Disk Partitioning Technique for Reducing Multimedia Access Delay

Paul Bocheck, Henry Meadows, and Shih-Fu Chang

Department of Electrical Engineering and Center for Telecommunication Research
Columbia University, New York, N.Y. 10027

Abstract

With the introduction of future multimedia services such as HDTV, digital systems able to store and retrieve a huge amount of video and audio information will play an important role. One of the key problems is the multimedia data access optimization. In this paper we compare several techniques proposed for storage of multimedia information on the hard disk and introduce a Disk Partitioning Technique allowing an increase in the number of concurrent users while minimizing the required buffer size. Also the support for interactive control such as pause or reverse is demonstrated.

Keywords: multimedia retrieval, disk storage, video server

I. Introduction

Problem of storage and retrieval of delay sensitive information recently received a lot of attention. This paper investigates a problem of efficient placement and retrieval of multiresolution video streams (HDTV) under constraints of real-time and interactive service. Real-time retrieval generally involves two issues. The first addresses the actual physical placement of data on the medium and the second the retrieval scheduling algorithm. The basic principles and definitions of delay sensitive data retrieval and some existing placement strategies were described by Gemmell and Christodoulakis [1] with some extensions to multichannel playback. Rangan and Vin in [2,3] presented issues involving design of multiuser HDTV storage server. They proposed a constrained block allocation mechanism as the efficient way to represent and store multiple video streams on disk. Chen, Kandlur and Yu in [4] and Gemmell [5] independently developed functionally equivalent algorithms based on grouping streams into independent sets to reduce buffering requirements. Reddy and Wyllie [6] proposed a new disk scheduling technique combining SCAN seek optimization and EDF (Earliest Deadline First) algorithms.

In this paper we focus on disk storage and analyze performance of several proposed disk systems. We introduce a new Disk Partitioning Technique suitable for storage of interactive video sequences. This work is part of Columbia's Video on Demand prototyping project in the Image and Advanced TV Laboratory [7].

II. Multimedia retrieval model

At this moment, we focus on the following configuration: single multimedia server with multiple requesters (users). We assume that all users have same resolution requirements and therefore same playback rates r_p. The users are interactively retrieving multimedia streams consisting of a sequence of compressed video frames. There is no assumption about the coding technique used and no restriction about start time of each individual video stream and user interaction. The optimization objective is the overall cost performance ratio, e.g., to support maximum number of users at the minimum buffer cost.

We define *retrieval cycle* (in short *cycle*) to be the fixed period during which all serviced streams retrieve *one* unit of information u_i (also called *block*) containing enough data to satisfy continuity requirement of each particular stream. In order to satisfy playback continuity at the receiver, time for retrieval of each information unit must be less or equal to its playback. We can formulate this for the case of one disk drive and random block allocation:

$$\sum_{i=1}^{s} \left(t_{sx} + t_{rx} + u_i / r_d\right) \le \frac{u_i}{r_p} \qquad \text{(Eq. 1)}$$

where s is maximum number of playback streams, t_{sx} is maximum seek time, t_{rx} is a maximum rotation time, u_i is a size of information unit (*block*), r_d and r_p are disk reading and playback rates respectively. From the above expression we can readily see that for relatively small units of information, major limiting factors for the concurrent retrieval are actual physical storage parameters. Maximum seek, rotation times, and retrieval data rate depend on particular device type. Influence of physical parameters tend to disappear for large u_i. On the other hand the size of information unit is limited by buffer availability and also bounded by delay constraint. Having fixed u_i, we will try to service as many users as possible.

