

Performance Analysis and Fundamental Performance Trade Offs for CLV Optical Disks

*Stavros Christodoulakis
Daniel Alexander Ford*

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

ABSTRACT

CLV type optical disks is a very large and important class of optical disk technology, of which CD-ROM disks form a subclass.

In this paper we present a model of retrieval from CLV optical disks. We then provide exact and approximate results analyzing the retrieval performance from them. Our analysis takes into account disks with and without a mirror in the read mechanism, small objects completely placed within block boundaries, placement that allows block boundary crossing, as well as very large objects (such as documents) placed within files.

In the second part of the paper we describe some fundamental implications of physical data base design for data bases stored on CLV optical disks. We show that very significant performance gains may be realized by appropriate design.

1. Introduction

The immense storage capacities, economical costs and good retrieval performance presently offered by various forms of optical disk technology are making it an attractive choice as a storage medium for large data base applications. They are also making possible new and more demanding applications such as multimedia databases which require large amounts of high performance storage. Further advances in the technology expected in the near future such as even greater storage capacity, still lower costs and faster access times, and the possibility of erasable optical disks, promise to ensure that it will continue to grow as a prominent storage technology for some time to come [1, 2, 4, 7].

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and / or specific permission.

© 1988 ACM 0-89791-268-3/88/0006/0286 \$1.50

Optical disk technology takes two basic forms, that of *Constant Angular Velocity* (CAV) disks, which rotate at a fixed rate but have a variable storage density and that of *Constant Linear Velocity* (CLV) disks, which rotate at a variable rate but have a fixed storage density. The two forms embody the trade-offs of speed and storage; CAV type disks have faster access times while in general CLV type disks have greater storage capacity.

Currently, CLV technology is being used extensively in all CD-ROM's (read only disks) to publish and distribute large volumes of data and information. It is also used in some WORM (Write Once Read Many times) [8] designs. WORM disks are used in data base or multimedia archival applications.

Since the retrieval performance of an application's storage system plays a very important role in the success of the application, it is critical to understand and produce analytic estimates of the performance of optical disks. Such knowledge will be useful in the development of data structures, query optimizers and other techniques and tools important to the enhancement of the retrieval performance of systems employing optical disk technology.

For optical disks, the seek time is generally much larger than the rotational latency. Though some WORM disks are faster, delays of up to one second or more are possible with the current CD-ROM technology and so the seek time is usually the most important factor in the time required to answer a query. The issue is complicated slightly however by some optical disk drives which are capable of reading from more than one track without performing a seek. The access mechanism on some drives is equipped with an adjustable mirror that allows slight deflections of the beam of laser light used to read the data. This enables it to be aimed at any one of a small set of tracks immediately beneath the head. This set of tracks is called a *span* and the number of tracks in the set is called the *span size*. This capability can be likened to cylinders on magnetic disk packs except that for optical disks the sets of tracks in spans can be overlapping. This span access capability is provided by manufactures of both CAV and CLV disk drives (although currently, it is more frequently encountered with CAV drives).

We examine three cases of retrieval from CLV disks and determine the required number of seeks in each. The first two, for *non-crossing* objects and *crossing* objects, involve the retrieval of objects close in size to a disk sector. A *non-crossing* object is completely contained within a sector and hence can be retrieved with one sector access, such placement of objects may frequently be employed when storing records. A *crossing* object may overlap sector boundaries and so could possibly require more than one sector to be accessed for its retrieval. The third case we examine is for large objects, of which bitmaps and digitized sound are prime examples. The size of such large objects can be of the order of 300 to 500K or more, and can exceed the capacity of several tracks on the disk and require more than one seek to be retrieved.

Because there is a one-to-one relationship between the minimum number of seeks required to retrieve a set of objects from a file and the minimum number of non-overlapping spans (sets of tracks) required to "cover" those objects¹, it is sufficient to restrict our analysis to determining that number of spans. Note that a span size of one track corresponds to the conventional case of no deflecting mirror (although the characteristics of CLV disks make the analysis different than CAV disks).

