RESIDUAL CLASS STORAGE BASE FOR VIDEO SERVERS

Hailiza Kamarulhaili[+] *,Putra Sumari*[*]
[+]*School of Mathematical Sciences,*
[*]*School Of Computer Science*
University Science of Malaysia, Minden, Penang, 11800 Malaysia
e.mail: {hailiza, putras }@cs.usm.my

ABSTRACT
Many concepts in number theory have been adopted in various disciplines for problem solving. In this paper a method for data storage scheme in video on demand (VOD) servers is proposed. This method uses the theory of residual classes for data placement. Video on demand is a system that provides a service to users to browse and watch videos within a computer network. One of the most challenging aspects in such system is to have a VOD server with a capability of retrieving and transmitting different blocks of videos simultaneously. A part from this VOD server, it should also capable of producing high number of simultaneous streams with a low average waiting time. The proposed scheme is able to eliminate latency and hence producing maximum number of simultaneous streams. In this scheme the disk is divided into regions and blocks of videos are scattered within the regions based on residual theory rule. In addition, we also developed a stream caching technique for extra stream to help reducing the average waiting time of the system.

KEY WORDS
Video on demand server, data placement, simultaneous streams, caching streams, average waiting time.

1. INTRODUCTION

A video on demand (VOD) is an electronic video rental service over the network [1-6]. It consists of three main components namely, VOD server, transport network and user display equipments. The VOD servers store a large number of videos and act as an engine to distribute videos to users upon requests. Transport network is a medium for video delivery between servers and users. The users display equipment is a television type device for viewing purposes. VOD eliminates a specific time schedule characteristic as in traditional television program [1]. With VOD, users can view any video, any time with any interactive facilities (rewind, forward, fast rewind, fast forward, pause) and no longer constraint by the specific programmed schedule. The important task of video on demand server is to retrieve different blocks of video and sends them simultaneously to users. This is not an easy task due to the characteristic of video data. Video has a stringent real time characteristic that requires a continuous presentation in time [7]. The blocks of data have to be retrieved from the server (particularly the disk subsystem) and sent to users in real time mode. Otherwise users will experience disturbance during viewing. Video also consumes a large space due to the nature of digitized processing. For example, a frame with color depth of 24-bit-colour with 640 pixels x 480 pixels resolution produces 900 Kbytes of data. At 30-frame-per-second presentation speed, the amount is 27 Mbytes [4].

To tackle the problem we have looked into the aspect of video storage architecture [2, 3, 8, 9]. Storage architecture, we refer here is a layout of blocks of video on the disks and the supporting components. The storage architecture is usually designed for the following objectives namely, (i) to increase the number of simultaneous streams and at the same time, (ii) to minimize the average waiting time (the duration period between the moment of users press the button of particular video title and the moment playback starts). The former is

achieved by having storage architecture with a smallest latency while the latter is achieved by having an immediate attention from r/w head to retrieve the block to serve requested video. These two objectives are related to each other is some ways. Reduction of latency has to compensate with an increment of average waiting time. When a new request arrives the r/w head will be forced to finish the current job to serve the new requests.

In this paper we present a video data storage architecture that produces maximum number of simultaneous steams with a small waiting time. We introduce disk-region concept to divide the disk into equal size region. The block of video are scattered within region with real time constraint. The policy to position each block in region we use a mathematical number theory branch called residue classes as a placement policy. In this scheme latency is eliminated and hence maximum number of simultaneous streams is achieved. To reduce the average waiting time we employ streams caching technique. Ram as a caching component is added and new request is served by this cache. The new request is served by creating a replicated stream for the cache. The cache component also increases the number of simultaneously stream. The scheme is simple in retrieval and has a good load balancing.

This paper is organized as follows. Background and related works are described in section 2. The video-on-demand server and its storage component are presented in section 3. Cache

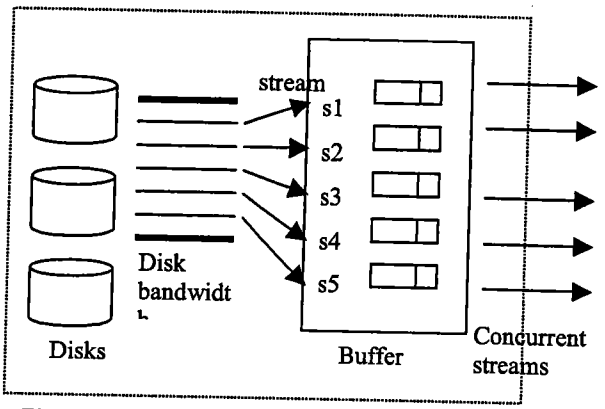

Figure 1: Video-on-demand servers architecture

component called replicated buffer to reduce the average waiting time is elaborated in section 4. Case study and simulation results are presented in section 5. Finally, the conclusion is presented in section 6.

