

# 계층적 VOD 서버에서의 효율적인 비디오 데이터 저장 방법

전 성 배<sup>†</sup> · 이 원 석<sup>††</sup>

## 요 약

주문형 비디오(VOD) 서버는 대용량의 비디오 데이터를 저장하고 사용자에게 이를 실시간으로 전송하여야 하는 제약 조건을 만족시켜야 한다. 기존의 VOD 서버 시스템은 실시간 반응 시간을 보장하기 위해 대부분 비디오 데이터의 영구 저장 장치로 하드 디스크를 사용하지만 대용량의 비디오 데이터를 저장하는 데에는 경제성과 신뢰성의 문제점이 대두된다. 본 논문은 이러한 문제점을 보완하기 위해 대용량 저장 장치인 팟자기 디스크 주크박스를 사용하는 복합 다단계 VOD 서버에서 다양한 비디오 데이터 저장 방법들을 제안하고 각 방법의 저장 비용 및 성능을 비교분석한다. 또한, 복합 다단계 시스템과 하드 디스크를 기반으로 하는 서버의 특성을 고려하여 비디오 데이터를 저장함으로써 비용 효과적인 VOD 서버 구성 방법을 제안한다.

## Efficient Video Data Allocation Methods in a Hierarchical VOD Server

Sung Bae Jun<sup>†</sup> · Won Suk Lee<sup>††</sup>

### ABSTRACT

A Video-On-Demand(VOD) system has a large amount of video data in its server storage and sends the data to users via communication networks interactively. In this paper, we discuss the problems of a server which uses disks as its permanent storage for a video database. In order to avoid these problems, this paper proposes several video data allocation methods for a VOD server that utilizes the jukeboxes of optical disks, and compares their performance and implementation costs. Jukeboxes of optical disks can be more suitable for the storage of a large video database since it is reliable and inexpensive. In addition, this paper discusses the cost-effective integration of a VOD server that combines the advantages of the hierarchical and disk-based servers.

### 1. 서 론

최근 컴퓨터 및 통신 기술의 발전으로 고속망을 통해 사용자에게 비디오 데이터를 제공하는 주문형 비디오(Video-On-Demand) 서비스가 가능하게 되었다. 주문형 비디오 서비스를 위한 시스템은 크게 사용자의 요구에 따라 저장된 비디오 데이터를 검색 및 송출하는 비디오 서버(video server), 비디오 및 제어 테

이터를 전송하는 네트워크 부분과 전송된 비디오를 재생하는 가입자 장치로 구성된다. 또한 부가적 구성 요소로서 사용자 등록·요금 청구 등의 사용자 데이터 관리와 비디오 프로그램 관리 및 사용자 서비스 허가 제어와 같은 전체 VOD 시스템을 관리하는 운영 센터가 비디오 서버의 일부분 혹은 독립적인 요소로서 구성될 수 있다[1].

비디오 서버에 저장되는 비디오 데이터는 텍스트나 수치적 타입의 기존 데이터와는 다른 특성을 갖는다. 비디오 데이터는 압축된 경우라도 그 크기가 매우 크다. 압축률이 1:25 이상이고 1.5Mbps의 데이터 전송

<sup>†</sup> 준 회 원:연세대학교 공과대학 컴퓨터학과  
<sup>††</sup> 정 회 원:연세대학교 공과대학 컴퓨터학과  
논문접수:1997년 7월 1일, 심사완료:1998년 1월 14일

를 지원하는 MPEG-1을 기준으로 할 때 2시간 길이의 영화 한편에 대한 데이터를 저장하기 위하여 약 1.3GBytes의 용량이 필요하며, 서비스시 네트워크를 거치는 비디오 데이터는 끊어지지 않고 이어지는 스트림(stream) 형태로 발생해야 하고 미디어 형태에 따라 요구되는 대역폭에 맞게 실시간(real-time)으로 검색 및 재생이 이루어져야 한다. 또한 요구 즉시형(on demand)이므로 동시에 여러 사용자들을 지원할 때 새로운 스트림에 대하여 어떤 특정한 한계 내의 시작 지연(start-up latency) 시간이 보장되어야 한다. 사용자측에서 비디오 데이터의 재생은 갱신이 필요 없으므로 읽기 전용(read-only)의 성격을 지니며 빨리 감기, 되감기, 멈춤 등과 같은 기능의 제공이 요구된다.

이러한 비디오 데이터의 특성으로 인해 비디오 서버는 방대한 크기의 비디오 데이터베이스를 저장하기 위한 대용량 저장 장치와 많은 수의 사용자들 동시에 지원하기 위해 요구되는 입출력 처리량 및 CPU 처리 능력을 가져야 한다. 기존의 디스크는 비교적 적은 크기의 텍스트 데이터를 저장할 때는 적당하나 용량이 큰 여러 데이터를 동시에 실시간으로 검색하는 VOD 서비스를 위해서는 적당하지 않으므로 디스크의 입출력 성능과 신뢰성을 보장하기 위해 디스크 스트라이핑(striping)[2]과 RAID(Redundant Arrays of Inexpensive Disks) 기술[3,4]의 적용이 제안되었다. 또한, 대부분의 VOD 서버에 관한 연구는 사용자에 대한 빠른 반응 시간과 정해진 소비율로 데이터를 전송하기 위해 디스크 탐색 연산을 최적화시키기 위한 여러가지 실시간 디스크 스케줄링 방법[5, 6, 7, 8]이나 하드 디스크에서의 비디오 데이터 저장 방법[4, 9]에 집중되었고 이와 더불어 현재 디스크 기술상의 낮은 입출력 대역폭에 따른 데이터 전송량의 한계를 극복하기 위해 디스크 멀티태스킹과 데이터 중복[9]의 활용이 제안된다.

하드 디스크는 일반적으로 테잎이나 광자기 디스크에 비해 바이트당 저장 가격이 매우 높다. 그러므로 대용량의 비디오 서버에서 비디오 데이터베이스의 저장 비용을 줄이기 위하여 비디오 데이터베이스의 영구 저장 장치를 테잎 라이브러리를 활용하는 방안이 연구 되었다[10, 11, 14]. 그러나 테잎 드라이브에서는 데이터의 임의 접근(random access)이 불가능하기 때문에 테잎에 저장된 데이터를 디스크로 옮길

때 테잎 마운팅과 테잎상의 알맞은 위치를 찾는 작업이 수행되어야 하므로 사용자의 임시 접근 제어 요구에 빠른 반응 시간을 보장할 수 없다. [11]에서는 테잎 단위로 저장된 비디오 데이터에 요구되는 서비스들의 연속적인 비디오 재생을 위한 테잎 스케줄링 방법이 연구되었고 [14]에서는 CD 주크박스에 저장된 비디오 데이터의 연속 재생을 위해 사용되는 스테이징 디스크 공간의 버퍼 관리 방법에 대해 연구되었다.

비디오 데이터의 저장 장치로 하드 디스크가 아닌 광자기 디스크 장치를 사용하게 되면 광자기 디스크는 하드 디스크보다 저렴하기 때문에 하드 디스크보다 낮은 비용으로 많은 비디오 데이터를 저장할 수 있으며 긴 디스크 실패간 평균 시간(Mean Time Between Failure: MTBF)을 가지므로 서비스되는 비디오 데이터의 양이 방대할 때 안정적인 비디오 서버 시스템을 구축할 수 있는 장점을 가진다. 하지만, 광자기 디스크 주크박스에서 광자기 디스크를 교환하는데는 수초(8-12초)의 시간이 필요하므로 실시간 처리 시스템인 VOD 서버에서는 시스템의 성능 저하를 초래하게 된다. 본 논문은 비디오 데이터의 영구 저장 장치로서 광자기 디스크 주크박스를 사용하여 디스크를 저장 매체로 사용하는 시스템보다 경제적이고 신뢰성을 지원할 수 있는 계층적 저장 구조를 갖는 복합 다단계 VOD 서버 시스템[12]에서 광자기 디스크의 교환 시간을 은폐하면서 효율적인 데이터 전송을 지원할 수 있는 비디오 데이터 저장 방법들을 제안하고 각 방법들의 특성을 분석한다.

