On Design of a Scalable Video Data Placement Strategy for Supporting a Load Balancing Video-on-demand Storage Server

By
Kelvin Kwok-wai Law

Supervised By :
Prof. John Chi-shing Lui

Submitted to the Division of Department of Computer Science & Engineering
in partial fulfillment of the requirements for the
degree of Master of Philosophy
at the
Chinese University of Hong Kong
June 1997

On Design of a Scalable Video Data Placement Strategy for Supporting a Load Balancing Video-on-demand Storage Server

submitted by

Kelvin Kwok-wai Law

for the degree of Master of Philosophy
at the Chinese University of Hong Kong

Abstract

Although advances have been made on video compression and high speed network technologies, designing a Video-on-demand (VOD) server is still a very challenging task. For one thing, the delivery of digital video must meet the real-time schedule; for another, digital video is large in volume and therefore demands high storage and transfer bandwidth. These two fundamental characteristics of digital videos make it difficult to design a load balanced VOD storage server that supports continuous video data retrieval for various video playback modes.

It has been recognized that VCR operations such as fast forward / rewind, pause and resume are important functionalities in providing VOD service. Most often, techniques proposed in solving VCR operations may require additional system resources such as extra disk I/O, additional buffer as well as the necessity of allocating extra network bandwidth for delivering the video object. In the dissertation, we propose a data blocks placement strategy for our VOD storage server. The use of subband video coding technique makes the placement strategy feasible for supporting various VCR functionalities and load balanced feature. In addition, we propose a disk scheduling scheme as well as an admission control policy so as to manage VOD server system resources efficiently. Finally, we determine the buffer requirement for delivering video object in our system.

The main contribution of the dissertation is that the proposed data blocks placement strategy, which strips multi-resolution video components across numbers of storage devices, allows the VOD storage server to support VCR functionalities easily. More importantly, no additional system resources such as I/O and network bandwidth are required to support VCR display. Our video storage server also maintains load balancing feature throughout the period of normal and VCR playback.

Acknowledgments

I would like to begin by thanking my parents, Kun-yuet Law and Bik-shan Tam, for the life and support they have given me. Without their nurture and care, I would not have been able to follow this path.

My research advisor, John Chi-shing Lui, has been my primary professional and intellectual influence during my time here at the Chinese University of Hong Kong. I am grateful to him not only for his ideas, but also for the mental rigor and the way of research which he taught me.

I would like to thank Leana Golubchik for her valuable advice during my research. My thanks and appreciations also go to Man-hon Wong.

I would like to thank my closest friends here at the Chinese University of Hong Kong who have taught me and given me support in their own ways: Sze-kin Lam, Steve Tai-man Wong, Alan Lung-sing Tung, Mary Yuk-yee Leung, Dilys Yin-hui Hung, Cedric Chiu-fai Fong, Chan-man Kuok, Chi-chiu Cheung and Peter Wai-kwok Lie.

Finally, I would also like to acknowledge my friends outside of the field who have continued to support me throughout the years: Sarah Yasson Chang, Hanford Hang-fung Chan, Ka-yue Chiu and Elsa Chui-ying Yan.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Advances on distributed multimedia system and networking technologies have made it feasible for providing interactive television services. Interactive television services allow geographically distributed users the flexibility to select and receive specific information through remote control units via the information superhighways. A well interactive service provider usually offers various services ranging from banking service, distant learning education, home shopping, electronic newspaper, financial transactions, game service for single-user and multi-user , Internet netvigation to Video-on-demand service (VOD)[1].

Video-on-demand service is considered as one of the premier applications in interactive services. Unlike traditional television broadcasting service in which subscribers have no controls during the program delivery sessions, VOD users have more control flexibilities. Users can tailor their view preferences such as viewing dimensions and quality of a video playback for any playback duration. In addition, the subscribers can control the video playback rate during the program delivery session. Besides, the on demand service saves the clients travel time to and from the video rental stores. Instead, they can select their favorite videos from the digital video library at any time and begin the playback service immediately. Therefore, providing VOD service is an attractive business in the entertainment market.

A number of commercial VOD system prototypes have been built in the recent years. Bell Atlantic is one of the first companies that have developed a commercial VOD ITV system [2]. Another up-coming commercial VOD system is being developed by Hong Kong Telecom and IBM [3].

1.2 Motivation

Designing a VOD server is a very challenging task due to the fundamental characteristics of digital video. For one thing, the delivery of digital video must meet the real-time schedule because video frames data convey meaning only when presented continuously in time. Another reason is that digital videos are generally large in volume and thus have demanding transfer bandwidth and storage requirement. Therefore, a storage system that supports continuous video data retrieval as well as the network subsystems that ensure timely delivery of video data are the critical components in the design of a VOD server.

It has been recognized that providing VCR operations is a major requirement in VOD service. However, implementing various VCR operations is not trivial, especially when the movies are coded based on the MPEG-1 standard. Under the normal video playback, the display bandwidth requirement of the MPEG-1 video is around 1.5 Mbps. To support a VCR fast forward or fast rewind function, which is several times faster than the normal display speed, the system might need to retrieve video data at a faster rate and therefore an additional I/O retrieval bandwidth is needed. This implies that during the period which VCR functions are employed, we are adding more workload to both the VOD storage server and the communication network. This additional workload adds complexity to the design of the VOD system as well as to the network management algorithm.

In designing a VOD system, it is a challenging task to design a system that can support the multi-resolution viewing configuration at the end-user and various VCR speed-up rates. Since digital video without scalable compression does not support multi-resolution nor multi-rate feature, the video server has to replicate the same set of videos with various resolutions and rates to the storage system. On receiving the demand for using a particular resolution or VCR rate, the corresponding video data are then retrieved from the storage system. It is clear that this implementation wastes a great deal of storage space and hence such a design is not cost-effective.

1.3 Scope

Based on the above consideration, this dissertation proposes a video data block layout scheme for addressing a number of issues. First of all, the development of the proposed file system is based on a scalable video compression technique – subband video coding. The appealing feature offered by the subband video coding is that the digitized compressed video has multi-resolution as well as multi-rate properties. These properties are very useful in providing the cost-effective VCR display service, in that the extraction of subsets of video stream makes the VCR functions demand no additional system resources such as I/O bandwidth on the VOD server and hence introduces no additional traffic into communication network.

The presented video data block layout scheme is based on the concept of striping multi-resolution components across numbers of cooperating storage devices. The layout strategy offers load balancing feature during the normal and VCR display period. The load balancing property is a very important feature because the VOD system will not have a situation in which a single disk is highly loaded while other disks are lightly loaded. Due to this load balancing property, the VOD system can accept more viewing users, and thereby the system becomes more cost-effective.

Finally, the dissertation deals with system resources management. System resources such as buffer and disk I/O are limited when the video server supports many simultaneous users. Based on the proposed video block layout scheme, an I/O scheduling scheme and an admission control policy have been developed. In addition, the memory buffer requirement for each user in various display modes is also evaluated in this dissertation.

1.4 Dissertation Outline

The organization of the dissertation is as follows.

In Chapter 2, we review the technology as well as the recent development of the VOD server. The chapter begins with the description of various interactive services provided by the interactive television providers. Then the discussion of the general architecture of VOD system and some popular video compression schemes are presented. At the end of the chapter, we briefly describe some related works of the VOD system.

In Chapter 3, we propose a data placement policy in supporting video that has a homogeneous multi-resolutions ratio [1]. The chapter begins with the description of disk model on which our proposed VOD file system is based. Then an example is given to illustrate how the assignment works for a video file which supports four resolutions. At the end of the chapter, a general framework on the assignment of video files with n resolutions support is presented.

In Chapter 4, we explore the issues of disk scheduling and admission control policy for the proposed homogeneous multi-resolutions file system. The chapter begins with the presentation of the proposed disk scheduling scheme – cyclical scheduling policy. Then we illustrate how a new video stream for the newly admitted client is scheduled in the existing data retrieval schedule. Finally, the

[1]The formal definition of video file supporting multi-resolution ratio is given in Chapter 3

conditions as well as the policy in deciding the admission of a new client is discussed.

In Chapter 5, we show the proof of the load balancing property of the presented multi-resolution file system.

In Chapter 6, we deal with the buffer arrangement for various display modes, normal and VCR display. We begin with the description of buffer organization for the VOD system. Then we present some examples for the interaction between the disk array system and the dual buffer system for providing smooth video playback services during various display modes. Lastly, we evaluate the buffer requirement for normal and VCR display in the VOD system.

Finally, Chapter 7 concludes the dissertation with a summary of the presented work.

Chapter 2

Background and Related Researches

In this chapter, we describe some background works on the VOD system. We begin with a review on different VOD interactive services provided by the information service providers in Section 2.1. We present a general architecture for VOD system in which we illustrate the functions of the video server, high speed network services and set-top box in Section 2.2. In Section 2.3, we list various popular video compression technologies that can be applied to VOD system. Finally, a few related research works are briefly discussed in Section 2.4.

2.1 Interactive Services

According to the amount of interactivity of which a customer can control, interactive television services can be categorized into several aspects [1, 4]

- **Broadcast services**
  The services are similar to the traditional television broadcasting services

in which the users have nearly no control on over the service time.

- **Pay-Per-View services**
  The services are quite similar to the existing cable television services. A clients signs a contract with the cable television services provider and pays for each specific program. In this type of service, users have limited control over the viewing time and no control of VCR functions.

- **Quasi Video-On-Demand services**
  In this type of services, a number of users with the same interest of a video are batched into a group, the interactive television provider starts a video session whenever the number of users being batched at that time fulfills the threshold value. During the video playback session, users are allowed to have limited playback controls by switching to a different group.

- **Near Video-On-Demand services**
  Users who are subscribed to this services have greater flexibility in playback controls such as forward and rewind than those who are subsribed to the services provided by Quasi VOD services. The forward and rewind functions are simulated by transmitting video in discrete time units.

- **True Video-On-Demand services**
  Users have full-function of VCR controls over the playback sessions. That is to say that they can issue the fast forward / rewind play, pause and random positioning of a video at any time during the video playback.

2.2 VOD Architecture

A general architecture of a VOD system is shown in Figure 2.1. The VOD system roughly consists of three main components : set-top box, high speed network and video server.

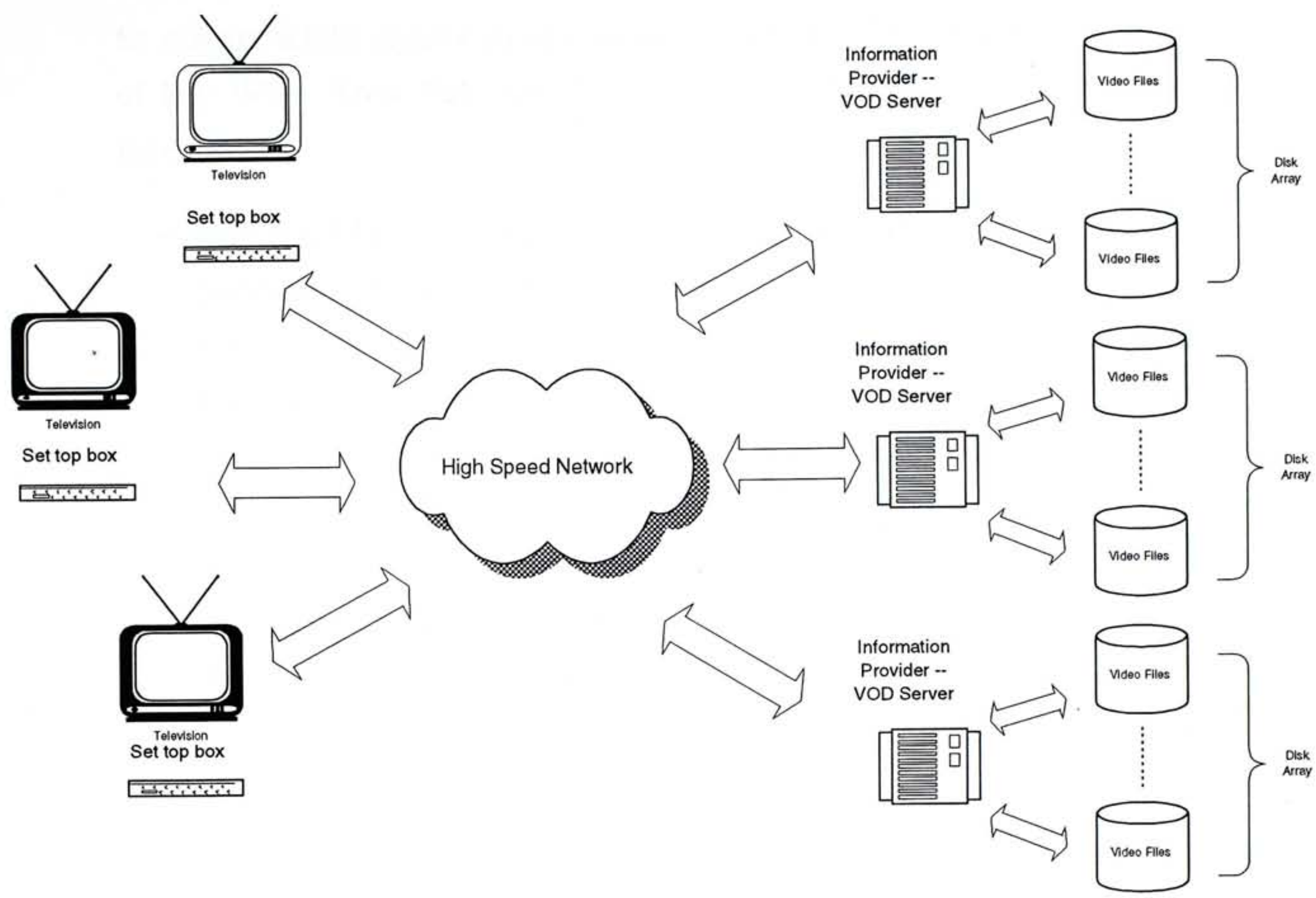


Figure 2.1: A general architecture of VOD system

- Set-top box

  From the clients' perspective, the set-top box is a front-end device which bridges the communication channel between the interactive television providers and the users. It receives and processes signal from both the video servers and the end clients. Through the set-top box, the compressed / encrypted video frames sent by the video server can be decompressed / decrypted, the digital information is then presented on the television. In addition, clients are able to tailor their viewing preferences through the set-top box during the video playback period. By using remote control unit, clients send control signal such as VCR commands to the set-top box. The input signal is processed and then sent to the video server via the high speed network.

- High speed network

  The network system sets up a path for the communication between a video server and the clients of the server. Video server transfers video information to clients while clients send control signals to the video server. A number of key Wide Area Network (WAN) services for multimedia [5] are briefly introduced:

– Switched Multimegabit Data Services (SMDS)
  Switched Multimegabit Data Service is a connectionless high-speed, high quality packet service provided by the Bell Operating Companies (BOCs) and by other carriers. It enables accesses at DS1 and DS3 speeds and supports an end-to-end throughput close to the transmission speed. However , the users are required to perform their own multiplex data for multimedia applications. SMDS has become commercially available in 1991 and a number of vendors including Cisco and ACC announced equipments.

– Frame Relay Service (FRS)
  Frame relay service is a connection-oriented service operating at 1.544 Mbps (2.048 Mbps in Europe). It is a multiplex service supporting connectivity between user devices (such as routers) as well as the connectivity between users and a public network. The service is designed to reduce network delays, to provide more-efficient bandwidth utilization and to decrease communication equipment cost. Currently, FRS is available in approximately forty U.S. cities and in some major European cities.

– Dedicated-Line Services
  Dedicated-Line Services is generally leased by companies or national enterprises from the local / international telecommunication service providers. Since the line is not generally shared with others, there is no delay variation and no frame discard over the WAN. Hence, the users can utilize the line with its maximum throughput 1.544 Mbps (some also at 45 Mbps).

– Asymmetric Digital Subscriber Lines (ADSL)
  The ADSL technology enables the exiting ISDN twisted copper-pair to deliver T1 service of throughput 1.544 Mbps from network to user direction, and 160Kbps low speed information from user-to-network direction. Therefore, ADSL is suitable for delivering VCR-quality

video or gaining access to remote libraries of CD-ROM material over existing network.

- Video server

  The video server is the core device that provides interactive services. It processes customers' requests, sets up and maintains the video stream according to the requests. The video server usually consists of a number of computing machines or processing units with interfaces for communicating with groups of tertiary storage devices and network subsystems. The computing units require fast processing speed, large memory space to handle and to process the requests from a large number of simultaneous users. In addition, the enormous system bus bandwidth enables the system to handle input and output traffic from storage system as well as network interface. The network interface receives the clients' requests and directs the signal to the computing unit for further processing. Through the same interface, video information retrieved from the tertiary storage and processed by the computing units is sent to clients via the Internet.

2.3 Video Compression

It is important to note that video data differ from numeric and textual data in their characteristics. The volume of video files is very large and consume a great deal of storage space. A video frame, typically digitized at a resolution of 512 x 480 pixels with a color depth of 24 bits per pixel, will result in 780KB of information per frame. A video of length 90 minutes and frame rate 30 frames per second occupies 111GB ! Moreover, a video, which is a sequence of media quanta such as video frames, is meaningful to human only when presented continuously in time. Therefore, a VOD server has to ensure that each of this media quanta must be retrieved from disk storage at the specified real-time playback rate. The conventional slow storage devices do not allow raw video data playback to meet the specified real-time playback. Furthermore, it is nearly impossible to transmit raw video data via the current network in real time. In order to cope with this problem, a considerable amount of research has been conducted on the compression schemes with the result that raw data bits can be dramatically reduced in the order of 100 to 200.

There are a multitude of video compression techniques developed in the past decade. Two of the principle compression schemes are discrete cosine transformation (DCT) based compression technique and subband compression technique.

