Printing Requisition/Graphic Services

15054

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3451.

TITLE OR DESCRIPTION: CS-88-06

DATE REQUISITIONED: July 14/88
DATE REQUIRED: ASAP
ACCOUNT NO.: 1 2 6 | 8 9 4 1 | 4 1

REQUISITIONER – PRINT: S. Christodoulakis
PHONE: 4452
SIGNING AUTHORITY: S. DeAngelis

MAILING INFO – NAME: Sue DeAngelis
DEPT.: C.S.
BLDG. & ROOM NO.: DC 2314
☒ DELIVER
☐ PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES: 30
NUMBER OF COPIES: 30

TYPE OF PAPER STOCK
☒ BOND ☐ NCR ____ PT. ☒ COVER ☐ BRISTOL ☒ SUPPLIED ☐ ________

PAPER SIZE
☒ 8½ x 11 ☐ 8½ x 14 ☐ 11 x 17 ☐ ________

PAPER COLOUR
☒ WHITE ☐ ________

INK
☒ BLACK ☐ ________

PRINTING
☐ 1 SIDE ____ PGS. ☒ 2 SIDES ____ PGS.

NUMBERING
FROM TO

BINDING/FINISHING
☒ COLLATING ☒ STAPLING a down left side ☐ ____ HOLE PUNCHED ☐ PLASTIC RING

FOLDING/PADDING

CUTTING SIZE

Special Instructions

Math fronts and backs enclosed

COPY CENTRE

OPER. NO.	BLDG.	MACH. NO.

DESIGN & PASTE-UP

OPER. NO.	TIME	LABOUR CODE
		D01
		D01
		D01

TYPESETTING

	QUANTITY			
PAP000000				T01
PAP000000				T01
PAP000000				T01

PROOF

PRF				
PRF				
PRF				

	QUANTITY	OPER. NO.	TIME	LABOUR CODE
NEGATIVES				
FLM				C01
FLM				C01
FLM				C01
FLM				C01
FLM				C01
PMT				
PMT				C01
PMT				C01
PMT				C01
PLATES				
PLT				P01
PLT				P01
PLT				P01
STOCK				
				001
				001
				001
				001
BINDERY				
RNG				B01
RNG				B01
RNG				B01
MIS00000				B01

OUTSIDE SERVICES

$ ________
COST

TAXES – PROVINCIAL ☐ FEDERAL ☐

Printing Requisition/Graphic Services

10358

1. Please complete unshaded areas on form as applicable.
2. Distribute copies as follows: White and Yellow to Graphic Services. Retain Pink Copies for your records.
3. On completion of order the Yellow copy will be returned with the printed material.
4. Please direct enquiries, quoting requisition number and account number, to extension 3451.

TITLE OR DESCRIPTION
Performance Analysis and Fundamental Performance Trade Offs for CLV Optical Disks CS-88-06 ~~XXxXXxXX~~

DATE REQUISITIONED Feb. 22/88
DATE REQUIRED Feb. 26/88
ACCOUNT NO. 1 2 6 6 0 7 4 4 1

REQUISITIONER- PRINT S. Christodoulakis
PHONE x3688
SIGNING AUTHORITY Yvonne Fink

MAILING INFO –
NAME Sue De Angelis
DEPT. Computer Science
BLDG. & ROOM NO. MC 6081E
☒ DELIVER
□ PICK-UP

Copyright: I hereby agree to assume all responsibility and liability for any infringement of copyrights and/or patent rights which may arise from the processing of, and reproduction of, any of the materials herein requested. I further agree to indemnify and hold blameless the University of Waterloo from any liability which may arise from said processing or reproducing. I also acknowledge that materials processed as a result of this requisition are for educational use only.

NUMBER OF PAGES 28
NUMBER OF COPIES 60

TYPE OF PAPER STOCK
☒ BOND □ NCR ____ PT. □ COVER □ BRISTOL □ SUPPLIED □ ______

PAPER SIZE
☒ 8½ x 11 □ 8½ x 14 □ 11 x 17 □ ______

PAPER COLOUR
☒ WHITE □ ______

INK
☒ BLACK □ ______

PRINTING
□ 1 SIDE ____ PGS. ☒ 2 SIDES ____ PGS.

NUMBERING
FROM TO

BINDING/FINISHING 3 down left side
☒ COLLATING ☒ STAPLING □ ____ HOLE PUNCHED □ PLASTIC RING

FOLDING/ PADDING

CUTTING SIZE

Special Instructions
Math fronts and backs enclosed.

COPY CENTRE
OPER. NO. BLDG. MACH. NO.

