File Organizations and Access Methods for CLV Optical Discs

Stavros Christodoulakis

Daniel Alexander Ford

Department of Computer Science
University of Waterloo,
Waterloo, Ontario, N2L 3G1

ABSTRACT

A large and important class of optical disc technology are CLV format discs such as CD ROM and WORM. In this paper, we examine the issues related to the implementation and performance of several different file organizations on CLV format optical discs such as CD ROM and WORM. The organizations examined are based on hashing and trees.

The CLV recording scheme is shown to be a good environment for efficiently implementing hashing. Single seek access and storage utilization levels approaching 100% can be achieved for CD ROM's. It is shown that a B-tree organization is not a good choice for WORM discs (both CAV and CLV), but a modified ISAM approach can be appropriate for WORM discs. We describe clustered BIM's, a class of tree organizations appropriate for CD ROMS. Expressions for the expected retrieval performance of both hashing and trees are also given.

The paper concludes by outlining recent results and future directions on buffered implementations of access methods for WORM discs, as well as advantages of signature based access methods for text retrieval in WORM disc architectures.

1. Introduction

Continuing progress in the development of optical disc technology is resulting in ever increasing capacities and lower costs for direct access secondary storage. This new availability of large inexpensive storage is fueling the development of more ambitious and demanding applications such as multimedia data bases [4, 6, 12], which until recently have not been technically nor financially feasible.

Along with new applications, the advance of optical disc technology is spawning a need to understand the issues involved in providing efficient file organizations and access methods for the new storage mediums it is producing. Optical discs are similar in nature to conventional magnetic discs, but some types have differences that significantly affect the efficiency and feasibility of implementing conventional file organizations such as hashing, B-trees and ISAM. For example, the format used to organize storage space on the surface of an optical disc can be different from that used on magnetic discs. Many optical discs use a CLV (Constant Linear Velocity) recording scheme rather than the CAV (Constant Angular Velocity) scheme used on virtually all magnetic discs. Also, at the current level of technology, most generally available optical discs are not erasable. Two common examples of optical discs that combine both of these characteristics are CD ROM (Compact Disc Read Only Memory) and CLV WORM (Write Once Read Many times) discs.

CLV optical discs have the advantage over their CAV cousins of maximizing, within the limits of the recording technology, the utilization of a disc's recording surface. For the same size disc platter, more data can be stored on a CLV disc than a CAV disc. This is achieved by using a uniform recording density throughout the disc and varying the speed at which the disc rotates to ensure that all recordings, regardless of their position on the disc surface, pass beneath the sense mechanism at the same rate. The data recordings on a CLV disc are usually laid down in a single spiral pattern.

The disadvantage of this approach is that it makes movements of the access mechanism or seeks on CLV discs slightly slower than on CAV discs. Reading the data stored on a CLV format disc requires the speed at which the disc platter rotates to accurately match the position of the sense mechanism, but determining the position with the required precision is difficult without first being able to read the identification information stored in each disc sector. This "chicken and egg" problem is solved in a time consuming but effective fashion by reading the identification information as the access mechanism is moved incrementally across the surface of the disc, adjusting the rotation rate to match. Other delays come from problems of determining the exact position of a track, as adjustments might be required for proper alignment. The details of the techniques used by individual optical disc drive manufacturers are proprietary information and generally not available.

For most optical discs, the seek time is generally much larger than the disc's rotational latency. Though some WORM discs are faster, delays of up to one second or more are possible with the current CD ROM technology. As a result, the seek time is usually the most important factor in the time required to answer a query.

The issue of optical disc seek performance is complicated however, by the existence of some optical disc drives that are capable of reading from more than one track without performing a seek. The viewing mechanism (in the access mechanism) of such a drive is equipped with an adjustable mirror that allows slight deflections of the beam of laser light used to scan the disc. This enables it to be aimed at any one of a small set of the spirally concentric tracks immediately beneath the viewing mechanism. This set of tracks is called a *span* and the number of tracks in the set is called the *span size* [5]. Typical values of the span size are 10, 20 and 40 tracks. This span access capability can be likened to cylinders on magnetic disc packs except that for optical discs the sets of tracks or sectors in a span can be overlapping. It is provided by manufacturers of both CAV and CLV disc drives (although currently, it is more frequently encountered with CAV drives). For example, the Alcatel Thomson GIGADISC GD-1001, which uses the CAV format, has a span size of 40 tracks and an access time for tracks within the span of 5 milliseconds. For tracks outside the span, the access time is 200 milliseconds.