2.3.1 DCT Based Compression

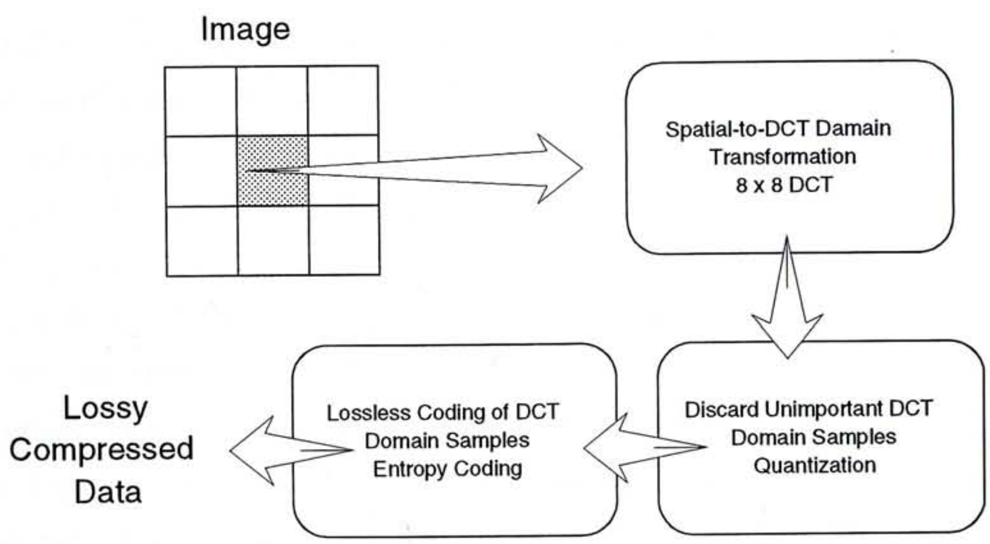


Figure 2.2: DCT based coding system

In a DCT based coding scheme (Figure 2.2), the DCT process transforms each 8×8 block of source image into 64 DCT coefficients. Each of these coefficients is the weight associated with corresponding DCT basis waveform. Some of the weights will be removed in the lossy compression mode and hence the corresponding waveforms will not be used during the decompression process. This process is referred to quantization. After quantization, the 8×8 block is then entropy encoded. H.261, JPEG, MPEG-1 and MPEG-2 are typical video compression standards using DCT based compression techniques. A detailed description of H.261, JPEG, MPEG-1 and MPEG-2 is out of the scope of this dissertation and therefore we direct the readers to [6, 7, 8, 9] for further discussion.

2.3.2 Subband Video Compression

The idea of subband coding was originally introduced by Crochiere et al. [10] for the encoding of speech. The coding technique is called 1-D subband coding because frequency bandwidth is split along with single dimension frequency domain. Woods et al [11] extended the 1-D subband coding concept to two dimensional and applied it to the image coding which is called 2-D subband coding. However, the 2-D subband coding alone cannot efficiently compressed images with temporal redundancy and hence leads to the development of 3-D subband coding [12, 13, 14].

Subband coding technique is a class of scalable video compression algorithm capable of producing multiple resolution of video bit stream [15]. This type of coding technique is based on multi-stage algorithms, where each stage uses the Quadrature Mirror Filtering (QMF) technique to separate the input streams into two frequency spectrums, one for the high band and the other for the low band frequency spectrum. Spatial compression is achieved by applying the video signals to the two stages of the algorithms, one for the x direction and the other for the y direction. Temporal compression is achieved by filtering and downsampling successive frames. By cascading several stages of filtering and downsampling, further compression of the video signal can be achieved [16, 17]. For example, cascading ten stages will produce 40 outputs, each corresponding to a different frequency spectrums. By combining these outputs into groups, various degree of spatial and temporal resolutions of the original video can be reconstructed. A full resolution of the original video stream can be constructed by combining all outputs together.

An illustration of how subband coding technique accepts the input signal is as follows: This original signal is first being downsampled into a set of streams, each of which is limited to a range of spatial frequencies. The filtered subband streams are further encoded by one or more coders for optimization. Reconstruction is achieved by first decoding the incoming encoded streams, the decoded subband streams are then added together to form an image. Figure 2.3 shows an example of

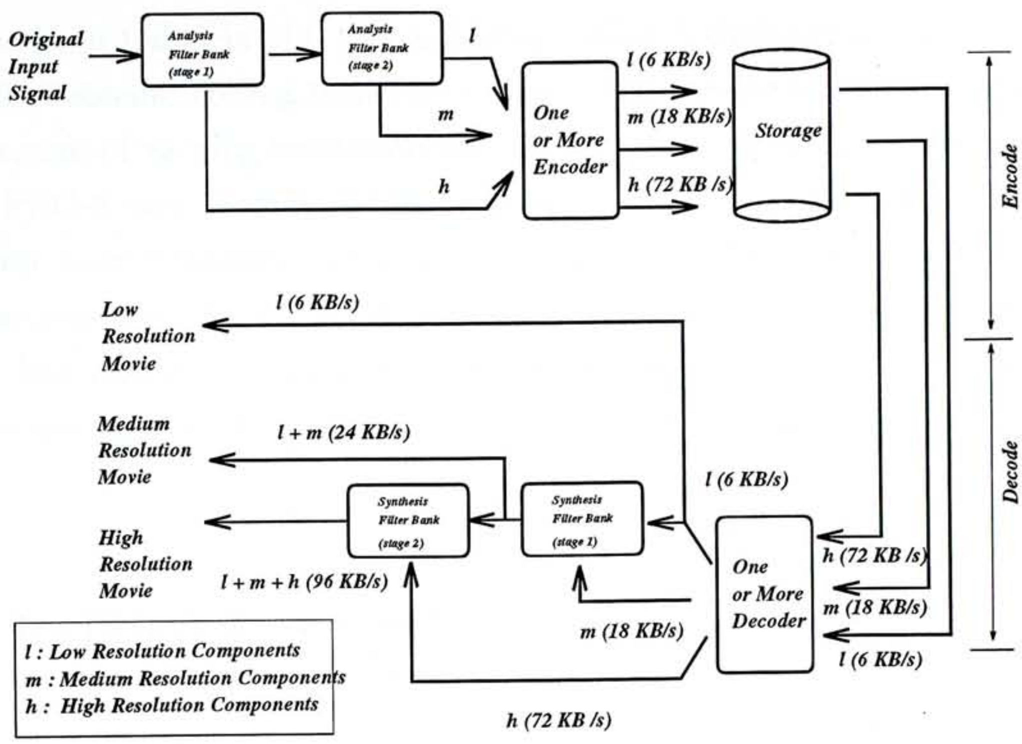


Figure 2.3: Typical subband coding example

subband compression with two cascading filters scheme. On the encoding side, the original video signal is decomposed by analysis filters, in which quadrature mirror filters (QMF) are implemented for alias free reconstruction, into three subband streams. After passing through the first stage of analysis filter, high resolution components (h) are filtered out. Medium resolution components (m) and low resolution components (l) are obtained by passing through the second analysis filter. These downsampled streams are coded by possibly multiple coders before being stored in the storage server. In this example, the throughput of the encoded high, median and low resolution components are 72 KB/s, 18 KB/s and 6 KB/s respectively. During data retrieval, the encoded streams are retrieved from the storage server and are decoded by the corresponding decoders. The low resolution video (l), which has a throughput of 6 KB/s, is obtained by simply using the decoded low resolution components. By adding the decoded low resolution and medium resolution components, the medium resolution video stream ($l+m$) now has a throughput of 24 KB/s. The high resolution video stream ($l + m + h$)is obtained by combining the decoded low, medium and high resolution components and this high resolution video stream has a throughput of 96 KB/s.

It has been found in [18] that subband coding techniques are comparable to MPEG-2. Subband coding techniques offer a high degree of scalability, allowing video streams of varying resolution and bit rate to be constructed, far more than what MPEG-2 can provide. The techniques can be used to decompose high bandwidth source images , such as video frames, which requires a great deal of storage capacity, into a great number of subbands. Each of these subbands requires less resources such as storage and network bandwidth. Therefore, we intend to use these techniques in our proposed file system.

2.4 Related Research

There are various research results on supporting interactive VCR functions for the VOD system. For example, authors in [19] proposed two methods, known as the segment sampling and segment placement, in order to support variable bit rate browsing for MPEG-like video streams. In segment sampling, one out of m segments (a retrieval unit) is selected for display when a fast forward (or fast rewind) with viewing is desired and the fast forwarding speed is m times the normal display speed. In contrast, segment placement scheme allocates segments to disks judiciously such that a completely uniform segment sampling across the disk array occurs. The drawback of this approaches is that the quality of display during these VCR functions is not acceptable. For example, viewers will experience jitterness and large jump in video frames. In [17, 20], authors discussed the implementation of I/O scheduling for VOD systems. They pointed out that if I/O scheduling is not carefully designed, starvation of data blocks might occur. Two algorithms were proposed and the main idea was to have a safe state transition (from normal display to VCR function) and to avoid starvation. It should be noted that, under this scheme, an additional buffer resource is required and the communication network will experience a sudden surge of traffic; consequently, data blocks might be dropped by the communication network. In [21], authors proposed using several buffer management techniques for providing constant viewing intervals as well as a limited form of VCR functions for viewers. As long as the data blocks were pre-fetched and are available in the memory buffer, no additional I/O resource is needed. However, a large amount of memory buffers are necessary to implement this scheme. At the same time, traffic workload fluctuation will occur in the communication network. Other works based on exploiting the characteristic of MPEG coding technique were presented in [22, 23] but all these require additional I/O resources, such as buffer memory as well as the potential of introducing traffic workload fluctuation into the network.

Recently, the subband coding techniques [16, 17, 15] were proposed in implementing a VOD system that can support multiple video resolutions. In [24], authors studied two video layout strategies on the disk arrays in order to achieve different degrees of parallelism and concurrency.

Chapter 3

Multiple Resolutions Video File System

In this chapter, we propose a data blocks placement strategy for multi-resolution video files. In section 3.1, we outlines the physical disk storage system and the disk parameters used in the later chapters. Some of the terminologies appeared on the proposed multi-resolution data blocks placement strategy are introduced in section 3.2. In order to give a clearer concept on how the data blocks placement strategy works, we use an example in section 3.5 to illustrate the assignment of a video file supporting four resolutions. Finally, the framework of assigning video files with n resolutions is presented in section 3.4.

3.1 Physical Disk Storage System

It is commonly agreed that video files are large in size and require a large amount of disk storage space, and thereby a single disk drive is unable to hold hundreds of videos of which a commercial video server is required. In addition, the bandwidth requirement for normal and VCR video playback makes a single disk become the major limitation in offering multiple simultaneous users playback service. The

RAID (Redundant Arrays of Inexpensive Disks) [25] is a feasible solution to the storage problem.

Using RAID has an advantage over a single disk. Storage space is an obvious one. Redundant disk array is composed of a number of independent disks, the disk capacity is, therefore, increased by the number of disks in the array when RAID level 0 is used. Besides, RAID allows data striping which transparently distributes data over multiple disks. This implies that a video stream can be served by multiple I/Os in the RAID. As a result, the aggregate I/O performance can be enhanced.

The RAID system is an array of d homogeneous disk. That means the characteristics of each disk, in terms of seek time, rotational latency and transfer latency, are the same. We further assume that the maximum bandwidth of each disk is B^{max} bytes / second, $T^{max}_{seek}(\mathcal{U})$, $T^{max}_{rotational}(\mathcal{U})$ and $T^{max}_{transfer}(\mathcal{U})$ are the the maximum seek time, rotational latency and transfer latency of $\mathcal{U}$ bytes of data from the disk respectively. Therefore, we can define the total disk latency of retrieving $\mathcal{U}$ bytes of data as :

$$T^{max}(\mathcal{U}) = T^{max}_{seek}(\mathcal{U}) + T^{max}_{rotational}(\mathcal{U}) + T^{max}_{transfer}(\mathcal{U}) \tag{3.1}$$

3.2 Multi-resolution Video Data Placement Scheme

Suppose the VOD system can support $n > 0$ resolutions of display and any given video file in the system is labelled as $\mathcal{V}$ having g segments, $\mathcal{V}$ can be expressed as :

$$\mathcal{V} \ = \ \{\mathcal{V}_0 \cup \mathcal{V}_1 \cup \cdots \mathcal{V}_i \cdots, \cup \mathcal{V}_{g-1}\}$$

where $\mathcal{V}_i$ is the i^{th} segment in video $\mathcal{V}$ and $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. Each segment has up to n resolutions of data:

$$\mathcal{V}_i \ = \ \{\mathcal{V}_i^0 \cup \mathcal{V}_i^1 \cup \cdots \cup \mathcal{V}_i^{n-1}\}$$

where $\mathcal{V}_i^j$ represents the additional data blocks of the j^{th} resolution to the $(j-1)^{th}$ resolution for the i^{th} segment, and we call these additional data blocks as the j^{th} *enhancement.* The enhancement $\mathcal{V}_i^{j+1}$ contains data blocks required to form a higher display resolution than the enhancement $\mathcal{V}_i^j$ for $0 \le j < n-1$. Each $\mathcal{V}_i^j$ is composed of b_j disk blocks[1] such that:

$$\mathcal{V}_i^j = \{r_{i,0}^j, r_{i,1}^j, \ldots, r_{i,b_j-1}^j\}$$

where $r_{i,k}^j$ is the k^{th} disk blocks for $\mathcal{V}_i^j$. Figure 3.1 illustrates the data block layout of the first 8 segments of a video file. Based on our defined notation, we have:

$$\begin{aligned}
\mathcal{V}_0 &= \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1 \cup \mathcal{V}_0^2 \cup \mathcal{V}_0^3\} \\
\mathcal{V}_0^0 &= \{r_{0,0}^0\} \ ; \ b_0 = 1 \\
\mathcal{V}_0^1 &= \{r_{0,0}^1\} \ ; \ b_1 = 1 \\
\mathcal{V}_0^2 &= \{r_{0,0}^2, r_{0,1}^2\} \ ; \ b_2 = 2 \\
\mathcal{V}_0^3 &= \{r_{0,0}^3, r_{0,1}^3, r_{0,2}^3, r_{0,3}^3\} \ ; \ b_3 = 4
\end{aligned}$$

All enhancement $\mathcal{V}_i^j$, $\forall 0 < i \le n-1$, by itself cannot form a complete segment

Segment / Disk	***0***	***1***	***2***	***3***	***4***	***5***	***6***	***7***
0	$r^0_{0,0}$	$r^1_{0,0}$	$r^2_{0,0}$	$r^2_{0,1}$	$r^3_{0,0}$	$r^3_{0,1}$	$r^3_{0,2}$	$r^3_{0,3}$
1	$r^3_{1,0}$	$r^3_{1,1}$	$r^3_{1,2}$	$r^3_{1,3}$	$r^0_{1,0}$	$r^1_{1,0}$	$r^2_{1,0}$	$r^2_{1,1}$
2	$r^2_{2,0}$	$r^2_{2,1}$	$r^0_{2,0}$	$r^1_{2,0}$	$r^3_{2,0}$	$r^3_{2,1}$	$r^3_{2,2}$	$r^3_{2,3}$
3	$r^3_{3,0}$	$r^3_{3,1}$	$r^3_{3,2}$	$r^3_{3,3}$	$r^2_{3,0}$	$r^2_{3,1}$	$r^0_{3,0}$	$r^1_{3,0}$
4	$r^1_{4,0}$	$r^0_{4,0}$	$r^2_{4,0}$	$r^2_{4,1}$	$r^3_{4,0}$	$r^3_{4,1}$	$r^3_{4,2}$	$r^3_{4,3}$
5	$r^3_{5,0}$	$r^3_{5,1}$	$r^3_{5,2}$	$r^3_{5,3}$	$r^1_{5,0}$	$r^0_{5,0}$	$r^2_{5,0}$	$r^2_{5,1}$
6	$r^2_{6,0}$	$r^2_{6,1}$	$r^1_{6,0}$	$r^0_{6,0}$	$r^3_{6,0}$	$r^3_{6,1}$	$r^3_{6,2}$	$r^3_{6,3}$
7	$r^3_{7,0}$	$r^3_{7,1}$	$r^3_{7,2}$	$r^3_{7,3}$	$r^2_{7,0}$	$r^2_{7,1}$	$r^1_{7,0}$	$r^0_{7,0}$

Figure 3.1: An example of assigning video data blocks into disk array

to be display to the clients. In other words, $\mathcal{V}_i^j$ is only a portion of the whole

[1]The data block size is chosen such that the overhead due to seek and rotational latency is not more than 10% of the overall disk response time.

displayable segment. We define $\mathcal{V}_i^0$ as the *base resolution* at segment i. Any display segment of resolution $j > 0$ consists of video blocks of base resolution as well as video blocks from resolution 1 to j. We denote the set of *display resolution* j at segment i as :

$$\mathcal{R}_i^j = \bigcup_{k=0}^{j} \mathcal{V}_i^k$$

The following is an example from Figure 3.1, we have

$$\begin{aligned}
\mathcal{R}_0^3 &= \mathcal{V}_0 = \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1 \cup \mathcal{V}_0^2 \cup \mathcal{V}_0^3\} \\
\mathcal{R}_0^2 &= \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1 \cup \mathcal{V}_0^2\} \\
\mathcal{R}_0^1 &= \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1\} \\
\mathcal{R}_0^0 &= \{\mathcal{V}_0^0\}
\end{aligned}$$

Clearly, the number of disk blocks for any segment of display resolution j is equal to $\sum_{k=0}^{j} b_k$. In addition, we let $disk(r_{i,k}^j)$ denote the disk number, ranging from 0 to $d-1$, that has been allocated for storing the disk block r_{ik}^j. We also define $DISK(\mathcal{R}_i^j)$ to be the disk set for storing $\mathcal{R}_i^j$. Using the example in Figure 3.1 , we have $disk(r_{0,0}^1) = 1$ and $DISK(\mathcal{R}_0^2) = \{0, 1, 2, 3\}$.

Definition 1 $\mathcal{R}_i^j$ *and* $\mathcal{R}_k^l$ *are non-overlapping iff* $DISK(\mathcal{R}_i^j) \cap DISK(\mathcal{R}_k^l) = \emptyset$.

