Scalable MPEG2 Video Servers with Heterogeneous QoS on Parallel Disk Arrays.

Seungyup Paek, Paul Bocheck, and Shih-Fu Chang.

Department of Electrical Engineering & Center for Telecommunications Research,

Columbia University, New York, NY 10027.

{syp, bocheck, sfchang} @ctr.columbia.edu

ABSTRACT

In this paper we focus on the video storage unit of a video server. We present a new, flexible data placement strategy for independent parallel disk arrays. The trade-off between utilization efficiency and interactive delay is investigated for this data placement strategy. Based on this trade-off, we show the advantage of video servers supporting a range of *interactivity QoS*. For our data placement strategy, we show that using scalable video improves the utilization and interactivity performance of a video server. We use three-layer, scalable MPEG2 digital video to support *resolution QoS* at a video server. Finally, we show that the data placement strategy reduces the complexity of admission control at the video server to that of a single disk system.

1. Introduction

In designing Columbia's Video On Demand testbed system [1,2] we are investigating advanced image and video technologies as components of a VoD system. One critical component is the design of an optimized real time video storage unit in the video server. Previously, we presented a disk partitioning technique to reduce the access delay for a single disk based single resolution video storage unit [5]. In this paper, we present a flexible strategy for the placement of video data on a parallel array of disks. For this placement strategy, we show that using scalable video improves the overall utilization and interactivity performance of a video server. Work on scalable video data placement in which utilization of the disk system is maximized is studied in [4]. However, the proposed scheme incurs large maximum start up and interactivity delays. In [3], a multiresolution video data placement scheme is presented in which fewer disks service low resolution requests and all disks in a parallel array service high resolution requests. We show that the performance of the low resolution requests is the same as in [4], whereas the performance of high resolution requests is not maximized. In real time retrieval of multiple video streams, we show that there is a trade-off between maximum interactive delay and utilization of disks. For the placement strategy that we present, different videos can have a range of interactivity and utilization performance based on this trade-off. On one end of the spectrum, the utilization of the disks is maximized (hence increasing the number of concurrent connections) and on the other end, the maximum interactivity delay is minimized. The flexibility of our strategy is that different videos can operate at different points of this retrieval performance spectrum to provide a range of interactivity QoS. We investigate the use of scalable video in a video server. It is shown that using scalable video based on the proposed data placement strategy improves the overall utilization and interactivity performance of a video server. The data placement strategy is optimized for the MPEG2 video coding structure (conforming to both main profile and high profile of MPEG2) to reduce the quality degradation during congestion. An admission control strategy based on the proposed data placement strategy is developed. It is shown that our data placement strategy reduces the complexity of admission control to that of a single disk system.

Section 2 describes the MPEG2 scalable video compression technology [7]. We show the advantage of supporting video resolution QoS at a video server. Section 3 briefly describes the system operation of the video storage unit. In section 4, a new strategy for the placement of video data on a parallel array of disks is presented. The trade-off relations based on this placement strategy show the advantage of supporting interactivity QoS in video servers. It is shown how scalable video can improve the overall utilization and interactivity performance of a video server based on the proposed data placement strategy, which provides further advantages of supporting resolution QoS in video servers. Finally, we present the admission control strategy for the video server based on the placement strategy that is presented.

