ELSEVIER

Information Processing Letters 80 (2001) 225–231

Information Processing Letters

www.elsevier.com/locate/ipl

Phased RGSS: An improved disk array scheduling for continuous media retrieval ☆

Han Wook Lee, Dae Woong Kim, Chan-Ik Park *

Department of Computer Science and Engineering, PIRL, San 31 HyoJa Dong, Pohang, 790-784 South Korea

Received 23 October 2000; received in revised form 12 February 2001
Communicated by K. Iwama

Abstract

This paper proposed a disk array scheduling algorithm called Phased RGSS which improves the initial service latency over RGSS presented by Kang and Park [IEICE Trans. Inform. Systems E81-D (8) (1998)] without increasing the buffer requirement for jitter-free service for continuous media retrieval. A specific data layout should be provided in order for Phased RGSS to be applied. The data layout makes disk arm movement more effective. The characteristics of Phased RGSS are proved by analysis of buffer requirement as well as initial startup delay.

Keywords: Continuous media; Disk array; Operating system; Scheduling

1. Introduction

Due to recent advances in high speed networking and storage technologies, retrieving multimedia data remotely has become a quite common operation in most distributed applications. One example system includes a multimedia server that supports the continuous playback of video stored in local disks to client via networks. Continuous retrieval of video data imposes two timing requirements to be satisfied: startup delay and playback rate. The startup delay represents the time interval during which a client should wait for video data playback after the client makes a request to a video server. We do not consider the delay coming from network bandwidth since the issue can be handled separately from disk subsystems. The playback rate specifies how fast a client consumes the video data stored in a buffer. Having a large buffer enables a loose timing requirement on the playback rate. The problem is how to design a disk subsystem for continuous video retrieval which supports a certain number of concurrent streams without violating the timing requirements of startup delay and playback rate while minimizing the required buffer size.

The disk scheduling and the underlying data layout scheme play an important role on guaranteeing the timing requirements under a certain amount of buffer space. Disk scheduling algorithms for continuous video retrieval include SCAN-EDF [1] and GSS (Group-Sweeping-Scheduling) [2,3]. And, Tobagi et al. presented StreamingRAID [4] where they applied the GSS algorithm to RAID. In particular, the GSS algorithm shows better performance than the fixed order scheduling scheme [5] and SCAN [6] in the

☆ This research has been supported partially by LG Yonam Foundation, and partially by the Ministry of Education of Korea through the BK21 program.

* Corresponding author.
E-mail address: cipark@postech.edu (C.-I. Park).

0020-0190/01/$ – see front matter
PII: S0020-0190(01)00178-8

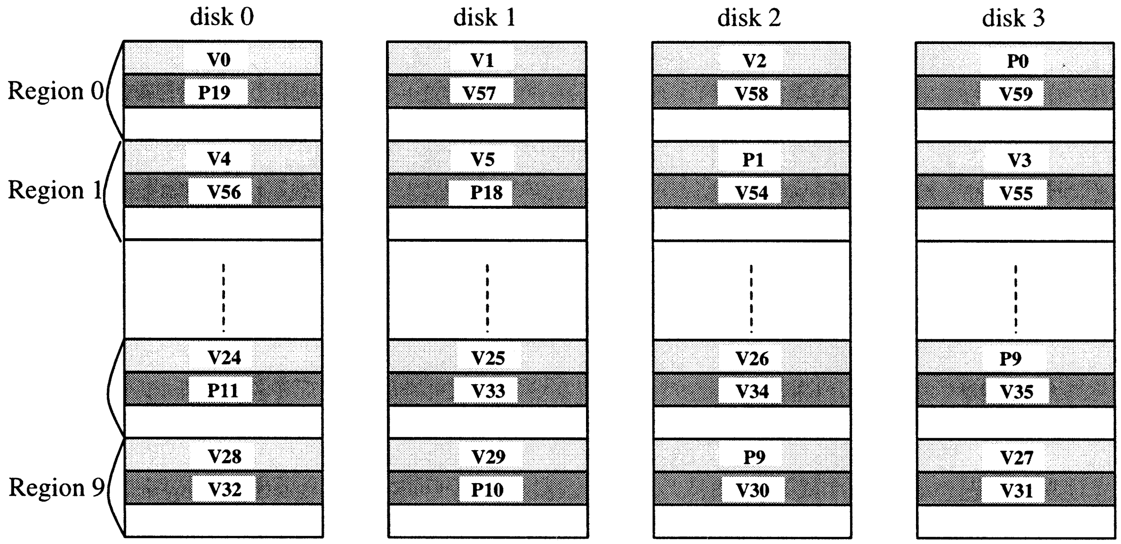

Fig. 1. An example of the region-based data layout when the number of disks is four and a disk is partitioned into ten regions.

