1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

An Effective Data Placement Scheme for Supporting VCR Functionality in an Video Server

Shin-Tse Cheng, Yu-zhan Hou, Cheng Chen, and Suh-Yin Lee

Department of Computer Science and Information Engineering,
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
Email:{sjcheng, yzhou, cchen, sylee}@csie.nctu.edu.tw

ABSTRACT

Recently, the video server plays an important role in VOD or near VOD system. In general, a good video server can not only provide start/stop operations for playback of each request, but also support VCR functionality. In this thesis, we will propose an effective data placement method, called Enhanced-Segment-Placement (ESP) scheme, to support VCR operations built in our video server management platform. Its placement pattern provides uniform load balance for arbitrary retrieval rate as well as normal rate playback. Some distinguished features are explored by appropriate analytic approach. Meanwhile, experiments show that our scheme will have higher throughput, lower reject probability as well as better load balance factors compared with other methods. The detailed description of ESP scheme and its implementation and evaluations will be given in the literature.

1. Introduction

Recent progress in digital video technology has led to large-scale multimedia systems. This kind of systems store videos on disks and provide video-on-demand (VOD) services to many clients over wired or wireless network [1][2][3]. In VOD services, clients are allowed to apply VCR-like operations, such as pause, resume, fast-forward (FF) and fast-rewind (FW) [4][5]. Usually, the performance of a video server is critically dependent on hot spots (i.e. overloaded disks) in the disk array [4]. This becomes more complicated with VCR-like playback request. The obvious remedy is to remove hot spots using load-balancing mechanism on the disks, which is usually achieved by efficient disk placement and retrieval schemes [4][5].

So far, several placement methods for supporting VCR-like operations have been proposed [4][5]. The first kind of methods is that a special file is created specifically for use in fast-forward/rewind situations [16]. The drawback of this method is both of the increase of the amount of disk space used by videos and permanent fast-forward/rewind playback rate. The second method is to reorder the video file and simply send I/P frames—called skipping frames [21]. It is tough to implement in our video server. Considering normal play, the video server has to retrieve both of versions of a GOP, and then reconstructs entire look of the GOP. The third kind of method is to skip entire groups of frames – GOP (Group of Picture) [7][8]. Basically, it can be further divided into three different schemes [4][5]. (i) Segment-Sample-Method: GOPs are stored in an round-robin manner. While fast-forward operation is performed, workload will be shifted across disks by calculating which GOP should be retrieved. But the fast-forward rate is not stable [4]. (ii) Segment-Placement-Method: This method allocates GOPs to disks judiciously, so that the GOP can be uniformly sampled in fast-forward at some pre-determined speed [4]. (iii) Prime-Round-Robin Method: At first, we find the largest prime number N_p on the value of N disks, and then we divide N disks into two groups of disks. One contains $N_p - 1$ disks, while the other contains $N - N_p + 1$ disks. Finally, we store GOPs according to if the GOP number is divisible by N_p [5].

In fact, the size of a GOP varies in real world [7][8]. For the ease of implementation, the size of retrieval block in our video server is fixed and smaller than a GOP [7][8]. Our video server has to retrieve several blocks in consecutive service rounds, so as to approximate a GOP for playback. By this way, we will develop an effective data placement method – called ESP (Enhanced-Segment-Placement) scheme – for supporting VCR operations in our video server, which is produced by Mentor Data System Inc. [9]. By this way, ESP scheme places blocks in disks by an improved segment-placement method and round-robin placement pattern. It can effectively achieve well-designed load balance and larger throughput during either normal-rate or fast-forward playback as well as the video quality is acceptable. After we simulated and implemented this scheme on our video server, we find that this scheme provide lower reject probability about twenty percent less than other methods. This scheme also makes the load imbalance factor about thirty percent less than other methods.

The remaining part of this paper is organized as follows. In Section 2, we will describe the architecture of our video server at first and then introduce the issues, concepts and principles of our ESP scheme in some detail. In Section 3, several properties of our scheme will be analyzed. In Section 4, simulation results and performance evaluations are shown. Finally, the concluding remarks are given in Section 5.

2. Principle of Enhanced-Segment-Placement Method

2.1 Overview of Our Video Server Architecture

Our video server is produced by Mentor Data System Inc. [9]. It can be separated into two portions, as shown in Fig. 1. One is the video server management subsystem and the other is the video server engine. The video server management subsystem handles matters of system resource management, stream management, admission control management, data placement management, and a video database [2]. The video server engine handles matters of disk scheduling management, stored data retrieval, and buffer management [22]. The size of basic retrieval block is fixed on 94 Kbytes in video server engine.

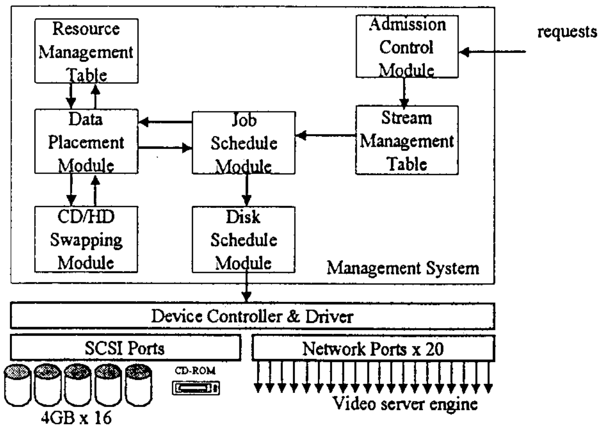


Figure 1. Our Video Server Architecture

Based on the video server engine, we have developed an useful and effective video server management system for VOD applications [1]. Fig. 1 illustrates that our video server management system is divided into several modules. When clients' requests are coming in the video server, admission control module guards them from entering the remaining module to be served and keeps quality of displaying video streams good. The stream management table keeps accepted video streams and makes sure each one of them available to job schedule module immediately. Job schedule module maintains continuous playback of video streams in a cycle fashion. Data placement module is responsible for a content-updating stream in job schedule module. Disk server module receives retrieval commands from job schedule module, and transforms them to SCSI- commands. For supporting VCR-like operations, we have to enhance some functions in data placement module. Video content will be placed in and retrieved from resource management table properly in the way of our placement scheme.

