

# 비디오 저장 서버의 초기 대기시간 감소 방법

조창식, 마평수, 진윤숙, 신규상  
한국전자통신연구원 컴퓨터소프트웨어기술연구소  
e-mail : {cscho\_pmah\_yjsjin\_gsshin}@etri.re.kr

## Minimizing Initial Latency in Video Storage Servers

Changsik Cho, Pyeongsoo Mah, Yunsook Jin, Gysang Shin  
ETRI - Computer & Software Technology Laboratory

### 요 약

초기 대기시간의 감소는 동시에 지원할 수 있는 사용자 수의 최대화와 함께 비디오 저장 서버가 상용 서비스에 사용되기 위해서 갖추어야 할 필수적인 요소이다. 본 논문에서는 각 디스크의 부하가 균형을 이루고, 새로운 사용자의 디스크 읽기 요구를 우선적으로 처리하는 방법으로 초기 대기시간이 일정 시간을 넘지 않도록 하는 디스크 배치 방법과 스케줄링 방법을 제안하고, 이러한 방법을 구현하는데 필요한 시스템 변수들의 설정 방법을 제안한다. 또한 기존의 방법과의 비교를 통하여 제안한 방법의 우수한 성능 향상이 있음을 보인다.

## 1. 서론

비디오 데이터는 매우 큰 저장 공간과 전송 대역폭을 필요로 하므로 다중 디스크(disk array)의 사용은 필수적이다. 그러나 다중 디스크를 사용하면서 디스크의 read unit 을 크게 하면 라운드의 주기가 커지게 되므로, 새로운 사용자의 디스크 read 요구가 도착하였을 때부터 해당 비디오가 디스크에서 처음으로 읽어질 때까지의 초기 대기시간(initial latency)이 너무 길어지게 된다는 단점이 있다. 이러한 초기 대기시간은 사용자가 비디오를 처음으로 선택하였을 때는 물론이고, fast forward 또는 fast rewind 와 같은 VCR 제어를 선택한 경우나 임의의 위치로 점프하여 재생하는 경우에도 필요하게 된다.

초기 대기시간을 감소하기 위해서 설정된 주기별로 여러 사용자의 디스크 read 요구를 일괄적으로 처리하는 방식을 택하지 않고 사용자의 디스크 read 요구를 개별적으로 처리하는 방법을 채택한 시스템[1]도 있으나, 이러한 방법은 비디오 저장 서버가 동시에 지원할 수 있는 사용자의 수가 현저하게 감소하여 비디오 저장 서버의 경제성이 작아진다는 단점이 있다. 일부 시스템[5]에서는 개별 사용자 별로 디스크 read 요청을 하는 주기만 있고 시스템 전체적으로 여러 사용자가 공유하는 주기는 없는 경우도 있는데, 이러한 방법 역시 초기 대기시간의 감소 효과는 있으나 디스크를 first-come first-served 방식으로 처리하는 결과를 초래하여 동시에 지원 가능한 사용자의 수가 현저하게 감소하게 된다는 단점이 있다.

기존의 연구에서는 보다 많은 사용자를 지원하거나[2, 3, 4] 초기 대기시간을 감소하는 방법[1]을 별개의 연구 목표로 하였으나, 이 두 가지는 저장 서버의 경제성과 활용성에 반드시 필요한 기능이다. 본 논문에서는 비디

오 저장 서버에서 지원할 수 있는 사용자 수를 최대화 하면서 동시에 초기 대기시간을 적절한 시간 이내로 할 수 있는 방법을 제시한다.

## 2. 제안하는 방법

### 2.1 비디오 데이터 배치 및 Retrieval 방법

디스크에서 읽어오는 read unit 이 클수록 디스크 액세스의 오버헤드 비율이 작아지게 되어 디스크를 효율적으로 활용할 수 있기 때문에, 디스크의 수가  $d$ 개일 경우 본 논문에서는  $d/2$  개의 디스크에 한 라운드에서 읽어오 할 read unit 만큼의 데이터를 저장하는 방식을 제안한다. 이러한 방식은 라운드 마다  $d$  개의 디스크 전체에서 데이터를 읽어오는 기존 스트라이핑 방식에 비하여 각 디스크에서 읽어오는 데이터의 크기가 2 배가 되므로, 디스크 액세스 시간 중에서 회전 대기시간(rotational latency)의 비율을 감소하여 결과적으로 보다 많은 수의 사용자를 지원할 수 있게 한다. 그러나 이러한 방법은 각 디스크 간의 부하가 불균형되는 문제가 발생할 수 있으므로 일부 비디오를 중복으로 저장하여 load balancing 하는 방법을 제안한다.