area of continuous video retrieval. All these scheduling algorithms do not put any constraints on the underlying data layouts. Recently, the authors have proposed the RGSS (Region-based GSS) in [7] which improved the GSS by restricting its underlying data layout with the help of *region* concept [8]. The basic idea is to partition a disk space into several regions, i.e., several small disks and apply the GSS to each region. It can support a larger number of concurrent streams than the GSS for a given buffer size. However, the main pitfall of this algorithm is that the startup delay grows significantly as the number of regions increases.

This paper proposes another scheduling algorithm called PRGSS (*Phased Region-based GSS*) which has less sensitive startup delay than the algorithm in [7] while requiring equal buffer size. In the next section, previous works are explained. We will then describe how PRGSS works in detail. In Section 4, the analysis on the buffer requirement and startup delay is given. Conclusions are described in Section 5.

2. Region-based GSS

In order to improve the behavior of GSS, Kang and Park [7] proposed region-based data layout which requires that a disk space is partitioned into several regions and contiguous video data segments must be placed in consecutive regions. However, the startup delay is significantly increased as the number of regions grows in this scheme, thus making it difficult to find out the appropriate number of regions.

Fig. 1 shows how the region-based data layout works in the case that the number of disks is four and a disk space is partitioned into ten regions.

In GSS, the set of outstanding streams is also partitioned into several groups. The service period in GSS is defined as the full-stroke disk seek time multiplied by the number of stream groups since one full-stroke disk seek time is required for a single stream group in the worst case. Therefore, as the number of the outstanding streams gets increased, that is, the number of stream groups gets increased, the service period increases. Increased service period means larger buffer size for supporting the required playback rate. In order to keep the required buffer size as low as possible while achieving the required playback rate, we should keep the service period unchanged even though the number of outstanding streams increases. The RGSS [7] solved this problem by restricting the underlying data layout such that all the required video data for any stream groups can be found in a certain region of a disk space. With this restricted data layout scheme, it was shown that the

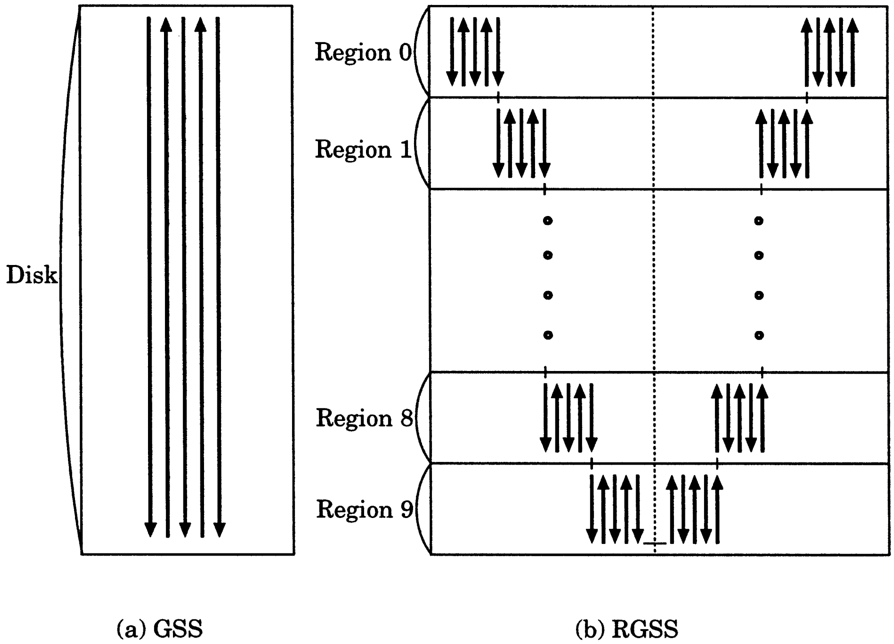

Fig. 2. GSS vs. RGSS: (a) GSS in a single period, (b) RGSS in twenty periods.

