Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

A Frame Converting Strategy Based on Compensatory Region-based Data Placement Scheme for Interactive Video Servers

Shiao-Li Tsao Yueh-Min Huang Shiang-Chun Liou Yih-Wei Liang*
Department of Engineering Science
National Cheng Kung University, Tainan, Taiwan 701 R.O.C
Industrial Technology Research Institute, Computer & Communication Research Lab. *
E-mail : sltsao@system.es.ncku.edu.tw

Abstract

Of late, zoning technique has been applied on disks to increase their capacities. An effect of disk-zoning is that a disk has higher throughput (bandwidth) in outer zones than in inner zones. To efficiently utilize various bandwidths from zones, we propose a compensatory retrieval method based on region-based data placement scheme. Due to the same retrieving of a data block, we can read more data in the outer zones to compensate for the deficit read in the inner zones, which can significantly enhance the disk throughput. From the simulation results and the performance testing of the prototype, it can be seen that our proposed method can improve disk throughput by 25%-35% compared to the original region-based data placement. In addition, we design a frame converting strategy to reconstruct the MPEG data. With the frame converting technique, the compensatory retrieval method can facilitate the interactive operations of video playout for a video server

1. Introduction

Recently, the development of video servers has been growing rapidly. There are many issues on developing a video server, for example, how to store the large amount of the video data, how to provide large and guaranteed disk and network bandwidth, and how to retrieve and transfer the video data in the real-time manner. Focusing on the storage subsystem of a video server, we find that one of the challenges of designing the storage subsystem is providing large and guaranteed bandwidth. To achieve this requirement, the issue of the load balance, the parallelism and the concurrence of multiple hard disks, and the high utilization of disk bandwidth must be considered[8]. The data placement and the disk scheduling can be used to improve disk throughput[1][4][5]. A new technique called disk zoning was proposed to increase the disk capacity. A zoned-disk contains several zones in which a zones is a group of continuous tracks with the same sectors/track ratio[3]. Another effect of disk zoning is that, since a hard disk spins at a constant angular velocity, the disk has higher bandwidth in outer zones than in inner zones[2]. Previous studies on region-based data placement do not realize this problem and do not use the various bandwidths from different zones[6][9][12] [13]. In [11], a solution was presented to group complementary tracks into track pairs, and blocks are stored on the basis of track pair. Thus, the average bandwidth can be obtained. However, one additional seek is introduced while accessing a block since a block is further partitioned into two sub-blocks and stored on complementary tracks. In order to utilize the bandwidths from zones, we design a compensatory retrieval method based on region-based data placement to efficiently utilize the zoned-disk bandwidth. In addition, we design a frame converting strategy to reconstruct video data. Thus, the proposed method can also facilitate fully interactive operations of video playout.

The compensatory retrieval method not only limits the seeking boundary but also uses the advantage of disk zoning. According to the simulation results and the performance testing of the prototype, our method can improve the disk throughput by 25%-35% compared to the original region-based data placement schemes. Our method can also provide the interactive operations of video playout. The total waste of the disk space is less than 5% of the original space, which is inconsiderable compared to the improvement of the disk throughput.

The rest of this paper is organized as follows. In Section 2, we elaborate the compensatory retrieval method based on region-based data placement scheme. In Section 3, we present a frame converting technique to extend our retrieval method to support interactive operations of video playout. In Section 4, the simulation results are described and discussed. Finally, we conclude the study in Section

5.

2. Compensatory retrieval method

2.1 Region-based data placement and problem description

The main benefit of region-based data placement is that the seeking distance of the disk head is bounded in a region of the disk rather than the whole one. The first step of region-based data placement is that a disk is partitioned into several regions. Then, a video file is divided into blocks which are distributed onto each region. The requests from clients are only admitted when the disk head locates on the region with the desired block. Therefore, the disk head is confined within a region while serving every user, and the seek time can be minimized. However, if a request is made for the data that is not on the current serving region, the request is queued until the disk head moves to the region with the desired block, and the response time becomes longer.

In general, the sequence of storing blocks in the region-based data placement scheme can be classified into two major types. One is C-SCAN-like data placement, another is SCAN-like data placement. Figure 1 illustrate the SCAN-like data placement scheme.

