

# 주문형 비디오 시스템에서 효율적 버퍼 할당을 위한 동적 버퍼 할당 기법

이상호<sup>\*</sup>, 이영구, 황규영

한국과학기술원 전산학과, 첨단정보기술 연구센터

## A Dynamic Buffer Allocation Scheme for Efficient Buffer Allocation in Video-on-Demand Systems

Sang-Ho Lee<sup>\*</sup>, Young-Koo Lee, and Kyu-Young Whang

Department of Computer Science and

Advanced Information Technology Research Center (AITrc)

Korea Advanced Institute of Science and Technology (KAIST)

### 요 약

주문형 비디오 시스템에서 사용자 요청의 초기대기시간과 메모리 요구량을 줄이기 위해서는 각 사용자 요청에 할당되는 버퍼의 크기를 최소화하는 것이 필요하다. 이는 초기대기시간과 메모리 요구량이 사용자 요청에 할당되는 버퍼의 크기에 따라 지수적으로 증가하기 때문이다. 그러나 기존의 버퍼 할당 기법은 시스템이 완전 부하인 상태를 고려하여 버퍼 크기를 결정하고 할당함으로써 필요이상의 큰 버퍼를 각 사용자 요청에 할당한다. 그래서 본 논문에서는 시스템의 실행시간 정보(runtime information)를 활용하여 버퍼 크기를 결정하고 할당함으로써 불필요한 메모리 할당을 없애는 동적 버퍼 할당 기법을 제안한다. 동적 버퍼 할당 기법은 특정 버퍼 스케줄링 방식에 의존된 것이 아니기 때문에 기존의 모든 버퍼 스케줄링 방식에 적용이 가능하다. 본 논문에서는 성능 평가를 통해 동적 버퍼 할당 기법의 우수성을 보인다.

### 1. 서 론

최근의 통신 기술과 비디오/오디오 데이터의 압축과 디지털화 기술의 발전으로 네트워크를 통한 비디오 데이터의 송수신이 가능하게 되었고 이와 같은 발전은 주문형 비디오, 주문형 오디오, 온라인 지침서(on-line tutorial), 비디오 게임 등을 위한 주문형 비디오 시스템(video-on-demand systems)을 등장시켰다.

주문형 비디오 시스템은 사용자의 요청에 따라 비디오 데이터를 사용자에게 제공하는 시스템이다. 비디오 데이터의 특성으로는 사용자에게 연속적으로 데이터가 제공되어야 한다는 시간적 연속성과 데이터의 양이 크다는 대용량성이 있다. 주문형 비디오 시스템에서는 대용량인 비디오 데이터 전체를 메모리에 읽어 들일 수 없기 때문에 각 사용자가 요청한 비디오 데이터를 블록단위로 제공하는 버퍼 관리가 필요하다. 버퍼 관리에서는 비디오 데이터의 시간적 연속성을 보장하기 위해 사용자가 버퍼에서 데이터를 가져 가기 전에 필요한 데이터 블록을 버퍼로 읽어 들여야 한다.

주문형 비디오 시스템의 버퍼 관리에서는 사용자의 요청이 시스템에 도착한 후에 시스템이 비디오 데이터를 사용자에게 제공하기 시작하는 시간적인 간격인 초기대기시간(initial latency time) [Cha97b, Cha97c]과 메모리 요구량을 최소화하는 것이 중요하다. 메모리 요구량을 최소화하면 시스템은 더 적은 메모리량으로 더 많은 사용자를 지원할 수 있다. 그리고 초기대기시간의 최소화는 비디오 게임과 같이 사용자에게 빠른 응답시간을 제공해야 하는 응용에 유용할 뿐만 아니라, 빨리 보내기(fast forward)나 빨리 뒤로 보내기(fast backward)와 같은 VCR 기능의 응답시간을 줄일 수 있다. 이는

VCR 기능을 새로운 사용자 요청으로 처리하기 때문이다. [Cha97b, Ber94, Dey94, Dan94].

