行政院國家科學委員會專題研究計畫成果報告

視訊伺服器之設計與製作(IV)
Design and Implementation of a Video Server (IV)

計畫編號：NSC 88-2218-E-009-043
執行期限：87 年 8 月 1 日至 88 年 7 月 31 日
主持人：張瑞川　　國立交通大學資訊科學學系

1. Chinese Abstract

目前關於視訊伺服器的研究及商業化的視訊伺服器產品，大都假設其所儲存的視訊資料是以 MPEG-1/2 壓縮的視訊資料，然而新一代影音壓縮標準 MPEG-4 將音效、影像等以物件方式表示成影音物件 AVO(Audio/Visual Object)，這使得接收端對於視訊資料有更多的互動式功能。相對的，這也使得以儲存 MPEG-4 為主的視訊伺服器在設計及製作上比傳統以 MPEG-1/2 為主的視訊伺服器更為複雜。

為了使視迅伺服器能更有效的支援 MPEG-4，我們改進了硬碟上的即時磁碟讀寫頭排程以克服儲存系統的瓶頸。同時，配合 MPEG-4 更強大的互動式功能，我們也設計並製作一個具有互動式功能的主從式架構播放平台。在本報告中，我們詳述了此一即時磁碟讀寫頭排程及互動式功能的主從式架構播放平台的設計與製作。

關鍵詞：視訊伺服器、MPEG-4、隨選視訊、多媒體

Abstract

Currently, most researches and commercial products on video servers all assume hat the media files are encoded in the MPEG-1/2 format. However, with the introduction of the new generation of audio/video compression standard, MPEG-4, that represents audio and video as Audio/Video Objects (AVO), will support more interactive functionality during playback in clients. As a result, the design and implementation of an MPEG-4 VOD systems are more complicated than traditional MPEG-1/2 ones.

In order to make video servers more efficient for MPEG-4 streams, we propose a new real-time disk scheduling to overcome the bottleneck of storage subsystem. At the same time, we design and implement a interactive playback platform based on the client-server architecture to accommodate with the more powerful interactive functionality of MPEG-4. In this report, we describe the new real-time disk scheduling algorithm and the design and implementation of our interactive client-server playback system.

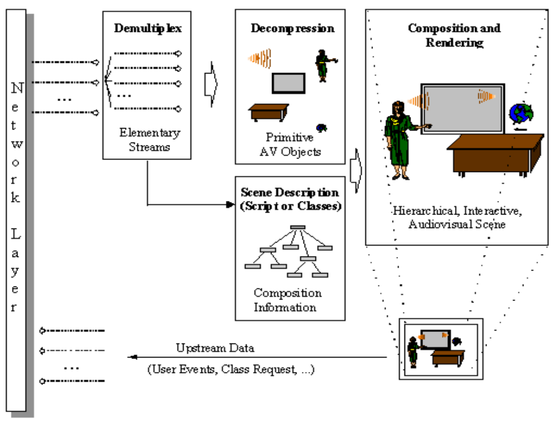
Keywords: Video server, MPEG-4, VOD, Multimedia

2. Motivation and Goal

In past years, we have designed and implemented a VOD server [1-2] based on MPEG-1/2 compression standard and the playback just consists of continuous display of frames. However, the new generation compression standard MPEG-4 introduces the concept of objects to support for more interactive functionality.

As showed in Figure 1, after receiving media data from network, the MPEG-4 stream is demultiplexed into a number of elementary streams. These elementary streams may be scene descriptions or compressed AVOs(Audio/Visual Objects) [3-5]. The AVO can be decompressed into the original primitive AV Objects. And the scene description is a hierarchy data structure, which describes the relationship of AVOs. Finally, by means of the composition, the scene description and its associated AVOs are reconstructed into the hierarchical, interactive, and audiovisual scene.

Figure 1. The MPEG-4 client playback architecture



As above illustrated, the design and implementation of MPEG-4 video severs are more complicated than traditional MPEG-1/2 ones. Therefore, our goal is to investigate the influence of MPEG-4 compression characteristics for video servers and improve the performance. First, we introduce a new real-time disk scheduling algorithm to obtain better disk throughput, and consequently, the number of supported disk requests. Also, we design and implement an interactive client-server architecture playback platform to accommodate with the more powerful interactive functionality of MPEG-4.

