

Video Server Retrieval Scheduling and Resource Reservation for Variable Bit Rate Scalable Video

Seungyup Paek and Shih-Fu Chang

Department of Electrical Engineering, Columbia University
1312 S.W. Mudd Building, New York, NY 10027, USA.
{syp, sfchang}@ee.columbia.edu

Abstract

State of the art digital video compression produces bursty, variable bit rate video. The bursty nature of compressed video raises challenges in the design of video servers. In this paper, we first present a method for the efficient retrieval of bursty video data from the disk system to the memory of a digital video server. For a *single* video stream, the proposed retrieval schedule minimizes the buffer requirement for continuous retrieval, given that a fixed disk bandwidth is reserved for the entire duration of retrieval. Secondly, we present an optimal resource reservation algorithm for *multiple* video streams based on the proposed retrieval schedule. The resource reservation algorithm maximizes the number of bursty video streams that can be supported by a video server, given any disk bandwidth and memory resource. Thirdly, we present a progressive display scheme for scalable video that is based on the retrieval schedule and resource reservation algorithm. Performance evaluations based on simulations using MPEG-2 trace data are presented. For a PC with four disks and a memory resource of 120 MBytes, our approach can support 50%-275% more video streams than previously proposed approaches, depending on the pre-fetch delay that users are willing to tolerate in interactive viewing of videos.

I. Introduction

In recent years, the advent of powerful computing, networking and consumer electronics technologies have made it possible to develop *multimedia networks for the home*. Amongst the myriad of technologies, it is likely that homes of the future will have multimedia enabled Personal Computers (PCs), Digital Versatile Disk (DVD) players, TVs, and digital video camcorders. PCs and consumer electronics devices are expected to be seamlessly connected by digital interconnects such as IEEE 1394, enabling new multimedia applications for the home. IEEE 1394 technology is a high-speed data pipe with speeds starting at 100 Mbps. Current specifications allow for speeds of up to 800 Mbps and speeds beyond 1600 Mbps are being developed.

In a home based multimedia network, PCs can be used as a digital video server. In other words, the PC can be used to store multiple videos and provide interactive viewing of videos to multiple devices concurrently.

To fully utilize the storage resources of the PC, the videos stored on the PCs will be compressed. State of the art digital video compression produces bursty, variable bit rate (VBR) video. Figure 1 shows trace data for MPEG-2 VBR video data. The video sequence is from the movie 'Forrest Gump'. VBR video provides several advantages over constant bit rate (CBR) video, including consistent video quality and lower encoder complexity. However, in general, the bursty nature of compressed video raises challenges in disk storage, disk retrieval and network transmission of video servers.

For a home based multimedia network, based on current technologies, we can expect the main bottleneck for PC based digital video servers to lie in the disk retrieval scheduling of bursty video. This is because current disk I/O bandwidths for PCs are generally lower than relatively high speed digital interconnects such as IEEE 1394, which is expected to be used in home based multimedia networks. Therefore, it is important to develop methods for the efficient retrieval of bursty video data from the disk system to the memory of a PC based digital video server. Video data has to be retrieved from the disk system to memory before transmission into a network.

Most video servers operate in cycles, and during each cycle time, data is retrieved by a disk retrieval scheduler from the disk system to memory for each stream that is supported. For CBR data, the disk retrieval is relatively simple. During each cycle, a constant amount of data is retrieved for each stream. However, for VBR video, it is not clear how data should be retrieved for each stream.

In Constant Time (CT) retrieval of VBR data [5, 12], data corresponding to a constant time is retrieved for each stream during each cycle of operation of a video server. Let v be the video that a stream s is retrieving. For continuous, lossless retrieval, it is necessary to reserve a disk bandwidth equal to the peak data rate of video v for stream s . In this paper, we will show that video servers based on CT retrieval cannot always fully utilize the memory resources of a video server in maximizing the number of supported streams.

In Constant Data (CD) retrieval of VBR data [1, 5], a constant amount of data is retrieved for each stream during each cycle of operation of a video server. A disk bandwidth equal to the average data rate of video v is reserved for stream s . In general, this scheme requires data to be pre-fetched into a buffer before transmission to ensure that buffer starvation does not occur. Buffer starvation can occur because the amount of data transmitted during each cycle is variable for VBR video, while the amount of data retrieved from the disk system is constant during each cycle. In this paper, we will show that video servers based on CD retrieval cannot always fully utilize the bandwidth resources of a video server in maximizing the number of supported video streams.

To overcome the limitations of CD and CT retrieval, we developed a *Minimal Resource* (MR) retrieval schedule and an associated optimal resource reservation algorithm that can fully utilize a given set of video server resources to maximize the number of supported video streams.

In MR retrieval, a range of disk bandwidths can be reserved for the retrieval of VBR data. This is in con-

trast to CD and CT retrieval, in which a disk bandwidth equal to either the average data rate or peak data rate of a VBR video is reserved, respectively. In MR retrieval, as the reserved bandwidth for a stream is reduced, the worst case buffer requirement for a stream increases. A buffer is required since data has to be pre-fetched to ensure continuous, lossless transmission. For a given bandwidth reservation, MR retrieval *minimizes* the worst case buffer memory requirement for continuous, lossless transmission into the network.

MR retrieval leads to a *buffer-bandwidth resource relation* for each VBR video. The resource relation shows the worst case buffer requirement for a given bandwidth reservation. We can show that increasing the reserved bandwidth reduces the worst case pre-fetch delay. Based on this, we introduce the concept of a *Pre-fetch Delay Tolerance Quality of Service* (PDT QoS). A PDT QoS is specified for each stream that a video server supports, and it specifies the worst case pre-fetch delay that can be tolerated during interactive retrieval. It is shown that a PDT QoS specified for a stream s that retrieves video v is equivalent to placing a lower bound on the bandwidth that can be reserved for stream s . The lower bound for the bandwidth can be determined from the buffer-bandwidth resource relation of video v .

For MR retrieval, we developed an optimal resource reservation algorithm for multiple streams. The resource reservation algorithm determines what buffer and bandwidth should be reserved for each stream. The objective of the resource reservation algorithm is to maximize the number of video streams for a given set of video server resources, while ensuring the PDT QoS and retrieval constraints of each stream. It can be shown that the resource reservation algorithm based on MR retrieval fully utilizes the bandwidth resources and maximizes the number of supported video streams.

The MR retrieval scheme essentially defines a spectrum between CD and CT retrieval. The MR retrieval scheme and the optimal resource reservation algorithm make optimum use of the computing resources for a given server configuration.

We present performance evaluations based on simulations using MPEG2 trace data. It is found that the optimal resource reservation algorithm based on MR retrieval improves the performance of video servers compared to resource reservation algorithms based on CT or CD retrieval. For a video server based on a PC with four disks and a memory resource of 120 MBytes, our approach supports at least 50% more video streams than previously proposed approaches. As the PDT QoS is increased, the number of streams supported by a video server is increased.

Compared to simulcast coding, scalable coding schemes can provide multiple levels of video with a minimal cost of extra bandwidth or storage capacity. In scalable video coding, subsets of the full resolution bitstream are used to obtain subsets of the full resolution video. In this paper we present a scheme for the *progressive display* of scalable video for interactive viewing. In this scheme, a progressively increasing PDT QoS is specified for the progressively higher scalable layers of a video. Each scalable layer is consid-

ered as an independent video. This is in contrast to non-progressive display of scalable video in which a single PDT QoS is specified for all the scalable layers. For a video server with a single disk and a memory resource of 120 MBytes, progressive display supports 17% more scalable layer streams than non-progressive display.

This paper is organized as follows. In section 2 we first discuss related research in retrieval scheduling and resource reservation. Section 3 presents our basic assumptions and the video server system model. Section 4 discusses the main differences between CBR and VBR video. Section 5 outlines the retrieval constraints to ensure continuous, lossless retrieval of VBR video in video servers. Section 6 and 7 examines two approaches for retrieval scheduling of VBR video. We discuss the limitations of each approach. Section 8 presents the minimal resource (MR) retrieval schedule for VBR video data. We discuss how this retrieval schedule overcomes the limitations of previously proposed approaches. Section 9 presents a resource reservation algorithm based on MR retrieval that is necessary to maximize the number of video streams that a server can support. Section 10 briefly overviews scalable video and then presents a scheme for the progressive display of scalable video. Progressive display improves the performance of video server supporting scalable video. Section 11 presents performance evaluations of the MR retrieval schedule in comparison to other proposed approaches.

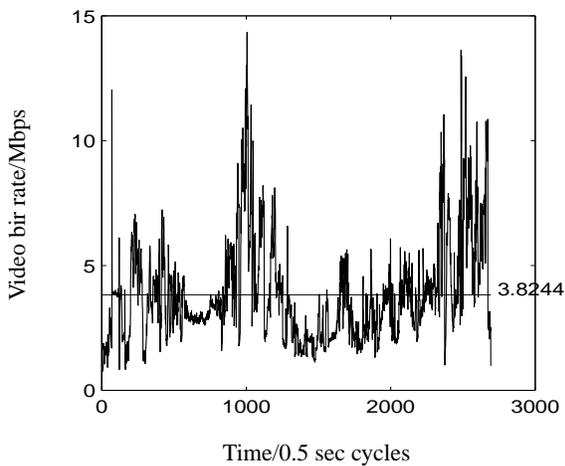


Fig. 1. MPEG-2 VBR video trace.

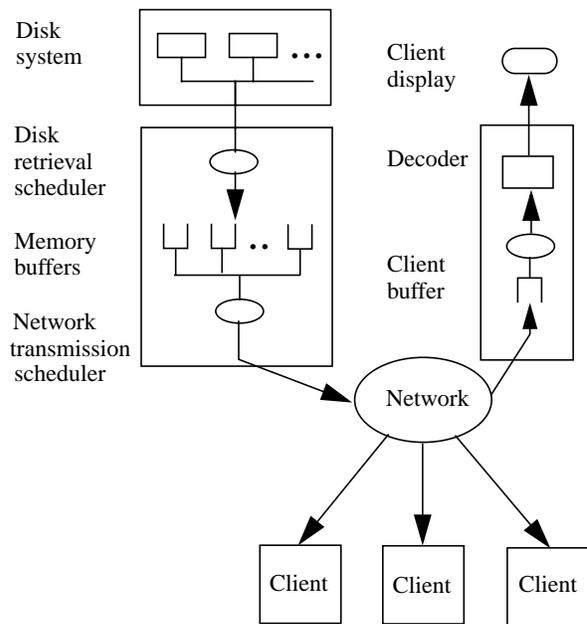


Fig. 2. Video server system model.

II. Related Work

In [12], CT retrieval is the basis for the admission control scheme in multimedia servers. The primary contribution of the work in [12] is that *statistical service guarantees* are provided to all streams. In other words, for each stream, a continuous retrieval is guaranteed to a fixed percentage of the video data. It is proposed that a certain percentage of video data can have the continuity requirement violated without significantly affecting the quality of the video. This leads to an improvement in the utilization of the server. New clients are admitted for service as long as the statistical estimate of the aggregate data rate requirement (rather than the peak data rate requirement) can be met. The approach of ensuring statistical guarantees, rather than deterministic guarantees is certainly effective in increasing the performance of a multimedia server. However, the approach proposed in [12] is based on CT retrieval and has the drawback that memory resources cannot be fully utilized to maximize the number of supported streams.

In [5], CD retrieval is the basis for both a deterministic and statistical admission control scheme. The retrieval scheme is referred to as Constant Data Length (CDL) retrieval. In [1], two retrieval schemes called traditional CDL and generalized CDL (GCDL) are presented. In the traditional CDL scheme of [1], a constant amount of data is retrieved from the disk for a video stream in the first disk cycle. In the second cycle, the same constant amount of data is retrieved only if it is required to prevent buffer underflow. Otherwise, no data is retrieved. This process repeats throughout the retrieval. Therefore, each retrieval cycle is either an idle or active round. Although a constant data amount is retrieved during an active round, the overall retrieval can be variable bit rate. This is different from [5] in which the overall retrieval is constant bit rate i.e. there are no idle rounds. In [1], the GCDL scheme is an extension of the traditional CDL scheme in which the retrieval round can be different for different video streams and which are a multiple of the disk cycle. This is shown to reduce the buffer requirements compared to traditional CDL.

In our video server model, we have assumed a cycle based operation of the video server in which the disk retrieval scheduler retrieves data for multiple video streams from the disk system to the memory during each cycle. It is important to note that not all video servers operate on a cyclic basis. For example, [14] proposes a deadline based scheme.

In the following sections, it will be seen that our approach performs *smoothing* in retrieving video data from disks to memory. There have been several researchers who have also proposed smoothing techniques for bursty pre-recorded video data [15, 16]. Our approach differs because we are also concerned with the retrieval of *multiple* video streams across a common channel (disk to memory channel), using a common memory resource (server memory). The techniques proposed in [15, 16] deal with smoothing techniques of a single video stream from the server to the client across a high performance network, utilizing the client buffer for smoothing. Therefore, our work considers the dynamics between multiple video streams, whereas the other approaches are only concerned with per-stream dynamics.

III. System Model

In this section we describe the video server system model relevant to our research. The system model is shown in figure 2. The video server has a fault tolerant disk array for the the storage of video data, a memory resource, and a network interface for transmission into the computer network.

The disk retrieval scheduler has a cyclic operation [4, 8, 9, 10]. Each cycle, the disk retrieval scheduler retrieves data for multiple video streams from the disk system to the memory. The network stream scheduler also has a cyclic operation, although the cycle time of the network stream scheduler will typically be much smaller than the cycle time of the disk retrieval scheduler. The network stream scheduler transmits data for multiple video streams into the network during each cycle.

The video data is interleaved over all the disks of the array. Such a distribution technique is referred to as “wide striping” or “fine grained striping”. It has been shown that this results in buffer requirements that increase with the number of disks in the array. Therefore, using fine grained striping can lead to high buffer requirements for large scale commercial video servers. However, the focus of this paper is on small scale, PC based digital video servers for the home.

During each disk cycle, the disk head completes one cycle of a SCAN disk head schedule. The disk head scans the disk starting from the inner most track to the outer most track [2, 6]. While scanning the disk, data blocks belonging to different streams are read from the disk. Upon reaching the outer most track, the head is returned back to its initial position. The performance analysis is given in [2]. Although multiple blocks for each video stream are being retrieved from the disks each cycle, the overhead of disk head seeks is greatly reduced by using the SCAN disk head scheduling algorithm.

There is no special data layout of the videos on the disks. There has been extensive research that addresses the layout of video data on disks. This paper does not address how data are actually laid out on disk. Instead, we have assumed that each disk operates with the SCAN disk head schedule, which does not assume any special data layout on disk. We have done detailed analysis, performance evaluation and simulation in [2].

The network can accommodate the peak bandwidth of all video streams and introduces zero delay and zero jitter. The limitations of the network are not considered. In future research we will consider the limitations of the network, and we will also consider the interaction between the disk retrieval scheduler and network stream schedulers. In particular, we will consider the interaction of our disk retrieval scheduler with network stream scheduling algorithms to smooth VBR video data for transmission in networks.

The video server supports completely interactive viewing and continuous, lossless retrieval. Video servers supporting completely interactive viewing allow viewers to pause and resume playback at any time during a viewing session. Playback can also resume at any point of a video. Video servers supporting continuous, lossless retrieval provision resources so that once playback of a portion of a video has started, no delay is

introduced and no data is lost. Note that this does not mean that there can be no delay *before* playback begins e.g. after a pause.

The video server has a fixed bandwidth and fixed buffer reserved for the entire duration of interactive retrieval. There are no renegotiations of resources for each video stream during a viewing session. This is a key simplifying assumption in our research. There is no renegotiation of resources for the retrieval during a viewing session.

IV. CBR and VBR Compressed Digital Video

In our research we used MPEG2 compressed digital video. However the retrieval schedule and the resource reservation algorithm are directly applicable to any video codec that results in VBR video. The variable bit rate of MPEG2 video is dependent on the encoding structure of the MPEG2 coding algorithm. In MPEG2 digital video, compression is achieved by the combination of techniques such as the discrete cosine transformation (DCT), variable length codes, quantization of DCT coefficients, motion estimation and motion compensated inter-frame prediction. MPEG2 has a buffer control mechanism in which the quantization parameter can be varied adaptively in order to achieve a constant average bit rate of the compressed video. The disadvantage of this mechanism is that the subjective visual quality will be variable, since the quantization parameter is continually varied. An alternative is to maintain a constant quantization parameter during the encoding of video. This results in variable bit rate video, in which the amount of data to represent different time scales of video (macroblock, slice, frame, group of pictures etc.) are variable. The research presented in this paper maximizes the number of VBR streams that can be supported by a video server.

V. Time Constraints

In the following sections, although the operation of a video server is based on discrete time cycles, we will use continuous time notation to clearly convey the core ideas. Table 1 defines the notation we will use in the following sections.

Table 1: Notation

| | |
|--------|--|
| $r(t)$ | Data rate of stored video |
| t_n | Start of network transmission |
| t_d | Start of disk retrieval |
| T | Time duration of video |
| $o(t)$ | Total data output from memory into network |

Table 1: Notation

| | |
|--------|--|
| $i(t)$ | Total data input from disk into memory |
| m | Buffer memory reserved for retrieval |
| b | Disk bandwidth reserved for retrieval |

The retrieval constraints for the retrieval of a video are as follows:

- 1) $i(t) \geq o(t)$ (continuous retrieval constraint)
- 2) $i(t) - o(t) \leq m$ (Buffer constraint)
- 3) $\frac{di(t)}{dt} \leq b$ (Disk bandwidth constraint)

In the following, we will assume that $o(t) = \int r(t - t_s)$. In other words, the data is output into the network according to the data rate of the stored video, $r(t)$, starting at time t_n .

VI. Constant Time Retrieval

Constant time (CT) retrieval is described by the equation $i(t) = o(t)$. The accumulated input data and accumulated output data are equivalent. In an actual system, the network stream scheduler waits one cycle time t_c after disk retrieval begins before it can start transmission into the network.

The delay of one cycle exists because we assume that the video server uses a double buffer scheme for retrieval and transmission (figure 3). During each cycle, data is read from the disk into the disk read buffers for each stream. Each disk executes one SCAN cycle. Concurrently, data is written from the network write buffers into the network for each stream. At the end of each cycle, the contents of the disk read buffers are written onto the network write buffers, and the process repeats. No pre-fetch buffer is required for CT retrieval. In the first cycle, the disk read buffer is being filled, while the network write buffer is empty.

