論文
5-2-13

Advanced Fast-Forward/Backward Algorithm for VOD Server
(VOD서버를위한개선된Fast-Forward/Fast-Backward 알고리즘)

김진욱(Jin-Uok Kim)[1] 정수목(Soo-Mok Jung)[2]

Abstract

In this paper, we devise both methods supporting interactive VCR-like functions for MPEG stream and minimizing the storage space. Specifically, we proposed both methods to efficiently stripe MPEG stream into multiple disks, and an access strategy to retrieve the distributed streams over the disks. To accurately locate the streams to be referred to, we introduce a table indexing the location. When client request Fast Forward(FF)/Fast Backward(FB), VOD(Video on Demand) server refer to this index table supporting FF/FB functions. Experimental result shows that the proposed method can reduce the additional storage space needed when VOD server services FF/FB functions.

요약

본 논문에서는 MPEG 스트림에 대하여 VCR과 같은 기능을 제공하고 저장 공간을 최소화하는 알고리즘을 제안하였다. 특히 MPEG 스트림을 여러 개의 디스크에 효과적으로 스트라이핑하는 방법과 전체 디스크에 걸쳐 분산되어 있는 스트림을 검색하기위한 접근 방안을 제안하였다. 제안된 알고리즘에 대한 실험을 통하여 VOD 서버가 FF/FB기능을 제공하는 경우에 필요한 추가적인 저장 공간을 감소시킬 수 있음을 확인하였다.

논문접수 : 2004. 2. 10
심사완료 : 2004. 2. 17.

1) 정회원 : 한국증권전산
2) 정회원 : 삼육대학교 컴퓨터과학과 부교수

1. Introduction

VOD service is a technology that sends MPEG(Moving Pictures Export Group) or other video format data to many users in real time.

One of the key component in the above application is the VOD server which is responsible for the storage and transmission of videos. Depending on the application, a VOD server may be required to store hundreds of videos, and may be required to concurrently transmit to clients, data for a few hundred videos. MPEG is a widely used multimedia stream in VOD server, because Inter-frame compression techniques of MPEG provide significant advantages in both storage and transmission. Consequently, it is welcomely accepted for VOD service[1][2].

In a VOD server, in addition to providing the basic "start/stop" functions, it is highly desirable to provide the user with VCR-like functions such as fast forward(FF) and fast backward(FB). There are several possible approaches to implementing these functions. However, each of these approaches implies additional resource requirements on the system.

Therefore, we address the problem of supporting VCR-like functions for inter-frame compressed video streams such as MPEG and minimize the additional resource requirement.

In this paper, to meet such purpose, we propose GOP block placement method and I-frame access method to remove the additional storage requirement. Furthermore, it is desirable to support interactive access to MPEG stream and support VCR-like functionality.

2. Previous work

To implement FF/FB without considering about MPEG stream reference in previous, the storage is placed MPEG stream both original MPEG stream for normal playout mode and additional MPEG stream that is organized with I frames for using FF/FB. For example we can use MPEG-2 stream and store movie of 100 minutes, and the size of original MPEG stream is 3GB, encoded with 4Mbits per second, then additional storage space is occupied with about between 15 percent and 30 percent of original MPEG stream. Therefore that is not suitable to implement to support FF/FB because this system has additional storage space.

In general, schemes that support fast forward and fast backward operations display frames at a rate higher than normal playout[3]. Or skip frames. In the former scheme, fast-forward at n-times the normal playout rate required n-times as many frames to be retrieved, yielding n-fold increase requirements, then the client request must be delay until the necessary resource become available. In the latter scheme, on the other hand, fast-forward at n-times the normal playout rate is achieved by displaying every n-th frames at the normal playout rate. Although conceptually elegant, such frame skipping schemes may not be directly applicable for MPEG stream that are encoded using compression techniques that exploit temporal redundancy between successive frames. This is because, such compression techniques create inter-frame dependencies, which may prevent every n-th frame to be independently decoded.

