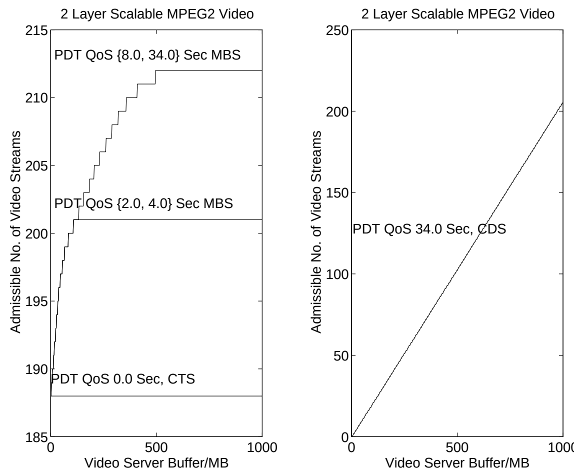

FIGURE 3: Performance evaluation

transfer bandwidth of a current disk system were improved by a factor of two to project the performance characteristics of the next generation of magnetic disk systems.

Table 1: Disk performance characteristics

Max. Seek Latency	9 ms
Min. Seek Latency	0.75 ms
Max. Rotation Latency	7.1 ms
Max. Disk Transfer Rate	120 Mbps

Figure 3 shows the maximum number of video streams of scalable video that can be supported concurrently by a video server using three different scheduling schemes: ***minimum buffer (MBS), constant data (CDS), and constant time (CTS)***. The figure shows the total number of admissible video streams to the video server system as the on-board memory resource is increased. The disk system based on the 16 disk array with performance characteristics as in table 1 is kept the same. Therefore, the performance evaluation shows the increase in the number of video streams as the memory resource of a video server is increased. This performance evaluation has two limitations. Firstly, we assumed that all clients access only one video with two scalable layers. This was due to the fact that we only had scalable MPEG2 trace data of one sequence. Secondly, we assumed that all the clients specify the same PDT QoS. In an actual video server system, all clients can select from a variety of videos with completely heterogeneous PDT QoS requirements. However, this simple performance evaluation demonstrates the main dynamics and advantages of this research.

We first consider the constant time schedule. It can be seen that the constant time schedule cannot take advantage of any increase in the memory resource of a video server. The advantage of this 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, but that any pre-fetch delays within the video server is zero. It can be seen that the performance of this scheme is the same as the minimum buffer schedule in which clients specify a PDT QoS of zero.

In the minimum buffer schedule, we can see that the number of video streams that can be supported by the video server increases as the video server memory resources are increased. This is due to the fact that the minimum buffer schedule alleviated the disk I/O bandwidth bottleneck by optimally utilizing the memory resource. For completely interactive and guaranteed retrievals, the optimal resource reservation framework guarantees that no other schedule can support more video streams for a given memory resource. We can also see that for two layer scalable video, increasing the PDT QoS values for the scalable layers can lead to a larger number of video streams supported by a video server. It is seen that for low PDT QoS values, increasing the memory resource may not lead to increases in the number of supported video streams, since the PDT QoS is a constraint on the feasible region of the resource reservation set.

Finally we can see that the number of video streams supported by the constant data schedule is much lower than the minimum buffer schedule. This scheme is essentially memory limited. The bandwidth is not fully utilized since the memory requirements are the limiting factor in the resource reservation and admission control.

6. Summary

In this research we have presented our results for the optimal retrieval scheduling of video data across the disk-memory interface of a video server. This retrieval schedule was shown to be the basis of a framework for the optimal utilization of video server on-board memory resources for the interactive retrieval of videos.

References:

1. S.- F. Chang, A. Eleftheriadis, D. Anastassiou, “Development of Columbia’s Video On Demand Testbed,” Image Communication Journal, Special issue on Video on Demand and Interactive TV, 1996.
2. E. Chang and A. Zakhor, “Admissions Control and Data Placement for VBR Video Servers”, IEEE International Conference on Image Processing, 1994.
3. H. Vin, P. Goyal, A. Goyal(1), and A. Goyal(2), “A Statistical Admission Control Algorithm for Multimedia Servers”, ACM Multimedia ‘94, San Francisco, USA, 1994.
4. T. Chiang and D. Anastassiou, “Hierarchical Coding of Digital Television,” IEEE Communications Magazine, May 1994.
5. S. Paek and S.-F. Chang, Video Server Retrieval Scheduling for VBR Scalable Video, CU/CTR/TR 429-95-35.
6. Y. Yu, “Columbia MPEG Software Release 6.5 User’s Manual,” Technical report of Image & Advanced Television Laboratory, Columbia University, 1994

$$M_t = \sum_{j=0}^{N_s-1} m[j], B_t = \sum_{j=0}^{N_s-1} b[j]$$

3. Admission Control.

(a) If $M_t > M^n$: the video server cannot support all the video streams specified (memory limited).