본 논문의 구성은 다음과 같다. 2절에서는 복합 다단계 시스템의 구조와 주크박스내의 광자기 디스크를 영구 저장 매체로 사용함으로써 발생하는 문제점과 이에 대한 해결책을 제시하고, 3절에서는 복합 다단계 시스템에서의 다양한 비디오 데이터 저장 방법들을 제안한다. 4절에서는 복합 다단계 시스템의 특성을 설명하고 3절에서 기술된 다양한 저장 방법들을 비교하여 분석하고, 5절에서는 복합 다단계 시스템을 이용한 비용 효과적인 VOD 서버 구조를 제안하며, 6절에서는 종합적인 결론을 맺는다.

## 2. 복합 다단계 VOD 서버 시스템

복합 다단계 VOD 서버의 논리적인 구조는 (그림 1)

과 같으며 다수의 광자기 디스크 드라이브들과 다수의 공유 디스크, 운영 센터, 생산자, 소비자로 구성된다. 복합 다단계 VOD 서버에서 비디오 데이터베이스는 드라이브의 광자기 디스크에 영구 저장된다. 광자기 디스크의 교환 시간을 보상하기 위해 한번에 로딩하는 데이터 블록의 크기는 일정 수준 이상이 되어야 하며 데이터 블록의 크기가 증가함에 따라 메모리 버퍼 요구량이 증가하게 된다. MPEG-1 비디오 데이터의 소비율 1.5Mbps를 고려할 때 상당한 크기의 버퍼가 요구되므로 데이터를 임시적으로 저장하기 위한 저장 공간으로 메모리 버퍼만을 이용하기에는 메모리 요구량과 그 비용이 크기 때문에 버퍼를 하드 디스크에 위치시키는 스테이징 디스크(staging disk) 방법[13]을 이용할 수 있다. 본 논문에서는 드라이브에 있는 데이터 블록을 공유 디스크에 로딩하는 단계를 디스크 로딩이라고 정의하며 한번의 디스크 로딩 작업에 의해 전송되는 데이터 블록을 디스크 블록이라 정의한다. 또한 공유 디스크에 버퍼링된 데이터 블록을 소비자의 메모리로 전송하는 단계를 메모리 로딩이라 정의하며 한번의 메모리 로딩 작업에 의해 전송되는 데이터 블록을 메모리 블록이라 정의한다.

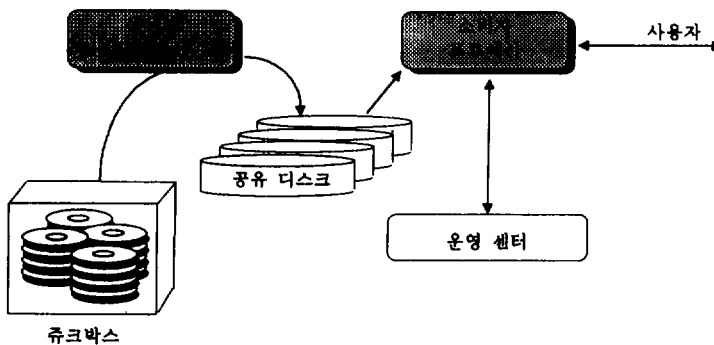
일반적으로 광자기 디스크로부터 한번에 많은 양의 데이터를 전송하는 것이 효율적이므로 하나의 디스크 블록은 여러개의 메모리 블록으로 나뉘어 메모리 로딩된다. 그러므로 공유 디스크에 버퍼링되는 데이터는 디스크 로딩 작업에 의해 생산되고 메모리 로딩 작업에 의해 소비되는 관계로 볼 수 있다. 소비자

의 메모리 로딩 작업에 의해 소비자의 메모리로 전송된 데이터 블록은 통신 네트워크를 통하여 가입자 장치로 전달된다.

생산자와 소비자 프로세서의 실질적인 하드웨어적 구성은 구축하고자 하는 VOD 서버의 서비스 용량에 따라 개인 컴퓨터, 중/대형, 병렬 컴퓨터 등으로 구현할 수 있다. 즉 네트워크 처리 능력, 입출력 버스의 대역폭, 디스크의 수 등 하드웨어적인 요소에 따라 생산자와 소비자의 구성이 다를 수 있다. 또한 생산자 프로세서와 소비자 프로세서가 독자적인 일의 처리에 간섭을 받지 않으려면 각각 독립적인 시스템 버스와 입출력 버스, 메모리를 갖고 있어야 하며, 이로 인해 디스크 로딩 작업은 메모리 로딩 작업과 중복되어 수행될 수 있다.

광자기 디스크 교환 시간이란 하나의 광자기 디스크로부터 데이터를 생산자의 메모리로 로딩한 후, 이 광자기 디스크가 적재되어 있는 광자기 디스크 드라이브에서 광자기 디스크를 꺼내어 다시 다음 사용자를 위한 광자기 디스크를 로딩하기까지의 시간을 말한다. 복합 다단계 시스템에서는 드라이브에서의 긴 광자기 디스크 교환 시간이 서버의 성능에 많은 영향을 주게 된다. 이러한 교환 시간은 다수의 드라이브를 사용하여 스케줄링을 최적화함으로써 은폐가 가능하다. 본 절에서는 광자기 디스크 교환 시간을 스페이킹 방법을 설명하기 위하여 다음의 <표 1>과 같은 시스템 변수를 설정하였다.

디스크 로딩 작업은 두 가지 서브로딩 작업으로 구



(그림 1) 복합 다단계 VOD 서버의 논리적인 구조  
(Fig. 1) Logical structure of a complex multi-level VOD server

분할 수 있으며 이들은 주크박스의 광자기 디스크에서 생산자의 메모리 버퍼로의 전송인 주크박스-메모리 로딩 작업과 생산자 메모리 버퍼에서 공유 디스크로의 전송인 메모리-공유 디스크 로딩 작업이다. <표 1>의  $D_{jm}$ 과  $D_{md}$ 는 각각 한번의 주크박스-메모리 로딩 작업에 소요되는 시간과 한번의 메모리-공유 디스크 로딩 작업에 소요되는 시간을 의미한다.  $n_{max}$ 는 시스템에서 하드웨어적으로 동시에 서비스 가능한 최대 사용자의 임계치로서 실제로 서비스할 수 있는 최대 사용자 수는 여러 가지 요인에 의하여  $n_{max}$ 보다 적을 수 있다. 각 사용자에 대한 데이터는 끊어짐 없이 연속적으로 전달 되어져야 하므로 생산자는 최대 서비스 가능한 사용자수  $n_{max}$ 명 각각에 대해 하나씩의 디스크 블럭을 디스크 로딩하는 작업들이 수행되는 시간을 한 주기로 하여 동일한 작업을 반복한다. 마찬가지로 소비자도 생산자의 한 주기 동안 로딩된 디스크 블럭들에 대해 메모리 로딩 작업과 네트워크 전송 작업을 수행하는 시간을 한 주기로하여 로딩 작업을 반복하여 수행한다. 따라서 서버 시스템은 생산자의 주기와 소비자의 주기가 같을 때 가장 효율적이다. 주기가 같지 않을 때에는 이중 큰 주기가 서버의 로딩 주기이며 이를 로딩 사이클로 정의하여  $T$ 로 표시하였다.  $S_i$ 는 각 스트림이 필요로 하는 서비스 전송률로서 MPEG-1으로 압축된 비디오 데이터에 대하여서는 1.5Mbps가 된다.

디스크 로딩의 서브로딩 중 광자기 디스크 드라이브가 관련되는 부분은 주크박스-메모리 로딩이므로 2

개 이상의 드라이브가 하나의 군을 형성하여 생산자가 하나의 드라이브로부터 주크박스-메모리 로딩 작업을 수행할 때 다른 드라이브에서는 광자기 디스크를 교체함으로써 각 군에서는 주크박스-메모리 로딩이 끊어짐 없이 발생하여 전체적으로 광자기 디스크 교환 시간을 감추는 효과를 얻을 수 있다. <표 1>에서  $k$ 는 광자기 디스크 교환을 온메시키기 위하여 군을 형성하는 데 사용되는 광자기 디스크 드라이브의 갯수를 의미한다.