RGSS can support the required playback rate with relatively small changes of buffer size. However, the RGSS still has a significant pitfall that the startup delay is very sensitive to the number of regions in a single disk. Fig. 2 shows how the GSS and the RGSS work differently in data retrieval operation, given that the number of outstanding stream groups is five and the number of regions is ten. Note that Fig. 2(a) shows how the GSS works in a single service period whereas Fig. 2(b) shows how the RGSS works in twenty service periods, not a single period. That is, the service period in the RGSS can be significantly reduced by region-based layout, thus reducing the required buffer size.

3. Phased RGSS

In the RGSS, the playback time for video frames of a stream retrieved from a single region should be kept equal to the time for serving one frame of all the outstanding streams in a single period. Since the RGSS considers only one region in a single period, a newly arrived stream request should wait for $2R$ regions to pass through in the worst case (see Fig. 2) when a disk is partitioned into R regions.

That is, if we denote the time for serving a single frame of a video as T_{frame}, a newly arrived stream request should wait for $2RT_{frame}$ in the worst case and RT_{frame} in average before starting to get served. This time is called the *startup delay*. The worst or average length of the startup delay is very important in characterizing a video storage server. If, as in the RGSS, the playback time for video frames of a stream retrieved from a single region should be kept equal to the time for serving one frame of all the outstanding streams in a single period, the resulting startup delay is inevitably bound to the number of regions. Our proposed PRGSS scheme decouples these two values to make the startup delay less sensitive to the number of regions.

Fig. 3 shows the difference between the operations of the RGSS and the PRGSS when the number of regions is four and the number of stream groups is three. Note that eight service periods are considered in the RGSS whereas only three service periods are defined in the PRGSS. Each arrow represents a disk seek operation covering a region, i.e., the SCAN

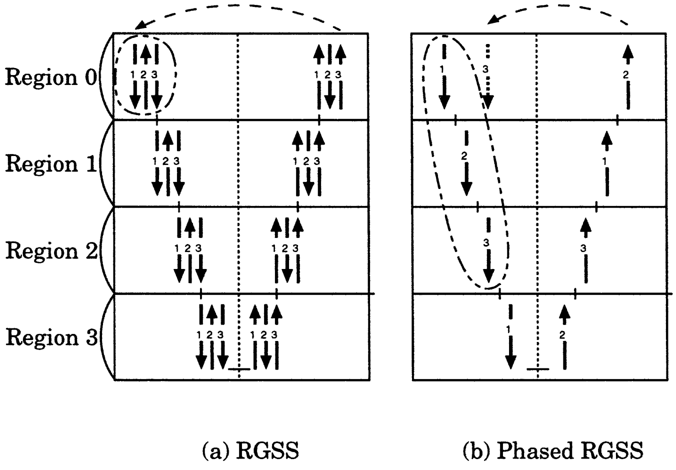

Fig. 3. The sequences of video data retrieval in RGSS and PRGSS.

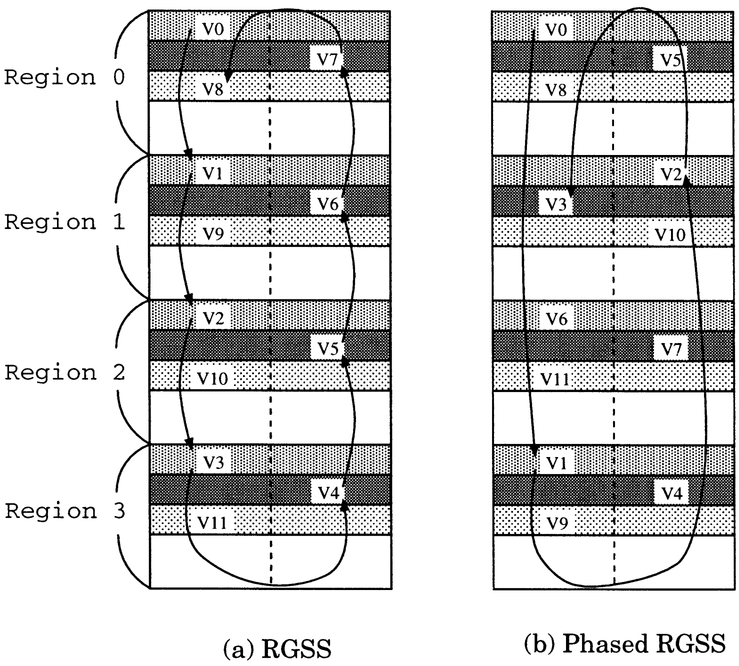

Fig. 4. Comparisons of data layouts in RGSS and Phased RGSS.

operation over the region. In the RGSS, each frame of all outstanding streams is retrieved from the same region which is currently being accessed. However, in the PRGSS, each frame of all outstanding streams is retrieved from distinct regions. It is easily seen that the distance of disk head movement remains equal in a single service period and the service order is also unchanged, regardless of scheduling schemes.