As mentioned earlier, network transmission can only begin after a delay of one cycle. This delay is different from the pre-fetch delay mentioned in the following sections and is common to all the retrieval schedules. We shall ignore this delay in the following sections. The pre-fetch delay for CT retrieval is zero.

For each stream, a disk bandwidth of b equal to the peak data rate of the video being retrieved must be reserved for the entire duration of an interactive viewing session:

$$b = \max \{r(t), 0 \leq t \leq T\}.$$

In the following sections, we ignore the disk read buffers and network write buffers and only consider the pre-fetch buffers.

The major disadvantage of CT retrieval is that it cannot take advantage of memory resources to maximize

the number of streams supported by a video server. Consider a video server that uses CT retrieval. Each stream has a bandwidth reservation equal to the peak data rate of the video being retrieved. Assume that the total bandwidth reserved for all streams is equal to the total bandwidth of the video server. Assume that a large buffer memory exists in the video server. Using CT retrieval, no more streams can be supported by the video server due to the bandwidth limitation. It will be shown that if all viewers can tolerate a pre-fetch delay, the bandwidth reserved for all the streams can be substantially reduced from the peak bandwidth. Reducing the reserved bandwidth for each stream will be shown to require an increase in pre-fetch buffer requirements for each stream, if continuous, lossless retrieval is to be guaranteed. In this way, memory resources can be utilized to alleviate the I/O bandwidth bottleneck. By reducing the total bandwidth reserved for all the streams, the video server can potentially increase the number of streams that are supported. CT retrieval does not take advantage of viewers that can tolerate a pre-fetch delay and always assigns a pre-fetch delay of zero to all streams.

For the MPEG-2 VBR video shown in figure 1, CT retrieval has to reserve a bandwidth of 14 Mbps for the entire duration of retrieval. The buffer requirement for a video server operating at 0.5sec cycle time is 1.75 MB.

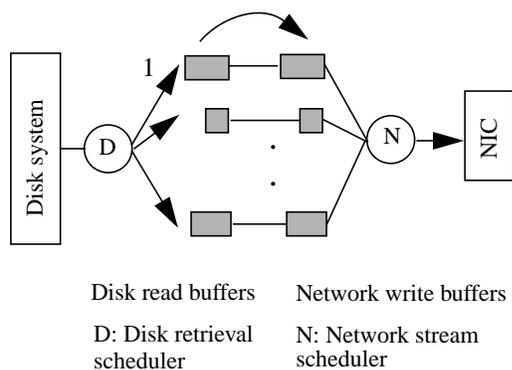


Fig. 3. Buffers for CT retrieval.

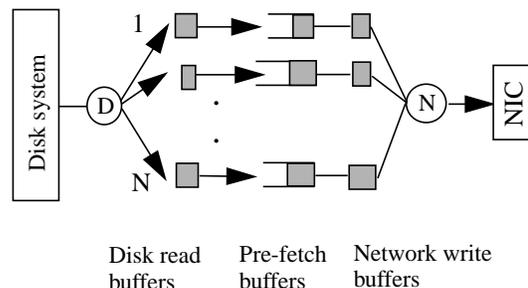


Fig. 4. Buffers for CD retrieval.

VII. Constant Data Retrieval

In constant data (CD) retrieval, a fixed bandwidth of b Mbps is reserved for each stream. The reserved bandwidth is typically much smaller than in CT retrieval, in which the peak bandwidth is reserved. CD retrieval retrieves a fixed amount of data $b \cdot t_c$ during each cycle until all the data has been transmitted (figure 4, 5). This differs from CT retrieval, in which a variable amount of data is retrieved in each cycle. In this scheme, data has to be pre-fetched to ensure that continuous, lossless retrieval of the video is guaranteed. The continuous, lossless retrieval constraint is violated if buffer starvation occurs. This can occur because the amount of data transmitted during each cycle is variable, while the amount of data retrieved

from the disk system is constant during each time cycle. Since a pre-fetch data has to be retrieved, there is a pre-fetch delay associated with CD retrieval. The worst case pre-fetch delay $t_n - t_d$ can be determined for stored video because the entire trace is known a-priori:

$$t_n - t_d = \frac{p}{b},$$

$$p = \text{MAX}\{-(b \cdot \Delta - o(t_n + \Delta))\} \quad 0 \leq \Delta \leq T$$

The major disadvantage of CD retrieval is that it cannot fully utilize video server resources to maximize the number of streams. Consider a video server that uses CD retrieval. Each stream has a pre-fetch buffer requirement. Assume that the total buffer memory reserved for all streams is equal to the total memory resource of the video server. However, assume that the total bandwidth reserved for all streams is less than the total disk bandwidth of the video server. Using CD retrieval, no more streams can be supported by the video server because of the memory limitation. In the next section, it will be shown that by increasing the bandwidth reserved for each stream, the memory requirement for each stream can be substantially reduced. In this way, the bandwidth resources can be utilized to alleviate the memory bottleneck of a video server. By reducing the total memory reserved for all the streams, the video server can potentially support more streams.

CD retrieval continues to retrieve a constant amount of data until all data has been retrieved. Data is sometimes retrieved earlier than is required, leading to potentially unnecessarily large buffer requirements. For the MPEG-2 VBR video shown in figure 1, CD retrieval reserves a bandwidth of 3.8 Mbps for the entire duration of retrieval. The buffer requirement for a video server operating at 0.5sec cycle time is found to be 50 MB.

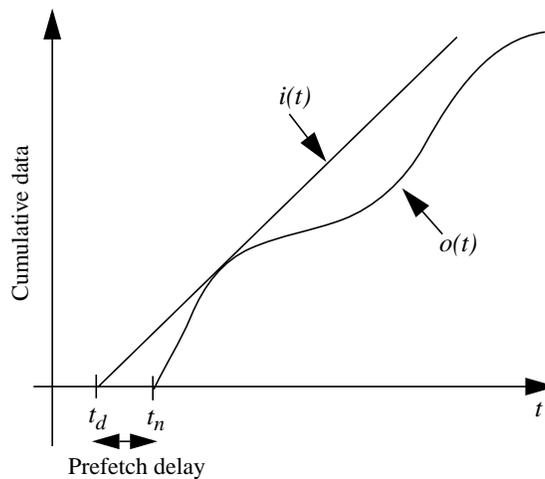


Fig. 5. Cumulative data for CD retrieval.

VIII. Minimal Resource Retrieval

In this section, we present a new Minimal Resource (MR) retrieval schedule for continuous, lossless retrieval of VBR video. MR retrieval is similar to CD and CT retrieval because it alternates between intervals of CT and CD retrieval. However, it differs in that a *range* of bandwidths can be reserved for the retrieval. CD retrieval requires that a bandwidth equal to the average data rate is reserved, while CT retrieval requires a bandwidth reservation equal to the peak data rate.

If the bandwidth reserved for retrieval of VBR video is less than the peak data rate, then data has to be pre-fetched to ensure continuous, lossless retrieval. Therefore, a buffer for pre-fetch data is required. In order to minimize the buffer requirement, data should be pre-fetched *just-in-time*. For the retrieval of VBR video, MR retrieval minimizes the worst case buffer requirement that is required for a given disk bandwidth reservation. It will be shown that this property of MR retrieval fully utilizes the resources of a video server.

We first describe the MR retrieval schedule and then discuss its properties. As before, let t_n be the start of transmission of a stream into the network. The accumulated data output from the memory to network is $o(t)$. $o(t) = 0, t \leq t_n$. The bandwidth reserved for the retrieval is b . We define a function $a(\alpha, \beta) = b \cdot (t - \alpha) + \beta$ that we will use in describing MR retrieval. We also use two new variables t_b, t_e to mark the beginning and end of each *maximum retrieval* interval. MR retrieval of stored video is described in table 2 and is shown graphically in figure 6.

Table 2: MR retrieval

| |
|--|
| 1. Set $t_e = t_n + T$ |
| 2. Decrease t_e until $\left. \frac{do(t)}{dt} \right _{t_e} \geq b$ |
| 3. Find the intersection of $a(t_e, o(t_e))$ with $o(t)$ |
| 4. Let $t_b < t_e$ equal the intersection point |
| 5. Mark the interval $[t_b, t_e]$ as a <i>maximum retrieval</i> interval |
| 6. If $t_b \leq t_n$ then stop |
| 7. Set $t_e = t_b$ and return to step 2 |

MR retrieval is defined by the maximum retrieval intervals. Figure 7 shows $i(t)$ for MR retrieval. If the time t falls inside any of the maximum retrieval intervals, the retrieval rate is at the maximum bandwidth

b. Otherwise, the retrieval amount is equal to the data rate. The buffer status at time t is $m(t) = i(t) - o(t)$.

The worst case pre-fetch delay is:

$$d = \frac{\text{MAX}\{m(t)\}}{b}, t_n \leq t \leq t_n + T.$$

Figure 9 shows MR retrieval of the MPEG-2 VBR video shown in figure 1.

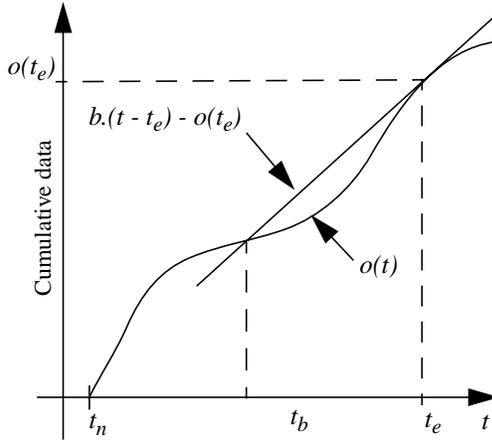


Fig. 6. Determination of MR retrieval.

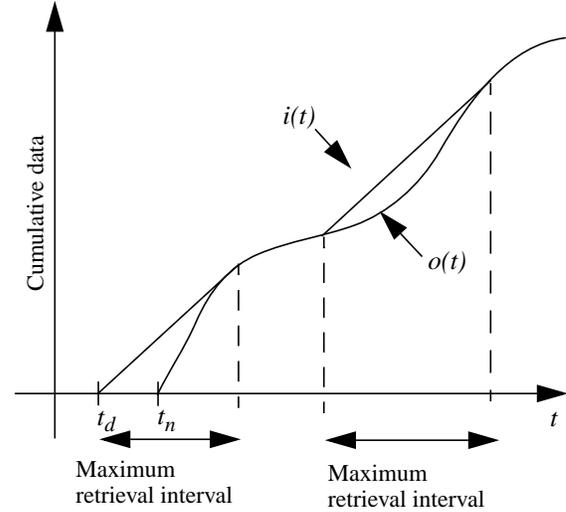


Fig. 7. Cumulative data for MR retrieval.

Proof of MR retrieval optimality

MR retrieval can be shown to minimize the worst case pre-fetch buffer requirement for a given bandwidth reservation for the continuous, lossless retrieval of a video. We can prove this by showing that MR retrieval is based on *just-in-time* retrieval. Consider the first maximum retrieval interval $[t_b, t_e]$. In MR retrieval, the retrieval rate during a maximum retrieval interval is b . We define a small time interval Δ . Consider the start of retrieval to be delayed to $t_b + \Delta$. In this case, it can be seen that even if the retrieval is at the maximum retrieval rate of b , the continuous retrieval constraint will be violated (figure 8). Therefore, t_b is the latest time at which pre-fetch of data can start if continuous retrieval is to be guaranteed up to t_e . Consider the start of retrieval to start earlier at $t_b - \Delta$. It can be seen that the buffer requirement will increase for any possible retrieval schedule as data is retrieved earlier than required. This analysis can be done iteratively for all the maximum retrieval intervals. Therefore, since MR retrieval is based on *just-in-time* retrieval, it minimizes the buffer requirement while satisfying the constraints for continuous, lossless retrieval.

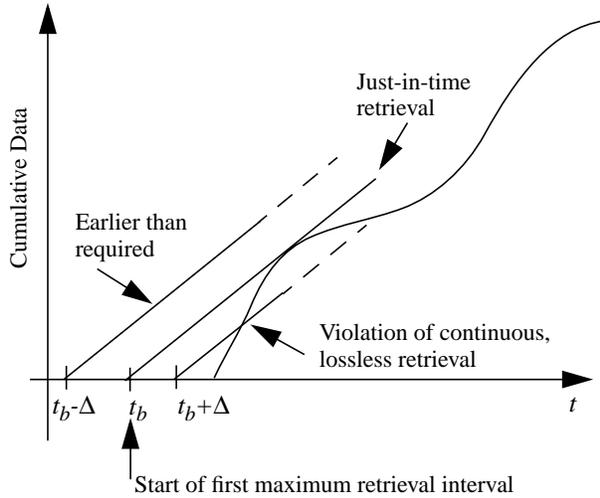


Fig. 8. MR retrieval optimality.

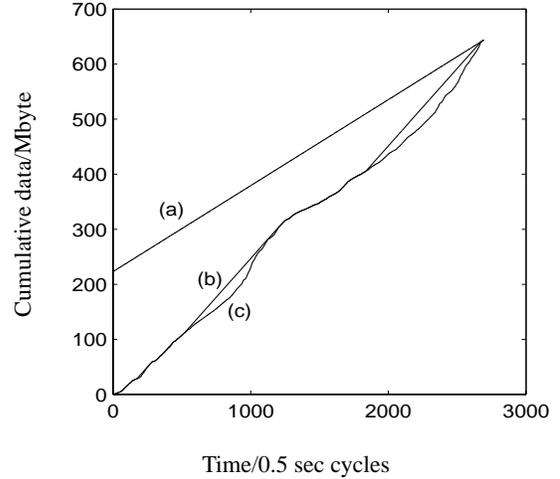


Fig. 9. MR retrieval of MPEG-2 VBR video.

- (a) $i(t)$ for MR retrieval with reserved bandwidth of 2.5 Mbps
 (b) $i(t)$ for MR retrieval with reserved bandwidth of 4.5 Mbps
 (c) $o(t)$

Buffer-bandwidth resource relation of MR retrieval

For a given bandwidth reservation, MR retrieval minimizes the worst case buffer that is necessary for continuous, lossless retrieval. It can also be shown that increasing the reserved disk bandwidth will reduce the minimized worst case buffer requirement. Therefore, MR retrieval leads to a buffer-bandwidth resource relation for the retrieval of a video. Figure 10 shows the buffer-bandwidth resource relation for the MPEG2 encoded video trace data shown in figure 1. From this relation, the corresponding worst case pre-fetch delay can be found. If the worst case buffer requirement for the retrieval of a given video is m , then the worst case pre-fetch delay is m/b , where b is the corresponding reserved bandwidth.

For the interactive viewing of videos, we introduce the concept of a *Pre-fetch Delay Tolerance Quality of Service* (PDT QoS). A PDT QoS is specified for each stream that a video server supports, and it specifies the worst case pre-fetch delay that can be tolerated during interactive viewing. It is shown that a PDT QoS specified for a stream s that retrieves video $v(s)$ is equivalent to placing a lower bound on the bandwidth that can be reserved for stream s . The lower bound for the bandwidth can be determined from the buffer-bandwidth resource relation of video $v(s)$.

The primary strength of MR retrieval is the flexibility to optimally trade bandwidth and buffer. This is captured in the buffer-bandwidth resource relation. While CD and CT retrieval are each represented by a single point on the resource relation for the stored video, MR retrieval can operate at multiple operating points on the resource relation. MR retrieval can set any bandwidth reservation. As the bandwidth reservation is

reduced, it is necessary to increase the reserved buffer. Consider a video server that uses MR retrieval. We present two cases to demonstrate the advantage of using MR retrieval over CD or CT retrieval.

Case 1. Assume that initially, each stream has a bandwidth reservation equal to the peak data rate of the video being retrieved. Assume that the total bandwidth reserved for all streams is equal to the total bandwidth of the video server. Assume that a large buffer memory exists in the video server. Using CT retrieval, no more streams can be supported by the video server because of the bandwidth limitation. In MR retrieval, if all viewers can tolerate a pre-fetch delay, the bandwidth reserved for all the streams can be substantially reduced from the peak bandwidth. Reducing the reserved bandwidth for each stream requires an increase in pre-fetch buffer requirements for each stream, if continuous, lossless retrieval is to be guaranteed. In this way, memory resources can be utilized to alleviate the I/O bandwidth bottleneck. By reducing the total bandwidth reserved for all the streams, the video server can potentially increase the number of streams that are supported.

Case 2. Assume that initially each stream has a bandwidth reservation equal to the average data rate of the video being retrieved. Each stream has a pre-fetch buffer requirement. Assume that the total buffer memory reserved for all streams is equal to the total memory resource of the video server. However, assume that total bandwidth reserved for all streams is less than the total disk bandwidth of the video server. Using CD retrieval, no more streams can be supported by the video server because of the memory limitation. In MR retrieval, by increasing the bandwidth reserved for each stream, the memory requirement for each stream can be substantially reduced. In this way, the bandwidth resources can be utilized to alleviate the memory bottleneck of a video server. By reducing the total memory reserved for all the streams, the video server can potentially support more streams.

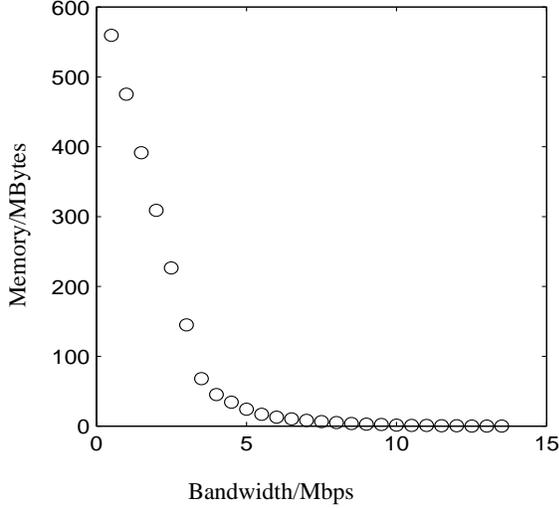


Fig. 10. Resource relation of MPEG-2 VBR video.