주문형 비디오 시스템의 버퍼 관리에 관한 기존 연구에서는 메모리 요구량과 초기대기시간을 줄이기 위해서 사용자에게 할당된 버퍼에 데이터를 채우는 순서를 결정하는 여러 버퍼 스케줄링 방식들을 제안하고 [Cha97a, Cha97b, Yu93, Dan94, Gol96], 각 사용자 요청에게 버퍼를 할당하기 위해 정적 버퍼 할당 기법을 사용하였다. 그러나 이들 버퍼 스케줄링 방식들이 사용하는 정적 버퍼 할당 기법은 서비스 중인 사용자 요청 수에 변화가 없는 정상상태(steady-state)만을 고려하여 버퍼 크기를 결정하고 할당함으로써 메모리를 비효율적으로 사용한다는 단점이 있었다 [To99]. 즉, 정적 버퍼 할당 기법은 시스템이 지원 가능한 최대수의 사용자 요청을 서비스하여 완전 부하(full load)가 된 정상상태에서 각 사용자 요청이 필요로 하는 버퍼의 최소 크기를 구하고 이를 시스템의 부하에 관계없이 동일하게 각 사용자에게 할당하였다. 그래서 이 기법은 서비스 중인 사용자 요청 수가 작은 경우에도 지원 가능한 최대수의 사용자 요청을 서비스할 때에 필요한 크기의 버퍼를 할당하기 때문에 각 사용자 요청에 필요이상으로 큰 버퍼가 할당되었다. 이로 인해 시스템의 평균 초기대기시간이 증가하고 메모리 요구량이 증가하였다. 이는 주문형 비디오 시스템에서 사용자 요청 수가 증가하면 각 사용자 요청이 필요로 하는 버퍼의 크기는 지수적으로 증가하고 이에 따라 메모리 요구량과 초기대기시간도 지수적으로 증가하기 때문이다 [Cha97a].

본 논문에서는 시스템의 평균 초기대기시간을 줄이고 불필요한 메모리 할당이 없는 동적 버퍼 할당 기법(Dynamic Buffer Allocation Scheme)을 제안한다. 동적 버퍼 할당 기법은 기존의 정적 버퍼 할당 기법과는 달리 시스템의 동적 환경을 고려하여 시스템의 실행시간 정보(runtime information)인 각 디스크에 비디오 데이터를 요청한 사용자 요청 수에 따라 각 사용자 요청에 할당할 버퍼의 크기를 구하

\* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았다.

고 이를 각 사용자 요청에 할당한다. 그래서 동적 버퍼 할당 기법은 불필요한 크기의 버퍼 할당을 제거하여 평균 초기대기시간을 감소시키고 메모리 요구량을 줄인다. 본 논문에서 제안한 동적 버퍼 할당 기법은 특정 버퍼 스케줄링 방식에 의존된 것이 아니기 때문에 기존의 버퍼 스케줄링 방식에 모두 적용이 가능하다.

본 논문은 먼저 제 2장에서 본 연구를 수행하는 주문형 비디오 시스템의 모델에 관해서 기술하고 제 3장에서는 본 논문에서 제안한 동적 버퍼 할당 기법에 관해서 논하도록 한다. 이 장에서는 동적 버퍼 할당 기법의 버퍼 할당 알고리즘과 각 사용자 요청에 할당할 버퍼의 크기에 관해서 논하도록 한다. 그리고 제 4장에서는 성능 평가를 하고 제 5장에서는 결론을 맺는다.

**2. 주문형 비디오 시스템 모델**

주문형 비디오 시스템의 기본 아키텍처는 그림 1과 같으며, 시스템은 비디오 데이터를 저장한 디스크와 각 사용자에게 할당된 버퍼, 디스크로부터 데이터를 읽어 버퍼를 채우는 서버로 구성된다. 그리고 서버는 시스템에 도착한 각 사용자 요청에 버퍼를 할당한다.