Referring to the example in Figure 3.1, $\mathcal{R}_0^1$, $\mathcal{R}_1^1$, $\mathcal{R}_2^1$ and $\mathcal{R}_3^1$ are non-overlapping because $DISK(\mathcal{R}_0^1) = \{0,1\}$, $DISK(\mathcal{R}_1^1) = \{4,5\}$, $DISK(\mathcal{R}_2^1) = \{2,3\}$ and $DISK(\mathcal{R}_3^1) = \{6,7\}$.

Definition 2 *The normal display of a video file with respect to a user is defined as the display resolution* r*, where* $0 \leq r \leq n-1$*, set by that user upon his admission into the VOD system.*

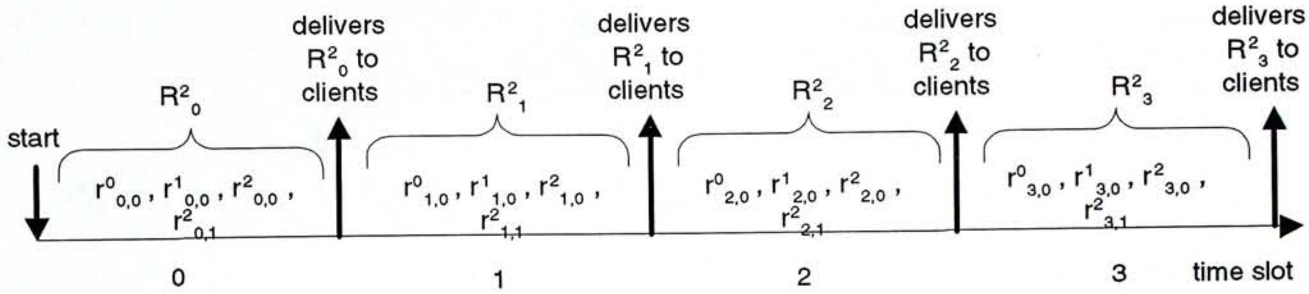


Figure 3.2: Normal display of display resolution 2

Upon admission , the user can set his normal display to be at any resolution r where $0 \leq r \leq n-1$. Under this circumstance, the normal display of the i^{th} segment of a video file requires the VOD system to retrieve all disk blocks in $\mathcal{R}_i^r$. Referring to the example in Figure 3.1, a user set the normal display rate at $r = 2$ upon his admission into the system, We also assume that he starts viewing the video at segment 0. Figure 3.2 illustrates the data block retrieval and delivery process. In this figure, the horizontal line represents the time slot, and the vertical arrow marks the time instance when the server has finished retrieving a segment of video of display resolution 2. The time interval between any two vertical arrows is called a time slot. Any video blocks of a segment must be retrieved within the time slot.

In this example, the server retrieves data blocks of display resolution 2 , $\mathcal{R}_i^2$, from the parallel disk system in every time slot. To view the video at the normal display, the system needs to retrieve data blocks $\mathcal{R}_i^2$ at time slot i, $\forall i \geq 0$. The data blocks retrieved are first put into the system buffer before delivery, Such buffer management will be discussed later in the dissertation. The viewer can request any VCR functions such as fast forward or fast rewind, during the normal display viewing period. This can be accomplished by retrieving more segments per time slot, with these segments having a lower display resolution than the normal display resolution r.

Definition 3 *The fast forward or fast rewind VCR function supported by the system made feasible by retrieving multiple consecutive segments of the video file of resolution r' (where $0 \leq r' < r$) to the end user.*

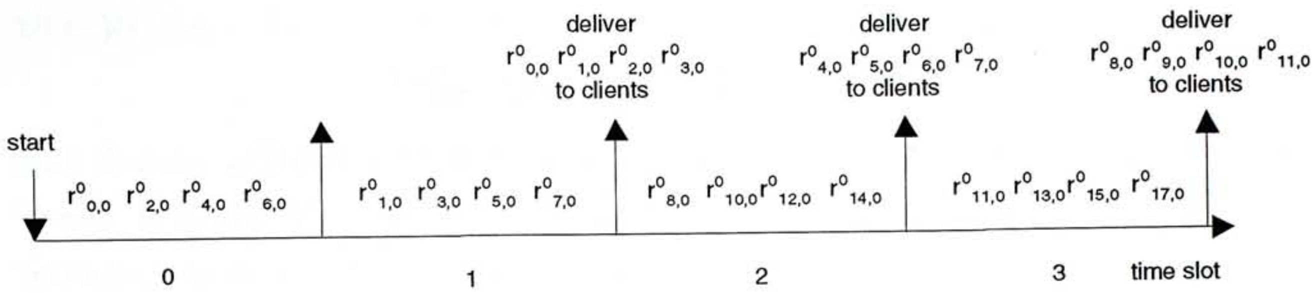


Figure 3.3: VCR display of display resolution 0

In general, if the normal display is set at display resolution r, a user has the option of up to $r-1$ different VCR viewing speeds, which can be expressed as:

$$VCR\_Speed(r,r') = \left\lfloor \frac{\sum_{j=0}^{r} b_j}{\sum_{j=0}^{r'} b_j} \right\rfloor \qquad \text{where } 0 \leq r' < r \leq n-1 \quad (3.2)$$

According to the user's selection of normal display resolution, he can only select VCR speeds that can be expressed by Equation 3.2. In other words, he can choose up to r' resolutions (where $0 \leq r' < r \leq n-1$) for different VCR speeds. The number of disk blocks for display resolution r' is $\sum_{i=0}^{r'} b_i$ which is smaller than the number of blocks of display resolution r, $\sum_{i=0}^{r} b_i$. Therefore, the VOD system can retrieve more segments per unit time and thereby obtains a faster viewing speed. It is important to note that the VCR speed-up introduces a constant traffic workload to the communication network.

To illustrate, consider the example in Figure 3.1. If upon admission, the user requests for normal display resolution $r = 3$, then in order to view segment $\mathcal{V}_0$, the system needs to retrieve disk blocks in $\mathcal{R}_0^3 = \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1 \cup \mathcal{V}_0^2 \cup \mathcal{V}_0^3\}$ or:

$$\bigcup_{l=0}^{3} \bigcup_{j=0}^{b_l} r_{0,j}^l$$

If VCR function is desired, the user can choose the following VCR speeds: $VCR\_Speed(3,2) = 8/4 = 2$, $VCR\_Speed = (3,1) = 8/2 = 4$ and $VCR\_Speed(3,0)$ $= 8/1 = 8$. Suppose the viewer chooses the VCR speed that is two times faster than the normal display; he actually makes a subscription of display resolution $r' = 2$. The system can retrieve disk blocks in $\mathcal{R}_0^2 \cup \mathcal{R}_1^2 = \{\mathcal{V}_0^0 \cup \mathcal{V}_0^1 \cup \mathcal{V}_0^2\} \cup \{\mathcal{V}_1^0 \cup$

$\mathcal{V}_1^1 \cup \mathcal{V}_1^2\}$ or:

$$\{\cup_{l=0}^{2} \cup_{j=0}^{b_l} r_{0,j}^{l}\} \bigcup \{\cup_{l=0}^{2} \cup_{j=0}^{b_l} r_{1,j}^{l}\}$$

and thereby achieve a VCR viewing speed which is twice the normal display speed. It is important to note that during the VCR period, the system does not introduce more workload traffic into the network and therefore the design of the buffer and network management algorithms can be simplified.

Apart from offering normal display and VCR functionality, the multi-resolution file system achieves the load balancing property during the normal and VCR display periods.

Definition 4 *We define the retrieval of a display resolution r as* load balanced *over any $\mathcal{P}$ consecutive segments if every disk in the parallel disk system has the same amount of transfer load. The video data blocks retrieved are to satisfy the following conditions :*

1. $\cup_{i=k}^{\mathcal{P}+k-1} DISK(\mathcal{R}_i^r) = \{0, 1, \ldots d\}$, *and*
2. $\cap_{i=k}^{\mathcal{P}+k-1} DISK(\mathcal{R}_i^r) = \emptyset$

where k is any segment number in the video.

We use the example in Figure 3.1 to illustrate the idea. If $r = 3$, the VOD system is load balanced for every segment retrieved. However, if a user starts a VCR function with $r' = 1$ at segment 0, then the VOD system needs to retrieve data blocks $\{r_{0,0}^0, r_{0,0}^1\}$, $\{r_{1,0}^0, r_{1,0}^1\}$, $\{r_{2,0}^0, r_{2,0}^1\}$, $\{r_{3,0}^0, r_{3,0}^1\}$. Since these four segments are non-overlapping, they can be retrieved in parallel and a VCR speed which is twice the speed for the normal display is archieved. Using the example in Figure 3.1, no matter what the display resolution r' is at the request of VCR, the disks remain balanced after retrieval 8 consecutive segments.

3.3 Example of our Video Block Assignment Algorithm

We use the example in Figure 3.1 to us illustrate our video block assignment algorithm. For the normal display, since the retrieval size of each display resolution is not the same, every segment retrieval adds a different workload to different disks. If upon admission, the viewer chooses display resolution $r = 3$, then each segment retrieval requires reading one data block from each disk, the load balancing feature is therefore maintained after each segment retrieval. On the other hand, if $r = 2$, then retrieval of segment $\mathcal{V}_0^2$ requires reading one data block from disk 0 to 3 and the retrieval of segment $\mathcal{V}_1^2$ requires reading one data blocks from disk 4 to 7. , the load balancing property is maintained after reading two consecutive segments. Similar argument can be observed for other values of r. In general, the load balancing property is maintained after reading $P_r = \frac{d}{\sum_{i=0}^{r} b_i}$ consecutive segments for $0 \leq r < n$. Given the example illustrated in Figure 3.1, we can observe that no matter what display resolution r the user selected upon his admission, all disks should retrieve the same number of disk blocks after reading 8 consecutive segments.

Before we present the data blocks assignment algorithm, let us illustrate how we perform the data blocks assigned for the 4−resolution VOD system which we have in Figure 3.1. Later, we can generalize the design principle to n−resolution VOD system. The assumption we make for the underlying video file system is:

$$\sum_{i=0}^{r} b_i \bmod \sum_{i=0}^{r-1} b_i = 0 \qquad 1 \leq i \leq n-1 \tag{3.3}$$

, assuming that we start with $d = \sum_{i=0}^{n-1} b_i = 8$ disks and we try to assign disk blocks starting from the highest resolution first.

resolution $= 3$: Since the total number of data blocks in $\mathcal{R}_i^3$, for any i^{th} segment is equal to $\sum_{i=0}^{3} b_i = 8$, we have to use up all d disks. Although we know that we have to assign every ith segment of $\mathcal{R}_i^3$ to all disks, we cannot determine which disk blocks in $\mathcal{R}_i^3$ is assigned to which disk until disk blocks with lower resolution have been assigned.

resolution $= 2$: To assign data blocks in $\mathcal{R}_i^2$, we have to ensure that if user selects $r = 3$ upon his admission, he can choose $r' = 2$ for his VCR speed later. To provide this particular VCR service, the VOD system must be able to retrieve $VCR\_Speed(3, 2) = 8/4 = 2$ consecutive segments of display resolution r' in parallel. This implies that two consecutive segments of display resolution 2, $\mathcal{R}_i^2$ and $\mathcal{R}_{i+1}^2$, must be *non-overlapping.* One way to accomplish this is to assign the i^{th} segment to disk $0, 1, 2, 3$ and the $(i+1)^{th}$ segment to disk $4, 5, 6, 7$. We can repeat the assignment in a round robin fashion so that these two consecutive segments of display resolution 2 are non-overlapping. Once we have decided the disk mapping for the j^{th} display resolution, we can determine where to put the enhancement data blocks of resolution $j + 1$. According to our example, if we choose to assign $\mathcal{R}_0^2$ to the first four disks, enhancement data blocks $\mathcal{V}_0^3 = \{r_{0,0}^3, r_{0,1}^3, r_{0,2}^3, r_{0,3}^3\}$ could then be assigned to disk $4, 5, 6$ and 7. The data block assignment of enhancement to resolution 3 is illustrated in Figure 3.4

<table>
<tr><th>Segment / Disk</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>0</td><td colspan="4">R^2_0</td><td>$r^3_{0,0}$</td><td>$r^3_{0,1}$</td><td>$r^3_{0,2}$</td><td>$r^3_{0,3}$</td></tr>
<tr><td>1</td><td>$r^3_{1,0}$</td><td>$r^3_{1,1}$</td><td>$r^3_{1,2}$</td><td>$r^3_{1,3}$</td><td colspan="4">R^2_1</td></tr>
<tr><td>2</td><td colspan="4">R^2_2</td><td>$r^3_{2,0}$</td><td>$r^3_{2,1}$</td><td>$r^3_{2,2}$</td><td>$r^3_{2,3}$</td></tr>
<tr><td>3</td><td>$r^3_{3,0}$</td><td>$r^3_{3,1}$</td><td>$r^3_{3,2}$</td><td>$r^3_{3,3}$</td><td colspan="4">R^2_3</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr>
</table>

Figure 3.4: Data block assignment for enhancement to resolutions= 3.

resolution $= 1$ To assign data blocks of $\mathcal{R}_i^1$, we have to consider two cases:

- **case 1:** If user selects $r = 3$ upon his admission and if he later chooses $r' = 1$ for the VCR speed, the VOD system must be able to retrieve $VCR\_Speed(3, 1) = 8/2 = 4$ consecutive segments of display resolution $r' = 1$ in parallel. Consequently, any four consecutive segments, $\mathcal{R}_i^1$, $\mathcal{R}_{i+1}^1$, $\mathcal{R}_{i+2}^1$ and $\mathcal{R}_{i+3}^1$ must be non-overlapped.

- **case 2:** In case the user selects $r = 2$ upon his admission, and if he chooses $r' = 1$ for the VCR speed, the VOD system must be capable of retrieving $VCR\_Speed(2,1) = 4/2 = 2$ consecutive segments of display resolution $r' = 1$ in parallel. As a result, any two consecutive segments, $\mathcal{R}_i^1$ and $\mathcal{R}_{i+1}^1$, must be non-overlapped.

Segment / Disk	0	1	2	3	4	5	6	7
0	R^1_0		$r^2_{0,0}$	$r^2_{0,1}$	$r^3_{0,0}$	$r^3_{0,1}$	$r^3_{0,2}$	$r^3_{0,3}$
1	$r^3_{1,0}$	$r^3_{1,1}$	$r^3_{1,2}$	$r^3_{1,3}$	R^1_1		$r^2_{1,0}$	$r^2_{1,1}$
2	$r^2_{2,0}$	$r^2_{2,1}$	R^1_2		$r^3_{2,0}$	$r^3_{2,1}$	$r^3_{2,2}$	$r^3_{2,3}$
3	$r^3_{3,0}$	$r^3_{3,1}$	$r^3_{3,2}$	$r^3_{3,3}$	$r^2_{3,0}$	$r^2_{3,1}$	R^1_3	
...	...	...	...	...	...	...	...	...

Figure 3.5: Data block assignment of enhancement to resolution= 2

Clearly, if the condition in case 1 is satisfied, the condition in case 2 also holds. One way to accomplish the assignment is to put data blocks of $\mathcal{R}_i^1$ to disk 0 and 1, $\mathcal{R}_{i+1}^1$ to disk 4 and 5, $\mathcal{R}_{i+2}^1$ to disk 2 and 3 and $\mathcal{R}_{i+3}^1$ to disk 6 and 7. Once the assignment is made, the assignment of enhancement data blocks of resolution $r = 2$ can be determined. For example, we can assign $\{r_{0,0}^2, r_{0,1}^2\}$ to disk 2 and 3, $\{r_{1,0}^2, r_{1,1}^2\}$ to disk 6 and 7, $\{r_{2,0}^2, r_{2,1}^2\}$ to disk 0 and 1 and $\{r_{3,0}^2, r_{3,1}^2\}$ to disk 4 and 5 (Figure 3.5).

resolution $= 0$ To assign data blocks in $\mathcal{R}_i^0$, there are three cases:

- **case 1:** $r = 3$ and $r' = 0$. In this case, the VOD system must be able to retrieve $VCR\_Speed(3,0) = 8$ consecutive segments in parallel. Therefore, $\mathcal{R}_i^0 \cdots \mathcal{R}_{i+7}^0$ must be non-overlapped.
- **case 2:** $r = 2$ and $r' = 0$. In this case, the VOD system must be able to retrieve $VCR\_Speed(2,0) = 4$ consecutive segments in parallel. Therefore, $\mathcal{R}_i^0 \cdots \mathcal{R}_{i+3}^0$ must be non-overlapped.
- **case 3:** $r = 1$ and $r' = 0$. In this case, the VOD system must be able to retrieve $VCR\_Speed(1,0) = 2$ consecutive segments in parallel. Therefore, $\mathcal{R}_i^0$ and $\mathcal{R}_{i+1}^0$ must be non-overlapped.

As long as we can satisfy case 1, all other cases can be satisfied. The following disk assignment makes the condition in all cases hold: $\mathcal{V}_0^0 = \{r_{0,0}^0\}$ to disk 0, $\mathcal{V}_1^0 = \{r_{1,0}^0\}$ to disk 4, $\mathcal{V}_2^0 = \{r_{2,0}^0\}$ to disk 2, $\mathcal{V}_3^0 = \{r_{3,0}^0\}$ to disk 6, $\mathcal{V}_4^0 = \{r_{4,0}^0\}$ to disk 1, $\mathcal{V}_5^0 = \{r_{5,0}^0\}$ to disk 5, $\mathcal{V}_6^0 = \{r_{6,0}^0\}$ to disk 3 and $\mathcal{V}_7^0 = \{r_{7,0}^0\}$ to disk 7. Again, we know the assignment for enhancement to resolution $r = 1$ once the assignment of $\mathcal{R}_i^0$ is determined. The final assignment is shown in Figure 3.1.

3.4 An Assignment Algorithm for Homogeneous Video Files

The number of display resolutions that the assignment algorithm can support is not limited to four. Indeed, the algorithm can support any number of display resolutions as long as the coding of the video satisfies certain conditions, which we will state later. In this section, we show how we can generalize the assignment algorithm which is presented in the previous example to support n display resolutions.