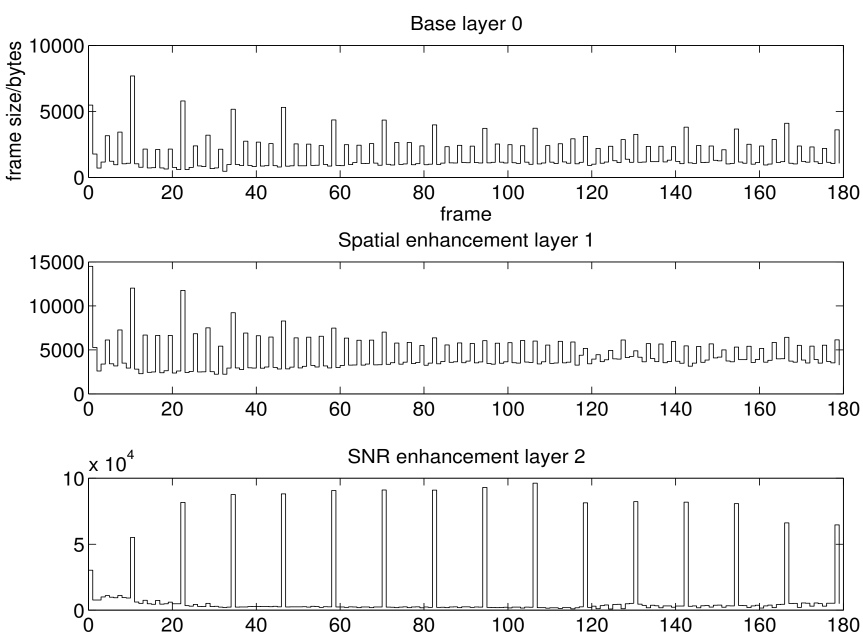

Figure 1. Data trace of scalable MPEG2 digital video.

2. MPEG2 Scalable Digital Video for Video Resolution QoS

In this section we overview the scalable MPEG2 digital video technology that we propose to use in our video server, and we show the advantage of supporting video resolution QoS based on scalable video at a video server. In section 4 we present further advantages of using scalable video in video servers. In scalable MPEG2 video coding, a subset of the full resolution bitstream can be used to obtain a subset of the full resolution video [6]. The MPEG2 standard allows a hybrid spatial and SNR scalability for three layer coding of video sequences. In such a scheme, the base layer provides the initial resolution of video, an additional spatial enhancement layer allows for the upsampling and hence increase in frame size of the base layer, and a further SNR enhancement layer provides for an increase in the visual quality of the base+spatial enhancement layers of video. For our video server design, a possible allocation of bit rates for each layer of scalable video can be chosen as in table 1. In this paper, we focus on constant bit rate, variable quality MPEG2 video. Optimal allocation of bit rates for constant bit rate and variable bit rate scalable MPEG2 is being researched as part of the video server research, and will be covered in a separate paper.

Table 1: Bitrate allocation for scalable MPEG2 video (All frame rates are 24fps)

Layer	Avg. Bit Rate (Mbps)	Frame Size	Visual Quality	Avg. PSNR (dB)
Base	0.32	304 x 112	VHS	35
Spatial Enhancement	0.832	608 x 224	Super VHS	34
SNR Enhancement	1.856	608 x 224	Super VHS	37

The values in table 1 and figure 1 above are from a sequence of 3,000 frames from the movie Ben Hur. The burstiness in the trace data comes from the encoding structure of MPEG2 [7]. For simulation, trace data for MPEG2 scalable video was prepared for 15,000 frames, by using Columbia's full-profile, standard-conforming MPEG2 software encoder. The selection of the bit rates for each layer was based on the approximation that, for video of frame size 720x480 and 30 fps, a bit rate of 4.0 Mbps provides VHS quality video. For video frame sizes of m x n, at f fps, we approximate the required bit rate b Mbps for VHS quality in the following way: $\frac{m \times n}{720 \times 480} \times \frac{f}{30} \times 4.0 = b$. The bit rates in the table are incremental bit rates (not accumulated rates).

In advanced video server systems it will be necessary to support different video resolutions and video stream data rates to accommodate clients with different network bandwidth, display resolutions and processing power. Storing multiple independent video streams of the same video in order to support different video resolutions will be inefficient in terms of video storage requirements. Scalable video provides an effective way to store multiple resolutions of videos with the same storage requirements of single resolution videos.

3. Overview of System Operation

We consider a parallel array of independent disks each connected in parallel to a central memory. When a request for an I/O operation from a single disk is placed, two types of overhead are incurred: the time it takes for the head to move to the appropriate cylinder (referred to as the *seek time*), and the time it takes for the first sector to appear under it (referred to as the *rotation latency*). Following this overhead, the transfer of the video data begins. The transfer time is a function of the data requested. For every cycle of the video server, one 'retrieval block' of video data has to be retrieved for every video stream. For a larger retrieval block, a higher utilization efficiency can be achieved, but a larger buffer size will be required. Each video stream is serviced in a round robin fashion during each cycle. The retrieval block is a fixed number of frames that is referred to as a *group of frames* (gof).