Neither CD ROM nor WORM discs allow the information stored on them to be altered. Information on CD ROM discs is registered by a physical pressing process similar to that used to create audio LP's. A master disc is created using a photochemical etching process from which other pressing masters can be made. The pressing masters are then used to imprint the CD ROM disc with the desired data recording patterns. The recording surface is given a thin coating of aluminum to make it reflective and is then covered with a protective plastic layer.

On a WORM disc, data is recorded on the disc surface by way of permanent reflective changes made by a laser beam. WORM discs are manufactured and received by the user in a blank state; information is then permanently registered on them in an incremental sector-by-sector manner at the user site. To improve error rates, error detection and correction information is incorporated within a sector when it is written. That makes it impossible for an application to erase, reuse or even add to data that has been previously stored on a WORM disc, as any attempt to write over a previously written sector (if the drive will allow such an operation) will make the sector contents inconsistent with the error detection and correction information. This, in turn, will cause an error to be reported during later retrievals of the rewritten sector(s).

In this paper, we examine the efficiency, feasibility and possible implementation strategies of providing some of the conventional file organizations and access methods used on CAV format magnetic discs. We also give approximate expressions for the expected retrieval performance of these organizations when implemented on CLV optical discs.

In the next section of this paper, we briefly present a model of CLV format optical discs. In section 3, we examine the implementation and performance of the hashing file access method for the CLV format. The implications for retrieval performance and space utilization of using pointer linked tree file organizations, such as B-trees and ISAM (Indexed Sequential Access Method), are discussed in sections 4 and 5 of the paper; particular attention is paid to the difficulties of using such structures on WORM type optical discs.

2. A Model and a Schedule

In this section we present an abstract model [5, 7, 8] of optical discs upon which our retrieval performance analysis is based and give a physical description of CLV discs. We also present an optimal schedule for answering a query.

An optical disc is a device composed of T ordered tracks, an access mechanism and a viewing mechanism. A track is represented by its sequence number i within the tracks of the device, $i=1,2,\ldots,T$ (to avoid discussion about boundary conditions, we assume that track numbers are extended above T and below 1). Each track is composed of several sectors (or blocks). The access mechanism can be positioned at any track. When the access mechanism is positioned at a certain track i, the device can read data that exist completely within Q consecutive tracks (track i is one of them). To do that, the viewing mechanism *focuses* to a particular track with qualifying data (within the Q tracks). We call the Q consecutive tracks a *span* and this capability of optical discs *span access capability*. An *anchor point* a of a span, is the smallest track number within a span. The largest track number within this span is $a + Q - 1$. The anchor point of a span completely defines the tracks of the span. A span can therefore be described by its anchor point.

A number N of objects (or records) may qualify in a query. N is called the *object selectivity* (or record selectivity) of the query. For optical discs, the number of times that the access mechanism has to be moved for accessing the data that qualifies in a query is called the *span selectivity* of the query. Therefore, a first approximation of the cost of evaluating a query is given by the span selectivity of the query.

When the access mechanism is moved, a seek cost and a rotational delay cost are incurred, as for magnetic discs. Transferring a track of data from the device involves a track transfer cost, as is also the case for magnetic discs. When more than one track within a given span has to be transferred to main memory, the access mechanism does not have to be moved, as we mentioned before. However, there is a small additional delay (about five milliseconds) involved for focusing the viewing mechanism on each additional track (within a span) that has to be read. We call this delay a viewing cost and will ignore it for the remainder of this paper because it is much smaller than the seek cost. This model reduces to a model of magnetic discs when the number of tracks in a span is one and the track size is equal to the cylinder size.

A spiral scheme is generally used on CLV optical discs for the storage of data, rather than the concentric rings found on CAV discs. The spiral consists of one long physical track of data recordings that begins at a distance from the centre of the disc called the *principal radius*, and continues until close to the outer edge of the disc, where it terminates. The principal radius is defined by the centre of the disc and the start point of the track.

A track, for purposes of our analysis, starts from the intersection of the spiral with the radial line that starts at the centre of the disc and passes through the beginning of the physical spiral. The track ends at the next intersection with this radial line.

A scheduling algorithm for retrieving the data that qualify in a query is an ordered sequence of anchor points that define the spans used for answering the query (and their order). Observe that there are many different sets of anchor points (and sequences of anchor points/schedules) that can be used for answering a given query. The span selectivity of the query (as well as the distance travelled by the access mechanism) depends on the scheduling algorithm used for answering the query.

In the following, we show optimal criteria for a scheduling algorithm when retrieving data in this model. In the algorithm, the access mechanism moves continuously in one direction. The criteria minimizes the number of times that the access mechanism is moved (span selectivity), as well as the total distance travelled by the access mechanism.