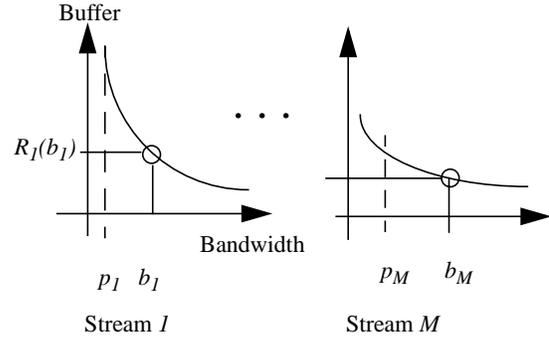


Fig. 11. Resource reservation problem.

IX. Resource Reservation for MR Retrieval

In the previous section we developed and presented the MR retrieval schedule. It was found that the worst case memory buffer requirement for the retrieval of a VBR video decreases as the bandwidth reservation increases. It was seen that MR leads to a buffer-bandwidth resource relation. In this section, we develop the resource reservation algorithm based on MR retrieval for multiple streams in a video server.

In a video server, data for *multiple, concurrent* streams is retrieved from the disk system to memory and then transmitted into the network. The video server has resources of disk bandwidth and memory that have to be shared amongst all streams. If MR is used for the retrieval of each stream, the important question remains as to what buffer and bandwidth reservations should be made for each stream i.e. the operating point on the resource relation of the video retrieved by each stream must be determined. For each incoming stream, the reservations should be made to maximize the number of streams that can be supported by a video server while guaranteeing the continuous, lossless retrieval and PDT QoS of each stream. Before describing the reservation problem, we present some definitions in table 3.

Table 3: Notation for resource reservation

| | |
|--|------------------------------|
| M | Total number of streams |
| $k \in \{1, \dots, M\}$ | Stream index |
| B_I | Bandwidth increment |
| $b_k \cdot B_I, b_k \in \{1, 2, \dots\}$ | Stream bandwidth reservation |

Table 3: Notation for resource reservation

| | |
|--|--|
| $b_k^o \cdot B_I, b_k^o \in \{1, 2, \dots\}$ | Optimal stream bandwidth reservation |
| p_k | Lower bound for stream bandwidth reservation |
| q_k | Peak bandwidth of video accessed by stream k |
| $R_k(b_k)$ | Stream buffer-bandwidth resource relation |
| d_k | Stream PDT QoS |
| C_M | System memory resource constraint |
| C_B | System disk bandwidth resource constraint |
| $\vec{B} = (b_1, \dots, b_M)$ | Reservation vector for all streams |
| $\vec{B}^o = (b_1^o, \dots, b_M^o)$ | Optimal reservation vector for all streams |
| S | Reservation vectors with $p_k \leq b_k, k = 1, \dots, M$ |

Note that $R_k(b_k)$ is determined for a video stream by the MR retrieval schedule. In other words, for a given bandwidth b_k , we find the worst case buffer requirement $R_k(b_k)$ for the video stream based on the MR retrieval schedule.

The lower bound on the stream bandwidth reservation is determined as follows:

$$p_k = \text{MIN} \{ b_k \}, \text{ subject to } \frac{R_k(b_k)}{b_k} \leq d_k,$$

where $\frac{R_k(b_k)}{b_k}$ is the pre-fetch delay.

Problem formulation

Our objective in a video server is to maximize the number of streams that can be supported. Therefore, we formulate the resource reservation problem as follows:

For a given set of streams, determine if there exists a reservation vector \vec{B} for all streams that satisfies the following constraints:

$$M_T = \sum_{k=1}^M R_k(b_k) \leq C_M \text{ (memory constraint),}$$

$$B_T = \sum_{k=1}^M b_k \leq C_B \text{ (bandwidth constraint),}$$

$$\vec{B} \in S \text{ (PDT QoS constraint).}$$

If the reservation vector exists, the set of streams can be supported by a video server, otherwise the set of streams cannot be supported. Note that in an actual system, the computation of the total bandwidth constraint is not a simple summation as shown above, since the SCAN disk head schedule is assumed. The actual computation is an approximation based on simplifying assumptions [2, 9].

In the SCAN scheduling algorithm, the scanning cycle consists of two phases. During the first cycle, the head scans the disk from the inner most track to the outer most track. While scanning the disk, data blocks belonging to different streams are read from the disk. Upon reaching the outer most track, the head is returned to the initial position. Several assumptions are made in the above analysis. Firstly, any stream accessed during the first phase will add to the total retrieval cycle the maximum rotational latency, the data reading time and the minimum seek time. Secondly, since the retrieval cycle consists of two phases of head movement, we add two maximum seek delays to the total cycle time. For the SCAN scheduling algorithm, we determine if the bandwidth constraint is satisfied by calculating the total cycle time T required for all the bandwidth reservations, as follows:

$$T = \sum_{k=1}^M \left(\frac{t_c \cdot b_k}{n_d \cdot r_d} + t_{rx} + t_{sm} \right) + 2 \cdot t_{sx}$$

t_c is the round robin cycle time, t_{sx} is maximum seek time, t_{sm} is minimum seek time and t_{rx} is maximum rotation latency. r_d is maximum disk transfer rate and n_d is the total number of disks in an array. For the bandwidth constraint, we require that $T \leq t_c$. This is equivalent to the constraint $B_T \leq C_B$.

The resource reservation problem above can be shown to be equivalent to the following two step algorithm:

1) Find the reservation vector \vec{B}^o which is the solution to the following constrained minimization problem:

$$\text{MIN } M_T = \sum_{k=1}^M R_k(b_k) \text{ subject to } B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

2) If $M_T \leq C_M$ (system memory constraint), the set of all streams can be supported by the system, otherwise the set of streams cannot be supported.

Let A be the set of *all reservation vectors* that meet both the PDT QoS constraints $\{d_k, k = 1, \dots, M\}$ and the system bandwidth constraint C_B specified above. The non-linear minimization problem gives us \vec{B}^o , which is the reservation vector in A that minimizes the total system memory requirement M_T . If M_T is greater than the system memory constraint, then there can be no reservation vector in A that will also meet

the memory resource constraint C_M . This is because the reservation vector \vec{B}^o is the vector in A that *minimizes* the memory requirement. This means that there can be no reservation vector that meets all system resource constraints C_M , C_B , and the PDT QoS constraint. Therefore the two step algorithm is equivalent to the optimal resource reservation algorithm.

Optimal resource reservation algorithm

In this section we outline the main approach we used in finding an optimal resource reservation algorithm. The algorithmic solution to the optimal resource reservation problem is based on a version of the Lagrange multiplier method as applied to bit allocation problems in video coding [11]. The approach places no restriction on the form of the stream resource relations e.g. the resource relations need not be strictly convex. In this section, we only present the main theorem associated with this method. In appendix A, we present the development of the optimal algorithm for completeness.

As part of the two step resource reservation algorithm we have to solve the following *constrained* minimization problem:

$$MIN M_T = \sum_{k=1}^M R_k(b_k), \text{ subject to } B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

The solution to this constrained minimization problem is based on the following theorem:

Theorem: For any $\lambda \geq 0$, the solution $\vec{B}^o(\lambda)$ to the *unconstrained* minimization problem

$$MIN \left\{ \sum_{k=1}^M R_k(b_k) + \lambda \sum_{k=1}^M b_k \right\}, \text{ subject to } \vec{B} \in S$$

is also the solution to the *constrained* minimization problem with the constraint

$$C_B^o(\lambda) = \sum_{k=1}^M b_k^o(\lambda).$$

The proof of this theorem is presented in [11]. This theorem does not guarantee a solution to the constrained minimization problem. It just says that for every λ , there is a corresponding constrained problem for which the solution is the same as that of the unconstrained problem. The main point is that if $C_B^o(\lambda)$ happens to equal C_B , then $\vec{B}^o(\lambda)$ is the desired solution to the original constrained problem.

In solving the constrained problem, we can sweep over λ from 0 to ∞ and try to find a corresponding $C_B^o(\lambda)$ that equals C_B . A typical dependency of the solution rate $C_B^o(\lambda)$ on the lagrange multiplier λ can be shown to have the form of figure 12. It shows a decreasing staircase curve where the discontinuities correspond to singular values of λ .

The key question then becomes the problem of finding the corresponding λ efficiently. This forms the main portion of the optimal resource reservation algorithm. The development of the algorithm and the optimal resource reservation algorithm is presented in appendix A.

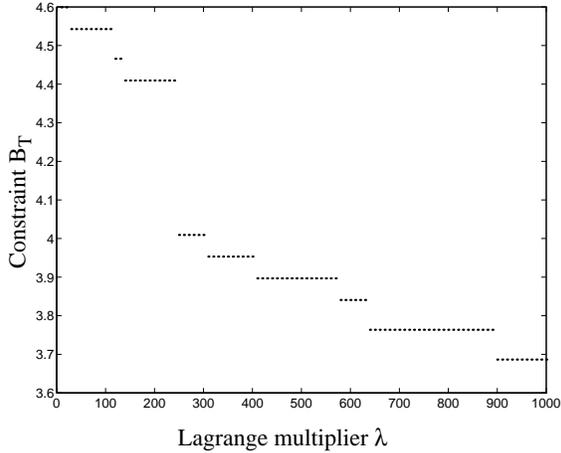


Fig. 12. Variation in constraint function with λ .

Heuristic resource reservation algorithm

The optimal resource reservation algorithm presented above gives us an upper bound on the performance of the video server architecture proposed in this paper. In [10] we presented a low complexity heuristic resource reservation algorithm. The algorithm is presented here, and in section 11, we compare the performance of this algorithm with the optimal resource reservation algorithm presented above. We found by simulation that the performance of this algorithm typically performs within 10% of the optimal resource reservation algorithm, although it is much lower in complexity than the optimal algorithm. Although, the optimal algorithm only performs typically 10% better than the heuristic algorithm, it was important to develop the optimal algorithm for two reasons. Firstly, solving the optimal resource reservation problem gave us an in-depth understanding of the resource reservation process. Secondly, the optimal algorithm servers as a benchmark to compare the performance of any heuristic algorithm.

- 1) For all video streams, the retrieval bandwidths \hat{B} are initially set to the peak data rate of the videos.
- 2) Compute total memory and bandwidth requirement:

$$M_T = \sum_{k=1}^M R_k(b_k), B_T = \sum_{k=1}^M b_k$$

- 3) If $M_T > C_M$: the video server *cannot* support all the video streams specified (memory limited). Stop the algorithm.

4) If $M_T < C_M$:

If $B_T \leq C_B$: The video server *can* support all the video streams specified. Stop the algorithm.

If $B_T > C_B$: Reduce total bandwidth by increasing buffer requirements (step 5).

5) Reduce total bandwidth.

First find the video stream for which there is a minimal increase in buffer with an incremental decrease in bandwidth. Define g_k as the incremental increase in buffer of each video stream k with an incremental decrease in bandwidth of the bandwidth reservation for the video stream.

$$g_k = R_k(b_k - 1) - R_k(b_k) , G = \{k; p_k < b_k\}$$

G defines the set of all streams for which the bandwidth reservation can be incrementally decreased. We find the video stream k_{min} for which the above is minimized i.e.

$$g_{k_{min}} = \min \{g_k\} , k \in G$$

If the set G is null, then the set of video streams cannot be accommodated by the video server. Stop the algorithm.

If a minimum is found, step 2 is repeated with the following resource reservation:

$$\vec{B} = (b_1, \dots, b_{k_{min}} - 1, \dots, b_M)$$

X. Scalable Video

In this section we integrate scalable video with MR retrieval and the resource reservation algorithm to present a scheme for the *progressive display* of scalable video. Progressive display improves the performance of a video server supporting interactive viewing of video.

Compared to simulcast coding, scalable coding schemes can provide multiple levels of video with a minimal cost of extra bandwidth or storage capacity. In scalable video coding, subsets of the full resolution bit-stream are used to obtain subsets of the full resolution video [4, 7, 9]. Scalable video will be used in advanced computer networks to support heterogeneous clients. Mobile wireless clients may only have computing resources to receive the lowest layer of video, while high performance workstations will request all the scalable layers of video.

The MPEG2 standard allows a combination of *spatial*, *SNR (signal-to-noise ratio)* and *temporal scalability* for up to three layer coding of video sequences. In one possible hybrid, three layer scalable coding scheme, the base layer provides the initial resolution of video. The spatial enhancement layer enables the upsampling and hence increase in frame size of the base layer. Finally, the SNR enhancement layer increases the visual quality of the (base+spatial enhancement) layers of video [7]. In another scheme for MPEG-2 video, three layer temporally scalable video is achieved as follows. The lowest scalable layer is

comprised of the I frames of a video (I layer). The P frames (P layer) enable an increase in the temporal resolution of the I frame layer. Finally, the B frames (B frames) increase the temporal resolution of the I+P layers. We refer to this as IPB scalable video. In this scheme, scalability is inherently provided by the MPEG-2 encoding structure. We have developed and tested a simple but robust system to extract the I, P, B layers from an MPEG-2 sequence and also to combine the multiple layers (I, I+P, I+P+B) for decoding. There are various ways for scalable video data to be placed on disks [4, 9]. In this research, we assumed that each scalable layer is stored separately as an independent ‘video’ and that each layer is interleaved over all disks.

In the progressive display of scalable video for interactive viewing, a progressively increasing PDT QoS is specified for the progressively higher scalable layers of a video. Each scalable layer is considered as an independent video. This is in contrast to non-progressive display of scalable video in which a single PDT QoS is specified.

In progressive display, the pre-fetch data for progressively higher layers are retrieved simultaneously since each layer is considered to be an independent video. At any given time, a video is transmitted only with all the scalable layers for which the pre-fetch data have been fully retrieved. For example, suppose that transmission is to be resumed at some point of a video after an interactive function. If enough time has elapsed only for the pre-fetch data of the lowest scalable layer to have been retrieved, only the lowest scalable layer is transmitted. If enough time has elapsed for the pre-fetch data of the first two layers to have been retrieved, the first two scalable layers are transmitted.

Progressive display improves the performance of the video server supporting scalable video. For scalable video, let the lowest layer of video have a PDT QoS of l sec. The higher layers will have PDT QoS values larger than l sec. In non-progressive display of scalable video, to achieve the same degree of interactivity, all the scalable layers have the same PDT QoS value of l sec. We will demonstrate the performance improvement in the next section.

XI. Performance Evaluation

For performance evaluation, we used the disk performance characteristics of three disk systems as shown in table 4. Disk system 1 has the disk performance characteristics of a current magnetic disk. In disk system 2, the performance parameters were improved by a factor of two to project the performance characteristics of the next generation of magnetic disk systems. For performance evaluation, trace data for MPEG2 scalable and non-scalable video was obtained using Columbia’s full-profile, standard-conforming MPEG2 software encoder/decoder [13].

In the following simulations, the video server receives requests for videos from clients. Each new request specifies a certain video which is stored on the video server, for which there exists a resource relation. Each

request also has an associated PDT QoS. For each new request, the video server determines if it can accept the request or not. If the video server can accept the new request, a stream is established for the new request. For each simulation the total number of video streams that can be supported by the video server is found for a given set of available video server resources. The simulations find the total number of admissible streams to the video server system as the on-board memory resource is increased, while maintaining a fixed disk bandwidth resource.

Performance comparison of MR and CT retrieval

Figure 13 and 14 compares the performance of MR and CT retrieval scheduling. Each line corresponds to a single simulation. In each simulation, all streams are accessing the same MPEG-2 VBR video which has the trace data shown in figure 1. Also, in each simulation, each stream specifies the same PDT QoS value. For disk system 1, if a video server has 120 MB, MR retrieval supports 50% more streams than CT retrieval, if users can tolerate a pre-fetch delay of 10.0 sec.

In the case of MR retrieval scheduling, we used the heuristic resource reservation algorithm to maximize the number of streams that can be supported concurrently by a video server. In the case of CT retrieval, the resource reservation algorithm is based on the fact that each stream requires a bandwidth reservation equal to the peak data rate of the requested video. Figures 13 and 14 show the total number of admissible video streams at the video server system as the total memory resource is increased, while keeping the disk system the same.

For MR retrieval, we can see that the number of streams that can be supported increases as the video server memory resources are increased. For continuous, lossless retrieval in interactive viewing, the resource reservation algorithm based on MR retrieval guarantees that no other retrieval schedule can support more video streams for a given set of video server resources.

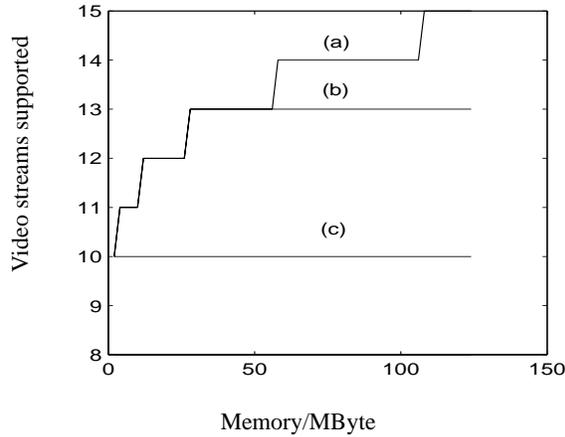
It can be seen that CT retrieval cannot take advantage of any increase in the memory resource of a video server. The advantage of the CT retrieval schedule is that the PDT QoS is always zero. This does not mean that the total delay that the client experiences before receiving its requested video is zero. However, the pre-fetch delay is zero. It can be seen that the performance of this scheme is the same as MR retrieval in which clients specify a PDT QoS of zero.