(b) If $M_t < M^n$:

If $B_t \le B^d$: The video server *can* support all the video streams specified.

If $B_t > B^d$: Reduce total bandwidth by increasing buffer requirements (step 4).

4. 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[j]* as the incremental increase in buffer of each video stream *j* with an incremental decrease in bandwidth Δb of the bandwidth reservation for the video stream.

$$g[j] = f^{v[j]}(b[j] - \Delta b) - f^{v[j]}(b[j])$$

$$G = \left\{ g[j]; \frac{f^{v[j]}(b[j] - \Delta b)}{b[j] - \Delta b} \le d[j] \right\}$$

G defines the set of the incremental increase in buffer for all video streams for which the PDT QoS will be maintained even for the new resource reservation defined by the incremental decrease in reserved bandwidth. We find the video stream for which the above is minimized i.e. $j_{min} = \underset{j}{Arg} \{Min(g[j])\}$.

If the set *G* is null, then the set of video streams cannot be accommodated by the video server. However, if a minimum is found, step 2 is repeated with the resource reservation of the appropriate video stream modified.

4. Scalable video

In this section we overview the MPEG2 scalable digital video technology and discuss how scalable video improves the performance of a video server in relation to the optimal resource reservation framework.

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 [4]. 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. With practical considerations and subjective evaluation, we chose a hybrid, three layer scalable coding scheme. In this hybrid 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.

The variable bit rate of scalable MPEG2 video is dependent on the encoding structure of scalable MPEG2 video. In the MPEG2 digital video technology, 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 interframe 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 scheduling scheme proposed in this paper is developed to efficiently retrieve such VBR video from the disk system of a video server. For performance evaluation of the proposed minimum buffer scheduling scheme, trace data for MPEG2 scalable video was obtained using Columbia's full-profile, standard-conforming MPEG2 software encoder/decoder [6].

In section 2, it was shown that each video sink retrieves a video object from the storage medium with an associated PDT QoS. In a video server with scalable videos, each client specifies a ***PDT QoS for each scalable layer*** of the video that it requests. The client can either specify the ***same*** PDT QoS for the different layers or specify ***progressively increasing*** PDT QoS values for the higher layers. The PDT QoS value sets a lower bound on the possible disk bandwidth reservations (and a corresponding upper bound for the on-board memory reservations) in the buffer-bandwidth relation of a scalable layer.

Consider a set of clients that concurrently request multiple scalable videos. Each client specifies a PDT QoS for each scalable layer of the video that it requests. Suppose that there is no feasible solution for the reservation of resources for the multiple videos in order to support all the clients concurrently. This means that there is no resource reservation set that can allow all the clients to be supported by the video server under the resource and PDT QoS constraints. If some clients increase the PDT QoS values for its higher layers while maintaining the same PDT QoS for its base layers, this may possibly lead to a feasible solution for the resource reservations, since the pre-fetch delay constraints are relaxed. In this way, each client can still achieve the same PDT QoS for its base layer, and the set of clients can be supported concurrently by the video server.

5. Performance evaluation

In this section, the optimal resource reservation algorithm based on the minimum buffer schedule is compared to current approaches.

The disk system used for the performance evaluation is a disk array with 16 disks, each with the performance characteristics as in table 1. The seek latencies and

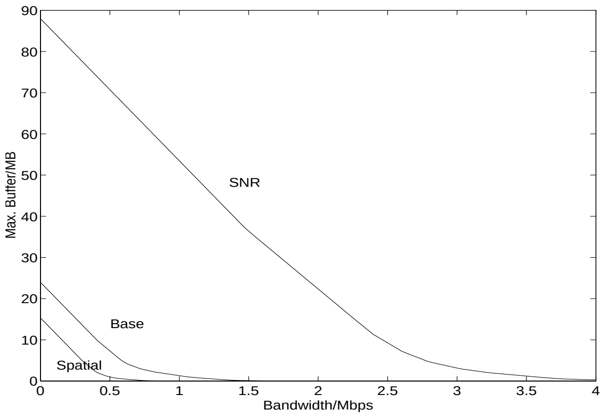

FIGURE 2: Buffer-bandwidth relation for scalable MPEG2 video based on minimum buffer scheduling

lem can be solved deterministically because the function $a_O(t)$ is known a priori for ***stored video data***. In brief, the minimum buffer schedule is based on the constant data schedule, with the difference that data is only retrieved ***just in time*** for consumption at the sinks. This removes the main disadvantage of the constant data schedule, in which data can be retrieved earlier than is required, leading to potentially large buffer requirements.

