A Continuous Media Placement Scheme in Multi-Zone Disks

Li-Chun Sung
Department of Computer Science and Information Engineering
Tamkang University,
Taiwan ROC
g7190030@tkgis.tku.edu.tw

Chin-Hwa Kuo
Department of Computer Science and Information Engineering
Tamkang University,
Taiwan ROC
chkuo@mail.tku.edu.tw

Meng-Chang Chen
Institute of Information Science
Academia Sinica
Taiwan ROC
mcc@iis.sinica.edu.tw

Ming-Yi Lai
Department of Computer Science and Information Engineering
Tamkang University,
Taiwan ROC
g8190304@tkgis.tku.edu.tw

ABSTRACT

A systematic continuous data placement scheme in distributed multi-zone disks is developed for video on demand. The developed scheme maximizes not only the averaged data transmitted rate, but also the number of simultaneous accesses. The scheme consists of the following components: (1)the *Retrieval Sequential Model* to achieve disk load balancing, maximize data throughput, and simplify the issue of admission control; (2)the *Idle Round* scheme to reduce the buffer size required at the client site for VBR video stream. Finally we perform experimental tests to evaluate the performance of the proposed scheme.

1.Introduction

As a result of advances in computer system and network technologies, real time transmission of multimedia through networks becomes feasible. In the mean while it also poses new challenges to designers.

Generally speaking, the VoD server design is a very popular research issue today. At the present time, most PC-based multimedia servers make use of SCSI bus to connect multiple disks as data storage systems. For such a system, in principle, it can allow more than hundred of simultaneous requests. However, due to the hardware characteristics of disk, extra service time, e.g., disk seek time and rotational latency, does require. Furthermore, due to the characteristics of video, compression may lead to *Variable Bit Rate* (VBR) stream. Under such a circumstance, designs that based on the peak rate condition may not fully utilize the available system resources. As a result, the services that a server can support is much below the theoretic value. Therefore, an effective data placement and retrieval scheme for VoD is of importance.

Related works

Research devoted to VoD design can be categorized into two types. The one elaborates disk hardware characteristics in reducing data access time and the one focuses on variable bit rate video characteristics to develop continuous media access technique. In the first approach, disk scheduling techniques such as SCAN, C-SCAN, and GSS are proposed [1][2]. Furthermore, region-based data storage strategy is developed to reduce the disk seek time [3]. By integrating C-SCAN and region-based data storage strategy, one can reduce the access startup latency significantly. In multi-disk system, SCSI is a cost effective selection [4]. The disconnect and reconnect features supported in SCSI allow concurrent accesses of different disks, hence increasing the number of video playbacks. The data granularity and disk loading balance issues are explored in [5][6]. To increase the storage capacity, disk zoning scheme is used. This scheme used in VoD server was studied in [7][8]. In the second approach, *Constant Time Length* (CTL) and *Constant Data Length* (CDL) are the most data access schemes for continuous media streams. The disadvantages of CDL scheme are that increases the buffer size and startup latency. In case of CTL, it may results in poor system resource utilization or complex system management scheme. Smoothing technique is introduced to overcome the above weakness. Feng et al. proposed the ABA scheme [9] and SWS scheme. However, these schemes are not suitable for the case when limited buffer size available at the client end. To deal with the above problem, MVBA, CBA, and MCBA schemes were developed [10][11].

In our approach, we elaborated the characteristics of disk hardware and video streams. We designed and implemented an efficient video placement scheme in multi-zone disks. This proposed scheme integrates two schemes, namely (1) *Retrieval Sequential Model* and (2) *Idle Round*. First, based on the concept *Constant Read Time* (CRT)[12], we rearrange a multi-zone disk into several logical zones and design a data placement and retrieval sequence model. By using this model not only results in efficient disk resource utilization and large data transmission rate but also simplifies the issues such as bandwidth management and admission control. Second, by using the *idle round* technique, we proposed a scheme, which reduces the buffer size required at the client end. The scheme is extremely useful to the case that only limited buffer size is available at the client end. Finally we perform the experimental tests to verify our schemes.

2. System Overview

In this section, first we briefly introduce the architecture of multi-disk VoD system and design parameters of concern in our design. Then we describe the concepts of disk zoning and *Constant Read Time* data access scheme to facilitate our later presentation.

2.1 Multi-disk VoD system architecture

At the present time, computer memory systems usually organize in hierarchy manner. Up to date, magnetic disk is still the main secondary storage device in support of on-line access multimedia services [8]. Due to the limitation of storage capacity in single disk, as shown in fig. 1, a media server connects to several magnetic disks through fast wide SCSI-2 bus [13]. It may also connect to a tertiary storage system, such as optical disks stack. On the other end, this server connects to a network. Users can access the material of interest through networks.