(그림 2)는  $D_{jm}$ 이  $D_{md}$ 보다 길거나 같다는 가정하에  $k$ 개의 드라이브가 한 군을 형성하여 주크박스-메모리 로딩과 광자기 디스크 교환을 동시에 수행하여 광자기 디스크 교환 시간을 온메시키는 과정을 설명한 것이다.  $D_{jm}$ 과  $D_{md}$ 의 크기 비율은 I/O 채널 및 버퍼 크기와 같은 시스템의 하드웨어 구성에 따라 변화되며  $D_{jm}$ 이  $D_{md}$ 보다 짧을 경우에는 데이터 전송에 있어서 주크박스에서의 다양한 비디오 저장 방법에 대한 효과가 나타나지 않는다. 각 주크박스에 기계적인 팔이 한 개씩 존재한다는 가정하에 광자기 디스크 교환 작업을 중첩 시키기 위하여  $k$ 개의 드라이브는 각기 다른 주크박스에 있는 경우로 가정한다. 이러한 드라이브 군에 대하여 하나의 데이터 채널을 할당하면 각각의 데이터 채널에서는 데이터 로딩 작업을 끊임없이 수행할 수 있다. 그러므로 (그림 2)와 같은 스케줄링 방법을 사용하면 광자기 디스크 주크박스에서의 광자기 디스크 교환 시간을 온메시킬 수 있다.

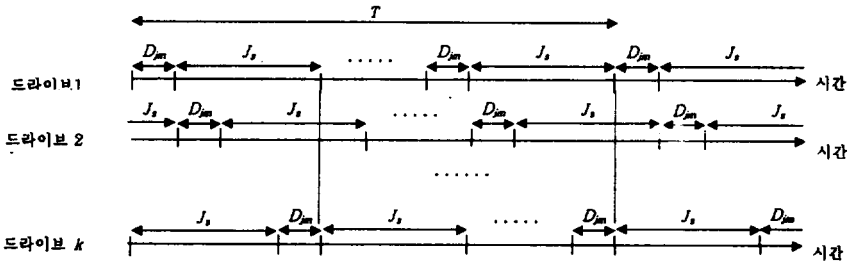
각각의 주크박스 군에서는  $D_{jm} + J_s$  시간 동안  $k$ 개의 디스크 블럭이 생산자의 메모리로 로딩되고 비디오 서버에는  $J/k$ 개의 주크박스 군이 있으므로 각각의 주크박스 군마다 독립적인 입출력 채널이 존재한다면 동일한 시간에  $J$ 개의 디스크 블럭이 로딩된다. 생산자의 주기가 서버의 로딩 사이클을 결정한다면 로딩 사이클  $T$ 는 다음과 같이 표현된다.

$$T = n_{max} \times \frac{k}{J} \times D_{jm} \tag{식 2.1}$$

한 로딩 사이클에서  $n_{max}/J$ 개의 디스크 블럭이 하나의 주크박스로부터 차례로 로딩되며 이는 하나의 주크박스에서 수용 가능한 최대 사용자 수가 된다. 그러므로 로딩 사이클  $T$ 는  $T/D_{jm}$ 개의 슬롯으로 나뉘어진다. 이러한 슬롯을 로딩 슬롯(load slot)이라 정의

<표 1> 서버 시스템 변수  
<Table 1> System parameters of the server

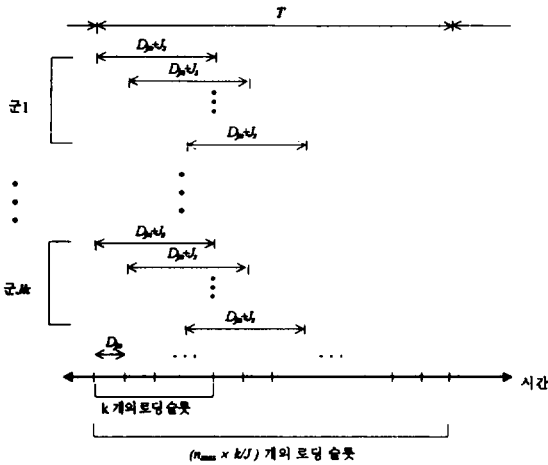
|                                    |           |
|------------------------------------|-----------|
| 사용하는 주크박스의 수                       | $J$       |
| 광자기 디스크 교환 시간                      | $J_s$     |
| 한번의 주크박스-메모리 로딩 작업에 걸리는 시간         | $D_{jm}$  |
| 한번의 메모리-공유 디스크 로딩 작업에 걸리는 시간       | $D_{md}$  |
| 디스크 블럭의 크기                         | $D_b$     |
| 동시에 지원 가능한 최대 사용자 수(최대 비디오 스트림의 수) | $n_{max}$ |
| 로딩 사이클                             | $T$       |
| 서비스에 필요한 전송률                       | $S_i$     |
| 한 군에 속하는 광자기 디스크 드라이브의 수           | $k$       |



(그림 2) 광자기 디스크 교환 시간을 은폐시키는 스케줄링 방법  
 (Fig. 2) A scheduling method to hide the optical disk swapping time

한다. 각각의 로딩 슬롯에서  $J/k$ 개의 디스크 블럭이 로딩된다. (그림 3)은 한 주기에서 생산자의 로딩 슬롯을 나타낸 것이다. 비디오 서버에서는 각 사용자에 대하여 비디오 데이터를 끊어짐 없이 스트림 형태로 발생 시켜야 하는 실시간 요구 조건을 만족 시켜야 하므로 로딩 사이클  $T$ 는 다음을 만족해야 한다.

$$T \leq \frac{D_b}{S_t} \quad (\text{식 2.2})$$



(그림 3) 로딩 사이클과 로딩 슬롯  
 (Fig. 3) Loading slots in a loading cycle

복합 다단계 VOD 시스템에서 사용자가 새로운 요구를 한 후 이에 대한 서비스가 시작되기까지의 반응 시간은 생산자 프로세서가 주크박스에 있는 광자기

디스크를 교환하여 해당 비디오 데이터를 광자기 디스크 드라이브에 로딩하고 디스크 로딩을 통해 소비자 프로세서에게 넘겨 준 후 소비자 프로세서가 메모리 로딩 및 네트워크 전송 작업 동안의 시간이 된다. 다른 사용자를 위하여 생산자와 소비자가 서비스하고 있을 때에는 이 시간이 더욱 길어질 수 있으며 이중 가장 긴 시간을 차지하는 것은 주크박스 내 광자기 디스크 교환 시간이다. 이를 최소화시키기 위한 방법으로 공유 디스크에 모든 비디오 데이터의 적정한 양의 시작 부분을 프리페칭(pre-fetching)하여 저장함으로써 사용자의 요구에 대해 디스크 시스템과 동등한 시간 내에 반응할 수 있는 방법을 고려할 수 있다. 이렇게 공유 디스크에 저장되는 각 비디오 데이터의 시작 부분을 시작 블럭이라고 정의한다. 시작 블럭의 크기는 시작 블럭의 데이터가 서비스되는 동안 주크박스에 저장된 다음 블럭을 정상적으로 서비스할 수 있는 상태를 보장하기 위한 최소한의 데이터 양으로 결정되며, 시작 블럭을 서비스하는 시간은 다음 블럭이 다른 사용자를 위해 서비스 중인 데이터 블럭들과 함께 스케줄링되어 계속적으로 서비스가 가능한 상태가 되기까지의 시간보다 길거나 같은 조건을 만족해야 한다.

비디오 서비스 중 빨리 감기 및 빨리 되감기와 같은 비디오 데이터의 임시 접근 제어에 대한 사용자 요구가 발생하면 사용자의 요구는 시작 대기시간과 동일한 이유로 인하여 대기시간이 필요하다. 근접한 곳으로의 임시 접근 제어 요구는 시스템내의 버퍼 관리 문제임으로 본 논문에서는 고려하지 않기로 한다. 일반적으로 임시 접근 제어는 사용자가 비디오 스트림

의 정확한 위치를 표현하기 보다는 시간 축에서의 편차에 비례하여 재생되어야 할 위치가 결정되며 MPEG과 같이 프레임간에 압축율이 다른 경우 정확한 위치를 찾아내는 것은 거의 불가능하다. 따라서, 사용자의 요구에 빠르게 대처하기 위해 시작 블록을 저장하는 방식과 유사하게 주크박스에 저장된 비디오 데이터 블록 중에서 시간 축에 대해 일정 간격의 블록들을 공유 디스크에 미리 저장하는 방법을 고려할 수 있다. (그림 4)는 스트림으로 이어진 비디오 데이터의 저장 형태와 소비 순서와의 관계를 나타낸 것이다. 빗금 친 부분은 공유 디스크에 프리패칭되어 상주하여야 할 블록들을 나타내며, 빗금 치지 않은 부분은 주크박스의 광자기 디스크에서 상주하여 필요할 때마다 생산자가 공유 디스크로 로딩하는 블록들을 나타낸다. 이와 같이 공유 디스크에 상주하는 각 데이터의 중간 부분을 중간 블록이라고 정의한다.