2.2 Basic Design Issues

Basically, there are two main design issues to be deliberated here. One is Quality of Service (QOS) and the other is storage utilization [1][13]. For each client requesting video data, good quality of service implies a jitter-free video display. There are two considerations to achieve good quality of service, while VCR-like operations are performed. One is the jitter-free and the other is load- balancing [23]. Jitters would happen because the retrieved block might consist of a series of frames in a GOP, which cannot be decoded independently. In order to overcome this problem, we establish our placement scheme based on multiple blocks as a fundamental unit. The consequence still results to slight jitters, but displayed scenes are acceptable. Storage utilization means that how many percentage of storage space in a disk array is used and how often it is accessed. Some placement schemes introduced well-designed methods but ignored the issue of storage utilization, such as Prime-Round-Robin (PRR) scheme [5]. On the other hand, we prefer an small FAT (File Allocation Table) to store the location of videos. We compute the location of the retrieval block run-time by our placement scheme, in order to reduce the size of FAT and upraise storage utilization.

Before we describe our ESP scheme, we will give some assumptions and notations at first. We assume that our placement pattern maps to a disk array. We use several blocks as a fundamental unit. Table 1 summarizes the symbols we will use.

Symbol	Description
N_{disk}	the number of disks in a disk server
D_i	i^{th} disk
R_{ff}	the fast-forward rate
n_r	the ratio of a fundamental unit to a fundamental block
H^j_i	the logical j^{th} fundamental block to be accessed in a disk
B^j_i	the j^{th} fundamental block of the i^{th} video, V_i
S^j_i	the j^{th} fundamental unit of the i^{th} video, V_i
$T^{R_{ff}}_{ij}$	the transaction to retrieve S^j_i with fast-forward rate R_{ff}
t_i	i^{th} time unit to display fundamental units

Table 1. Notations

2.3 Concept and Principle

For the ease of implementation in real video server, the video server usually adopts small fixed-size blocks for retrieval and storage. We use multiple-block-one-unit policy in order to enclose at least one GOP in a fundamental unit for playback. Basically, there are three key factors to be considered. The first is the number of disks N_{disk}, the second is fast-forward rate R_{ff} and the third is the unit/block ratio n_r [4][5].

In multi-block-one-unit policy, other placement methods either afford less fast-forward rate or bias the workload [4][5][25]. In order to achieve the load balance, we propose a new placement pattern – called rotational striping, as shown in Fig. 2. Consider the condition of $n_r = 4$, $R_{ff} = 4$, and $N_{disk} = 8$, we stripe rotationally the fundamental blocks in a fundamental unit on adjacent disks, instead of placing them in the same disk. Rotational striping balances in a shorter time than other methods.

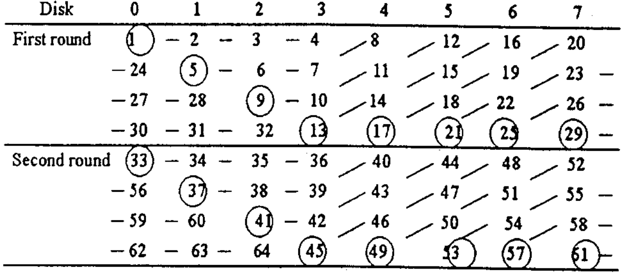
Disk	0	1	2	3	4	5	6	7
	1*	2*	3*	4*				
		5	6	7	8			
			9	10	11	12		
				13	14	15	16	
					17*	18*	19*	20*
	24					21	22	23
	27	28					25	26
	30	31	32					29
Load	1	1	1	1	1	1	1	1

Figure 2. Rotational Striping with unit/block ratio = 4

We can derive a general rule for different cases to use appropriate placement patterns to balance the workload. This implies two alternative schemes – either ESP scheme without shift or ESP scheme with shift, which will be described in the following.

2.4 Scheme without Shift

As shown in Fig. 2, only half of storage space is allocated for video data. Hence, we will not only allocate fundamental blocks to fill the spare space, but also ensure that the merit of rotational striping is maintained. For illustrative purpose, Fig. 3 shows the fundamental blocks retrieved in the first two rounds of retrieval by ESP-NS (Enhanced-Segment-Placement scheme with Non-Shift). In Fig. 3, $N_{disk} = 8$, $R_{ff} = 4$, and $n_r = 4$. Our ESP-NS algorithm moves up some blocks to fill the spare space and utilizes storage well. By this way, the arrangement of fundamental blocks will be tight over all the disks. And in each round, N_{disk} disks contain $n_r \times N_{disk}$ blocks.



Disk	0	1	2	3	4	5	6	7
First round	①	2	3	4	8	12	16	20
	24	⑤	6	7	11	15	19	23
	27	28	⑨	10	14	18	22	26
	30	31	32	⑬	⑰	㉑	㉕	㉙
Second round	㉝	34	35	36	40	44	48	52
	56	㊲	38	39	43	47	51	55
	59	60	㊶	42	46	50	54	58
	62	63	64	㊺	㊾	(53)	(57)	(61)

Figure 3. ESP without Shift

By this way, we can give the following algorithm formally.