DESIGN & PASTE-UP

OPER. NO.	TIME	LABOUR CODE
		D01
		D01
		D01

TYPESETTING

	QUANTITY			
PAP000000				T01
PAP000000				T01
PAP000000				T01

PROOF

PRF				
PRF				
PRF				

NEGATIVES

	QUANTITY	OPER. NO.	TIME	LABOUR CODE
FLM				C01
FLM				C01
FLM				C01
FLM				C01
FLM				C01

PMT

PMT				C01
PMT				C01
PMT				C01

PLATES

PLT				P01
PLT				P01
PLT				P01

STOCK

				001
				001
				001
				001

BINDERY

RNG				B01
RNG				B01
RNG				B01
MIS00000				B01

OUTSIDE SERVICES

$ ______
COST

TAXES – PROVINCIAL □ FEDERAL □

PERFORMANCE ANALYSIS AND FUNDAMENTAL PERFORMANCE TRADE OFFS FOR CLV OPTICAL DISKS

Stavros Christodoulakis
Daniel Alexander Ford

Research Report CS-88-06

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

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, 6].

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

*To appear in SIGMOD '88 Proceedings, June 1988 trade-offs of speed and storage; CAV type disks have faster access times while in general CLV type disks have greater storage capacity.

CAV disks have a lower capacity because they must compensate for the different speeds at which data stored in their tracks pass under the read head. They do this by elongating the recordings made on the outer tracks so that even though they move at a faster rate they have the same duration under the sense mechanism as the recordings made on the inner tracks. This means the tracks on the outer edge of the recording surface of CAV disks hold the same amount of data as do the inner tracks despite the greater physical length of the outer tracks.

CLV disks avoid this inefficiency by having a uniform recording density throughout the disk and then relying upon the drive's ability to vary the speed at which the disk rotates to ensure that all recordings, regardless of their position on the disk, pass beneath the sense mechanism at the same rate. This approach allows maximum utilization of the recording surface within the limits of the recording technology, but has the drawback that it makes movements of the read head or seeks on CLV disks slightly slower than on CAV disks. Reading the data on the disk requires the speed at which the disk rotates to match the position of the sense mechanism, but determining the position with the required accuracy is difficult without first being able to read the identification information stored in each sector. This "chicken and egg" problem can be solved by reading the identification information as the read head is moved incrementally across the surface of the disk, adjusting the rotation speed to match. Other delays come from problems in determining the exact start of a track as adjustments might be required for proper alignment.

A hybrid type of optical disk is available which combines the characteristics of both types of disks. The recording surface of the hybrid disk is divided into bands of tracks. All the tracks within a band are read at the same angular velocity, and in effect, form a miniature CAV disk. To allow a more uniform recording density (and therefore more capacity) through out the disk, the angular velocities differ from band to band, slower on the outer bands, faster on the inner. This approach attempts to increase the storage capacity over that available using the conventional CAV technique while at the same time reducing the amount of rotational speed adjustments required for a purely CLV scheme.

Currently, optical disk 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) [7] 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).

An analysis of the expected retrieval performance from CAV type optical disks (with or without a mirror) is provided in [3]. It gives exact and approximate results for determining the expected number of seeks for several different retrieval situations. In this paper we are concerned with modeling and analyzing the performance of CLV type optical disks, as well as describing fundamental data base performance tradeoffs.

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 objects[1], 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).

The next two sections presents a model of CLV optical disks, and an exact analysis of the expected number of seeks needed for the retrieval of a given number of objects from CLV optical disks. The analysis is general enough to include as a special case CAV disks. A lower bound on the computation cost required to compute an exact solution is also given.

Approximation techniques employing results derived for the CAV case that require less computation than the exact solutions are presented in the sections after that. They give results for both small (i.e. non-crossing and crossing objects) and large objects and discuss the effects of various factors on the accuracy of their results.

Finally, the last section discusses some fundamental data base performance tradeoffs for designing physical data bases (or file organizations) residing on CLV optical disks.

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

2. A Model and a Schedule

In this section we present an abstract model of optical disks which we will be using for analyzing their retrieval performance and give a physical description of CLV disks. We also present a schedule for answering a query and we show that it is optimal.

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, $i=1,2, \ldots ,T$ (to avoid discussion about boundary conditions we assume that track numbers are extended above T and below 1). Each track is composed of 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 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. 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.

When the access mechanism is moved a seek cost and a rotation delay cost are incurred as in the case of magnetic disks. The seek cost depends on the distance travelled by the access mechanism. Transferring a track of data from the device involves a track transfer cost as is also the case in magnetic disks. 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 (of the order of one millisecond) 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 remaining part of this paper. This model reduces to a model of magnetic disks 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 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 at a distance from the centre of the disk called the *principal radius,* 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.

The number of sectors in the *i'th* track is given by:

$$\left\lceil \frac{L_i}{l_s} \right\rceil$$

Where

$L_i = 2\pi r_1 + (2i - 1)\pi \, \Delta r$ - is the length of the *i'th* track.

l_s - is the length of a sector.

r_1 - is the principle radius.

Δr - is the distance between tracks.

The first sector of a track is by convention the first whole sector encountered after the intersection of the radial line and the track.