III. Disk Partitioning Technique (PAR/CSCAN)

Following is a description of a proposed system utilizing constrained disk placement technique. Suppose that single disk surface is divided into several circular partitions *1*, *2*,...,n_p (see Figure 1). Each partition is then further divided into blocks of fixed size u_i. As an example, each block can contain multiple video frames. Then each stream can be represented as a sequence of blocks b_0, b_1, ..., b_M. Consecutive blocks b_{i-1}, b_i, b_{i+1}, of the stream will be written into different partitions in such a way that during each *scanning cycle*, the head will scan all partitions in the same direction retrieving consecutive blocks of particular streams from each partition. Upon reaching the end of the last partition n_p, the head will move back scanning the disk in reverse direction. Since consecutive blocks of streams, stored at different partitions follow predictive pattern, streams will be retrieved in the fixed order during each cycle. This is the reason for eliminating double buffering requirement compared to random placement, where order of blocks during scanning cycle is not predictable. It is interesting to point out that blocks inside each partition grow from both sides toward the center of the partition. Figure 2 depicts this block allocation policy for different number of partitions.

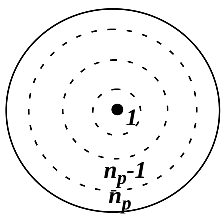

Figure 1. Disk partitioning

As an example, assume playback of one stream from the disk with four partitions (see Figure 2b). At the beginning, the head will start moving from the center of the disk reading block 1 from partition 1 and continue to block 2 from partition 2, until the block 4 from partition 4. At the end of the fourth partition, the head direction will reverse and block 5 will be read. Following will be retrieval of block 6 from partition 3, until the block 8 will be read from the partition 1. Then the cycle will repeat.

Let’s focus now on the retrieval of multiple streams. As we already pointed out, since the head is always scanning the disk in one particular direction until it reaches the end of the last partition in its direction (circular SCAN algorithm), multiple streams must be read during the scan of each partition. This is accomplished by synchronizing the retrieval of multiple streams in such a way that the request for the start of additional stream will be intentionally delayed (see Figure 3) until the disk head scans the partition containing the first block of the new request. From that point, the additional block read corresponding to new stream will be performed during the scan of each partition in the described way.

	Partition #1	**Partition #2**
a.	**1,5,9,13,.............16,12,8,4**	**2,6,10,14,............15,11,7,3**

	Partition #1	**Partition #2**	**Partition #3**	**Partition #4**
b.	**1,9,.......16,8**	**2,10,.....15,7**	**3,11,....14,6**	**4,12,.....13,5**

Figure 2. Partitioning Block Allocation

The artificial start-up delay introduced at the beginning of the new request can be eliminated by beginning the retrieval of the new stream right from the very next partition. Then the retrieval can synchronize on either past or future blocks. Even though this will cause addition or loss of few beginning blocks (frames) as shown in Figure 4, such synchronization of the retrieval will introduce only unnoticable effect to user comparing to the advantage of faster interactive start-up delay. Figure 4 depicts this improved synchronization mechanism where reverse playback starts from blocks not retrieved yet. In both examples reverse request is started right from the next partition. Note that the reverse retrieval request, after initial synchronization, is undistinguished from the forward playback stream retrieval.

reverse request
for stream 1

(a) 2 partitions
Stream 1: 1 2 3 4 5 6 7 8 9 10 ***6 5 4*** ...
Stream 2: 1 2 3 4 5 6 7 8 9 ...

playback request
for stream 2

(b) 4 partitions
Stream 1: 1 2 3 4 5 6 7 8 9 10 11 12 ...
Stream 2: ... 9 10 11 12 13 14 ***10 9 8 7 6 5*** ...

reverse request
for stream 2

Figure 3. Multiple stream synchronization

reverse request
for stream 1

(a) 2 partitions
Stream 1: ...2 3 4 5 6 7 ***9 8 7 6 5 4 3*** ...
Stream 2: 1 2 3 4 5 6 7 8 9 10 ...

playback request
for stream 2

(b) 4 partitions
Stream 1: 1 2 3 4 5 6 7 8 9 10 11 12 ...
Stream 2: ... 9 10 11 ***13 12 11 10 9 8 7 6 5*** ...

reverse request
for stream 2

Figure 4. Improved synchronization

V. Analysis

In this section, we present models of several block placement and seek optimization algorithms and compare their buffer requirements and performance. For our comparative analysis we use high performance disk (Table 1: IBM 3390, Model 1 [8]). We assume block size k (number of sectors) to be chosen as one of system parameters during the multimedia system design.