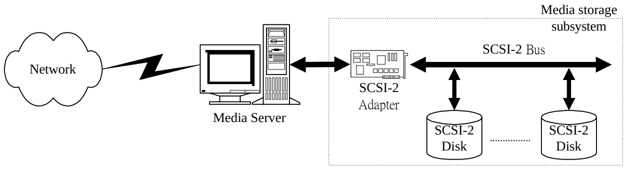


Fig.1 Multi-disk VoD server architecture

2.2 Constant Read Time

In the disk systems, each disk drive rotates in constant angular velocity. To maintain a constant data throughput, the disk has high data storage density in the inner track than the outer track. Thus, to meet the demands for higher storage capacity, disk-zoning technique is introduced. By using this technique, the storage space towards the outer portions of a disk platter is larger than the inner portions. More data may be recorded in the outer portions as a result. Since the disk rotation speed is constant, disk zoning technique produces multiple transfer rates, and hence, increasing the design complexity.

Continuous media access techniques are divided into two categories, namely *Constant Data Length* (CDL) and *Constant Time Length* (CTL). Both schemes are based on round-based data access method. Readers are referred to [14] for further details. In conventional disks, the CDL scheme can be illustrated by Eq. (1). In each service round,

$$B=R\,T_s \qquad (1)$$

where B denotes the amount of data transmitted (Bytes), R represents the data transfer rate (Bytes/sec), and T_S indicates the service time (sec).

That is, due to the data transfer rate R and the service time T_S are treated as constants, the amount of data transmitted B in each service round is fixed. In other words, the disk system delivers constant amount of data and maintains a fixed service time in each service round.

In the case that disk employs zoning technique, Eq. (2) is generalized as

$$B(Z_i)=R(Z_i)\,T_s(Z_i). \qquad (2)$$

That is the transmission rate $R(Z_i)$ depends on which physical disk zone Z_i is accessed. So to maintain the same amount of data transmitted in each round, the service time is different for different physical zone. Thus, the admission control in this case, like CTL, becomes complex to manipulate.

In the case of CRT, constant service time is used in each service round. As indicated in Eq. (3), the transmission rate depends on which logical zone LZ_i is accessed. These logical zones are obtained from the rearrangement of the original physical zones. Since the service time T_S is a constant, the total amount of data transfer B also depends on logical zone. In multi-zone disk systems, the amount of data transfer will be different in each service round. Hence Eq. (2) can be modified as

$$B(LZ_i)=R(LZ_i)\,T_s. \qquad (3)$$

The differences of CDL and CRT are indicated in Fig. 2. In the case of variable bandwidth disk systems, the transfer rate and service time both are different, but the amount of data access in each round is the same for CDL. On the other hand, both the transfer rate and the amount of data access are different, but the service time is the same in each round for CRT.

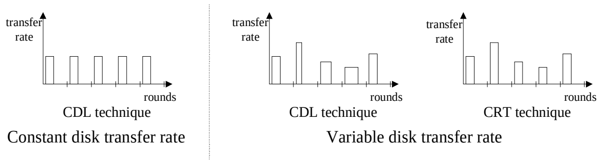


Fig.2 Comparison of CDL and CRT

3. Retrieval Sequential Model

In this section, based on the concept of CRT, we divide a multi-zones disk into several logical zones in the sense that the averaged disk throughput is maximized. Then, we developed a continuous media placement scheme, named *Retrieval Sequential Model*(RSM), which is able to achieve disk load balancing in media access process. As a result, the admission control and playback interactivity of the VoD system can be done in an effective manner.

3.1 Logical zone partition

In order to maximize the disk throughput, we use a logical zone partition strategy [15] in the design of our data placement scheme. The partition strategy is based on the CRT concept. To result in the disk loading balance purpose, each logical zone contains the same number of data blocks. It needs to satisfy

$$\mu=\frac{S(LZ_1)}{R(LZ_1)} = \frac{S(LZ_2)}{R(LZ_2)} = \Lambda = \frac{S(LZ_{LZ\#})}{R(LZ_{LZ\#})}$$

where μ represents the data access time for a complete logical zone, LZ_i represents the i-th logical zone, $LZ\#$ denotes the total number of logical zone, $R(LZ_i)$ indicates the data transmission rate of the i-th logical zone, and $S(LZ_i)$ shows the storage capacity of the i-th logical zone.

3.2 Design of the Retrieval Sequence Model

Usually, the partition of a common disk results in more than one logical disk zone. At the present time, video servers require more than one disk to match large volume storage requirement. Hence the video storage system configures as a multi-disk and multi-zone environment. If each video stream starting logical disk zone is the same [16], a *double round-robin* placement scheme results in data transmission throughput out of balance. If disk throughput in each round changes significantly, it will affect the number of accesses at the same time. In other words, disk bandwidth may not be fully used in most rounds. Therefore, the data placement in the above situation is a crucial design issue.

We proposed a data *Retrieval Sequence Model* (*RSM*) to resolve the above issue. To facilitate our presentation, the notation LZ_j^i represents the j-th logical zone of the i-th disk. For a given disk storage system, it has N_{disk} disks and each has $LZ\#$ logical zones. Then, we can construct up to N_{disk} logical data access sequences. Each logical sequence contains a set of unique $LZ\#$ logical zones. The above can be represented as

$$for\ i = 1, \ldots, N_{Disk}\ and\ j = 0, 1, \ldots, LZ\# - 1$$

$$RS_i(j) = \left\{ LZ_{f'(i,j)}^{f(i,j)} \middle| \begin{array}{l} f(i,j) = (i + j - 1) \bmod N_{Disk} + 1 \\ f'(i,j) = (f(i,j) + j - 1) \bmod LZ\# + 1 \end{array} \right\}$$

where RS_i represents the logical data access sequence starting from the i-th disk.