2. A Model and a Schedule

An optical disk 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, $i=1,2,\dots,T$ (to avoid discussion about boundary conditions we assume that track numbers are extended above T and below 1). Each track is composed of a number of 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 which completely exist within Q consecutive tracks (track i is one of them). In order 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 disks *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 number N of objects (or records) may qualify in a query. In the case of optical disks the number of times that the access mechanism has to be moved for accessing the data which 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.

A spiral scheme is generally used on CLV optical disks for the storage of data rather than the concentric rings found on CAV disks. The spiral consists of one long physical track of data recordings that begins on the inner part of the disk and continues until close to the outer edge of the disk where it terminates.

A track for our purposes starts from the intersection of the spiral with the radial line that starts at the centre of the disk and passes through the very beginning of the physical spiral. The track ends at the next intersection with this radial line. Since sectors are of fixed length, the radial line may intersect the body of a sector.

¹ provided all the seeks are in one direction, either into the centre or out to the disk edge.

3. Exact Solutions

Calculating the exact solution for the expected number of spans when the span size is equal to one track is fast to compute. When the span size exceeds one track the situation becomes more complex as the possibility of spans covering tracks with different capacities must be addressed. This makes the computation much more expensive.

3.1. Span Size Equal to One Track

When the size of a span is restricted to one track, as is the case when the access mechanism is not equipped with a mirror, we get the following expression for the exact number of spans expected to be required to access a number of (non-crossing) objects. Let R be the number of equal track capacity segments in the file and let t_i be the number of tracks in segment i . Let M be the object capacity of the file and let n_i be the object capacity of tracks in segment i . Let N be the number of objects selected and

Then the expected retrieval cost is given by:

$$\sum_{i=1}^R t_i \left(1 - \frac{\binom{M - n_i}{N}}{\binom{M}{N}} \right)$$

The division in the above expression gives the probability of one of the tracks in the i 'th segment not containing any of the objects selected for retrieval, subtracting this from 1 gives the probability of one track in segment i containing at least one object and requiring a span (of size one track) to retrieve it. We multiply this probability by the number of tracks in the i 'th segment to get the number of spans required to retrieve objects from it, and sum over the number of segments in the file for the final result.

3.2. Span Size of Greater Than One Track

Span sizes of more than one track introduce the possibility of having variable capacity tracks within a span, requiring more involved analysis to compute an exact result.

For each of the possible numbers of spans that could be used to retrieve a given number of objects for a file, we determine the probabilities of each being required. This probability is determined by computing the number of object arrangements a number of spans could retrieve and dividing by the total number of object arrangements possible in the file.

Let T be the number of tracks in the file, N the number of objects to be retrieved and M the maximum number of objects in the file. Let Q be the number of tracks in a span (the span size), K the number number of spans being used to retrieve the objects and \bar{K} the expected number of spans (the retrieval cost). Then the exact result is given by:

$$\bar{K} = \sum_{K=1}^{\left\lfloor \frac{T}{Q} \right\rfloor} K \frac{\text{arrange}(1, N, T, K)}{\binom{M}{N}}$$

In this formula, $\text{arrange}(i, n, t, k)$ is the number of object arrangements of n objects that can be retrieved by k spans from the t tracks that begin at track i .

The main component of this process is the determination of the number of object arrangements that a given number of spans can be used to retrieve. This problem is different for non-crossing and crossing objects, as the existence of crossing objects requires the possibility of a span boundary falling on the sector boundary crossed by an object, requiring a second span to completely retrieve the object.

3.2.1. Non-Crossing Objects

For non-crossing objects arrange is given recursively in the following expression. Let Nt_i be the object capacity of track i and Ns_i , the object capacity of the span anchored at track i . Then:

$$\text{arrange}(i, n, t, k) = 0, \text{ if } n = 0, n < k, k = 0 \text{ or } t < (k-1)Q + 1$$

$$\begin{aligned}
\text{arrange}(i,n,t,k) = & \text{arrange}(i+1,n,t-1,k) + \\
& \sum_{l=1}^{\min(n,Ns_i)} \text{arrange}(i+Q,n-l,t-Q,k-1) \sum_{j=1}^{\min(Nt_i,Ns_i-(l-1))} \binom{Ns_i-j}{l-1}
\end{aligned}$$

Where

l - is the number of retrieved objects from this span.

j - is the index position of the one object that must be retrieved from first track of the span.