Table 1: Disk storage (IBM 3390 [10]) and retrieval parameters

Parameter	Value	Description
t_{sx}	**18 ms**	**maximum seek time**
t_{sm}	**1.5 ms**	**minimum seek time**
t_{rx}	**14.2 ms**	**maximum rotation time**
r_d	**33.6 Mbps**	**maximum disk transfer rate**

Table 1: Disk storage (IBM 3390 [10]) and retrieval parameters

Parameter	Value	Description
c_s	**512 bytes**	**sector capacity**
r_p	**1.5 Mbps**	**playback rate**
n_p	**4**	**number of partitions**
l_{min}	**0 ms**	**scattering parameter, see [3]**

The analysis of performance characteristics will include maximum number of concurrent streams (*s*), utilization (ρ), and the size of memory buffer (*b*) required for uninterrupted playback.

Contiguous block placement with SCAN (CON/SCAN)

In this technique, the blocks are written on the disk as one contiguous sequence. Multiple sequences (movies) will be written one after each other. The retrieval cycle consists of two phases. During the first one, head scans the disk starting from the inner most track until it reaches the outermost track. While scanning the disk, the data blocks, belonging to different streams are read from the disk. Upon the reaching the outer most track the head is returned back to its initial position without reading any data. The continuity requirement for this technique (Eq. 1) can be then rewritten as follows:

$$kc_s/r_p = s\left(kc_s/r_d + t_{rx} + t_{sm}\right) + 2t_{sx} \qquad \text{(Eq.2)}$$

In obtaining the (Eq. 2) the following assumptions were made: any stream accessed during the first phase will add to the total retrieval cycle the maximum rotation time t_{rx}, time to read the block of size k sectors and minimum seek time t_{sm} used as an approximation to the head positioning. Finally, since the retrieval cycle consists of two phases of head movement, we have to add to the total cycle time two maximum seek delays t_{sx}. The equation for maximum number of supported streams s (Eq. 3) can be then readily obtained from (Eq. 2). Defining utilization as $\rho \equiv sr_p/r_d$ we can also express the maximum utilization:

$$s = \left(r_d/r_p\right)\frac{1 - 2\frac{r_p}{kc_s}t_{sx}}{1 + \frac{r_d}{kc_s}\left(t_{rx} + t_{sm}\right)} \qquad \rho = \frac{1 - 2\frac{r_p}{kc_s}t_{sx}}{1 + \frac{r_d}{kc_s}\left(t_{rx} + t_{sm}\right)}$$

$$b = k = \frac{1}{c_s}\left[\rho r_d\left(t_{rx} + t_{sm}\right) + 2r_p t_{sx}\right]\frac{1}{(1-\rho)} \qquad \text{(Eq.3)}$$

Noting that for scan technique we have $b = k$, one can easily obtain the memory buffer requirement *b*. Graphs, corresponding to *s*, ρ, and *b* are depicted in Figures 5, 6, and 7 respectively.

Contiguous block placement with circular SCAN algorithm (CON/CSCAN)

This technique is very similar to *CON/SCAN* with an exception that data is read in both directions of the disk head movement. Retrieval cycle consists of only one phase during which data will be read from the disk. Note however, since the video sequence is placed in contiguous fashion and blocks from different streams are not read in a fixed order, we need to double memory buffer to support uninterrupted playback. This requirement can be written as: $b = 2k$. Also, the basic continuity requirement (Eq. 1) can be described as:

$$kc_s/r_p = s\left(kc_s/r_d + t_{rx} + t_{sm}\right) + t_{sx} \qquad \text{(Eq. 4)}$$

Scattered block placement (SCA)