Algorithm ESP-NS (V_i, N_{disk}, R_{ff}, n_r).

In the r^{th} round of retrieval, the k^{th} disk, $0 \leqq k \leqq N_{disk} - 1$ and $N_{disk} \geqq n_r$, retrieves the fundamental block with block number j in video i, where $r = \lfloor B^j_i / (N_{disk} \times n_r) \rfloor \times n_r$ and $z = \lfloor B^j_i / n_r \rfloor + [B^j_i]_{nr} > 0$.

```
Step 1:   $J = 1$;
    Repeat step 2 to 5 until ($V_i$ is completely stored or retrieved from disks)
Step 2:   set current accessed block $j$ of video $i$ to $B^j_i$
Step 3:   set current accessed disk to $D_k$, where
          $D_k = [\, [\lfloor B^j_i / n_r \rfloor]_{Ndisk} + [B^j_i]_{nr} - 1 \,]_{Ndisk}$
Step 4:   set current accessed block $H^j_i$ in $D_k$, where
          If ($z \geqq N_{disk}$) then
              $H^j_i = \lfloor B^j_i / n_r \rfloor - (N_{disk} - n_r) + r$
          Else if ($n_r < z < N_{disk}$) then
              $H^j_i = \lfloor B^j_i / n_r \rfloor - [z]_{nr} + r$
          Else then
              $H^j_i = \lfloor B^j_i / n_r \rfloor + r$
Step 5:   $J = J + 1$
```

where $[\cdot]_n$ means value in the bracket mod n.

Consider Fig. 3, if current fundamental block B^j_i is equal to 1, we find this block belongs to the first fundamental unit and it is also the first block in that unit. According to algorithm ESP-NS, the current accessed disk is 0. The z value is smaller than n_r, so that the current accessed block locates in the first row indicated by 0. By the same way, the access sequence will be $\{D_k, H^j_i\} = (\{0, 0\}, \{1, 0\}, \{2, 0\}, \{3, 0\}, \{4, 3\}, \{5, 2\}, \{6, 1\}, \{7, 0\}, \{0, 4\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{4, 7\}, \{5, 6\}, \{6, 5\}, \{7, 4\})$ in fast-forward mode. It can be seen that workload is distributed and balanced.

Obviously, the complexity of ESP scheme without shift is $O(n^2)$, which is the same as other placement methods [4][5].

2.5 Scheme with Shift

While in the case of $(R_{ff}, N_{disk}) \neq 1$ and n_r does not divide R_{ff}, most of disk accesses hit on some of disks in a disk array more frequently. As shown in Fig. 4 for example, assume that $N_{disk} = 8$ and $R_{ff} = 4$ with $n_r = 6$. We have to shift entire placement pattern in order to make the workload balanced. To achieve this objective, we will devise the ESP with shift procedure, namely, algorithm ESP-S (Enhanced-Segment-Placement scheme with Shift). In the following, we give the algorithm formally.

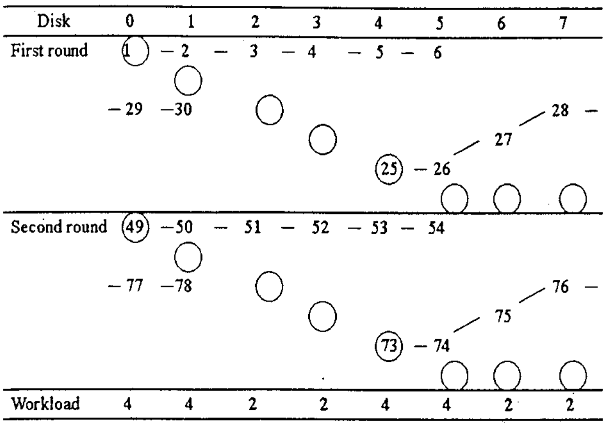


Figure 4. A fast retrieval without shift for $N_{disk} = 8$, $R_{ff} = 4$, and $n_r = 6$

Algorithm ESP-S (V_i, N_{disk}, R_{ff}, n_r).

In the r^{th} round of retrieval, the k^{th} disk, $0 \leqq k \leqq N_{disk} - 1$ and $N_{disk} \geqq n_r$, retrieves the fundamental block with block number j in video i, where $r = \lfloor B^j_i / (N_{disk} \times n_r) \rfloor \times n_r$, and $z = \lfloor B^j_i / n_r \rfloor + [B^j_i]_{nr} > 0$. Assume that ND_{shift} represents the number of disks skipped when shifting; NR_{shift} represents the number of rounds which should perform the shifting; n_{round} represents the round number. Let g.c.d(n, m) denote the greatest common divisor of n and m, which are two positive integers.

```
Step 1:  $j = 1$;
    Repeat step 2 to 8 until ($V_i$ is completely stored or retrieved from disks )
Step 2:     set current accessed block $j$ of video $i$ to $B^j_i$
Step 3:     Set the number of disks skipped to $ND_{shift}$ when shifting,
            If ( $n_r > R_{ff}$ ) then
                $ND_{shift} = [n_r - R_{ff}]_{g.c.d(Ndisk, Rff)}$
            Else then
                $ND_{shift} = [n_r]_{g.c.d(Ndisk, Rff)}$
Step 4:     Set the number of shifts to $NR_{shift}$, where
            If ( $[g.c.d(N_{disk}, R_{ff})]_{NDshift} > 0$ ) then
                $NR_{shift} = g.c.d(N_{disk}, R_{ff})$
            Else then
                $NR_{shift} = g.c.d(N_{disk}, R_{ff}) / ND_{shift}$
Step 5:     Set the round number $n_{round}$, where
            $n_{round} = [\lfloor B^j_i / (N_{disk} \times n_r) \rfloor]_{NRshift}$
Step 6:     set current accessed disk to $D_k$, where
            $D_k = [\, [\lfloor B^j_i / n_r \rfloor + ND_{shift} \times n_{round}]_{Ndisk} + [B^j_i]_{nr} - 1\, ]_{Ndisk}$
Step 7:     set current accessed block $H^j_i$ in $D_k$, where
            If ( $z \geqq N_{disk}$ ) then
                $H^j_i = \lfloor B^j_i / n_r \rfloor - (N_{disk} - n_r) + r$
            Else if ( $n_r < z < N_{disk}$ ) then
                $H^j_i = \lfloor B^j_i / n_r \rfloor - [z]_{nr} + r$
            Else then
                $H^j_i = \lfloor B^j_i / n_r \rfloor + r$
Step 8:     $j = j + 1$
```