Figure 2 shows the buffer-bandwidth relation for the scalable video layers [4, 6] of an MPEG2 encoded video. From this relation, the corresponding pre-fetch delay tolerance (PDT) QoS for interactivity can be found. This QoS indicates the tolerance of the maximum pre-fetch delay for the video that is accessed. This value is usually specified by users or applications. The PDT QoS is derived directly from the buffer-bandwidth relation. If the buffer requirement for the retrieval of a given video is m_{max}, then the PDT QoS is m_{max}/b, where b is the corresponding reserved retrieval bandwidth.

The details of relating the retrieval schedule to the disk memory interface (including relevant disk system architecture considerations) are given in [5]. It is shown that the optimal retrieval schedule is directly applicable to RAID storage architectures. In brief, the storage medium corresponds to the disk system, the initial link corresponds to the disk I/O, the buffers correspond to the on-board memory of a video server, and the second link corresponds to the network bandwidth.

3. Optimal resource reservation

The buffer-bandwidth relation based on the optimal retrieval schedule developed in the previous section is the basis for an ***optimal resource reservation framework.*** The optimal resource reservation of resources is critical in maximizing the number of sinks that can concurrently retrieve video objects from the storage medium.

The minimum buffer schedule optimally minimizes the maximum buffer requirement for the retrieval of a video object, given that a fixed link bandwidth is reserved for the entire duration of interactive retrieval. The schedule gives us a *buffer-bandwidth relation* for the retrieval of each video object. This relation $m = f(b)$ indicates the buffer reservation m required for a video object, given that a bandwidth b is reserved for the retrieval of a video object. Because the minimum buffer schedule optimally minimizes the buffer requirement, the buffer-bandwidth relation f' of any other retrieval schedule is related as: $f'(b) \geq f(b)$, $b > 0$. The important question to answer now is to determine what bandwidths (and corresponding buffers) we should reserve for the ***multiple concurrent retrievals*** of video objects in the communication model of figure 1.

Consider a set of N_S video sinks with each video sink requesting the interactive retrieval of video object $v[j]$, $0 \leq j \leq N_S - 1$. Each requested video object by sink j has an associated buffer-bandwidth relation $f^{v[j]}$. We will also assume that each video sink has a PDT QoS requirement $d[j]$. The bandwidth, memory resource reservation for each video sink j that must be determined is: $\{b[j], m[j] = f^{v[j]}(b[j])\}$.

In this research, we present the optimal resource reservation framework that can be used to solve the optimization of different objective functions F. This framework is based on the buffer-bandwidth relation derived from the optimal retrieval scheduling scheme:

Determine $\{b[j], m[j] = f^{v[j]}(b[j])\}$, $0 \leq j \leq N_S - 1$ to minimize $F(b[0], m[0], \ldots b[N_S - 1], m[N_S - 1])$, subject to the constraints:

$$\sum_{j=0}^{N_S-1} m[j] \leq M^n \quad \textit{(buffer constraint)}$$

$$\sum_{j=0}^{N_S-1} b[j] \leq B^d \quad \textit{(link constraint)}$$

$$m[j]/b[j] \leq d[j] \quad \textit{(PDT QoS constraint)}$$

Note that if there is no feasible solution to the above optimization problem, then the simultaneous access to this set of data objects cannot be supported. Examples of possible objective functions could be to minimize the sum of the pre-fetch delays or to minimize the sum of total link bandwidth.

The algorithm for resource reservation to ***optimally minimize*** the sum of the pre-fetch delays using the resource reservation framework above is as follows:

1. For all video sinks, the reserved retrieval bandwidths are initially set to the peak data rate of the video objects.
2. Compute total memory and bandwidth requirement:

object beginning with the group n_s must consume $\{r[n_s], r[n_s+1], r[n_s+2], ...\}$ Mbits of data in consecutive time cycles. Since the constraints determine how much data each video sink must receive during each cycle, the scheduler transferring data to the video sinks is determined by the arrival time constraints. The research goal is to determine the optimal scheduling scheme for the retrieval of data out of the storage medium.