The first term of *arrange* counts the number of arrangements which do not have objects located in track i that can be retrieved using the same number of spans.

The second term counts the number of arrangements that can be retrieved with the given number of spans that have objects located in track i . The first summation computes the number of object arrangements in the file for each of the different possible numbers of objects l in the first span. For a given value of l , the second summation in the term calculates the number of their arrangements within the sectors of the span that have at least one object in the first track.

3.2.2. Crossing Objects

Calculating *arrange* for crossing objects, we must account for the possibility of an object extending past the bounds of a span by crossing the boundary of the last sector in that span and requiring a second span to completely cover it. This makes the calculation of *arrange* much more complex although its expression is similar to that for Non-crossing objects. It is omitted for brevity.

4. Approximate Solutions

Although CLV disks are complex to analyze exactly, they are not so different from CAV type disks that one can rule out the success of extending the approximations valid for CAV disk to the CLV case. The rate at which track capacity increases in CLV disks is very slight, requiring, for example, some 1400 tracks on a CD-ROM disk to increase the capacity by one sector. This leads one to suspect that for objects with sizes that are considerably smaller than the capacity of a track, such as non-crossing and crossing objects, the results for both types of disks might be fairly close. We examine this possibility in the next section.

4.1. Average CAV Approximation

We can compute upper and lower bounds on the approximate solution to the expected number of spans required to cover a given number of objects on a CLV disk by computing two corresponding CAV solutions, one for a CAV disk with the same track capacity as the minimum found on the CLV disk and one for a CAV disk with the maximum found on the CLV disk. For the upper bound, no CLV solution would require a larger number of spans since the generally larger capacity spans on the CLV disk would cover more objects so fewer would be required. Similarly, for the lower bound the generally lower capacity spans on the CLV disk would cover fewer objects so more would be required.

Computing the upper and lower bounds for the case of small objects and 10 tracks per span, we find that our suspicions about the closeness of the CLV and CAV solutions was justified. For a CLV disk with a minimum track capacity of 9 sectors and a maximum capacity of 23 sectors (typical values for CD-ROM), an upper bound is the CAV solution for a disk with uniform track capacity of 8 sectors and a lower bound is the CAV solution for 24 sectors².

What this shows is that if we use any value of track capacity (the average being a good choice) between the minimum and maximum track capacities of a CLV disk to produce a corresponding CAV solution, we can expect it to be very close to the desired CLV solution.

4.2. Multiple CAV Approximation

An alternate method of calculating the expected number of spans is to use the expected distributions of objects to the equal track capacity segments of the file and calculate the CAV solution for each, summing the solution to produce the final result.

For R segments, let \bar{K}_i be the expected number of spans for subsegment i , and \bar{N}_i be the expected number of objects that fall in subsegment i , then this is expressed as:

² these values are actually one more and one less than the maximum and minimum track capacities, producing even more extreme bounds.

$$\bar{K} = \sum_{i=1}^R \bar{K}_i(\bar{N}_i)$$

This method gives results for non-crossing and crossing objects comparable to those obtained above using a straight average track capacity.

Computing the expected retrieval cost for small objects in this manner is not as computationally efficient as using a single CAV solution. This situation is different for computing retrieval costs when objects which are large compared to the retrieval capacity of a span. We examine this in the section on computing the retrieval cost for large objects.

4.3. Expected Spans for Non-Crossing Objects

The approximate solution for the expected number of spans required for the retrieval of non-crossing objects from a CAV disk is found in [3] :

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

Where \bar{B} is the expected number of tracks to be retrieved.

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 :

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

4.4. Expected Spans for Crossing Objects

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

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

Where

$$\bar{B}_c = \frac{N}{TC} \left[(T-1) \left(2 - \frac{N}{TC} \right) + 1 \right]$$

5. Large Objects (with respect to span size)

When we consider the existence of large objects in a file, we find that the retrieval performance for the two types of disks can differ considerably. The number of spans required to retrieve large objects from a CAV type disk will be independent of the objects' placements on the disk. This is not true for CLV type disks where the retrieval capacity of a span depends on its location, and hence, objects stored in the outer tracks may require fewer spans to be retrieved than if they were stored on the inner tracks of the disk or anywhere on a comparable CAV type disk. This is particularly true when object size is large with respect to the retrieval capacity of a span.