Vi,j:means block j of CM file i

	Region 1	Region 2	Region 3	Region 4	Region 5
Disk 1	V1.1,V2.1,.. V1.10,V2.10	V1.2,V2.2,.. V1.9,V2.9,..	V1.3,V2.3,.. V1.8,V2.8,..	V1.4,V2.4,.. V1.7,V2.7,..	V1.5,V2.5,.. V1.6,V2.6,...
Disk 2	V1.11,V2.11	V1.12,V2.12	V1.13,V2.13	V1.14,V2.14	V1.15,V2.15

Sequence of CM block placement

Figure 1 The stored sequence of blocks for SCAN-like data placement

However, studies of region-based data placement all assume that the size of a retrieved block is the same; so is the disk bandwidth from different regions. Under these assumptions, the reading time for each block should be the same. Based on those concepts, the ideal service round for each region is shown in Figure 2. Here, we define that the total time spent for seeking blocks and retrieving blocks for all users once is called a service round. In that figure, we need to seek the proper track, rotate to the exact sector, and read data from the disk to the memory. The amount of data read must be large enough to maintain the continuity of the video playback for every client. In fact, zoned-disks have various bandwidths from zones. For the same amount of retrieved data, the disk head will spend less reading time in outer zones than in inner zones. To maintain the continuity of video playback, the maximum number of users that the server can support is bounded by the bandwidth of the innermost zone. The upper part of figure 3 illustrates the actual situation. As a result of the fact, the disk bandwidth of other zones on the disk is not fully utilized. This waste becomes more serious when the difference of bandwidths between inner and outer zones becomes large.

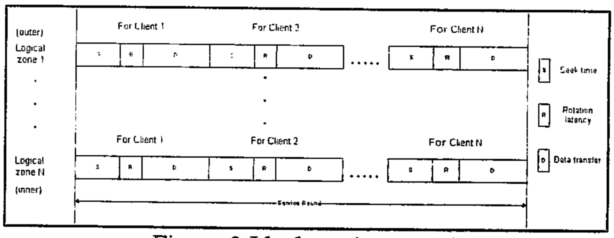

Figure 2 Ideal service round

2.2 Basic concept

As mentioned above, there are several alternatives to place blocks on the regions of a disk. Our method is applicable for any region-based strategy. In this section, we take SCAN-like data placement of Figure 1 as an example to describe the basic concept of the compensatory retrieval method. We use multiple independent hard disks as a disk array, and apply the coarse-grain striping strategy to the disk array. First, hard disks are partitioned into several logical zones. A logical zone is defined as a group of continuous tracks, which is similar to the region defined in [12] and the cluster indicated in [6]. Each logical zone has the same number of tracks and may cover several physical zones. The physical zone is an area on a hard disk with the same bandwidth. We partition the video files into blocks of different sizes and place those blocks onto logical zones on multiple disks. Here, the video block is defined as the unit of one time access. We let the block size be larger in outer logical zones than in inner logical zones. The sequence of data placement is like the strategy shown in Figure 1.

Then, we consider the retrieval process. The retrieval process is simplified by ignoring the overheads of the operating system and the network delivery in our current study. We also assume that the consumption rate of a video stream is a constant. The basic concept behind our proposed scheme is that the block size of one time access for a client is fixed multiple tracks. The number of tracks in a block is the same in every logical zone.

We reduce the amount of data read in inner logical zones, compared to the original region-based data placement scheme, and thus the number of users accommodated will be increased in a fixed service round. However, the problem is that the amount of data read in inner logical zones may not be enough to keep the continuity of video playback. For the same reading time, the same number of tracks but more data can be read in outer logical zones to compensate for the consumption by clients when the data in inner logical zones is read. To accomplish the process, a buffer is thus needed to temporarily store the data used for the compensation since the original idle time in outer zones is used to read ahead more data. Consequently, the actual service round is shown on the upper part of Figure 3. Our strategy is shown on the lower part of the same figure. The partition of hard disk, the block size in different logical zones, and the analysis of the buffer management are described in the ensuing section.

