

실시간 멀티미디어 데이터의 캐싱에 관한 연구

장정희*, 이경오*

*선문대학교 컴퓨터정보학부

e-mail:leeko@rainbow.sunmoon.ac.kr

A Study on the Realtime Multimedia Caching Schemes

Jung-Hee Jang*, Kyung-Oh Lee*

*Dept of Computer Science, Sunmoon University

요 약

실시간 멀티미디어 시스템을 위한 캐싱 기법이 몇 가지 제안되었지만 그러한 기법들은 캐쉬 히트에 따른 디스크 대역폭 절약이라는 실질적인 효과에 초점을 맞추기보다는 적중률을 높이는 데 주력을 하여왔다. 끊임이 없는 멀티미디어 데이터의 상영이 보장되지 않는다면 높은 캐쉬 적중률만으로는 성능이 우수하여 졌다고 보기는 힘들다. 본 논문에서는 이러한 멀티미디어의 특성을 고려하여 SPT-L, PSIC, DIC 등을 소개하고 있으며 이 기법에서는 끊임 없는 상영이 보장됨과 동시에 절약된 디스크 대역폭을 이용하여 더 많은 스트림을 서비스할 수 있는 기법들이다.

1. 서론

음성이나 영상을 서비스하는 멀티미디어 서버에서의 캐싱 기법은 문자나 숫자 등을 기반으로 하는 전통적인 캐싱 기법과는 여러 가지 점에 있어서 차이가 난다. 음성이나 영상 등의 스트림에서는 항상 정해진 시간 이내에 데이터가 가용하여야 하며 이 시간이 지난 후에는 의미가 없어지게 되는 실시간적인 특성을 지니고 있다.

실시간 멀티미디어 데이터를 서비스하기 위해서는 많은 버퍼 용량과 높은 디스크 대역폭이 필요하다. 따라서 서버를 구축하는 비용이 많이 들게 된다. 따라서 주어진 자원이나 비용을 가지고 보다 많은 서비스를 제공하기 위한 기법[1, 4, 6, 7]들에 대한 제공이 필요하며 멀티미디어를 위한 캐싱 기법이 이 중의 하나라 할 수 있다.

Dan 등[2]이 제안한 Interval Caching이라는 기법은 스트림들의 시간적 국소성을 이용한 기법으로 같은 멀티미디어 객체를 상영하고 있는 연속적인 두 스트림 간에 존재하는 간격(Interval) 전체를 캐쉬에 두어 두 번째 스트림은 디스크를 접근하지 않고 서

비스가 가능하게 하는 기법이다. 하지만 이 기법은 캐쉬에 있는 간격이 더 중요한 간격에 의해 대체될 수 있기 때문에 끊임 현상(hiccup)이 발생할 가능성이 있다는 문제가 있다.

Ozden 등[5]이 제안한 기법에서는 이 Interval Caching을 개선하여 간격 단위가 아니라 간격에 포함된 블록 레벨의 캐싱 기법을 제안하여 더욱 좋은 캐쉬 적중률을 갖게 하였지만 이 기법역시 끊임 현상(hiccup)이 발생할 가능성이 있다는 문제가 있다.

본 논문에서는 끊임 현상이 발생하지 않으면서도 높은 캐쉬 적중률을 갖는 기법인 STP-L, NIC, PSIC 등에 대하여 소개하고 있다.

먼저 STP-L(Space Time Product of Logical Blocks)이라는 기법은 블록 차원의 캐싱 기법으로 높은 캐쉬 적중률을 갖지만 다른 기법에 비해 계산에 대한 부담이 많다는 단점이 있다.

NIC(Non-Preemptive Interval Caching) 기법은 Interval Caching의 변종으로 한번 캐싱된 간격은 다른 간격에 의해 절대로 치환되지 않도록 하는 기법으로 끊임 현상을 방지할 수는 있으나 시스템 성

능적인 측면에서 떨어지는 문제가 있다.

PSIC(Preemptive but Safe Interval Caching)이라는 기법은 이 NIC가 갖는 성능적인 단점을 보완하기 위해 끊임 현상이 발생할 가능성이 없을 경우만 치환을 허용하는 기법이다.

