{40} > 1$, or duplicate if $d \leq \frac{\bar{l}}{4}$. Since we expect that in many office environments the distribution of document lengths will be bimodal with a high peak at 1 to 4 kbytes and a flatter and smaller peak at 50-100 Kbytes (for documents with bitmaps), this criterion would imply that documents with bitmaps are almost never duplicated.

Having answered above the question of when to duplicate, we must still examine how to keep track of where segments of where segments of different versions of documents are located.

The question of how is a more difficult one. We will have to keep track of this information within the document descriptor of the documents. The document descriptor as we mentioned before makes the mapping from the internal representation to presentation. If the internal representation of the text part occupies a number of consecutive bytes, then a pointer to the beginning of the text and a count of the number of bytes is enough. This scheme will have to be replaced with a set of pointers and counts to the various places that the data is stored. This is done within the multimedia document editor. Some parts of documents (say images or documents which are only annotated) may stay completely unchanged in the new document. Thus a single pair of (pointer, count) will be enough. More details on the implementation aspects of the multimedia documents and the way that MINOS deals with the problem of duplication control appears in [Christodoulakis et al. 85].

9. Summary and Further Issues

In this paper we have described and analyzed several problems related to the architecture of a multimedia document archiver using optical disk technology. Simulation results show that clustering into smaller files may considerably improve performance especially when the optical disk supports a span access capability. The improvement is better for larger selectivities and span sizes. We presented simulation results on the performance of algorithms which deside where to place a new document when the space allocated originally to its file is exhausted.

The contention in the optical disk which may result in increased user response times can be reduced by combining (ORing) the queries of more than one users. The combined request is examined against the signatures and qualifying documents are retrieved from the optical disk. (The order of signatures in the signature files is the same with the order of documents in the optical disk.) The documents which are retrieved from the optical disk are examined against the user queries and they are sent to the appropriate user (or users if the same document qualifyies in more than one query). This way the average response time per user decreases because the user does not have to wait until the system finishes examining the request of another user. Due to the larger selectivity of the combined query clustering also improves. Finally the access mechanism moves in one direction for longer periods at a time.

Qualitative and analytical results were presented for some storage hierarchy and migration problems which appear in this environment. Migration problems are different due to the write once property of the optical disk. These results give some insight into the tradeoffs which appear in this environment. Requirements and methods for version support were described and an analysis of the information duplication problem was presented. The organization of MINOS, a prototype information system which supports archiving and editing facilities for multimedia documents were briefly outlined. More details appear in ([Christodoulakis et al. 84], [Christodoulakis et al. 85]).

There are several more performance issues of importance which we are currently investigating in this environment. We have assumed in our presentation that documents are placed one after the other so that they make a maximum use of space in the optical disk. This approach may result in more seeks than necessary because even small documents may cross the boundaries of two tracks. An alternative is to decide where to place a document based on the amount of space that will be left empty if the document is moved to the next track. Of course this decision also depends on the document size because the move may result in another track boundary crossing. Analysis and experimentation is necessary to determine disk utilization and performance improvements. In a more complicated scheme the space left empty by the move of a document to a different block may be (partially) utilized by a new incoming document. Alternatively, in order to avoid pointers, only the last few blocks may be looked to find if there is enough space to fit the document. This can be done without additional accesses due to the span access capability. This scheme requires more complicated software but it may result in better space utilization and better response times.

In yet another scheme, document placement in the optical disk is done in batches from the documents of the file which were first inserted in the magnetic disk. This is done so that the write cost is minimized. (We assume that some space may be left unused to avoid track boundary crossing.) In this case the order of placement of the documents of the batch on the optical disk will have to be determined so that the number of times that documents of the batch cross track boundaries is minimized. In addition clustering may improve with appropriate rearrangment of the documents within the batch [Christodoulakis 84b]. This will result in a reduction of the average response time.

Finally analytic formulae of the cost will have to be developed. These estimates may be used for design decisions in particular environments. We have provided so far several exact and approximate analytic estimates of the expected system cost as function of the system and document parameters [Christodoulakis 85a].

In addition to the issues of data placement and storage hierarchies, issues of providing efficient access methods in this environment are also very important. We have studied so far a variety of signature access methods ([Tsichritzis and Christodoulakis 83], [Christodoulakis and Faloutsos 84], [Faloutsos and Christodoulakis 84], [Faloutsos 85], [Christodoulakis and Elles 85]). Further research in this area will have to consider clustering techniques for signatures and hardware implementation of signatures.