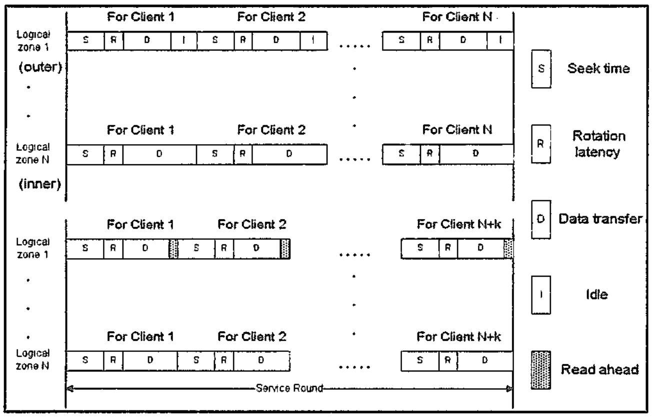

Figure 3 Basic concept of our proposed strategy

2.3 Partition logical zone

A logical zone may cover several physical zones. Since the number of sectors per track is different for each physical zone, the sizes of logical zones are also different. A disk can be partitioned into any number of logical zones, and the choice of the number depends on two factors: the initial response time and the utilization of disk bandwidth. Obviously, more partitions of a disk can reduce the seeking distance. However, it brings the penalty of longer initial response time, since if the disk head just misses the proper logical zone with the requested video block, the user must wait for the disk head coming back to the logical zone with desired block.

According to the underlying strategy of our scheme, we spend the same reading time for each logical zone. The size of reading data is the unit of multiple tracks. A logical zone may cover several physical zones, so we let the sectors per track for a logical zone be the minimum number of sectors per track in a physical zone that the logical zone covers. This is shown in Figure 4. The solid line illustrates the number of sectors per track of physical zones, the dashed line identifies the number of sectors per track of logical zones. We take 4 logical partitions as an example. In this case, some disk space is wasted. In Section 4, we will show that the wasted space is negligible compared to the throughput improvement.

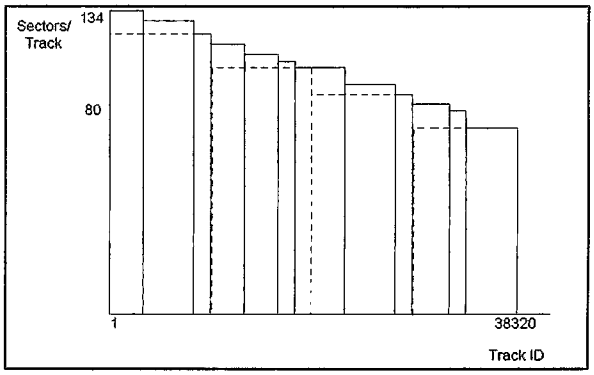

Figure 4 Partition of logical zone

2.4 Alternation of block size

The amount of data of one time access is defined as a block. A block is made up of fixed multiple tracks. To determine the number of tracks in a block, the following factors must be considered. First, the block size must be large enough to meet the consumption by clients between the time period of a service round. Second, the larger the block size than the minimum requirement, the less the total time of disk accesses that is needed. However, more memory is required for a server. The following inequality can be derived to guarantee the property of the continuity. The inequality illustrates the fact that the time of data consumed by any client must be larger than the total time for the server to access the data in a service round. Table 1 presents the meaning of notations.

Table 1 Meaning of notations

Symbol	*Meaning*
N	Number of requests
M	Number of logical zones
B	Block size (KB)
Tm	Minimum disk bandwidth (KB/sec)
Smiz	Maximum seek time within a logical zone (sec)
Smbz	Maximum seek time between two contiguous logical zones (sec)
Stt	Maximum track-to-track seeking or head switching time (sec)
Bt	Consumption rate of a video stream (KB/sec)
Bi	Block size in logical zone i

Nt	Number of tracks in a block
Ti	The minimum bandwidth in logical zone i. (KB/sec)

$$\frac{B}{Tm}\cdot N + Smbz + (N-1)\cdot Smiz + N\cdot(Nt-1)\cdot Stt < \frac{B}{Bt}$$

When accomplishing a service round in a logical zone, the worst movement of the disk head is to move across the logical zone and begin a new service round in the next logical zone. In other words, in the worst case the disk head will move across up to two logical zones when entering a new service round. After entering a new service round, the disk head must find a proper block within a logical zone, and the worst possible moving distance is across the entire logical zone. Besides seeking a block, the track-to-track seeking is also required when the block size is more than one track. In the above inequality, we do not consider the rotational latency since the block size is fixed multiple tracks. The time must be less than the time for consuming a block by any client. Other studies have regarded the block size as a constant[6][10][13][14]. However, this assumption results in the waste of bandwidth for a zoned-disk as described in the previous section. Therefore, we assert that block size should be different for different logical zones in order to obtain the better throughput.