3. Results and Discussions

3.1 Real-Time Disk Scheduling

A VOD system must not only support lots of users, but also guarantee real-time requirement [6] for continuous playback. The EDF (Earlier Deadline First) is the disk scheduling algorithm for satisfying the real-time constraints of multimedia system. However, the employment of EDF results in poor I/O throughput. SCAN, on the other hand, tries to optimize the disk throughput but being short of the real-time consideration. Thus, SCAN is not suitable for real-time applications. SCAN-EDF [7], the best-known real-time disk scheduling algorithm, reschedule the requests with the same deadlines by the seek-optimization SCAN scheme to reduce disk service time. But the seek-optimizing scheme is applied only for requests with the same deadline, the obtained improvements are limited.

Therefore, we propose a new real-time disk-scheduling algorithm, called DM-(Deadline-Modification-SCAN) algorithm with an idea of MSG (Maximal-Scannable-Group) for rescheduling [8]. We define MSG as a set of consequent requests that can be successfully rescheduled by SCAN under guaranteed real-time requirements. Be means of MSG, we can increase the number of requests that can be rescheduled by SCAN, no longer limited by requests with the same deadline. Notations used in this report are listed in Table 1.

Table 1. Notations used in this paper.

n	number of the input requests
r_i	release time of T_i
d_i	deadline of T_i
a_i	the track location of T_i

$c_{j,i}$	service time of T_i when served after T_j
e_i	the start-time of T_i
f_i	the fulfill-time of T_i
T_Z	the schedule result of T by the algorithm Z, $T_Z = T_{Z(0)}T_{Z(1)}\ldots T_{Z(n)}$
G_i	maximum-scannable-group of T_i

Definition: MSG

Given an EDF schedule $T_E = T_{E(0)}T_{E(1)}\ldots T_{E(n)}$, the maximum-scannable-group G_i started from request T_i is defined as the maximum consequent request group $G_i = T_{E(i)}T_{E(i+1)}\ldots T_{E(i+m)}$ which satisfies: $f_{E(k)} \leq d_{E(i)}$ and $r_{E(k)} \leq e_{E(i)}$ for $k = i$ to $i+m$. ❑

Given a feasible EDF schedule, we can prove that the seek-optimized reschedule result of MSG could successfully improve the I/O throughput under guaranteed real-time requirement. Note that our algorithm is based on the assumptions that the input schedule is an EDF schedule. To iteratively improve the scheduling result by the same algorithm, the input schedule should be kept as an EDF schedule. To satisfy this requirement, we develop a deadline-modification-SCAN (DM-SCAN) algorithm to modify the deadlines of requests. After the deadlines are modified, we have a new EDF schedule. Therefore, we can iteratively reschedule the new MSG by the same approach to improve the I/O throughput. The original deadlines are restored after DM-SCAN. This entire algorithm can be described as follows.

Algorithm: *DM-SCAN*

/* INPUT: the scheduling result of EDF.
OUTPUT: a real-time disk schedule. */
Store $d'_k := d_k$ for all requests T_k
/* d'_k backup real deadline d_k of T_k */
repeat
  Identify all MSG groups from input
  Identify all MU_MSG groups from MSG
  /* local group seek-optimization */
  for $i := 1$ **to** n **do**
    if (G_i is MU_MSG) **then**
      reschedule G_i by SCAN;
  /* deadline modification */
  for $k := n-1$ **down to** 1 **do**
    if ($d_k > d_{k+1}$) **then** $d_k := d_{k+1}$;
until (no deadline is modified);
Recover $d_k := d'_k$ for all requests T_k.
/* restore real deadlines */

3.2 Interactive Functionality Support

As stated in section 1, the scene description information is carried within potentially hundreds of data channels. This calls for a high rate of establishment and release of numerous short-term transport channels with appropriate quality of service (QoS) [9].

Thus, the emergence of MPEG-4 requires the need for transport protocols with different characteristics to coexist and be integrated with single applications. By means of these different characteristic transport protocols and hence scene description, users can be supplied with more interactive support than MPEG-1/2 by MPEG-4's object description techniques.

Therefore, we implement a client-server architecture for interactive playback with MPEG-4 streams on Windows 98 and Windows NT. We implement the client and server based on WinSock Application Interface [10-11]. Below the WinSock is the appropriate transport protocol among which we choose from based on the requirement of scene description.

4. Evaluations

The experimental results for DM-SCAN are presented to compare conventional approaches. We use the HP 97560 hard disk for our performance evaluations. HP 97560 contains 1962 cylinders. It has 19 tracks-per-cylinder and 72 sectors-per-track with 512 bytes sector size. Given the seek distance $d_{i,j}$ as $|a_j - a_i|$. We have service time $c_{i,j} = 8 + 0.008 d_{i,j}$ if $d_{i,j} > 383$. Otherwise, $c_{i,j} = 3.24 + 0.4(d_{i,j})^{1/2}$.