However, the PRGSS requires a more restricted data layout than the RGSS does because all regions are not considered equally for storing frames of a stream. For example in Fig. 3, the regions with frames of the first stream group are ordered as 0, 3, 1, 1, ... in the case of PRGSS, but as 0, 1, 2, 3, 3, 2, ... in the case of RGSS. Fig. 4 shows an example of a data layout in the RGSS and the PRGSS.

The data layout for PRGSS can be represented by a layout parameter denoted as c and we assume that the first frame of a stream is randomly stored. The layout parameter c specifies how many regions are skipped to locate the next frame for a stream. Fig. 4 shows an example of $c = 3$. c can be any odd value which is larger than or equal to three but not the multiple of the number of regions.

4. Analysis of Phased RGSS

4.1. Buffer requirement

Fig. 5 is drawn for buffer requirement analysis. Video segments of Group i (i.e., stream group) are to be ready in the buffer at the time t_{i+1}, and make the buffer empty by playback during the period denoted as T_{frame}. Thus, all of GSS, RGSS, and PRGSS require $n + s$ buffer blocks at worst case, where n is the number of the outstanding streams and s is the number of streams in a stream group. That is, $n = gs$ where g is the number of stream groups.

The buffer size required for a stream, m, is determined by the playback time (which is mT_p where T_p is the playback rate, i.e., consumption rate). The playback time should be made equal to the time to fill up the buffer of the size m for each stream in a single period.

Now we derive the equation on the time to fill up the buffer of the size m for each stream in a single period. Let's denote it as T_{RD}. Because, in the case of GSS, SCAN operations are required g times and the minimum seek time is inevitably required among streams, we get

$$T_{RD_{gss}} = n\left\lceil \frac{m_{gss}}{M_t} \right\rceil t_r + gS_{max} + nS_{min}, \tag{1}$$

where M_t is the track size, t_r is the amount of time taken for one rotation, S_{min} is the minimum seek overhead required for any request switching, and S_{max} is the full-stroke disk seek time. However, in the case of RGSS and PRGSS, the full-stroke disk seek time

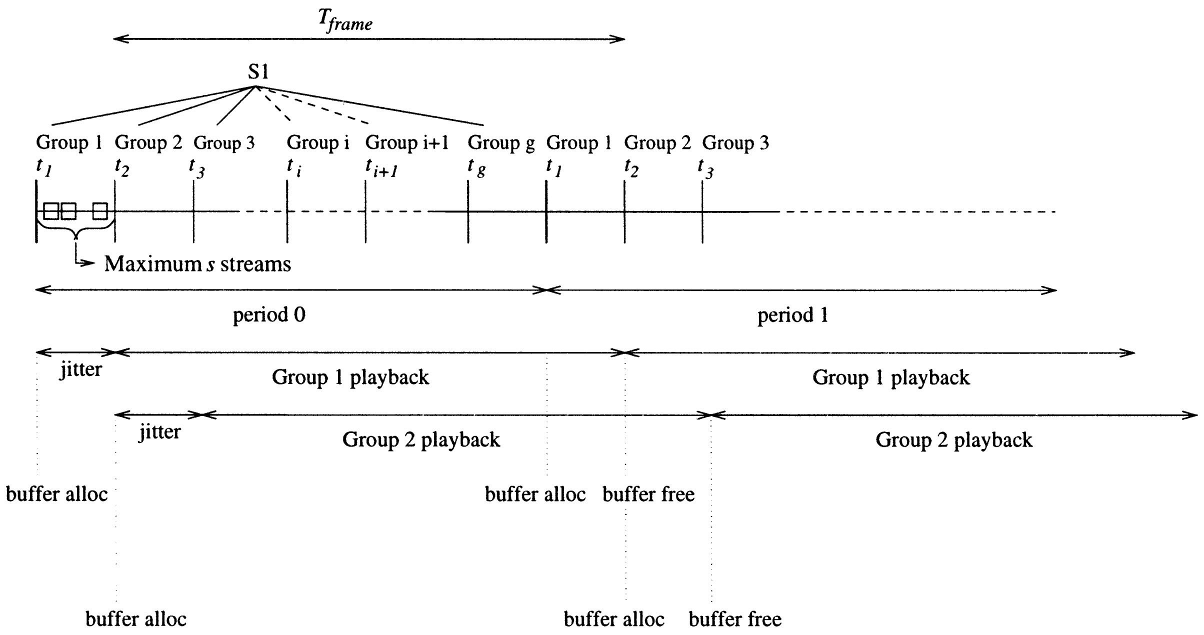

Fig. 5. The timing diagram for GSS, RGSS, and Phased RGSS.