where $[\cdot]_n$ means value in the bracket mod n.

For illustrative purpose, Fig. 5 shows the fundamental blocks retrieved in the first two rounds of retrieval by algorithm ESP-S for the case of $N_{disk} = 8$, $R_{ff} = 4$, and $n_r = 6$. As indicated in Fig. 5, we got { ND_{shift}, $NR_{shift} - 1$} = {2, 1}. That means we need to shift the placement pattern by 2 in the even round, as shown in Fig.5. The current accessed disk will be adjusted by the factor of 2 while the current accessed block is not adjusted in the even round. Step 6 and step 7 is just the same as step 3 and step 4 in algorithm ESP-NS. We can find the access sequence of fundamental blocks is $\{D_k, H^j_i\}$ = ({0, 0}, {1, 0}, {2, 0}, {3, 0}, {4, 0}, {5, 0}, {4, 4}, {5, 4}, {6, 3}, {7, 2}, {0, 2}, {1, 2}, {2, 6}, {3, 6}, {4, 6}, {5, 6}, {6, 6}, {7, 6}, {6, 10}, {7, 10}, {0, 9}, {1, 8}, {2, 8}, {3, 8}) in fast-forward mode. Even though the video server retrieves the fundamental block B^j_i as the same order as before, the workload distributes to every disks equally due to static placement scheme of ESP with shift, when fast-forward operation is performed.

Because all fundamental blocks in a fundamental unit stripe, the blocks retrieved in each round scattered approximately evenly across disks. The advantage is the same as described in ESP without shift. It is claimed that, for each disk, ESP scheme reserves more bandwidth for outputting video data, and the number of clients, whose requests are accepted, would increase. The complexity of ESP scheme with shift is also $O(n^2)$.

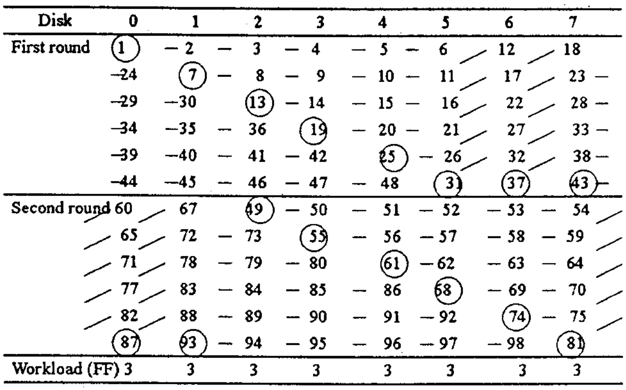


Figure 5. ESP with Shift

Now we can summarize the ESP algorithm in the following by combining ESP-NS and ESP-S.

Algorithm ESP.

The number of disks in the disk array is N_{disk}. The fast-forward rate in the system is R_{ff}. The unit/block ratio is set to n_r.

```
If ( $(R_{ff}, N_{disk}) = 1$ ) then
    For each requested video $i$, $V_i$
        ESP-NS( $V_i$, $N_{disk}$, $R_{ff}$, $n_r$ )
Else if ( $(R_{ff}, N_{disk}) \neq 1$ ) then
    If ( Either $N_{disk}$ divides $R_{ff}$ or $N_{disk}$ does NOT divide $R_{ff}$ ) then
        If ( $n_r$ divides $R_{ff}$ ) then
            For each requested video $i$, $V_i$
                ESP-NS( $V_i$, $N_{disk}$, $R_{ff}$, $n_r$ )
        Else if ( $n_r$ does NOT divide $R_{ff}$ ) then
            For each requested video $i$, $V_i$
                ESP-S( $V_i$, $N_{disk}$, $R_{ff}$, $n_r$ )
```

3. Properties of ESP Scheme

In this section, we will describe several properties of our ESP scheme. We analyze each condition about load balance, throughput, placement strategy and quality of service.

3.1 Analysis of Load Balance

In this subsection, we will show that our ESP scheme would balance the workload over disks in the disk array. We conclude no matter there is prime number relationship or not between the number of disks and the fast-forward rate, load will be distributed evenly.

Lemma 1. In ESP scheme placement, all fast retrieval as well as normal play with speed $R_{ff} \neq kN_{disk}$ ($k = 1, 2, \ldots$) require accesses to N_{disk} distinct disks for N_{disk} time units regardless of the value of R_{ff}, if and only if N_{disk} is prime number [5].