Theorem: {Optimal Scheduling}

The number of spans required to access all qualifying objects in a query is minimized if:

A. Spans do not overlap and

B. The anchor point is always positioned on a track with sectors containing qualifying objects.

In addition, the total distance travelled by the access mechanism is also minimized (within one span length) if the access mechanism is moved so that in addition to the conditions A and B, the anchor points $a_1, \ldots, a_s$ of the schedule satisfy $a_1 \leq a_2 \cdots \leq a_s$.

Proof: found in [5].

The theorem shows that if a schedule satisfies the conditions A and B, it results in a minimum number of spans. Note that for magnetic discs, the number of moves of the access mechanism is constant and equal to the number of tracks which contain the qualifying records. This not the case with optical discs where the number of moves of the access mechanism (long seeks) depends on the order of the data retrieval (schedule).

This theorem has an immediate implication on the data structures that are appropriate for optical discs. It suggests that organizations that are appropriate for optical discs should have unidirectional pointers to avoid additional block accesses. There for, organizations such as multilists are not appropriate for optical discs.

In [8], exact and approximate results have been derived for retrievals from CLV optical discs that follow optimal schedules. It was shown that the conventional CAV solution for the expected number of spans required for the retrieval of objects from a CAV disc, as found in [5], was a good approximation to the CLV solution. This is true for both cases where objects are and are not restricted from crossing sector boundaries.

The expected number of spans $\overline{K}$ is given by:

$$\overline{K}(\overline{B}) = \frac{\overline{B}}{1 + \frac{Q-1}{T-1}(\overline{B}-1)}$$

Here $\overline{B}$ is the expected number of tracks to be retrieved. It is calculated differently for non-crossing and crossing objects.

The approximate expected number of tracks for non-crossing objects where: t is the number of tracks in the file, n is the number of objects to be retrieved and c is the track capacity, is given by :

$$\overline{B} = b(t,n,c) = t\left(1 - \frac{\binom{(tc-n)}{n}}{\binom{tc}{n}}\right)$$

For crossing objects, the same formula for spans is used but the approximate expected number of tracks is calculated differently.

$$\overline{B} = \overline{B}_c + b(T-\overline{B}_c, N\left(1-\frac{T-1}{TC}\right)\frac{T-\overline{B}_c}{T}, C)$$

where

$$\overline{B}_c = \frac{N}{TC}[(T-1)\,(2-\frac{N}{TC}) + 1]$$

It is shown in [8] that the relative placement of files on the platter of CLV discs may have significant impact on retrieval performance.

3. Hashing on CLV Format Optical Discs

As a primary access mechanism, hashing has the ability of directly determining the location of a selected record without the need to perform disc accesses to consult file indices. This is of particular advantage on CLV optical discs, as they typically require relatively long periods of time to reposition their access mechanisms. By eliminating the seeks required to consult indices, hashing offers the possibility of a considerable speed advantage over other file organizations.

This advantage will only be present if bucket overflow can be reduced or eliminated. While the slow seek times of CLV optical discs make hashing attractive as a file access method, they also increase the cost of using overflow chaining to handle the inevitable bucket overflows (or the cost of any other collision resolution method for that matter that requires the access mechanism to move). The process of following a list of pointers from overflow block to overflow block across the surface of the disc could cause considerable delay in resolving a query.

The reason bucket overflow occurs in conventional implementations is that a bucket is usually associated with a physical division of the storage device, either a disc track or cylinder. This association allows the entire bucket to be retrieved with a single movement of the access mechanism. When the storage capacity of the physical division is exhausted however, bucket overflow occurs and additional seeks are required to complete the retrieval of the bucket.

On a CLV format optical disc, the sectors are usually arranged to form a single spiral track. This characteristic allows a hashing implementation to avoid bucket overflow by varying the physical capacity of buckets to match the number of assigned objects. Any number of

consecutive disc sectors (up to the capacity of the disc) can be allocated, and because they are arranged in a spiral, they can all be retrieved with a single seek (following the spiral does not incur a seek cost).

If all the data of a file is available for preprocessing before the file is written on the disc, as is the case for CD ROM, bucket overflow can be completely eliminated. Tracks can be expanded to accommodate any number of records assigned to a bucket so that all records can be allocated to sequential disc sectors and read with one disc access. Overflow can also be eliminated for WORM discs, but at the expense of increased disc space consumption caused by the duplication of records needed to restore sequentiality after a change to a file [9].

Note that it is still convenient for the purposes of analysis to use the CAV definition of a track as an approximation to the CLV case. As mentioned above, it has been shown [8] that expected retrieval performance from a CAV disc is close to the performance of the CLV disc for small files and record sizes. We will use this result in the analysis given below.