With the developed logical access model, we are in the position to present our data placement scheme as follows:

$$RSM = \left\{ RSS_i \mid for\ i = 1, 2, \ldots, N_{Disk} \right\}$$

$$RSS_i = \left\{ (RS_{M(i)}\ \ RS_{M(i+1)}\ \cdots\ RS_{M(i+N_{Disk})})^+ \right\}$$

$$RSS_i(j) = RS_{M'(i,j)}((j-1) \bmod LZ\# + 1)$$

$$where\ M(i) = (i-1) \bmod N_{Disk} + 1$$

$$M'(i,j) = \left(i + \left\lceil \frac{j}{LZ\#} \right\rceil - 2\right) \bmod N_{Disk} + 1$$

In the above, *RSM* represents the data sequential model, RSS_i denotes the i-th logical data access model, and $RSS_i(j)$ indicates the j-th logical zone in the i-th logical access model.

As shown in Fig. 3, depending on the choice of logical data access model, the scheme constructs different data sequential access sequences. It is a *double round-robin* scheme with *skew zone* retrieval pattern as indicated in Fig. 4(a). In other words, the starting access logical zone of different disks is organized as skew pattern. When video streams are accessed concurrently, the above data access model will access the corresponding disks to ensure disk-loading balance. In this way, the number of accesses of a video server can increase. As shown in Fig. 4(b), by following the design algorithm, each disk throughput is different in each service round. However, the total disk throughput is maintained as a constant.

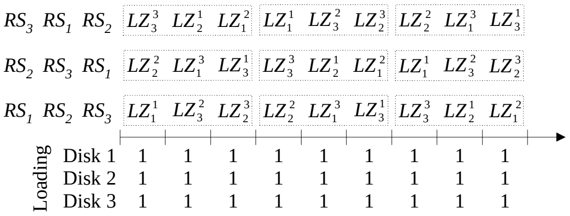
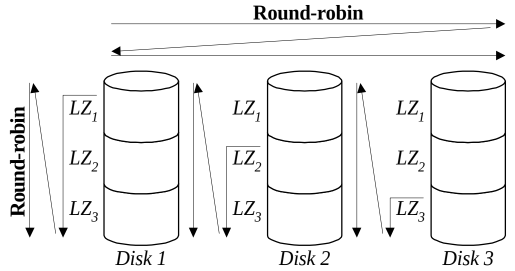
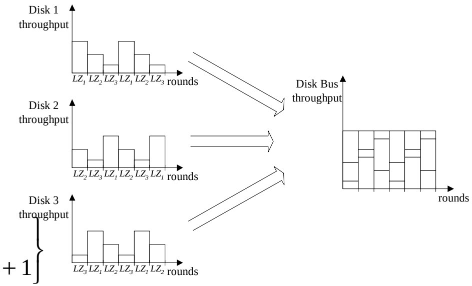


Fig.3 Data access model in multi-zone disks



(a) access scheme



(b) Disk bus throughput

Fig.4 Skew zone retrieval scheme

3.3 Service time for continuous playout

To determination the service time T_S for each service round, we work in backward manner. For a given video stream, first we find out the service round that most likely to causes the starvation phenomena. We call this service round as $P_{Critical}$. To prevent the starvation phenomena, before video playout reaches this round, the system needs to access more data than required in each service round to prepare for the data required at the $P_{Critical}$. Thus, for a given video stream in a N_{Disk} multi-disk system we have

$$\begin{array}{l} for\ i = 1, \ldots, N_{Disk} \\ \quad for\ j = 2, 3, \ldots, RN(Stream) \\ DRF_i(j) - DCF(j+1) \geq 0 \end{array} \tag{4}$$

where *RN(stream)* represents the amount of service rounds required, DRF_i represents video data retrieval function for the *i*-th sequential data access model, and *DCF* represents the data consumption function.

During the playout, the $P_{Critical}$ round for a given video is determined by Eq. (5), i.e., the maximum averaged data consumption rate

$$\begin{array}{l} for\ j = 2, 3, \ldots, RN(Stream) + 1 \\ \dfrac{DCF(P_{Critical})}{P_{Critical}} = \max\left(\dfrac{DCF(j)}{j-1}\right) \end{array} \tag{5}$$

To ensure the playout continuity, $P_{Critical}$ has to satisfy Eq.

(4). Substitute $P_{Critical}$ into Eq. (4), we have

$$DRF_i(P_{Critical}-1)=\sum_{r=1}^{P_{Critical}-1} R(RSS_i(r))\times T_S \ge DCF(P_{Critical})$$

And the minimum T_S is $\dfrac{DCF(P_{Critical})}{\sum_{j=1}^{P_{Critical}-1} R(RSS_i(j))}$.