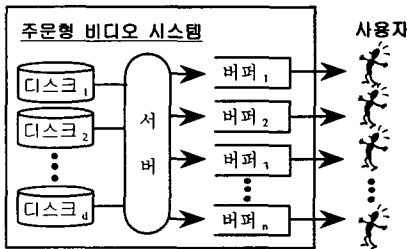


그림 1. 주문형 비디오 시스템의 기본 아키텍처.

주문형 비디오 시스템은 사용자에게 연속적으로 비디오 데이터를 제공해야 한다는 제약 조건이 있으며, 이 제약 조건을 만족시키기 위해서 서버는 각 사용자에게 할당된 버퍼가 비지 않도록 주기적으로 각 버퍼에 비디오 데이터를 채운다. 여기서 서버가 디스크로부터 데이터를 읽어 버퍼를 채우는 행위를 서비스라 하고 서버가 서비스 중인 모든 버퍼들에 비디오 데이터를 한번씩 채우는 시간 간격을 서비스 주기라 한다. 그리고 버퍼를 할당하는 시점부터 서비스 주기내에 도착할 사용자 요청을 추가 요청이라 한다.

서버가 서비스를 수행하는 방식에는 각 버퍼를 돌아 가면서 서비스하는 라운드 로빈(Round Robin) 방식[Cha97a]과 비디오 데이터가 디스크상에 위치한 순으로 서비스를 하여 디스크의 탐색 시간(seek time)을 최소화하는 스윙 방식[Cha97a], 라운드 로빈 방식과 스윙 방식을 접목시킨 GSS(Grouped Sweeping Scheduling) 방식[Yu93]이 있다. GSS 방식은 버퍼들을 여러 그룹으로 구성하여 각 그룹 내에서는 스윙 방식으로 서비스를 하고 각 그룹들은 라운드 로빈 방식으로 서비스한다.

본 논문에서는 모든 사용자가 동일한 속도로 비디오 데이터를 소비한다고 가정하고[Cha97a], 이를 비디오 데이터의 소비 속도라고 한다. 그리고 디스크로부터 데이터를 전송할 때에 발생하는 탐색 시간과 회전 지연 시간을 디스크 지연 시간(disk latency time)[Cha98a]이라 한다. 또한 본 논문에서는 디스크 지연 시간을 줄이기 위해서 비디오 데이터는 디스크에 연속적으로 저장되어 있고 하나의 버퍼에 데이터를 채우는 연산은 비선점(non-preemption)으로 수행된다고 가정한다. 그래서 하나 버퍼를 서비스할 때에는 한번의 디스크 지연만이 발생한다. 본 논문에서 사용하는 변수는 아래와 같다.

- TR: 디스크의 전송 속도.
- DR: 비디오 데이터의 소비 속도.
- DL: 디스크 지연 시간.
- T: 서비스 주기.
- BS: 버퍼 크기.
- N: 디스크가 지원 가능한 최대 사용자 요청 수.

**3. 동적 버퍼 할당 기법**

동적 버퍼 할당 기법은 정적 버퍼 할당 기법과는 달리 정상상태에서만 뿐 만 아니라, 과도상태(transient state)에서도 각 사용자 요청에 최소 크기의 버퍼를 할당하여 평균 초기대기시간을 최소화하고, 메모리 사용 효율을 높여 같은 메모리량으로도 더 많은 사용자를 서비스 할 수 있도록 하는 버퍼 할당 기법이다.

본 장에서는 먼저 제 3.1절에서 동적 버퍼 할당 기법의 버퍼 할당 알고리즘에 관해서 설명하고 이 할당 알고리즘에서 각 사용자에게 할당하는 버퍼의 크기를 결정하는 조건과 이 조건을 만족하는 버퍼 크기 결정 수식을 제 3.2절에서 설명하도록 하겠다.