4. Data Placement Strategy for Interactivity QoS

In this section, two extreme strategies for the placement of video data on a parallel array of disks is compared. These strategies are presented to provide a framework to evaluate various placement strategies. For each scheme, advantages and disadvantages are compared, and a new, flexible strategy for the placement of video data is presented. The new scheme is shown to support a range of interactivity QoS for a parallel array of disks.

In advanced digital video systems of the future, we reconsider the commonly accepted notions of interactivity. Our goal for interactivity in the video server is not to 'simulate' VCR functions exactly but to achieve effective search mechanisms while efficiently utilizing the limited resources of a video server. We propose that the critical functions of interactivity that are required for video servers are location of specific scenes and multiple rate 'scanning' of video segments (fast/slow forward/reverse). For our data placement strategy, the maximum interactivity delay values of all the interactivity functions (start, request gof, resume) (figure 2) are equal and will be shown to depend on a single parameter (segmentation level). The scan granularity (the minimum gof interval between consecutive gof retrieved during a scan) is directly dependent on this same parameter.

4.1 Balanced placement.

This scheme represents one end of the interactivity QoS. The interactivity delay (defined in figure 2) of this scheme is one cycle, but the utilization of each disk is low. For example, if a user pauses the playback of a video stream and after some time requests that the video stream be resumed, the video stream would be able to resume in the following round robin cycle. In this scheme, each group of frames (gof) of each resolution of video is divided into N_d equal segments and placed over all N_d disks. In

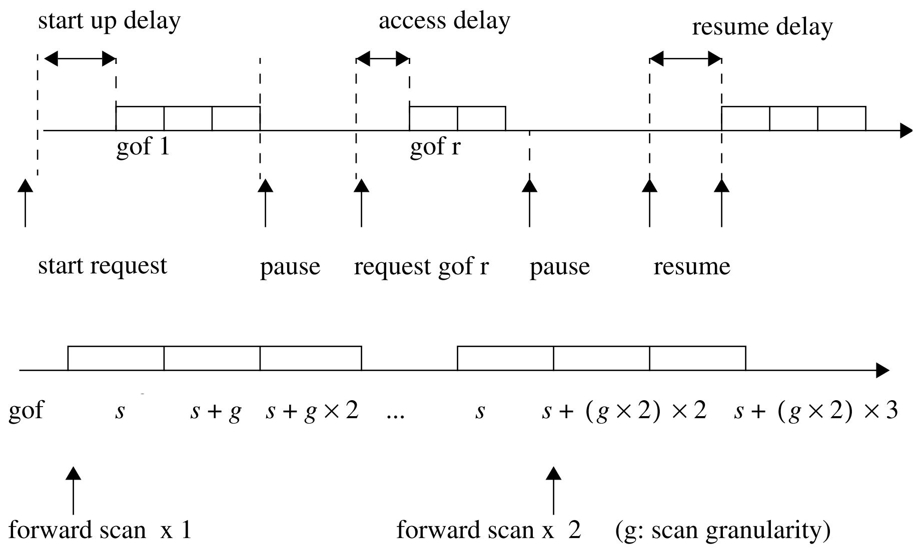

Figure 2: Interactivity functions of video server.

[3] a similar data placement strategy is presented, in which the full resolution gof is segmented to N_d segments. For the balanced placement strategy, the excessive number of disk seeks leads to under utilization of the disks. For an approximate analysis, we first consider all video streams in the scheduler round robin to be of one resolution. For our analysis, each disk is assumed to use the SCAN disk scheduling algorithm during each round robin cycle. It is shown that utilization $\rho = \frac{R_p \times S_{max}}{N_d \times R_d}$ (S_{max} is the maximum number of video streams of a given resolution that the parallel array of disks can support, R_p is the video playback rate and R_d is maximum disk transfer rate). As shown in our prior work [5], S_{max} can be derived from the following equation, where T_{cycle} 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.

$$T_{cycle} = S_{max} \times \left(\frac{T_{cycle} \times R_p}{N_d \times R_d} + T_{rx} + T_{sm} \right) + 2 \times T_{sx} .$$

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. Based on this analysis, we show the trade-off between utilization of the disk array and interactivity delay in figure 6. This will be explained further in section 4.3. The analysis shows that increasing the retrieval block size (gof) increases the utilization of the disks, but also results in an increase in the maximum interactivity delay. The analysis above is also used to evaluate the periodic placement strategy and the multiple segmentation strategy.