Note that the service time T_S is dependent on the sequential data access model. In the optimization, it needs to go through all N_{Disk} retrieval sequences.

From the above analysis, we developed an algorithm, called *min_service_time* algorithm, in the computation of T_S. The pseudo code is given in Fig. 5.

```
min_service_time( )
    $T_S = 0$
    for j = 2 to RN(Stream) + 1
        if $\sum_{t=1}^{j-1} R(RSS_i(t))\times T_S < DCF(j)$
            $T_S = DCF(j) \Big/ \sum_{t=1}^{j-1} R(RSS_i(t))$
        end
    end
    return($T_S$)
end
```

Fig.5 Pseudo code of *min_service_time* algorithm

4. *Idle Round* scheme

In a common playout device such as set top box, it only consists of limited buffer sizes. When the data access quantity in each service round over the available buffer sizes, the playout continuity is damaged. Generally speaking, the maximum buffer sizes required at the client site B_{Client} depends on the maximum difference of the accumulated data retrieval and the data consumption in the video playout process. This can be represented as

$$\begin{aligned} &for\ \ j=1,2,\ldots\ldots,RN(Stream)\\ &B_{Client}=\max\{\,DRF_i(\,j\,)-DCF(\,j+1\,)\,\}\end{aligned} \qquad (6)$$

Since DCF is dependent on the characteristics of corresponding video stream, i.e., CBR or VBR, and DRF_i is dependent on the data access model and the service time, the determination of buffer size at the client end needs to take these into account. In the following, we propose the schemes in the determination of buffer size for CBR and VBR video streams.

4.1 Determination of buffer size for CBR

In the case of CBR video stream, as shown in Fig. 6, the DCF is linear and the DRF is piecewise linear. The maximum buffer size required, by following Eq. (6), is the product of the maximum disk transfer rate and the service time T_S. Thus, by using the *min_service_time* algorithm developed in Section 3.3, the required buffer size can be computed.

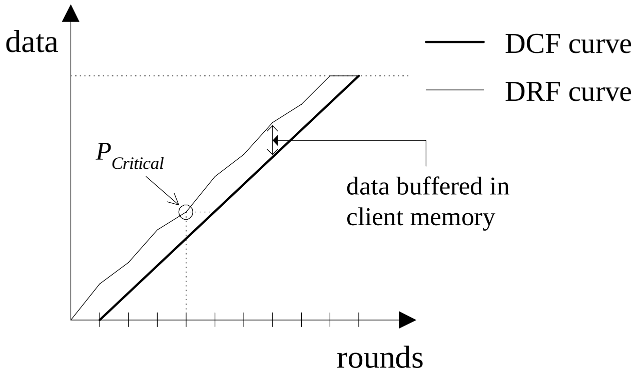


Fig.6 Determination of buffer size in CBR case

4.2 Determination of buffer size for VBR

In the case of VBR video stream, as shown in Fig. 7, if the *min_service_time* algorithm is employed the buffer size required in both sides of $P_{Critical}$ may be much larger than the amount of data consumption in a single round. The *min_service_time* algorithm becomes inadequate in the case of VBR. To resolve this problem, based on the ON-OFF scheme proposed in [17], we developed the *idle round* technique. In the idle rounds, the system does not access any data to avoid extra buffer size required. In other words, if the system does not access any data in the service round and this will not cause any playout starvation in the next round, then this service round becomes an idle round. As shown in Fig. 8, the buffer size that required is reduced when the *idle round* scheme is employed. After showing the usefulness of *idle round* technique, next we present the scheme that using idle round in reducing the buffer required at the client end.

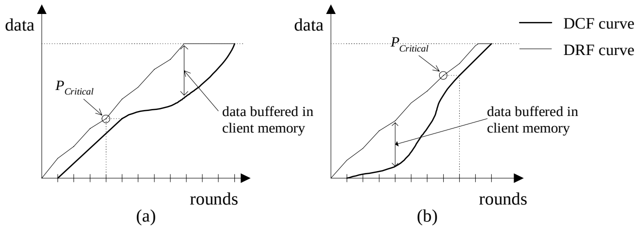


Fig.7 Determination of buffer size by using the minimal T_S algorithm

Depending on the data consumption function, the situation can be divided into two cases, (1) low data consumption rate after $P_{Critical}$ (see Fig. 7(a)) and (2) low data consumption rate before $P_{Critical}$ (see Fig. 7(b)).