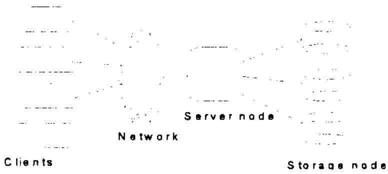
Sakamoto *et al.*[4] proposed "Decoding only I frame"on servicing user request FF/FB. This method has another storage space per one movie. One is used to store a MPEG stream for play mode, and the other is only I frame for FF/FB mode. In this method we can consider two problems. First, VOD server sends client only I frame of MPEG stream when client station requests FF/FB. In the client station, the client decodes only I frame in real time, and the volume of data (I frames) sent to client is larger than MPEG stream of normal play since the size of I frames is larger than P and B frames. As a result, more network bandwidth is required during FF/FB period. Second, the VOD server has to store additionally another MPEG stream consisted with I frames only. The amount of additional space reaches about 15 percent to 30 percent of original MPEG stream.

The chain of problem and as its solutions has unfolded these evolutionary efforts. Then, widely used approach is Sakamoto's method due to its inherent simplicity. Excessive network bandwidth required to support Sakamoto's method is attacked by Park[5] with the cooperation of client's node. The client decode an I frame repeatedly in several times. Accordingly, the I frame as much as the number of repetition is not needed. Meanwhile, the second penalty, additional storage space has been pending problem.

In this paper, to reduce the required additional storage space of Sakamoto's scheme in FF mode, we propose two schemes. One is a method to store MPEG stream, that is called GOP blockplacement method, for accessing Blocks in normal playout mode and efficiently accessing I-frame in FF mode, the other is to access the blocks, called I-frame access method for efficient mechanism to extract I frame out of block. Through those proposed methods, we can reduce additional storage space, and access requirements are evenly distributed all disks stored MPEG stream when VOD server service FF/FB functions.

3. Proposed Fast-Forward/Backward Algorithm

There are two major types of parallel VOD servers: shared multiprocessors and distributes memory cluster architectures. The MPEG stream is sent to the memory buffer through a high-speed network or bus, and then to clients. A mass storage system has presented the capacity of supporting hundreds of requests. However, it is not yet clear that a multiprocessor VOD server can be scalable. A clustered architecture is easy to scale to thousands of server nodes. In such a system, between a set of storage nodes and a set of server nodes are connected by a high-speed network such as Myrinet. Data is retrieved form the storage nodes and sent to the server nodes that send the data to clients.



[Fig. 1] System model

In this paper, we assume a clustered architecture as fig. 1. The storage nodes, each with a local disk array, are responsible for storing MPEG stream in some storage medium, such as disks. The server nodes are responsible for client's requests. On receiving a request from a client, a server node will schedule it to a time interval and deliver appropriate data blocks within some time deadline to the client during the playout of a video.

To comply with the reference among various types of frames as mentioned before, the multimedia stream is divided into blocks. It is called "GOP-block". Each block begins with an I frame and consecutive B or P frames. The block is the elementary unit of storage and retrieval when Play mode, and I frame is elementary unit of retrieval when FF/FB mode.

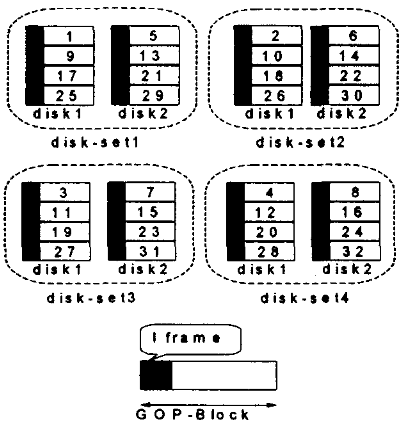
We consider a VOD server that contains *n* disk-set with blocks distributed in a round-robin manner across the disks. Formally, we can represent it as follows:

$F(b,n) = (b \bmod n)$, where b is a block and n is the number of disk-set.

An example of the block placement for $n = 4$ is given in Fig. 2.

Each storage node has a disk-set. In normal playout mode, client requests a GOP-block for playout that data. After server node receive client's request, then server node request the requested GOP-block to target storage node. The storage node well knows the next request GOP-block of client because the MPEG stream is continuous and client requests GOP-block one by one. Therefore the storage node can pre-fetch next GOP-block in memory and, at the same time, send requested GOP-block from storage mode's memory to server node through high speed network.

In FF mode, the retrieval process is same as normal playout mode. But, in this mode, requested blocks are only I frames. The advantage of this pre-fetching scheme in storage node is to overlap the time between sending time of one block to server node and retrieval time of block from disk.