Scattered block placement technique introduced in [3] performs well during the playback of synchronized streams. Low performance retrieval of interactive streams is caused by not using the scan technique during the retrieval. Instead, maximum seek and rotation time is assumed during the switch between different streams. Also, application of scattering to playback of unsynchronized streams lead to increase of cycle time due to inefficient use of scattering parameter l_{min}. Therefore, for our analysis we assume $l_{min} = 0$ *ms*. With previous assumption, the scattered block placement will then transform to random block placement with random access. Choosing l_{min} greater than *0* would cause even further decrease in number of supported streams. The continuity requirement equation can be expressed as:

$$kc_s/r_p = s\left((k-1)c_s/r_d + t_{rx} + t_{sx} + (k-1)l_{min}\right) \qquad \text{(Eq. 5)}$$

Random block placement with SCAN algorithm (RAN/SCAN)

Random block placement with SCAN algorithm implies the need for double buffering, since records can be accessed in any order during the cycle time. The continuity requirement will be the same as Eq. 2.

Grouped Sweeping Scheme (GSS)

Based on the number of simultaneous streams GSS scheme effectively combines round-robin and SCAN scheduling techniques. Dividing streams into several groups can reduce buffering requirements. It was concluded in [4] that for the large number of streams this technique tends to converge into the SCAN disk scheduling. Since we are interested only in the maximum number of simultaneous streams, for our analysis the GSS scheme is equivalent to RAN/SCAN algorithm.

Disk Partitioning Technique block placement with circular SCAN algorithm (PAR/CSCAN)

In PAR/CSCAN placement technique the retrieval cycle consists of time reading blocks in single partition plus the time to move head over this partition. The later is reduced n_p times and can be expressed as: t_{sx}/n_p. Also, since blocks on the disk are stored in fixed, prearranged fashion, double buffering is not required. The continuity requirement equation can be expressed as:

$$kc_s/r_p = s\left(kc_s/r_d + t_{rx} + t_{sm}\right) + t_{sx}/n_p \qquad \text{(Eq. 6)}$$

Utilization (ρ) and buffer requirement (b) can be derived in a way similar to that in Eq. 3.

Figures 5, 6 and 7 compare performance parameters of discussed playback techniques. The actual parameters were used from Table 1 with exception of maximum rotational delay $t_{rx} = 0.2$ *ms*. Choice of this value can be justified for block sizes of multiple tracks in which case data retrieval