S_{max} is reduced to S_{max}/R due to the restricted data layout. Then, we get

$$T_{RD_{rgss,prgss}} = n\left\lceil \frac{m_{rgss,prgss}}{M_t} \right\rceil t_r + g\frac{S_{max}}{R} + nS_{min}. \quad (2)$$

Since the condition of $mT_p = T_{RD}$ should hold to support continuous playback, we can derive the followings from Eqs. (1) and (2),

$$m_{gss} \geqslant \frac{M_t}{M_tT_p - nt_r}(gS_{max} + nS_{min}),$$

$$m_{rgss,prgss} \geqslant \frac{M_t}{M_tT_p - nt_r}\left(g\frac{S_{max}}{R} + nS_{min}\right).$$

4.2. Startup delay

In the RGSS, the disk head can be positioned at the same region after $2R$ periods in the worst case. Thus, a newly arrived stream request can begin playback after elapsing

$$\frac{1}{2R}\left(\sum_{i=1}^{2R} i\right)T_{frame}$$

in average, i.e.,

$$T = \left(R + \frac{1}{2}\right)T_{frame}.$$

Even though there are physically R regions, we can assume that there are logically $2R$ regions according to the direction of the disk head movement. In the PRGSS, suppose that the ith region among $2R$ logical regions is currently being served when a new stream arrives. Then, we have to wait for the disk head to access the region which contains the first video frame of the newly arrived stream. Let us denote this time as T_1. If the stream group currently under service can accommodate the new stream, then the startup delay for the new stream is equal to T_1. Otherwise, we have to wait for another stream group to come, which can accommodate the new stream. We denote this time as T_2. The startup delay T is then equal to the sum of T_1 and T_2. We now derive the expected value of T_1 and T_2, and then construct an equation of the startup delay.

The probability that current region under service is the ith when a new stream arrives will be $1/2R$ because there are logically $2R$ regions. And the time for servicing a single region in the PRGSS is T_{frame}/c because T_{frame} is equivalent to the time for servicing c regions. Then, we can derive $E(T_1)$ as follows.

$$E(T_1) = \frac{1}{2R}\sum_{i=1}^{2R}\frac{T_{frame}}{c}i = \left(R + \frac{1}{2}\right)\frac{T_{frame}}{c}.$$

In order to derive $E(T_2)$, we need to get the probability of having a stream group which can accommodate the newly arrived stream. Assume that the maximum number of streams is N and the number of the outstanding streams is n. Then, the average density of a stream group will be n/N. The set of stream groups which are served in the same region is called a *phase group*. The number of stream groups in a phase group is $\lceil g/c \rceil$ where c is the data layout parameter and g is the number of stream groups. Each phase group can accommodate upto $\lceil N/c \rceil$. This number is denoted as k. We now derive the probability of finding a stream group which can accommodate the newly arrived stream. Without loss of generality, assume that we can find a stream group in the ith phase group, that is, we can find a stream group after passing all the $i-1$ phase groups. Let's denote the probability that none of stream groups in all phase groups from the first upto the $(i-1)$th can accommodate the new stream as $P_a(i)$ and the probability that one of stream groups in the ith phase group can accommodate the new stream as $P_b(i)$.

$$P_a(i) = \frac{\binom{n}{(i-1)k}}{\binom{N}{(i-1)k}},$$

$$P_b(i) = 1 - \frac{\binom{n-(i-1)k}{k}}{\binom{N-(i-1)k}{k}}.$$

Then, the probability of finding a stream group which can accommodate the new stream in the ith phase group will be $P_a(i)P_b(i)$, we get $E(T_2)$ as follows.

$$E(T_2) = \sum_{i=1}^{c} 2R\frac{T_{frame}}{c}(i-1)P_a(i)P_b(i) = \frac{2RT_{frame}}{c}\sum_{i=1}^{c}(i-1)P_a(i)P_b(i).$$