4.2 Periodic placement [4].

This scheme represents the opposite side of the interactivity QoS. This scheme maximizes disk utilization, and the maximum access delay will be shown to be N_d cycles. For each video, consecutive gof are placed on consecutive disks in a round robin fashion. For every cycle, one gof is retrieved for every video stream connected to the video server. Each gof is retrieved from a single disk (compared to multiple disks in the balanced placement scheme). If a single disk can support n gof retrievals in one cycle, then N_d disks support (N_d.n) video streams concurrently.

The observation is made that for a video stream starting retrieval of video data at cycle r, the video stream accesses a different single disk during each cycle. However, the video stream accesses the same single disk during each cycle as all video streams with start cycles in the following set: $\{r_i, i=1,2,...N_s \mid (r_i \textit{ mod } N_d = r \textit{ mod } N_d)\}$, where r_i denotes the start cycle of video stream i and we assume the first gof of all videos are stored on the same disk. This observation shows that for all video streams connected to the video server, we can group the video streams into N_d *video stream sets*.

All video streams in a video stream set retrieve data from the same disk during any given cycle (figure 3). Based on this, it can be shown that the worst-case interactivity delay for a video stream is N_d cycles for any of the equivalent interactivity functions. To prove this, we first note that each of the interactive functions are equivalent in that a *specific required gof* must be retrieved from the array of disks. The number of video streams being serviced on the disk that contains the required gof is the number of video streams in the video stream set that is accessing the disk during a particular cycle. The required gof cannot be accessed until a video stream set that can accommodate a new video stream is accessing the appropriate disk. Since the total number of sets are N_d, the maximum access delay before retrieval is N_d. We can also show that the scan granularity for this scheme is N_d. It has been shown that for regular playback, a video stream j accesses consecutive disks to retrieve consecutive gof. If video stream j requires a forward scan while only utilizing the resources reserved in its video stream set, we can show that the scan granularity is $g=N_d+1$ i.e. consecutive gof retrieved for a scan starting at gof s are: s, $(s + g \times 1)$, $(s + g \times 2)$,... In other words, the finest forward scan this scheme can support is (N_d+1).

4.3 Multiple Segmentation

In this section, we present a new flexible strategy for the placement of video data on a parallel array of disks. This scheme allows videos to take on a range of maximum interactive delay and scan granularity values. It is shown that decreasing interactivity delay can be achieved at the cost of decreasing utilization efficiency. Therefore, there is a design range for the placement of video data. In this section we also show how scalable video has advantages for video servers based on this placement strategy. We also develop an admission control framework based on this data placement strategy. The new scheme presented here uses different degrees of segmentation of gof blocks for the placement of gof blocks across a parallel array of disks.

Multiple segmentation (MS) scheme:

1. For a parallel array of N_d disks, we define ($\log_2 N_d$) + 1 segmentation levels: $S = \{S_i = 2^i, i=0,1,...\log_2 N_d\}$

2. For a given segmentation level S, divide each gof into S equal segments.

3. For a given segmentation level S, specify (N_d / S) sets of disks.

4. For each video sequence, the consecutive retrieval blocks (gof) which were each divided into S equal segments are stored on consecutive sets of disks as in table 2.

Table 2: Multiple Segmentation N_d=8 disks

Disk	1	2	3	4	5	6	7	8
S=1	gof 1	gof 2	gof 3	gof 4	gof 5	gof 6	gof 7	gof 8
S=2	gof 1	gof 2	gof 3	gof 4	gof 1	gof 2	gof 3	gof 4
S=4	gof 1	gof 2	gof 1	gof 2	gof 1	gof 2	gof 1	gof 2
S=8	gof 1	gof 1	gof 1	gof 1	gof 1	gof 1	gof 1	gof 1

Balanced placement is a special case of multiple segmentation with S =8, while peri-