2. BACKGROUND AND RELATED STUDIES

Video-on-demand server is an important component in video on demand application. It stores large numbers of videos and acts as the engine to retrieve and distribute video to users. A more specific responsibility is to retrieve different blocks either belonging to the same or different video and then send them simultaneously to the network. This is a challenging task since video holds real time characteristic. Blocks of video should be retrieved and sent to the network within the deadline to guarantee the continuity of simultaneous streams (stream is referred to the transmission channel of video from the server to users). One technique to handle this problem is to have a better storage scheme. Data storage is the method to lay blocks on disks along with intelligent retrieval scheduling with aims to reduce latency and to produce higher number of simultaneous streams. Another aspect that plays important role is the average waiting time. The design and enhancement of new data storage should not jeopardize the existing waiting time performance.

Numerous researchers have done great contribution to the fields. In [9, 10], storage issues in supporting real time data are investigated. The work focused on traditional random and contiguous data allocation. These approaches however are facing to much latency and not suitable for the read only category like VOD services. In [3], a circular skip cluster scheme is proposed to enhance data retrievals technique in the VOD server. In [9, 11], disks are partitioned into regions. Data are distributed within regions with aim to reduce latency and hence increase the number of simultaneous streams. In [12], the disks are divided into zones data are distributed according to the residual theory rule with aim to reduce the latency. This scheme however provides a small number of zones and limit the number of blocks sit in each region. In [2, 8, 13], proposed Phase based storage scheme. The video is organized as a column-row representation. Data are stored and read column by column. Latency is eliminated and number of concurrent streams is maximized. This scheme however has left extra space at the inner of the disk that could not be read or written. In [6, 14] then enhance the scheme by freeing the inner space of the disk and use it to storage purposes.

3.THE VIDEO SERVER ARCHITECTURE

The architecture of video on demand server in our study is shown in figure 1. The video server consists of array of disks and buffer. Videos in form of blocks are fetched from the disks and placed temporarily in the buffer before being sent to the network. These blocks form simultaneous streams as shown in figure 1. Figure 1 shows an example of five simultaneous streams labeled by $s1$, $s2$, until $s5$. The number of blocks read is determined by the data storage policy implemented in the disks. The data storage scheme normally has the characteristics of small latency (or zero latency), higher number of simultaneous streams and small waiting time. Clearly indicate advantages, limitations and possible applications.

3.1. Disk-region layout

In this scheme disks are divided into regions equal size as shown in figure 2. The figure shows n regions labeled as $R1$ to Rn. The first region is labeled by $R1$ toward the last region is labeled number with Rn. We now derive the number of regions and with $[x]$ indicates the greatest integer less than or equal to x, we have the following notation.

$$N = [disk\_size/disk\_rate]. \quad (1)$$

We then derive the number of simultaneous stream p as follows:

$$P = [disk\_rate/video\_rate]. \quad (2)$$

With equation (2) and d (block size) is equal to *video_rate*, we can conclude that p is the maximum number of simultaneous stream the system can support. Each region contains p blocks to represent p simultaneous streams. The disk bandwidth (*disk_rate*) is fully occupied by p simultaneous streams.

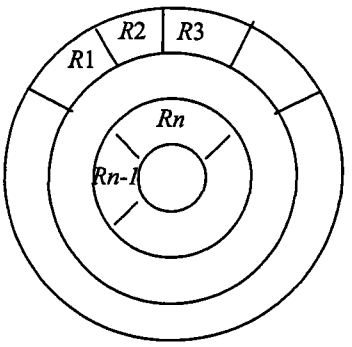

Figure 2 : Disk_region concept: disk is divided into *n* regions

Ri	Region *i*th
disk_size	Disk capcity
disk_rate	Disk bandwidth
video_rate	Playback rate
p	Number of simultaneous streams
d	Block size
a_i	First block of stream *a*
a_{i+1}	Second block of stream *a*
m	Matrix row

Table 1 : Parameter used in the study

Figure 3 shows *p* blocks labeled by a_1, a_2, a_3, ..., and a_p are placed in region *r1*. The next subsequent *p* blocks labeled by a_{1+1}, a_{2+1}, a_{3+1}, ..., a_{p+1}, are placed in region *R2*. The gap between two subsequent blocks is bounded by real time constraint. E.g. Block a_i and a_{i+1} (see figure 3) are at the right distance for preserving the real time. Equation (1), (2) and elaborated explanation of figure 3 shows that latency is eliminated. *P* blocks located at region *R1* are sent simultaneously to the network. The following subsection explains the disk-region concept in multiple disks.