References

[Bell 83] Bell, A : "Critical Issues in High Density Magnetic and Optical Storage", Proceedings of the SPIE vol. 382, Optical Data Storage, Jan. 1983.

[Christodoulakis 84a] Christodoulakis, S.: "Framework for the Development of a Mixed-Mode Message System", Proceedings ACM-BCS Symposium on Research and Development in Information Retrieval, Camidge, England, 1984.

[Christodoulakis 84b] Christodoulakis, S.: "Implications of Certain Assumptions in Data Base Performance Evaluation", ACM TODS, June 1984.

[Christodoulakis and Faloutsos 84] Christodoulakis, S. and Faloutsos, C.: "Performance Analysis of a Message File Server", IEEE Transactions on Software Engineering, March 1984.

[Christodoulakis et al. 84] Christodoulakis, S., Vanderook, J., Li, J., Li, T., Wan, S., Wong, Y., Papa, M., and Bertino, E.: "Development of a Multimedia Information System for an Office Environment", Proceedings VLDB 84, Singapore, 1984.

[Christodoulakis et al 85] Christodoulakis, S., Theodoridou, M., Papa, M: "Multimedia Document Presentation, Information Extraction and Document Formation in MINOS", submitted for publication, 1985.

[Christodoulakis 85] Christodoulakis, S.: "Office Filing", in Office Information Systems, D. Tsichritzis editor, 1985.

[Christodoulakis 85a] Christodoulakis, S:, "Performance Analysis of a Document Archiver", submitted for publication, 1985.

[Christodoulakis and Elles 85] Christodoulakis, S., and Elles, K.: "Similarity Retrieval of Images in a Multimedia Archiver", in preparation, 1985.

[Clayook 83] Clayook, B.,G.: "File Management Techniques", Wiley, 1983.

[Faloutsos 85] Faloutsos, C.: "Design and Performance Comparisons of some Signature Extraction Methods", Proceedings ACM SIGMOD 85 (this issue).

[Faloutsos and Christodoulakis 84] Faloutsos, C. amd Christodoulakis, S.: "Signature Files: An Access Method for Documents and its Analytic Performance Evaluation", ACM TOOIS, Oct. 1984.

[Fujitani 84] L. Fujitani: "Laser Optical Disk: The coming revolution in on-line storage", CACM 27,6, June 1984, pp.546-554.

[Izawa 84] Izawa, K.: "Document Image Filing System Utilizing Optical Disk Memories", IEEE Database Engineering 7,3, 1983, pp.3-7.

[Katz and Lehman 84] Katz, R., and Lehman, T: "Database Support for Versions and Alternatives of Large Design Files", IEEE Transactions on Software Engineering 10,2, 1984, pp.191-200.

[Kenney et al 79] Kenney, C., Lou, D., McFarlane, R., Chou, A., Nadah, J., Kohler, T., Wanger, J., Zernike, F.: "An Optical Disk Replaces 25 mag tapes", IEEE Spectrum, Fe. 1979, pp.33,38.

[Lorie 77] Lorie, R.A.: "Physical Integrity in a Large Segmented Database", ACM TODS 2, 1977.

[Maier 82] Maier, D.: "Using Write-Once Memory for Database Storage", Proc. ACM PODS 1982, pp.239-264.

[Smith 81a] Smith, A.J.: "Optimization of I/O Systems by Cache Disks and File Migration. A Summary", Performance Evaluation 1, 1981, pp.249-262.

[Smith 81b] Smith A.J.: "Long Term File Migration: Development and Evaluation of Algorithms", CACM 24,8, 1981.

[Philips 84] "Megadoc .. An effective reply to the information flood", systems description, Philips, 1984.

[Rochkind 75] Rochkind, M.: "The source Code Control Systems", IEEE Transactions on Software Engineering, SE-1, Dec. 1975.

[Severene and Lohman 76] Severene, D. and Lohman, G.: "Differential Files: Their Application to the Maintenance of Large Databases", ACM TODS 1, 1976.

[Teorey and Fry 82] Teorey, T. and Fry, J.: "Design of Database Structures", Prentice Hall, 1982.

[Tsichritzis and Christodoulakis 83] Tsichritzis, D. and Christodoulakis, S.: "Message Files", ACM TOOIS 1,1, 1983.

[Tsichritzis et al 83] Tsichritzis, D., Christodoulakis, S., Economopoulos, P., Faloutsos, C., Lee, A., Lee, D., Vanderoek, J., and Woo, C.: "A Multimedia Office Filing System", Proceedings VLDB 83, Florence, Italy, 1983.

[Wiederhold 77] Wiederhold, G.: "Database Design", McGraw-Hill, 1977.