The assignment algorithm has to maintain certain properties that the multi-resolution file system has. The file system must be capable of offering normal and VCR display as well as maintaining load balancing feature. In the normal display, data blocks distributed among the disks of a displayable segment can be retrieved at a time. This indicates that we cannot assign two data blocks belonging to a segment to the same disk. The argument is also true for VCR display. If a client subscribes any display resolution r as the normal display, the system must be able to offer VCR service of display resolution r', where $0 \leq r' < r$, to the user at any instant. Since the VCR operations are mimicked by displaying several consecutive segments of lower resolution to user at a time, these consecutive segments must also be non-overlapping. Besides providing normal and VCR services, the layout of a video file created by the algorithm must be load balanced for any display resolution. In Chapter 5, we show the proof that the proposed assignment algorithm has achieved all these properties.

The server system is based on the following assumptions in order to provide the multi-resolution display services and maintain the disk load balancing feature:

C 1 *The minimum number of disks for a video file is* $d = \sum_{i=0}^{n} b_i$.

C 2 *The number of disk blocks of display resolution* j, $\sum_{i=0}^{j} b_i$, *is an integer multiple of the number of disk blocks of display resolution* $j-1$, $\sum_{i=0}^{j-1} b_i$, *such that* $\sum_{i=0}^{j} b_i \bmod \sum_{i=0}^{j-1} b_i = 0$.

The condition C1 ensures that no two data blocks belonging to a segment of highest display resolution are assigned to the same disk. As a result, it provides the feasibility of normal and VCR implementation. The condition C2 eliminates the possiblity that several consecutive segments of VCR display overlap. Based on these two conditions, we propose the following video file assignment algorithm.

The assignment algorithm has two main steps: 1) creates a mapping table, 2) maps the video data blocks to the disks using the mapping table. A mapping table which we call *template* is first created when the VOD server performs the disk assignment for a video. A template is actually a map of data blocks to disks with a limited number of segments. The number of segments depends on the number of disks d in the disk array and the maximum number of display resolution n that the server supports. Usually, the template is in the form of table or array of size $\frac{d}{b_0} \times d$, where $\frac{d}{b_0}$ is the row size (or the number of initial segments of the video file) while d is the number of disks in the system. A typical example of template is shown in Figure 3.1. The template maps the disk blocks of a video supporting 4 display resolutions into 8 disks, and hence the size of the template is 8×8. Once the template is created, the i^{th} segment of the video file will follow the assignment of the $(i \bmod \frac{d}{b_0})^{th}$ row of the template. To visualize the effect, take the assignment of video block r_{80}^{0} as an example. The template shown in

Figure 3.1 has 8 segments which means that the layout of data block repeats for every 8 consecutive segments. We know from the notation of r^0_{80} that the data block is located at the 8^{th} segment of the video file; thus, the disk assignment can be found in the first segment in the template. Also, the block is the first block of display resolution 0; therefore, it should be in the same disk as the block r^0_{00}. Consequently, the data block will be assigned to the 8^{th} segment of disk 0. The rest of the video file can be assigned to disks in a similar way.

There are two advantages of using template. In the first place, the size of template is usually so small that it can be stored into the server main memory without utilizing too many resources. Another reason is any data block assignment or retrieval can be translated into a simple template look up. Therefore, it is computationally efficient to locate any data block in any segment of a video file.

The idea of constructing a template is similar to the procedure described in Section 3.3. We start the assignment from the highest display resolution. The fact that the higher display resolution has more data blocks than lower display resolution in a segment implies that it requires more disks than lower one. Thus, we can determine the placement of enhancement data blocks to the higher display resolution once the disk assignment of the two consecutive display resolutions is determined. According to this principle, the template construction consists of two steps : 1) set up *base templates*, 2) assign enhancement data blocks based on the base template. The first step of template construction is to build a set of base template. A base template is a collection of non-overlapping disk assignment sets dedicated to a display resolution. Therefore, the disk assignments in a base template dedicated to display resolution j satisfy the following conditions:

1. $\cup_{i=0}^{P_j} DISK(\mathcal{R}_i^j) = \{0, 1, \ldots, d-1\}$
2. $\cap_{i=0}^{P_j} DISK(\mathcal{R}_i^j) = \emptyset$

where $P_j = \frac{d}{\sum_{i=0}^{j} b_i}$. Once the base template is created, the data block assign-

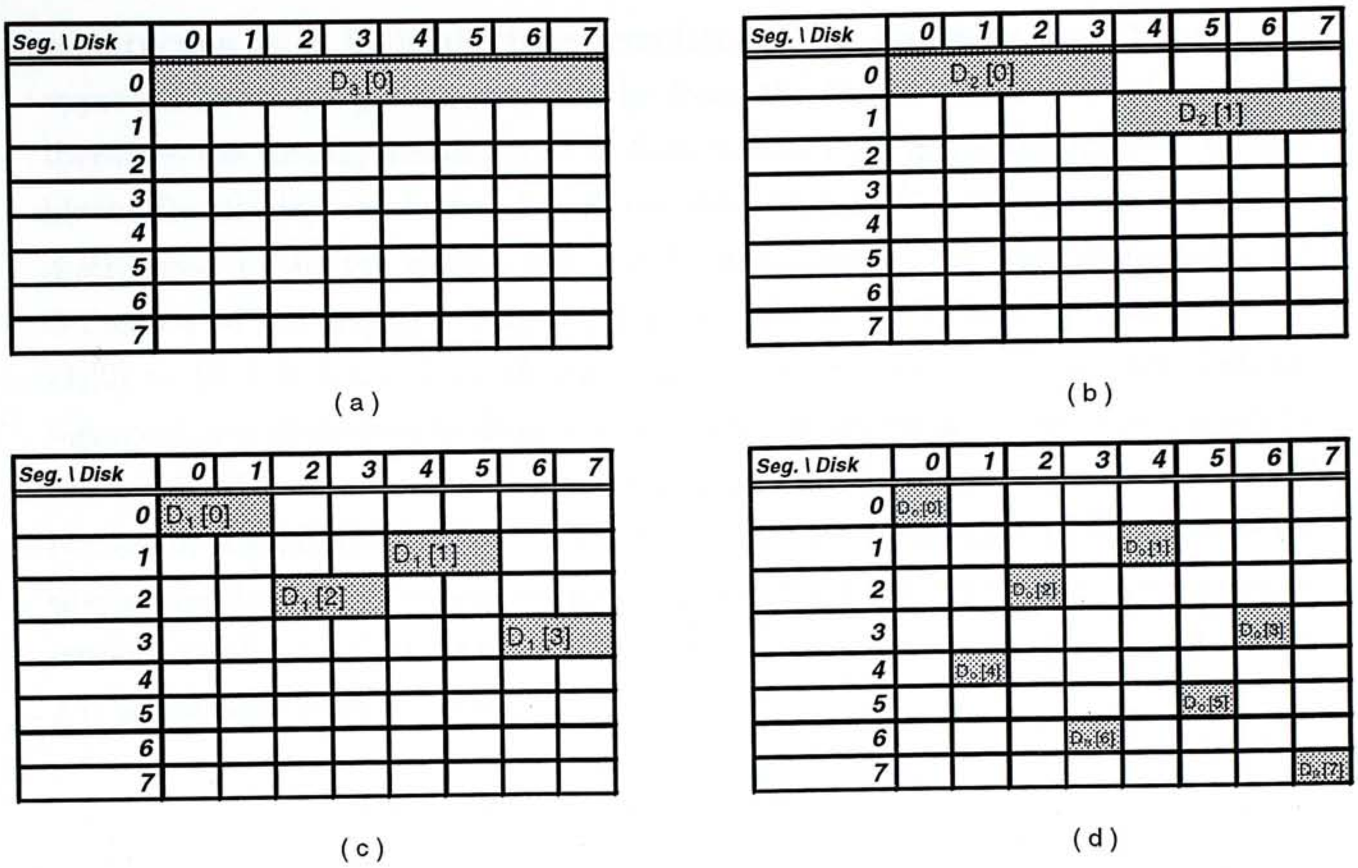


Figure 3.6: The construction of base template of 4 display resolutions

ment of display resolution j for P_j consecutive segments can be determined. In addition, since the number of rows for a template is $\frac{d}{b_0}$, which is the number of disk assignment sets of base template dedicated to display resolution 0, the disk assignment sets of display resolution j at segment i', where $i' \bmod i = 0$ and $P_r \le i' < \frac{d}{b_0}$, can be determined by applying the base template $\frac{\sum_{i=0}^{j} b_i}{b_0}$ times. Whenever the base templates dedicated to two consecutive display resolutions , say j and $j-1$, are made, we can then determine the enhancement of resolution j at segment i by applying the following formula:

$$DISK(\mathcal{V}_i^j) = DISK(\mathcal{R}_i^j) - DISK(\mathcal{R}_i^{j-1}) \tag{3.4}$$

The template can be constructed by recursively applying Equation (3.4) to all display resolution j, where $1 \le j \le n-1$, and all segments i, where $0 \le i \le \frac{d}{b_0}$.

Let us illustrate the construction of the base template by an example shown in Figure 3.6, which assumes that the VOD system to assign data blocks of video supporting 4 display resolutions into $d = 8$ disks. The first step of template construction is to build the base templates of the 4 display resolutions. Our approach is to assign the data blocks from the highest display resolution and therefore the display resolution 3 is first considered. Since the number of disk blocks for display resolution 3 is 8, the data blocks of each segment are evenly distributed to all the disks. We use $D_3[0]$ to denote the disk assignment set dedicated to the display resolution 3 at segment 0 in the base template. Hence, $D_3[0] = \{0,1,2,3,4,5,6,7\}$ (Figure 3.6(a)). In general, $D_j[i]$ denotes disk assignment sets dedicated to display resolution j at segment i in the base template. Next, we will create the base template of display resolution 2. Since the number of blocks for display resolution 2 is 4, which is half of the number of data blocks for display resolution 3, the base template for display resolution 2 has two segments, namely $D_2[0]$ and $D_2[1]$. One of the possible assignments for display resolution 2 is as follows (Figure 3.6(b)):

$$D_2[0] = \{0,1,2,3\} \text{ and } D_2[1] = \{4,5,6,7\}$$

Similarly, there are four segments in the base template of display resolution 1 because $b_2 = 2$. The data block of the display resolution can possibly be assigned as (Figure 3.6(c)):

$$D_1[0] = \{0,1\}, D_1[1] = \{2,3\}, D_1[2] = \{4,5\} \text{ and } D_1[3] = \{6,7\}$$

Finally, the eight segments of display resolution 0 becomes (Figure 3.6(d)):

$$D_0[0] = \{0\}, D_0[4] = \{1\}, D_0[2] = \{2\}, D_0[3] = \{6\},$$

$$D_0[4] = \{1\}, D_0[5] = \{5\}, D_0[6] = \{3\} \text{ and } D_0[7] = \{7\}$$

After creating the base templates of these 4 display resolutions, we proceed to assign the enhancement data block of each resolution. We can apply Equation (3.4) Since $D_j[i] = DISK(\mathcal{R}_i^j)$ and $D_{j-1}[i] = DISK(\mathcal{R}_i^{j-1})$, the equation becomes :

$$DISK(\mathcal{V}_i^j) = D_j[i \bmod P_j] - D_{j-1}[i \bmod P_j] \tag{3.5}$$

By applying this equation, we can get the template shown in Figure 3.1.

The video placement algorithm for a general $N-$resolution is given in Figure 3.7. The variables n and d are the user-defined parameter that specified the number of display resolutions supported by the video file and the number of parallel disks in the VOD system respectively.

Algorithm $\mathcal{P}$ *Video Object Placement Algorithm*
Procedure N_RESOLUTION_VIDEO_OBJECT_ASSIGNMENT (n, d)

Begin
  Let $D_{n-1}[0] \leftarrow \{0, 1, ..., d-2, d-1\}$;
  For $k \leftarrow n-2$ to 0 Do

    For $l \leftarrow 0$ to $P_{k+1} - 1$ Do
      $X \leftarrow D_{k+1}[l]$;
      For $i \leftarrow 0$ to $\frac{P_k}{P_{k+1}} - 1$ Do
        For $j \leftarrow 0$ to $\sum_{m=0}^{k} b_m - 1$ Do
        $x \leftarrow$ the least element in X;
        $D_k[P_{k+1} \times i + l] \leftarrow D_k[P_{k+1} \times i + l] \cup \{x\}$;
        $X \leftarrow X - \{x\}$;
        Endfor
      Endfor
    Endfor

    For $l \leftarrow 0$ to $\frac{P_0}{P_k} - 1$ Do
      For $i \leftarrow 0$ to $P_{k+1} - 1$ Do
        For $j \leftarrow 0$ to $\frac{P_k}{P_{k+1}} - 1$ Do
          $\mathcal{V}^{k+1}_{l*P_k + P_{k+1} \times j + i} \leftarrow D_{k+1}[i] - D_k[P_{k+1} \times j + i]$;
        Endfor
      Endfor
    Endfor

  Endfor

  For $l \leftarrow 0$ to P_0 Do
    $\mathcal{V}^0_l \leftarrow D_0[l]$
  Endfor

  Assign R^k_j components ,$\forall k \in \{0, 1, \ldots, k\}$,
  for segment j according to $\mathcal{V}^k_j$

End N_RESOLUTION_VIDEO_OBJECT_ASSIGNMENT

Figure 3.7: N-resolution video placement algorithm

Chapter 4

Disk Scheduling and Admission Control

In the previous chapter, we have developed the framework of the multi-resolution video file system as well as the data block assignment algorithm. The framework does not make any assumption on how the system admits a new client and how it schedules the retrievals of videos. In this chapter, we give a disk scheduling algorithm for deciding how the video server schedules the data retrievals of different video streams so that the aggregate bandwidth of the disk array is fully utilized. Then an admission control algorithm, which decides whether a new client can be admitted into the system, is presented.

4.1 Disk Scheduling Algorithm

Suppose a VOD server stores c homogeneous videos in the disk system. Let us denote each video file as m_x where $0 \leq x < c$. Upon receiving the first playback request for video file m_x from a client, the server creates a stream $S_{id}^{m_x}$ for that client, where id is a unique number so that one can identify a particular video stream in the system. Since it is possible that several clients may view the same video in different segments at the same time, the corresponding video streams share the same video file and thus the same m_x. In addition, each stream $S_{id}^{m_x}$ associates with a transfer rate PR data sets/second. Data set consists of a segment or several segments of video data, depending on the user's display modes. If a user is currently in normal display mode, the data set consists of a segment of video. That is to say that the playback rate of normal display is PR segments/second. However, if a user is viewing the video in the VCR mode, the data set probably contains the video data blocks from several consecutive segments of video. Therefore, although the video server transfers PR data sets/second, it allows the users to view video in the VCR mode by including several consecutive segments of video into the same data set.

The VOD system employs a *cyclical scheduling policy* and retrieves data for each video stream in *service rounds.* During each service round, the server retrieves a fixed amount of data for each client from the disks. The maximum duration of a service round is determined by the data sets transfer rate, PR, of the streams. That is to say that the duration of a service round is not longer than $\frac{1}{PR}$ seconds. The value is justified because the time interval between the transmissions of two consecutive data sets is $\frac{1}{PR}$ seconds regardless the client's display mode.

Usually, the aggregate bandwidth of the disk array is much larger than the bandwidth requirement for retrieving a video segment , which implies that the server can retrieve more than one video segment during a service round. For presentation convenience, we divide a service round into a smaller unit called *session.* We have already shown in Equation (3.1) that the total disk latency of retrieving $\mathcal{U}$, which is the size of a video data block in unit of bytes, is $T^{max}(\mathcal{U})$ seconds, and therefore the minimum time duration for a session is $T^{max}(\mathcal{U})$. Each disk should start to fetch the required data set at the beginning of a session and the data set should be available in the buffer by the end of the session. We denote the data blocks of display resolution k of video stream $S_{id}^{m_x}$ retrieved at session l of service round i as $Segment(S_{id}^{m_x}, i, l, k)$. The resultant set is equivalent to $\mathcal{R}_j^k$ of video m_x. Consider the example shown in Figure 4.1, suppose the parallel disk system contains a video m_0 using the layout in Figure 3.1. Let $user_1$, $user_2$ and $user_3$ be requesting to playback video m_0 with normal display resolution 3, 1 and 1 respectively. During service round 0, $user_1$ and $user_2$ start to playback video at segment 0 while $user_3$ requests to start at segment 1. Upon receiving these requests, the system initializes three streams, $S_0^{m_0}$, $S_1^{m_0}$ and $S_2^{m_0}$, by allocating enough buffer to hold the retrieving data. Afterwards, the video streams are periodically retrieved from disks in *service round*. Take stream $S_1^{m_0}$ for example, blocks r_{00}^0 and r_{00}^1 are retrieved at service round 0, blocks r_{10}^0 and r_{10}^1 are taken at service round 1 and so on.