**3.1. 버퍼 할당 알고리즘**

주문형 비디오 시스템에서 각 버퍼들이 메모리를 최대한 공유하도록 하기 위해서 사용자 요청이 버퍼에 있는 데이터를 사용한 후에 바로 해당 메모리를 반환하도록 한다. 그래서 반환된 메모리를 다른 버퍼들이 사용하도록 하여 시스템 전체의 메모리 요구량을 줄이도록 한다[Cha97a]. 그래서 서버는 버퍼를 서비스하기 전에 항상 버퍼를 할당하여야 하며 이 때에 버퍼 할당 알고리즘이 수행된다. 동적 버퍼 할당 기법은 서비스 중인 사용자 요청 수가 변화하는 과도상태(transient state)에서도 최소 크기의 버퍼를 할당하기 위해 실행시간 정보(runtime information)인 서비스 중인 버퍼의 수를 유지하여야 한다. 그래서 동적 버퍼 할당 기법의 버퍼 할당 알고리즘은 서비스 중인 버퍼의 서비스가 완료되었을 때에도 수행된다. 동적 버퍼 할당 기법의 버퍼 할당 알고리즘은 아래와 같다.

1. 버퍼 스케줄링 방식에 따라 서비스할 버퍼 B를 선택한다.
2. 버퍼 B가 새로 도착한 것이면 서비스 중인 버퍼의 수를 하나 증가 시키고 스텝 4를 수행한다.
3. 버퍼 B가 서비스 완료된 것이면 서비스 중인 버퍼의 수를 하나 감소 시키고 스텝 1을 수행한다.
4. 서비스 중인 버퍼의 수와 추가 요청 수에 따라 버퍼 크기를 결정하여 사용자에게 버퍼를 할당한다.

스텝 4의 버퍼 크기 결정은 제 3.2절에서 유도할 수식을 사용하여 수행된다. 그리고 버퍼 크기 결정에서 사용되는 추가 요청 수는 버퍼를 할당 받는 시점부터 서비스 주기내에 새로 도착할 사용자 요청이기 때문에 버퍼 할당하는 시점에서는 정확히 알 수 없다. 그러나 시스템의 과거 사용 기록을 사용하여 이를 추정할 수 있다.

**3.2. 버퍼의 크기**

본 절에서는 동적 버퍼 할당 기법으로 할당하는 버퍼의 크기를 구하도록 한다. 동적 버퍼 할당 기법은 정상상태만을 고려하여 버퍼를 할당하는 기존 버퍼 할당 기법과는 달리 시스템의 과도상태까지를 고려하여 버퍼를 할당한다. 시스템의 과도상태는 서비스 중인 버퍼의 수가 변화하는 상태를 말하며 이 상태에서 버퍼 크기가 만족해야 할 조건은 아래와 같다.

**조건 1:** 버퍼의 크기는 사용자 요청이 서비스 주기동안에 소비하는 데이터량보다는 커야 한다.

**조건 2:** 서비스 주기내에 서비스 중인 버퍼와 서비스 주기내에 도착하는 사용자 요청(추가 요청)을 한번씩 서비스 할 수 있어야 한다.

조건 1은 비디오 데이터의 시간적 연속성을 보장하기 위해 필요하고 조건 2는 새로 도착한 사용자 요청의 초기대기시간을 최소화하기 위해서 필요하다. 만약 조건 2가 만족하지 않으면 새로 도착한 사용자 요청은 현재의 서비스 주기내에는 서비스되지 못하고 서버가 서비스할 수 있을 때까지 기다려야 한다. 그래서 서비스 주기 T내에 도착할 추가 요청 수가 k이고 현재 서비스 중인 버퍼의 수가 n인 경우의 버퍼의 크기 BS(m)은 조건 1과 2를 만족하여야 한다. 그리고 조건 1과 2는 수식 1과 2로 표현할 수 있다. 수식에서 T×DR은 서비스 주기동안 사용자 요청이 소비하는 데이터량이고 BS<sub>s(n+k)</sub>/TR + DL은 하나의 버퍼를 서비스하는데 걸리는 시간이다.