3.2. Data placement on multiple disks

Section 3.1 has shown a disk is divided into regions. In multiple disks, assuming all disks are identical in capacity, the number of regions of each disk is equal. With *m* disks and *n* regions, we can represent this as (*m* x *n*) matrix as shown in figure 4. The row indicates number of regions and the column indicates number of disks. The (*m,n*) is the indicator of block location. E.g. Position (*m,n*) means block *i* sits on disk *m* and at region *n*.

Our data placement scheme is based on a concept of number theory known as residual classes. Here we denote the set of all integers such that $m \equiv i \bmod (y)$ as residual of classes *i*

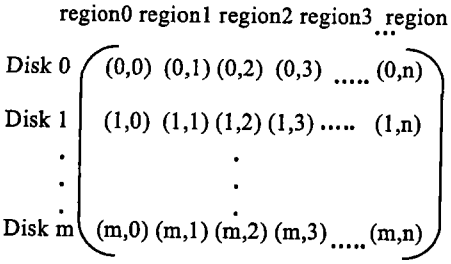

Figure 4 : Matrix representation for region concept

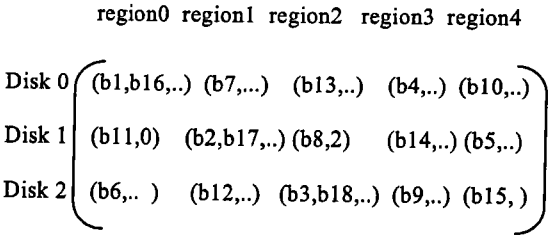

Figure 5 : block b1, b2, ..., bk are striped across disks and regions

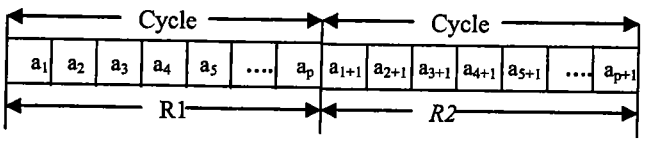

Figure 3: The bounded of gap between block a_i and a_{i+1} for real time characteristic.

m modulo y. For instance if $y = 3$, then we have 3 residual classes which are 0, 1,and 2. The class of 0 contains all integers that when divided by 3 leave no residue. The 1 is all the integers when divided by 3 leaves 1 as residue and so on. For our case study, we have two integers m and y with $(m, y) = 1$. Another word, the greatest common factor of m and y is equal to 1, with an additional condition that $y = [n]_m$, where $[n]_m$ means the greatest relative less or equal to n which is relatively prime to integer m. From the following formula, the i-th block of video is then stripped and scattered across m disk and n regions. That is, the i-th block is place in

(region number $\equiv i(mod)m$, and disk number $\equiv i(mod)n$) .

With condition: maximum number of blocks in each region is equal or less than p blocks. For example, let the number of disk (m) is 3 and the number of region (n) is 5. As shown in figure 5, the video is composed of blocks b1, b2, b3,.., bk. The data placement of the video is shown in figure 5. With equation (3), block b1, b16 are stored at location (0,0), subsequent block b2, b17 are stored at location (1,1) and so on until end of block. In our scheme the r/w heads of all disks should be synchronized and should be simultaneously read the same block at the same region. By the above placement, blocks of video are striped across disks and spread over all disks. It is a good property of load balancing during reading.

3.3. Retrieval and buffering

In this scheme the policy the r/w head will read all blocks at one region before move to the next region for the next subsequent blocks. When reached the last region the r/w head returns back to the first region to repeat the operation. We use double buffering concept due to simplicity. In this scheme buffers of size $2pd$ is required. To visualize the buffering operation, lets buffer is divided into two rooms of size pd each. While the first buffer is filled with data of size pd, the second buffer sent out data of size pd. Then the role switch that is the second buffer now is filled with data of size pd and at the same time the first buffer sent out data of size pd. This preserved the real time constraint of video data.