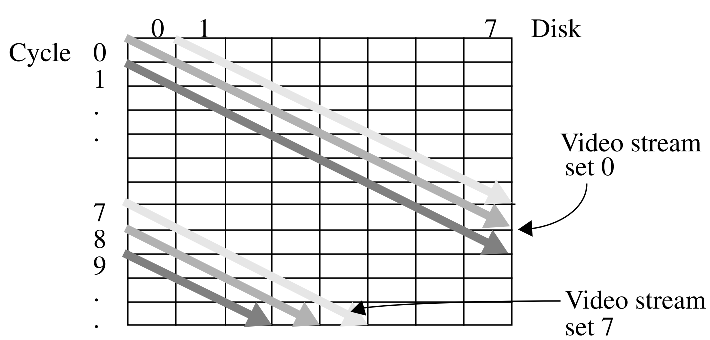

Figure 3: Video stream sets for periodical placement strategy.

odical placement is a special case with S=1. We can show that increasing the segmentation S reduces the maximum interactivity delay at the price of utilization efficiency.

For a segmentation level S, a video stream accesses S disks during each cycle. Extending the structure of video stream sets, we develop the structure of *component video stream sets*. For a parallel array of N_d disks, we define N_d component video stream sets. For a video j that is stored with segmentation level S, we say that S component video streams are required for a single video stream of video j. Therefore, resources are reserved on S component video stream sets for the retrieval of a single video stream for video j. Figure 4 shows which component video stream sets are used for the retrieval of a given video stream at cycle r stored with segmentation level S.

All component video streams in a video stream set retrieve data from the same disk during any given cycle. Based on this, it can easily be shown that the worst-case interactivity delay for a video stored with segmentation level S is N_d/S cycles for any of the equivalent interactivity functions. Suppose that a video j is stored with segmentation level S=2 on an N_d=8 disk array (figure 5). Consider a video stream that has reserved resources on component video stream sets (0, 4), with all other video stream sets exhausted by other video streams. Assume a request for a gof stored on disks (0,4) is

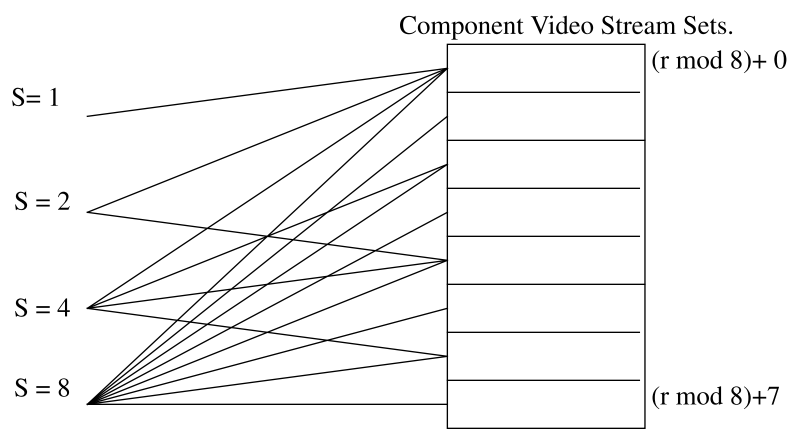

Figure 4:Component video streams sets for the retrieval of video stream at cycle r

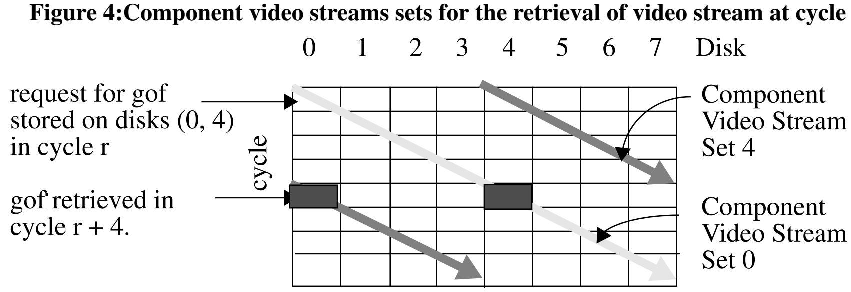

Figure 5: Example to demonstrate maximum interactivity delay.