Proof. Let $T^{Rff}{}_{ij}$ be a transaction as described in Table 1. $S^{j}{}_{i}$, $S^{j+Rff}{}_{i}$, $S^{j+2Rff}{}_{i}$, ... will be presented at t_1, t_2, t_3, ... and are stored on $D_{(i+j) \bmod Ndisk}$, $D_{(i+j+Rff) \bmod Ndisk}$, $D_{(i+j+2Rff) \bmod Ndisk}\cdots$ respectively. If $S^{j+dRff}{}_{i}$ ($d = 0, 1, \ldots$) is also stored on disk $D_{(i+j) \bmod Ndisk}$ storing $S^{j}{}_{i}$, then $dR_{ff} = kN_{disk}$ ($k = 1, 2, \ldots$). Assuming that N_{disk} is a prime number, the minimum value of d is equal to N_{disk}, because N_{disk} is not divisible by R_{ff} (= 1, 2, ..., $N_{disk}-1$). That is, $T^{Rff}{}_{ij}$ needs to access the same disk only in every $N_{disk}{}^{th}$ time unit for $R_{ff} \neq kN_{disk}$.

Corollary 1. In ESP scheme placement, all fast retrieval as well as normal play with speed $R_{ff} \neq kN_{disk}$ ($k = 1, 2, \ldots$) require accesses to N_{disk} distinct disks for N_{disk} time units, if and only if N_{disk} and R_{ff} is relatively prime number.

Proof. From Lemma 1, we deduce this corollary directly.

Theorem 1. In a ESP scheme placement, all fast retrieval as well as normal play with speed $R_{ff} \neq kN_{disk}$ ($k = 1, 2, \ldots$) require accesses to N_{disk} distinct disks and the workload is evenly distributed across N_{disk} distinct disks for $d \times l.c.m(N_{disk}, R_{ff}) / R_{ff}$ ($d = 1, 2, \ldots$) time units, if and only if N_{disk} and R_{ff} is not relatively prime and n_r divides R_{ff}. Here, l.c.m(n, m) denote the least common multiple of n and m, which are two positive integers.

Proof. Without loss of generality, only the very first fundamental block of a fundamental unit is considered. In a service round, $l.c.m(N_{disk}, R_{ff}) / R_{ff}$ out of $l.c.m(N_{disk}, R_{ff})$ such blocks are accessed. Thus, there are $n_r \times l.c.m(N_{disk}, R_{ff}) / R_{ff}$ out of $n_r \times l.c.m(N_{disk}, R_{ff})$ blocks retrieved for playback in fact. Assuming that $n_r = R_{ff}$, $S^{j}{}_{i}$, $S^{j+Rff}{}_{i}$, $S^{j+2Rff}{}_{i}$, ... are presented at t_1, t_2, t_3, ... and the very first fundamental block of each S are stored on $D_{(i+j) \bmod Ndisk}$, $D_{(i+j+Rff) \bmod Ndisk}$, $D_{(i+j+2Rff) \bmod Ndisk}\cdots$ respectively, it means $B^{j}{}_{i}$, $B^{j+1}{}_{i}$, ..., $B^{j+nr-1}{}_{i}$, $B^{j+Rff}{}_{i}$, $B^{j+Rff+1}{}_{i}$, ..., $B^{j+Rff+nr-1}{}_{i}$, ... are retrieved to play out and these blocks are stored on $D_{(i+j) \bmod Ndisk}$, $D_{(i+j+1) \bmod Ndisk}$, ..., $D_{(i+j+nr-1) \bmod Ndisk}$, $D_{(i+j+Rff) \bmod Ndisk}$, $D_{(i+j+Rff+1) \bmod Ndisk}$, ..., $D_{(i+j+Rff+nr-1) \bmod Ndisk}$. By assumption, there are $l.c.m(N_{disk}, R_{ff})$ blocks out of $R_{ff} \times l.c.m(N_{disk}, R_{ff})$ blocks retrieved from N_{disk} distinct disks once respectively for $l.c.m(N_{disk}, R_{ff}) / R_{ff}$ time units. Thus, We can extend this lemma to $n_r = dR_{ff}$ ($d = 1, 2, \ldots$) without effort.

3.2 Analysis of Throughput

In the following theorem, we not only show that load is balanced across disks but also show that algorithm ESP-S can achieve maximal throughput if shift is necessary.

Theorem 2. The number of shifts incurred by algorithm ESP-S is the minimum among all FF retrieval schemes that could achieve the maximal throughput, if each media segment is constructed by multiple blocks which are distributed across the disks.

Proof. In fast-forward mode, l.c.m(N_{disk}, R_{ff}) / R_{ff} implies how many fundamental units are retrieved, before the succeeding fundamental unit accesses the same series of disks, without any shift in placement. Considering two cases, one is the case with $n_r > R_{ff}$; the other is with $0 < n_r < R_{ff}$. In the former case, there are $(n_r - R_{ff}) \times$l.c.m(N_{disk}, R_{ff}) / R_{ff} overloaded disks among N_{disk} disks in a disk array. In other words, average workload spreading over each disk is $(n_r - R_{ff}) \times$l.c.m(N_{disk}, R_{ff}) / $(R_{ff} \times N_{disk})$, and this is equivalent to $(n_r - R_{ff})$ / g.c.d(N_{disk}, R_{ff}) in a round. For that accumulated workload of each disk ought to be $(n_r - R_{ff})$ and distributed evenly, after introducing shift scheme, the load would be balanced in g.c.d(N_{disk}, R_{ff}). Following the same rule, we could prove the other case as mentioned in the beginning.