In the above inequality, only one service round is considered. Furthermore, we consider M service rounds which are the sum of the service round from the first logical zone to the last logical zone. In fact, M is the number of logical zones that we have defined. The relation between the number of users and the amount of retrieved data by M service rounds can be derived in the inequality (A). The inequality can be applied to different strategies, for example the strategies described in Figures 1 and 2. Because the time period of the service round is bounded by the lowest bandwidth of a disk, the minimum bandwidth of a disk must be used to calculate the upper bound of the number of users.

$$N\cdot\sum_{i=1}^{M}\frac{B}{Tm} + Smiz\cdot(N-1)\cdot M + M\cdot Smbz + M\cdot N\cdot(Nt-1)\cdot Stt < \frac{M\cdot B}{Bt} \quad .....(A)$$

$$N < \left\lfloor \frac{Tm\cdot[\cdot B - Bt\cdot(Smbz - Smiz)]}{Bt\cdot[B + Tm\cdot Smiz + (Nt-1)\cdot Stt]} \right\rfloor$$

When the facts of different block sizes and different transfer rates in logical zones are applied, the inequality (A) becomes the inequality (B)

$$N\cdot\sum_{i=1}^{M}\frac{Bi}{Ti} + Smiz\cdot(N-1)\cdot M + Smbz\cdot M + M\cdot N\cdot(Nt-1)\cdot Stt < \frac{\sum_{i=1}^{M} Bi}{Bt} \quad .....(B)$$

From inequality (B), the block size must be the size of fixed multiple tracks in logical zone i. Although the number of tracks for all blocks is the same, the size is different due to varying numbers of sectors for tracks. We assume the same amount of time for reading a block in different logical zones. In such a design, the data is read less in inner logical zones and is not enough to maintain video continuity. However, the consumed data in inner logical zones can be compensated for by reading more in outer zones. Thus, we can use the bandwidth to a great extent for each logical zone since the idle time of the disk head has been reduced to the minimum while reading outer zones.

For the detailed relationship between the number of served users and block size, we further explore it from inequality (B). We know the reading time is the same in different logical zones, and we assume that :

$$\frac{B_1}{T_1} = \frac{B_2}{T_2} = ... = \frac{Bm}{Tm} = C \quad ....(C).$$

Hence, C is a constant value. For equation (C), we know $B_i = C\cdot T_i$, and we can derive inequality (D) :

$$N < \left\lfloor \frac{\sum_{i=1}^{M} Bi + Bt\cdot M\cdot(Smiz - Smbz)}{Bt\cdot[C\cdot M + Smiz\cdot M + (Nt-1)\cdot Stt]} \right\rfloor \quad .....(D).$$

3 Supporting interactive playout

In [7], they reconstructed and layout MPEG data on a single hard disk. However, they did not consider multiple hard disks and region-based data placement. In [9], they presented a placement sequence based on the region-based data placement to support FF/FB searching of a video. The method needs no additional buffer but supports limited number of FF/FB playing speeds. In [13], they can provide fully interactive playout of video, but additional buffer is needed while the video is played in FF/FB manner. The above studies adopt the segmentary playing of a MPEG video, and the access block is in the unit of group of picture (GOP)[10].

However, using GOP as a unit for an access block is not applicable for our retrieval method. In our study, we take the advantage of different bandwidth of the disk zoning, and we read more data in outer logical zones to compensate for the insufficient data read in inner logical zones. We let the size of one time access be multiple tracks in each logical zone, and the size is different in every logical zone. The access block of the compensatory retrieval method may not be the multiple of GOPs, and if we assume that the block size should be the multiple of GOPs, the loss of the disk space is serious.