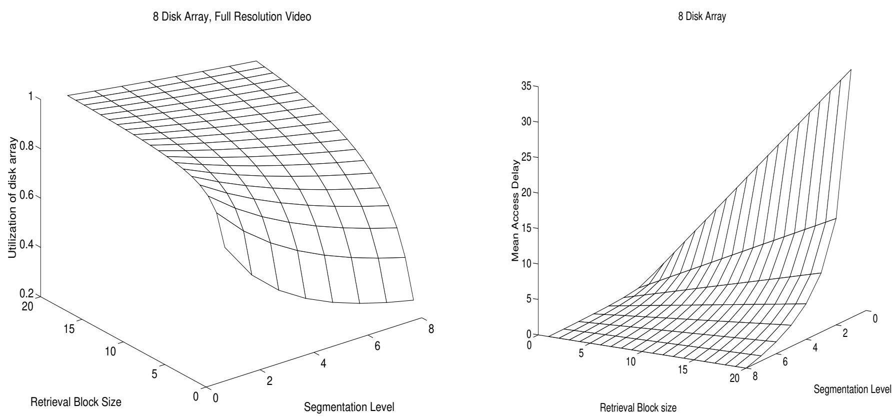

Figure 6: Performance of utilization and mean access delay vs retrieval block size and segmentation level

made during cycle r, and the desired start gof is stored on the disk(s) which have just been accessed during cycle r. The delay before the desired gof can be accessed from the appropriate disk(s) is $N_d/S=4$. In summary, a video stored with segmentation level S requires S component video streams. If resources are reserved on S component video stream sets, a maximum of N_d/S cycles are required before a given set of S component video stream sets has accessed all disks. Using a similar analysis, we can also show that the scan granularity for this scheme is N_d/S.

This scheme has advantages in flexibility in that videos with high interactivity delay tolerance can be stored with a smaller segmentation level, and videos requiring low interactivity delay are stored with higher segmentation levels. Multiple levels of segmentation can be supported on the same array of disks in a video server to provide a range of interactivity QoS.

4.4 Multiple Segmentation Based on scalable MPEG2 Video

We use the proposed multiple segmentation scheme to segment each gof of each layer of scalable video into S segments. The specific value of S to use is a design parameter that can be chosen by the system designer for each video in the video server, depending on its access requirements. We now consider how to divide each gof of each resolution (layer) of video into S segments. Each gof of each layer consists of a sequence of I, B, P frames of MPEG2 video. There are two basic ways to segment the gof of a given layer, as shown in figure 7. Using method 1, we see that if one segment is not retrieved, all frames of a gof will be affected. Method 2 is clearly a better option. Furthermore, based on method 2, we may group together frames of the same type (I, P, B) before segmentation. In this way, we assign the highest priority to segments containing I frames, intermediate priority to segments containing P frames, and the lowest priority to segments containing B frames. Segmentation does not occur exactly at frame boundaries. Each segment has an associated priority, and the priorities can be used in the video server scheduler to selectively drop segments to achieve *graceful degradation* in the case of congestion. In addition, further granularity in interactive scan functions can be easily achieved by skipping B and/or P frames. In many real time applications or near real time applications for which fast responses are critical, lower layers may be segmented with a higher level (method 3, figure 7) so that lower layers can be retrieved with shorter delays for a high degree of interactivity. This can be used for progressive retrieval in which lower layers are displayed before full resolution layers are fully retrieved (figure 8).

4.5 Admission Control Framework Based on Multiple Segmentation.

Given the multiple segmentation placement strategy presented above, we develop an admission control framework for the video server. It was shown that each incoming video stream can be decomposed into a number of *component video streams*. Higher segmentation levels require more component video streams for a single video stream. Admission control at the video server is an operation at the call establishment level for a video stream request at a video server. Given an incoming request with a specific QoS requirement, the admission control must decide to accept or reject the call. The policy has to determine if the request can be serviced by the video server while maintaining heterogeneous QoS requirements of all video streams already connected to the video server. The challenge is to maximize the utilization of the video server resources while ensuring heterogeneous QoS requirements of connected video streams. For a parallel array of N_d disks, we define N_d component video stream sets (CVSS). All component streams in a given CVSS retrieve video data from the same single disk during a given cycle. The component video streams in a CVSS are said to be connected to the same *logical disk*. The CVSS simplification provides a strategy for admission control in the video server. In the video server, we maintain a single CVSS admission control table. For each incoming video stream, we update the corresponding CVSS entries accordingly. Note that depending on the resolution of the video stream, we calculate whether the incoming video stream can be supported on the each logical disk associated with each CVSS. All logical disks are assumed to be identical with the same disk characteristics.