A scheduling algorithm for retrieving the data which qualify in a query is an ordered sequence of anchor points which 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 an optimal scheduling algorithm for retrieving data in this model. In this algorithm the access mechanism can be moving continuously in one direction. This algorithm 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 qualifying sectors.

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 [3].

The theorem shows that if a schedule satisfies the conditions A and B it results in a minimum number of spans.

If a schedule $A = a_1, a_2, \ldots, a_s$ satisfies $a_1 \leq a_2 \cdots \leq a_s$ the access mechanism is moved in one direction. Since each of the spans with anchor points a_1 and a_s contain at least one track with qualifying sectors, such a schedule will always result in a distance travelled which is within at most the length of two spans from the minimum possible distance that the access mechanism has to travel in order to retrieve all qualifying tracks. If, in addition, condition B of the theorem is satisfied, the schedule A will result in a distance travelled which is at most larger by the length of one span from the minimum possible distance travelled (because the tracks of the first span are between the first and the last qualifying tracks of the file). Thus, a schedule $A = a_1, \ldots, a_s$ which satisfies conditions A and B of the theorem, and in addition satisfies $a_1 \leq a_2 \cdots \leq a_s$, results in a minimum number of spans (number of moves of the access mechanism) and a minimum distance travelled (within the length of one span).

Note that if we change the definition of the anchor point to indicate the track with the largest number within a span, we can show that a schedule $Z = z_1, \ldots, z_s$, which satisfies conditions A and B

and also $z_1 \geq z_2 \cdots \geq z_s$, is also optimal. In fact, it can be shown that the two schedules A and Z will always result in the same number of spans, and that the distance travelled by the two schedules will, at most, differ by the length of one span. We call schedule Z a *backward* schedule and schedule A a *forward* schedule. The key observation for this proof is that the last span of A has to overlap (has some common tracks) with the first span of Z, at least at the last qualifying track. The second from the end span of A also has to overlap with the second span of Z, otherwise the last and the second from the end span of A would overlap. Thus, the second span of Z overlaps with the second from the end span of A at least in the anchor point of A. Continuing the induction this way, we find that the two schedules result in the same number of spans.

In the following sections we will only consider optimal schedules and we will derive cost estimates for various environments.

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. We present solutions for both cases below.

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 $\overline{K}$ the expected number of spans (the retrieval cost).

Then the exact result is given by:

$$\overline{K} = \sum_{K=1}^{\left\lceil \frac{T}{Q} \right\rceil} K \frac{arrange(1,N,T,K)}{\binom{M}{N}}$$

In this formula, *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. We give treatments of both cases below.

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:

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

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

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.

For crossing objects the definition of *arrange* is:

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

$$arrange(i,n,t,k) = nospans(i,n,t,k) + sncnc(i,n,t,k) + sncc(i,n,t,k)$$

Where

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

$$nospans(i,n,t,k) = arrange(i+1,n,t-1,k)$$

nospans(i,n,t,k) is the number of arrangements of n objects retrievable by k spans in the t tracks that begin with track i when none of the n objects occupy sectors in track i (and hence "nospan" starts at track i).

$$sncnc(i,n,t,0) = 0$$

$$sncnc(i,n,t,1) = 0, \text{ if } n > Ns_i$$

$$else = \sum_{j=1}^{min(Nt_i,Ns_i-(n-1))} \begin{pmatrix} Ns_i - j \\ n-1 \end{pmatrix}$$

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

sncnc(i,n,t,k) (*Non-continuing, Non-crossing*) is the number of arrangements of n objects retrievable by k spans in the t tracks that begin with track i when the span anchored at track i does not retrieve a continuing object from the previous span and does not have an object crossing the boundary of the last sector it covers into the next span.

$$sncc(i,n,t,k) = 0, \text{ if } i = 1,\ k = 0,\ or\ k = 1$$

$$sncc(i,n,t,k) = 0, \text{ if } t \leq Q\ (spansize)$$

$$sncc(i,n,t,k) = \sum_{l=1}^{min(n-1,Ns_i)} (\ scnc(i+Q,n-l-1,t-Q,k-1) + scc(i+Q,n-l-1,t-Q,k-1)\ )$$

$$\sum_{j=1}^{min(Nt_i,Ns_i-(l-1))} \binom{Ns_i - j}{l-1}$$

sncc(i,n,t,k) (*Non-continuing, Crossing*) is as for *sncnc(i,n,t,k)* above, except that an object does cross the boundary of the last sector covered by the span (requiring another span to retrieve it completely).