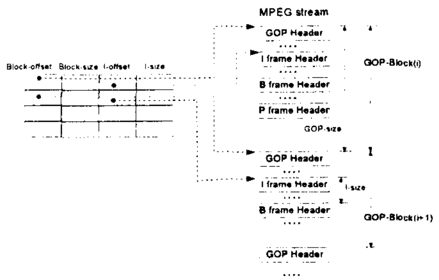


[Fig. 2] GOP block placement layout

In the proposed algorithm, the VOD server can service FF/FB request using the index table in I-frame access method. When a client requests a FF/FB operation, a VOD server sends I-frame of each GOP block in Fig 1. To extract the location of each I-frame, we use an index table, we can know the location of each I-frames in disk for the rapidity of service to client requests of FF/FB functions. This information is called "GOP block index table". This index table supports both the interactive serviceability of FF/FB functions, and the prompt switching between FF/FB and normal playout.

The index table has two parts. One is a structure for playout mode, and the other a structure for FF/FB mode. The information of play mode is consists of two fields, Block-offset and Block-size. Block-offset is the offset of selected block in disks, and the block-size is the size of that block. The information of FF mode also consists of two fields both I-offset and I-size. I-offset is the offset of I-frame in selected block from disks, and I-size is the size of that I-frame. In the index table, the value of Block-offset and I-offset is not equal. The reason is as follows. One GOP block has the headers of a frame or a GOP(Group of Pictures), and one block has headers in this order, GOP header , I-frame header, and other frame headers such as P frame headers or B frame headers. Strictly speaking, the Block-offset is pointed to GOP header and the I-offset is pointed to I-frame header.

Fig 3 illustrates the mapping relation between index table and MPEG stream in disks. We assume that the MPEG stream consists of consecutive GOP blocks and each GOP has one I frame and other frames such as P frames and B frames. Block offset indicates the GOP header of the block. Meanwhile, I-offset indicates the header of I frame of the block. Using the index table in I-frame access method, the VOD server can service FF/FB request.



[Fig. 3] Mapping relation between index table and MPEG stream

4. Experimental results

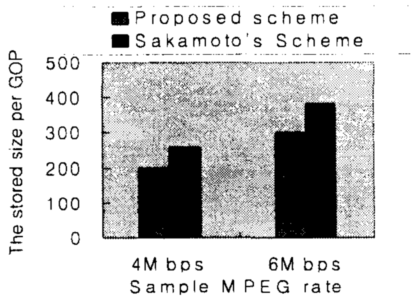
In this section, we compare and evaluate the performance of the proposed method and Sakamoto's method. For examining the practicality of our methodology, we evaluated the jitter delay of the two methods in the case of bursty disk status based on client-server model.

Using the proposed method, we simulated the arrival of video request by a uniform process. Assume that there are M different videos available in the VOD server. Upon arrival in the VOD server, a client selects in order to watch the *i*th video with probability $P_i = 1/M$, $1 \le i \le M$, we also assume that the request probabilities are steady through the simulation.

We assume that VOD server has 6 disks and disk parameters used in the simulation is Atlas XP34300[6]. Atlas is fabricated by Quantum, 5Gbyte SCSI hard disk, widely used for multimedia applications. We assumed the disk parameters such as average random seek(read) is 8ms, maximum seek time is 19ms, rotational delay time is 8.3ms, heads/drive are 20, tracks/surface are 3820, sectors/track are 137, sector size is 512B.. We assume that all disks'capability is same and disk queue works in FIFO(first In First Out) manner, with FCFS(First Come First Serve) disk scheduling.

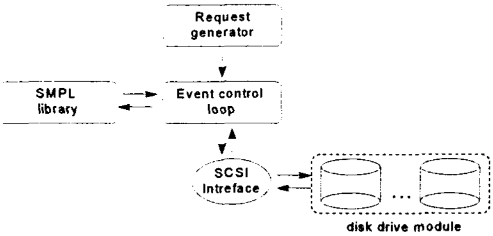
Fig. 4 shows the space of storage resource needed to store one GOP. The proposed scheme store only one GOP, but Sakamoto's scheme store both one GOP to support normal playout and one I frame to support FF. In 4Mbps encoding rate with MPEG-2, proposed scheme need 199.8 Kbytes, but Sakamoto's scheme need 258.1 Kbytes of storage space per GOP. Therefore, our strategy is highly superior to the Sakamoto's scheme since it does not need extra storage space to separately store I frames.