새로운 비디오 스트림이 비디오 데이터베이스에 추가될 때에 그 시작 블록과 중간 블록들이 공유 디스크에 프리패칭 되어져야 하며 이러한 블록들은 해당 비디오 스트림이 비디오 데이터베이스에서 삭제될 때까지 공유 디스크에 존재하게 된다. 그러므로 공유 디스크는 논리적으로 두개의 부분으로 나뉘어 지는데, 하나는 데이터 버퍼링을 위한 부분이며 나머지는 비디오 스트림들의 시작 블록들과 중간 블록들을 상주시키기 위한 부분이다.

(그림 4)에서 빨리 감기나 빨리 되감기와 같은 임시 접근 제어가 발생한 경우 소비자는 가입자 장치에서 현재 재생되고 있는 비디오 프레임의 위치를 고려하여 사용자가 요구한 위치에 가장 가까운 중간 블록 t

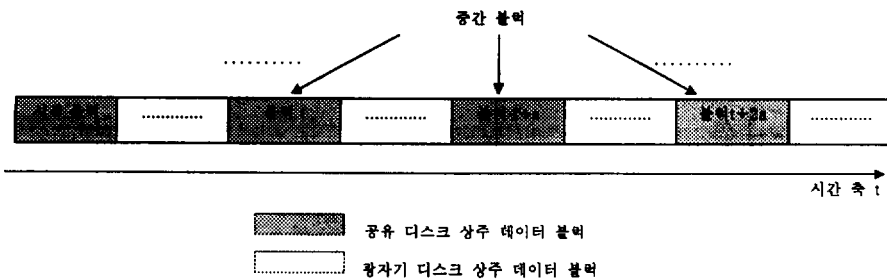
를 찾아 이 블록 t의 데이터를 사용자에게 서비스하고 이와 동시에 생산자는 동일한 비디오의 블록 t+1을 공유 디스크로 로딩한다. 중간 블록의 크기는 연속된 다음 블록이 생산자로부터 로딩될 수 있는 시간 만큼의 데이터 양이 되어야 하며 각 비디오 스트림에 중간 블록의 갯수가 많을수록 임시 접근 요구에 대해 질 높은 서비스가 가능하다. 또한 이러한 시작 블록과 중간 블록의 크기는 서버의 로딩 사이클이 클수록 증가하게 되므로 서버의 로딩 사이클이 짧을수록 동일한 디스크 용량에 더 많은 중간 블록을 수용할 수 있다.

### 3. 비디오 스트림 저장 방법

복합 다단계 서버 시스템에서 비디오 스트림들을 광자기 디스크 주크박스에 저장하는 방법으로 크게 분산 저장 방법과 비분산 저장 방법을 고려할 수 있다. 이 두 가지 저장 방법은 비디오 스트림을 구성하는 데이터 블록들을 주크박스에 할당하는 방법에 차이가 있으며 복합 다단계 시스템의 비디오 스트림 관리뿐만 아니라 성능에 많은 영향을 준다. 본 절에서는 이러한 두 가지 저장 방법을 자세히 설명하고 각각의 특성과 문제점을 제시하고 이의 해결 방안을 제안한다.

#### 3.1 비분산 저장 방법

비분산 저장 방법은 비디오 스트림의 연속된 디스크 블록들을 하나의 주크박스에 있는 광자기 디스크에 연속적으로 저장하는 방식이다. 비디오 스트림의 크기가 광자기 디스크의 용량을 초과하여 하나의 광



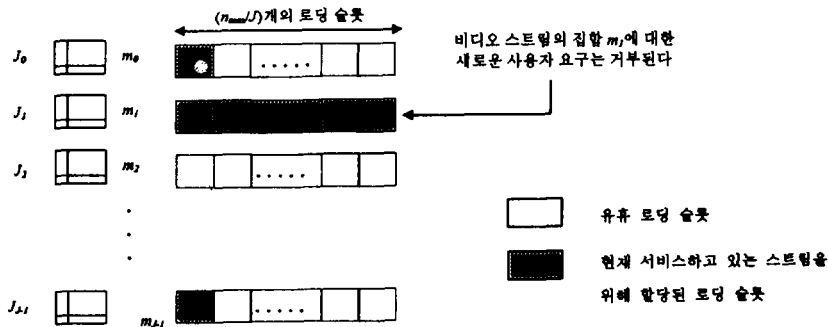
(그림 4) 비디오 데이터 블록 저장 형태  
(Fig. 4) Storage pattern for a video stream

자기 디스크에 전체 비디오 스트림을 저장할 수 없을 경우에는 동일한 주크박스에 있는 다른 광자기 디스크에 나머지 데이터 블록을 저장한다. 비분산 저장 방법에서 하나의 비디오 스트림을 추가하거나 삭제할 때 기존의 다른 비디오 스트림의 저장 장소에 영향을 받지 않으므로 비디오 데이터베이스의 관리와 광자기 디스크의 고장에 대비한 백업용 광자기 디스크 관리가 용이하다. 반면에, 2절에서 기술된 바와 같이 한 주크박스에서 동시에 서비스할 수 있는 최대 스트림의 수는 고정되어 있으므로 각 주크박스는 한 로딩 사이클안에  $n_{max}/J$ 개의 로딩 슬롯을 갖는다. 따라서 비분산 저장 방법에서 새로운 사용자의 요구에 대한 서비스 허가는 그 사용자가 요구한 비디오 스트림이 저장된 주크박스에 유휴 로딩 슬롯이 존재하는가에 의해 결정된다. 각각의 비디오 스트림은 하나의 주크박스에 저장되므로 서비스를 위해 로딩 슬롯을 할당 받는 주크박스가 고정된다. 또한 각각의 주크박스는 최대  $n_{max}/J$ 개의 스트림을 처리할 수 있으므로 요구된 비디오 스트림을 저장하고 있는 주크박스에 할당된 유휴 슬롯이 존재하면 사용자 요구는 수용되고 그렇지 않으면 그 사용자의 서비스 요구는 거부된다.

(그림 5)는 비분산 방법에서 J개의 주크박스  $J_0, J_1, \dots, J_{J-1}$ 에 할당된 생산자의 로딩 슬롯을 보여준다. 빗금 친 부분은 현재 서비스 중인 스트림을 위해 할당된 로딩 슬롯을 표시하고 빗금 치지 않은 부분은 할당되지 않은 로딩 슬롯을 나타낸다. 각 주크박스는

한 로딩 사이클에  $n_{max}/J$ 개의 로딩 슬롯을 가지므로 하나의 주크박스에서 한 로딩 사이클 T 동안 수용 가능한 최대 사용자 수는  $n_{max}/J$ 명이 된다. 즉 서버 전체적으로는 최대  $n_{max}$  스트림을 동시에 처리할 수 있는 능력을 갖고 있지만 특정 주크박스에 저장된 비디오 스트림에 대하여 사용자의 요구가 집중되면 실제 서비스 가능한 사용자 수는 그 이하가 될 수 있다. 예를 들면, (그림 5)와 같은 로딩 형태에서 새로운 사용자가  $J_1$ 에 저장된 비디오 스트림의 서비스를 요구하면 기존에  $J_1$ 에 저장된 비디오 스트림에 대한 서비스들 중 어느 한 서비스가 종료될 때까지 더 이상의 사용자 요구를 수용할 수 없다. 따라서, 비분산 방법에서 새로운 사용자의 서비스 허가 여부는 새롭게 요구된 비디오 스트림이 저장된 장소와 다른 사용자들의 요구 분포에 의존적으로 결정된다.