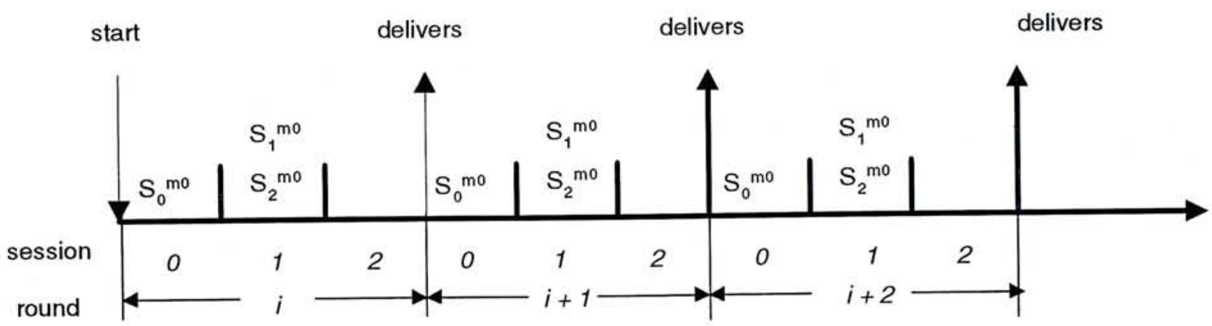


Figure 4.1: Schedules the disk retrievals of 3 video streams

The number of streams sustained during a service round depends on the display QOS requirement from clients. The display quality of video playback is directly proportional to the display resolution that the system uses to serve the client. That is to say, the better the display quality, the higher the display resolution the system uses. Since a video segment of a higher display resolution requires more data blocks than the lower display resolution, more disks are involved in the data retrieval of one video segment. If a client selects his normal display resolution as n, all d disks are involved every time when a segment of video is retrieved and hence no spare disks are available to serve other clients in the same session. If the normal display resolution, however, is less than n, the server needs not involve all d disks in each session. Consequently, more than one video stream retrievals can probably be scheduled in a session of the same service round. The degree of parallel retrievals, hence, depends on the display resolution with which the streams associated. For presentation convenience, we divide video streams into two categories: 1) simultaneous access streams and, 2) sequential access streams. If two or more video streams are scheduled to access the disks simultaneously in a session of the same service round, we call these the simultaneous access streams, otherwise we call these the sequential access streams. Formally, we have:

Definition 5 *Two video streams can simultaneously be retrieved from a parallel I/O system if*

$$DISK(Segment(S_i^{m_x}, a, b, k)) \cap DISK(Segment(S_j^{m_y}, a, b, k')) = \emptyset \quad \forall a, b, k, k'$$

where a is the service round number, b is the session number of service round a and k, k' are the normal display resolutions.

Definition 6 *Two video streams can only be retrieved sequentially from a disk array if*

$$DISK(Segment(S_i^{m_x}, a, b, k)) \cap DISK(Segment(S_j^{m_y}, a, b, k')) \neq \emptyset \quad \forall a, b, k, k'$$

where a is the service round number, b is the session number of service round a and k, k' are the normal display resolutions.

It is clear that the number of video stream retrievals sustained in a service round is bounded by the following two cases. If all users subscribe display resolution n as the normal display, the lower bound of the number of streams in a service round becomes $\frac{1}{PR \times T^{max}(\mathcal{U})}$. On the other hand, if all users select display resolution 0 as the normal display upon admission, then the maximum number of video streams in a service round becomes $\frac{P_0}{PR \times T^{max}(\mathcal{U})}$.

Using the example in Figure 4.1, $S_0^{m_0}$ must be sequentially retrieved when stream $S_1^{m_0}$ and $S_2^{m_0}$ are in the system because
$DISK(Segment(S_0^{m_0}, i, 0, 3)) \cap DISK(Segment(S_1^{m_0}, i, 1, 1)) =$

$$DISK(Segment(S_1^{m_0}, i, 1, 1))$$

and$DISK(Segment(S_0^{m_0}, i, 0, 3)) \cap DISK(Segment(S_2^{m_0}, i, 1, 1)) =$

$$DISK(Segment(S_2^{m_0}, i, 1, 1)).$$

On the other hand, since $DISK(Segment(S_1^{m_0}, i, 1, 1)) \cap DISK(Segment(S_2^{m_0}, i, 1, 1)) = \emptyset$. Therefore, stream $S_1^{m_0}$ and $S_2^{m_0}$ can be scheduled in the same session of the same service round.

Segment / Disk	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	$r^3_{0,1}$	$r^3_{0,2}$	$r^3_{0,3}$	$r^0_{0,0}$	$r^1_{0,0}$	$r^2_{0,0}$	$r^2_{0,1}$	$r^3_{0,0}$
1	$r^1_{1,0}$	$r^2_{1,0}$	$r^2_{1,1}$	$r^3_{1,0}$	$r^3_{1,1}$	$r^3_{1,2}$	$r^3_{1,3}$	$r^0_{1,0}$
2	$r^3_{2,1}$	$r^3_{2,2}$	$r^3_{2,3}$	$r^2_{2,0}$	$r^2_{2,1}$	$r^0_{2,0}$	$r^1_{2,0}$	$r^3_{2,0}$
3	$r^2_{3,1}$	$r^0_{3,0}$	$r^1_{3,0}$	$r^3_{3,0}$	$r^3_{3,1}$	$r^3_{3,2}$	$r^3_{3,3}$	$r^2_{3,0}$
4	$r^3_{4,1}$	$r^3_{4,2}$	$r^3_{4,3}$	$r^1_{4,0}$	$r^0_{4,0}$	$r^2_{4,0}$	$r^2_{4,1}$	$r^3_{4,0}$
5	$r^0_{5,0}$	$r^2_{5,0}$	$r^2_{5,1}$	$r^3_{5,0}$	$r^3_{5,1}$	$r^3_{5,2}$	$r^3_{5,3}$	$r^1_{5,0}$
6	$r^3_{6,1}$	$r^3_{6,2}$	$r^3_{6,3}$	$r^2_{6,0}$	$r^2_{6,1}$	$r^1_{6,1}$	$r^0_{6,0}$	$r^3_{6,0}$
7	$r^2_{7,0}$	$r^1_{7,1}$	$r^0_{7,0}$	$r^3_{7,0}$	$r^3_{7,1}$	$r^3_{7,2}$	$r^3_{7,3}$	$r^2_{7,0}$

Figure 4.2: Template of video 2

It is important to point out that a stream may not be scheduled in the same session of each service round, i.e. retrieval of stream $S_i^{m_x}$ scheduled in session l of service round i may be scheduled in session j of service round $i+1$ where $l \neq j$. This is because two simultaneous access streams in the current service round may become sequential access streams in the next or any service round later. For example, if another video m_1 in the storage system exists and its layout template is shown in Figure 4.2. Consider the scenario in Figure 4.1, suppose a new client requesting the video service of video m_1 with normal display setting at display resolution 1 comes into the system. The VOD server creates a stream $S_3^{m_1}$ for the new incoming client and schedules the disk resources for the stream. It is clear that the stream $S_3^{m_1}$ must be scheduled in session 2 of service round i because $DISK(Segment(S_0^{m_0}, i, 0, 3)) \cap DISK(Segment(S_3^{m_0}, i, 2, 1)) \neq \emptyset$, and $DISK(Segment(S_1^{m_0}, i, 1, 1)) \cap DISK(Segment(S_2^{m_0}, i, 1, 1)) \cap$ $DISK(Segment(S_3^{m_0}, i, 2, 1)) \neq \emptyset$. However, the three streams $S_1^{m_0}$, $S_2^{m_0}$ and $S_3^{m_1}$ can be scheduled in session 2 of service round i+1 since $DISK(Segment(S_1^{m_0}, i, 1, 1))$

$\cap DISK(Segment(S_2^{m_0}, i, 1, 1)) \cap DISK(Segment(S_3^{m_0}, i, 2, 1)) = \emptyset$. As a result, the streams $S_1^{m_0}$, $S_2^{m_0}$ and $S_3^{m_1}$ are simultaneous access streams and the scenario is demonstrated in Figure 4.3. In order to record the scheduling order of the data

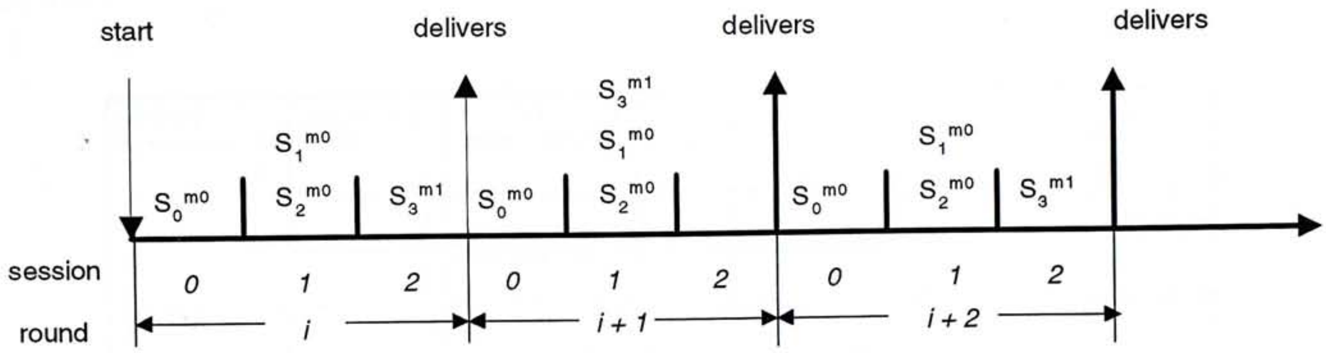


Figure 4.3: Add a stream $S_3^{m_1}$ to the existing scheduling

block retrievals for various streams as well as to simplify the scheduling of a new request, the system uses a *retrievals scheduling map* to accomplish the job. A typical sample is illustrated in Figure 4.4 which represents the schedule for the four streams in Figure 4.3 in the form of retrievals scheduling map. The first column is the service round number, the second column is the session number and the remaining columns are disk numbers, where $d = 8$. If a disk is being utilized in session l of service round i, the corresponding entry in the map is marked with the stream number. For example, stream $S_0^{m_0}$ requires disk set $DISK(Segment(S_0^{m_0}, i, 0, 3)) = \{0, 1, 2, 3, 4, 5, 6, 7\}$ to transfer data at session 0 of service round i and, consequently, all disks are marked with $S_0^{m_0}$ at session of service round i. It is important to point out that the length of the retrievals scheduling map is of P_0 service rounds because all the disk assignment patterns are repeated for every P_0 segments as explained in the previous section. Therefore, the retrievals scheduling map can be stored into the main memory in the system without utilizing much memory resources.

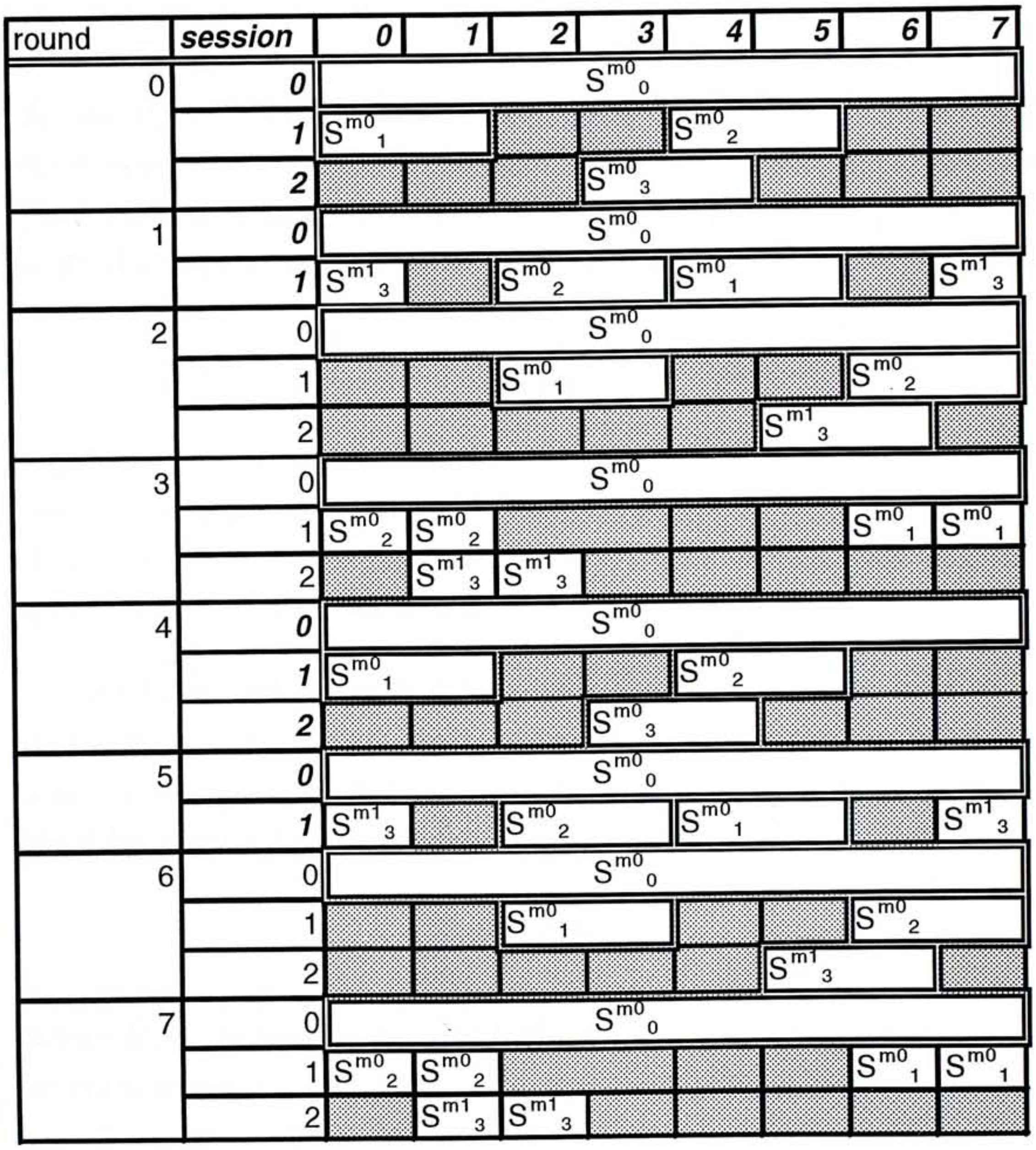


Figure 4.4: Retrievals scheduling map of Figure 4.3

4.2 Admission Control

There are two conditions for the system to consider whether it should admit a new user into the system or not. These conditions are:

1. *To satisfy the continual display requirement of a stream.* The time between displaying two continuously segment is $\frac{1}{PR}$. Therefore, the system has to place the next segment of video data into the buffer within $\frac{1}{PR}$ seconds. To fulfill the requirement, the system needs to satisfy the following condition:

$$\sum_{i=1}^{k+1} T_j^{max}(\mathcal{U}) < \frac{1}{PR} \quad 0 \leq j < d \tag{4.1}$$

   where k is the number of streams that the system has currently admitted and j is the index to the disk subsystem. The condition states that the total disk latency of $k+1$ streams should be less than the playback time interval between consecutive segments of any stream.

2. *To satisfy the individual disk bandwidth requirement.* Since the bandwidth of a disk is limited, it can only support a maximum number of streams at a given time interval. If k streams are using disk j for data retrieval, then the following condition needs to be satisfied:

$$\sum_{i=1}^{k+1} PR \cdot \mathcal{U} < B_j \tag{4.2}$$

   where B_j is the maximum bandwidth of disk j and PR is the playback rate of video streams.

Based on the two conditions described above, we can obtain the admission algorithm as shown in Figure 4.5. The algorithm consists of two parts. The first part (line 1 to line 7 of Figure 4.5) examines whether the video template subscribed by the to-be-admitted client, $DISK(R^{res}_{seg\_start+i})$, of stream $k+1$ will violate Equation (4.1) and (4.2). If the two conditions are satisfied, the algorithm

Algorithm *Admission Control Algorithm*
Procedure Admit_User($k+1, stream, res, seg\_start$)

/* $k+1$: the to-be admitted $k+1^{th} streams$ */
/* $stream$: the stream, $S_i^{m_x}$, that associate with video m_x */
/* res : the normal display resolution the client subscribed */
/* seg_start : the starting segment number where the client issued the playback command */

Begin

For $i \leftarrow 0$ to $P_0 - 1$ Do
  For all $j \in DISK(R^{res}_{seg\_start+i})$ Do
    If ($\sum_{i=0}^{k} T_j^{max}(\mathcal{U}) > \frac{1}{PR}$) or ($\sum_{i=0}^{k} PR \cdot \mathcal{U} > B_j$)
      Return false
    Endif
  Endfor
Endfor

For $i \leftarrow 0$ to $P_r - 1$ Do
  For all session $l \in$ round i Do
    If $\neg\exists$ a slot for $DISK(R^{res}_{seg\_start+i})$
      create a new session
    Endif
    marked the slot of disk j at session l of round i with $stream$
  Endfor
Endfor
Return true

End Admit_User

Figure 4.5: Admission control algorithm

advances to the second part (line 8 to line 18 of Figure 4.5) to schedule the incoming stream to disk slots that are available. For instance, consider the scenario in Figure 4.3 again. Suppose the number of maximum sessions supported by each disk is 4 and we assume streams $S_0^{m_0}$, $S_1^{m_0}$ and $S_2^{m_0}$ are already admitted before service round i. After the system has received request for allocating resource for a new stream $S_3^{m_1}$ at service round i, then it starts the Admit_User algorithm to see if the request of stream $S_3^{m_1}$ can be fulfilled. The first part of the algorithm verifies for the possibility of adding load to the disk set $DISK(R_0^1) = \{3,4\}$, $DISK(R_1^1) = \{7,0\}$, $DISK(R_2^1) = \{5,6\}$, $DISK(R_3^1) = \{1,2\}$, $DISK(R_4^1) = \{3,4\}$, $DISK(R_5^1) = \{7,0\}$, $DISK(R_6^1) = \{5,6\}$ and $DISK(R_7^1) = \{1,2\}$ at service round i, $i+1$, $i+2$, $i+3$, $i+4$, $i+5$, $i+6$ and $i+7$ respectively. In this case, the test is passed because the maximum number of load to each disk at any given service round is 2. The second part of the admission algorithm is to perform the scheduling computation. Take the first segment as an example, when the system schedules for segment R_0^1 at service round i, it is found that $DISK(R_0^3)$ of video m_0, $DISK(R_0^1)$ of video m_1 and $DISK(R_1^1)$ of video m_0 overlap at disk 4. Therefore, a new session, session 2, is created at service round i and the corresponding slot in session 2 of service round i is marked with $S_3^{m_1}$. The result of the map for the first P_0 segments is shown in Figure 4.4.

Chapter 5

Load Balancing of the Disk System

Apart from offering normal playback and VCR capabilities, our file system layout can also balance the load among the disks for different display resolutions. In other words, the disk system achieves the load balancing property whenever the video file is accessed, both in normal or VCR display. The following is the proof of the load balancing property of our VOD system.