Table 4: Disk performance parameters

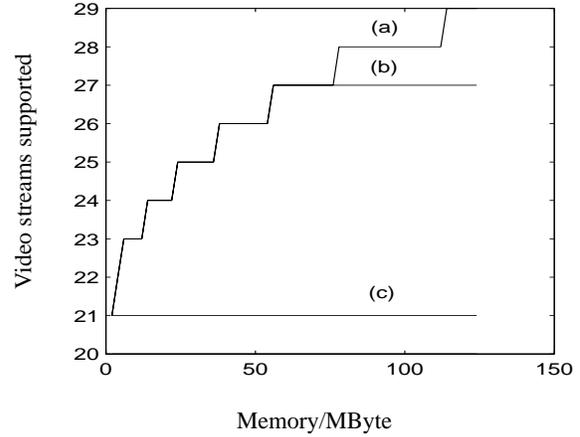
| Parameter | Disk system 1 | Disk system 2 |
|--------------------------|---------------|---------------|
| Disk cycle time/second | 0.5 | 0.5 |
| Max. rotation latency/ms | 14.2 | 7.1 |
| Max. seek latency/ms | 18.0 | 9.0 |
| Min. seek latency/ms | 1.5 | 0.75 |

Table 4: Disk performance parameters

| | | |
|------------------------------|------|-------|
| Max. disk transfer rate/Mbps | 60.0 | 120.0 |
| No. of disks in array | 4 | 4 |

**Fig. 13.** Performance comparison of MR and CT retrieval (disk system 1).

- (a) MR retrieval PDT QoS 10.0
- (b) MR retrieval PDT QoS 2.0
- (c) CT retrieval (PDT QoS 0.0)

**Fig. 14.** Performance comparison of MR and CT retrieval (disk system 2).

- (a) MR retrieval PDT QoS 10.0
- (b) MR retrieval PDT QoS 2.0
- (c) CT retrieval (PDT QoS 0.0)

Performance comparison of MR and CD retrieval

Figure 15 compares the performance of MR and CD retrieval scheduling. Each line corresponds to a single simulation run. In each simulation all streams are accessing the same MPEG-2 VBR video which has the trace data shown in figure 1. Also, in each simulation, each stream specifies the same PDT QoS value. For disk system 1, if a video server has 150 MB, MR retrieval supports 275% more video streams than CD retrieval. CD retrieval is memory bound.

In the case of MR retrieval scheduling, we used the fast heuristic resource reservation algorithm to maximize the number of streams that can be supported concurrently by a video server. In the case of CD retrieval, the resource reservation algorithm is based on two facts. Firstly, each stream requires a bandwidth reservation equal to the average data rate of the requested video. Secondly, there is a fixed memory requirement for the retrieval of the video. Figure 15 shows the total number of admissible streams at the video server system as the on-board memory resource is increased, while keeping the disk bandwidth the same.

We can see that the number of streams supported by CD retrieval is generally much lower than MR retrieval. This scheme is essentially *memory bound*. The bandwidth is not fully utilized since the memory

requirements are the limiting factor in the resource reservation.

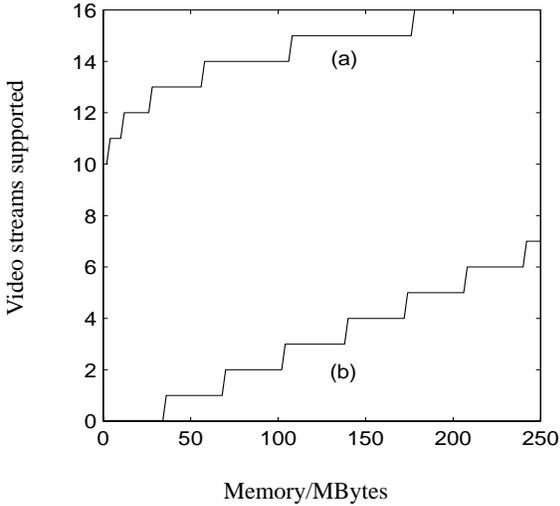


Fig. 15. Performance comparison of MR and CD retrieval (disk system 1).

- (a) MR retrieval (PDT QoS 135.0 sec)
- (b) CD retrieval (PDT QoS 135.0 sec)

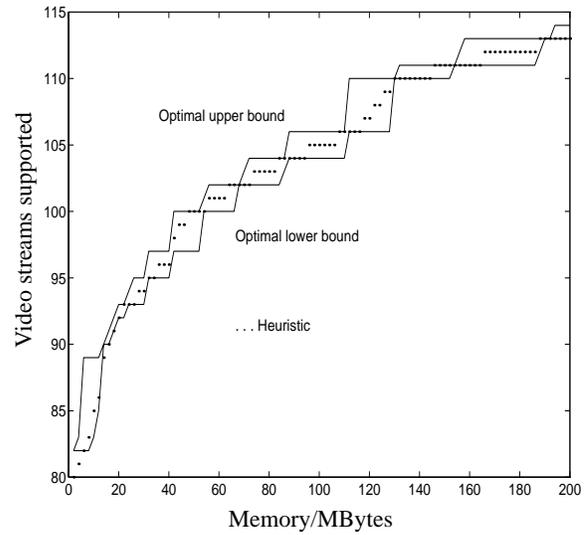


Fig. 16. Performance comparison of optimal and heuristic resource reservation algorithms.

Performance comparison of optimal and heuristic resource reservation algorithm

Figure 16 compares the maximum number of streams that can be supported concurrently by a video server for the optimal resource reservation algorithm and the heuristic resource reservation algorithm. The simulation parameters are given in table 5. The request pattern for videos is uniform. The videos are sequences from the movie ‘Forrest Gump’ and ‘Ben Hur’. For each request for a video, the mean PDT QoS is shown in table 5. For this simulation, we used the performance characteristics of disk system 2, however, we considered a large scale disk system with 16 disks.

For the optimal algorithm the exact optimal solution is not computed due to high computation requirements. Instead, an upper and lower bound on the number of admissible streams is found. The derivation of these bounds are discussed in appendix A.

Figure 16 shows that as the memory resource is increased, the number of streams also increases. It was found that the difference in the number of streams that can be admitted between the heuristic and optimal algorithm is less than 10% for this simulation. Therefore, the performance of the heuristic algorithm is very close to that of the optimal algorithm.

Table 5: Simulation parameters

| | |
|---|------|
| No. of videos | 5 |
| Mean of Avg. bit rates of all videos/Mbps | 2.52 |
| Mean of peak rate of all videos/Mbps | 8.86 |
| Mean PDT QoS/sec | 5.0 |

Performance comparison of progressive and non-progressive display of scalable video

In figure 17, we compare the progressive display for the interactive viewing of scalable video with non-progressive display of scalable video. All streams are retrieving the same MPEG-2 scalable video. The video sequence is from the movie ‘Ben Hur’ and the related information is shown in table 6. The request pattern for the scalable layers of the video is uniform. For progressive display, progressively higher PDT QoS values are specified for the progressively higher scalable layers (table 6). In non-progressive display, the same PDT QoS is specified for the full resolution of video to achieve the same degree of interactivity. Figure 17 shows that using progressive display for the interactive viewing of scalable video increases the number of streams supported by a video server. For disk system 1, if a video server has a memory resource of 120 MB, progressive display supports 17% more scalable layer streams than non-progressive display.

Table 6: Simulation parameters

| Scalable layer | Base | Spatial | SNR |
|--|------|---------|------|
| Avg. rate/Mbps | 0.7 | 0.44 | 2.5 |
| Peak rate/Mbps | 1.8 | 1.0 | 4.7 |
| Non-progressive display PDT QoS/Sec | 2.0 | 2.0 | 2.0 |
| Progressive display PDT QoS/Sec | 2.0 | 10.0 | 20.0 |

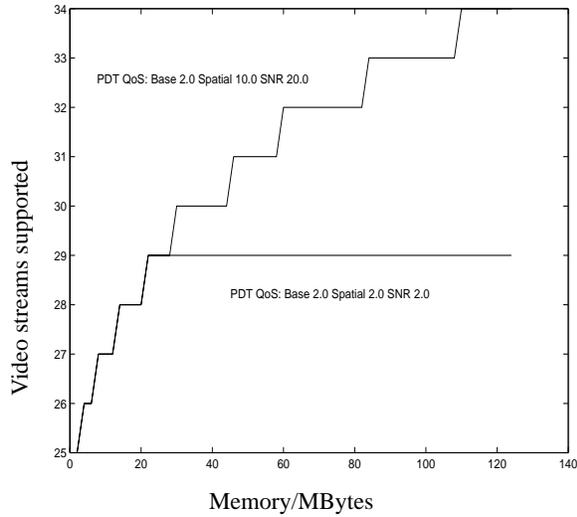


Fig. 17. Performance evaluation for progressive display of scalable video (disk system 1).

XII. Conclusions

In this research we have presented a new retrieval schedule for the retrieval of bursty VBR video data from the disk system to the memory of a video server. MR retrieval allows a range of bandwidths to be reserved for retrieval and guarantees a minimal buffer requirement for each bandwidth reservation. We presented a resource reservation algorithm for video server resources based on the MR retrieval schedule. We provided theoretic proofs for the optimality of the resource allocation algorithm. We also presented a fast, low complexity heuristic algorithm which approximates the theoretical optimal solution within 10%. The retrieval schedule and resource reservation algorithms are flexible enough to be implemented on general purpose computers. Performance evaluations based on simulations using MPEG2 trace data were presented. We found that the retrieval schedule and resource reservation algorithm dramatically improves the performance and flexibility of video servers compared to previous approaches.

To support heterogeneous clients, we also applied scalable video coding to our retrieval schedule and resource reservation algorithm. We proposed a new retrieval method for scalable video- progressive display, and showed a significant increase in system utilization and efficiency.

Appendix A Optimal Resource Reservation Algorithm for MR Retrieval

For the notation defined in table 3, the optimal resource reservation problem was shown in section 9 to be equivalent to the following two step algorithm:

1) Find the reservation vector \vec{B}^o which is the solution to the following non-linear constrained minimization problem:

$$\min M_T = \sum_{k=1}^M R_k(b_k)$$

Subject to

$$B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

2) If $M_T \leq C_M$ (system memory constraint), the set of all streams can be supported by the system, otherwise the set of streams cannot be supported.

The solution to the constrained minimization problem is based on the following theorem:

Theorem: For any $\lambda \geq 0$, the solution $\vec{B}^o(\lambda)$ to the *unconstrained* minimization problem

$$\min \left\{ \sum_{k=1}^M R_k(b_k) + \lambda \sum_{k=1}^M b_k \right\}, \text{ subject to } \vec{B} \in S$$

is also the solution to the *constrained* minimization problem with the constraint $C_B^o(\lambda) = \sum_{k=1}^M b_k^o(\lambda)$.

The proof of this theorem is presented in [11]. This theorem does not guarantee a solution to the constrained minimization problem. It just says that for every λ , there is a corresponding constrained problem for which the solution is the same as that of the unconstrained problem. The main point is that if $C_B^o(\lambda)$ happens to equal C_B , then $\vec{B}^o(\lambda)$ is the desired solution to the original constrained problem.

In solving the constrained problem, we can sweep over λ from 0 to ∞ and try to find a corresponding $C_B^o(\lambda)$ that equals C_B . The key question then becomes the problem of finding the corresponding λ efficiently. This forms the main portion of the optimal resource reservation algorithm.

The crucial point in solving the unconstrained problem is that the solution is obtained by minimizing each term of the sum separately.

If $\vec{B}^o(\lambda)$ is the solution to the optimization problem, we denote by $b_k^o(\lambda)$ the k th component of $\vec{B}^o(\lambda)$, and define $S_k = \{p_k, p_k + 1, \dots, q_k\}$. Then $b_k^o(\lambda)$ solves

$$\min \{R_k(b_k) + \lambda \cdot b_k | b_k \in S_k\}.$$

Given λ , one can solve for all $b_k^o(\lambda)$, sum them all up to get $C_B^o(\lambda)$, and then compare this value to the desired constraint C_B . If $C_B^o(\lambda) = C_B$, the desired solution has been found. Note that $b_k^o(\lambda)$ may not be unique. Therefore, the solution $\vec{B}^o(\lambda)$ is also not necessarily unique.

The following points show that it is not necessary to sweep over all λ (all values on the non-negative real line) in order to find the solutions to the unconstrained problem. The proof of these points are given in [11]

- 1) $b_k^o(\lambda)$ either decreases or remains unchanged as λ increases.
- 2) Any given λ can either have a *singular* or *non-singular* solution $b_k^o(\lambda)$ to the unconstrained problem. Singular points are values of λ for which more than one solution exists. Non-singular points only have one solution.
- 3) Two adjacent singular values have one and only one solution in common.
- 4) All non-singular values of λ between two adjacent singular values have the same (single) solution.

A typical dependency of the solution rate $C_B^o(\lambda)$ on the lagrange multiplier λ can be shown to have the form of figure 12. It shows a decreasing staircase curve where the discontinuities correspond to singular values of λ .

The main conclusion from the above points is that to find a constraint which satisfies $C_B^o(\lambda) = C_B$, it is not necessary to search over all singular λ . It is sufficient to check only the singular points. Locating the desired solution requires the knowledge of the singular λ . The following lemma are used to locate all the singular λ .

Lemma 1: Let λ_1 be singular, $\lambda > \lambda_1$, and $\vec{B}^o(\lambda)$ be a common solution associated with λ and λ_1 . Define the set

$$S_k^I = \{b_k^o + 1, \dots, q_k\}.$$

This set is non-empty if $b_k^o + 1 \leq q_k$. Also, define the set

$$M^I = \{k \in \{1, \dots, M\} | S_k^I \neq \emptyset\}.$$

This defines the set of all resource relations for which an increase in bandwidth is possible. Then, the singular value

λ_1 , which is necessarily the closest to λ from below is given in terms of λ by

$$\lambda_1 = \max \left\{ \frac{R_k(b_k^o(\lambda)) - R_k(b_k)}{b_k - b_k^o(\lambda)} \mid k \in M^I, b_k \in S_k^I \right\}.$$

Lemma 2: Let λ_2 be singular, $\lambda < \lambda_2$, and $\vec{B}^o(\lambda)$ a common solution associated with λ and λ_2 . Define the set

$$S_k^D = \{p_k, \dots, b_k^o - 1\}.$$

This set is non-empty if $b_k^o - 1 \geq p_k$. Also, define the set

$$M^D = \{k \in \{1, \dots, M\} | S_k^D \neq \emptyset\}.$$

Then, the singular value λ_2 , which is necessarily the closest to λ from above is given in terms of λ by

$$\lambda_2 = \min \left\{ \frac{R_k(b_k) - R_k(b_k^o(\lambda))}{b_k^o(\lambda) - b_k} \mid k \in M^D, b_k \in S_k^D \right\}.$$

The last two lemmas say that by knowing a singular value, the next from below or from above can be found, and thus, all singular values and all solutions $\vec{B}^o(\lambda)$ can be located.

We can now present the optimal resource reservation algorithm used in section 9:

- 1) Obtain an initial value of λ .
- 2) Solve the unconstrained problem. If λ is not singular, there is only one such solution and one constraint $C_B^o(\lambda)$. If λ is singular, then there are at least two different solutions. Find the two solutions from $\{\vec{B}^o(\lambda)\}$ with greatest and smallest constraints denoted by $C_B^l(\lambda)$ and $C_B^h(\lambda)$, respectively.
- 3) If the desired constraint C_B is such that $C_B^l(\lambda) \leq C_B \leq C_B^h(\lambda)$, then obtain all solutions in $\{\vec{B}^o(\lambda)\}$ and find the one for which the constraint, denoted by $C_B^a(\lambda)$, is the closest to C_B from below. If $C_B = C_B^a(\lambda)$, an exact optimal solution has been found. If not, an approximate solution has been found. Go to step 6.
- 4) If $C_B \leq C_B^o(\lambda)$ [or $C_B \leq C_B^l(\lambda)$], find the next singular λ using the current solution which corresponds to $C_B^o(\lambda)$ [or $C_B^l(\lambda)$]. Then, go to step 2.
- 5) If $C_B \geq C_B^o(\lambda)$ [or $C_B \geq C_B^h(\lambda)$], find the next singular λ using the current solution which corresponds to $C_B^o(\lambda)$ [or $C_B^h(\lambda)$]. Then, go to step 2.
- 6) If $M_T = \sum_{k=1}^M R_k(b_k^o) \leq C_M$ the set of all video streams can be supported by the video server, otherwise the set cannot be supported.

References:

1. E. Biersack, F. Thiesse, and C. Bernhardt, "Constant Data Length Retrieval for Video Servers with VBR Streams." Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems. June, 1996.
2. P. Bocheck, H. Meadows, and S.-F. Chang, "A Disk Partitioning Technique for Reducing Multimedia Access Delay." International Conference on Distributed Multimedia Systems and Applications. August, 1994.
3. A. J. Chaney, I. D. Wilson and A. Hopper, "The Design and Implementation of a RAID-3 Multimedia File Server." Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video. April. 1995.
4. E. Chang and Avidesh Zakhor, "Scalable Video Data Placement on Parallel Disk Arrays." SPIE Symposium on Imaging Technology. 1994.
5. E. Chang and A. Zakhor, "Admissions Control and Data Placement for VBR Video Servers." IEEE International Conference on Image Processing. 1994.
6. M. S. Chen, D. D. Kandlur, P. S. Yu, "Optimization of Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams." ACM Multimedia 93. 1993.
7. T. Chiang and D. Anastassiou, "Hierarchical Coding of Digital Television." IEEE Communications Magazine. May, 1994.
8. B. Ozden, R. Rastogi, A. Silberschatz, "Disk Striping in Video Server Environments." Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems. June, 1996.
9. S. Paek, P. Bocheck and S.-F. Chang, "Scalable MPEG2 Video Servers with Heterogeneous QoS on Parallel Disk Arrays." Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video. April. 1995.
10. S. Paek and S.-F. Chang, "Video Server Retrieval Scheduling for VBR Scalable Video." Proceedings of the Third IEEE International Conference on Multimedia Computing & Systems. June, 1996.
11. Y. Shoham and Allen Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers." IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. 36, No. 9. Sept. 1988.
12. H. Vin, P. Goyal, A. Goyal(1), and A. Goyal(2), "A Statistical Admission Control Algorithm for Multimedia Servers." ACM Multimedia 94. 1994.
13. Y. Yu, "Columbia MPEG Software Release 6.5 User's Manual." Technical report of Image & Advanced Television Laboratory, Columbia University. 1994.
14. S. Sahu, Z. Zhang, J. Kurose and D. Towsley. "On the Efficient Retrieval of VBR Video in a Multimedia Server." Proceedings of the IEEE International Conference on Multimedia Computing and Systems. June 1997.
15. J. D. Salehi, Z. Zhang, J. F. Kurose and D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements Through Optimal Smoothing." University of Massachusetts Technical Report TR 95-98. November, 1995.
16. J. Mc Manus, K. W. Ross, "Video on Demand over ATM : Constant-Rate Transmission and Transport", IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 1087-1098, 1996.