이와 같은 문제점을 보완하기 위하여 모든 비디오 스트림에 대하여 활용 가능성(availability)을 위한 데이터 복사본을 한 부씩 서버에서 저장하는 방법을 고려할 수 있다. 새로운 사용자가 요구한 비디오 스트림들을 서비스하기 위해서 요구된 비디오 스트림을 저장하고 있는 2개의 주크박스의 슬롯들 중 적어도 하나의 유휴 슬롯이 존재하면 새로운 요구에 대한 서비스가 가능하지만 데이터를 저장하는데 요구되는 공간이 2배로 늘어나는 단점을 갖는다. 따라서, 이러한 복사본을 많이 가질수록 사용자 요구의 집중으로 인해 서비스가 거부되는 비율이 감소하게 되는 반면에 복사본의 수가 증가함에 따라 주크박스에서의 비



(그림 5) 비분산 저장 방법과 로딩 슬롯  
(Fig. 5) Loading slots in undistributed allocation method

디오 데이터 저장 비용이 증가한다.

### 3.2 분산 저장 방법

분산 저장 방법은 디스크 스트라이핑[3]과 유사한 방식으로 하나의 비디오 스트림을 디스크 블록 단위로 나누어 다수의 쥬크박스에 스트라이핑하여 저장한다. 분산 저장 방법에서는 각각의 비디오 스트림이 다수의 쥬크박스에 나누어 저장되므로 새로운 비디오 스트림을 비디오 데이터베이스에 추가하거나 삭제할 때 전체 데이터베이스의 변경이 불가피하며 활용 가능성을 위한 복사본 또는 백업용 복사본 관리가 비분산 방법에 비하여 복잡해진다.

비분산 방법에서는 각 사용자가 서비스를 받는 동안에 생산자의 디스크 로딩 작업을 위한 로딩 슬롯은 해당 스트림이 저장된 쥬크박스의 슬롯으로 고정되지만 분산 방법에서는 서버의 로딩 사이클이 바뀔 때마다 스트림이 할당받는 로딩 슬롯은 다른 쥬크박스에 할당된 로딩 슬롯으로 변경한다. 예를 들면,  $n$ 번째 로딩 사이클  $T_n$ 에 쥬크박스  $J_i$ 에서 서비스를 받은 스트림들은  $T_{n+1}$ 에서는 쥬크박스  $J_{(i+1) \bmod J}$ 에 할당된 로딩 슬롯에서 서비스된다. (그림 6)은 이러한 로딩 형태를 나타낸 것이다. (그림 6)에서는 쥬크박스  $J_1$ 만이 로딩 사이클  $T_1$ 에서 유틸 로딩 슬롯을 갖는다. 로딩 사이클  $T_1$ 에서 새로운 사용자의 요구가 없고 서비스를 중단하는 사용자가 없는 경우 다음 로딩 사이클  $T_2$ 에서는 쥬크박스  $J_2$ 만 유틸 로딩 슬롯을 갖게 된다. 이러한 로딩 형태에서 만약 새로운 사용자가 첫 번째 디스크 블록이 쥬크박스  $J_0$ 에 저장된 스트림을 로딩 사이클  $T_1$ 에서 요구한 경우에 그 사용자를 위한 최초의 디스크 로딩 작업은 쥬크박스  $J_0$ 가 유틸 로딩 슬롯을 갖게 되는 로딩 사이클  $T_2$ 에서야 비로소 수행될 수 있다.

분산 저장 방법은 비분산 저장 방법과는 달리 비디오 스트림이 저장된 쥬크박스와 관계없이 서버의 사용자 수용 능력을 최대한으로 활용할 수 있다. 하지만, (그림 6)에서와 같이 최악의 경우에는 새로운 사용자의 요구에 대한 첫 번째 디스크 로딩 작업이  $J$  로딩 사이클 이후에 수행된다. 사용자가 비디오 서비스를 받는 도중에 빨리 감기나 빨리 되감기 등과 같은 임시 접근 제어 서비스를 요구한 경우에도 이러한 상황이 발생하게 된다. 그러므로 사용자 요구에 대해 비분산

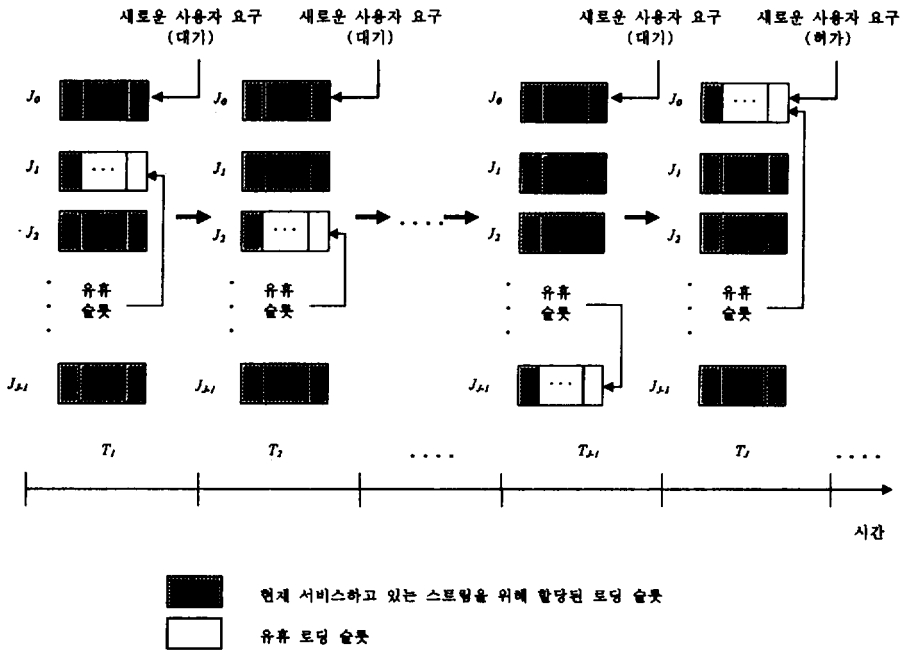
방법과 동등한 반응 시간을 보장하기 위해서는 공유 디스크에 저장 되는 시작 블록과 중간 블록의 크기는 비분산 방법의  $J$ 배가 되어야 한다.

분산 저장 방법에서 공유 디스크에 저장 되는 상주 블록의 크기를 줄이기 위하여 쥬크박스 클러스터링(jukebox clustering) 방법과 활용 가능성을 위한 복사본을 저장하는 방법을 고려할 수 있다. 쥬크박스 클러스터링이란  $d$ 개의 쥬크박스를 하나의 논리적인 쥬크박스 클러스터로 구성하여 한 비디오 스트림의 디스크 블록들을 동일한 클러스터를 구성하는 쥬크박스들에 스트라이핑하며 저장한다. 즉 클러스터간에는 비분산 방법이 적용되며 클러스터 내에서는 분산 저장 방법이 적용된다.

비디오 스트림의 디스크 블록들이 하나의 클러스터를 구성하는  $d$ 개의 쥬크박스에 스트라이핑되므로 상주 블록의 크기를 감소시킬 수 있다. 다시 말하면 쥬크박스 클러스터링에서의 상주 블록의 크기는 비분산 방법의  $d$ 배( $d < J$ )가 된다. 반면에 서버에서 수용 가능한 최대 비디오 스트림의 수는 비분산에서의 동일한 이유로 인하여 감소할 수 있다. 즉 하나의 쥬크박스 클러스터에 사용자 요구가 집중된 경우에 그 클러스터에 속한 비디오 스트림을 요구한 사용자에게 서비스는 비록 서버의 다른 클러스터에 유틸 슬롯이 존재하더라도 거부된다.

분산 저장 방법에서 공유 디스크 상주 블록의 크기를 줄이기 위한 방법으로 활용 가능성을 위한 복사본을 저장 하는 방법을 고려할 수 있다. 문제를 간단히 하기 위하여 각각의 비디오 스트림에 대하여 한 부씩의 복사본을 저장하는 경우를 가정하기로 한다.  $J$ 개의 쥬크박스를  $J_0$ 부터  $J_{J-1}$ 까지 정렬하여 동일한 비디오 스트림에 대한 두 개의 복사본을  $J/2$  만큼의 간격을 갖는 쥬크박스에 저장한다. 예를 들면, (그림 6)에서  $J_0$ 와  $J_{J/2-1}$ 가 동일한 비디오 스트림들을 저장하게 된다. 이때 로딩 주기  $T_1$ 에 요구된 새로운 사용자의 요구는  $T_{J/2-1}$ 에서 서비스될 수 있다. 따라서 분산 저장 방법에서 활용 가능성을 위한 복사본을 지원하는 방법에서는 복사본의 갯수가 늘어남에 따라 첫 번째 디스크 로딩을 수행하기까지의 대기 시간이 짧아진다. 결과적으로 사용자의 서비스 시작 요구나 빨리 감기, 되감기 등과 같은 임시 접근 제어 요구 시 빠른 반응 시간을 보장하기 위해 쥬크박스로 부터 정상적인 데





(그림 6) 분산 저장 방법과 로딩 슬롯  
(Fig. 6) Loading slots in distributed allocation method

이더 로딩 작업이 수행될 때까지 소비되어야 하는 공유 디스크 상주 불력의 크기를 감소 시킬 수 있는 장점이 있다. 반면에 이 방법은 광자기 디스크 큐크박스에 저장되는 비디오 스트림의 수가 증가하게 되므로 큐크박스에서의 저장 비용이 증가되는 단점이 있다. 또한, 이 방법은 하나의 스트림에 대한 모든 복사본들을 동일한 클러스터에 저장함으로써 큐크박스 클러스터링 방법과 병합하여 사용할 수 있다.