본 논문에서는 인기도 편차가 존재함을 활용하여 저장 서버에 저장되는 비디오 중 선택 빈도가 높으리라고 예상되는 상위 10% 정도의 비디오를 인기 비디오로 분류하여 중복 저장하는 방법을 제안한다.

예를 들어 디스크가 8개인 경우 라운드 마다 읽어오는 read unit 을 4 개의 디스크에 분할하여 저장하며, 인기 비디오  $X$  의 시작 위치와 복사본인  $X'$  의 시작 위치가 다르도록 중복 저장한다. 하나의 디스크 내에서  $X, X', Y$  의

상대적 위치에 대한 제한은 없으며, 단지 각각의 데이터가 연속적인(contiguous) 공간에 저장되어 있기만 하면 된다. 인기 비디오가 시간이 지남에 따라 비인기 비디오로 분류되면 중복 저장된 2개의 파일 중 하나를 삭제하면 된다. 인기 비디오의 경우에는 좌측 디스크들과 우측 디스크들을 액세스하고 있는 사용자 수를 비교하여 사용자 수가 적은 쪽에서 시작되도록 스케줄링하는 방법으로 디스크를 사용하는 사용자의 수가 균형을 이루도록 하였다. 이러한 방법은 인기 비디오를 선택한 사용자 수가 좌측과 우측의 디스크들을 액세스하는 수의 편차보다 크기만 하면 완전하게 load balancing 이 이루어지게 된다.

### 2.2 디스크 스케줄링 방법

초기 대기시간(initial latency)이 일정한 시간 이내가 되도록 하기 위해 본 논문에서는 라운드를 여러 개의 slot 으로 나누고, 각 slot 에서는 디스크 read 요구를 SCAN 방식으로 처리하는 방식을 택하였다. 이처럼 라운드를 여러 개의 slot 으로 나누는 방법은 Philip S. Yu 등이 제안한 Grouped Sweeping Scheme(GSS)[6]에서도 사용된 방법이지만, GSS 에서는 라운드에서 처리할 모든 디스크 read 는 라운드 시작 전에 각 slot 에 균등하게 할당하며 라운드 시작 이후에 요구된 새로운 사용자의 디스크 read 요구는 다음 라운드에서 처리되기 때문에, 최악의 경우에 새로운 사용자는 라운드 시간의 2배 동안 대기한 후에 비디오의 시작 부분을 읽어오게 되는 문제가 발생한다.

이러한 문제를 해결하기 위해 본 논문에서는 각 slot 이 시작될 때 새로 시작한 비디오 스트림을 우선적으로 처리하는 방식으로 스케줄링하여 새로운 사용자를 가장 빠른 slot 에서 처리되도록 하였다. 이와 같은 스케줄링 방법에 의하여 특정 slot 이 시작된 직후에 도착한 새로운 비디오 스트림도 다음 slot 에서는 반드시 처리되므로, 본 논문의 방법에서는 새로운 사용자는 최악의 경우에도 slot 시간의 2배 이내에 디스크에서 데이터를 읽어오게 된다.

### 3. 변수 설정 방법

본 논문에서 제안하는 방법을 적용한 비디오 저장 서버를 구현하기 위해서는 라운드 시간( $T_c$ ), slot 의 수( $g$ ), 디스크 read unit 의 크기(readUnitSize), 비디오 저장 서버의 메모리 크기( $B$ ), 최대 지원 가능한 사용자 수( $N$ )를 결정하는 방법이 필요하다.

비디오 저장 서버에서 사용할 디스크를 선택하면 해당 디스크의 특성에 따라 디스크 트랙의 크기(trackSize), seek time( $t_{seek}$ ), rotation time( $t_{rot}$ ), minimum startup time( $t_{min}$ ) 등의 값이 결정되며, 디스크의 수( $d$ )는 PCI bus, SCSI card, disk 의 data bandwidth 가 균형을 되도록 적절한 수가 선택된다. 허용 가능한 초기 대기시간( $t_{latency}$ )과 비디오의 재생 속도( $V_p$ )의 값도 각각 시스템 설계자의 판단과 선택된 비디오의 타입에 의하여 그 값이 결정되므로 상수

로 처리한다.