In our experiments, the data access for each input request is assumed to be on one disk track (36 KB in HP 97560) without splitting across different cylinders. We assume that the workloads of input requests are uniformly distributed on the disk surface. Requests' release times are randomly generated with proper deadlines assigned.

For each approach, the same 100 test examples are presented. Each test example contains 10 randomly generated requests. The mean values for the obtained schedule fulfill-times are evaluated.

Table 2 shows the minimum, the maximum, and the average schedule fulfill-time obtained for different real-time disk scheduling approaches. In this table, we apply the same 100 test examples (each test example contains 10 randomly generated requests) to different solution approaches to keep the comparison as fair as possible. For each solution approach, the related improvements are also presented. In this paper, the measurement of I/O throughput improvement is compared to the obtained results of SCAN-EDF. The proposed DM-SCAN approach can achieve 14% improvements. Besides, 9% improvements are obtained at the first iteration (DM-SCAN-1).

Table 2. The minimum (min), maximum (max), and average (avg) schedule fulfill-time, and the improvement (imp) of the I/O throughput.

Algorithm	Schedule Fulfill-Time (msec)			
	min	max	avg	imp
DM-SCAN	289.27	402.36	356.74	14%
DM-SCAN-1	317.24	460.18	375.12	9%
SGR	291.37	462.70	385.63	7%
SCAN-EDF	377.06	462.60	414.84	=0%

** The applied group size of SGR is 5.*

** The improvement is related to EDF.*

Table 3. The number of request supported for different scheduling policies

Algorithms	Number of Supported Requests		
	min	max	avg
DM-SCAN	7	25	13.75
SCAN-EDF	4	24	10.21
EDF	3	24	9.87
SFTF	1	10	4.38
FCFS	1	10	4.11

** SFTF = shortest fulfill-time first.*

** FCFS = first come first serve.*

To further study the number of requests supported, Table 3 shows the minimum, the maximum and the average number of real-time requests that can be successfully scheduled by different approaches. To each test approach, the same 100 examples with a sequence of randomly generated requests are applied. In each test example, we try to increase the number of input requests to the test procedures until the related schedule result is not valid for real-time requirements. According to our experiment, the proposed DM-SCAN approach can support more requests than the conventional approaches.

For the interactive functionality support, we have already implemented the client-server architecture platform and can supply with good interactive playback. In addition, it will be used as our research platform for furthermore investigation of MPEG-4 streams interactive characteristics.

5. References

[1] W. Y. Chung and R. C. Chang, "An Implementation of the CMSS VOD Server," In 1997 Workshop on Consumer Electronics: Digital Video and Multimedia and NSC Annual Seminar on HDTV Research and Development, Oct. 1997.

[2] W. Y. Chung, M. L. Chiang, and R. C. Chang, "Reducing the Redundant Memory Copying in VOD Server," In 1998 IEEE International Symposium on Consumer Electronics (ISCE'98), Oct. 1998.

[3] ISO/IEC 14496-1 CD, "MPEG-4 Systems," Oct. 1997.

[4] ISO/IEC 14496-2 CD, "MPEG-4 Visual," Oct. 1997.

[5] ISO/IEC 14496-3 CD, "MPEG-4 Audio," Oct. 1997.

[6] A. Dan, D. Sitaram, and P. Shahabuddin, "Scheduling Policies for an on-demand video server with batching," Proc. ACM Multimedia Conf., pp.15-22 1994.

[7] A. L. N. Reddy and J. Wyllie, "Disk scheduling in a multimedia I/O system," *Proc. ACM Multimedia Conf.*, pp. 225-233, 1993.

[8] R. I. Chang, W. K. Shih, and R. C. Chang, "Deadline-Modification-SCAN with Maximum-Scannable-Groups for Multimedia Real-Time Disk Scheduling," Proc. Real-Time Systems Symposium (RTSS'98), 1998, pp. 40-49.

[9] J. F. Huard, A. A. Lazar, K.S.Lim, and G.S.Tselikis, "Realizing the MPEG-4 Multimedia Delivery Framework," IEEE Network, Nov. 1998, pp. 35-45.

[10] M. Hall, M. Towfiq, G. Arnold, D. Treadwell, and H. Sanders, "Windows Sockets Specification V1.1," 1993.

[11] The WinSock Group, "Windows Sockets 2 Application Programming Interface," 1996.