The disadvantage of using tracks with variable numbers of sectors lies in possibly increased retrieval times due to more complicated main memory management schemes. Tracks with many sectors will take longer to retrieve than those with fewer sectors. Buffer management also becomes an issue when potentially large amounts of buffer space could be required. Organizing the space to ensure the ability to buffer the largest track may not be a simple matter, especially in a storage system shared among many concurrent users. Fixed size tracks do not have these problems.

3.1. Hashing Implementation

Implementing the variable track capacity hashing scheme for CLV discs requires a small amount of storage overhead to hold a table that stores the mapping function between the hash buckets and the CLV spiral track. If hash buckets are laid out on the spiral in their logical order, the table will contain one sector address per hash bucket. Successive table entries will delimit the boundaries of each track. If a more complex physical arrangement is called for, perhaps because of a need to reduce sector crossings, it can be accommodated by including more information in each table entry. It is expected that the mapping tables, even for many hash buckets, will be small enough to fit in main memory, but could be maintained on magnetic disc if so required by an implementation.

3.2. Hashing on CD ROM

CD ROM discs have similar characteristics to those used for audio compact discs (CD's). The nature of the manufacture and use of CD ROM type optical discs make them a prime platform for implementing hashing. The contents of a CD ROM disc are never updated or altered, so much time, effort and resources can be spent preprocessing the disc contents to produce an organization that will ensure good access performance. Once expended, this effort is not repeated until a new disc is issued to physically replace the old one. These characteristics and the preprocessing stage make it possible to achieve virtually 100% space utilization simply by eliminating the unused space that, for efficiency reasons, is usually present in files that are expected to undergo change. Having no need to handle insertions or deletions, implementing hashing on CD ROM optical discs becomes a one time job of preprocessing the file by computing the contents of each hash bucket and constructing the hash bucket/sector address mapping table.

Before the disc is physically pressed, a certain degree of optimization and improvement can be injected into the expected access performance during the preprocessing stage. During this period, it is possible to examine and experiment with the performance of a variety of different combinations of hashing functions and numbers of hash buckets, selecting the one that provides the greatest expected access performance for the file contents. We can also give attention to the bin packing problem of assigning hash buckets to physical locations to reduce sector boundary crossings (and therefore disc transfer times).

As there is no requirement to allocate disc space to hash buckets that have not been assigned records; a simple "NULL" entry in the table is all that is required to accomplish this. One can use many hash buckets in an attempt to reduce the expected number of records per bucket, and hence the time to retrieve and search each one. There is virtually no penalty for having many buckets other than having a larger mapping table. Even for large files, the mapping table size for a CD ROM implementation can be quite manageable. On a CD ROM disc, complete sector addresses require about twenty bits, 7 bits for the minute (0-99), 6 bits for the second (0-59) and 7 bits for the data block number (0-74), or two and a half bytes; fewer bits can be used if necessary since there are less than 500,000 sectors on a CD ROM disc, but that would involve more processing. Consider, a hashing scheme using 20,000 hash buckets would only require 45000 bytes. If each bucket was allocated a track of just one sector, the size of the file would be 40 megabytes and the table overhead, a mere 0.1%. If needed, the size of the table could be reduced by storing sector run lengths in fewer bytes per table entry (i.e. store the number of sectors in each track). The savings in space that would result need to be measured against the effort to calculate an absolute sector addresses.

3.3. Hashing on CLV WORM

Providing an efficient hashing access mechanism for a WORM type CLV optical disc will be more difficult than for CD ROM. The obvious complication is the existence of dynamically changing files and the necessity to allow for the insertion and deletion of file contents on a medium that does not allow for the reuse of previously allocated space. Also, for the applications in which WORM discs are typically used, there will usually be no preprocessing stage in which to explore alternate hashing strategies, so in general, it will not be possible to improve the expected access performance by adjusting the number of hash buckets or the hashing function to fit the data set as it was for CD ROM.

To improve efficiency, and in particular space utilization, implementing hashing for a file stored on a WORM disc will involve some degree of buffering on a magnetic disc for both the mapping table and the contents of hash buckets. A pointer in each entry of the mapping table would lead to what is essentially an overflow chain stored on the magnetic disc of records that are assigned to the hash bucket but which have not yet been archived on the optical disc. Information on which records have been logically deleted from the hash bucket (but which cannot be erased from the physically unerasable disc) might also be

stored in the bucket's entry in the mapping table. At some point, as the buffer space gradually fills up, a portion or all the buffered information would be flushed. The "new" records will be merged with the old (logically undeleted) ones already bon the optical disc and written together on new tracks; the mapping table will also be updated to reflect the changes. The expected disc space consumption for a buffered hashing organization (BHash) for WORM optical discs is analyzed in [9].