5. Conclusion and Future Work.

We have presented a new video data placement strategy for scalable video. In our testbed, the data placement strategy was applied specifically three layer, scalable MPEG2 video. The new strategy is a more flexible strategy than those proposed previously, and can accommodate a wide spectrum of video access characteristics in video servers. We have also showed the advantages of using scalable video based on the proposed data placement strategy. We have also presented a framework for admission control in a video server based on this new data placement strategy. The complexity of the admission control was shown to be reduced to that of a single disk system, even for parallel array of disks. Simulations are being to test the video storage unit design presented here using discrete event simulation [10]. In our simulation (figure 9), we model each disk in a disk array with specific disk characteristics to test our simplified approxima-

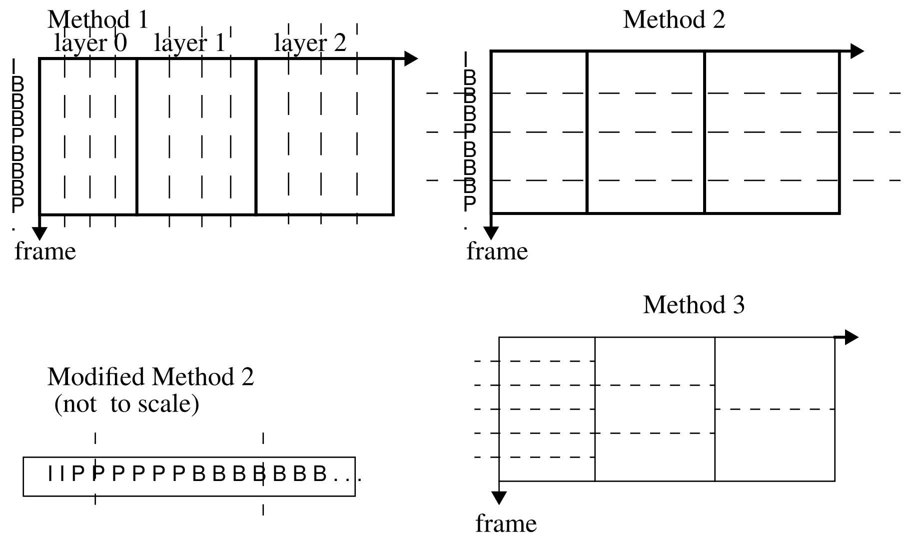

Figure 7: Multiple segmentation based on scalable MPEG2 video.

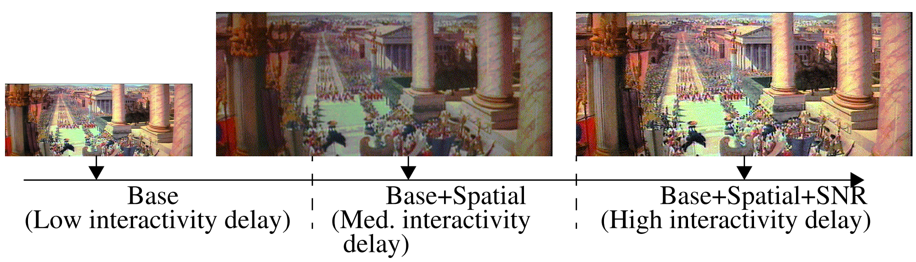

Figure 8: Improved utilization/interactivity performance using scalable MPEG2 video

tions. The simulations are based on actual trace data for three layer scalable MPEG2 video coding. Based on poisson arrival statistics of clients and simple interactivity request models, we are testing utilization and interactivity performance of our multiple segmentation data placement strategy for independent parallel disk arrays. Future plans include implementation of the video storage unit design into the video server for the Columbia campus-wide Video On Demand testbed.