We will assume that the video sinks have ***complete interactive control*** over the retrieval of video objects. The sink can request the data transfer of a video object beginning at any group within the video object. Once retrieval has begun, the sink can also pause the retrieval and restart at any other group of the video object. We will assume that the retrieval scheduler must guarantee arrival time constraints under such conditions.

For ***guaranteed arrival time constraints***, it is necessary to reserve bandwidth and buffer resources for each sink that is retrieving a video object. If renegotiation of resources is not possible, the assumption that sinks have complete interactive control over the retrieval dictates that the resource reservation must be fixed for the entire duration of retrieval, and must be based on the worst case retrieval portion of the video object.

In order to present the optimality of the retrieval schedule, we first present two widely accepted strategies.

2.1 Constant time scheduling

Consider one sink requesting the retrieval of a video object which has time constraint $\{r[n] : 0 \le n \le N_g - 1\}$, starting at group n_s. If t_r is the time the sink requests the retrieval of video, this schedule [3] retrieves data corresponding to a constant time during each cycle i.e. $r[n_s]$, $r[n_s+1]$, $r[n_s+2]$,.. The start of data consumption at the sink is $t_s = t_r + t_c$. A bandwidth of b Mbps equal to the peak data rate of the video object must be reserved at the link for the entire duration of interactive retrieval: $b = MAX\{r[n] : 0 \le n \le N_g - 1\}/t_c$.

This bandwidth is reserved for the entire duration of retrieval and is not reduced since the retrieval is assumed to be completely interactive and the portion of video with the peak data rate can be requested at any time by the sink.

The advantage of this scheme is that there is only one cycle time of delay before the sink can start consuming data. Furthermore, the required buffer size is only $2 \cdot b \cdot t_c$ (assuming double buffering). The disadvantage is that the reserved bandwidth on the link is under utilized, since the data retrieved in each cycle is usually less than the peak data rate.

2.2 Constant data scheduling

In this schedule [2], a bandwidth of b Mbps equal to the average data rate of the video object is reserved at the link connected to the storage medium. The bandwidth is reserved for the entire duration of retrieval. This schedule always retrieves a fixed data amount $b \cdot t_c$ each cycle. In this scheme, the sink can start consuming data only after a variable pre-fetch delay. The pre-fetch delay is necessary to ensure that buffer starvation does not occur after the sink has started consuming data. Buffer starvation can occur because the amount of data consumed at the sink each cycle is variable, while the amount of data retrieved from the storage medium is constant.

2.3 Optimal retrieval scheduling

In this section we present a retrieval schedule that optimally minimizes the buffer required for the interactive retrieval of a video. For the optimal retrieval schedule we first define the following:

$$a_o(t) = \sum_{n=0}^{\lfloor (t-t_s)/t_c \rfloor} r[n_s + n], \; t \ge t_s$$

This function represents the total accumulated data **output** from the buffer to the sink during the time $[t_s, t_s + \lfloor (t-t_s)/t_c \rfloor \cdot t_c)$, where t_s is the time the sink starts to consume data from the buffer.

$$a_i(t) = \sum_{n=0}^{\lfloor (t-t_r)/t_c \rfloor} s[n], \; t \ge t_r$$

This function represents the total accumulated data **input** from the storage medium to the buffer during the time $[t_r, t_r + \lfloor (t-t_r)/t_c \rfloor \cdot t_c)$, where t_r is the time the sink requests the retrieval of data. $s[n]$ represents the retrieval scheduling scheme and is the amount of data transferred from the storage medium to the buffer during time cycle $[t_r + n \cdot t_c, t_r + (n+1) \cdot t_c)$.

We assume that a link bandwidth of b Mbps which is less than or equal to the peak data rate of the video object is reserved for the entire duration of retrieval. The minimum buffer retrieval schedule optimally minimizes the buffer requirement for the entire retrieval. The minimum buffer schedule is the solution to the following optimization problem:

Define $m_{max}(n_s) = MAX\ \{a_i(t) - a_o(t)\}$,
$t_r \le t \le t_s + (N - n_s + 1) \cdot t_c$

Determine $s[n]$, $0 \le n \le N - n_s$ to minimize $m_{max}(n_s)$, subject to the constraints:
$a_i(t) - a_i(t - t_c) \le b \cdot t_c$ *(link bandwidth constraint)*
$a_i(t) \ge a_o(t + t_c)$ *(video object time constraint)*

$m_{max}(n_s)$ is defined as the maximum buffer requirement for the retrieval of the video object starting at group n_s. For the entire video object, we define the maximum buffer requirement for interactive retrieval as $m_{max} = MAX\ \{m_{max}(n_s) : 0 \le n_s \le N_g - 1\}$. The details of the minimum buffer schedule are presented in [5]. The key point is that the above optimization prob-

IEEE International Conference on Multimedia Computing & Systems 1996. Hiroshima, Japan.