Video Server Retrieval Scheduling and Resource Reservation for Variable Bit Rate Scalable Video

Seungyup Paek and Shih-Fu Chang

Department of Electrical Engineering, Columbia University
1312 S.W. Mudd Building, New York, NY 10027, USA.
{syp, sfchang}@ee.columbia.edu

Abstract

State of the art digital video compression produces bursty, variable bit rate video. The bursty nature of compressed video raises challenges in the design of video servers. In this paper, we first present a method for the efficient retrieval of bursty video data from the disk system to the memory of a digital video server. For a *single* video stream, the proposed retrieval schedule minimizes the buffer requirement for continuous retrieval, given that a fixed disk bandwidth is reserved for the entire duration of retrieval. Secondly, we present an optimal resource reservation algorithm for *multiple* video streams based on the proposed retrieval schedule. The resource reservation algorithm maximizes the number of bursty video streams that can be supported by a video server, given any disk bandwidth and memory resource. Thirdly, we present a progressive display scheme for scalable video that is based on the retrieval schedule and resource reservation algorithm. Performance evaluations based on simulations using MPEG-2 trace data are presented. For a PC with four disks and a memory resource of 120 MBytes, our approach can support 50%-275% more video streams than previously proposed approaches, depending on the pre-fetch delay that users are willing to tolerate in interactive viewing of videos.

I. Introduction

In recent years, the advent of powerful computing, networking and consumer electronics technologies have made it possible to develop *multimedia networks for the home*. Amongst the myriad of technologies, it is likely that homes of the future will have multimedia enabled Personal Computers (PCs), Digital Versatile Disk (DVD) players, TVs, and digital video camcorders. PCs and consumer electronics devices are expected to be seamlessly connected by digital interconnects such as IEEE 1394, enabling new multimedia applications for the home. IEEE 1394 technology is a high-speed data pipe with speeds starting at 100 Mbps. Current specifications allow for speeds of up to 800 Mbps and speeds beyond 1600 Mbps are being developed.

In a home based multimedia network, PCs can be used as a digital video server. In other words, the PC can be used to store multiple videos and provide interactive viewing of videos to multiple devices concurrently.

To fully utilize the storage resources of the PC, the videos stored on the PCs will be compressed. State of the art digital video compression produces bursty, variable bit rate (VBR) video. Figure 1 shows trace data for MPEG-2 VBR video data. The video sequence is from the movie ‘Forrest Gump’. VBR video provides several advantages over constant bit rate (CBR) video, including consistent video quality and lower encoder complexity. However, in general, the bursty nature of compressed video raises challenges in disk storage, disk retrieval and network transmission of video servers.

For a home based multimedia network, based on current technologies, we can expect the main bottleneck for PC based digital video servers to lie in the disk retrieval scheduling of bursty video. This is because current disk I/O bandwidths for PCs are generally lower than relatively high speed digital interconnects such as IEEE 1394, which is expected to be used in home based multimedia networks. Therefore, it is important to develop methods for the efficient retrieval of bursty video data from the disk system to the memory of a PC based digital video server. Video data has to be retrieved from the disk system to memory before transmission into a network.

Most video servers operate in cycles, and during each cycle time, data is retrieved by a disk retrieval scheduler from the disk system to memory for each stream that is supported. For CBR data, the disk retrieval is relatively simple. During each cycle, a constant amount of data is retrieved for each stream. However, for VBR video, it is not clear how data should be retrieved for each stream.

In Constant Time (CT) retrieval of VBR data [5, 12], data corresponding to a constant time is retrieved for each stream during each cycle of operation of a video server. Let v be the video that a stream s is retrieving. For continuous, lossless retrieval, it is necessary to reserve a disk bandwidth equal to the peak data rate of video v for stream s . In this paper, we will show that video servers based on CT retrieval cannot always fully utilize the memory resources of a video server in maximizing the number of supported streams.

In Constant Data (CD) retrieval of VBR data [1, 5], a constant amount of data is retrieved for each stream during each cycle of operation of a video server. A disk bandwidth equal to the average data rate of video v is reserved for stream s . In general, this scheme requires data to be pre-fetched into a buffer before transmission to ensure that buffer starvation does not occur. Buffer starvation can occur because the amount of data transmitted during each cycle is variable for VBR video, while the amount of data retrieved from the disk system is constant during each cycle. In this paper, we will show that video servers based on CD retrieval cannot always fully utilize the bandwidth resources of a video server in maximizing the number of supported video streams.

To overcome the limitations of CD and CT retrieval, we developed a *Minimal Resource* (MR) retrieval schedule and an associated optimal resource reservation algorithm that can fully utilize a given set of video server resources to maximize the number of supported video streams.

In MR retrieval, a range of disk bandwidths can be reserved for the retrieval of VBR data. This is in con-

trast to CD and CT retrieval, in which a disk bandwidth equal to either the average data rate or peak data rate of a VBR video is reserved, respectively. In MR retrieval, as the reserved bandwidth for a stream is reduced, the worst case buffer requirement for a stream increases. A buffer is required since data has to be pre-fetched to ensure continuous, lossless transmission. For a given bandwidth reservation, MR retrieval *minimizes* the worst case buffer memory requirement for continuous, lossless transmission into the network.

MR retrieval leads to a *buffer-bandwidth resource relation* for each VBR video. The resource relation shows the worst case buffer requirement for a given bandwidth reservation. We can show that increasing the reserved bandwidth reduces the worst case pre-fetch delay. Based on this, we introduce the concept of a *Pre-fetch Delay Tolerance Quality of Service* (PDT QoS). A PDT QoS is specified for each stream that a video server supports, and it specifies the worst case pre-fetch delay that can be tolerated during interactive retrieval. It is shown that a PDT QoS specified for a stream s that retrieves video v is equivalent to placing a lower bound on the bandwidth that can be reserved for stream s . The lower bound for the bandwidth can be determined from the buffer-bandwidth resource relation of video v .

For MR retrieval, we developed an optimal resource reservation algorithm for multiple streams. The resource reservation algorithm determines what buffer and bandwidth should be reserved for each stream. The objective of the resource reservation algorithm is to maximize the number of video streams for a given set of video server resources, while ensuring the PDT QoS and retrieval constraints of each stream. It can be shown that the resource reservation algorithm based on MR retrieval fully utilizes the bandwidth resources and maximizes the number of supported video streams.

The MR retrieval scheme essentially defines a spectrum between CD and CT retrieval. The MR retrieval scheme and the optimal resource reservation algorithm make optimum use of the computing resources for a given server configuration.

We present performance evaluations based on simulations using MPEG2 trace data. It is found that the optimal resource reservation algorithm based on MR retrieval improves the performance of video servers compared to resource reservation algorithms based on CT or CD retrieval. For a video server based on a PC with four disks and a memory resource of 120 MBytes, our approach supports at least 50% more video streams than previously proposed approaches. As the PDT QoS is increased, the number of streams supported by a video server is increased.

Compared to simulcast coding, scalable coding schemes can provide multiple levels of video with a minimal cost of extra bandwidth or storage capacity. In scalable video coding, subsets of the full resolution bitstream are used to obtain subsets of the full resolution video. In this paper we present a scheme for the *progressive display* of scalable video for interactive viewing. In this scheme, a progressively increasing PDT QoS is specified for the progressively higher scalable layers of a video. Each scalable layer is consid-

ered as an independent video. This is in contrast to non-progressive display of scalable video in which a single PDT QoS is specified for all the scalable layers. For a video server with a single disk and a memory resource of 120 MBytes, progressive display supports 17% more scalable layer streams than non-progressive display.

This paper is organized as follows. In section 2 we first discuss related research in retrieval scheduling and resource reservation. Section 3 presents our basic assumptions and the video server system model. Section 4 discusses the main differences between CBR and VBR video. Section 5 outlines the retrieval constraints to ensure continuous, lossless retrieval of VBR video in video servers. Section 6 and 7 examines two approaches for retrieval scheduling of VBR video. We discuss the limitations of each approach. Section 8 presents the minimal resource (MR) retrieval schedule for VBR video data. We discuss how this retrieval schedule overcomes the limitations of previously proposed approaches. Section 9 presents a resource reservation algorithm based on MR retrieval that is necessary to maximize the number of video streams that a server can support. Section 10 briefly overviews scalable video and then presents a scheme for the progressive display of scalable video. Progressive display improves the performance of video server supporting scalable video. Section 11 presents performance evaluations of the MR retrieval schedule in comparison to other proposed approaches.

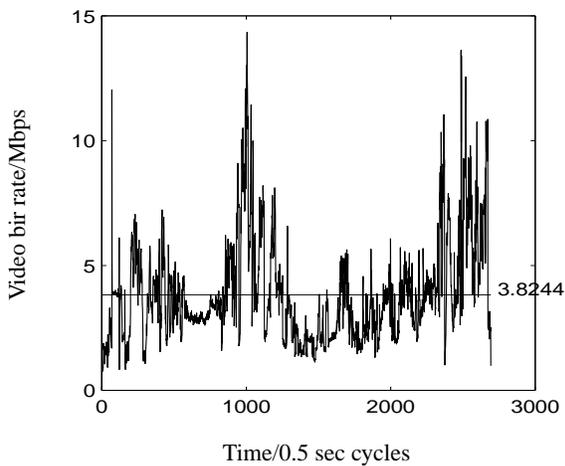


Fig. 1. MPEG-2 VBR video trace.

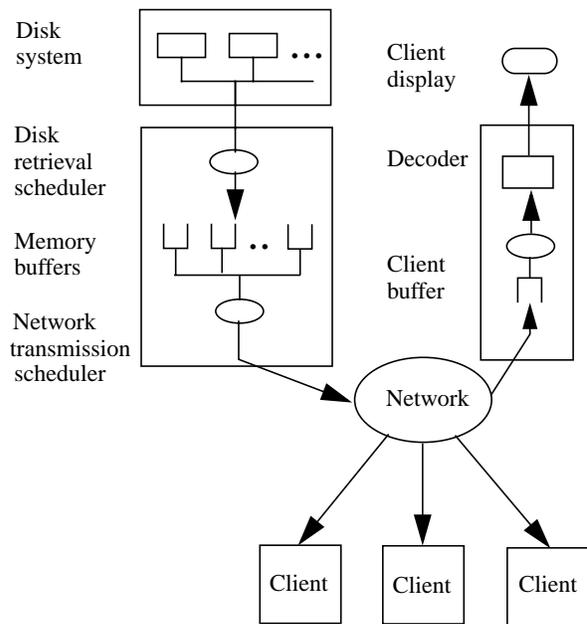


Fig. 2. Video server system model.

II. Related Work

In [12], CT retrieval is the basis for the admission control scheme in multimedia servers. The primary contribution of the work in [12] is that *statistical service guarantees* are provided to all streams. In other words, for each stream, a continuous retrieval is guaranteed to a fixed percentage of the video data. It is proposed that a certain percentage of video data can have the continuity requirement violated without significantly affecting the quality of the video. This leads to an improvement in the utilization of the server. New clients are admitted for service as long as the statistical estimate of the aggregate data rate requirement (rather than the peak data rate requirement) can be met. The approach of ensuring statistical guarantees, rather than deterministic guarantees is certainly effective in increasing the performance of a multimedia server. However, the approach proposed in [12] is based on CT retrieval and has the drawback that memory resources cannot be fully utilized to maximize the number of supported streams.

In [5], CD retrieval is the basis for both a deterministic and statistical admission control scheme. The retrieval scheme is referred to as Constant Data Length (CDL) retrieval. In [1], two retrieval schemes called traditional CDL and generalized CDL (GCDL) are presented. In the traditional CDL scheme of [1], a constant amount of data is retrieved from the disk for a video stream in the first disk cycle. In the second cycle, the same constant amount of data is retrieved only if it is required to prevent buffer underflow. Otherwise, no data is retrieved. This process repeats throughout the retrieval. Therefore, each retrieval cycle is either an idle or active round. Although a constant data amount is retrieved during an active round, the overall retrieval can be variable bit rate. This is different from [5] in which the overall retrieval is constant bit rate i.e. there are no idle rounds. In [1], the GCDL scheme is an extension of the traditional CDL scheme in which the retrieval round can be different for different video streams and which are a multiple of the disk cycle. This is shown to reduce the buffer requirements compared to traditional CDL.

In our video server model, we have assumed a cycle based operation of the video server in which the disk retrieval scheduler retrieves data for multiple video streams from the disk system to the memory during each cycle. It is important to note that not all video servers operate on a cyclic basis. For example, [14] proposes a deadline based scheme.

In the following sections, it will be seen that our approach performs *smoothing* in retrieving video data from disks to memory. There have been several researchers who have also proposed smoothing techniques for bursty pre-recorded video data [15, 16]. Our approach differs because we are also concerned with the retrieval of *multiple* video streams across a common channel (disk to memory channel), using a common memory resource (server memory). The techniques proposed in [15, 16] deal with smoothing techniques of a single video stream from the server to the client across a high performance network, utilizing the client buffer for smoothing. Therefore, our work considers the dynamics between multiple video streams, whereas the other approaches are only concerned with per-stream dynamics.

III. System Model

In this section we describe the video server system model relevant to our research. The system model is shown in figure 2. The video server has a fault tolerant disk array for the the storage of video data, a memory resource, and a network interface for transmission into the computer network.

The disk retrieval scheduler has a cyclic operation [4, 8, 9, 10]. Each cycle, the disk retrieval scheduler retrieves data for multiple video streams from the disk system to the memory. The network stream scheduler also has a cyclic operation, although the cycle time of the network stream scheduler will typically be much smaller than the cycle time of the disk retrieval scheduler. The network stream scheduler transmits data for multiple video streams into the network during each cycle.

The video data is interleaved over all the disks of the array. Such a distribution technique is referred to as “wide striping” or “fine grained striping”. It has been shown that this results in buffer requirements that increase with the number of disks in the array. Therefore, using fine grained striping can lead to high buffer requirements for large scale commercial video servers. However, the focus of this paper is on small scale, PC based digital video servers for the home.

During each disk cycle, the disk head completes one cycle of a SCAN disk head schedule. The disk head scans the disk starting from the inner most track to the outer most track [2, 6]. While scanning the disk, data blocks belonging to different streams are read from the disk. Upon reaching the outer most track, the head is returned back to its initial position. The performance analysis is given in [2]. Although multiple blocks for each video stream are being retrieved from the disks each cycle, the overhead of disk head seeks is greatly reduced by using the SCAN disk head scheduling algorithm.

There is no special data layout of the videos on the disks. There has been extensive research that addresses the layout of video data on disks. This paper does not address how data are actually laid out on disk. Instead, we have assumed that each disk operates with the SCAN disk head schedule, which does not assume any special data layout on disk. We have done detailed analysis, performance evaluation and simulation in [2].

The network can accommodate the peak bandwidth of all video streams and introduces zero delay and zero jitter. The limitations of the network are not considered. In future research we will consider the limitations of the network, and we will also consider the interaction between the disk retrieval scheduler and network stream schedulers. In particular, we will consider the interaction of our disk retrieval scheduler with network stream scheduling algorithms to smooth VBR video data for transmission in networks.

The video server supports completely interactive viewing and continuous, lossless retrieval. Video servers supporting completely interactive viewing allow viewers to pause and resume playback at any time during a viewing session. Playback can also resume at any point of a video. Video servers supporting continuous, lossless retrieval provision resources so that once playback of a portion of a video has started, no delay is

introduced and no data is lost. Note that this does not mean that there can be no delay *before* playback begins e.g. after a pause.

The video server has a fixed bandwidth and fixed buffer reserved for the entire duration of interactive retrieval. There are no renegotiations of resources for each video stream during a viewing session. This is a key simplifying assumption in our research. There is no renegotiation of resources for the retrieval during a viewing session.

IV. CBR and VBR Compressed Digital Video

In our research we used MPEG2 compressed digital video. However the retrieval schedule and the resource reservation algorithm are directly applicable to any video codec that results in VBR video. The variable bit rate of MPEG2 video is dependent on the encoding structure of the MPEG2 coding algorithm. In MPEG2 digital video, compression is achieved by the combination of techniques such as the discrete cosine transformation (DCT), variable length codes, quantization of DCT coefficients, motion estimation and motion compensated inter-frame prediction. MPEG2 has a buffer control mechanism in which the quantization parameter can be varied adaptively in order to achieve a constant average bit rate of the compressed video. The disadvantage of this mechanism is that the subjective visual quality will be variable, since the quantization parameter is continually varied. An alternative is to maintain a constant quantization parameter during the encoding of video. This results in variable bit rate video, in which the amount of data to represent different time scales of video (macroblock, slice, frame, group of pictures etc.) are variable. The research presented in this paper maximizes the number of VBR streams that can be supported by a video server.

V. Time Constraints

In the following sections, although the operation of a video server is based on discrete time cycles, we will use continuous time notation to clearly convey the core ideas. Table 1 defines the notation we will use in the following sections.

Table 1: Notation

| | |
|--------|--|
| $r(t)$ | Data rate of stored video |
| t_n | Start of network transmission |
| t_d | Start of disk retrieval |
| T | Time duration of video |
| $o(t)$ | Total data output from memory into network |

Table 1: Notation

| | |
|--------|--|
| $i(t)$ | Total data input from disk into memory |
| m | Buffer memory reserved for retrieval |
| b | Disk bandwidth reserved for retrieval |

The retrieval constraints for the retrieval of a video are as follows:

- 1) $i(t) \geq o(t)$ (continuous retrieval constraint)
- 2) $i(t) - o(t) \leq m$ (Buffer constraint)
- 3) $\frac{di(t)}{dt} \leq b$ (Disk bandwidth constraint)

In the following, we will assume that $o(t) = \int r(t - t_s)$. In other words, the data is output into the network according to the data rate of the stored video, $r(t)$, starting at time t_n .

VI. Constant Time Retrieval

Constant time (CT) retrieval is described by the equation $i(t) = o(t)$. The accumulated input data and accumulated output data are equivalent. In an actual system, the network stream scheduler waits one cycle time t_c after disk retrieval begins before it can start transmission into the network.

The delay of one cycle exists because we assume that the video server uses a double buffer scheme for retrieval and transmission (figure 3). During each cycle, data is read from the disk into the disk read buffers for each stream. Each disk executes one SCAN cycle. Concurrently, data is written from the network write buffers into the network for each stream. At the end of each cycle, the contents of the disk read buffers are written onto the network write buffers, and the process repeats. No pre-fetch buffer is required for CT retrieval. In the first cycle, the disk read buffer is being filled, while the network write buffer is empty.