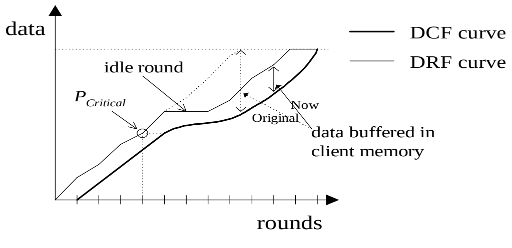


Fig.8 Reducing buffer size by *idle round* scheme

(1) low data consumption after $P_{Critical}$

In the case of low data consumption after $P_{Critical}$, the system requires to reduce the data retrieval rate after $P_{Critical}$. If the *j-th* round can be an idle round, it satisfies the following inequality.

$$for\ r = j, j+1, \ldots, RN(Stream)$$

$$DRF_i(r-1) + \sum_{t=j}^{r} B(RSS_i(t)) - DCF(r+1) \geq 0$$

Note that the above equation indicates that to inspect the j-th round is an idle round or not, we need to check the inequalities for the remaining rounds. This may become the computation burden. We make use of $P_{Critical}$ to simplify the computation. Defining that D_{Reduce} to be the difference of original data retrieval and the data retrieval after introducing idle round, then the new data retrieval function is given as

$$for\ j = 1, 2, \ldots, RN(Stream)$$

$$DRF_i'(j) = DRF_i(j) - D_{\mathrm{Reduce}}$$

Next, after obtaining the $P_{Critical}$ round, we determine the next starvation round, i.e., the next minimal value of the difference between data retrieval and data consumption. This round is called P_C. Repeat this process until *RN(Stream)*+1 round is reached. As shown in Fig. 9, the system may have many P_C rounds. Then in each P_C round, we compute the difference between the amount of data retrieval and data consumption. If this difference less than D_{Reduce}, then this round cannot be an idle round. Otherwise, the *idle round* technique is applied.

The *Idle Round* scheme described can reduce the buffer size at the client end. However, due to the logical zones consist of the same number of blocks, if some data blocks do not store data, this may lead to disk loading imbalance. Thus, the following constraint is added. If the *r*-th round is an idle round, it also satisfies Eq. (7).

$$BN^r(RSS_i(r)) \geq \left\lfloor \frac{\sum_{s=1}^{D_{Disk}} \sum_{t=1}^{LZ\#} BN^r(LZ_t^s)}{LZ\# \times D_{Disk}} \right\rfloor \qquad (7)$$

where r represents the r-th service round and $BN^r(z)$ represents the amount of data blocks in the z-th logical zone.

That is, for a given video stream, the amount of data blocks stored in the corresponding logical zone should be greater than the averaged number of data blocks. In summary, both Eq.(8) and Eq.(9) have to satisfied as the *r*-th service round to be an idle round. In this manner, client end buffer size is obtained without sacrificing disk loading balance.

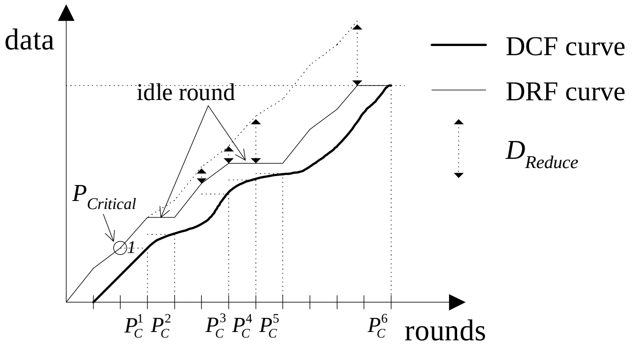


Fig.9 Determination of idle rounds by using the $P_{Critical}$

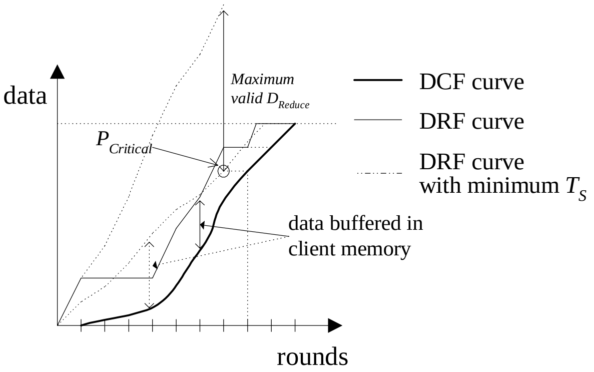


Fig.10 *Idle Round* scheme for low data consumption after $P_{Critical}$

(2) low data consumption rate before $P_{Critical}$

If low data consumption before $P_{Critical}$, due to the maximum value D_{Reduce} of is very small, the above scheme cannot apply straightforward. However, we can increase T_S, as a result of increasing D_{Reduce}. Then, as shown in Fig. 10, the scheme for low data consumption before $P_{Critical}$ is also applied. Note that the maximum value of T_S is bounded by the data access time for a logical zone μ or the startup latency for a given disk system.