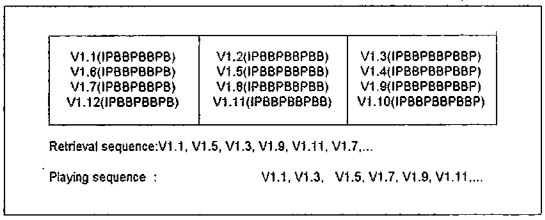

Figure 5 An example of FF searching by using compensatory retrieval method and frame converting technique

In order to provide FF/FB functions and reduce the loss of the disk space in our retrieval method, we present a frame converting method. When the video is stored in the storage device the first time, the access block in each logical zone contains unbroken frames. In other words, each frame would not be stored in two logical zones. For example, we store an access block with I(1), P(4), B(2), B(3), P(7), B(5) frames of video 1 in logical zone 1, and an access block with B(6), P(10), B(8), B(9), P(13), B(11), B(12), I(16) frames of video 1 in logical zone 2. The number before the I/P/B is denoted as the frame ID of a video. The number of frames stored in each logical zone is different since the size of an access block is not the same. However, if we store frames as above, the inter-frame dependence still exists. We can not skip the access block in logical zone 1, and play the access block in logical zone 2 directly. So, we propose a frame converting method to reconstruct frames in every access block. If we find that the starting frame in an access block is not I frame, we convert the frame to I frame and other frames in this block can be encoded by the first I frame. For the same example, we store an access block with I(1), P(4), B(2), B(3), P(7), B(5) frames of video 1 in logical zone 1, and convert and store an access block with I(6), P(10), B(8), B(9), P(13), B(11), B(12), P(16) frames of video 1 in logical zone 2. The P(10) is encoded by I(6), and B(8) is encoded by P(10) and I(6). Although the converting process is time-consuming, it is performed just once at the first time of storing video data. Moreover, the process can be done off-line without influencing the service time. After the frame converting method is applied, frames in an access block are independent from other blocks in other logical zones. Then, the compensatory retrieval method of region-based data placement can also support FF/FB searching of a video. Figure 5 shows an example of 2 times FF by using [13] method, the compensatory method and the frame converting technique. We take one disk as an example, assume there are three logical zones and that SCAN disk scheduling is applied. The block of the innermost logical zone contains 9 frames, the middle logical zone contains 10 frames, and the outermost logical zone contains 11 frames. If the video is requested to play at 2 times FF at the starting time, we retrieve and buffer the 1st, the 5th, and the 3rd blocks and start to play the 1st block while continuing to retrieve the 9th block. Since the frame converting technique is applied, the FF/FB searching based on the compensatory method is feasible.

Table 2 Terminology used to calculate the increment of the disk space

Symbol	*Meaning*
F	Total frames in a video
G	Number of frames in a GOP (We expect that the frame pattern is IPBBPBBPBB....IBBPBBP....)
Fi	Number of frames stored in logical zone i
Ii	Number of I frame stored in logical zone i
Pi	Number of P frame stored in logical zone i
Bfi	Number of B frame stored in logical zone i
I	Number of I frames before frame converting method is applied
P	Number of P frames before frame converting method is applied
Bf	Number of B frames before frame converting method is applied
I'	Number of I frames after frame converting method is applied
P'	Number of P frames after frame converting method is applied
Bf'	Number of B frames after frame converting method is applied
Is	Average size of I frame
Ps	Average size of P frame
Bs	Average size of B frame

The frame converting method brings additional cost of increment of the disk space to store a video. In order to get the increment of disk space, we derive the disk space to store a video before the frame converting method is applied and the space after the method is applied. Table 2 illustrates the terminology that we used.

Before the frame converting method is applied, the total space can be derived by

$I \cdot Is + P \cdot Ps + Bf \cdot Bs$ .....(H1)

where the I, P, B can be obtained from the following equations.

$$I = \left\lfloor \frac{N-G+1}{G} \right\rfloor + 2 \qquad P = \left\lceil \frac{N-1}{3} \right\rceil - \left\lfloor \frac{N-1}{G-2} \right\rfloor$$

$$Bf = N - I - P$$

After the frame converting method is applied, the space to store the converting frames is :

$$I'\cdot Is + P'\cdot Ps + Bf'\cdot Bs \quad .....(H2)$$

where the I', P', B' becomes

$$Ii = \left\lfloor \frac{Fi - G + 1}{G} \right\rfloor + 2 \qquad Pi = \left\lceil \frac{Fi - 1}{3} \right\rceil - \left\lfloor \frac{Fi - 1}{G - 2} \right\rfloor$$