큐크박스 클러스터링 방법이나 활용 가능성을 위한 복사본을 저장하는 방법을 사용할 경우에는 복사본을 큐크박스에 저장하는 비용과 공유 디스크에서 시작 불력과 중간 불력들을 저장 하는데 필요한 비용간의 관계를 고려하여야 한다. <표 2>는 각 저장 방법에서 임의 접근 제어 요구 시 사용자의 최대 대기 시간과 이를 은폐시키기 위해 공유 디스크에 상주시켜야 되는 시작 및 중간 불력의 크기를 비교한 것이다. 한 로딩 사이클에 하나의 디스크 불력이 생산되고 소비되므로 비분산 방법에서 한 로딩 사이클을 서비스하기 위한 공유 디스크 상주 불력의 크기는 하나의 디스크 불력의 크기인  $D_b$ 가 된다.

<표 2> 비디오 스트림 저장 방법 비교  
<Table 2> Comparison of allocation methods

| 저장 방법                    | 공유 디스크 상주 불력이 없는 경우 대기 시간 | 필요한 공유 디스크 상주 불력의 크기 |
|--------------------------|---------------------------|----------------------|
| 비분산 저장 방법                | $T$                       | $D_b$                |
| 순수 분산 저장 방법              | $JT$                      | $JD_b$               |
| 큐크박스 클러스터링 방법            | $dT$                      | $dD_b$               |
| m개의 복사본을 갖는 큐크박스 클러스터링방법 | $dT/m$                    | $dD_b/m$             |

#### 4. 비디오 스트림 저장 방법 분석

비디오 서버에서는 비디오 데이터의 크기가 매우 크고 사용자의 요구에 따른 서비스가 엄격하게 실시간으로 제공되어야 하므로 시스템의 설계는 시스템의 궁극적인 비용과 성능에 큰 영향을 주게 되며 적절하지 못한 시스템 구성은 적절한 구성에 비해 사용자에게 서비스 중단이나 기다림을 초래할 수 있고 동

일한 수의 사용자를 서비스하기 위해 더 많은 비용을 부담해야 한다.

본 절에서는 복합 다단계 시스템의 특성을 분석하고 3절에서 제시한 주크박스에서의 다양한 저장 방법들에서 사용자의 임시 접근 제어 요구에 대한 반응 시간을 분석한다. 광자기 디스크의 데이터 전송률을  $P_i$ 라고 표시하고 하고 2 절에서 기술된 바와 같이 주크박스에서의 광자기 디스크 교환 시간을 은폐시키는 스케줄링 방법을 사용하면 디스크 블럭의 크기는 다음의 조건을 만족해야 한다.

$$D_b \geq \frac{J_s}{k-1} \times P_i \quad (\text{식 4.1})$$

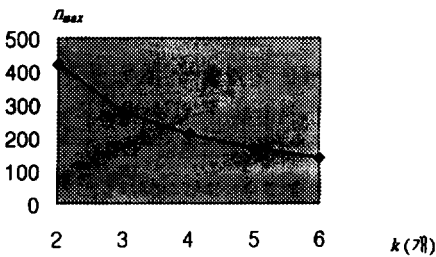
본 논문에서는 복합 다단계 VOD 서버의 특성을 분석하기 위해 HP SureStore 20xt 주크박스의 특성 값들을 사용하였다. 이 주크박스에서는 최대 16개의 광자기 디스크를 사용할 수 있고, 하나의 광자기 디스크 용량은 1.3GB이다. 또한, 광자기 디스크의 최대 교환 시간은 12초이며 데이터 전송률은 1.6MB/sec이다. 이러한 주크박스를 120개를 사용하면서 2시간 분량의

MPEG-1 비디오 스트림을 1920편 저장하는 복합 다단계 서버에서의 특성 값을 (그림 7)에 도시하였다. (그림 7-(a))는 식 (2.1)과 (2.2)에 의해 하나의 입력 채널에 할당되는 주크박스의 수  $k$ 가 2, 3, 4등으로 증가함에 따른  $n_{max}$ 의 변화를 보여주고 (그림 7-(b))는 (식 4.1)에 의해  $k$ 가 증가 함에 따른 최소의 디스크 블럭 크기의 최소값의 변화를 보여준다.

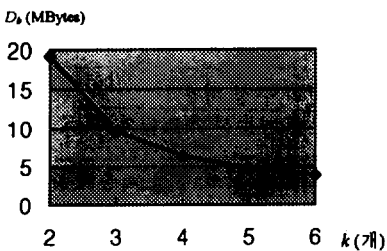
주크박스에서 다양한 비디오 데이터 저장 방법들에 대해 사용자의 임의 접근 요구 시 빠른 반응 시간을 보장하기 위해 필요한 공유 디스크 상주 블럭의 크기를 분석하기 위하여 다음과 같은 부가적인 변수를 설정하였다.  $R_i$ 는 하나의 공유 디스크 상주 블럭의 데이터가 가입자 장치에서 재생되는 시간을 나타내며,  $R_n$ 은 하나의 비디오 스트림에서 공유 디스크에 상주하는 블럭의 갯수를 표시한다. 문제를 간단히 하기 위하여 모든 비디오 스트림의 크기는 같다고 가정한다.  $V_i$ 는 하나의 비디오 스트림 전체가 가입자 장치에서 재생되는 시간을 나타낸다. 전체 비디오 스트림 중에서 공유 디스크 상주 블럭이 차지하는 비율을  $R_p$ 로 나타내면  $R_p$ 는 다음과 같이 정의된다.

$$R_p = \frac{R_i \times R_n}{V_i} \quad (\text{식 4.2})$$

사용자가 빨리 감기 또는 빨리 되감기와 같은 임의 접근 제어 서비스를 요구한 경우 사용자는 현재 사용자의 가입자 장치에서 재생되는 비디오 프레임과 사용자가 요구한 위치를 고려하여 새롭게 서비스를 시작할 공유 디스크 상주 블럭의 위치를 결정한다. 그러므로 사용자가 요구하는 위치와 실제 서버가 서비스하는 블럭의 시작 위치간에는 편차가 존재하게 되며 시간 축에서 이러한 편차의 평균을  $D_{avg}$ 로 나타낸다. (그림 8)에서는 여러 가지 시스템 변수가 변화함에 따라  $D_{avg}$ 의 변화를 도시하였다. X축은 하나의 클러스터에 속하는 주크박스의 수  $d$ 를 나타낸다. <표 2>에 제시된 각 방법에서 필요로 하는 최소의 공유 디스크 상주 데이터양을 고려하여 전체 비디오 스트림들에 대해 공유 디스크 상주 블럭이 차지하는 비율  $R_p$ 를 5, 10, 15, 20, 25, 30%로 변화시키면서  $D_{avg}$ 의 변화를 도시하였다.  $d=1$ 은 비분산 저장 방법을 의미하며  $d$ 의 값이 증가할 수록 순수 분산 저장 방법에서의  $D_{avg}$ 값으로 접근한다. (그림 8)에서는 활용 가능성을 위한



(a)

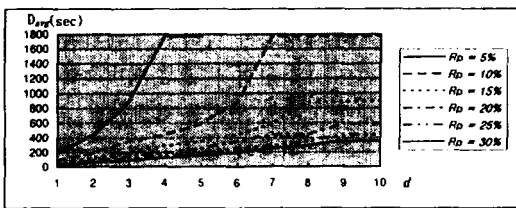


(b)

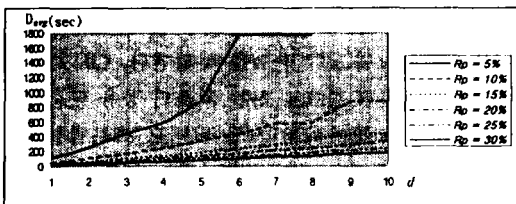
(그림 7) 시스템 변수간의 연관 관계  
(Fig. 7) Relationship among system parameters

복사본을 저장하는 경우는 고려되지 않았지만 만약 분산 저장 방법에서 각각의 비디오 스트림에 대하여  $m$ 개의 복사본을 저장한다면  $D_{avg}$ 는  $D_{avg}/m$ 으로 감소하게 된다.