Summarizing the above analysis, as shown in Fig. 11, we develop a *min_buffer* algorithm in the determination of buffer size at the client end.

```
min_buffer( )
    find all $P_C^1, P_C^2, \ldots$
    lookahead = 0
    service_time = min_service_time( )
    $B_{Client}$ = 0
    while lookahead < 5
        $DRF_i(1) = R(RSS_i(1)) \times$ service_time
        for j = 2 to RN(Stream)
        $DRF_i(j) = \min(S(Stream), DRF_i(j-1) + R(RSS_i(j)) \times$ service_time )
        end
        statisfy = 0
        $D_{Reduce}$ = 0
        t = 1
        $DRF_i'(1)$   $DRF_i(1)$
        for i = 2 to RN(Stream)
            if i > $P_C^t$
                t = t + 1
```

```
            end
            if $DRF_i(P_C^t) - D_{Reduce} - B(RSS_i(r)) \geq DCF(P_C^t)$
                statisfy = 1
            else
                statisfy = 0
            end
            if statisfy == 1 &&
                $BN^r(RSS_i(r)) \geq \left\lfloor \sum_{s=1}^{N_{Disk}} \sum_{t=1}^{LZ\#} BN^r(LZ_t^s) \Big/ (LZ\# \times N_{Disk}) \right\rfloor$
                $D_{Reduce} = D_{Reduce} + B(RSS_i(r))$
            end
            $DRF_i^{'}(r) = DRF_i(r) - D_{Reduce}$
        end
        for j = 1 to RN(Stream)
            $buffer\_size = \max\{DRF_i^{'}(j) - DCF(j+1)\}$
        end
        if $B_{Client}$ == 0 || buffer_size < $B_{Client}$
            $B_{Client}$ = buffer_size
            $DRF_i^{''} = DRF_i^{'}$
            $T_S$ = service_time
            look_ahead = 0
        else
            look_ahead = look_ahead + 1
        end
        service_time = service_time + 1
    end
    retuen($DRF_i^{''}$, $B_{Client}$, $T_S$)
end
```

Fig.11 Pseudo code of *min_buffer* algorithm

5. Simulation Results

In this section, we use the disk system as described in Section 2 to evaluate our design schemes. The disk system connects up to five Quantum XP34300W disks through Fast SCSI-2 bus. The parameters of Quantum XP34300W disk are given in Table 1. For comparison reason, we divided the each disk into several logical zones. The average transfer rates against different logical zones are plotted in Fig. 12. The average transfer rates distribute from 4.6875 MB/sec to 7.8516 MB/sec. As shown in Table 2, movies Volcano and Star Wars [18] are chosen as test video for CBR and VBR, respectively.

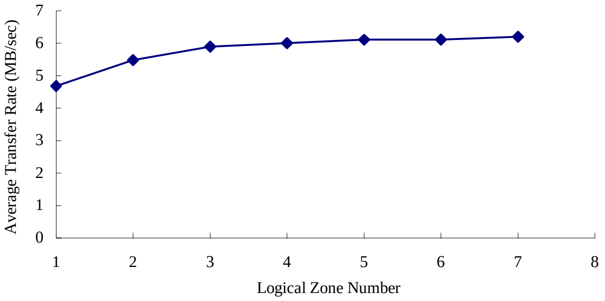


Fig.12 The average transfer rate of different logical zone numbers

Table.1 Parameters of Quantum XP34300W disk

Zone	Sectors per track	Cylinders per zone	Total sectors	Transfer rate(KBps)
1	134	208	557440	8040
2	131	262	686440	7860
3	128	248	634880	7680
4	125	304	760000	7500
5	123	176	432960	7380
6	120	232	556800	7200
7	116	248	575360	6960
8	113	232	524320	6780
9	107	528	1129920	6420
10	102	216	440640	6120
11	98	240	470400	5880
12	93	248	461280	5580
13	89	248	441440	5340
14	85	184	312800	5100
15	80	256	409600	4800

Table.2 Parameters of test video streams(1 round=10 sec)

Video Stream	STAR WARS	VOLCANO
Characteristics	VBR	Near CBR
Play time (sec)	7200	6000
Frames (f/sec)	24	24
Averaged data rate (Mbytes)	0.45	1.37
Maximal data rate (Mbytes)	1.01	1.40
Minimal data rate (Mbytes)	0.22	1.34

(1) Constant Bit Rate video stream

In the case of CBR, movie Volcano is chosen as test video. First we divide each disk into two, three, five, and seven logical zones. Then the *min_service_time* algorithm and the *min_buffer* algorithm to find the service time T_S and the buffer size at the client end B_{client}. As shown in Fig. 13, the service time decreases as the number of logical zones increases. In the mean while, buffer size is also decreased. Comparing the result with MVBA scheme [10], our design can support CBR video stream effectively.

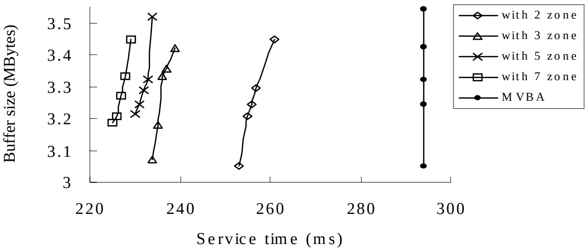


Fig.13 Experimental results for CBR video

(2) Variable Bit Rate video stream

In the case of VBR, movie Star Wars is chosen as test video. To illustrate the importance of the *idle round* technique, we compare the buffer sizes required at the client end for applying min_service_time algorithm only and applying both min_service_time and min_buffer algorithms. As shown in Fig. 14, in five logical zone cases, the *idle round* scheme can save more than 80% buffer size at the client end. The results for different logical zone case shown in Fig. 15. The relationship between service time and buffer sizes is plotted. These curves illustrate the tradeoffs between service time and buffer size at the client end. It is a good reference for designer. No matter which cases, up to a certain critical point, the increasing of service time will not results in the decreasing of buffer size. Again, comparing with MVBA scheme, our design is adequate for VoD system in the case of VBR.