이 두 수식을 전개하면 수식 3을 얻을 수 있고 수식 3은 과도상태에서 각 사용자 요청이 필요로 하는 최소 버퍼 크기가 되고 이를 동적 버퍼 할당 기법은 각 사용자 요청에 할당한다.

$$BS_k(n) \geq T \times DR \tag{1}$$

$$T \geq (n+k) \times \left( \frac{HS_1(n+k)}{TR} + DL \right) \times DR \tag{2}$$

$$BS_k(n) = DL \times \left[ \left( \frac{DR}{TR} \right) \times \prod_{i=1}^n (n+i \times k) \times \frac{N \times DR \times TR}{TR - N \times DR} + (n+k) \times DR \sum_{j=0}^{n-1} \left\{ \left( \frac{DR}{TR} \right)^j \times \prod_{s=2}^{j+1} (n+j \times k) \right\} \right] \tag{3}$$

$n + (e-1) \times k < N \leq n + e \times k$

**4. 성능 평가**

주분형 비디오 시스템의 버퍼 관리에서 초기대기시간과 메모리 요구량을 줄이기 위해서는 각 사용자 요청에 할당되는 버퍼의 크기를 최소화 시켜야 한다. 그래서 본 장에서는 기존의 정적 버퍼 할당 기법과 본 논문에서 제안하는 동적 버퍼 할당 기법이 할당하는 버퍼 크기를 비교하고자 한다.

본 절에서 수행하는 성능 평가는 Seagate Barracuda 9LP 디스크를 사용하는 주분형 비디오 시스템에서 평균 전송률이 1.5Bps 인 Mpeg-1 으로 인코딩된 비디오 데이터에 대해서 수행하였다. Seagate Barracuda 9LP 디스크는 전송 속도 TR이 120Mbps 이고 최대 회전 지연 시간이 8.33ms, 최대 탐색 시간이 13.4ms 인 디스크이다. 성능 평가에 사용한 버퍼 스케줄링 방식은 라운드 로빈 방식이고, 추가 요청 수는 8 이라고 가정하였다.

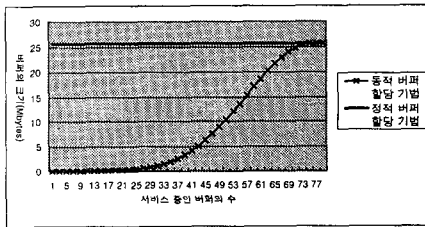


그림 2. 버퍼 할당 기법별로 할당되는 버퍼 크기 비교.

그림 2는 버퍼 할당 기법별로 할당되는 버퍼의 크기를 나타내고 있다. 그림에서 보는 바와 같이 동적 버퍼 할당 기법은 항상 정적 버퍼 할당 기법보다 적거나 같은 크기의 버퍼를 할당한다. 특히, 서비스 중인 버퍼의 수가 작은 경우에는 정적 버퍼 할당 기법보다 훨씬 적은 양의 버퍼를 할당하기 때문에 서비스 중인 버퍼의 수가 적은 경우에는 시스템의 평균 초기대기시간과 메모리 요구량이 감소하게 된다.

**5. 결 론**

본 논문은 주분형 비디오 시스템의 메모리 요구량과 평균 초기대기 시간을 줄이기 위한 새로운 버퍼 할당 기법을 제안하였다. 기존의 정적 버퍼 할당 기법은 서비스 중인 버퍼의 수에 변화가 없는 정상상태만을 고려하여 버퍼 크기를 결정하고 할당함으로써 필요이상으로 큰 버퍼를 사용자 요청에 할당하였다. 이로 인해 시스템의 평균 초기대기시간과 메모리 요구량이 증가하였다.