Lemma 1 *The number of data blocks per segment retrieved for display resolution r, is an integer multiple of the number of data blocks per segment retrieved for display resolution r', where $0 \leq r' < r \leq n$.*

Proof

The lemma can be proved by the transitivity that the number of data blocks of display resolution r, is an integer multiple of display resolution $r-1$, for $0 \leq r < n$. Suppose there exists l consecutive display resolutions exits between r and r' such that $r' < r_1 < r_2 \ldots r_l < r$, and $r_l = r-1$, $r' = r_1 - 1$. By condition

C2, for any two consecutive display resolutions, the following relationship exists:

$$\frac{\sum_{i=0}^{k} b_i}{\sum_{i=0}^{k-1} b_i} = C_{k-1}, \text{where } C_{k-1} \text{ is an integer}$$

It is clear that any two consecutive display resolutions between r and r', inclusively, can be expressed as :

$$\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r_l} b_i} = C_{r_l}, \qquad \frac{\sum_{i=0}^{r_l} b_i}{\sum_{i=0}^{r_l-1} b_i} = C_{r_{l-1}}, \qquad \frac{\sum_{i=0}^{r_l-1} b_i}{\sum_{i=0}^{r_l-2} b_i} = C_{r_{l-2}}, \qquad \ldots, \qquad \frac{\sum_{i=0}^{r_1} b_i}{\sum_{i=0}^{r'} b_i} = C_{r'}$$

where C_{r_l}, $C_{r_{l-1}}$, $\ldots$, $C_{r'}$ are integers.

By substitution, we can relate $\sum_{i=0}^{r} b_i$ and $\sum_{i=0}^{r'} b_i$, such that

$$\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} = C_{r_l} \cdot C_{r_{l-1}} \ldots C_{r'}$$

where $C_{r_l} \cdot C_{r_{l-1}} \ldots C_{r'}$ is an integer. Therefore, $\sum_{i=0}^{r} b_i$ is an integer multiple of $\sum_{i=0}^{r'} b_i$. ∎

Lemma 2 *Let the normal display resolution be r, the amount of data retrieved from every disk is the same over P_r consecutive service rounds, where $P_r = \frac{d}{\sum_{i=0}^{r} b_i}$.*

Proof

If a user has subscribed the maximum display resolution, that is $r = n - 1$, as the normal display, it is clear that the server retrieves equal amount of data from every disk at every service round. The reason is that data blocks of display resolution n in a segment are evenly distributed to each disk in the parallel disk system. However, the case is not trivial when the normal display resolution is set to be less than $n - 1$. In these cases, data blocks of display resolution r

$(0 \leq r < n-1)$ in *one* segment are no longer evenly distributed to every disk. As a result, not every disk is involved when one segment of a video file is retrieved by the server in a service round. We show below that transfer load of every disk is balanced after P_r consecutive service rounds. Since

$$\frac{\sum_{i=0}^{n-1} b_i}{\sum_{i=0}^{r} b_i} = \frac{d}{\sum_{i=0}^{r} b_i} = P_r$$

By Lemma 1, we know P_r must be an integer. This implies that the disk set $\mathcal{D} = \{0, 1, \ldots, d-2, d-1\}$ can be divided into P_r disjoint sub-sets, each of size $\sum_{i=0}^{r} b_i$. Each of this sub-set represents a segment of disk assignment for display resolution r. Without lost of generality, we assume a user starts to play video at a normal speed of resolution r from segment j at service round k. The P_r disjoint sub-sets from segment j to $j + P_r - 1$ can be expressed as :

$$\begin{aligned} DISK(R_j^r) \cap DISK(R_{j+1}^r) \cap \ldots \cap DISK(R_{j+P_r-2}^r) \cap DISK(R_{j+P_r-1}^r) \\ = \emptyset \quad (5.1) \end{aligned}$$

and

$$\begin{aligned} DISK(R_j^r) \cup DISK(R_{j+1}^r) \cup \ldots \cup DISK(R_{j+P_r-2}^r) \cup DISK(R_{j+P_r-1}^r) \\ = \{0, 1, 2, \ldots, d-1\} \quad (5.2) \end{aligned}$$

Equation (5.1) shows that the P_r sub-sets are disjoint and therefore, data blocks of display resolution r are non-overlapping in these P_r consecutive segments. However, Equation (5.1) alone does not guarantee that data blocks of display resolution r in P_r consecutive segments are evenly distributed to every disk and therefore, we need Equation (5.2) to ensure the condition. The *base template* construction (line 3 to line 12 in Figure 3.7) in our proposed assignment algorithm works under the principle of Equation (5.1) and (5.2); thus the layout of a video file also follows the property in Equation (5.1) and (5.2). According to our disk scheduling policy (section 5), one segment of a video is retrieved at one service round for a user. As a result the server retrieves only one data block from all disks over P_r consecutive service rounds data block retrievals. ■

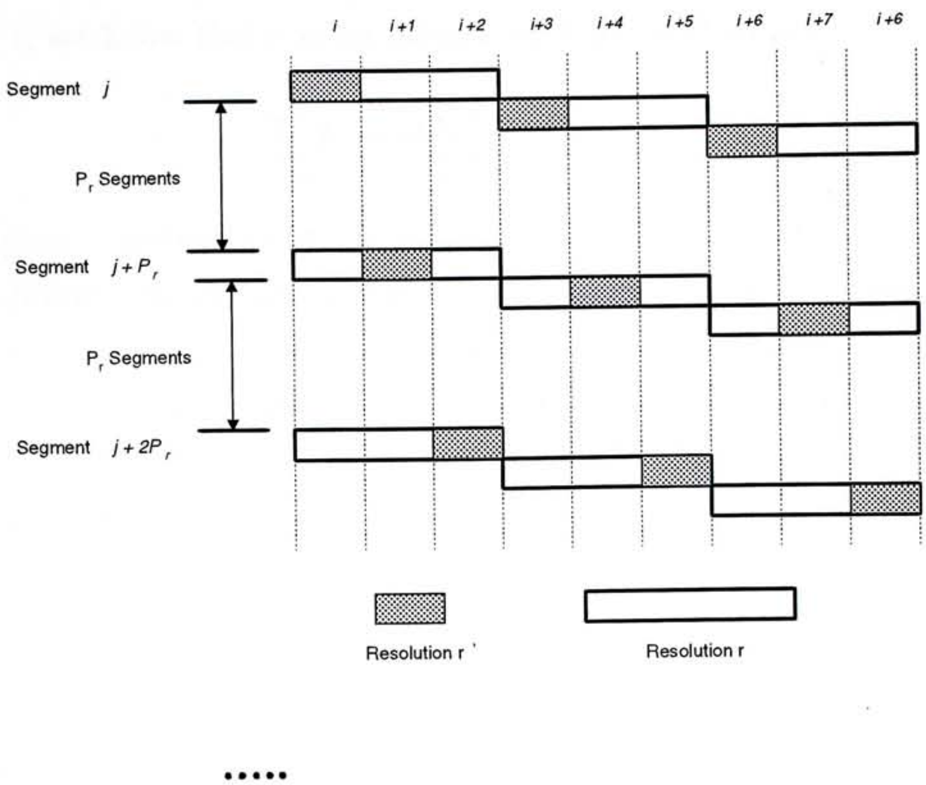


Figure 5.1: Normal and VCR display

Lemma 3 *If a user has subscribed a display resolution r as the normal display upon his admission, the amount of data transfer of every disk for any VCR display using display resolution r', where $0 \leq r' < r \leq n-1$, is the same over P_r consecutive service rounds, where $P_r = \frac{d}{\sum_{i=0}^{r} b_i}$.*

Proof

Since the disk schedule for a video stream has already been fixed upon the admission of a new client, the VCR requests issued during any normal display period do not alter the disk schedule. In other words, any data block for VCR display must be retrieved under the same disk schedule for normal display retrieval. However, instead of retrieving the data blocks from the normal display, the system retrieves the data blocks for the VCR display.

Without losing generality, we assume that a user, who has subscribed resolution r as the normal display, starts to play Fast Forward / Rewind with display resolution r' at segment j (as illustrated in Figure 5.1).

By Lemma 1, we know that r is an integer multiple of r' such that

$$\sum_{i=0}^{r} b_i \bmod \sum_{i=0}^{r'} b_i = 0$$

In the template construction (line 13-19 in Figure 3.7), the base template of display resolution r is repeated for every P_r segments. The scenario is shown in Figure 5.1, the disk assignment of data blocks for resolution r of the j^{th} segment is the same as the assignment of data blocks for segment $j + l \cdot P_r$ (l = 1,2,..etc). Furthermore, by Lemma 2, no components of resolution r are overlapped in this P_r consecutive segments. Let

$$\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} = C_r \tag{5.3}$$

The P_r disjoint sets property of video blocks of resolution r are expressed in Equations (5.1) and (5.2). Since the disk assignment sets of resolution $r-1$ are subsets of resolution r such that $DISK(\mathcal{R}_j^{r'}) \in DISK(\mathcal{R}_j^{r})$. Furthermore, Lemma 1 ensures that r is an integer multiple of r'. Therefore, the disk assignment set of resolution r at segment j is equal to the set union of resolution $r-1$ at segments $j + i \cdot P_r$, where $0 \leq i \leq C_r - 1$. Mathematically, we can express the statement as:

$$DISK(R_{j+i}^{r'}) \cap DISK(R_{j+P_r+i}^{r'}) \cap \ldots \cap DISK(R_{j+(C_r-1)\cdot P_r+i}^{r'}) = \emptyset \tag{5.4}$$

and

$$DISK(R_{j+i}^{r'}) \cup DISK(R_{j+P_r+i}^{r'}) \cup \ldots \cup DISK(R_{j+(C_r-1)\cdot P_r+i}^{r'}) = DISK(R_{j+i}^{r}) \tag{5.5}$$

where $0 \leq i \leq P_r - 1$.

Substitute Equation (5.5) into Equation (5.1), we get

$$DISK(R_j^{r'}) \cap DISK(R_{j+1}^{r'}) \cap \ldots \cap DISK(R_{j+P_{r'}-1}^{r'}) = \emptyset \tag{5.6}$$

Equation (5.6) implies that the retrieving of all video blocks of resolution r' in VCR mode, which is operated under the disk scheduling of the normal display of resolution r, are non-overlapped in consecutive $P_{r'}$ segments.

Also, if we substitute Equation (5.5) into Equation (5.2), we have

$$DISK(R_j^{r'}) \cap DISK(R_{j+1}^{r'}) \cap \ldots \cap DISK(R_{j+P_{r'}-1}^{r'}) = \{0, 1, 2, \ldots, d-1\}$$

$$\bigcup_{i=0}^{P_{r'}-1} DISK(R_i^{r'}) = \{0, 1, 2, \ldots, d-1\} \quad (5.7)$$

Equation (5.7), together with Equation (5.6) , tells us that there is only one unit of video block transferred for each disk during VCR display of resolution r' after $P_{r'}$ consecutive segments. Since the $\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i}$ segments are retrieved in one service round, the load of every disk is balanced in P_r consecutive service rounds. ∎

Theorem 1 *A VOD file system is said to have the* **load balancing** *property with respect to a video file if every disk has the same load over S consecutive service rounds retrievals, where S is an integer multiple of P_r and r is normal display resolution for a client.*

Proof

By Lemma 2, it is shown that the load of every disk is balanced over P_r consecutive service rounds retrievals under the normal display. In Lemma 3, we have shown that the load of every disk is also balanced over P_r consecutive rounds retrievals under the VCR display. Therefore, the claim holds

∎

Chapter 6

Buffer Management

In this chapter, we develop the framework of buffer management for our multi-resolution file system. In addition, we analyze the minimum buffer requirement and minimum starting time for a successful video playback. At the end of the chapter, we show the maximum buffer requirement for a newly admitted user.

6.1 Buffer Organization

The memory buffer of the VOD system is to hold the data retrieved from the disk system temporary before it is consumed by the clients. By using memory buffer, our disk scheduling algorithm can be simplified because the order of video stream retrievals can easily be re-sequenced. Moreover, the video server can smoothen data transmission from the disk system to the network subsystem by using buffer.

Usually, buffer space managed by an I/O scheduler can be either single buffer or dual buffer. In a single buffer system, there is only one set of buffer and data will be consumed as soon as the buffer is filled. On the other hand, the dual buffer system consists of two distinct sets of buffers which are used alternately.

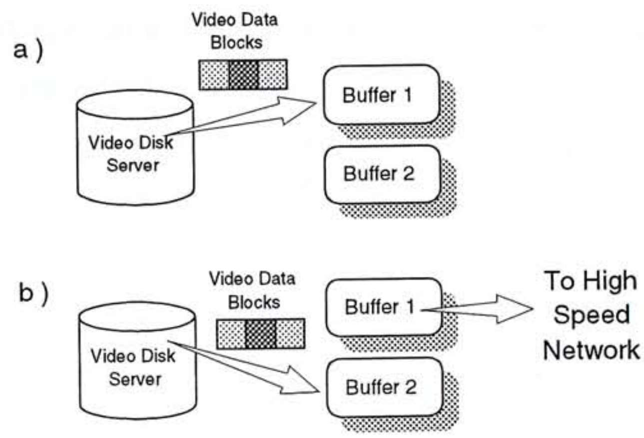


Figure 6.1: Dual buffer system

That is, data retrieved from disks are stored in one set of buffer while the data obtained in the previous service round will be consumed in other set of buffer. Buffer that stored data in service round k will be consumed in service round $k+1$ and verse visa.

The scenario of dual buffer system is illustrated in Figure 6.1. The video server reserves two sets of buffer, namely `buffer1` and `buffer2`, for each admitted client. The size of each buffer set depends on the display quality, namely the display resolution, that a client subscribes. Usually, more buffer resources are allocated for the client who subscribes a higher quality of video display. Upon receiving the video playback request for segment i from the client, the corresponding video data blocks are retrieved from the disks system and then put to the `buffer1`. However, the data blocks in `buffer1` are not immediately transferred to the network in the current service round. In the next service round, the disks read data blocks for the segment $i+1$ of video into the `buffer2`. Once the data blocks for segment $i+1$ are available in `buffer2`, segment i is delivered to the client.

The alternate produce-consume property offered by the dual buffer organization enables the system to take the full advantage of optimizing the disk scheduling. In the proposed disk scheduling scheme, video data retrieved in the previous service round are consumed once the data blocks retrieved in the current service round are ready in buffer. That means video data for different client streams are not consumed at the same time as when the current service round ends. The reason is to have a smooth transition of video data Also, the retrieval of video streams for the incoming clients are arranged to any free session slot in order to optimize the utilization of the aggregate bandwidth of the disk system. Therefore, it is possible that the interval between two consecutive retrievals of a video stream may be more than $\frac{1}{PR \times T^{max}(\mathcal{U})}$ sessions. In other words, retrievals scheduled in session i of round j may not be scheduled in session i of round $j+1$. If single buffer is used, the client may suffer from starvation because video data may be not be available in the buffer space since the last consumption. On the other hand, the interval between two consecutive data consumptions is guaranteed to be exactly $\frac{1}{PR \times T^{max}(\mathcal{U})}$ sessions in the dual buffer organization, and thus it frees the clients from starvation. The reason is that clients only consume data retrieved in the last service round rather than in the current service round, any retrieval delay due to the disk scheduler is compensated by the usage of buffer.

6.2 Buffer Requirement For Different Video Playback Mode

The total buffer required to be allocated for a newly admitted client depends on the video playback quality subscribed by that client. Generally, the size of buffer is directly proportional to the playback quality. That is to say that the higher the playback quality the client subscribes, the more buffer space the video server is required to reserve. Specifically, the amount of buffer to be allocated is determined by the selected display resolution because the display resolution determines the number of data blocks retrieved for normal and VCR display in a service round.

In the following section, we examine the buffer requirement for normal and VCR display :

- **Normal Display**

  The consideration of choosing the size of buffer required for the operation of normal display is quite self-explanatory. In each service round, a complete video segment of normal display is transferred from the disk array system to the buffer. As a result, the size of a buffer set is equal to the number of video data blocks of a video segment of normal display, $\sum_{i=0}^{r} b_i$ where r is the normal display resolution. Thus the total buffer for the dual buffer system is $2 \cdot \sum_{i=0}^{r} b_i$. For example, as in the scenario where a client subscribes video m_0, of which the layout is shown in Figure 3.1, and selects display resolution 1 as normal display. We assume that the client starts the normal playback at the segment 0 at service round 0. The timing diagram of the buffer operation during normal display for the first four service rounds is illustrated in Figure 6.2. The Figure 6.2 consists of five pictures:

  a) shows which video data blocks are retrieved in each service round.

  b) demonstrates the validity of the video blocks consumption in each service round. There are two lines, a thin and a thick one, in the diagram. The thin line indicates the lowest segment number of a video that is retrieved in the current service round. For instance, the lowest segment number of a video retrieved in service round 1 is 1. The other line, the thick one, indicates the highest number of a video segment consumed in the current service round. For example, the highest segment number of a video consumed in service round 3 is 1 in the figure. Thus, if it happens that the highest segment number of consumption is greater than the least segment number of retrieval in the same service round, and hence the value of the thick line is higher than the thin line, the video blocks consumption operation is considered to be invalid because it is impossible for the video blocks to be transferred to network before they are retrieved from the disk array.

  c) shows which video data blocks are transferred to `buffer1` as well as their utilization for each service round. The $y - axis$ of the diagram indicates the number of buffer blocks being utilized.

  d) depicts which video data blocks are transferred to `buffer2` for each service round. The diagram also shows the utilization of `buffer2`.