$$Bfi = Fi - Ii - Pi$$

$$I' = \left(\frac{N}{\sum_{i=1}^{M} Fi} \right) \cdot \sum_{i=1}^{M} Ii \qquad P' = \left(\frac{N}{\sum_{i=1}^{M} Fi} \right) \cdot \sum_{i=1}^{M} Pi$$

$$Bf' = \left(\frac{N}{\sum_{i=1}^{M} Fi} \right) \cdot \sum_{i=1}^{M} Bfi$$

Since the size of an access block is different in each logical zone, the number of frames is not the same. Firstly, we get the frames which have been converted in each logical zone. Then, we can get the total frames of a video. So the increment of space is the difference between Equation (H1) and (H2). From the above equations, three factors affecting the increment of the video space can be found. They are the number of logical zones(M), the number of frames in an access block (Fi), and the number of frames in a GOP(G). In the next section, the simulation results will show the impact of these factors. In general, the increment of the video space is acceptable.

In addition, the different size of one time access in different logical zones brings another influence on the FF/FB function. The influence is that the length of played video and the length of skipped video are different since the block sizes are different for every logical zone. In order to realize how much the influence is, we perform the following tests. We reconstruct the MPEG stream to skip some segments, and we can get the influence. We use the SCAN-like data placement and SCAN disk scheduling as an example. If we select 4 logical zones and one track is assumed as an access block, the 2 times FF searching shows 0.2 sec, skips 0.25 sec, shows 0.28 sec, skips 0.32 sec, shows 0.32 sec, skips 0.28 sec, shows 0.25 sec, skips 0.2 sec, and shows 0.2 sec again. The effect is not easy to be recognized by human eyes, even when the change of the video is significant. We increase the size of the access block to accommodate more frames. The change of the skipped segment and played segment is still acceptable when the size of the block is 5 tracks. However, the influence becomes sensitive when the block size is more than 5 tracks and the scene change of the video is large. In general, less than 10 tracks per access block is acceptable.

4. Simulation results and system prototyping

4.1 Simulation environment

In this Section, we simulate the improvement of the disk throughput by using the compensatory retrieval method, the increment of video space by using the frame converting method, and the impression of FF/FB functions. Before describing and discussing the simulation results, we describe the simulation environment.

As introduced in Section 2, our retrieval method can be applied in all kinds of region-based placement sequences. We take the SCAN-like data placement as an example to simulate the improvement of the disk throughput. Moreover, the SCAN disk scheduling is used. MPEG-I quality video is assumed as the bit rate of 1.5Mbps on average. Since the improvement of every disk in the disk array is the same, we just show the results of a single hard disk. Quantum XP32150W hard disk is used as the testbed for the simulation. Table 3 describes the hardware parameters of XP32150W and the zone information for the disk is listed in Table 4. In order to make the simulation results more practical, we use parameters from experiments on the real device. The track-to-track seek time and head switching time are average values of our experiments. The way we adopt can get a more accurate estimation of the seek time.

Table 3 Parameter of XP32150W

Track-to-Track seek time or head switching time(ms)	1
Seek time modeling	Mapping function

Table 4. Zone information of XP32150W

Zone	Sectors/track	Track/zone	Raw transfer rate(KB/s)
0	134	208	8040
1	131	262	7860
2	128	248	7680
3	125	304	7500
4	123	176	7380
5	120	232	7200
6	116	248	6960
7	113	232	6780
8	107	528	6420
9	102	216	6120
10	98	240	5880
11	93	248	5580
12	89	248	5340
13	85	184	5100
14	80	256	4800

4.2 Utilization of the disk bandwidth

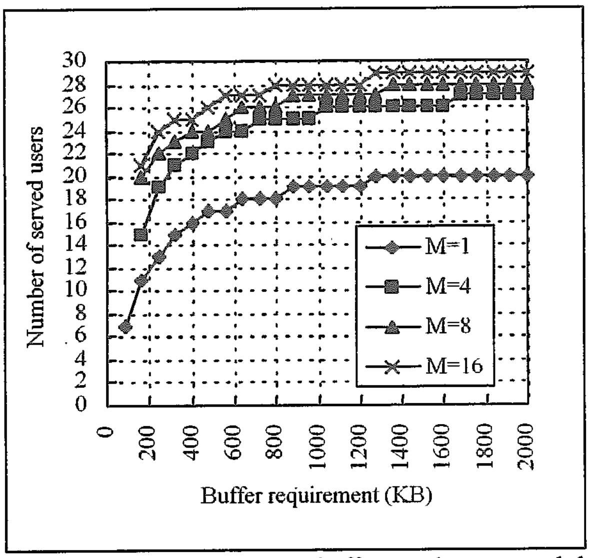

Figure 6 The relation between buffer requirement and the number of supported users by using compensatory region-based data placement