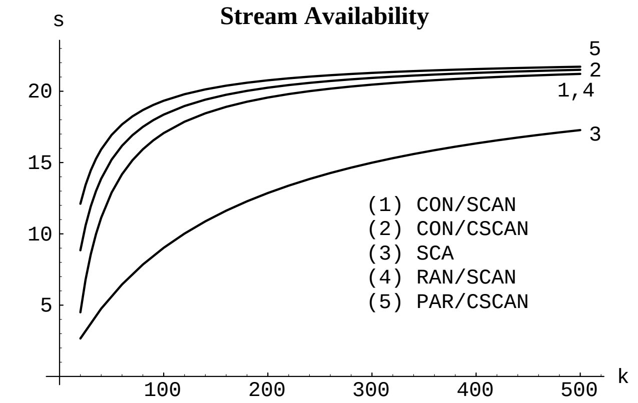

Figure 5. Variation of maximum number of simultaneous streams on buffer size

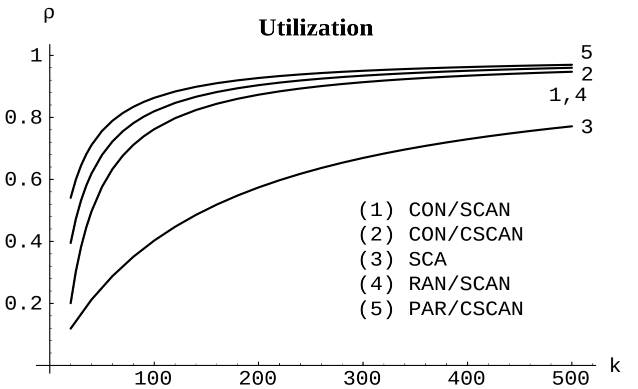

Figure 6. Variation of disk utilization on buffer size

could start right from the next sector and continue for multiple tracks. Figure 5 depicts the maximum number of simultaneous streams versus the block size. It clearly shows advantage of circular scan algorithms for both partitioned and random block placement. The better performance compared to simple scan technique can be explained by shorter read cycle time due to ability to read in both scanning directions. Low utilization of constrained block placement algorithm [3] is due to its assumption of maximum seek and maximum rotation delay between independent stream retrieval. Even though this particular technique can be successfully used for retrieval of multiple synchronized streams where scattering parameter can be correctly applied, it is inefficient for retrieval of unsynchronized interactive streams. The best performance of partitioned block placement technique is due to the reduced cycle time by means of separate partitions where less time is spent on moving the head during each cycle. The improvement can be observed especially for block size less than 100 corresponding roughly to one track. As it was already pointed out, using buffer size of multiple tracks has another potential advantage in reducing the rotation time. Some disk drives are able to start read and buffer the data as soon as the head crosses the next sector. This technique will almost eliminate the rotation time for the buffers of size multiple tracks. Figure 6 depicts more general dependence of disk transfer rate utilization on buffer size. Figure 7 depicts the buffer requirement versus utilization. Since both PAR and SEQ/SCAN do not require double

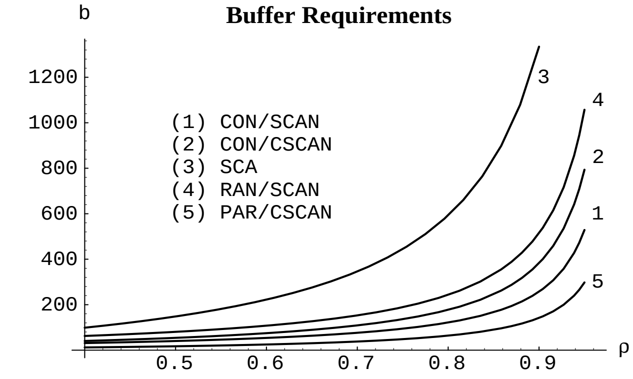

Figure 7. Variation of buffer requirements on utilization

buffering for continuous playback they are performing better than SEQ/CSCAN and RAN/SCAN techniques. The largest buffering is required for constrained block placement. From the above we can conclude, that for specific application such as playback of video streams arrangement of data on the disk plays an important role. Overall the continuous block allocation provide better performance than random block placement or scattered block placement. The Disk Partitioning Technique shows the best performance in both maximum number of supported streams and buffering requirements.

V. Conclusions

The efficient placement and retrieval of video streams and images is of high importance. In this paper requirements for multimedia servers were identified and new constrained block placement method was presented, analyzed and compared to others published in literature. Overall the Disk Partitioning Technique supports interactive functions such as pause or reverse and achieves higher stream availability and lower buffer requirement.

References

[1] J. Gemmell, and S. Christodoulakis, "Principles of Delay-Sensitive Multimedia Data Storage and Retrieval", ACM Transactions on Information Systems, vol. 10, No. 1, January 1992, pp. 51-90

[2] P. V. Rangan, H. M. Vin, "Efficient Storage Techniques for Digital Continuous Multimedia", IEEE Transactions on Knowledge and Data Engineering, August 1993

[3] H. M. Vin, P. V. Rangan, "Designing a Multi-User HDTV Storage Server", IEEE Journal on Selected Areas in Communications vol. 11, No. 1, January 1993

[4] Mon-Song Chen, D. D. Kandlur, P. S. Yu, "Optimization of the Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams", ACM Multimedia 93

[5] D. J. Gemmell, "Multimedia Network File Servers: Multi-channel Delay Sensitive Retrieval", ACM Multimedia 93

[6] A. L. N. Reddy, J. Wyllie, "Disk scheduling in a multimedia I/O system", ACM Multimedia 93 1993

[7] S. -F. Chang, A. Eleftheriadis, and D. Anastassiou, "Some Interoperability Issues in Columbia's Video on Demand Testbed", Contribution to International Digital Audio/Visual Council, April 1994

[8] IBM 3390 DASD High-performance, large-capacity storage solutions, G221-2703-03, 11-93