3.3 Analysis of Placement Strategy

Now we discuss the placement strategy, which is used while we intend to determine the fast-forward rate and the unit/block ratio for our video server. We consider two cases here.

Case 1. Assume that video data have not yet been stored in the disk array. With ESP algorithm, arbitrary values of N_{disk}, R_{ff} and n_r are able to be manipulated.

Case 2. Assume that video data have been stored in the disk array. (i) Assume that the placement pattern is for ESP without shift. According to the ESP-NS algorithm, the fast-forward rate can be either a sub-multiple of n_r or a relatively prime number of N_{disk}. (ii) Assume that the placement pattern is for ESP with shift. According to the ESP-S algorithm, we can get different rates of fast-forward by the following formula, g.c.d(N_{disk}, *new* R_{ff}) = g.c.d(N_{disk}, *original* R_{ff}). g.c.d(n, m) denote the greatest common divisor of n and m, which are two positive integers.

We find that $(R_{ff}, N_{disk}) \neq 1$ and $n_r = R_{ff}$ results to better performance than other cases, while R_{ff} is specified. In addition, the optimal options for fast-forward rate are sub-multiple of the number of disks.

3.4 Analysis of Quality of Service

In order to show that the display effect is acceptable, we initially assume that the size of a GOP is fixed for convenience, for example 112.5 KBytes. Any part of a GOP, which includes I frame, will be considered to be independently decoded. In addition, assume that L_{block} is the size of a block, which is 94 Kbytes in our video server, and L_{GOP} is the size of a GOP in kilobytes. We use the following rules to measure that how many percent of retrieved data is decoded.

(i) A fundamental retrieval unit contains $\lfloor (n_r \times L_{block})/L_{GOP} \rfloor$ complete GOPs. For illustrative purpose, the first three cases in Fig. 6 shows this condition. We can conclude that there are $\lfloor (n_r \times L_{block})/L_{GOP} \rfloor$ complete GOPs while,

$$((k \times R_{ff} \times n_r \times L_{block}) \bmod L_{GOP}) \le ((n_r \times L_{block}) \bmod L_{GOP}). \quad \textbf{(4-1)}$$

As shown in Fig. 6, $((n_r \times L_{block}) \bmod L_{GOP})$ means the difference between the last retrieved block and the third GOP in case 1. The above condition ensures case 1-3 occur.

(ii) A fundamental retrieval unit contains $(\lfloor (n_r \times L_{block})/L_{GOP} \rfloor - 1)$ complete GOPs. The case 4 in Fig. 6 shows this condition. We can conclude that there are $(\lfloor (n_r \times L_{block})/L_{GOP} \rfloor - 1)$ complete GOPs while,

$$((k \times R_{ff} \times n_r \times L_{block}) \bmod L_{GOP}) > ((n_r \times L_{block}) \bmod L_{GOP}). \quad \textbf{(4-2)}$$

The above condition ensure case 4 occurs. k is an arbitrary positive integer, which represents the k^{th} retrieved unit for fast-forward play.

According to the simulation, the statistic about percent of decoded video data reveals that display effect is acceptable, as shown in Fig. 7. We find that different fast-forward rates will not affect the percentage of decoded video data, namely the quality of service. In general, at least fifty percent of retrieved data can be decoded. With the growth of n_r, more percent of retrieved data can be played out without jitters.

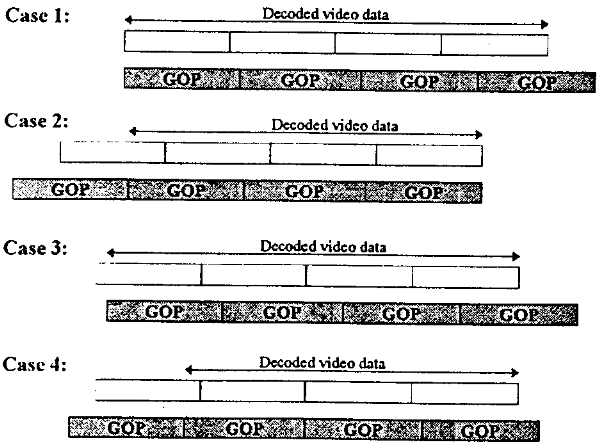


Figure 6. Cases of data retrieval by blocks, where $n_r = 4$, $L_{block} = 94$, $L_{GOP} = 112.5$

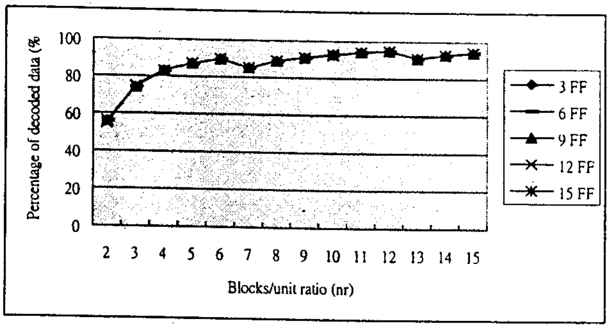


Figure 7. Percent of decoded video data, where $L_{block} = 94$ and $L_{GOP} = 112.5$

4. Simulation, System Implementation and Performance Evaluations

In order to measure the performance gains, we have not only developed a simulation environment but also implement of our ESP scheme on our video server for showing its applicability. In this section, several main evaluation gains are measured and collected. We use the simulation to measure the performance of four placement schemes: simple striping (SS), segment-placement-method (SP), prime-round-robin scheme (PRR), and our ESP scheme.

4.1 Video Data and Input Model

The system parameters of our simulation package can be classified into two categories as follows.