An approach to the problem of calculating the expected cost (number of spans) of retrieving a number of large objects, is to partition the file into segments of equal track/span capacity and combine the CAV solutions computed for each to produce a final result. Doing so we are able to derive methods for producing both exact and approximate results.

5.1. Exact Solution

Consider a partition of the tracks of the file into R segments. Let M_i be the object capacity of each segment, $i=1,2,\dots,R$. Let N_i be the number of objects distributed to each segment ($\sum_{i=1}^R N_i = N$) and $P(U) = P(N_1, N_2, \dots, N_R)$ be the probability of such a distribution..

Then,

$$P(U) = \frac{\prod_{i=1}^R \binom{M_i}{N_i}}{\binom{M}{N}}$$

If each partition has a constant number of objects per track the expected number of spans required for retrieving N objects is:

$$\sum_U P(U) \sum_{i=1}^R \bar{K}_i(U) + \Delta_R(U)$$

Where $\bar{K}_i(U)$ is the expected spans completely contained within segment i , and $\Delta_R(U)$ is a correction term that represents the expected number of spans selected that cross segment boundaries.

5.2. Approximate Solutions for Large Objects

An approximate solution can be derived using the model for the exact one presented above. The computation cost can be adjusted by using less than the maximum number of segments and calculating the CAV solutions for each using the average of the track capacities in the segments. This approach allows one to trade-off accuracy and computation cost. The greater the resolution of the approximation (i.e. the number of segments) the greater is the accuracy of the result, but also greater is the number of computations required to produce it. In the limit the calculation of the retrieval cost is exact because the track capacities used will be the actual capacities of the tracks, not an average. This makes the method intuitively very satisfactory.

The approximation is calculated by combining the results for each of the possible distributions of the objects to the segments weighted by the probability of the distribution occurring.

An approximation of the expected cost then, for the case of large objects and for a resolution level having R segments, is given by:

$$\sum_U P(U) \sum_{i=1}^R \bar{K}_i(U)$$

Where

$$\bar{K}_i(U) = \bar{K}_e \cdot N_i + S(N_i, T - (N_i \cdot \bar{K}_e \cdot Q), \text{olength}_i - (Q \cdot \bar{K}_e \cdot \text{tlength}_i))$$

is the number of spans required for the i 'th partition given the distribution U .

and

$$\bar{K}_e = \left\lfloor \frac{\text{olength}_i}{Q \cdot \text{tlength}_i} \right\rfloor$$

$$\text{olength}_i = \text{objectsize} + \frac{\text{tlength}_i}{2}$$

$$S(n, t, l) = \frac{n}{1 + \frac{N_{\text{span}} - 1}{N_{\text{file}} - 1} (n - 1)}$$

Where \bar{K}_e is the number of spans that are completely occupied by an object in the i 'th partition and olength_i is the effective size of the object. Half the length of one track is added to the object size to account for the fact that on average an object will start half way into the first track of the first span and so will extend that much farther into the file from the very beginning of the first span. N_{span} is the number of objects that can be retrieved by one span (a function of t and l) and N_{file} is the total number of objects that the file can contain. Finally, N_i is the number of objects distributed to the i 'th partition.

The first summation in the approximation adds the expected retrieval costs for each possible distribution U of objects to the segments of the file weighted by the probability of the distribution occurring. The second summation adds the individual costs expected for each partition given a distribution of objects U .

5.3. Multiple CAV Approximation for Large Objects

The usefulness of the multiple CAV approximation becomes apparent when employed to approximate the retrieval cost for large objects. Its application is identical to that presented above for non-crossing and crossing objects except that the formula used for the expected spans is that for large objects.

6. Fundamental Performance Tradeoffs for CLV disks

The varying track capacity found on CLV disks has some fundamental implications for the placement of data and the selection of both access methods and file organizations.