$$scc(i,n,t,k) = 0, \text{ if } i = 0,\ k = 0,\ or\ k = 1$$

$$scc(i,n,t,k) = 0, \text{ if } t \leq Q\ (spansize)$$

$$scc(i,n,t,k) = \sum_{l=0}^{min(n-1,Ns_i)} (\ scc(i+Q,n-l-1,t-Q,k-1) + scnc(i+Q,n-l-1,t-Q,k-1)\ ) \binom{Ns_i}{l}$$

scc(i,n,t,k) (*Continuing, Crossing*) is the number of arrangements of n objects retrievable by k spans in the t tracks that begin with track i when the span anchored at track i does retrieve a continuing object from the previous span and does have an object crossing the boundary of the last sector it covers into the next span.

$$scnc(i,n,t,0) = 0$$

$$scnc(i,n,t,1) = 0, \text{ if } n > Ns_i$$

$$else = \binom{Ns_i}{n}$$

$$scnc(i,n,t,k) = \sum_{l=0}^{min(n,Ns_i)} arrange(i+Q,n-l,t-Q,k-1) \binom{Ns_i}{l}$$

scnc(i,n,t,k) (*Continuing, Non-crossing*) is as for *scc(i,n,t,k)* above, except that no object crosses the boundary of the last sector.

The graph in Figure 1 below shows the exact results computed for a small example of a file with twenty tracks and up to seven objects.

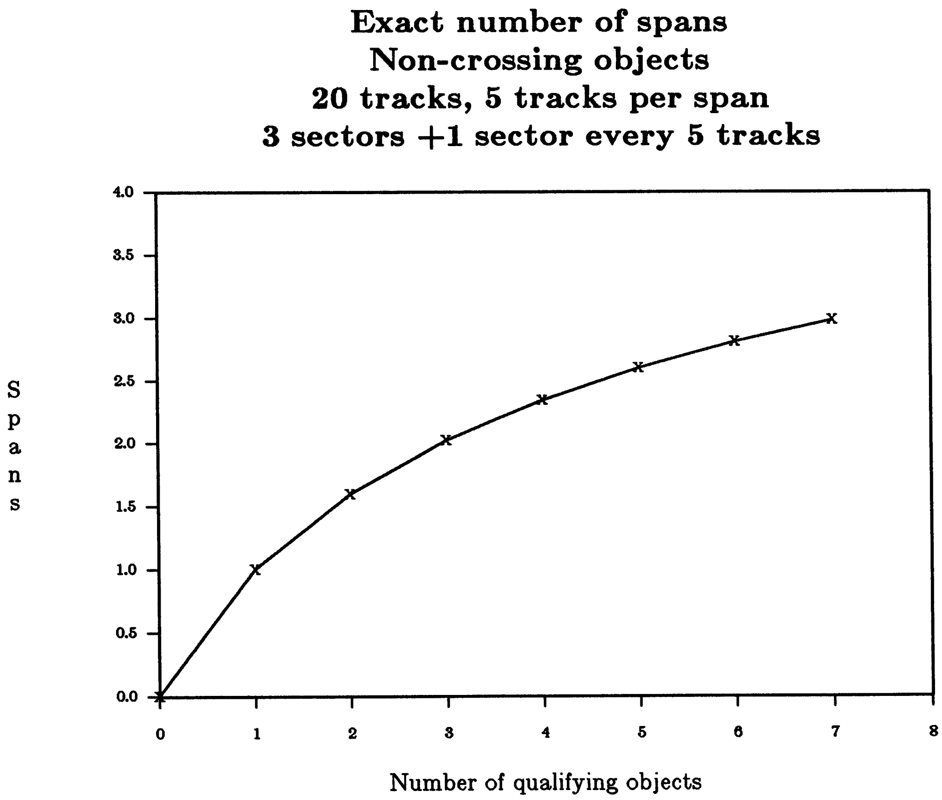

Figure 1.

3.3. Lower Bound on Computation Cost

The exact solution for the expected number of spans for both non-crossing and crossing objects is very efficient for a span size of one track, but for span sizes greater than this it is quite expensive to compute. The process examines every possible arrangement of the K spans on the T tracks. The number of such arrangements is a lower bound on the computation cost.

Ignoring the possibility of a span overlapping the end of the file, there are:

$$\binom{T - KQ + K}{K}$$

different arrangements.

To see that this is so, one can view an arrangement uniquely as a string of 0's and 1's. The ones represent the relative positions of the spans in the file and the zeros represent the uncovered tracks between them. The string "010001" represents a span anchored on the second track of a file followed by three uncovered tracks and then another span. The different number of such strings for a given number of 1's and 0's is the same as the number of arrangements of the spans in the file.

For a fixed number of 0's, the strings are represented uniquely by the index positions of the 1's. There are K such indices chosen from the length of the string. The string has $T - KQ$ 0's and K 1's and hence a length of $T - KQ + K$. The equation follows.

4. Approximate Solutions

The computational cost of the exact solution prohibits its practical application for large files. The possibility of deriving an alternate exact method having a lower computation cost seems unlikely. The variable track capacity of CLV disks means that the exact positions of the spans are required to know number of objects they could retrieve. Hence, it seems that any solution would still need to examine all arrangements of spans to produce an exact answer.