(i) Video data: We use 23 video files of movies with average content size from 5000x94KB to 7000x94KB [9].

(ii) Input model: We determined to adopt that average request arrival rate is equal to 1, and Zipf-like distribution follows 70-30 popularity [11]. In round-robin job scheduling policy, the service round time is 250ms. We generated 1000 requests for our evaluations.

4.2 Performance Evaluations of Load Imbalance

Fig. 8 shows the changes of imbalance factor by the time, while $N_{disk} = 8$, $R_{ff} = 4$, and $n_r = 4$. The ESP scheme differs from other methods about 30 percent in average imbalance factor. It means that 30 percent of workload is not distributed evenly.

4.3 Performance Evaluations of Reject Probability

Reject probability indicates the percentage of rejected requests by the video server before the users' requests are serviced. It is important that a good placement scheme will provide lower reject probability and let more users be serviced.

Fig. 9 depicts the maximal reject probability by the number of disks, while $R_{ff} = 4$, and $n_r = 4$. We can see, in general, the maximal reject probability decreases while the number of disks increases. This is because the video server is capable to offer more bandwidth for accessing. In Fig. 9, we can see our ESP scheme support average 20-30 percent less reject probability than other methods.

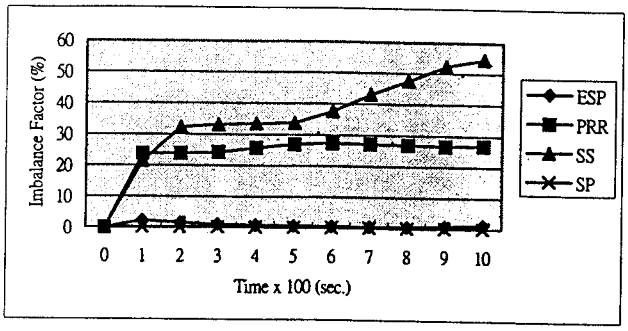


Figure 8. Imbalance factor, when ESP without shift is adopted

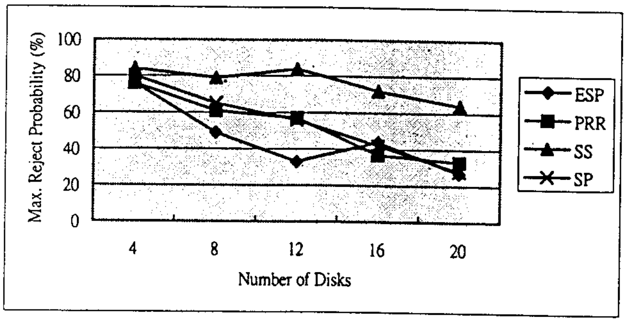


Figure 9. Reject probability, when ESP without shift is adopted

Fig. 10 shows the maximal reject probability by the unit/block ratio, while $R_{ff} = 4$, and $N_{disk} = 8$. Our ESP scheme supports average 10-20 percent less reject probability than other methods, when we adjust the value of n_r. The result of Fig. 10 reminds us that our ESP scheme benefits from striping the blocks of a fundamental unit across disks, instead of place them in the same disk. By this way, the video server can offer more requests to be permitted in a round because the workload is distributed.

Fig. 11 illustrates the maximal reject probability by fast-forward rate, while $N_{disk} = 8$ and $n_r = 4$. For our ESP scheme, the average maximal reject probability is about 75 percent. It can be seen that the reject probability is lower while R_{ff} is equal to 2 and 4. In these cases, ESP scheme without shift is adopted. As we discuss in Section 3, it will provide good load balance. Thus, more users are permitted to play out the video.

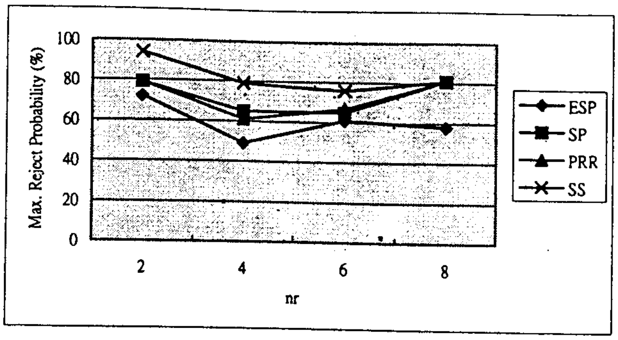


Figure 10. Reject probability of distinct placement methods, while n_r varies

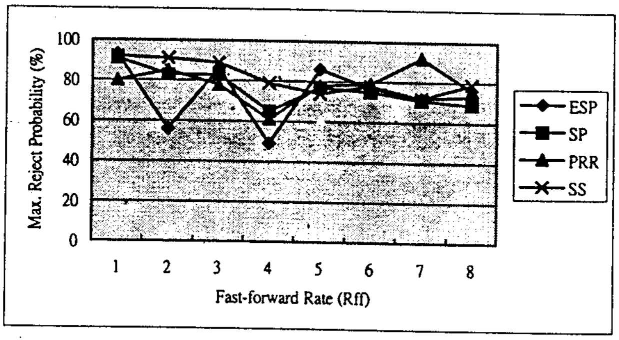


Figure 11. Reject probability of distinct placement methods, when R_{ff} varies

5. Concluding Remarks

In this paper, we have presented a new and efficient data placement scheme to support VCR operations for our video server. The proposed placement scheme is called Enhanced-Segment-Placement (ESP) scheme. The major idea is to introduce the unit/block ratio in addition to the number of disks and the fast-forward rate, which are called three key factors. We also show that ESP scheme can offer maximal throughput. In addition, we propose a strategy for our ESP scheme to determine how to adjust three key factors optimized, when we intend to place videos or change the fast-forward rate on our video server. With performance evaluations, ESP scheme is shown that workload is more balanced than other methods. Meanwhile, ESP has higher throughput with lower reject probability. Moreover, our scheme can be easily implemented on any real video server by selecting appropriate block size, GOP size, the number of disks, the fast-forward rate and unit/block ratio.