본 논문에서 제안한 동적 버퍼 할당 기법은 기존 버퍼 할당 기법과는 달리 시스템의 과도상태(transient state)까지를 고려하여 버퍼의 크기를 결정하고 사용자 요청에 버퍼를 할당하기 때문에 필요이상으로 큰 버퍼가 할당되는 것을 막을 수 있었다. 그래서 동적 버퍼 할당 기법은 시스템에 서비스 중인 버퍼가 작은 경우에는 기존의 정적 버퍼 할당 기법보다 훨씬 작은 크기의 버퍼를 할당하였다. 이로 인해 동적 버퍼 할당 기법은 각 사용자 요청에 할당되는 버퍼의 크기를 줄여 주분형 비디오 시스템의 평균 초기대기시간과 메모리 요구량을 줄일 수 있었다.

향후 연구에는 동적 버퍼 할당 기법을 각 버퍼 스케줄링 방식에 적용하여 이를 수치적으로 분석하고 실험을 통한 성능 평가를 수행할 예정이다. 그리고 동적 버퍼 할당 기법에서 버퍼 크기를 결정하는데 중요한 역할을 하는 추가 요청 수를 추정하는 기법을 고안할 예

정이다.

**참고 문헌**

[Ber94] Berson, S., Ghandeharizadeh, S., Muntz, R., and Ju, X., "Staggered Striping in Multimedia Information Systems," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, pp. 79-90, Minneapolis, MN, 1994.

[Cha97a] Chang, E. and Garcia-Molina, H., "Effective Memory Use in a Media Server," In Proc. 23rd Int'l Conf. on Very Large Data Bases, pp. 496-505, Athens, Greece, 1997.

[Cha97b] Chang, E. and Garcia-Molina, H., "BubbleUp: Low Latency Fast-Scan for Media Servers," In Proc. 5th ACM Int'l Conf. on Multimedia, pp. 87-98, Seattle, WA, 1997.

[Cha97c] Chang, E. and Garcia-Molina, H., "Reducing Initial Latency in Media Servers," IEEE Multimedia, Vol. 4, No. 3, pp. 50-61, 1997.

[Cha98a] Chang, E. and Garcia-Molina, H., "Cost-Based Media Server Design," In Proc. 8th Int'l Workshop on Research Issues in Data Engineering, pp. 76-83, Orlando, FL, 1998.

[Dey94] Dey-Sircar, J. K., Salehi, J. D., Kurose, J. F., and Towsley, D., "Providing VCR Capabilities in Large-Scale Video Servers," In Proc. 2nd ACM Int'l Conf. on Multimedia, pp. 25-32, San Francisco, CA, 1994.

[Dan94] Dan, A., Sitaram, D., and Shahabuddin, P., "Scheduling Policies for an On-Demand Video Server with Batching," In Proc. 2nd ACM Int'l Conf. on Multimedia, pp. 15-23, San Francisco, CA, 1994.

[Gol96] Goluchik, L., Lui, J. C. S., and Muntz, R. R., "Adaptive Piggybacking: A Novel Techniques for Data Sharing in Video-on-Demand Storage Servers," Multimedia Systems, Vol. 4, No. 3, pp. 140-155, 1996.

[Pan96] Pan, H., Ngoh, L. H., and Lazar, A. A., "A Buffer-Inventory-Based Dynamic Scheduling Algorithm for Multimedia-on-Demand Servers," Multimedia Systems, Vol. 6, No. 2, pp. 125-136, 1998.

[To99] To, T.-P. and Hamidzadeh, B., "Dynamic Real-Time Scheduling Strategies for Interactive Continuous Media Servers," Multimedia Systems, Vol. 7, No. 2, pp. 91-106, 1999.

[Sea98] Seagate, Inc., Seagate Barracuda 9LP Family Product Specification, 1998 (Available at URL: <http://www.seagate.com>).

[Yu93] Yu, P. S., Chen, M.-S., and Kandlur, D. D., "Grouped Sweeping Scheduling for DASD-based Multimedia Storage Management," Multimedia Systems, Vol. 1, No. 1, pp. 99-109, 1993.