This problem does not exist in the analysis of CAV disks as their uniform track capacities mean that spans cover the same number of objects regardless of their position in the file. Such solutions have been derived in [3].

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 one to the other. 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 sectors[2]. For non-crossing objects, the graph in Figure 2 shows the two virtually coincident lines of the two bounds on the CLV solution indicating that it must be very close to the CAV solution. The graph for crossing objects is similar.

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.

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

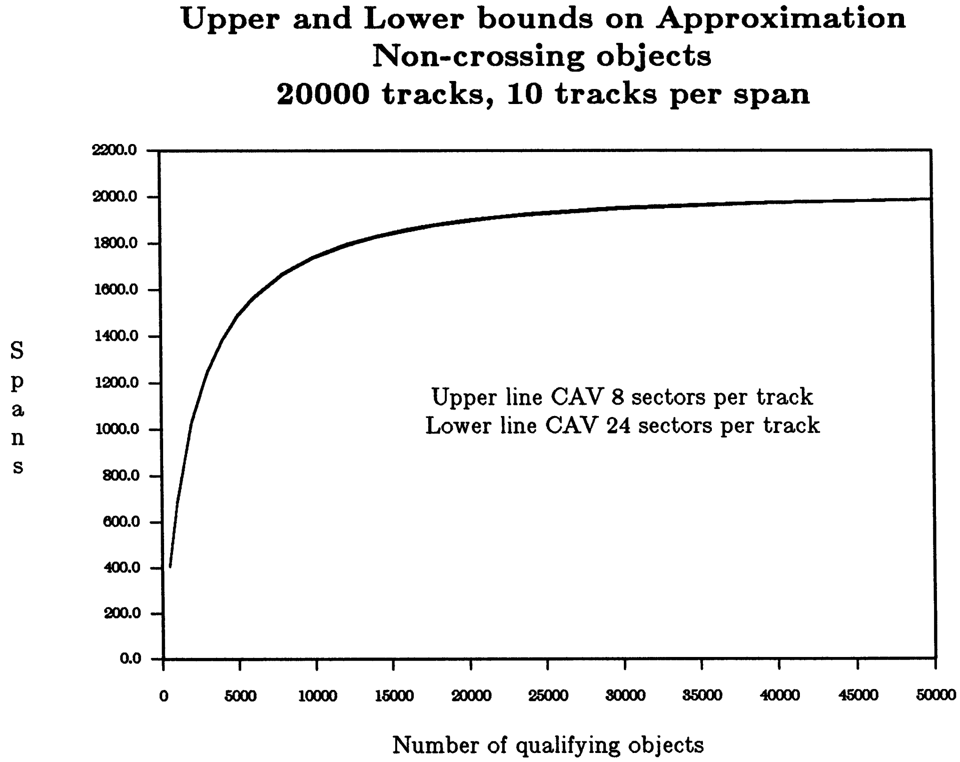

Figure 2.

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 $\overline{K_i}$ be the expected number of spans for subsegment i, and $\overline{N_i}$ be the expected number of objects that fall in subsegment i, then this is expressed as:

$$\overline{K} = \sum_{i=1}^{R} \overline{K_i}(\overline{N_i})$$

This method gives results for non-crossing and crossing objects comparable to those obtained above using a straight average track capacity. This is illustrated in the graph in Figure 3 below.

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.