6.1. Sequential versus Random Access

If a file is being accessed using a sequential scan the delay in retrieving data is dominated by the transfer time. This is even more so if the disk has spiral tracks and seeks to the next track are transparent. For such a situation, it is clear that the retrieval cost in terms of time delay is the same for data located on the inside or outside tracks.

If the disk has concentric tracks (rare for CLV disks) the sequential scan will be less costly on the outside tracks because fewer seeks will be required. The difference may be marginal because the cost will still be dominated by the data transfer time. If there is a mirror in the read head of the device, any differences (even for concentric tracks) may completely disappear.

The situation is slightly different when a file is being accessed randomly. Assuming no problems with the availability of buffer space, we will typically allocate a buffer which is a multiple of the size of a sector. The random access is then characterized by its track selectivity d , which is the number of tracks with qualifying objects. For simplicity and without loss of generality we identify a block with a sector.

Let G be the number of sectors of the file. If the file is allocated in the inside tracks the N sectors will be allocated in many different tracks. Therefore, many more moves of the access mechanism may be required for the retrieval of all the qualifying sectors. Since the seek time dominates by far the data transfer times for the case of random access, the cost of accessing the data from the outside tracks is less.

The graph in Figure 1 below illustrates the difference in the number of movements of the access mechanism for equal sized files located on the inner and outer tracks.

The ratio of the costs will depend upon the selectivity of the queries. Note that rotational delays for a single record access per track will be greater for the file in the outside tracks. At the same time the average distance traveled by the access mechanism will be less in the outside tracks. Both factors are less important than the number of times that the access mechanism moves and their effect tends to cancel out. The above discussion leads us to two conclusions.

Firstly, the placement of a sequentially accessed file on inside or outside tracks of a CLV disk will not affect the retrieval performance significantly. However, the opposite is true for randomly accessed files whose choice of placement on the inside or outside tracks of a CLV disk will significantly affect the retrieval performance of accesses from those files. In particular placement of the randomly accessed file on the outside tracks of the disk gives better retrieval performance. As a result files which are accessed randomly should be placed towards the outside tracks of the CLV disk, while files that are accessed sequentially should be placed in the inside tracks of the disk.

Secondly, between two randomly accessed files of the same characteristics (e.g. record lengths, file sizes, selectivity of retrievals), the one with the highest frequency of use should be placed on the outer tracks to realize higher savings in retrieval costs.

These results give us criteria for deciding upon the different tradeoffs of file placement. Let g be the number of objects (records) per block (sector). Let M be the number of records of the file. Let f be the frequency of queries on the file (e.g. the number of queries per unit of time). Then the cost of retrieval per unit of time is:

$$f \cdot \text{tracks}(M, N, g, \text{loc})$$

Where loc is the starting location of the file (track number of the first track, assuming that the most inside track is track 1), and tracks for the case of span size equal to one track is given in section 3.1. Given two randomly accessed files, F_1 , characterized by f_1, M_1, N_1, g_1 , and F_2 , characterized by f_2, M_2, N_2, g_2 , and a starting track location loc to allocate them on a CLV disk, the file F_1 should be allocated before the file F_2 if and only if:

$$f_1 \cdot \text{tracks}(M_1, N_1, g_1, \text{loc}) + f_2 \cdot \text{tracks}(M_2, N_2, g_2, \text{loc}_2) < \\ f_2 \cdot \text{tracks}(M_2, N_2, g_2, \text{loc}) + f_1 \cdot \text{tracks}(M_1, N_1, g_1, \text{loc}_1)$$

Where loc_2 is the starting location of the second file after the first file has been allocated in the location loc , and loc_1 is the starting location of the first file (F_1) after the second file (F_2) has been allocated in the location loc . The

**Approximate expected spans for files
at disk extremities
Non-Crossing objects
1956 tracks, 23 sectors at outer edge
5000 tracks, 9 sectors at inner edge
10 tracks per span**