(그림 8)에서  $R_b$  또는  $k$ 가 증가함에 따라  $D_{avg}$ 는 감소하며  $d$ 가 증가함에 따라  $D_{avg}$ 는 증가함을 알 수 있다.  $D_{avg}$ 는 사용자의 임시 접근 요구 위치와 실질적으로 비디오 스트림이 재생되는 위치간의 평균 편차이므로  $D_{avg}$ 가 작을 수록 임시 접근 제어에 대한 서비스의 질을 향상시킬 수 있다. 하나의 입력 채널에 할당되는 튜브박스의 수인  $k$ 를 증가 시킴에 따라 서버에서 동시에 발생 가능한 스트림의 수가 감소하게 되며  $R_b$ 를 증가시키면 비디오 스트림의 많은 부분이 공유 디스크에 상주하게 되므로 결국 공유 디스크에 상주하는 데이터에 대한 저장 비용이 증가하므로 비용-효과적인 면에서의 복합 다단계 VOD 서버의 장점이 줄어 든다. 또한, 클러스터링되는 튜브박스의 수인  $d$ 를 줄일 수록 사용자 요구의 집중에 따른 비분산 방법에서 나타나는 단점의 효과가 커지게 된다. 따라서, 효율적인 복합 다단계 VOD 서버의 설계를 위해서는 이와 같은 사항을 충분히 고려하여 요구되는 시스템의 성능에 적합하도록 시스템 변수들의 값을 설정해야 한다.



(a)  $k=4$



(b)  $k=6$

(그림 8)  $D_{avg}$  값의 변화  
(Fig. 8) Variation of  $D_{avg}$  value

### 5. VOD 시스템의 구현 비용 분석

일반적으로 디스크 시스템은 사용자의 임의 접근 요구에 대해 빠르게 반응할 수 있고 많은 수의 비디오 스트림을 동시에 서비스할 수 있는 장점을 갖고며 복합 다단계 시스템은 많은 수의 비디오 데이터를 안정적이고 저렴한 가격에 저장할 수 있는 장점이 있다. 비용 효과적인 서버를 구성하기 위해 이러한 두 VOD 서버 구조의 장점을 갖는 하이브리드(hybrid) 시스템 구조를 고려할 수 있다. 하이브리드 시스템에서는 자주 접근 되는 비디오 데이터를 디스크 시스템에서 수용하고 자주 접근 되지 않는 비디오들은 복합 다단계 시스템에서 저장한다. 따라서 디스크 시스템에서는 많은 요구 빈도를 갖는 적은 수의 비디오 스트림을 저장하여 실시간으로 서비스하며 복합 다단계 시스템에서는 요구 빈도가 낮은 많은 수의 비디오 스트림을 저장하고 이를 실시간으로 서비스 한다. VOD 서버는 하드웨어적으로 매우 다양하게 구성될 수 있으므로 이들의 구성 비용을 분석하여 비교한다는 것은 현실적으로 거의 불가능하다. 그러므로 본 절에서는 VOD시스템의 하드웨어적 구성을 가능한 추상화하여 디스크 시스템, 복합 다단계 시스템 및 하이브리드 시스템을 비교한다.

VOD 서버의 능력을 평가할 수 있는 두 가지 중요한 요소는 서버에서 저장할 수 있는 비디오 스트림의 수와 동시에 발생 가능한 스트림의 수이다. 일반성을 유지하면서 이하의 분석을 간단히 하기 위하여 모든 비디오 스트림의 크기는 동일하다고 가정하면 VOD 서버의 구성 비용은 하나의 비디오를 저장하기 위하여 필요한 단위 비용으로 비교가 가능하다. 각각의 구조에 대하여 비디오 스트림에 대한 단위 비용을 유효 비용 (effective cost :  $C$ )이라 정의하며 이는 다음과 같다.

$$C = \frac{\text{서버의 구성 비용}}{\text{서버에서 저장 가능한 스트림의 수}}$$

유효 비용은 서버의 전체 구성 비용에 대한 비디오 스트림의 평균 비용이라 할 수 있다. 마찬가지로 VOD 서버의 능력은 동시에 발생 가능한 스트림의 수에 의해 평가될 수 있다. 저장된 비디오 스트림의 수에 대한 서버에서 발생 가능한 스트림의 수를 유효 서비스 (effective service :  $S$ )라 하면 이는 다음과 같이 표시 된다.

$$S = \frac{\text{서버에서 동시에 발생 가능한 스트림의 수}}{\text{서버에서 저장 가능한 스트림의 수}}$$

디스크 시스템에서  $V$ 개의 다른 비디오 스트림을 저장하여 발생 가능한 스트림의 수, 유효 비용 그리고 유효 서비스를 각각  $n_d, C_d, S_d$ 로 표시하고 복합 다단계 시스템에서  $V$ 개의 다른 비디오 스트림을 저장하여 발생 가능한 스트림의 수, 유효 비용 그리고 유효 서비스를 각각  $n_m, C_m, S_m$ 으로 표시하기로 한다. 같은 수의 비디오 스트림을 저장할 때 일반적으로 디스크 시스템에서의 하드 디스크 드라이브의 수가 복합 다단계 시스템에서의 광자기 디스크 드라이브의 수보다 많으며 디스크의 데이터 전송률이 광자기 디스크의 전송률 보다 크기 때문에 디스크 시스템의 유효 서비스와 복합 다단계 시스템의 유효 서비스 간에는  $S_d > S_m$ 이 만족된다.

사용자들이 비디오를 접근하는 빈도는 각 비디오 스트림에 대하여 다르므로 이러한 비디오 접근 빈도를 고려하면 비디오 스트림들은 두개의 그룹으로 구분할 수 있다. 즉 비디오 스트림들은 사용자에 의해 자주 접근 되는 데이터(Frequently Accessed data: FAD)와 자주 접근 되지 않는 데이터(Infrequently Accessed data: IFAD)로 구분한다. 일반적인 80/20 법칙을 고려하면 80%의 비디오 스트림이 IFAD이며 20%의 비디오 스트림이 FAD이다. 전체 스트림 중에서 FAD와 IFAD에 속하는 스트림의 수를  $V_{FAD}$ 와  $V_{IFAD}$ 라고 표시하면 일반적으로 다음과 같은 식이 성립한다.

$$V = V_{FAD} + V_{IFAD}, \quad V_{FAD} < V_{IFAD}$$

<표 4>는 세가지 서버 구성 방식에 있어서 구성 비용과 동시에 발생 가능한 스트림 수를 나타낸 것이다.

<표 3> 서버의 구조에 따른 구성 비용과 발생 가능한 스트림의 수

<Table 3> Comparison of server structures

|            | 구성 비용                                      | 발생 가능한 최대 스트림의 수                           |
|------------|--|--|
| 디스크 시스템    | $(V_{FAD} + V_{IFAD}) \times C_d$          | $(V_{FAD} + V_{IFAD}) \times S_d$          |
| 복합 다단계 시스템 | $(V_{FAD} + V_{IFAD}) \times C_m$          | $(V_{FAD} + V_{IFAD}) \times S_m$          |
| 하이브리드 시스템  | $V_{FAD} \times C_d + V_{IFAD} \times C_m$ | $V_{FAD} \times S_d + V_{IFAD} \times S_m$ |

$C_d$ 와  $C_m$ 의 값에 따라서 각각의 서버의 능력은 변화한다. 만약  $C_d < C_m$ 이라면 디스크 시스템은 복합 다단계 시스템에 비하여 더 저렴한 비용으로 같은 용량의 비디오를 저장하여 더 많은 수의 스트림을 발생시킬 수 있으므로 구성 비용 측면에서 복합 다단계 시스템을 고려할 필요가 없다. 그러나 비디오 서버에서 수천 개 또는 그 이상의 비디오 스트림을 하드 디스크에만 저장하였을 때의 관리는 간단한 문제가 아니며 디스크 시스템에서 신뢰성을 얻기 위하여서는 데이터의 복사본을 따로 저장하거나 고장에 대비하여 복구를 위한 패리티를 가진 RAID시스템을 고려하여야만 한다. 그러나 위의 두 가지 방법은 각각 복사본이나 패리티 데이터를 위한 별도의 디스크 공간이 필요하다.