Figure 6 shows the relation of buffer requirement and the number of served users by proposed method and origin region-based data placement. The simulation results show that the improvement of disk throughput by using our compensatory method is worthwhile. The M value is denoted as the number of logical zones on a disk. From Figure 6, it can be seen that the maximum number of users is more than the number in the region-based data placement method under the same M value. This illustrates that our proposed method can efficiently utilize the different bandwidth of physical zones. Therefore, the maximum number of served users is never limited by the lowest bandwidth of a hard disk. Moreover, we can find that the improvement of compensatory method is not much when M>4, since the average bandwidth of M=4 is near to M>8. In other words, we can use the higher bandwidth of outer zones but meanwhile suffer the lower bandwidth of inner zones, when increasing the partition number. For example, if M is set to 4, the bandwidth of logical zones is 4.8MB/s, 5.88MB/s, 6.42MB/s, and7.5MB/s respectively, and the average bandwidth of this hard disk is 6.15MB/s. The bandwidth is improved by 28.1% more than the minimum bandwidth of a hard disk. However, if the M is set to 16, the bandwidth of logical zones is 4.8MB/s, 4.8MB/s, 5.34MB/s, 5.58MB/s, 5.88MB/s, 6.12MB/s, 6.42MB/s, 6.42MB/s, 6.78MB/s, 6.96MB/s, 6.96MB/s, 7.38MB/s, 7.5MB/s, 7.68MB/s, 7.68MB/s, and 7.86MB/s, and the average bandwidth is 6.5MB/s. The improvement is just 6% compared to M=4. Hence, the disk throughput is not improved very much when M>4.

4.4 Support interactive playout

To support FF/FB searching of a video on our retrieval method, the frame converting technique must be applied. However, the frame converting method will increase the space of a video. To investigate the increment of video space, we chose an MPEG-I stream to do some experiments. The famous movie "Forrest Gump", produced by Paramount Pictures, was chosen as the tested case. We convert the original Video-CD to MPEG-I stream, and parse the frames in the video. There are 45 frames in a GOP. Meanwhile, we get the average size of I, P, and B frames and these values are used for simulation of the increment of video space. Table 5 shows the frame size.

Figure 7 shows the relation of the block size and the percentage of lost space when the number of logical zones is 4. From this figure, increasing the block size means that the number of frames stored in a block is increased, and the percentage of lost space is reduced. In other words, if we increase the block size, the lost space can be reduced. Comparing 15 frames in a GOP (G=15) and 45 frames in a GOP (G=45), we can see that the number of frames in a GOP does not influence the increment of video space when the block contains more than 5 tracks. Although increasing the block size can reduce the lost space due to frame converting, it brings another problem: the time of played segment and the time to skip becomes more variant. We reconstruct the MPEG stream and measure the effect of FF and FB searching. We find that the FF/FB quality is acceptable when the block contains less than 5 tracks, no matter the scene change of the video is large or small. The alternation of the playing and the skipping is: play 1.05 sec, skip 1.28 sec, play 1.4 sec, skip 1.6 sec, play 1.6 sec, skip 1.4 sec, play 1.28 sec, skip 1.05 sec, and play 1.05 sec again, when the number of logical zones is 4. Furthermore, if the scene change of a video is small, the FF/FB quality is acceptable when there are 10 tracks in a block. However, the lost scene is significant when the scene change of a video is more.