4. AVERAGE WAITING TIME

The waiting time is the time taken by the r/w head to read all regions. The worst-case waiting time is the total length of regions. E.g. Let the first block of video i is located in region 1. When the new request arrives and the r/w head just passed region 1 then waiting time of the new request is the time length of the r/w heads to read all region and get back at region 1. If we look at scenario with 200 regions. The worst case waiting time is 200 seconds (3.3 minutes), that is the total length of all regions. The average waiting time is 100(200/2) seconds (1.6 minutes). With regard to this we also provide a method to decrease this waiting time using by stream caching technique. One characteristic of video is popularity. The most popular videos will be viewed by most of the users at any certain of time. Therefore by granting popular videos to be cached in the ram, it will allow a new stream, we referred to as replicated stream, to be created from the cache. Since the new request do not have to wait until r/w head get back at the first region instead it is served by cache therefore the average waiting time of the system is reduced.

To further detail of the technique, we use figure 6 with explanation. Figure 6a shows m disks with n regions. Blocks of video i are distributed within m disks using equation (3). A pool of ram is provided to support the technique. A pool of ram consists of replicated buffers as in figure 6b. Caching operation works as follow, when the r/w head reads the first block of video i, at the same time this block is copied into replicated buffer as shown by solid arrow in figure 6. When second block is read, at the same time this block is copied into the second room of replicated buffer. This operation goes on until all rooms in

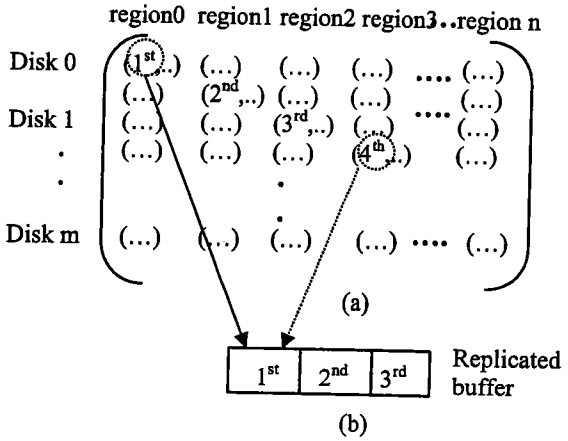

Figure 6 : (a)Data placement of video using residual classes theory
(b) Replicated buffer use to cached blocks of video

replicated buffer are occupied. At this moment, new requests are served using replicated stream. To sustain the continuity of replicated stream, the current block read by r/w head is copied into the first room at replicated buffer to replace the block just sent to network. This fashion will go on until end of video. If we set the size of replicated buffer is *n/2*, waiting time is reduced up to 50%. The replicated buffer will be used until nothing left and at this moment the operation will be back to original scheme as without replicated buffer.

5. SIMULATION

In this section we conduct a simulation for measuring waiting time performance of the system. We consider four disks with the following characteristics: Seagate Baracuda with 2.25 GB capacity and 75 Mbps bandwidth rate. We use five MPEG 1 standard videos with 60 minutes in length. With four disks we derive 240 regions to form (4 x 240) matrix. The size pool of RAM is 500 MB. The placement of blocks satisfy equation (1), (2) and (3). The simulation was developed using the CSIM/C++ discrete event simulation language [15]. The popularity of each video is determined based on Zifpian (*z*) [16] distribution and the characteristic of users arrival rates is based on Poisson (*p*) distribution. The popularity of each video and the characteristic of users arrival rate can be varied by changing the distribution factor *z* and *p* respectively. We adopt the default value of both Zipfian and Poisson parameter by $z = 0.2$ and $p = 8$ respectively [8].

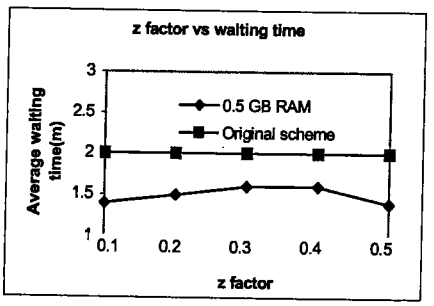

Figure 7: Popularity vs. avg. waiting time

Figure 7 shows the effect of different video popularity against the average waiting time. The value of *p* is fixed at 8 as default value. In overall the figure shows that original scheme gives two minutes average waiting time. With 0.5 GB replicated buffer, the average waiting time is decreased 20%. The decrement is because of most popular videos are cached in the replicated buffers and new served most of the new requests. Otherwise the R/W head has to roll back to the first region to serve a new stream. In the case of $z = 0.1$ the lowest waiting time is obtained. This is because as *z* become smaller, the access pattern more localized to individual video that is identified as the most popular. The case of $z = 0.3$, a slight increase of waiting time due to fair popularity among video and therefore the access pattern is distributed fairly to all videos.