만약  $C_d > C_m$ 이라면 디스크 시스템이 복합 다단계 시스템보다 더 많은 비용으로 더 많은 스트림을 발생시키는 것을 의미한다. 디스크 시스템이 고가의 비용으로 많은 사용자를 지원하므로 이 경우 디스크 시스템과 복합 다단계 시스템의 가격 대비 성능은 당연한 결과이다. 반면에 두 시스템의 하이브리드 시스템을 고려하면 하이브리드 시스템에서는 FAD는 디스크 시스템에 저장되고 IFAD는 복합 다단계 시스템에 저장되므로 하이브리드 시스템의 구성 비용은 디스크 시스템보다는 저렴하고 복합 다단계 시스템보다는 고가이며 발생 가능한 스트림의 수는 복합 다단계 시스템보다 많고 디스크 시스템보다는 적다. 반면에 디스크 시스템에서는 자주 접근 되지 않는 데이터인 IFAD와 자주 접근되는 FAD에 대하여 동일한 저장 비용이 필요하지만 IFAD인 비디오 데이터는 고가의 저장 비용에 비해 상대적으로 적은 요구를 서비스하므로 디스크 시스템에서는 IFAD에 대하여 필요 이상의 비용을 투자하게 된다. 이 경우의 하이브리드 시스템은 비디오 스트림을 IFAD와 FAD로 구분하여 각각의 특성에 맞는 저장 장치를 이용하여 디스크 시스템이나 복합 다단계 시스템의 장점을 수용하여 보다 경제적인 서버를 구성할 수 있는 방안이 된다.

## 6. 결 론

복합 다단계 VOD 서버는 일반적인 디스크 시스템보다 더욱 저렴한 가격으로 비디오 서버를 구성할 수

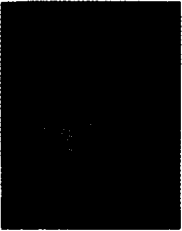
있는 방안이 될 수 있다. 이는 대용량의 비디오 데이터 베이스를 광자기 디스크에 저장하고, 다수의 프로세서가 생산자와 소비자의 관계를 가지며 생산자는 디스크 로딩 작업을 통하여 광자기 디스크에 저장된 데이터 불력을 공유 디스크에 저장하고, 소비자는 생산자에 의해 공유 디스크에 저장된 데이터를 사용자들에게 효율적으로 전달할 수 있는 시스템이다. 복합 다단계 시스템에서는 데이터의 영구 저장 장치로 광자기 디스크와 광자기 디스크 주크박스를 사용하기 때문에 광자기 디스크의 긴 교환 시간으로 인하여 발생 가능한 스트림의 수에 제약이 있으며 사용자의 요구에 대한 반응 시간이 긴 단점이 있다. 이와같은 단점은 테이어나 주크박스등의 대용량 저장 장치를 활용할 경우에 발생하는 일반적인 문제이다. 본 연구에서는 이러한 복합 다단계 VOD 서버의 문제점을 해결하기 위하여 광자기 디스크 교환 시간을 은폐시키기 위한 효율적인 스케줄링을 방법을 제안하였으며 사용자의 대기시간을 줄이기 위하여 공유 디스크에 비디오 스트림의 시작 부분과 중간 부분을 상주 시키는 방법을 제시하였다. 이와 같은 방법으로 대용량 저장 장치에 저장된 데이터를 짧은 대기시간에 실시간으로 서비스함으로써 대용량 저장 장치를 활용하여 디스크 시스템과 동등한 성능의 지원이 가능하다.

전송 및 저장 효율을 향상시키기 위해 복합 다단계 시스템의 주크박스에서 다양한 비디오 데이터 저장 방법들을 제안하였고 복합 다단계 시스템의 여러 가지 매개 변수 및 광자기 디스크 주크박스의 특성 값을 활용하여 복합 다단계 시스템의 특성과 다양한 저장 방법간의 차이를 분석하였다. 또한, VOD 서버의 다양한 구성 방식에 대하여 서버의 능력을 평가할 수 있는 두 가지 요소를 제시하고 사용자의 요구 분포에 따른 데이터 저장 비용과 발생 가능한 스트림의 수를 고려하여 비용 효과적인 VOD 서버 구성 방식을 제안하였다.

## 참 고 문 헌

- [1] D. Deloddere, W. Verbiest, and H. Verhille "Interactive Video On Demand", *IEEE Communications Magazine*, May 1994.
- [2] K. Salem and H. Garcia-Molina, "Disk Striping", *Proceeding of the Second International Conference on Data Engineering*, 1986.
- [3] D.A. Patterson, G. Gibson, and R.H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)", *ACM SIGMOD Conference*, 1988.
- [4] A.F. Tobagi, J. Pang, R. Baird, and M. Gang, "Streaming RAID™-A Disk Array Management system For Video Files", *ACM Multimedia 95*, 1995.
- [5] A.L.N. Reddy and J.C. Wyllie, "Disk Scheduling in a Multimedia I/O System", *ACM Multimedia 93*, June, 1993.
- [6] W.K. Shih, J.W.S. Liu, and C.L. Liu, "Modified Rate-Monotonic Algorithm for Scheduling Periodic Jobs with Deferred Deadlines", *Tech. Report, Computer Science Dept. Univ. of Illinois, Urbana-Champaign*, 1992.
- [7] A.L.N. Reddy and J.C. Wyllie, "I/O Issues in a Multimedia System", *IEEE Computer Magazine*, March 1994.
- [8] C.S. Freedman and D.J. DeWitt, "The SPIFFI Scalable Video-on-Demand System", *SIGMOD 95*, 1995.
- [9] S. Ghandeharizadeh and L. Ramos, "Continuous Retrieval of Multimedia Data Using Parallelism", *IEEE Trans. of Knowledge and Data Engineering*. Vol. 5, No. 4, Aug. 1993.
- [10] M.A. Kienzle, D. Sitaram, and W. Tetzlaff, "Using Tertiary Storage in Video-On-Demand Servers", *Compcon 95*.
- [11] S.W. Lau, J.C. Lui, and P.C. Wong, "A Cost-effective Near-line Storage Server for Multimedia System", 11<sup>th</sup> International Conference on Data Engineering, March 1995.
- [12] 서덕록의 4인, "복합 다단계 주문형 비디오 서버의 설계 및 구현", 한국정보처리학회 논문지, 제 4권, 제 3호, 1997년 3월.
- [13] Y. N. Doganata and N.A. Tantawi, "Making a Cost-Effective Video Server", *IEEE Multimedia*, Winter 1994.
- [14] J.Z. Wang, K.A. Hua, and H.C. Young, "A SEP

: A Space Efficient Pipelining Technique for Managing Disk Buffers in Multimedia Servers," International Conference on Multimedia Computing and Systems, June 1996.



**전 성 배**

1991년~1995년 연세대학교 컴퓨터학과(학사)

1995년~1997년 연세대학교 대학원 컴퓨터학과(석사)

1997년~현재 LG전자 미디어 통신 연구소(Media Communications Research Lab.) 재직중(연구원-Assistant Research Engineer)

관심분야: 주문형 비디오(Video On Demand), 내용기반 이미지검색기(Content-based Image Retrieval System), MPEG-4, MPEG-7



**이 원 석**

1985년 Boston Univ. 컴퓨터공학 학사

1987년 Purdue Univ. 컴퓨터공학 석사

1990년 Purdue Univ. 컴퓨터공학 박사

1992년 3월~1992년 12월 삼성전자 선임연구원

1993년~현재 연세대학교 컴퓨터학과 조교수  
관심분야: 비디오 데이터 모델링, 연역 데이터베이스, 멀티미디어 데이터베이스, 객체지향 데이터베이스