For the archiving of large objects in a hash file access scheme on WORM discs, it will be efficient in terms of space utilization to store a pointer to the object in the hash bucket rather than the object itself. This is because it would be wasteful to re-copy large objects (records), that might be megabytes in size, simply for the benefit of merging buffered information with that on the optical disc. A pointer would add another disc access to the retrieval process but the savings in storage space may be worth the extra effort.

3.4. Hashing as a Primary Access Method

When used as a primary access method, we expect that the random access retrieval performance of hashing will be about as good as is possible from a CLV format disc. For the scheme outlined above, the delay will consist of the time to do one seek and the time to read one hash bucket. We can use the results of [8] to analyze the expected retrieval performance.

The expected delay is given by:

$$expected\ delay = S_t + Bk_t$$

Where S_t is the expected seek time and Bk_t is the expected bucket transfer time.

A typical value for the expected seek time for CD ROM discs is 400msec. The expected bucket transfer time depends on the expected number of sectors occupied by the bucket, this in turn is determined by the size and average number of records hashed to a bucket. The time required to transfer one sector from a CD ROM disc is exactly 13.3msec (defined by standards).

If records are small enough to be contained in a sector and a sufficient number of hash buckets are used to reduce the expected number in a bucket to one, then at most two sectors will need to be retrieved if sector boundaries can be crossed by records, and at most one, if not. A delay between 413msec and 426msec is expected for the retrieval of one hash bucket.

3.5. Hashing as a Secondary Access Method

When used as a secondary file access method, the dispersal of qualifying records to different parts of the file (because they have been ordered by some other primary access method) will lower the expected retrieval performance of hashing. In that case, the contents of a hash bucket will not be records but pointers to records in the file. These pointers cause many more accesses beyond the one to retrieve the hash bucket contents.

Where $\overline{K}$ and $\overline{B}$ are as calculated in section 2, ignoring the transmission delay, the approximate expected retrieval performance is given by:

$$expected\ delay = (\overline{K}(\overline{B}) + 1)\, S_t$$

$\overline{B}$ depends on the expected number of records (pointers) per hash bucket and whether sector boundaries are crossed. The transmission delay is ignored because the delay owing to seeks is expected to be much larger.

4. Tree Index File Organizations on CLV Optical Discs

While hashing is an excellent file organization to use for CLV format discs, it does not allow the efficient processing of range and inequality queries, nor does it provide quick sequential access. A class of organizations that do allow these types of operations are tree indices such as B-trees [10] and ISAM.

Depending on the type of optical disc, either CD ROM or WORM, the task of providing an indexed sequential file structure such as B-trees or ISAM will be either straightforward or taxing. Just as for hashing, the static nature of the contents of a CD ROM disc make it easy to organize them into pointer linked tree organizations, but for WORM discs, the scenario is different. Modification of files, mostly insertions, is possible and is expected to occur frequently. To accommodate these changes, some modification of the file organization will be necessary. If the organization uses pointers to maintain its structure, then some of the pointers may have to change as well. This, in turn, will require new disc space to be consumed to store the new pointer values. The difficulty lies in devising techniques to keep the amount of consumed space to a minimum. A buffering scheme that reduces consumed space is described in [9].

4.1. B-trees on CLV

As stated above, the nature and applicability of a B-tree type organization to a CLV format disc depends heavily on the type of the disc, WORM or CDROM.

For a WORM disc, a direct, naive, implementation of a B-tree structure will consume disc space very quickly. The reason for this is that each modification will require space to be consumed to update some of the pointer values that link nodes in the tree together. The approach taken in [16] reduces this consumption by buffering the pointer values on magnetic disc where they can be updated without consuming storage space. A better approach is the Write-Once-B-tree (WOBT) [11] which does not change pointer values, but, rather, simply appends another value (space on the disc is reserved for appends) and then relies upon the physical sequencing of the different versions of individual pointers to disambiguate them (the current version is the last one in the node).

For CD ROM's, a pure B-tree organization is not appropriate. B-trees are designed to efficiently organize data sets which undergo considerable change over their life times. Because of this need to efficiently accommodate change, storage space in some nodes is left unused. This makes B-trees less efficient in terms of storage utilization and retrieval speed (because the tree is higher than it needs to be) than is possible if the data set remains static.