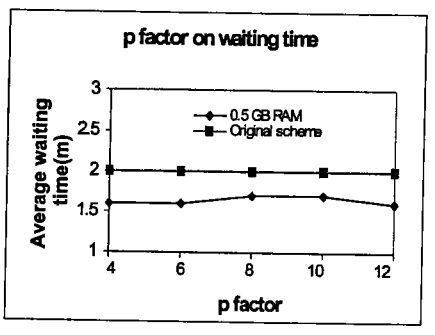

Figure 8: Arrival rate vs. avg. waiting time

Figure 8 shows the effect of different rate of users arrival rate against the average waiting time. The z is fixed at 0.2 as default value. The figure shows that the different access pattern due to p variability does not affect the average waiting time of the system.

6. CONCLUSION

This paper presents a data storage scheme for video on demand server. The scheme eliminates latency and hence produces maximum number of simultaneous streams. In this scheme disk-region is introduced. Disks are divided into small regions and data are striped in all regions. The striping policy within region is determined by number theoretical concept known as residual classes. The data placement within regions is designed as such that subsequent blocks of the same video are placed at different region bounded by real time characteristic. We also provide the RAM known as a cache for stream caching purposes. The ongoing blocks are copied into cache so that new requests can be served.

7. ACKNOWLEDGEMENT

This work was supported by the Universiti Sains Malaysia (USM) grant. Ref. number 304/PMATH/633114

REFERENCES

[1]. Srivastava, A., Kumar, A., and Singru, A., "Design and Analysis of a Video-on-demand Server," *Multimedia Systems , Vol. 5*, No. 4, pp. 238-254, 1997.

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

[3]. Chang, C. K., and Shih, C.-C., "A circular skip-cluster scheme to support video-on-demand services.," *Multimedia System, Vol. 7*, No. 2, pp. 107-118, 1999.

[4]. Furht, B., Kalra, D., Kitson, F. L., Rodriguez, A. A., and Wall, W. E., "Design Issues for Interactive Television Systems," *Computer, 1995, Vol. 28*, No. 5, pp. 25-39.

[5]. Li, V. O. K., and Liao, W., "Distributed multimedia system," *Proceeding of the IEEE , Vol. 85*, No. 7, pp. 1063-1108, 1997.

[6]. [6] Sumari, P., Merabti, M., and Pereira, R., "Video-on-demand Server: Strategies for improving performance.," *IEE Proceedings Software, Vol. 146*, No. 1, pp. 33-37, 1999.

[7]. Gemmell, D. J., Vin, H. M., Kandlur, D. D., Rangan, P. V., and Rowe, L. A., "Multimedia Storage Server: A Tutorial," *Computer, 1995, Vol. 28*, No. 5, pp. 40-49.

[8]. Chua, T. S., Li, J., Ooi, B. C., and Tan, K. L., "Disk Stripping Strategies for Large Video-on-Demand Servers," in *The 4th ACM International Multimedia Conference*, Boston, MA, pp. 297-306, 1996.

[9]. Ghandeharizadeh, S., Kim, S. H., and Shahabi, C., "On Configuring a Single Disk Continuous Media Server," in *SIGMETRICS Performance Evaluation, Ottawa Ontario Canada, Vol. 23*, No. 1, pp 37-46, 1995.

[10]. Liu, J. C. l., Du, D. H. C., and Schnepf, J. A., "Supporting random access on real-time retrieval of digital continuous media," *Computer Communications, 1995, Vol. 18*, No. 3, pp. 145-159.

[11]. Chang, E., and Garcia-Molina, H., "Reducing Initial Latency in Media Servers," *IEEE Multimedia*, Vol. pp. 50-61, 1997.

[12]. Wu, C.-S., Ma, G.-K., and Liu, M.-C., "A scalable storage supporting multistream real-time data retrieval," *Multimedia System, Vol. 7*, pp. 458-466, 1999.

[13]. Subrahmanium, V. S., and Jajoda, S., *Multimedia Database Systems: Issues and Research Directions*: Springer, New York, 1996, pp. 237-261.

[14]. Sumari, P., Merabti, M., and Pereira, R., "Video On Demand Server: Storage Architecture with Disk Arrays," in *Proceeding of the Eurograpics workshop (Multimedia '99)*, Milan, Italy, pp. 73-81, 1999.

[15]. Schwetman, H., "CSIM18 - The simulation Engine in Mesquite Software, Inc, Austin, TX.," , 1996.

[16]. Zipf, G. K., *Human Behaviour and the Principle of Least Effort* . Reading, MA: Addision-Wesley, 1949.