Since $T = T_1 + T_2$,

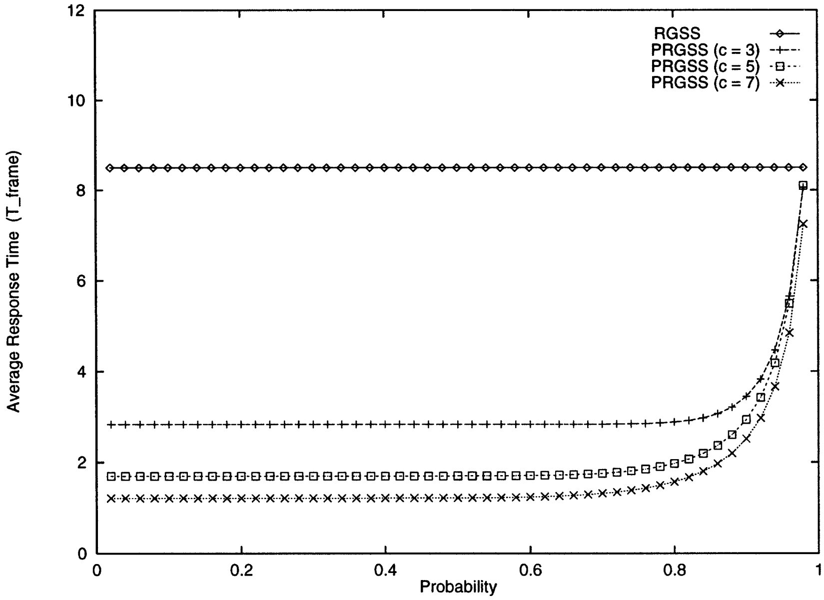

Fig. 6. $E(T)$ vs. n/N when $N = 50$ and $R = 8$.

$$\begin{aligned} E(T) &= E(T_1) + E(T_2) \\ &= \left(R + \frac{1}{2}\right)\frac{T_{frame}}{c} \\ &\quad + \frac{2R\ T_{frame}}{c}\sum_{i=1}^{c}(i-1)P_a(i)P_b(i). \end{aligned}$$

Note that the average startup delay is made less sensitive to the number of regions, i.e., R, by a factor of c in the case of PRGSS.

Fig. 6 shows $E(T)$ vs. n/N when $N = 50$ and $R = 8$. Note that the startup delay is dependent of n and c in the PRGSS while it is independent in the RGSS.

5. Conclusions

In this paper, we have proposed an improved disk array scheduling called Phased RGSS (PRGSS) for continuous video retrieval. The RGSS presented in [7] has a significant pitfall that the startup delay is very sensitive to the number of regions. Applying more restricted data layout leads us to develop the PRGSS which can make the startup delay less sensitive to the number of regions without changing the buffer requirement. The layout parameter is introduced to formulate the proposed data layout. We have derived analytic equations on buffer requirement and startup delay for the proposed PRGSS.

References

[1] A.L.N. Reddy, J. Wyllie, Disk scheduling in a multimedia i/o system, in: Proc. 1st Internat. Conf. on Multimedia, Aug. 1993.

[2] D.J. Gemmell, S. Christodoulakis, Principles of delay-sensitive multimedia data storage and retrieval, ACM Trans. Inform. Systems 10 (1993).

[3] P.S. Yu, M.S. Chen, D.D. Kandlur, Grouped Sweeping Scheduling (GSS) for DASD based multimedia storage management, Tech. Rept., IBM T.J. Watson Reseach Center, 1994.

[4] F.A. Tobagi, J. Pang, R. Baird, M. Gang, Streaming RAID^{TM} — a disk array management system for video files, in: Proc. 1st Internat. Conf. on Multimedia, Aug. 1993.

[5] P.V. Ranggan, H.M. Vin, S. Ramanathan, Designing an on-demand multimedia service, IEEE Comm. Magazine (1992).

[6] D.D. Kandlur, M.S. Chen, Z.Y. Shae, Design of a multimedia storage server, Tech. Rep., IBM T.J. Watson Reseach Center, 1993.

[7] D.Y. Kang, C.-I. Park, Synchronous RAID5 with region-based layout and buffer analysis in video storage servers, IEICE Trans. Inform. Systems E81-D (8) (1998).

[8] Y.J. Oyang, M.H. Lee, C.H. Wen, A video storage system for on-demand playback, Tech. Rept., National Taiwan University, 1994.