For guaranteed delivery of video streams from the video server, we can deterministic or statistical service. In general, statistical service achieves higher utilization of the disk array. In our current research, we are focusing on a statistical and deterministic service for the retrieval of variable bit rate (VBR) video streams from the disk array. Providing guaranteed service requires resource reservation at the disk array. For constant bit rate video streams, we can achieve a high utilization of the disk array resources. However, reservation of resources for VBR video based on the peak rate leads to low utilization of resources. We are developing schemes to increase the utilization of disk resources for VBR video based on multiple buffer management to sup

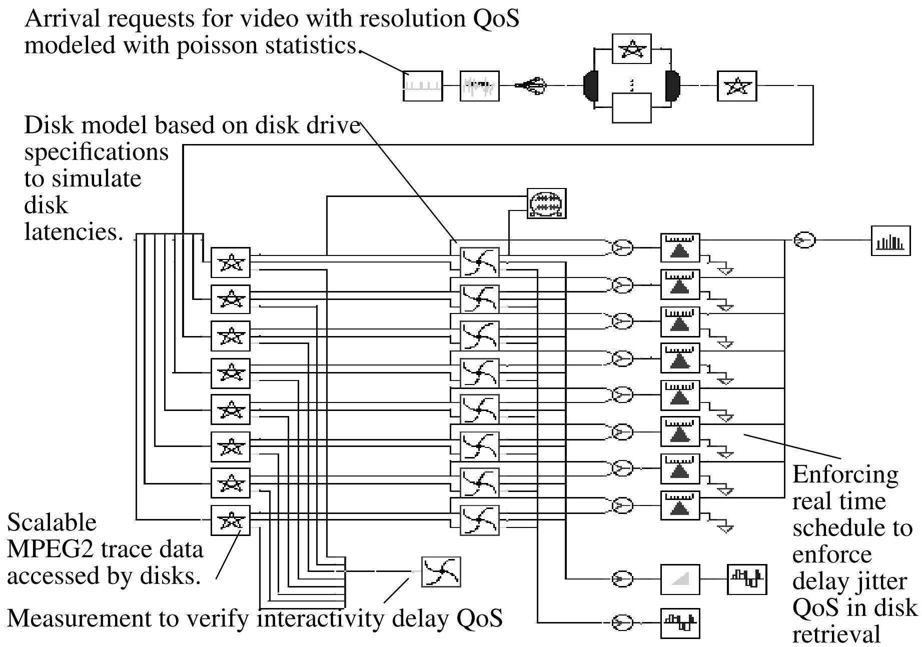

Figure 9: Parallel Disk Array Simulation

port delay jitter. This will introduce a *delay jitter QoS* dimension to the heterogeneous QoS suite of a video server.

References:

1. S.-F. Chang, D. Anastassiou, A. Eleftheriadis, J. Meng, S. Paek, S. Pejhan, J. Smith, “Development of Advanced Image/Video Servers in the Video On Demand Testbed,” IEEE Visual Signal Processing & Communications Workshop, Sept. 1994, New Brunswick, NJ.

2. S. -F. Chang, A. Eleftheriadis, and D. Anastassiou, “Some Interoperability Issues in Columbia’s Video on Demand Testbed”, Contribution to International Digital Audio/Visual Council, June 1994

3. K. Keeton and R. Katz, “The Evaluation of Video Layout Strategies on a High-Bandwidth File Server,” Intern. Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, England, UK, Nov. 1993.

4. E. Chang and A. Zakhor, “Scalable Video Data Placement on Parallel Disk Arrays,” SPIE Symposium on Imaging Technology, San Jose, 1994.

5. P. Bocheck, H. Meadows, and S.-F. Chang, “A Disk Partitioning Technique for Reducing Multimedia Access Delay,” Intern. Conference on Distributed Multimedia Systems and Applications, Honolulu, Aug., 1994.

6. Tihao Chiang and Dimitris Anastassiou, “Hierarchical Coding of Digital Television,” IEEE Communications Magazine, May 1994.

7. ISO/IEC 13818-2 Committee Draft (MPEG2)

8. “The Almagest,” Manual for Ptolemy, University of California at Berkeley, 1992.