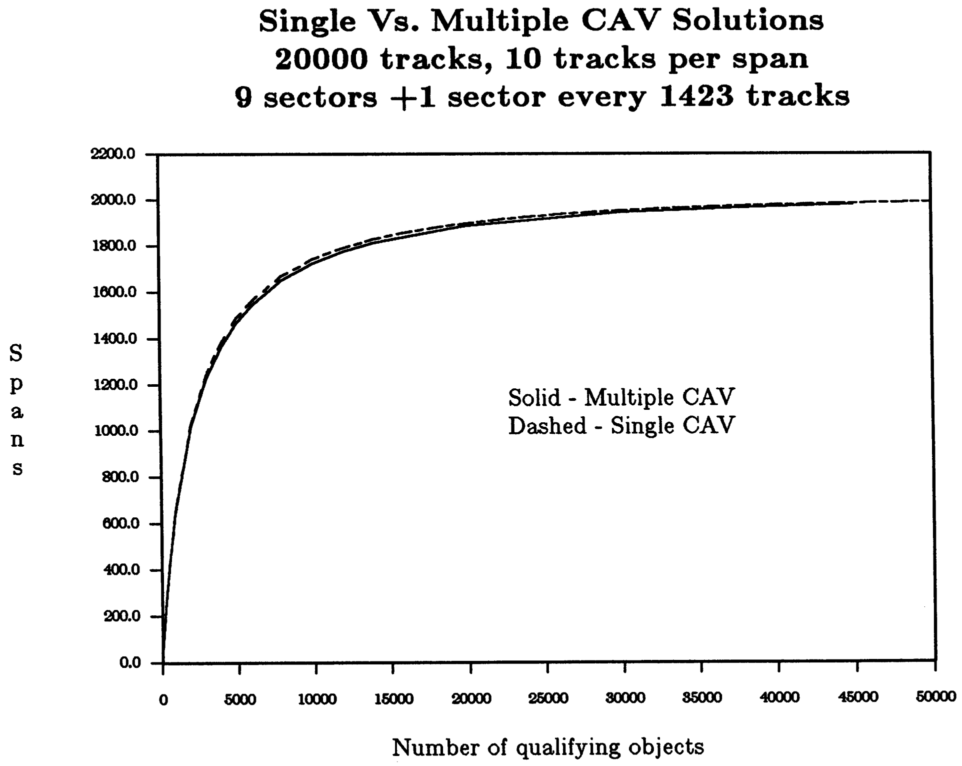

Figure 3.

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] :

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

Where $\overline{B}$ - 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 :

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

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.

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

Where

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

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 then 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 derived 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, $i=1,2,...,R$. Let N_i be the number of objects distributed to each segment $(\sum_{i=1}^{R} N_i = N)$ and $P(U) = P(N_1,N_2, \ldots, N_R)$ 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:

$$\sum_{U} P(U) \sum_{i=1}^{R} \overline{K_i}(U) + \Delta_R(U)$$

Where $\overline{K_i}(U)$ 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:

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

Where

$$\overline{S}_i(U) = \overline{K}_s \cdot N_i + S(N_i, T-(N_i \cdot \overline{K}_s \cdot Q), olength_i - (Q \cdot \overline{K}_s \cdot tlength_i))$$

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

and

$$\overline{K}_s = \left\lfloor \frac{olength_i}{Q \cdot tlength_i} \right\rfloor$$

$$olength_i = objectsize + \frac{tlength_i}{2}$$

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

$\overline{K}_s$ - is the number of spans that are completely occupied by an object in the *i'th* partition.

$olength_i$ - 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).

N_{file} - is the total number of objects that the file can contain.

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.

5.4. Comparison of Approximations

The results produced by the first approximation become more accurate as the resolution and the number of segments is increased and more and more accurate values for the track capacities are employed in the process. The results of the multiple CAV approximation improve as the number of objects increases and more accurate expected object distributions are used.

Errors in the first approximation can come from incorrectly estimating span capacity such that the size of a large object is between the actual capacity of a span and the estimated value. If the actual capacity of a span is below the estimated value then the span will erroneously be credited with the capability of completely containing one object that it cannot and thus for each object the number of spans is underestimated by at least one, and possible more. If the actual capacity of a span is above the estimated value, the opposite occurs and the number of spans is over estimated.

Increasing the resolution reduces the disparity between the two values giving better estimates for the required number of spans in each partition. Increasing it far enough will reduce the disparity to zero at which point other much smaller errors in the approximation will take precedence. One such source being omission of spans that cross segment boundaries. A resolution level of 4 (16 segments) is sufficient to completely encompass all of the track capacities found on a CD-ROM disk.

In the multiple CAV approximation, the main sources of error are inaccurate expected object distributions and the omission of the possibilities of spans that cross segment boundaries. The first becomes smaller as the number of objects increases and the second becomes smaller when the size of a segment is large compared to the size (number of tracks) of a span and becomes zero when the span size is one track.

A combinatorial explosion in the number of possible object distributions and a corresponding dramatic increase in the computation cost occurs in the first approximation technique when the number of objects and the level of resolution of the approximation is increased. To reduce the computation cost, each level of the first approximation can employ some of the computational results of the previous level. Also, since the approximation becomes progressively better the process can be stopped whenever successive approximations are near. The multiple CAV technique has none of these drawbacks, its computation cost is constant.

The graphs in Figures 4 and 5 show the results of the two approximations for objects of size 250k and 220k at different resolutions. They illustrate the marked changes obtained in the results of the first approximation as its resolution is varied and compare the accuracy of both.

The lower line on both graphs corresponds to resolution level 0 and a track capacity of 16 sectors. This gives a span capacity of 327680 bytes allowing all spans to completely contain either object. Increasing the resolution to level 1 gives two segments with track capacities of 12 and 19 sectors. A track capacity of 12 sectors gives a span capacity of 245760 bytes which cannot contain an object of size 250k but can still contain an object of size 220k. The effect of this change can be seen on the two graphs. In Figure 4, the line for resolution level 1 shows an increase in the number of required spans. The same line in Figure 5 is coincident with that for resolution level 0.