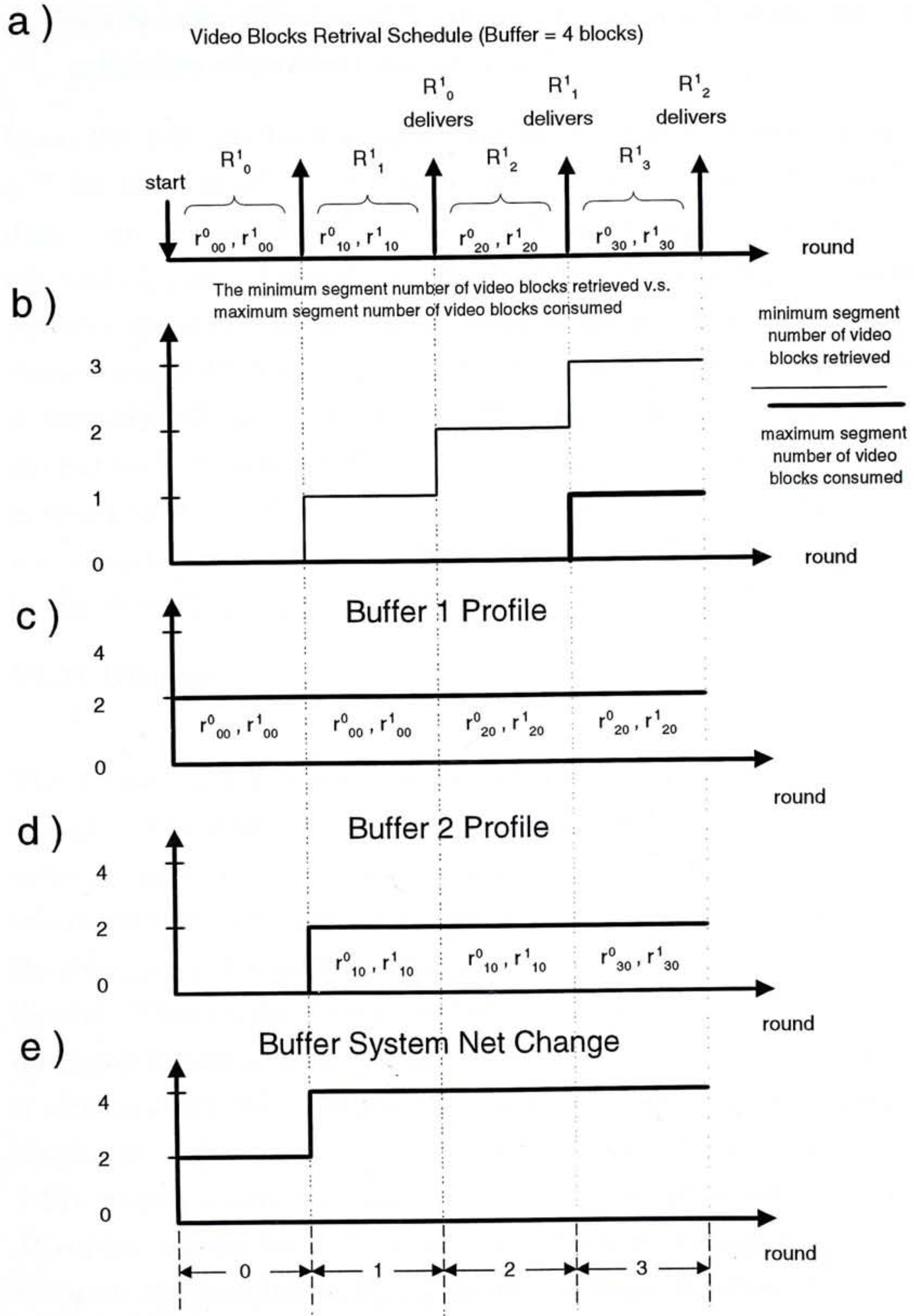


Figure 6.2: Timing diagram for normal display buffer operation

The $y - axis$ of the diagram indicates the number of buffer blocks being utilized.

e) describes the utilization of the dual buffer system. It is the sum of buffer utilization of `buffer1` and `buffer2`.

Upon the first playback request, the VOD server creates a video stream S_1^{mo} for that client. According to the disk scheduling policy, disk 0 and disk 1 are reserved for S_1^{mo} at service round 0 and the data blocks of R_0^1 (r_{00}^0 and r_{00}^1) are retrieved accordingly. The retrieved video segment R_0^1 is temporarily stored in the `buffer1` and is not consumed till the arrival of video segment R_1^1 at service round 1. Once the video segment R_1^1 is available in the `buffer2`, the video segment R_0^1 is then transferred to the client from the `buffer1`. In general, the video segment retrieved in service round k is buffered up and will be delivered in service round $k+1$. As a result, the size of each buffer set is $\sum_{i=0}^{1} b_i = 2$ blocks; the size of dual buffer system is, therefore, $2 \cdot \sum_{i=0}^{1} b_i = 4$ blocks.

- **VCR Display**

The deliberation of selecting optimal buffer size for VCR display is not as straight forward as normal display since the retrieval order of VCR display video segments is not in the same sequence as they are displayed. The data blocks retrieval schedule for a video stream has already been fixed during the admission, it cannot be altered for any display mode, normal or VCR display. That is, the server can only retrieve VCR data blocks under the operation of normal display disk schedule for each service round. In general, if a client starts the VCR playback that is $\frac{P_{r'}}{P_r}$ times faster than the normal display at segment j in service round k, where r and r' are normal and VCR display resolution respectively, then instead of retrieving segments R_j^r which are the normal display video segment at segment j, the server retrieves video segments $R_{j+i \cdot P_r}^{r'}$ in service round k, where $0 \leq i < \frac{P_{r'}}{P_r}$.

The ratio of VCR speed-up determines the size of buffer to be allocated. During the VCR display period, $\frac{P_{r'}}{P_r}$ consecutive video segments are transferred to the client in each service round. However, the retrieval of extra $\frac{P_{r'}}{P_r} - 1$ non-consecutive VCR video segments, $R_{j+i \cdot P_r}^{r'}$ for $1 \leq i < \frac{P_{r'}}{P_r}$, in

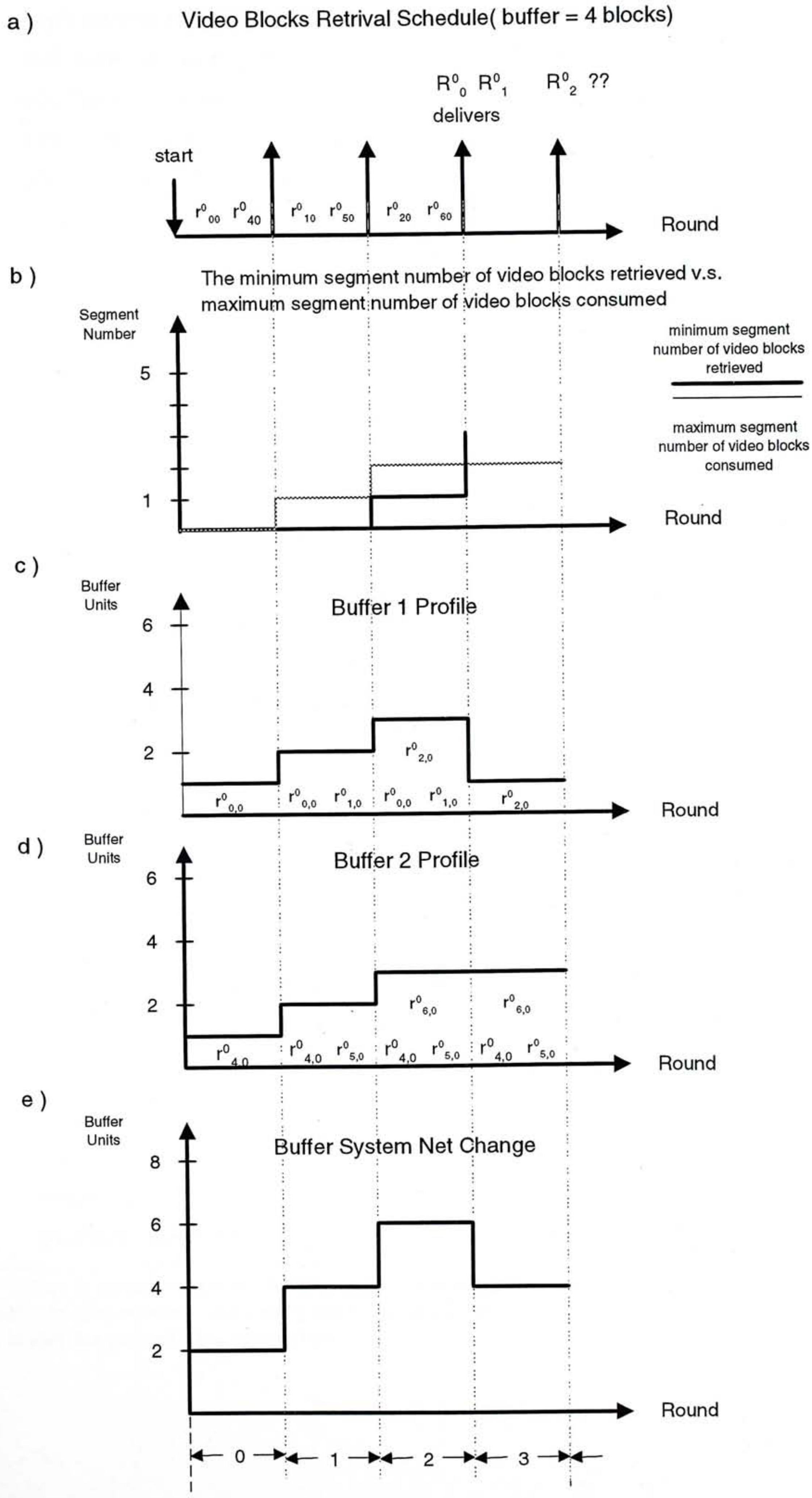


Figure 6.3: Timing diagram of VCR buffer operation when buffer = 6 blocks

each service round makes the server impossible to deliver the set of requested data on time. One of the possible solutions to ensure smooth VCR playback is by means of additional buffer space. All the retrieved VCR video segments are first being stored in the buffer space until the first complete set of requested data is available in the buffer. The out of sequence prefetched VCR video segments are then rearranged in order with the aid of additional buffer and are transferred to client accordingly.

In addition, the *start-up latency* for a VCR display command also depends on the ratio of VCR speed-up. The start-up latency, which is in terms of service rounds, is the time interval between the retrieval and delivery of the first set of requested data. Based on the video blocks retrieval schedule, the first complete set of VCR video segments generally cannot be read in one service round. The disks have to read data into the buffer in several service rounds in order to collect the first complete set of requested data and we define this period as the *prefetech interval.* Therefore, the VCR operation will be in effect with respect to the client after the latency of several service rounds. The length of start up latency is determined by the normal display resolution subscribed and the ratio of VCR speed-up and therefore the length of start-up latency is client dependent.

In order to better illustrate the scenario, consider the case in Figure 6.3 (a). Suppose the user in Figure 6.2 starts the VCR fast forward display that is 2 times faster than the normal speed at segment 0. Under the operation of the normal display disk scheduling, disk 0 and 1 are reserved in service round 0, disk 4 and 5 are reserved in service round 1, disk 2 and 3 are reserved in service round 2 and so on. During the VCR display period, the video server reads two segments of VCR display data blocks of display resolution 0, $R^0_{j+i\cdot 4}$ for $0 \leq i < 2$, into the buffer system. For instance, video blocks $r^0_{0,0}$ and $r^0_{4,0}$ are copied into `buffer1` and `buffer2` in service round 0 respectively[1]. However, the video blocks that currently in the buffer cannot form the first set of requested data, that is $r^0_{0,0}$ and $r^0_{1,0}$. The set is available only when the disks read r^0_{10} and r^0_{50} into the `buffer1` and `buffer2`

[1]There is no special reason for the buffer arrangement of video data blocks. The system can distribute the retrieved video data blocks to any buffer set. Our choice in Figures 6.3 (b) and (c) is just for presentation convenience.

at service round 1 respectively.

Our choice of buffer size in Figure 6.3 is based on the start-up latency as well as the normal display resolution. Figure 6.3 (a) and (c) show that $r^0_{0,0}$ and $r^0_{1,0}$ are entirely transferred into the `buffer1` after service round 1, which implies that the prefetech interval for the first set of requested VCR data is 2 service rounds. In addition, the server needs more resources to buffer up additional video data. Based on this argument, our data blocks retrieval policy only guarantees that the video data blocks are available only at the end of a service round. Therefore, starvation may occur at the time of data delivery, in that the interval between the two consecutive retrievals for a video stream may be over $\frac{1}{PR \times T^{max}(\mathcal{U})}$ seconds. Consequently, the start-up latency needs to be increased to 3 service rounds in order to avoid starvation. As a result, the total buffer in Figure 6.3 equals start-up latency $\times \sum_{i=0}^{r} b_i = 6$.

However, the way to calculate the prefetch interval in Figure 6.3 cannot always ensure that the video server provides smooth video playback service. Consider the Figure 6.3 again. In case the time interval between the data retrievals of service round 2 and 3 is over $\frac{1}{PR \times T^{max}(\mathcal{U})}$ second, $r^0_{3,0}$ and $r^0_{7,0}$ will be unavailable during the service round 3. Meanwhile, the server should transfer $r^0_{2,0}$ and $r^0_{3,0}$ to the client. Under these circumstances, the client will starve and the VCR display service is interrupted. Figure 6.3 (b) clearly demonstrates that the consumption buffer operation at the end of service round 2 is invalid because the highest segment number of consumption is greater the lowest segment number of retrieval.

In order to provide continuous VCR video playback service, the server has to ensure that the highest segment number of data consumption must be always smaller than the lowest segment number of data retrieval. Mathematically, we can define the relationship by the following formula:

$$\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t+1) - 1 < \lfloor \frac{s+t}{P_r} \rfloor \cdot P_{r'} + (s+t) \bmod P_r \tag{6.1}$$

where

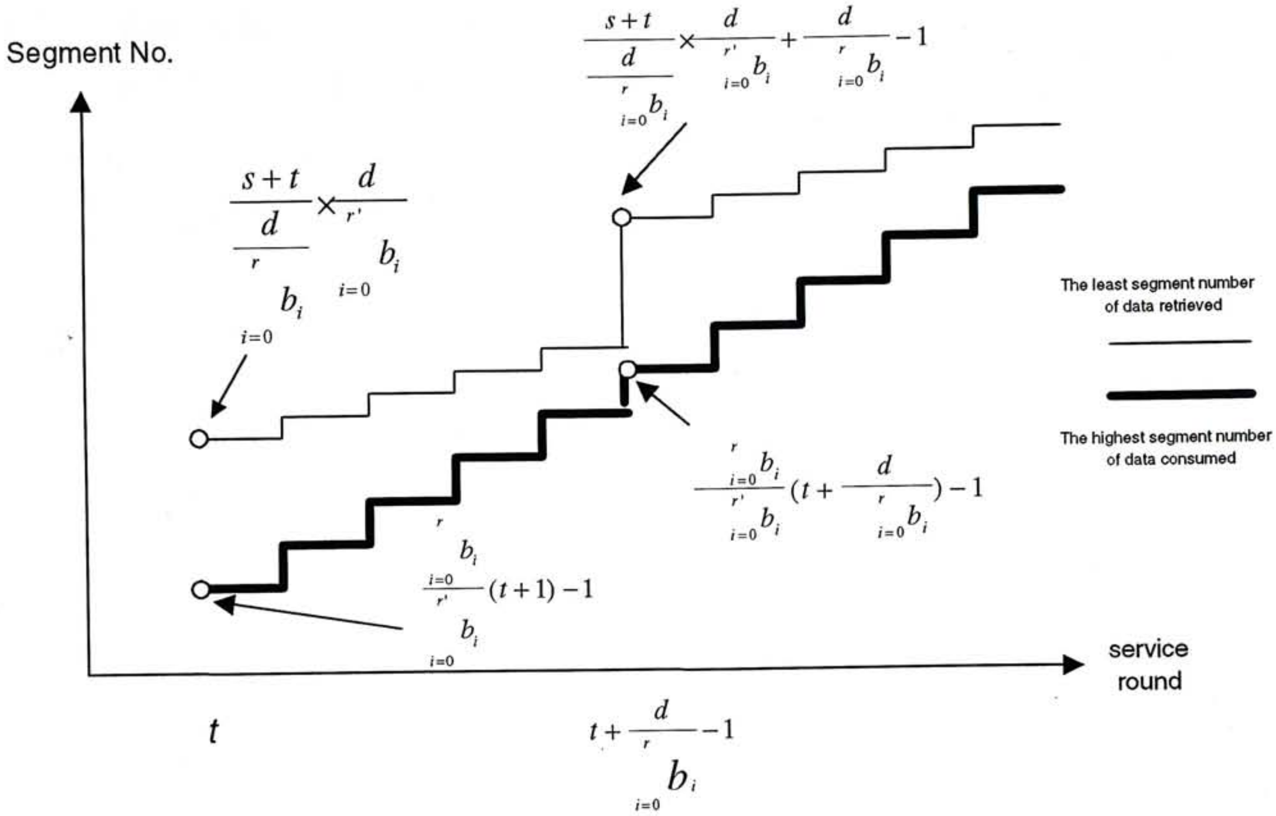


Figure 6.4: Pictorial representation of Equation 6.1

s is the prefetch interval
t is the current service round - s
r is the normal display resolution and,
r' is the VCR display resolution

The left hand side (L.H.S.) of Equation (6.1) is the highest segment number of video blocks consumed at the end of service round t. For each service round during the VCR display period, $\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i}$ consecutive VCR segments are consumed. Therefore, there are $\frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t+1)$ segments delivered at the end of service round t. The right hand side (R.H.S.) of Equation (6.1) is the least segment number of video blocks retrieval at service round $t+1$. The R.H.S. of the Equation (6.1) consists of two parts. The first part is $\lfloor \frac{s+t}{P_r} \rfloor \cdot P_{r'}$ which represents the segment number that has already been read into the buffer in the past $\lfloor \frac{s+t}{P_r} \rfloor$ service rounds. The second part is $(s+t) \bmod P_r$ which represents the least segment number of retrieved data during service round $s+t$ to $s+t+P_r-1$.