4.2. Clustered BIM-trees on CD ROM

A more efficient organization for CD ROM's would take full advantage of the fact that data sets on such discs never change. This leads directly to the development of a close relative of a B-tree for static data sets on CD ROM, namely, a perfectly balanced multiway tree with implicit addressing (there is no need for pointers since the tree

never changes). We will call such a tree a *Balanced Implicit Multiway tree* or a *BIM-tree*. BIM and B-trees are closely related, but because a BIM-tree is perfectly balanced, it may have nodes at its lowest level that are less than 50% full and hence may not always meet the definition of a B-tree. The advantage of a BIM-tree is that it will have 100% space utilization (above the lowest level) and possibly a lower height than a corresponding B-tree. Note that no pointers are required between nodes of the tree as it's regular structure allows implicit addressing (i.e., the position of a node at the next lower level can be computed from the current node).

As is true for B and B^+–tree's, there are two types of BIM trees, BIM and BIM^+. The difference being that a BIM^+–tree propagates key values in the nodes down to the lowest level, a BIM tree need not. This means that a BIM^+–tree will be faster for resolving range queries but will require more space to store.

Implementing a BIM-tree for a CD ROM disc involves the straightforward construction of a balanced multiway tree from the data. For a BIM^+–tree, before the construction of the tree, the data set being indexed can be stored sequentially, and in order, on the spiral track. Unlike a B^+–tree, no horizontal pointers are required at the lowest level of a BIM^+–tree since the data set is stored sequentially. Both types of BIM-trees are searched in a similar manner to B-trees.

To get the best retrieval performance from a BIM-tree, the nodes of the different levels of the tree should be clustered. For example, by placing them on the spiral sequentially in the order that they occur; namely, with the root node first, the nodes of the second level next, the nodes of the third level after that, and the rest of the levels in corresponding order. This requirement is a direct consequence of the optimal scheduling theorem of section 2. This ordering will ensure that the access mechanism will move in only one direction when accessing the tree so that an optimal number of seeks can be selected.

From [17], with the root of the BIM-tree not in main memory, we find the expected cost of retrieval to be about:

$$(S_t + B_t \left\lceil m \frac{E_s}{B_s} \right\rceil + D \log_2 m) \log_m N$$

Here, S_t is the expected seek time, B_t the block transfer time, m the branching factor (the number of links to nodes at the next level of the tree), E_s the size of one entry in the node and B_s is the size of a block (sector). $D \log_2 m$ is the time to perform a binary search on the node when it is in main memory and N is the number of records in the file.

Differentiating and setting the derivative to zero to find the branching factor (m) that will minimize the expected delay, we produce the following expression:

$$m \ln m - m = \frac{S_t}{B_t} \left\lfloor \frac{B_s}{E_s} \right\rfloor$$

4.3. BIM-tree Access Performance on CD ROM

Using typical performance parameters of a CD ROM disc, we can calculate the optimal branching factor for a given number of records. A typical value for the expected seek time S_t is 400msec. The time to transfer one block or sector, B_t, is exactly $1/75$ of a second, and the size of a sector, B_s is 2048 bytes when using the highest level of error correction, as will be the case for record oriented data. Lower levels of error correction are used for objects that can tolerate a small rate of error. Bit maps and recorded audio are good examples of such objects since a few wrong bits in either will not usually be noticed by a viewer or listener.

For an node entry size of 50 bytes we calculate the following:

$$m \ln m - m = \frac{0.4}{1/75} \left\lfloor \frac{2048}{50} \right\rfloor = 1200$$

which corresponds to a value of m equal to 263 (i.e., each node should have 262 or $m-1$ entries).

As an example, consider indexing an entire CD ROM disc consisting of 1000000 records. Reserving 50 megabytes ($50 \cdot 100000$) for the BIM, 500 megabytes (at the highest level of error correction) would still be left for the storage of the records. The expected number of accesses would be:

$$= \left\lceil \log_m N \right\rceil = \left\lceil \frac{\ln N}{\ln m} \right\rceil$$

$$= \left\lceil \frac{\ln 1000000}{\ln 263} \right\rceil = 3 \; accesses$$

Using the performance parameters given above, the expected delay due to disc accesses would be:

$$expected\ delay = (0.4 + (1/75) \left\lceil 262 \frac{200}{2048} \right\rceil) \cdot 3$$

$$= 1.48 \; seconds$$

Because the 262 entries do not completely fill all seven ($(263 \cdot 50)/2048 = 6.4$) of the sectors required to store them, we can improve the expected performance and space utilization by increasing the branching factor m (and hence the number of entries in each node) to a value that does. Here, we can increase m to $1 + \left\lfloor 7 \frac{2048}{50} \right\rfloor = 287$, which corresponds to 286 entries per node. A similar analysis can be done for when the root of the BIM tree is stored in main memory.