In the future, we will give further study on the following aspects. We expect to develop a job scheduling method such that, we can look forward to effectively allocate a video stream for several users. We expect to extend the placement scheme to support VCR playback and fault tolerant design simultaneously.

Acknowledgement

This research was supported by the National Science Council of the Republic of China under contract numbers: NSC 85-2622-E009-006R.

References

[1] B. Ozden, R. Rastogi, A. Silberschatz, "On the design of a low cost video-on-demand storage system," Multimedia Systems, Vol. 4, No. 1, pp. 40-54, 1996.

[2] A. Mourad, "Issues in the design of a storage server for video-on-demand," Multimedia Systems, Vol. 4, No. 2, pp. 70-86, 1996.

[3] D. Jadav, A. Choudhary, P. B. Berra, "An evaluation of design trade-offs in a high-performance media-on-demand server," Multimedia Systems, Vol. 5, No. 2, pp. 53-68, 1997.

[4] M. S.Chen, D. D. Kandlur, and P. S. Yu, "Storage and retrieval methods to support fully interactive playout in a disk-array-based video server," Multimedia Systems, Vol. 3, No.3, pp. 126-135, 1995.

[5] T. G. Kwon, Y. Choi, S. Lee, "Disk placement for arbitrary-rate playback in an interactive video server," Multimedia Systems, Vol. 5, No. 1, pp. 271-281, 1997.

[6] L. Golubchik, C. S. Lui, and R. R. Muntz, "Adaptive piggybacking: a novel technique for data sharing in video-on-demand storage server," Multimedia Systems, Vol. 4, No. 2, pp. 140-155, 1996.

[7] A. H. Wong and C. T. Chen, "A comparison of ISO MEPG 1 and MPEG 2 video coding standards," Proc. of SPIE, Vol. 204, pp. 1436-1440, 1993.

[8] D. L. Gall, "MPEG: A video compression standard for multimedia applications," Communications of the ACM, Vol. 4, No. 34, pp. 46-58, 1991.

[9] C. H. Wang, *Final report of innovative research technique development project of Science Park*, May 1996.

[10] A. L. N. Reddy and J. Wyllie, "I/O issues in a multimedia system," IEEE COMPUTER, Vol. 27, No. 3, pp. 69-74, 1994.

[11] J. W. Kim, Y. U. Lho, K. D. Chung, "An effective video block placement scheme on VOD server based on multi-zone recording disks," IEEE Trans. On Knowledge and Data Engineering, Vol. 3, No. 3, pp. 29-36, 1997.

[12] T. L. Kunii, Y. Shinagawa, R. M. Paul, M. F. Khan, and A. A. Kbokhar, "Issues in storage and retrieval of multimedia data," Multimedia Systems, Vol. 3, No. 1, pp. 298-304, 1995.

[13] P. V. Rangan and H. M. Vin, "Efficient storage techniques for digital continuous multimedia," IEEE Trans. On Knowledge and Data Engineering, Vol. 5, No. 4, pp. 564-573, 1993.

[14] Y. Wang, D. H. C. Du, "Weighted striping in multimedia servers," IEEE Proc. of Multimedia, Vol. 5, No. 1, pp. 102-109, 1997.

[15] A. Merchant and P. S. Yu, "Design and Modeling of Clustered RAID," IEEE Trans. On Knowledge and Data Engineering, pp. 140-149, 1992.

[16] S. Berson, R. Muntz, S. Ghandeharizadeh, and X. Ju, "Staggered striping in multimedia information systems," Proc. Of ACM SIGMOD, 1994, pp. 79-90.

[17] A. Silberschatz, P. B. Galvin, *Operating system concepts*, Addison-Wesley Publishing Company, Inc., pp. 410-417, 1994.

[18] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," Multimedia Systems, Vol. 4, No. 1, pp. 112-121, 1996.

[19] H. J. Lee, D. H. C. Du, "The effect of disk scheduling schemes on a video server for supporting quality MPEG video accesses," IEEE Proc. of Multimedia, Vol. 5, No. 1, pp. 194-201, 1997.

[20] P. S. Yu, M. S. Chen and D. D. Kandlur, "Grouped Sweeping Scheduling for DASD-based multimedia storage management," Multimedia Systems, Vol. 1, No. 3, pp. 99-109, 1993.

[21] C. F. Wu, *A cost-effective video-on-demand server supporting VCR operations*, Master thesis, Institute of Computer Science and Information Engineering, NCTU, June 1997.

[22] R. T. Ng and J. Yang, "An analysis of buffer sharing and prefetching techniques for multimedia systems," Multimedia Systems, Vol. 4, No. 2, pp. 53-69, 1996.

[23] A. Srivastava, A.Kumar, A. Singru, "Design and analysis of a video-on-demand server," Multimedia Systems, Vol. 5, No. 1, pp. 238-254, 1997.

[24] C. H. Chen, *Storage management of a KTV video server*, Master thesis, Institute of Computer Science and Information Engineering, NCTU, June 1996.

[25] M. M. Buddhikot, G. M. Parulkar, "Efficient data layout, scheduling and playout control in MARS," Multimedia Systems, Vol. 5, No. 3, pp. 199-212, 1997.