Video Server Retrieval Scheduling for Variable Bit Rate Scalable Video†

Seungyup Paek and Shih-Fu Chang

Image & Advanced Television Laboratory & Department of Electrical Engineering,
Columbia University, New York, NY 10027, USA
E-mail: {~syp, ~sfchang}@ctr.columbia.edu

Abstract

In advanced multimedia networks, video servers with real time operating systems and high performance storage architectures will deliver multiple concurrent video streams to clients. The video server has a limited disk I/O bandwidth. This research presents a new framework for ***optimally utilizing*** *the on-board memory to alleviate the disk bandwidth bottleneck. The optimal utilization of the memory resource is achieved by using an* ***optimal retrieval schedule*** *for video data across the disk-memory interface. The optimal retrieval schedule is directly applicable to video servers that are based on general storage architectures (e.g. RAID). Performance evaluation using real MPEG2 trace data verifies that this approach can provide large increases in the number of supported video streams.*

1. Introduction

Usually, digital video compression/encoding techniques result in variable bit rate (VBR) video. The problem with retrieving multiple concurrent VBR videos from disk systems is that reserving disk bandwidths based on peak data rates can lead to under utilization of the disk system, while reducing the bandwidth reservation for each video stream can either lead to disruptions in the continuity of video presentation or large pre-fetch delay requirements.

This research presents a framework for ***optimally utilizing*** an arbitrary amount of on-board memory to alleviate the disk bandwidth bottleneck in a video server. We show that the optimal utilization of the memory resource depends directly on an ***optimal retrieval schedule*** for video data across the disk-memory interface.

The optimal resource reservation (utilization) of memory maximizes the number of video streams supported by a video server under the constraints of video server resources (bandwidth, memory) and the clients' tolerance to interactivity delays. We also show how scalable video relates to the optimal resource reservation framework.

This work is part of the research and development of advanced video servers for the Video-on-Demand testbed at Columbia University [1].

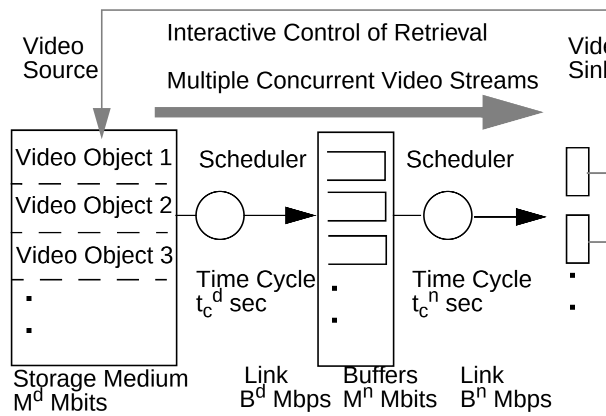

FIGURE 1. Communication model for retrieval scheduling in video servers

2. Retrieval scheduling in video servers

In this section a new optimal approach for retrieval scheduling in the disk-memory interface of video servers for the ***interactive viewing*** of video is presented. We first present the retrieval scheduling scheme for the general communication model of figure 1, and then relate it to the disk-memory interface of a video server.

In the model, multiple real time video objects in the storage medium have to be transferred to multiple video sinks via two links in series (one from the disk to memory and another from the memory to clients through a network). Each link has an associated retrieval scheduler that services multiple video streams in a round robin fashion. The time cycles for each link specify the time interval of one round robin cycle. For simplicity of notation, we will assume that the time cycles are the same and equal to t_c.

Each video object can be considered to be composed of groups of image frames of variable size. The arrival time constraint at the sink $\{ r[n] : 0 \le n \le N_g - 1 \}$ specifies the data amount of each group that must arrive at the sink during consecutive time cycles (N_g represents the total number of groups). A video sink starting to consume the video

† The long version of this paper can be obtained from the web page http://www.ctr.columbia.edu/~syp.

This work was supported in part by the National Science Foundation under a CAREER award, and the ADVENT project of Columbia University.