In the above description, we assumed one particular clustering algorithm for the nodes of BIM and BIM^+–trees. This algorithm will allow the disc head to move in one direction. However, there are alternative clustering algorithms possible that still satisfy the requirement of moving the disc head in one direction, but at the same time better exploit the span access capability of optical discs and the optimal scheduling algorithm to reduce expensive long seeks. We are currently experimenting with such algorithms.

5. ISAM on CLV WORM Optical Discs

An alternative sequential access method to B-trees is ISAM. Like B-trees, it allows sequential access to file contents and primarily consists of a pointer linked tree structure. Unlike B-trees, the index structure of ISAM has the useful property of remaining unchanged when the file undergoes modification. This property is ideal for data sets stored on WORM discs.

It is not easy to implement a "pure" ISAM file organization on a WORM disc, but a slight variation can be easily accommodated. The changes necessary to adapt to the characteristics of WORM discs lie in the method of handling overflows and in the frequency of periodic file maintenance.

Insertion of records can cause the disc space allotted to a particular range of keys to overflow. The conventional ISAM approach to this problem is to reserve an overflow area and shuffle the positions of old and new records in the primary and overflow areas to maintain physical sequential ordering of the records. Periodic maintenance on conventional ISAM files is performed to clear the overflow areas by reorganizing the file and its index. The maintenance improves the expected retrieval performance by reducing the expected amount of overflow processing required.

On WORM discs, it is not possible to "shuffle" the position of records to maintain physical as well as logical ordering. Rewriting the new and old records to do so would quickly use up the disc space and decrease expected insertion performance.

Much like the arrangement previously described for hashing, a modified ISAM approach for CLV WORM optical discs could use magnetic disc storage as the overflow medium [7, 9]. It can also take advantage of the variable track capacities (i.e., portions of the spiral) to reduce the need for periodic file reorganization. In such an approach, the top level of the ISAM index would be kept in main memory or magnetic disc and the rest on the optical disc. A small table would also be kept in main memory to map between the lowest levels of the index and both physical (optical) disc locations and overflow chains stored on magnetic disc.

When a record is inserted into the file, it will be placed on the overflow chain kept for its index position on the magnetic disc. When a record is retrieved, the optical disc will be accessed. If the record is not found, then the overflow chain on the magnetic disc will be searched. Sequential access will require both discs to be accessed if there are records buffered on the magnetic disc.

The deletion of a record could be handled by storing information in the mapping table or on magnetic disc. The exact method should not prove critical as the type of applications that use WORM discs tend to be archival in nature and deletions are expected to be rare.

When an overflow chain becomes excessively long (by some criteria), its contents and the contents of its corresponding primary area on the optical disc will be merged and stored on a new WORM disc track. The mapping table would be updated to reflect the change and the buffer space on the magnetic disc flushed. Note that the index will not change.

The availability of variable capacity tracks on CLV WORM optical discs allows the capacity of a track to expand to meet the load imposed on it. Thus, the flushing and merging operations continue without requiring a reorganization of the file and its index until the capacity of the disc is exhausted. When the disc is full and further insertions are pending, the file and its index (along with the new records) can be transferred to a new disc and be reorganized in the process.

Reorganization may be desirable before the disc is full to adjust the index to better match the actual contents of the file. This will shorten the length of tracks that have grown excessively long because of insertion patterns that did not match the organization reflected in the current index. A track may be considered to be too long if its capacity exceeds the size of the main memory buffer or if the transmission time required to read it from the disc exceeds some threshold. Shortening the track length by file reorganization will lessen buffering problems and improve the expected retrieval performance by reducing the transmission delay and the expected length of overflow chains. The trade off between retrieval performance and disc space consumption for buffered ISAM organizations is given in [9]. In short, to maintain a high level of retrieval performance (few seeks) disc space must be consumed to maintain the sequentiality of data sets. Lower levels of sequentiality require lower levels of space consumption.

6. Summary, Conclusions and Future Research

We have shown that some conventional file access methods used for a magnetic disc that use a CAV recording scheme can be adapted to the characteristics of CLV optical discs, such as CD ROM and WORM.

In particular we have shown that particularly good retrieval performance and file utilization close to 100%, can be achieved by using hashing as a primary file organization method on CLV format optical discs such as CD ROM. We have provided analytic cost estimates of the performance and of the organization for CD-ROM's and WORM's.