Theorem 2 *The prefetch interval for VCR operation of speed-up rate* $\frac{P_{r'}}{P_r}$ *is* $\lfloor P_r \cdot (1 - \frac{P_r}{P_{r'}}) \rfloor + 1$ *service rounds, where* r *and* r' *are the normal and VCR display resolution respectively.*

Proof

We use Equation (6.1) to obtain the result.

Let

$$f(x) = \frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (x+1) - 1 \text{ and,}$$
$$g(x) = \lfloor \frac{s+x}{P_r} \rfloor \cdot P_{r'} + (s+x) \bmod P_r.$$

$f(x)$ is linear and is strictly increasing while $g(x)$ has the following properties:

- it is strictly increasing.
- $g(x)$ is linear except at the point of integer multiple of P_r. At these points, the segment number of data retrieved increases abruptly by $P_{r'}$ because the video blocks between segment $\frac{s+x}{P_r} \cdot P_{r'} + (s+x) \bmod P_r$ and $(\frac{s+x}{P_r} + 1) \cdot P_{r'} + (s+x) \bmod P_r$ are already in the buffer.

If the highest segment number of data consumed is always smaller than the lowest segment number of data retrieved at any given service rounds t and $t + P_r - 1$, then the inequality of Equation 6.1 holds for any service round.

Consider the scenario at service round t of Figure 6.4. Since $s + t$ is an integer multiple of P_r, the term $(s+t) \bmod P_r$ is canceled and we can simplify $f(t)$ and $g(t)$ as

$$f(t) = \frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t+1) - 1 \qquad \text{and} \qquad g(t) = \frac{s+t}{P_r} \cdot P_{r'}$$

Substituting into Equation 6.1, we get

$$\begin{aligned} g(t) &> f(t) \\ \frac{s+t}{P_r} \cdot P_{r'} &> \frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t+1) - 1 \\ s &> 1 - \frac{P_r}{P_{r'}} \end{aligned} \qquad (6.2)$$

At round $t + P_r - 1$, $f(t + P_r - 1)$ and $g(t + P_r - 1)$ can be simplified as

$f(t + P_r - 1) = \frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t + P_r) - 1$ and $g(t + P_r - 1) = \frac{s+t}{P_r} \cdot P_{r'} + P_r + 1$

Substituting $f(t + P_r - 1)$ and $g(t + P_r - 1)$ into Equation 6.1

$$\begin{aligned} g(t + P_r - 1) &> f(t + P_r - 1) \\ \frac{s+t}{P_r} \cdot P_{r'} + P_r - 1 &> \frac{\sum_{i=0}^{r} b_i}{\sum_{i=0}^{r'} b_i} \cdot (t + P_r) - 1 \\ s &> P_r \cdot (1 - \frac{P_r}{P_{r'}}) \end{aligned} \tag{6.3}$$

Since these two equations are applied to the Equation 6.1 , an AND operation must be applied to Equation 6.2 and Equation 6.3. The result then becomes

$$s > P_r \cdot (1 - \frac{P_r}{P_{r'}}) \tag{6.4}$$

Therefore, the prefetch interval for VCR operation of speed-up rate $\frac{P_{r'}}{P_r}$ is $\lfloor P_r \cdot (1 - \frac{P_r}{P_{r'}}) \rfloor + 1$ ∎

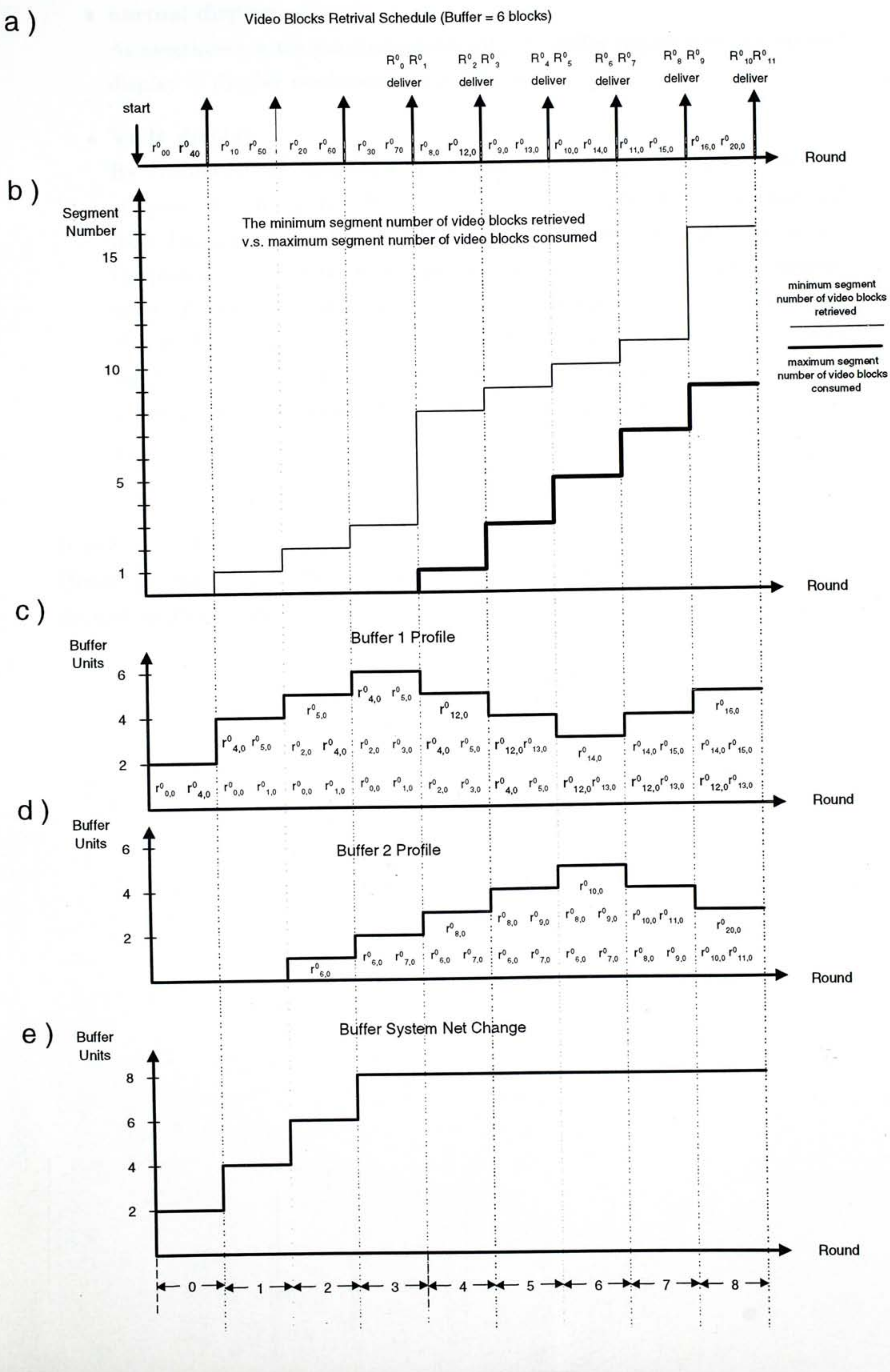
Example

Consider the scenario in Figure 6.3 again. By using the result in Theorem 2, given $r = 1$, $r' = 0$ and $d = 8$, the prefetch interval is $\lfloor \frac{8}{2}(1 - \frac{4}{8}) \rfloor + 1 = 3$ service rounds. Therefore, the start-up latency is 4 service rounds and the size of the dual buffer system that ensures smooth VCR playback of display resolution 0 is $4 \cdot \sum_{i=0}^{1} b_i = 8$. The timing diagram of the VCR buffer operation at the buffer size = 8 blocks is illustrated in Figure 6.5.

Theorem 3 *The maximum buffer requirement for a client who has subscribed r as the normal display resolution is $(s + 1) \cdot \sum_{i=0}^{r} b_i$, where $s = \lfloor P_r \cdot (1 - \frac{P_r}{P_0}) \rfloor + 1$.*

Proof

To prove the theorem, we consider the buffer required for the two display modes, normal and VCR display, and select the maximum requirement between them.


a)
Video Blocks Retrival Schedule (Buffer = 6 blocks)
deliver
start
Round
b)
Segment Number
The minimum segment number of video blocks retrieved
v.s. maximum segment number of video blocks consumed
minimum segment number of video blocks retrieved
maximum segment number of video blocks consumed
Round
c)
Buffer Units
Buffer 1 Profile
Round
d)
Buffer Units
Buffer 2 Profile
Round
e)
Buffer Units
Buffer System Net Change
Round
0
1
2
3
4
5
6
7
8

- **normal display**

  As mentioned in the previous paragraph, the buffer requirement for normal display of display resolution r is $2 \cdot \sum_{i=0}^{r} b_i$.

- **VCR display**

  By Theorem 2, we can deduce that the total buffer required for VCR display of speed-up rate $\frac{P_{r'}}{P_r}$ is $(\lfloor P_r \cdot (1 - \frac{P_r}{P_{r'}}) \rfloor + 2) \cdot \sum_{i=0}^{r} b_i$. It can be observed from Equation (6.4) that the greater the VCR speed-up ratio, the longer the prefetch interval is. The implication is that we can obtain the greatest value of prefetch interval under any normal display resolution by assigning $r' = 0$. As a result, the maximum prefetch interval under the normal display resolution r is $\lfloor P_r \cdot (1 - \frac{P_r}{P_0}) \rfloor + 1$ service rounds. Therefore, the maximum buffer requirement during VCR playback period is $(\lfloor P_r \cdot (1 - \frac{P_r}{P_0}) \rfloor + 2) \cdot \sum_{i=0}^{r} b_i$.

It is clear that $(\lfloor P_r \cdot (1 - \frac{P_r}{P_0}) \rfloor + 2)$ is always greater than 2 as long as $P_r \neq P_0$. Hence, the maximum buffer requirement for a client who has subscribed r as the normal display resolution is

$$(\lfloor P_r \cdot (1 - \frac{P_r}{P_0}) \rfloor + 2) \cdot \sum_{i=0}^{r} b_i.$$

∎

Chapter 7

Conclusions

It is commonly agreed that designing a VOD server is a challenging task. Unlike traditional information server in which textual and numeric data are stored, millions of image sequences are stocked in the video storage server. By nature, the size of images, though in compressed format, is voluminous and hence they occupy a considerable amount of storage space. This sizeable image volume results in the requirement of large data transfer rate for storage and network subsystem during the video playback and VCR display period. Moreover, video data convey meaning only when presented continuously in time. The real-time retrieval requirement in addition to large data transfer rate requirement make the design of a cost-effective video server more difficult.

In this dissertation, we mainly address the problem due to the requirement of large data transfer rate. We propose a video data blocks placement scheme which aims at providing a load balanced and cost-effective storage environment in designing a video server (Chapter 3). Our data blocks placement scheme takes the advantage of subband video coding. The band splitting characteristic of subband video coding enables us to strip the video frame components across the cooperating disks; hence the disk bandwidth is more effectively utilized. In addition, the subband video coding scheme offers the multi-resolution and multi-rate property for each encoded video files (Chapter 2). The multi-resolution coding embeds video data of different display resolutions in a single video stream; thus the system can provide multi-resolution viewing services without replicating several sets of the same video with different resolutions support. The multi-rate property allows the video server system to extract subsets of the embedded video stream to become VCR display. These two properties enable the system to store one set of video to support multi-resolution as well as VCR display services.

The principle of the data block placement scheme is the attempt to allocate video blocks belonging to the same display resolution of two consecutive video segments in disjoint set of disks. The purpose of this scheme is to enhance the possibility of keeping constant transfer bandwidth of disk and network subsystem during the normal and VCR display period. In order to make the placement strategy feasible, two conditions have to be satisfied(Chapter 3.4): The first condition is that the minimum number of disks for a video file is $d = \sum_{i=0}^{n} b_i$ while the second condition is the number of disk blocks of display resolution j, $\sum_{i=0}^{j} b_i$,is an integer multiple of the number of disk blocks of display resolution $j-1$. These two conditions guarantee that video blocks of any two consecutive video segments belonging to the same display resolution, except for the highest display resolution, can be stored on different sets of disks. The video blocks placement algorithm is presented in Chapter 3.

Apart from offering constant data transfer requirement during normal and VCR display period, the data block placement scheme also maintains the load balanced features for various display modes. The load balanced feature of the storage system avoids the occurrence of "hot spots" in conventional storage architectures and it is important in providing cost-effective services. The proof of this important feature in our storage system is illustrated in Chapter 5.

System resources management is a critical design issue in the video server system. Generally, system resources, including computing power, I/O bandwidth as well as memory buffer, are very expensive. If they are not well managed, the system will not be cost effective. To deal with this problem, we have developed the retrieval scheduling and admission control policy in Chapter 4. With the aid of retrieval scheduling map, the video server can admit appropriate number of users effectively so that the system resources are not over utilized. The retrieval scheduling scheme employs the cyclical scheduling policy in scheduling the data retrievals for each video stream. The cyclical scheduling policy means data retrievals, which are proceeded in service rounds, and it ensures that the computing power an, d disk I/O can be equally shared by each video stream.

Finally, we present the buffer management scheme in Chapter 6. In our system, we use the dual buffer system approach. Our retrieval scheduling scheme assumes that the time interval between two consecutive data retrievals of the same video stream may not be constant; thus the transfer rate of the video stream may not be guaranteed. By using the dual buffer system, video data retrieved in the current service round can be consumed in the next service round so that the dilivery of video data can be smoothened. In addition, under this retrieval scheme, the sequence of video data blocks retrievals for VCR display is generally not in the same order of their display. Using our proposed buffer scheme, we can rearrange these out of sequence video segments in order before delivery to the users. In Chapter 6, we also determine the buffer requirement for various display modes and the maximum buffer size for each client.

Bibliography

[1] A.D. Gelman, H. Kobrinski, L.S. Smoot, S.B. Weinstein, M.Fortier, and D. Lemay. A store and forward architecture for video on demand service. *proc. IEEE ICC*, pages 27.3.1 – 27.3.5, 1991.

[2] J. Sutherland and L. Litteral. Residential video services. *IEEE Communications Magazine*, pages 36–41, July 1992.

[3] Roger L. Haskin and Frank L. Stein. A system for delivery of interactive television programming. Technical report, IBM Almaden Research Center, 1994.

[4] T.D.C. Little and D.Venkatesh. Prospects for interactive video-on-demand. *IEEE Multimedia*, 1(3):14–24, 1994.

[5] Danial Minoli and Robert Keinath. *Distributed Multimedia Through Broadband Communication Services*, chapter 5. Artech House, 1993.

[6] ITU-T Recommendation H.261. Line transmission of non-telephone signals. *Video codec for audiovisual services at px64 kbits*, March 1993.

[7] G.K. Wallace. The JPEG still-picture compression standard. *Communication of the ACM*, 34(4), April 1991.

[8] ISO/IEC JTC1 CD 11172. Coding of moving pictures and associated audio for digital storage media up to 1.5Mbits/s. *International Organization fro Standardization (ISO)*, 1992.

[9] ISO/IEC JTC1 CD 11172. Generic coding of moving pictures and associated audio. *International Organization fro Standardization (ISO)*, 1994.

[10] Crochiere R.E., Webber S.A., and Flanagan J.L. Digital coding of speech in subbands. *Bell Systems Technical Journals, BSTJ*, 55:1069–1085, 1976.

[11] Woods J.W. and O'Neil S.D. Subband coding of images. *IEEE TRansactions of Acoustics, Speech and Signal Processing*, 34(5):1278–1288, 1986.

[12] David Taubman and Avideh Zakhor. Multirate 3-d subband coding of video. *IEEE Transactions on Image Processing*, 3(5):572–588, September 1994.

[13] Jens-Rainer Ohm. Three-dimensional subband coding with motion compensation. *IEEE Trasaction on Image Processing*, 3(5):559 – 571, September 1994.

[14] Christine I. Podilchuk, Nikil S. Jayant, and Nariman Farvardin. Three-dimensional subband coding of video. *IEEE Trasaction on Image Processing*, 4(2):125 – 139, February 1995.

[15] J. Woods. *Subband Image Coding.* Kluwer Academic. Boston, MA, 1991.

[16] E. Chang and A. Zakhor. Subband video coding based on velocity filters. *IEEE Visual Signal Processing Workshop*, pages 2288–2291, September 1992.

[17] A. Zakhor E. Chang. Scalable video data placement on parallel disk arrays. In *Proceedings of SPIE Symposium on Storage and Retrieval for Image and Video Databases II, San Jose, California*, volume 2185, pages 208–223, 1994.

[18] David Pegler, David Hutchison, Phillip Lougher, and Doug Shepherd. A scalable architecture for multimedia storage. In *High-Speed Networking for Multimedia Applications*, chapter 16, pages 127–263. Kluwer Academic Publishers, 1996.

[19] M.S. Chen and P.S. Yu D. D. Kandlur. Storage and retrieval methods to support fully interactive playout in a disk-array-based video server. In *ACM Multimedia Systems*, volume 3(2), pages 126–135, 1995.

[20] Deepak R. Kenchammana-Hosekote and Jaideep Srivastava. An I/O scheduler for continuous media, part i: Steady state. Technical Report 95-039, University of Minnesota, Computer Science Department, 1995.

[21] D. Rotem and J.L. Zhao. Buffer management for video database systems. In *The* 11^{th} *International Conference on Data Engineering*, March 1995.

[22] T.D.C. Little H.J. Chen, A. Krishnamurthy and D. Venkatesh. A scalable video on demand service for the provision of vcr-like functions. In *Proc. 2nd International Conference on Multimedia Computing Systems, Washington DC.*, May 1995.

[23] Prashant Shenoy and Harrick M. Vir. Efficient support for scan operations in video servers. In *Proceedings of the Third ACM Conference on Multimedia*, 1995.

[24] K. Keeton and R.H. Katz. Evaluating video layout strategies for a high performance storage server. In *ACM Multimedia Systems*, volume 3(2), pages 43–52, 1995.

[25] P. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson. Raid: High performance, reliable secondary storage. Technical Report 93-778, University of California, Berkeley, 1993.