At resolution level 2, we see further changes in the results. The track capacities are 10, 14, 18, and 21 sectors. The lowest of which corresponds to a span capacity of 204800 bytes and cannot contain an object of size 220k. The second lowest track capacity corresponds to a span capacity of 286720 bytes which can contain an object of size 250k. We find that in the graph for object size 250k in Figure 4, the line for the level 2 solution shows a lower number of expected spans as now three of the four estimated span capacities can contain a 250k sized object. In the graph for object size 220k in Figure 5, the line for the resolution level 2 solution is above that of levels 0 and 1 since now one of the four span capacities is less than the object size of 220k resulting in more spans being required. In fact, the relative positions of the two object sizes with respect to the span capacities at resolution level 2 are the same giving virtually identical solutions for the two sizes.

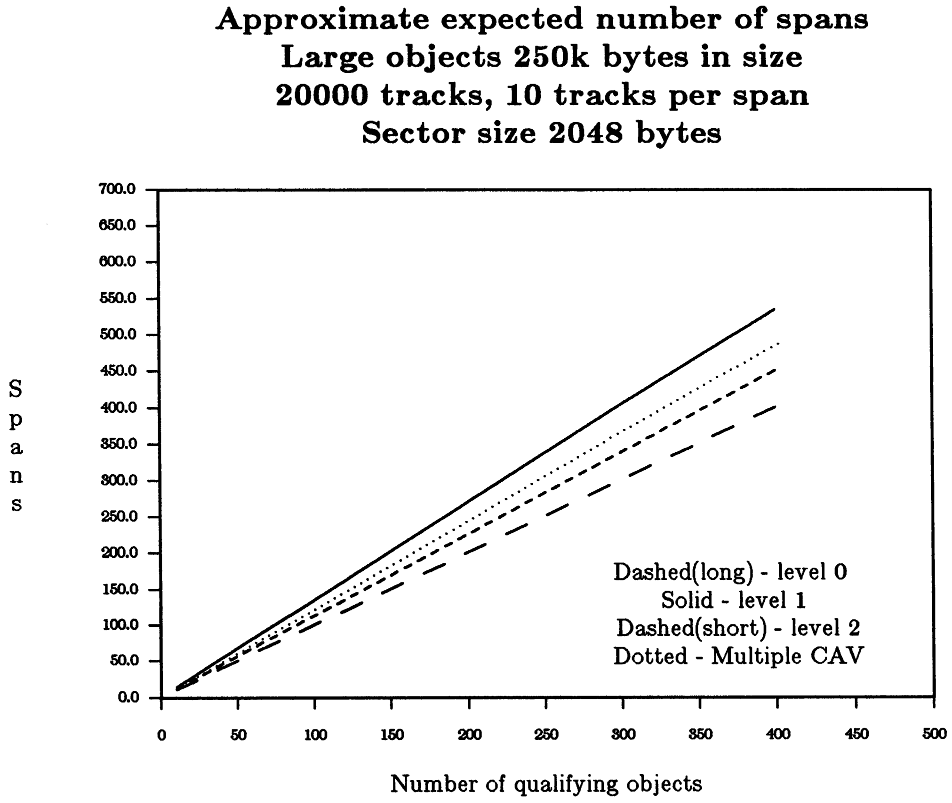

Figure 4.

Further increases in the resolution cause similar but smaller changes in the solution as the estimated span capacities move closer to their actual values.

The lines depicting the results for the multiple CAV approximation on the two graphs agree very well with the higher resolution, more accurate, results of the first approximation. Both results of the first approximation are converging to the those of the multiple CAV approximation.

This agreement, and the lower computation cost of the multiple CAV approximation, lead us to conclude that it is the method of choice for approximating the retrieval cost of the objects when they are large compared to the capacity of a span.