To further illustrate the design step, we make use of a disk system with five disks as an example. Parts of data consumption function and data retrieval function are shown in Fig. 16 (a) and Fig. 16(b) for CBR and VBR respectively. These indicates the function of idle round in reducing the buffer size required at the client end, in particular for the case of VBR. Also, we show the data block distribution for each test streams in Fig. 17. We present the distribution by calculating the mean absolute deviation (MAD) of allocated blocks for each logical zone. As Shown in Fig. 17, the MADs are mostly below two blocks and this figure shows the disk load balancing is achieved for both CBR and VBR in our schemes.

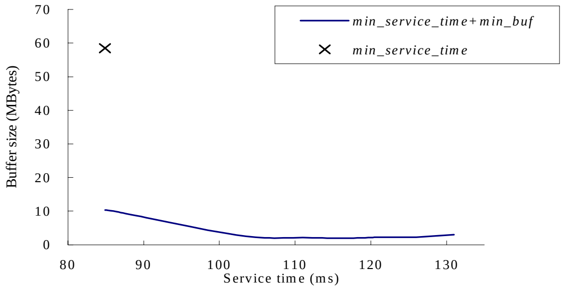


Fig.14 The buffer required by the *min_service_time* algorithm w/wo *min_buffer* algorithm when *LZ#*=5

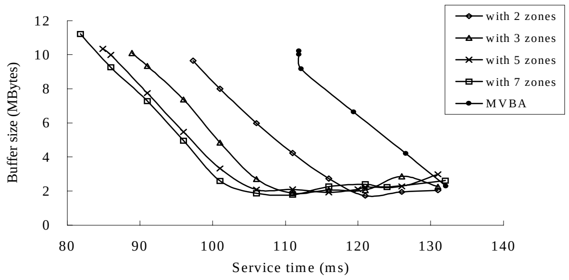
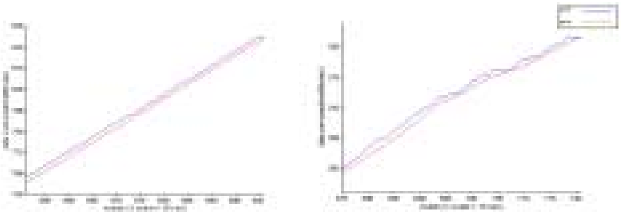
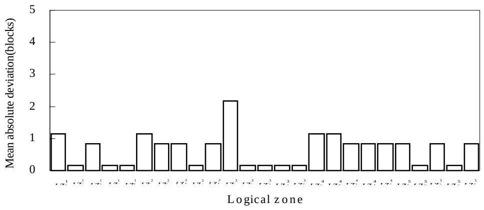


Fig.15 Experimental results for VBR

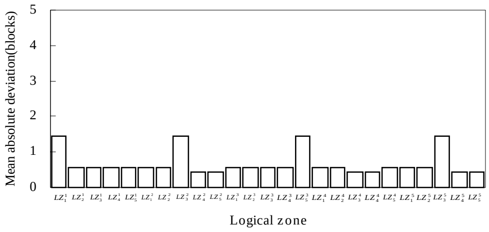


(a)Volcano (b)Star Wars

Fig.16 Relationship between DCF and DRF



(a) Star Wars(RSM=RSS_4,T_S=106ms)



(b)Volcano(RSM=RSS_1,T_S=230ms)

Fig.17 Mean absolute deviation of blocks allocated for logical zones

6. Implementation

Based on our proposed schemes, we have developed a video on demand system. It includes two subsystems: (1) Media server and (2) Media player. The whole system is now implemented on the Microsoft Win32 platform and its architecture is shown in fig. 18. The corresponding implementation details are described as follow:

(1) Media server

The server is designed and implemented for video storing, video retrieving, and network streaming. To achieve high efficiency, the *Preprocessing Unit* is developed to format the video sources according to RSM model and the buffer reducing algorithms before storage. After storage, the *Data Retrieving Unit* can easy perform admission control and video data retrieval according to RSM format. In addition, because the rate variability is reduced through *Preprocessing Unit*, the *Network Transfer Unit* can further create a QoS channel for network streaming through the Microsoft GQoS interface.

(2) Media player

The player is designed and implemented for video receiving and video playback. To simplify the implementation complexity and achieve future extension, as shown in fig. 18, we adopt the Microsoft DirectShow filter technique for video/audio decoding. In the media player, the Network Retrieving Filter is designed for receiving video data from network and forward these data to MPEG-1 decoding filters provided by Microsoft DirectX subsystem. Because new filters can be easy added to the DirectX subsystem, the player can support new video types by adding extra filters.