본 논문에서는 각 라운드를  $g$  개의 slot 으로 분할하여 각 비디오의 실제 초기 대기시간이 허용 가능한 초기 대기시간( $t_{latency}$ )보다 작도록 하였으므로 각 라운드 마다  $g$  번의 디스크 seek 가 발생하게 되며, 좌측과 우측의 디스크들에서는 각각 최대  $N/2$  개의 디스크 read 요구가 처리될 수 있다.

read unit 이 하나의 디스크에만 저장된 경우 이를 읽기 위해 필요한 디스크 회전수를 나타내는 가상의 변수를  $k$  라고 할 경우, 디스크에서 read unit 크기의 데이터를 읽어오기 위해서는 이처럼 실제로 데이터를 읽는 시간(data transfer time) 외에도 디스크 헤드가 데이터의 시작 부분 위에 도달할 때 까지 기다리는 회전 대기시간(rotational latency)이 필요하므로, 한 스트림의 read unit 을 읽어오기 위해 각 디스크는 최악의 경우  $\frac{k}{d/2} + 1$  회전을 하게 된다.

따. 그러므로 본 논문의 방법에서 라운드 시간( $T_c$ )은 다음과 같은 식으로 표현된다.

$$T_c = \frac{N}{2} t_{min} + g t_{seek} + \frac{N}{2} \left( \frac{k}{d/2} + 1 \right) t_{rot} \dots\dots\dots (1)$$

또한 비디오가 중단없이 연속적으로 재생되기 위해서는 read unit 의 데이터가 클라이언트 시스템에서 재생되는 시간(readUnitSize/ $V_p$ )이 디스크 read 주기( $T_c$ ) 보다 커야 하며,  $d/2$  의 배수여야 하므로 다음과 같은 식을 도출할 수 있다.

$$ReadUnitSize = \left\lceil \frac{V_p T_c}{d/2} \right\rceil \frac{d}{2} \dots\dots\dots (2)$$

본 논문에서 제시한 방법이 사용될 경우에 최대 초기 대기시간은 slot 이 수행되는 시간( $T_c/g$ )의 2배이며, 이는 사용자가 허용하는 초기 대기시간( $t_{latency}$ ) 보다 작아야 하므로, 다음과 같은 식을 도출할 수 있다.

$$\frac{2T_c}{g} \leq t_{latency} \dots\dots\dots (3)$$

가장 경제성이 있는 메모리 크기와 라운드 시간을 결정하기 위해 위의 식을 바탕으로 slot 의 수( $g$ )가 1인 경우부터 순차적으로 계산한다.

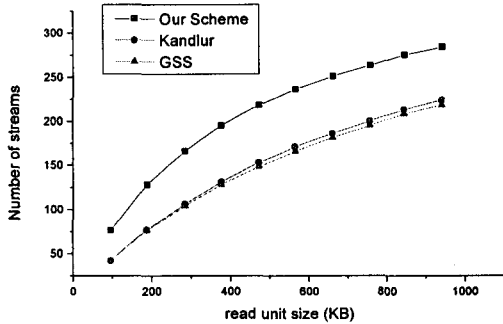
먼저 식 (3)으로부터 최대 크기의  $T_c$ 를 계산한 다음 이를 식(2)에 대입하여 readUnitSize 의 값을 구한 후, 이를 이용하여  $k$  값을 구한다. 지원 가능한 최대 사용자 수는 식 (1)에서  $N$ 을 구하는 식을 도출하여 계산한다. 즉,

$$N = \frac{2(T_c - g t_{seek})}{t_{min} + \left( \frac{2k}{d} + 1 \right) t_{rot}}$$

비디오 저장 서버에서는 read unit 크기의 데이터를 읽어오는 과정과 이를 네트워크를 통하여 전송하는 과정이 동시에 수행되므로 각 사용자 별로 2개의 read unit 를 저장하는 버퍼(buffer)가 필요하게 되며, 저장 서버의 메모리 크기를 다음 식을 사용하여 계산한다.  $B = 2 \text{ readUnitSize } N$

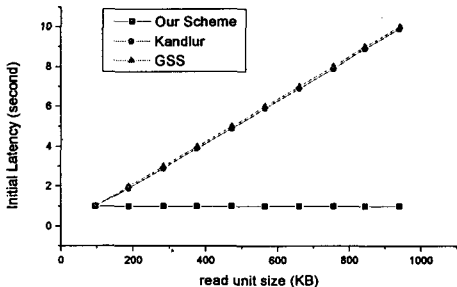
#### 4. 성능 분석 및 관련 연구와의 비교

본 절에서는 8 개의 multi-zone disks 를 사용한 경우를 기준으로 제안한 방법의 성능을 기존의 방법들과 비교하여 분석한다. (그림 1)은 기존의 방법인 [3, 6]와 제안한 방법의 최대 지원 가능한 사용자 수를 비교하였다.