As mentioned earlier, network transmission can only begin after a delay of one cycle. This delay is different from the pre-fetch delay mentioned in the following sections and is common to all the retrieval schedules. We shall ignore this delay in the following sections. The pre-fetch delay for CT retrieval is zero.

For each stream, a disk bandwidth of b equal to the peak data rate of the video being retrieved must be reserved for the entire duration of an interactive viewing session:

$$b = \max \{r(t), 0 \leq t \leq T\}.$$

In the following sections, we ignore the disk read buffers and network write buffers and only consider the pre-fetch buffers.

The major disadvantage of CT retrieval is that it cannot take advantage of memory resources to maximize

the number of streams supported by a video server. Consider a video server that uses CT retrieval. Each stream has a bandwidth reservation equal to the peak data rate of the video being retrieved. Assume that the total bandwidth reserved for all streams is equal to the total bandwidth of the video server. Assume that a large buffer memory exists in the video server. Using CT retrieval, no more streams can be supported by the video server due to the bandwidth limitation. It will be shown that if all viewers can tolerate a pre-fetch delay, the bandwidth reserved for all the streams can be substantially reduced from the peak bandwidth. Reducing the reserved bandwidth for each stream will be shown to require an increase in pre-fetch buffer requirements for each stream, if continuous, lossless retrieval is to be guaranteed. In this way, memory resources can be utilized to alleviate the I/O bandwidth bottleneck. By reducing the total bandwidth reserved for all the streams, the video server can potentially increase the number of streams that are supported. CT retrieval does not take advantage of viewers that can tolerate a pre-fetch delay and always assigns a pre-fetch delay of zero to all streams.

For the MPEG-2 VBR video shown in figure 1, CT retrieval has to reserve a bandwidth of 14 Mbps for the entire duration of retrieval. The buffer requirement for a video server operating at 0.5sec cycle time is 1.75 MB.

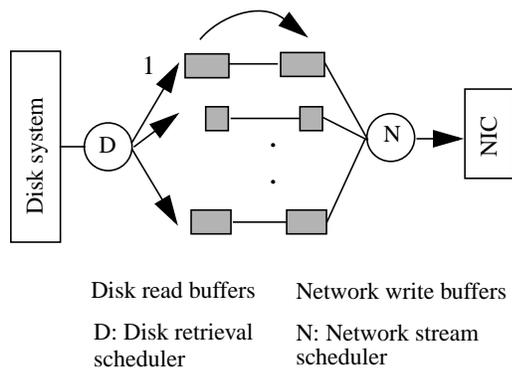


Fig. 3. Buffers for CT retrieval.

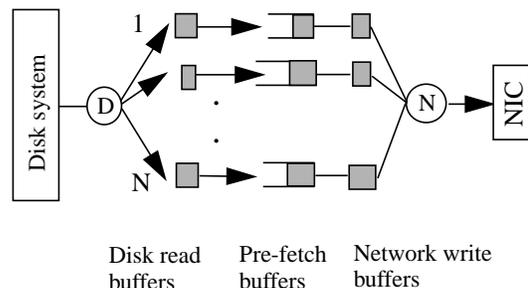


Fig. 4. Buffers for CD retrieval.

VII. Constant Data Retrieval

In constant data (CD) retrieval, a fixed bandwidth of b Mbps is reserved for each stream. The reserved bandwidth is typically much smaller than in CT retrieval, in which the peak bandwidth is reserved. CD retrieval retrieves a fixed amount of data $b \cdot t_c$ during each cycle until all the data has been transmitted (figure 4, 5). This differs from CT retrieval, in which a variable amount of data is retrieved in each cycle. In this scheme, data has to be pre-fetched to ensure that continuous, lossless retrieval of the video is guaranteed. The continuous, lossless retrieval constraint is violated if buffer starvation occurs. This can occur because the amount of data transmitted during each cycle is variable, while the amount of data retrieved

from the disk system is constant during each time cycle. Since a pre-fetch data has to be retrieved, there is a pre-fetch delay associated with CD retrieval. The worst case pre-fetch delay $t_n - t_d$ can be determined for stored video because the entire trace is known a-priori:

$$t_n - t_d = \frac{p}{b},$$

$$p = \text{MAX}\{-(b \cdot \Delta - o(t_n + \Delta))\} \quad 0 \leq \Delta \leq T$$

The major disadvantage of CD retrieval is that it cannot fully utilize video server resources to maximize the number of streams. Consider a video server that uses CD retrieval. Each stream has a pre-fetch buffer requirement. Assume that the total buffer memory reserved for all streams is equal to the total memory resource of the video server. However, assume that the total bandwidth reserved for all streams is less than the total disk bandwidth of the video server. Using CD retrieval, no more streams can be supported by the video server because of the memory limitation. In the next section, it will be shown that by increasing the bandwidth reserved for each stream, the memory requirement for each stream can be substantially reduced. In this way, the bandwidth resources can be utilized to alleviate the memory bottleneck of a video server. By reducing the total memory reserved for all the streams, the video server can potentially support more streams.

CD retrieval continues to retrieve a constant amount of data until all data has been retrieved. Data is sometimes retrieved earlier than is required, leading to potentially unnecessarily large buffer requirements. For the MPEG-2 VBR video shown in figure 1, CD retrieval reserves a bandwidth of 3.8 Mbps for the entire duration of retrieval. The buffer requirement for a video server operating at 0.5sec cycle time is found to be 50 MB.

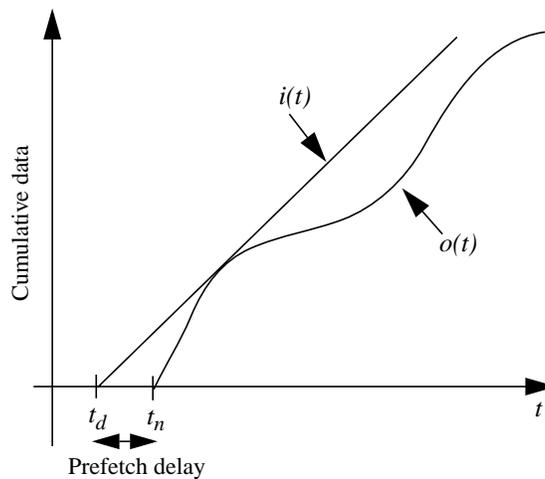


Fig. 5. Cumulative data for CD retrieval.

VIII. Minimal Resource Retrieval

In this section, we present a new Minimal Resource (MR) retrieval schedule for continuous, lossless retrieval of VBR video. MR retrieval is similar to CD and CT retrieval because it alternates between intervals of CT and CD retrieval. However, it differs in that a *range* of bandwidths can be reserved for the retrieval. CD retrieval requires that a bandwidth equal to the average data rate is reserved, while CT retrieval requires a bandwidth reservation equal to the peak data rate.

If the bandwidth reserved for retrieval of VBR video is less than the peak data rate, then data has to be pre-fetched to ensure continuous, lossless retrieval. Therefore, a buffer for pre-fetch data is required. In order to minimize the buffer requirement, data should be pre-fetched *just-in-time*. For the retrieval of VBR video, MR retrieval minimizes the worst case buffer requirement that is required for a given disk bandwidth reservation. It will be shown that this property of MR retrieval fully utilizes the resources of a video server.

We first describe the MR retrieval schedule and then discuss its properties. As before, let t_n be the start of transmission of a stream into the network. The accumulated data output from the memory to network is $o(t)$. $o(t) = 0$, $t \leq t_n$. The bandwidth reserved for the retrieval is b . We define a function $a(\alpha, \beta) = b \cdot (t - \alpha) + \beta$ that we will use in describing MR retrieval. We also use two new variables t_b, t_e to mark the beginning and end of each *maximum retrieval* interval. MR retrieval of stored video is described in table 2 and is shown graphically in figure 6.

Table 2: MR retrieval

| |
|--|
| 1. Set $t_e = t_n + T$ |
| 2. Decrease t_e until $\left. \frac{do(t)}{dt} \right _{t_e} \geq b$ |
| 3. Find the intersection of $a(t_e, o(t_e))$ with $o(t)$ |
| 4. Let $t_b < t_e$ equal the intersection point |
| 5. Mark the interval $[t_b, t_e]$ as a <i>maximum retrieval</i> interval |
| 6. If $t_b \leq t_n$ then stop |
| 7. Set $t_e = t_b$ and return to step 2 |

MR retrieval is defined by the maximum retrieval intervals. Figure 7 shows $i(t)$ for MR retrieval. If the time t falls inside any of the maximum retrieval intervals, the retrieval rate is at the maximum bandwidth

b. Otherwise, the retrieval amount is equal to the data rate. The buffer status at time t is $m(t) = i(t) - o(t)$.

The worst case pre-fetch delay is:

$$d = \frac{\text{MAX}\{m(t)\}}{b}, t_n \leq t \leq t_n + T.$$

Figure 9 shows MR retrieval of the MPEG-2 VBR video shown in figure 1.

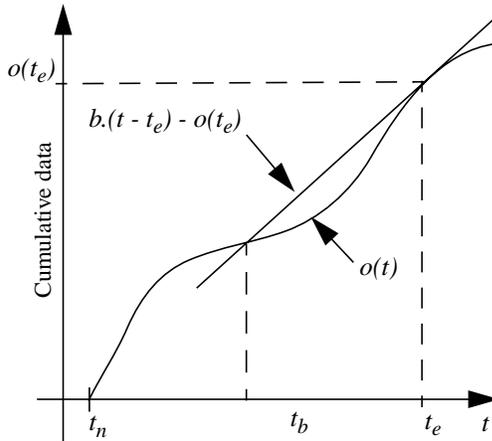


Fig. 6. Determination of MR retrieval.

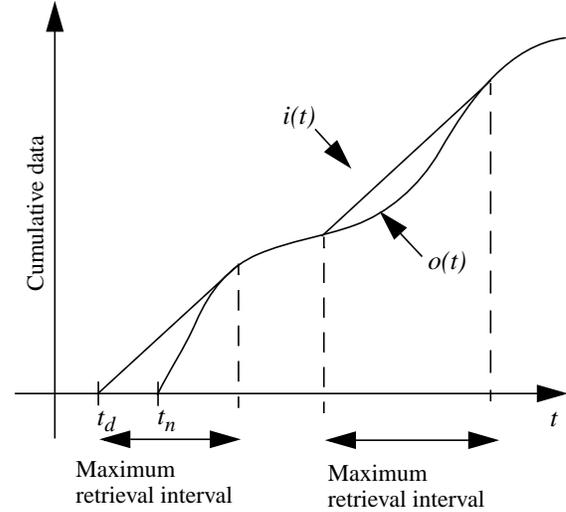


Fig. 7. Cumulative data for MR retrieval.

Proof of MR retrieval optimality

MR retrieval can be shown to minimize the worst case pre-fetch buffer requirement for a given bandwidth reservation for the continuous, lossless retrieval of a video. We can prove this by showing that MR retrieval is based on *just-in-time* retrieval. Consider the first maximum retrieval interval $[t_b, t_e]$. In MR retrieval, the retrieval rate during a maximum retrieval interval is b . We define a small time interval Δ . Consider the start of retrieval to be delayed to $t_b + \Delta$. In this case, it can be seen that even if the retrieval is at the maximum retrieval rate of b , the continuous retrieval constraint will be violated (figure 8). Therefore, t_b is the latest time at which pre-fetch of data can start if continuous retrieval is to be guaranteed up to t_e . Consider the start of retrieval to start earlier at $t_b - \Delta$. It can be seen that the buffer requirement will increase for any possible retrieval schedule as data is retrieved earlier than required. This analysis can be done iteratively for all the maximum retrieval intervals. Therefore, since MR retrieval is based on *just-in-time* retrieval, it minimizes the buffer requirement while satisfying the constraints for continuous, lossless retrieval.

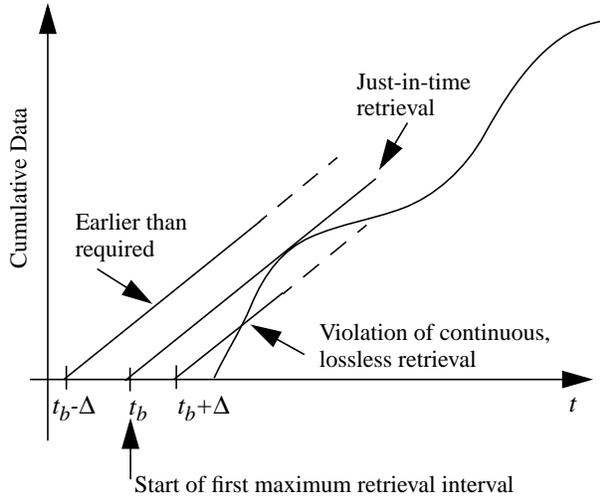


Fig. 8. MR retrieval optimality.

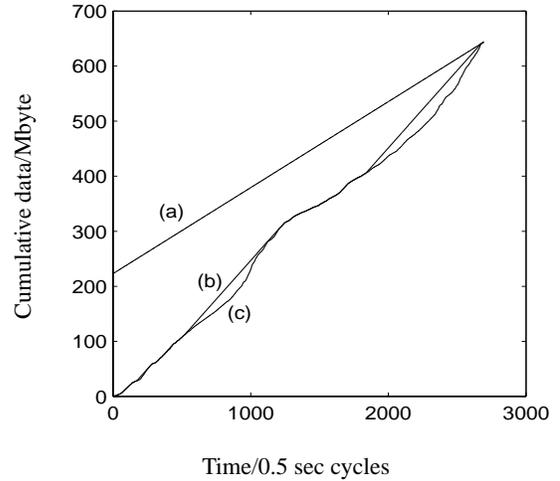


Fig. 9. MR retrieval of MPEG-2 VBR video.

- (a) $i(t)$ for MR retrieval with reserved bandwidth of 2.5 Mbps
 (b) $i(t)$ for MR retrieval with reserved bandwidth of 4.5 Mbps
 (c) $o(t)$

Buffer-bandwidth resource relation of MR retrieval

For a given bandwidth reservation, MR retrieval minimizes the worst case buffer that is necessary for continuous, lossless retrieval. It can also be shown that increasing the reserved disk bandwidth will reduce the minimized worst case buffer requirement. Therefore, MR retrieval leads to a buffer-bandwidth resource relation for the retrieval of a video. Figure 10 shows the buffer-bandwidth resource relation for the MPEG2 encoded video trace data shown in figure 1. From this relation, the corresponding worst case pre-fetch delay can be found. If the worst case buffer requirement for the retrieval of a given video is m , then the worst case pre-fetch delay is m/b , where b is the corresponding reserved bandwidth.

For the interactive viewing of videos, we introduce the concept of a *Pre-fetch Delay Tolerance Quality of Service* (PDT QoS). A PDT QoS is specified for each stream that a video server supports, and it specifies the worst case pre-fetch delay that can be tolerated during interactive viewing. It is shown that a PDT QoS specified for a stream s that retrieves video $v(s)$ is equivalent to placing a lower bound on the bandwidth that can be reserved for stream s . The lower bound for the bandwidth can be determined from the buffer-bandwidth resource relation of video $v(s)$.

The primary strength of MR retrieval is the flexibility to optimally trade bandwidth and buffer. This is captured in the buffer-bandwidth resource relation. While CD and CT retrieval are each represented by a single point on the resource relation for the stored video, MR retrieval can operate at multiple operating points on the resource relation. MR retrieval can set any bandwidth reservation. As the bandwidth reservation is

reduced, it is necessary to increase the reserved buffer. Consider a video server that uses MR retrieval. We present two cases to demonstrate the advantage of using MR retrieval over CD or CT retrieval.

Case 1. Assume that initially, each stream has a bandwidth reservation equal to the peak data rate of the video being retrieved. Assume that the total bandwidth reserved for all streams is equal to the total bandwidth of the video server. Assume that a large buffer memory exists in the video server. Using CT retrieval, no more streams can be supported by the video server because of the bandwidth limitation. In MR retrieval, if all viewers can tolerate a pre-fetch delay, the bandwidth reserved for all the streams can be substantially reduced from the peak bandwidth. Reducing the reserved bandwidth for each stream requires an increase in pre-fetch buffer requirements for each stream, if continuous, lossless retrieval is to be guaranteed. In this way, memory resources can be utilized to alleviate the I/O bandwidth bottleneck. By reducing the total bandwidth reserved for all the streams, the video server can potentially increase the number of streams that are supported.

Case 2. Assume that initially each stream has a bandwidth reservation equal to the average data rate of the video being retrieved. Each stream has a pre-fetch buffer requirement. Assume that the total buffer memory reserved for all streams is equal to the total memory resource of the video server. However, assume that total bandwidth reserved for all streams is less than the total disk bandwidth of the video server. Using CD retrieval, no more streams can be supported by the video server because of the memory limitation. In MR retrieval, by increasing the bandwidth reserved for each stream, the memory requirement for each stream can be substantially reduced. In this way, the bandwidth resources can be utilized to alleviate the memory bottleneck of a video server. By reducing the total memory reserved for all the streams, the video server can potentially support more streams.

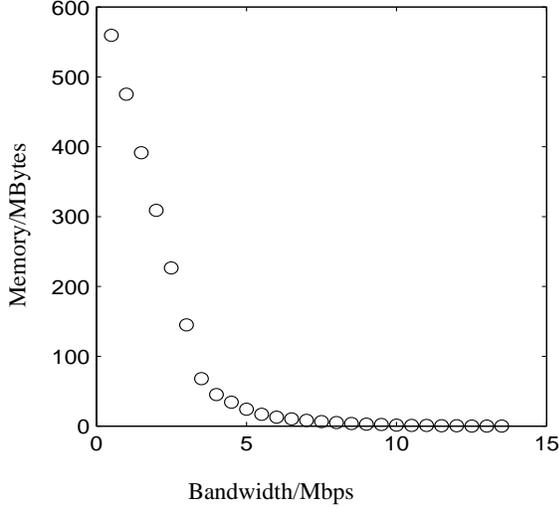


Fig. 10. Resource relation of MPEG-2 VBR video.