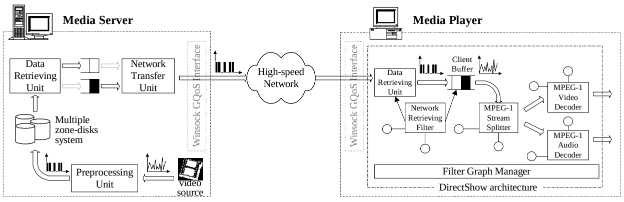


Fig.18 The architecture of system implementation

In addition, our developed system is adopted in the *Intelligent Web-based Interactive Language Learning* (IWiLL) system [19] and is named *Movie-based Conversation Showroom* (MCS). In the MCS system, English learners are able to learn, listen, and practice lift-like English conversation through networking English movie playback.

7. Conclusions

We propose a systematic design for a VoD system in the present paper. The design steps have two phases, (1) video placement, (2) video playout. In the continuous media placement stage, we developed a sequential data access model that based on CRT concept. The disk loading balance is achieved. So the number of simultaneous accesses increases. Furthermore, during video playout, the issues like admission control and playback interactivity can be manipulated effectively. Finally, by using the *Idle Round* technique, the buffer sizes required at the client ends can be further reduced. We perform experimental simulations to illustrate the adequate of our design.

As we have shown, our design considers the hardware characteristics of disk systems and video playout consumption. The system achieves its best performance when it stored the same type of video stream such as MPEG-1 VBR or MPEG-1 CBR. For systems that store different types of data streams, the same service time may not lead to the best performance. In addition, to support different bandwidth networks in Internet, the developments of video techniques tend to provide scalable video rates. To achieve high storing efficiency, the modified technique must be considered. We will investigate these issues in our future work.

REFERENCES

1. J. Gemmell, H. M. Vin, D. D. Kandler, P. V. Rangen, and L. A. Rowe, “Multimedia storage servers: A tutorial,” IEEE Computer, pp. 40-49, May 1995.
2. 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.
3. H. J. Lee and D. Du, “Cluster Placement : A Data Placement Scheme for A Mass Storage System of A Video-On-Demand Server,” Technical Report, UMN, 1995.
4. S. Ghandeharizadeh, J. Stone, and R. Zimmermann, “Techniques to Quantify SCSI-2 Disk Subsystem Specifications for Multimedia,” Technical Report, UCLA, 1995.
5. P. M. Chen, E. K. Lee, G. A. Gibon, R. H. Katz, and D. A. Patterson, “RAID: High-Performance, Reliable Secondary Storage,” ACM Computing Surveys, Vol. 26, No. 2, 1994.
6. G. Ganger et al., “Disk Subsystem Load Balancing: Disk striping vs. Conventional Data Placement,” in proc. of the 26th Hawaii International Conference on System Science, Vol. 1, IEEE CS press, pp. 40-49, 1993.
7. S. Chen ,and M. Thapar, "Zone-Bit-Recording-Enhanced Video Data Layout Strategies," in proc. of MASCOTS '96-4th International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 1996.
8. S. Ghandeharizadeh, S. H. Kim, C. Shahabi, and R. Zimmermann, “Placement of continuous media in multi-zone disks,” Book chapter 2 in Multimedia Information Storage and Management, Editor Soon M. Chung, Kluwer Academic Publishers, Boston, 1996, ISBN: 0-7923-9764-9.
9. W. Feng, and S. Sechrest, “ Smoothing and buffering for delivery of prerecorded compressed video,” in proc. of the ISET/SPIE Symposium on Multimedia Computing and Networking, pp. 234-242, 1995.
10. W. Feng, and S. Sechrest, “Critical bandwidth allocation for delievery of compressed video,” Computer Communications, Vol.18, pp. 709-717, 1995.
11. J. D. Salehi, Z.L. Zhang, J. F. Kurose, and D. Towsley, “Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing,” in proc. of ACM SIGMETRICS, pp. 222-231, 1996.
12. Yu-Chung Wang, Shiao-Li Tsao, Meng Chang Chen, Jan-Ming Ho and Ming-Tat Ko, "File Layout Design of VBR Video on Zoned-Disks", pp.49-62, Real-Time Database and Information Systems edited by Azer Bestavros and Victor Fay-Wolfe, Kluwer Academic Publishers, 1997.
13. Friedhelm Schmidt, The SCSI Bus and IDE Interface, second edition, Addison-Wesley Longman Limited, 1998, chapter 10, pp. 87-100.
14. Ed Chang and Avideh Zahkor, “Cost analyses for vbr video servers,” in proc. of SPIE Multimedia Computing and Networking, pp.381-397, 1996.
15. Chin-Hwa Kuo, Li-Chun Sung, and Meng Chang Chen, “ Design Issues in Multi-Zone Disks Video-on-Demand Systems,” has been accepted by the trans. of IEICE on Information and Systems.
16. S. Ghandeharizadeh, R. Zimmermann, W. Shi, R. Rejaie, D. Ierardi, and T. Li, “Mitra: A Scalable Continous Media Server,” Kluwer Multimedia Tools and Applications, 1997.
17. K. Sohraby, “On the theory of general ON-OFF source with applications in high-speed networks,” IEEE INFOCOM’93, pp. 401-410, 1993.
18. M. W. Garrett, W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," in proc. of ACM SigComm, London, Sept. 1994.
19. Intelligent Web-based Interactive Language Learning (IWiLL) system, http://163.13.127.201/iwill/