To compare proposed method with Sakamoto's method in same environment, we implemented simulator. The simulator is written in SMPL[7], and consists of request generator, event control loop, SCSI interface, and disk drive module. The composition of simulator is shown in Fig 5. The simulator periodically generates request in request generator, and the request enters event control loop. All event generated by simulator have transition to next event and is controlled by event control loop.



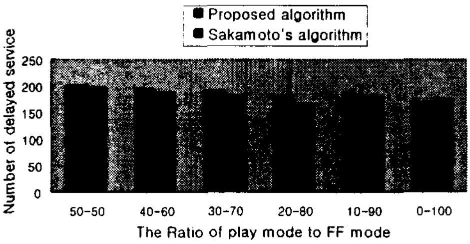
[Fig. 4] The stored size per GOB with sample MPEG rate

The request generated by request generator is transferred to disk drive module. Through this mechanism, request response time is evaluated. We assume that the number of total requests is 10000 in this simulator.



[Fig. 5] The configuration of simulator

In Fig 6, the pair of numbers beneath each bar means the normalized number of clients both in normal playout mode and FF/FB mode to 100. The first pair 50-50 implies that 10 clients are in normal playout mode and the rests in FF/FB mode within total 20 clients. The Y-axis gauges up the number of packets responded beyond the required time deadline.



[Fig. 6] Delayed service vs ratio of play mode to FF mode

The general trend of client preferences are skewed to normal playout rather than FF/FB. Therefore, the ratio of normal playout mode to FF mode from 50-50 to 0-100 is usual cases in commercial VOD server. In this range, proposed method shows little difference from Sakamoto's method.

5. Conclusions

In this paper, we have explored the GOP block placement method and I-frame access method to provide interactive VCR-like functions for MPEG stream.

Our GOP block placement method for all disks stored MPEG stream to service FF/FB divide original MPEG stream into each blocks, and I-frame access method service client's request interactively. The proposed method has index table to support rapidly accessing the location of I-frames when clients request FF/FB functions. Through this mechanism, the proposed method takes a great advantage in respect of storage space rather than Sakamoto's method. Together with it, The access time of our mechanism to support FF/FB has been also evaluated and compared with Sakamoto's way. Moderate performance was shown in the case of mixed both normal playout requests and FF/FB requests.

Reference

[1] L.A. Rowe, D.A. Berger, and J.E. Baldeschwieler, "The Berkeley Distributed Video-on-Demand System," Multimedia Computing Proc. of the Sixth NEC Research Symposium, Ed. T. Ishiguro, SIAM, 1996.

[2] H. Tezuka, T. Nakajima, "Simple Comtinuous Media Storage Server on Real-Time Mach," USENIX Technical Conf., San Diego, CA, 1996.

[3] J.K. Dey-Sircar, J.D.Salehi,J.F.Kurose, and D. Towsley,"Providing VBR Capabilities in Large-scale Video Servers," In Proc. of the Second ACM Int'l Conf. on Multimedia, Pages 25-32, Oct. 1994.

[4] H. Sakamoto, A. Uemori, H. Sugiyama, and K. Nishimura "Video Server Architecture Supporting Real-Time Input and Immediate Playback," In the Proc. of the Multimedia Japan 96, pp.224-231, March 1996.

[5] H. K. Park, and H.J. Cha,"An Efficient Implementation of Fast Forward/Reverse Functions for VOD Service,"In the Proc. Of the Transactions of the Korea Information Processing Society,pp.280-284, 1996.

[6] Atlas XP 34300 SCSI Hard Disk Drive

Product Brief, Oct., 1994.

[7] M.H. MaxDougall, "Simulating Computer Systems Techniques and Tools," The MIT Press, Cambrige, Messachusetts, 1987.

김 진 욱 (Jin-Uok Kim)

1997.2. 고려대학교

응용통계학과졸업(이학사)
1999.2. 고려대학교 대학원
컴퓨터학과 졸업(이학석사)
현재 한국증권전산
관심분야: 멀티미디어, VOD

정 수 목 (Soo-Mok Jung)

1984. 2. 경북대학교
전자공학과졸업(공학사)
1986. 2. 경북대학교 대학원
전자공학과 졸업(공학석사)
1986. 1. ~ 1991. 2.
LG정보통신 연구소 연구원
2002. 2. 고려대학교 대학원
컴퓨터 학과 졸업(공학박사)
현재 삼육대학교 컴퓨터 과학과 부교수
관심분야 : 멀티미디어, 영상처리,
컴퓨터아키텍처