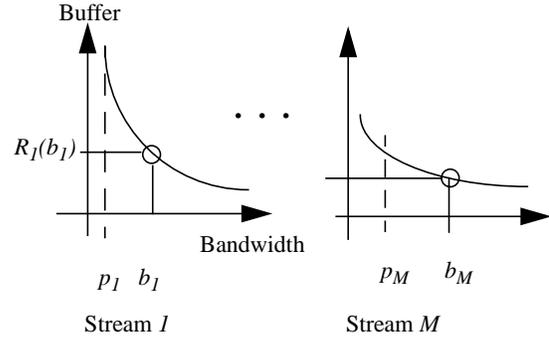


Fig. 11. Resource reservation problem.

IX. Resource Reservation for MR Retrieval

In the previous section we developed and presented the MR retrieval schedule. It was found that the worst case memory buffer requirement for the retrieval of a VBR video decreases as the bandwidth reservation increases. It was seen that MR leads to a buffer-bandwidth resource relation. In this section, we develop the resource reservation algorithm based on MR retrieval for multiple streams in a video server.

In a video server, data for *multiple, concurrent* streams is retrieved from the disk system to memory and then transmitted into the network. The video server has resources of disk bandwidth and memory that have to be shared amongst all streams. If MR is used for the retrieval of each stream, the important question remains as to what buffer and bandwidth reservations should be made for each stream i.e. the operating point on the resource relation of the video retrieved by each stream must be determined. For each incoming stream, the reservations should be made to maximize the number of streams that can be supported by a video server while guaranteeing the continuous, lossless retrieval and PDT QoS of each stream. Before describing the reservation problem, we present some definitions in table 3.

Table 3: Notation for resource reservation

| | |
|--|------------------------------|
| M | Total number of streams |
| $k \in \{1, \dots, M\}$ | Stream index |
| B_I | Bandwidth increment |
| $b_k \cdot B_I, b_k \in \{1, 2, \dots\}$ | Stream bandwidth reservation |

Table 3: Notation for resource reservation

| | |
|--|--|
| $b_k^o \cdot B_I, b_k^o \in \{1, 2, \dots\}$ | Optimal stream bandwidth reservation |
| p_k | Lower bound for stream bandwidth reservation |
| q_k | Peak bandwidth of video accessed by stream k |
| $R_k(b_k)$ | Stream buffer-bandwidth resource relation |
| d_k | Stream PDT QoS |
| C_M | System memory resource constraint |
| C_B | System disk bandwidth resource constraint |
| $\vec{B} = (b_1, \dots, b_M)$ | Reservation vector for all streams |
| $\vec{B}^o = (b_1^o, \dots, b_M^o)$ | Optimal reservation vector for all streams |
| S | Reservation vectors with $p_k \leq b_k, k = 1, \dots, M$ |

Note that $R_k(b_k)$ is determined for a video stream by the MR retrieval schedule. In other words, for a given bandwidth b_k , we find the worst case buffer requirement $R_k(b_k)$ for the video stream based on the MR retrieval schedule.

The lower bound on the stream bandwidth reservation is determined as follows:

$$p_k = \text{MIN} \{ b_k \}, \text{ subject to } \frac{R_k(b_k)}{b_k} \leq d_k,$$

where $\frac{R_k(b_k)}{b_k}$ is the pre-fetch delay.

Problem formulation

Our objective in a video server is to maximize the number of streams that can be supported. Therefore, we formulate the resource reservation problem as follows:

For a given set of streams, determine if there exists a reservation vector \vec{B} for all streams that satisfies the following constraints:

$$M_T = \sum_{k=1}^M R_k(b_k) \leq C_M \text{ (memory constraint),}$$

$$B_T = \sum_{k=1}^M b_k \leq C_B \text{ (bandwidth constraint),}$$

$$\vec{B} \in S \text{ (PDT QoS constraint).}$$

If the reservation vector exists, the set of streams can be supported by a video server, otherwise the set of streams cannot be supported. Note that in an actual system, the computation of the total bandwidth constraint is not a simple summation as shown above, since the SCAN disk head schedule is assumed. The actual computation is an approximation based on simplifying assumptions [2, 9].

In the SCAN scheduling algorithm, the scanning cycle consists of two phases. During the first cycle, the head scans the disk from the inner most track to the outer most track. While scanning the disk, data blocks belonging to different streams are read from the disk. Upon reaching the outer most track, the head is returned to the initial position. Several assumptions are made in the above analysis. Firstly, any stream accessed during the first phase will add to the total retrieval cycle the maximum rotational latency, the data reading time and the minimum seek time. Secondly, since the retrieval cycle consists of two phases of head movement, we add two maximum seek delays to the total cycle time. For the SCAN scheduling algorithm, we determine if the bandwidth constraint is satisfied by calculating the total cycle time T required for all the bandwidth reservations, as follows:

$$T = \sum_{k=1}^M \left(\frac{t_c \cdot b_k}{n_d \cdot r_d} + t_{rx} + t_{sm} \right) + 2 \cdot t_{sx}$$

t_c is the round robin cycle time, t_{sx} is maximum seek time, t_{sm} is minimum seek time and t_{rx} is maximum rotation latency. r_d is maximum disk transfer rate and n_d is the total number of disks in an array. For the bandwidth constraint, we require that $T \leq t_c$. This is equivalent to the constraint $B_T \leq C_B$.

The resource reservation problem above can be shown to be equivalent to the following two step algorithm:

1) Find the reservation vector \vec{B}^o which is the solution to the following constrained minimization problem:

$$\text{MIN } M_T = \sum_{k=1}^M R_k(b_k) \text{ subject to } B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

2) If $M_T \leq C_M$ (system memory constraint), the set of all streams can be supported by the system, otherwise the set of streams cannot be supported.

Let A be the set of *all reservation vectors* that meet both the PDT QoS constraints $\{d_k, k = 1, \dots, M\}$ and the system bandwidth constraint C_B specified above. The non-linear minimization problem gives us \vec{B}^o , which is the reservation vector in A that minimizes the total system memory requirement M_T . If M_T is greater than the system memory constraint, then there can be no reservation vector in A that will also meet

the memory resource constraint C_M . This is because the reservation vector \vec{B}^o is the vector in A that *minimizes* the memory requirement. This means that there can be no reservation vector that meets all system resource constraints C_M , C_B , and the PDT QoS constraint. Therefore the two step algorithm is equivalent to the optimal resource reservation algorithm.

Optimal resource reservation algorithm

In this section we outline the main approach we used in finding an optimal resource reservation algorithm. The algorithmic solution to the optimal resource reservation problem is based on a version of the Lagrange multiplier method as applied to bit allocation problems in video coding [11]. The approach places no restriction on the form of the stream resource relations e.g. the resource relations need not be strictly convex. In this section, we only present the main theorem associated with this method. In appendix A, we present the development of the optimal algorithm for completeness.

As part of the two step resource reservation algorithm we have to solve the following *constrained* minimization problem:

$$MIN M_T = \sum_{k=1}^M R_k(b_k), \text{ subject to } B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

The solution to this constrained minimization problem is based on the following theorem:

Theorem: For any $\lambda \geq 0$, the solution $\vec{B}^o(\lambda)$ to the *unconstrained* minimization problem

$$MIN \left\{ \sum_{k=1}^M R_k(b_k) + \lambda \sum_{k=1}^M b_k \right\}, \text{ subject to } \vec{B} \in S$$

is also the solution to the *constrained* minimization problem with the constraint

$$C_B^o(\lambda) = \sum_{k=1}^M b_k^o(\lambda).$$

The proof of this theorem is presented in [11]. This theorem does not guarantee a solution to the constrained minimization problem. It just says that for every λ , there is a corresponding constrained problem for which the solution is the same as that of the unconstrained problem. The main point is that if $C_B^o(\lambda)$ happens to equal C_B , then $\vec{B}^o(\lambda)$ is the desired solution to the original constrained problem.

In solving the constrained problem, we can sweep over λ from 0 to ∞ and try to find a corresponding $C_B^o(\lambda)$ that equals C_B . A typical dependency of the solution rate $C_B^o(\lambda)$ on the lagrange multiplier λ can be shown to have the form of figure 12. It shows a decreasing staircase curve where the discontinuities correspond to singular values of λ .

The key question then becomes the problem of finding the corresponding λ efficiently. This forms the main portion of the optimal resource reservation algorithm. The development of the algorithm and the optimal resource reservation algorithm is presented in appendix A.

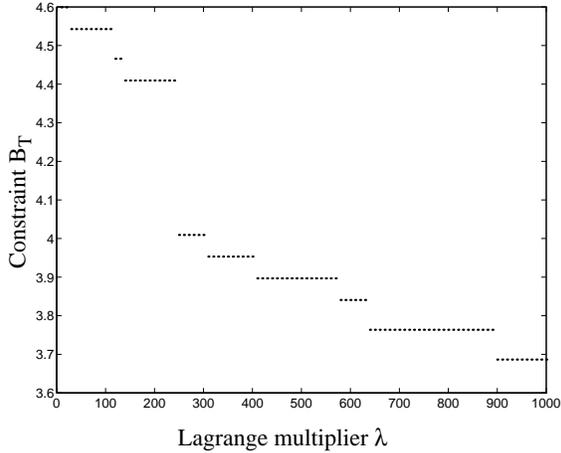


Fig. 12. Variation in constraint function with λ .

Heuristic resource reservation algorithm

The optimal resource reservation algorithm presented above gives us an upper bound on the performance of the video server architecture proposed in this paper. In [10] we presented a low complexity heuristic resource reservation algorithm. The algorithm is presented here, and in section 11, we compare the performance of this algorithm with the optimal resource reservation algorithm presented above. We found by simulation that the performance of this algorithm typically performs within 10% of the optimal resource reservation algorithm, although it is much lower in complexity than the optimal algorithm. Although, the optimal algorithm only performs typically 10% better than the heuristic algorithm, it was important to develop the optimal algorithm for two reasons. Firstly, solving the optimal resource reservation problem gave us an in-depth understanding of the resource reservation process. Secondly, the optimal algorithm servers as a benchmark to compare the performance of any heuristic algorithm.

- 1) For all video streams, the retrieval bandwidths \vec{B} are initially set to the peak data rate of the videos.
- 2) Compute total memory and bandwidth requirement:

$$M_T = \sum_{k=1}^M R_k(b_k), B_T = \sum_{k=1}^M b_k$$

- 3) If $M_T > C_M$: the video server *cannot* support all the video streams specified (memory limited). Stop the algorithm.

4) If $M_T < C_M$:

If $B_T \leq C_B$: The video server *can* support all the video streams specified. Stop the algorithm.

If $B_T > C_B$: Reduce total bandwidth by increasing buffer requirements (step 5).

5) Reduce total bandwidth.

First find the video stream for which there is a minimal increase in buffer with an incremental decrease in bandwidth. Define g_k as the incremental increase in buffer of each video stream k with an incremental decrease in bandwidth of the bandwidth reservation for the video stream.

$$g_k = R_k(b_k - 1) - R_k(b_k) , G = \{k; p_k < b_k\}$$

G defines the set of all streams for which the bandwidth reservation can be incrementally decreased. We find the video stream k_{min} for which the above is minimized i.e.

$$g_{k_{min}} = \min \{g_k\} , k \in G$$

If the set G is null, then the set of video streams cannot be accommodated by the video server. Stop the algorithm.

If a minimum is found, step 2 is repeated with the following resource reservation:

$$\vec{B} = (b_1, \dots, b_{k_{min}} - 1, \dots, b_M)$$

X. Scalable Video

In this section we integrate scalable video with MR retrieval and the resource reservation algorithm to present a scheme for the *progressive display* of scalable video. Progressive display improves the performance of a video server supporting interactive viewing of video.

Compared to simulcast coding, scalable coding schemes can provide multiple levels of video with a minimal cost of extra bandwidth or storage capacity. In scalable video coding, subsets of the full resolution bit-stream are used to obtain subsets of the full resolution video [4, 7, 9]. Scalable video will be used in advanced computer networks to support heterogeneous clients. Mobile wireless clients may only have computing resources to receive the lowest layer of video, while high performance workstations will request all the scalable layers of video.

The MPEG2 standard allows a combination of *spatial*, *SNR (signal-to-noise ratio)* and *temporal scalability* for up to three layer coding of video sequences. In one possible hybrid, three layer scalable coding scheme, the base layer provides the initial resolution of video. The spatial enhancement layer enables the upsampling and hence increase in frame size of the base layer. Finally, the SNR enhancement layer increases the visual quality of the (base+spatial enhancement) layers of video [7]. In another scheme for MPEG-2 video, three layer temporally scalable video is achieved as follows. The lowest scalable layer is

comprised of the I frames of a video (I layer). The P frames (P layer) enable an increase in the temporal resolution of the I frame layer. Finally, the B frames (B frames) increase the temporal resolution of the I+P layers. We refer to this as IPB scalable video. In this scheme, scalability is inherently provided by the MPEG-2 encoding structure. We have developed and tested a simple but robust system to extract the I, P, B layers from an MPEG-2 sequence and also to combine the multiple layers (I, I+P, I+P+B) for decoding. There are various ways for scalable video data to be placed on disks [4, 9]. In this research, we assumed that each scalable layer is stored separately as an independent ‘video’ and that each layer is interleaved over all disks.

In the progressive display of scalable video for interactive viewing, a progressively increasing PDT QoS is specified for the progressively higher scalable layers of a video. Each scalable layer is considered as an independent video. This is in contrast to non-progressive display of scalable video in which a single PDT QoS is specified.

In progressive display, the pre-fetch data for progressively higher layers are retrieved simultaneously since each layer is considered to be an independent video. At any given time, a video is transmitted only with all the scalable layers for which the pre-fetch data have been fully retrieved. For example, suppose that transmission is to be resumed at some point of a video after an interactive function. If enough time has elapsed only for the pre-fetch data of the lowest scalable layer to have been retrieved, only the lowest scalable layer is transmitted. If enough time has elapsed for the pre-fetch data of the first two layers to have been retrieved, the first two scalable layers are transmitted.

Progressive display improves the performance of the video server supporting scalable video. For scalable video, let the lowest layer of video have a PDT QoS of l sec. The higher layers will have PDT QoS values larger than l sec. In non-progressive display of scalable video, to achieve the same degree of interactivity, all the scalable layers have the same PDT QoS value of l sec. We will demonstrate the performance improvement in the next section.

XI. Performance Evaluation

For performance evaluation, we used the disk performance characteristics of three disk systems as shown in table 4. Disk system 1 has the disk performance characteristics of a current magnetic disk. In disk system 2, the performance parameters were improved by a factor of two to project the performance characteristics of the next generation of magnetic disk systems. For performance evaluation, trace data for MPEG2 scalable and non-scalable video was obtained using Columbia’s full-profile, standard-conforming MPEG2 software encoder/decoder [13].

In the following simulations, the video server receives requests for videos from clients. Each new request specifies a certain video which is stored on the video server, for which there exists a resource relation. Each

request also has an associated PDT QoS. For each new request, the video server determines if it can accept the request or not. If the video server can accept the new request, a stream is established for the new request. For each simulation the total number of video streams that can be supported by the video server is found for a given set of available video server resources. The simulations find the total number of admissible streams to the video server system as the on-board memory resource is increased, while maintaining a fixed disk bandwidth resource.

Performance comparison of MR and CT retrieval

Figure 13 and 14 compares the performance of MR and CT retrieval scheduling. Each line corresponds to a single simulation. In each simulation, all streams are accessing the same MPEG-2 VBR video which has the trace data shown in figure 1. Also, in each simulation, each stream specifies the same PDT QoS value. For disk system 1, if a video server has 120 MB, MR retrieval supports 50% more streams than CT retrieval, if users can tolerate a pre-fetch delay of 10.0 sec.

In the case of MR retrieval scheduling, we used the heuristic resource reservation algorithm to maximize the number of streams that can be supported concurrently by a video server. In the case of CT retrieval, the resource reservation algorithm is based on the fact that each stream requires a bandwidth reservation equal to the peak data rate of the requested video. Figures 13 and 14 show the total number of admissible video streams at the video server system as the total memory resource is increased, while keeping the disk system the same.

For MR retrieval, we can see that the number of streams that can be supported increases as the video server memory resources are increased. For continuous, lossless retrieval in interactive viewing, the resource reservation algorithm based on MR retrieval guarantees that no other retrieval schedule can support more video streams for a given set of video server resources.

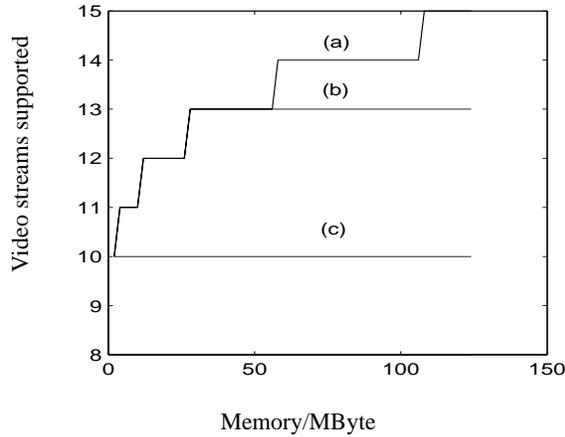
It can be seen that CT retrieval cannot take advantage of any increase in the memory resource of a video server. The advantage of the CT retrieval schedule is that the PDT QoS is always zero. This does not mean that the total delay that the client experiences before receiving its requested video is zero. However, the pre-fetch delay is zero. It can be seen that the performance of this scheme is the same as MR retrieval in which clients specify a PDT QoS of zero.