For tree like file organizations, particularly B-trees, it was shown that they would not have particularly good access times or space utilization when used on CLV optical discs. B-trees are not a good access method for WORM discs as they are expected to require large amounts of disc space to accommodate file changes. The ISAM organization for primary key retrieval was discussed and an implementation strategy that made better use of the disc was described. We described clustered BIM's, a class of tree organization appropriate for CD ROM's

We illustrated throughout the paper the implementation advantages, imparted by the static nature of files on CD ROM, to all the file access mechanisms; as well as the optimization that can be performed during the preprocessing stage of the disc's contents. It was also observed that the spiral recording scheme used on almost all CLV optical discs improved the expected retrieval performance and space utilization of file access methods compared with the concentric tracks of CAV format discs. This was because data sets (e.g., hash buckets) of arbitrary size could be allocated on the spiral and accessed with a single seek.

Both hashing and ISAM implementations on WORM discs can greatly benefit from the availability of read/write memory (magnetic discs, large main memory, etc.) for buffering incoming data before placing them on the write-once medium. Such a strategy would cluster the

data of a logical block (hash bucket, ISAM bucket) in WORM storage, although insertions to the bucket may have happened at different time intervals. Good clustering of data results in good retrieval performance for the bucket. Because of the limited rewritable storage capacity, data on WORM disc may have to be replicated to maintain good clustering. Alternative algorithms and analytic performance evaluations describing the retrieval performance against the WORM disc space utilization for such file organizations appears in [9].

It has been extensively discussed elsewhere that for text retrieval applications that use WORM optical disc storage as the storage medium, signature based access methods present particular advantages compared to other implementations [3, 11]. A large scale implementation of a WORM disc based multimedia document management system (MINOS project) uses access methods based on signatures and it uses buffered writes from magnetic to optical disc storage to optimize performance [4,6].

7. References

[1] Bell, A., Marrello, V., "Magnetic and Optical Data Storage: A Comparison of the Technological Limits", *Proceedings IEEE Compcon*, Spring 1984, 512-517.

[2] BYTE86, Collection of Articles, *Byte*, May 86.

[3] Christodoulakis, S., Faloutsos, C., "Design Considerations for a Message File Server", *IEEE Transactions on Software Engineering*, Vol. SE-10, No.2, pp. 201-210, March 1984.

[4] Christodoulakis, S., Theodoridou, M., Ho, F., Papa, M., Pathria, A. "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System", *ACM TOOIS*, Vol. 4, No. 4, October 1986, pp. 345-383.

[5] Christodoulakis, S., "Analysis of Retrieval Performance for Records and Objects Using Optical Disk Technology", *ACM Transactions on Data base Systems*, June 1987.

[6] Christodoulakis, S., Elliott, K., Ford, D.A., Hatzilemonias, K., Ledoux, E., Leitch, M., Ng, R., "Optical Mass Storage Systems and their Performance", *IEEE Database Engineering*, March 1988.

[7] Christodoulakis, S., Ford, D.A., "File organizations and Access Methods for CLV Optical Disks", Technical Report CS-88-21, Department of Computer Science, University of Waterloo, March 1988.

[8] Christodoulakis, S., Ford, D.A., "Performance Analysis and Fundamental Performance Trade Offs for CLV Optical Disks", *Proceedings ACM SIGMOD*, Chicago, June 1988.

[9] Christodoulakis, S., Ford, D.A., "Retrieval Performance Versus Disc Space Utilization on WORM Optical Discs", *Proceedings ACM SIGMOD*, Portland, 1989 (To appear).

[10] Comer, D., "The Ubiquitous B-tree", *Computing Surveys*, Vol. 11, No. 2, pp. June 1979.

[11] Easton, M.C., "Key-sequence data sets on indelible storage", *IBM J. Res. Develop*, Vol. 30, No. 3, (May 1986).

[12] Faloutsos, C., Christodoulakis, S., "Description and Performance Analysis of Signature File Methods for Office Filing", *ACM TOOIS*, Vol. 5, No. 3, July 1987.

[13] Fox, E.A., "ACM Press Database and Electronic Products - New Services for the Information Age", *Communications of the ACM*, Vol. 31, No. 8, pp. 948-951.

[14] Fujitani. L., "Laser Optical Disks: The Coming Revolution in On-Line Storage", *CACM 27*, 6 (June '84), 546-554.

[15] Maier, D., "Using Write-Once Memory for Data base Storage", *Proceedings ACM PODS 82*, 1982.

[16] "Optifile: Technical Reference Manual", KOM Inc. Ottawa, Version 1.0, February 1986

[17] Reingold, E.M., Hansen, W.J., "Data Structures in Pascal", Little Brown, Boston, 1986.