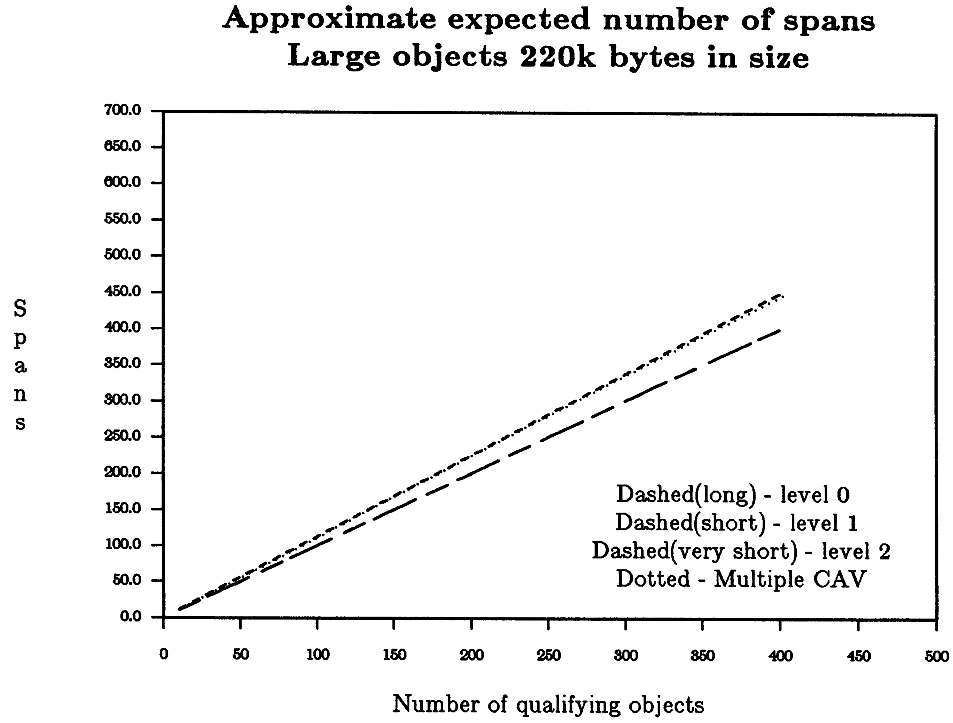

Figure 5.

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, and N the selectivity of a query. 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 6 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.

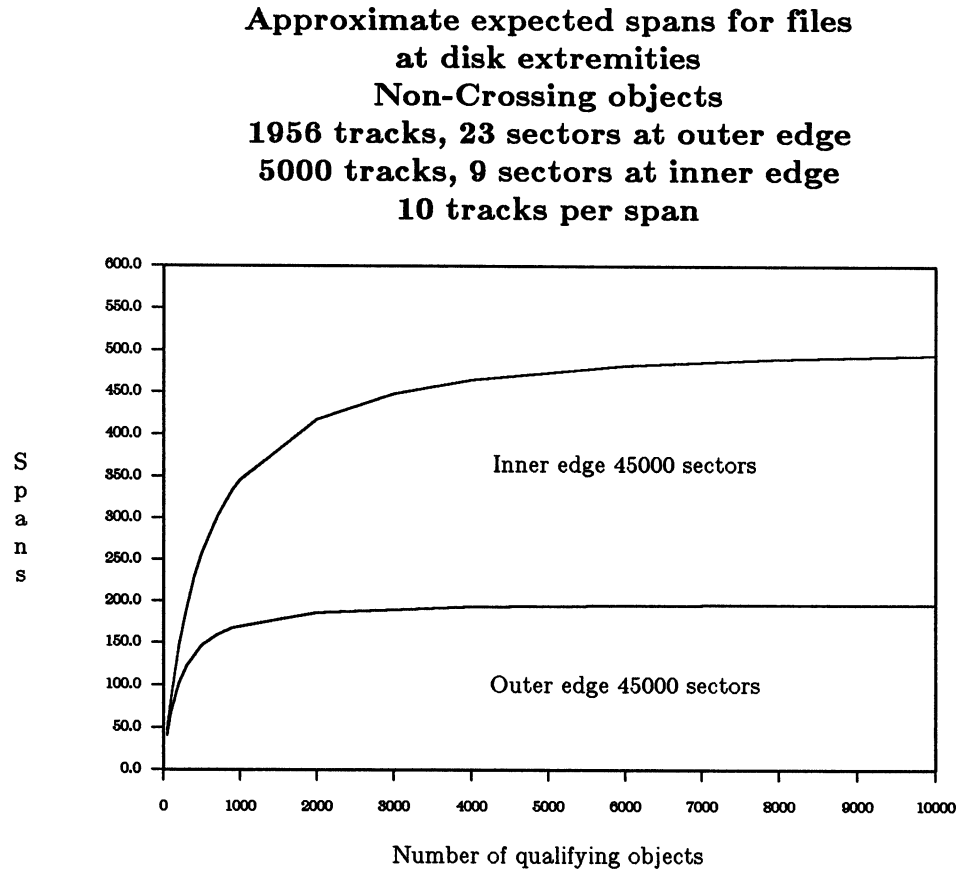

Figure 6.

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:

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

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:

$$f_1 \cdot tracks(M_1 \cdot N_1, g_1, loc) + f_2 \cdot tracks(M_2 \cdot N_2, g_2, loc_2) \;<\; f_2 \cdot tracks(M_2 \cdot N_2, g_1, loc) + f_2 \cdot tracks(M_1 \cdot N_1, g_1, loc_1)$$

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 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 retrieved 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.

We are currently pursuing work along several fronts related to data base performance for optical disk based data base applications. The results presented in this paper are basic to all such studies.

8. 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., “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] Hatzilemonias, K., “Minos Optical Disk Simulation Package (MODS)”, User’s Guide, Version 3.0, University of Waterloo, 1987.

[6] Maier, D., “Using Write-Once Memory for Database Storage”, *Proceedings ACM PODS 82*, 1982.

[7] OPTIMEM1000, “Optical Disk Drive (OEM MANUAL)”, Optimem, 435 Oakmead Parkway, Sunnyvale CA 94086.