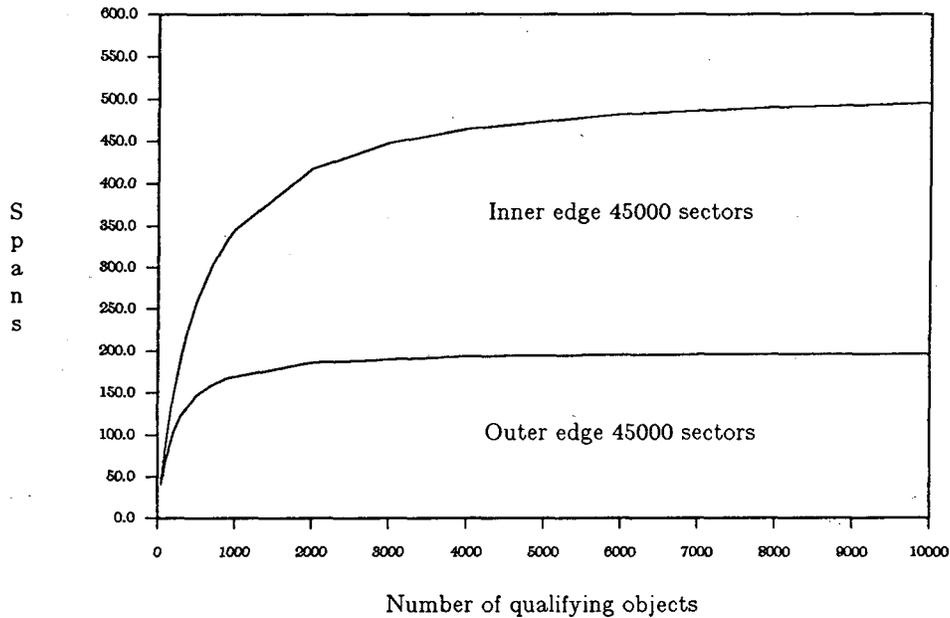


Figure 1.

calculation of loc_1 and loc_2 is given by the formula for calculating the spiral track lengths.

7. Conclusion

Our analysis shows the derivations of exact and approximate formulas for the expected number of spans required to retrieve a given number of objects from files stored on CLV type optical disks. The analytic techniques developed are needed for a variety of performance studies such as file organizations, query optimization, etc. For the case of computing the cost of retrieving objects (both non-sector boundary crossing and sector boundary crossing) that are small compared to the capacity of a span, we have shown that the CAV solution for an equivalent file having a uniform track capacity equal to the average of the file on the CLV disk is very close. When objects are large in size compared to the capacity of a span, we have shown that an approach employing multiple CAV solutions and the expected distribution of objects across the file on the CLV disk produces good results and is fast to compute (our results have been verified against simulation) [5].

We also discussed some fundamental implications of the nonuniform track capacities characteristic of CLV disk on the placement of files and showed that, in general, files with the greatest frequency of random accesses should be located on the outer tracks of the disk. The performance difference of alternative placements was shown to be very significant for files which are randomly accessed.

8. References

- [1] Bell, A., Marrello, V., "Magnetic and Optical Data Storage: A comparison of the Technological Limits", *Proceedings IEEE Comcon*, Spring 1984, 512-517.
- [2] BYTE86, Collection of Articles, *Byte*, May 86.
- [3] Christodoulakis, S., "Analysis of Retrieval Performance for Records and Objects Using Optical Disk Technology", *ACM Transactions on Database Systems*, June 1987.

- [4] Fujitani, L., "Laser Optical Disks: The coming Revolution in On-Line Storage", *CACM* 27, 6 (June '84), 546-554.
- [5] Christodoulakis, S., Hatzilemonias, K., "Optical Disk Simulation Package (ODS)", User's Guide, Version 3.0, University of Waterloo, 1987.
- [6] Christodoulakis, S., Ford, D. A., "Performance Analysis and Fundamental Performance Trade Offs for CLV Optical Disks", Tech. Report CS-88-06, University of Waterloo, 1988.
- [7] Maier, D., "Using Write-Once Memory for Database Storage", *Proceedings ACM PODS 82*, 1982.
- [8] OPTIMEM1000, "Optical Disk Drive (OEM MANUAL)", Optimem, 435 Oakmead Parkway, Sunnyvale CA 94086.