Table 4: Disk performance parameters

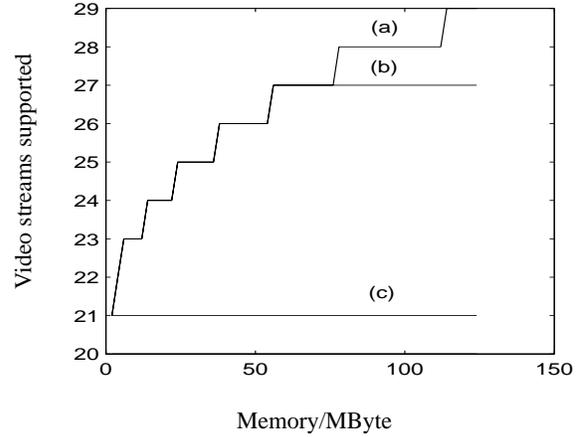
| Parameter | Disk system 1 | Disk system 2 |
|--------------------------|---------------|---------------|
| Disk cycle time/second | 0.5 | 0.5 |
| Max. rotation latency/ms | 14.2 | 7.1 |
| Max. seek latency/ms | 18.0 | 9.0 |
| Min. seek latency/ms | 1.5 | 0.75 |

Table 4: Disk performance parameters

| | | |
|------------------------------|------|-------|
| Max. disk transfer rate/Mbps | 60.0 | 120.0 |
| No. of disks in array | 4 | 4 |

**Fig. 13.** Performance comparison of MR and CT retrieval (disk system 1).

- (a) MR retrieval PDT QoS 10.0
- (b) MR retrieval PDT QoS 2.0
- (c) CT retrieval (PDT QoS 0.0)

**Fig. 14.** Performance comparison of MR and CT retrieval (disk system 2).

- (a) MR retrieval PDT QoS 10.0
- (b) MR retrieval PDT QoS 2.0
- (c) CT retrieval (PDT QoS 0.0)

Performance comparison of MR and CD retrieval

Figure 15 compares the performance of MR and CD retrieval scheduling. Each line corresponds to a single simulation run. In each simulation all streams are accessing the same MPEG-2 VBR video which has the trace data shown in figure 1. Also, in each simulation, each stream specifies the same PDT QoS value. For disk system 1, if a video server has 150 MB, MR retrieval supports 275% more video streams than CD retrieval. CD retrieval is memory bound.

In the case of MR retrieval scheduling, we used the fast heuristic resource reservation algorithm to maximize the number of streams that can be supported concurrently by a video server. In the case of CD retrieval, the resource reservation algorithm is based on two facts. Firstly, each stream requires a bandwidth reservation equal to the average data rate of the requested video. Secondly, there is a fixed memory requirement for the retrieval of the video. Figure 15 shows the total number of admissible streams at the video server system as the on-board memory resource is increased, while keeping the disk bandwidth the same.

We can see that the number of streams supported by CD retrieval is generally much lower than MR retrieval. This scheme is essentially *memory bound*. The bandwidth is not fully utilized since the memory

requirements are the limiting factor in the resource reservation.

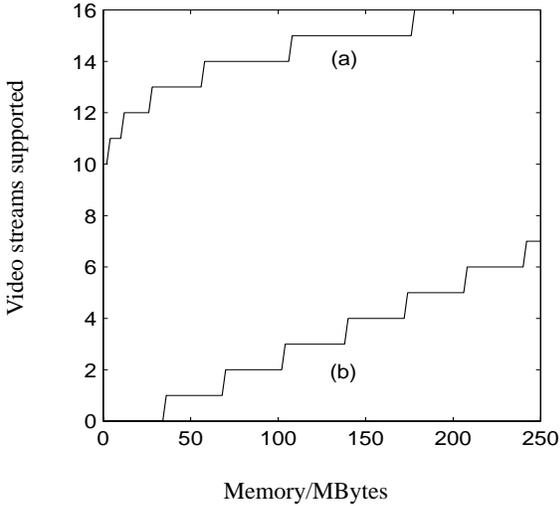


Fig. 15. Performance comparison of MR and CD retrieval (disk system 1).

- (a) MR retrieval (PDT QoS 135.0 sec)
- (b) CD retrieval (PDT QoS 135.0 sec)

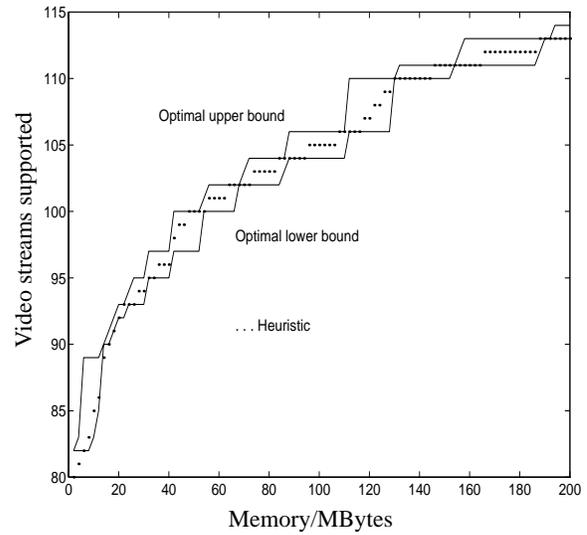


Fig. 16. Performance comparison of optimal and heuristic resource reservation algorithms.

Performance comparison of optimal and heuristic resource reservation algorithm

Figure 16 compares the maximum number of streams that can be supported concurrently by a video server for the optimal resource reservation algorithm and the heuristic resource reservation algorithm. The simulation parameters are given in table 5. The request pattern for videos is uniform. The videos are sequences from the movie ‘Forrest Gump’ and ‘Ben Hur’. For each request for a video, the mean PDT QoS is shown in table 5. For this simulation, we used the performance characteristics of disk system 2, however, we considered a large scale disk system with 16 disks.

For the optimal algorithm the exact optimal solution is not computed due to high computation requirements. Instead, an upper and lower bound on the number of admissible streams is found. The derivation of these bounds are discussed in appendix A.

Figure 16 shows that as the memory resource is increased, the number of streams also increases. It was found that the difference in the number of streams that can be admitted between the heuristic and optimal algorithm is less than 10% for this simulation. Therefore, the performance of the heuristic algorithm is very close to that of the optimal algorithm.

Table 5: Simulation parameters

| | |
|---|------|
| No. of videos | 5 |
| Mean of Avg. bit rates of all videos/Mbps | 2.52 |
| Mean of peak rate of all videos/Mbps | 8.86 |
| Mean PDT QoS/sec | 5.0 |

Performance comparison of progressive and non-progressive display of scalable video

In figure 17, we compare the progressive display for the interactive viewing of scalable video with non-progressive display of scalable video. All streams are retrieving the same MPEG-2 scalable video. The video sequence is from the movie ‘Ben Hur’ and the related information is shown in table 6. The request pattern for the scalable layers of the video is uniform. For progressive display, progressively higher PDT QoS values are specified for the progressively higher scalable layers (table 6). In non-progressive display, the same PDT QoS is specified for the full resolution of video to achieve the same degree of interactivity. Figure 17 shows that using progressive display for the interactive viewing of scalable video increases the number of streams supported by a video server. For disk system 1, if a video server has a memory resource of 120 MB, progressive display supports 17% more scalable layer streams than non-progressive display.

Table 6: Simulation parameters

| Scalable layer | Base | Spatial | SNR |
|--|------|---------|------|
| Avg. rate/Mbps | 0.7 | 0.44 | 2.5 |
| Peak rate/Mbps | 1.8 | 1.0 | 4.7 |
| Non-progressive display PDT QoS/Sec | 2.0 | 2.0 | 2.0 |
| Progressive display PDT QoS/Sec | 2.0 | 10.0 | 20.0 |

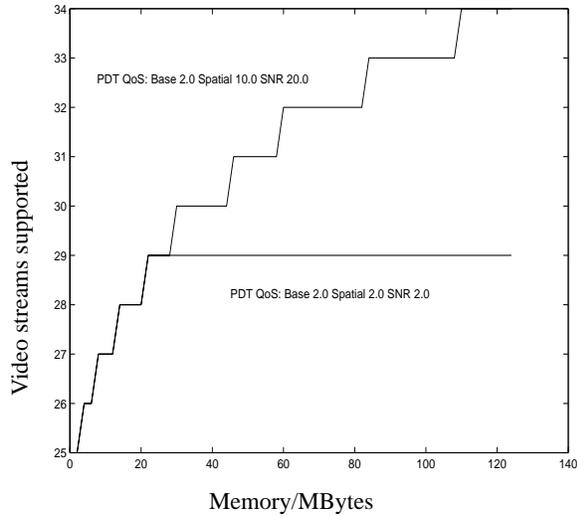


Fig. 17. Performance evaluation for progressive display of scalable video (disk system 1).

XII. Conclusions

In this research we have presented a new retrieval schedule for the retrieval of bursty VBR video data from the disk system to the memory of a video server. MR retrieval allows a range of bandwidths to be reserved for retrieval and guarantees a minimal buffer requirement for each bandwidth reservation. We presented a resource reservation algorithm for video server resources based on the MR retrieval schedule. We provided theoretic proofs for the optimality of the resource allocation algorithm. We also presented a fast, low complexity heuristic algorithm which approximates the theoretical optimal solution within 10%. The retrieval schedule and resource reservation algorithms are flexible enough to be implemented on general purpose computers. Performance evaluations based on simulations using MPEG2 trace data were presented. We found that the retrieval schedule and resource reservation algorithm dramatically improves the performance and flexibility of video servers compared to previous approaches.

To support heterogeneous clients, we also applied scalable video coding to our retrieval schedule and resource reservation algorithm. We proposed a new retrieval method for scalable video- progressive display, and showed a significant increase in system utilization and efficiency.

Appendix A Optimal Resource Reservation Algorithm for MR Retrieval

For the notation defined in table 3, the optimal resource reservation problem was shown in section 9 to be equivalent to the following two step algorithm:

1) Find the reservation vector \vec{B}^o which is the solution to the following non-linear constrained minimization problem:

$$\min M_T = \sum_{k=1}^M R_k(b_k)$$

Subject to

$$B_T = \sum_{k=1}^M b_k \leq C_B, \vec{B} \in S$$

2) If $M_T \leq C_M$ (system memory constraint), the set of all streams can be supported by the system, otherwise the set of streams cannot be supported.

The solution to the constrained minimization problem is based on the following theorem:

Theorem: For any $\lambda \geq 0$, the solution $\vec{B}^o(\lambda)$ to the *unconstrained* minimization problem

$$\min \left\{ \sum_{k=1}^M R_k(b_k) + \lambda \sum_{k=1}^M b_k \right\}, \text{ subject to } \vec{B} \in S$$

is also the solution to the *constrained* minimization problem with the constraint $C_B^o(\lambda) = \sum_{k=1}^M b_k^o(\lambda)$.

The proof of this theorem is presented in [11]. This theorem does not guarantee a solution to the constrained minimization problem. It just says that for every λ , there is a corresponding constrained problem for which the solution is the same as that of the unconstrained problem. The main point is that if $C_B^o(\lambda)$ happens to equal C_B , then $\vec{B}^o(\lambda)$ is the desired solution to the original constrained problem.

In solving the constrained problem, we can sweep over λ from 0 to ∞ and try to find a corresponding $C_B^o(\lambda)$ that equals C_B . The key question then becomes the problem of finding the corresponding λ efficiently. This forms the main portion of the optimal resource reservation algorithm.

The crucial point in solving the unconstrained problem is that the solution is obtained by minimizing each term of the sum separately.

If $\vec{B}^o(\lambda)$ is the solution to the optimization problem, we denote by $b_k^o(\lambda)$ the k th component of $\vec{B}^o(\lambda)$, and define $S_k = \{p_k, p_k + 1, \dots, q_k\}$. Then $b_k^o(\lambda)$ solves

$$\min \{R_k(b_k) + \lambda \cdot b_k | b_k \in S_k\}.$$

Given λ , one can solve for all $b_k^o(\lambda)$, sum them all up to get $C_B^o(\lambda)$, and then compare this value to the desired constraint C_B . If $C_B^o(\lambda) = C_B$, the desired solution has been found. Note that $b_k^o(\lambda)$ may not be unique. Therefore, the solution $\vec{B}^o(\lambda)$ is also not necessarily unique.

The following points show that it is not necessary to sweep over all λ (all values on the non-negative real line) in order to find the solutions to the unconstrained problem. The proof of these points are given in [11]

- 1) $b_k^o(\lambda)$ either decreases or remains unchanged as λ increases.
- 2) Any given λ can either have a *singular* or *non-singular* solution $b_k^o(\lambda)$ to the unconstrained problem. Singular points are values of λ for which more than one solution exists. Non-singular points only have one solution.
- 3) Two adjacent singular values have one and only one solution in common.
- 4) All non-singular values of λ between two adjacent singular values have the same (single) solution.

A typical dependency of the solution rate $C_B^o(\lambda)$ on the lagrange multiplier λ can be shown to have the form of figure 12. It shows a decreasing staircase curve where the discontinuities correspond to singular values of λ .

The main conclusion from the above points is that to find a constraint which satisfies $C_B^o(\lambda) = C_B$, it is not necessary to search over all singular λ . It is sufficient to check only the singular points. Locating the desired solution requires the knowledge of the singular λ . The following lemma are used to locate all the singular λ .

Lemma 1: Let λ_1 be singular, $\lambda > \lambda_1$, and $\vec{B}^o(\lambda)$ be a common solution associated with λ and λ_1 . Define the set

$$S_k^I = \{b_k^o + 1, \dots, q_k\}.$$

This set is non-empty if $b_k^o + 1 \leq q_k$. Also, define the set

$$M^I = \{k \in \{1, \dots, M\} | S_k^I \neq \emptyset\}.$$

This defines the set of all resource relations for which an increase in bandwidth is possible. Then, the singular value

λ_1 , which is necessarily the closest to λ from below is given in terms of λ by

$$\lambda_1 = \max \left\{ \frac{R_k(b_k^o(\lambda)) - R_k(b_k)}{b_k - b_k^o(\lambda)} \mid k \in M^I, b_k \in S_k^I \right\}.$$

Lemma 2: Let λ_2 be singular, $\lambda < \lambda_2$, and $\vec{B}^o(\lambda)$ a common solution associated with λ and λ_2 . Define the set

$$S_k^D = \{p_k, \dots, b_k^o - 1\}.$$

This set is non-empty if $b_k^o - 1 \geq p_k$. Also, define the set

$$M^D = \{k \in \{1, \dots, M\} | S_k^D \neq \emptyset\}.$$

Then, the singular value λ_2 , which is necessarily the closest to λ from above is given in terms of λ by

$$\lambda_2 = \min \left\{ \frac{R_k(b_k) - R_k(b_k^o(\lambda))}{b_k^o(\lambda) - b_k} \mid k \in M^D, b_k \in S_k^D \right\}.$$

The last two lemmas say that by knowing a singular value, the next from below or from above can be found, and thus, all singular values and all solutions $\vec{B}^o(\lambda)$ can be located.

We can now present the optimal resource reservation algorithm used in section 9:

- 1) Obtain an initial value of λ .
- 2) Solve the unconstrained problem. If λ is not singular, there is only one such solution and one constraint $C_B^o(\lambda)$. If λ is singular, then there are at least two different solutions. Find the two solutions from $\{\vec{B}^o(\lambda)\}$ with greatest and smallest constraints denoted by $C_B^l(\lambda)$ and $C_B^h(\lambda)$, respectively.
- 3) If the desired constraint C_B is such that $C_B^l(\lambda) \leq C_B \leq C_B^h(\lambda)$, then obtain all solutions in $\{\vec{B}^o(\lambda)\}$ and find the one for which the constraint, denoted by $C_B^a(\lambda)$, is the closest to C_B from below. If $C_B = C_B^a(\lambda)$, an exact optimal solution has been found. If not, an approximate solution has been found. Go to step 6.
- 4) If $C_B \leq C_B^o(\lambda)$ [or $C_B \leq C_B^l(\lambda)$], find the next singular λ using the current solution which corresponds to $C_B^o(\lambda)$ [or $C_B^l(\lambda)$]. Then, go to step 2.
- 5) If $C_B \geq C_B^o(\lambda)$ [or $C_B \geq C_B^h(\lambda)$], find the next singular λ using the current solution which corresponds to $C_B^o(\lambda)$ [or $C_B^h(\lambda)$]. Then, go to step 2.
- 6) If $M_T = \sum_{k=1}^M R_k(b_k^o) \leq C_M$ the set of all video streams can be supported by the video server, otherwise the set cannot be supported.

References:

1. E. Biersack, F. Thiesse, and C. Bernhardt, "Constant Data Length Retrieval for Video Servers with VBR Streams." Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems. June, 1996.
2. P. Bocheck, H. Meadows, and S.-F. Chang, "A Disk Partitioning Technique for Reducing Multimedia Access Delay." International Conference on Distributed Multimedia Systems and Applications. August, 1994.
3. A. J. Chaney, I. D. Wilson and A. Hopper, "The Design and Implementation of a RAID-3 Multimedia File Server." Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video. April. 1995.
4. E. Chang and Avidesh Zakhor, "Scalable Video Data Placement on Parallel Disk Arrays." SPIE Symposium on Imaging Technology. 1994.
5. E. Chang and A. Zakhor, "Admissions Control and Data Placement for VBR Video Servers." IEEE International Conference on Image Processing. 1994.
6. M. S. Chen, D. D. Kandlur, P. S. Yu, "Optimization of Grouped Sweeping Scheduling (GSS) with Heterogeneous Multimedia Streams." ACM Multimedia 93. 1993.
7. T. Chiang and D. Anastassiou, "Hierarchical Coding of Digital Television." IEEE Communications Magazine. May, 1994.
8. B. Ozden, R. Rastogi, A. Silberschatz, "Disk Striping in Video Server Environments." Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems. June, 1996.
9. S. Paek, P. Bocheck and S.-F. Chang, "Scalable MPEG2 Video Servers with Heterogeneous QoS on Parallel Disk Arrays." Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video. April. 1995.
10. S. Paek and S.-F. Chang, "Video Server Retrieval Scheduling for VBR Scalable Video." Proceedings of the Third IEEE International Conference on Multimedia Computing & Systems. June, 1996.
11. Y. Shoham and Allen Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers." IEEE Transactions on Acoustics, Speech, and Signal Processing. Vol. 36, No. 9. Sept. 1988.
12. H. Vin, P. Goyal, A. Goyal(1), and A. Goyal(2), "A Statistical Admission Control Algorithm for Multimedia Servers." ACM Multimedia 94. 1994.
13. Y. Yu, "Columbia MPEG Software Release 6.5 User's Manual." Technical report of Image & Advanced Television Laboratory, Columbia University. 1994.
14. S. Sahu, Z. Zhang, J. Kurose and D. Towsley. "On the Efficient Retrieval of VBR Video in a Multimedia Server." Proceedings of the IEEE International Conference on Multimedia Computing and Systems. June 1997.
15. J. D. Salehi, Z. Zhang, J. F. Kurose and D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements Through Optimal Smoothing." University of Massachusetts Technical Report TR 95-98. November, 1995.
16. J. Mc Manus, K. W. Ross, "Video on Demand over ATM : Constant-Rate Transmission and Transport", IEEE Journal on Selected Areas in Communications, Vol. 14, pp. 1087-1098, 1996.