Table 5 Average frame size of MPEG-I video

	I Frame	P Frame	B Frame	GOP	Frames in a GOP
Bytes	14430	7509	3134	213578	45 Frames

5. Conclusions

The study presents a new data placement and the corresponding retrieval method. The advantage of our approach is not only in limiting the seeking distance but also in utilizing the disk bandwidth efficiently. By using the frame converting technique based on compensatory retrieval method based-on region-based data placement scheme, the FF/FB searching of the video has been also provided. As for the disk space lost due to the concept of the logical zone being applied and the frame converting method being used, the loss is negligible compared to the improvement of the disk throughput. By the simulation results and further performance testing of our prototype, the improvement of the disk throughput by applying compensatory retrieval method is seen to be 25%-35%, compared to the original region-based data placement method. The total lost space due to the concept of the logical zone and the frame converting technique is about 5% of the original space.

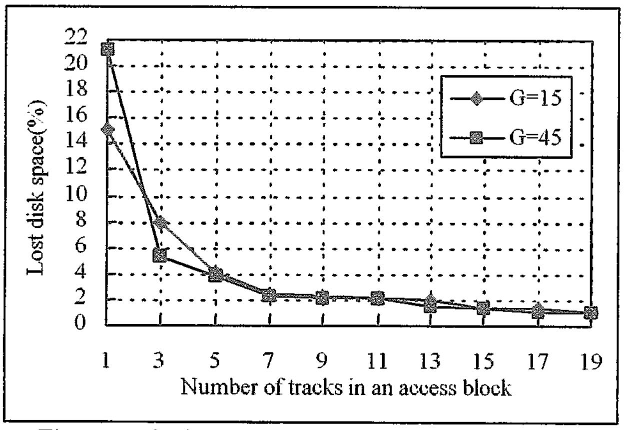

Figure 7 The increased space of a video when M=4

Reference

[1]Narasimha Reddy, James C.Wyllie, "I/O Issues in a Multimedia System", IEEE Computer, Vol. 27, No. 3, pp67-74, Mar. 1994

[2]Banu Ozden, Rajev Rastogi, Avi Silberschatz., "A Framework for the Storage and Retrieval of Continuous Media Data", Proc. of IEEE Conf. on Multimedia Computing Systems, pp2-13, May 1995

[3]Chris Ruemmler, John Wilkes, "An Introduction to Disk Drive Modeling", IEEE Computer, Vol.27, No.3, pp17-27, Mar. 1994

[4]Deepak R. Kenchammana-Hosekote, Jaideep Srivastava, "I/O Scheduling for Digital Continuous Media", University of Minnesota Technical Report, 1994

[5]D.James Gemmell, et al., "Multimedia Storage Servers : a Tutorial", IEEE Computer, Vol.28, No. 5, pp40-49, May 1995

[6]Horng-Juing Lee, et al. ,"Cluster Placement: A Data Placement Scheme for a Mass Storage System of a Video-on-Demand Server", University of Minnesota Technical Report,1995

[7]Huang-Jen Chen, "A Disk Scheduling Scheme and MPEG Data Layout Policy for Interactive Video Access From a Single Disk Storage Device", Boston University, Ph.D Thesis,1995

[8]Jenwei Hsieh, et al., "Performance Studies of a Mass Storage System for Video-on-Demand", University of Minnesota Technical Report,1994

[9]Meng-Huang Lee, et al., "Designing A Fully Interactive Video-on-Demand Server with A Novel Data Placement and Retrieval Scheme ", IEEE Trans. on Consumer Electronics , Vol. 41, No.3, pp851-858, Aug. 1995

[10]Ming-Syan Chen, Dilip D. Kandlur, Philip S. Yu., "Storage and Retrieval Methods to Support Fully Interactive Playout in a Disk-array-based Video Server", Multimedia System, No.3, pp126-135, 1995

[11]Yitzhak Birk, "Track-Pairing: a Novel Data Layout for VOD Servers with Multi-Zone-Recording Disks", Proc. of 1995 Multimedia Computing Systems, pp248-255, 1995

[12]Yen-Jen Oyang, Chun-Hung Wen, Chin-Yuan Cheng, Meng-Huang Lee and Jian-Tian Li, "A Multimedia Storage System for On-Demand Playback", IEEE Trans. on Consumer Electronics, Vol.41, No.1, Feb 1995

[13]Chih-Yuan Cheng, et al., "Effective Utilization of Disk Bandwidth for Supporting Interactive Video-on-Demand", IEEE Transactions on Consumer Electronics, Vol.42, No.1, Feb. 1996.