(그림 1) 최대 지원 가능한 사용자 수 비교

본 연구의 구현에서 사용한 3.5초의 라운드 시간과 660 KB 의 read unit 이 선택된 경우 Kandlur 의 방법[3]은 186 명의 사용자를 지원 할 수 있고, GSS[6]에서는 본 연구의 구현에서와 같이 7 개의 slot 으로 나누어진 경우 181 명의 사용자를 지원 할 수 있다. 이러한 수치는 제안한 방법이 이들 방법보다 각각 35%와 39% 더 많은 사용자를 지원할 수 있음을 나타낸다. 이러한 성능의 향상은 각 라운드 마다  $d/2$  개의 디스크에서 데이터를 읽어올 수 있는 데이터 배치 방법으로 디스크의 효율이 높아졌기 때문이며, 인기 비디오의 중복 저장으로 소비된 10%의 디스크 저장 공간의 비용보다 월등한 소득이라고 할 수 있다.



(그림 2) 최대 초기 대기시간의 비교

(그림 2)에서는 read unit 의 증가에 따른 최대 초기 대기시간을 비교하였다. Kandlur 의 방법과 GSS 의 초기 대

기시간은 라운드 시간의 2 배이며 read unit 의 증가에 따라 라운드 시간이 증가하므로, read unit 이 커지면 초기 대기시간도 커지게 된다. 반면에 본 논문의 방법은 라운드 시간이 클 경우에는 여러 개의 slot 으로 나누어 수행되며, 새로운 사용자의 디스크 read 요구는 우선적으로 가까운 slot 에서 처리되므로 초기 대기시간은 항상 일정하다.

#### 5. 결론

제안한 방법은 기존의 스트라이핑 방법에 비하여 각 디스크에서 읽어오는 데이터의 크기가 2 배이므로 디스크를 효율적으로 활용하게 되어 사용자 수가 크게 증대되었다. 성능 비교에 나타난 바와 같이 방대한 크기의 데이터를 처리하는 비디오 저장 서버에서는 디스크의 seek time 보다 rotation time 이 라운드 시간의 대부분을 차지하므로, read unit 의 크기를 크게 할 수 있는 방법이 동시에 지원할 수 있는 사용자의 수의 최대화에 결정적인 역할을 함을 확인하였다.

기존의 방법은 read unit 의 크기에 비례하여 초기 대기시간이 커지지만, 본 논문의 방법에서는 항상 일정한 시간 이내에 새로운 사용자의 디스크 read 요구가 처리된다. 이러한 성능의 향상은 대부분의 기존 연구와 마찬가지로 각 스트림 당 read unit 의 2 배에 해당하는 더블 버퍼를 사용하되, 새로 시작한 비디오 스트림을 우선 처리하는 스케줄링 방법으로 얻을 수 있었다.

#### 참고 문헌

- [1] E. Chang and H. Garcia-Molina, "BubbleUp : Low Latency Fast-Scan for Media Servers", Proc. of the 5<sup>th</sup> ACM International Conference on Multimedia, pp.87-98, Nov. 1997.
- [2] D.J. Gemmel, H.M. Vin, D.D. Kandlur, P.V. Rangan, and L.A. Rowe, "Multimedia Storage Servers: A Tutorial", IEEE Computer, Vol. 28, No. 5, pp.40-49, May 1995.
- [3] D.D. Kandlur, M.S. Chen, and Z.Y. Shae, "Design of a Multimedia Storage Server", IS&T/SPIE Int. Symposium on Electronic Imaging: Science and Technology, San Jose, pp. 164-178, Feb. 1994.
- [4] A.L.N. Reddy and J.C. Wyllie, "I/O Issues in a Multimedia System", IEEE Computer, Vol. 27, No. 3, pp. 69-74, Mar. 1994.
- [5] Y.S. Son et al., "CrownFS : A Clustered Continuous Media server", Proc. of the International Conference on Parallel and Distributed Systems, Dec. 1997.
- [6] P.S. Yu, M.S. Chen, and D.D. Kandlur, "Grouped Sweeping Scheduling for DASD-based Multimedia Storage Management", Multimedia Systems, Vol.1, No. 1, pp. 99-109, Jan. 1993.