2. 시스템 모델

그림 1은 실시간 멀티미디어 데이터를 위한 시스템 모델을 표현하고 있다.

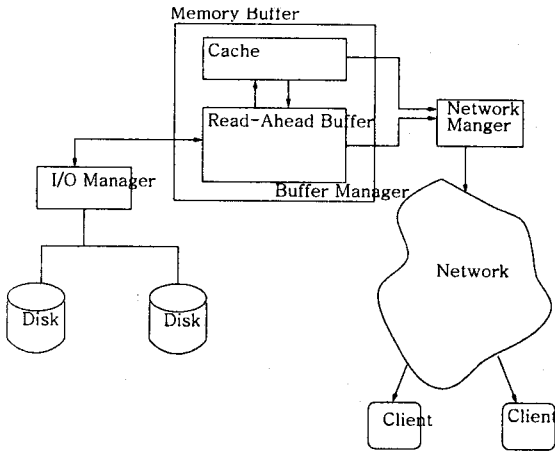


그림 1 실시간 멀티미디어 시스템 모델

디스크에 저장되어 있는 데이터는 고객의 요구에 따라 버퍼 메모리로 이동되어 네트워크를 통해 고객에게 전달되게 된다. 버퍼 메모리의 일부를 캐쉬로 사용하여 다른 고객이 다시 접근할 가능성이 높은 데이터를 메모리에 남겨둠으로써 디스크의 접근 횟수를 줄이고 결과적으로 성능을 향상시키자는 것이 본 캐싱 기법들이 지향하는 바이다.

최근의 실시간 멀티미디어 데이터의 경우 서비스의 질을 높이기 위해 가변 데이터율(VBR: Variable Bit Rate) 인코딩 방식이 많이 사용되고 있다. 이를 위한 멀티미디어 데이터 저장 기법으로 RTL(Round Time Length)[3] 기법이 있다. 이 기법에서는 VBR로 인코딩된 멀티미디어 객체를 각 주기에 필요한 데이터 블록으로 분할 한 후 Round Robin 방식으로 각 블록을 디스크 배열에 분산 배치하는 방식이다.

3. 캐싱 및 치환 기법

3.1 STP-L(Space Time Product of Logical Blocks)

이 기법은 Ozden 등[5]이 제안한 Basic이라는 기법이 32KB 블록 단위로 캐싱을 함으로 인해 계산 부담이 크게 발생하는 것을 줄이기 위해 제안된 기법으로 RTL(Rount Time Length)[3] 기법으로 저장된 멀티미디어 객체의 캐싱 단위를 물리적인 크기가 아닌 논리적 크기로 하자는 것이다. Basic에서는 각 논리적 블록의 TTNR(Time to Next Referece) 즉 해당 블록이 다음으로 참조될 때까지의 예상 시간을 계산하여 블록의 중요도를 결정한다. 이에 반해 STP-L에서는 VBR(Variable Bit Rate)로 코딩된 멀티미디어 객체에서는 동일한 시간을 상영하기 위한 데이터의 크기가 차이가 날수 있다는 점을 반영하여 TTNR에 논리적 블록의 크기를 곱한 값을 중요도의 척도로 사용함으로써 성능향상을 기할 수가 있게 되었다.

예를 들어 5초 후에 사용될 50KB 크기의 블록은 3초 후 사용될 100KB 블록보다 중요도가 높다는 것이다. Basic에서는 3초 후 사용되는 블록의 중요도가 높게 되지만 이는 전체적인 성능저하가 올 수 있다.

이 기법에서는 매 주기마다 논리적 블록에 대한 중요도를 판단하여 치환여부를 결정하여야 하고 또 논리적 블록의 숫자가 많기 때문에 계산 부담이 많이 든다는 단점이 있다.

3.2 NIC(Non-Preemptive Interval Caching)

NIC는 기본적으로 Interval Caching과 동일하게 동작한다. 그러한 한번 캐쉬가 된 간격은 상영이 끝날 때까지 치환이 일어나지 않는 점이 차이가 난다.

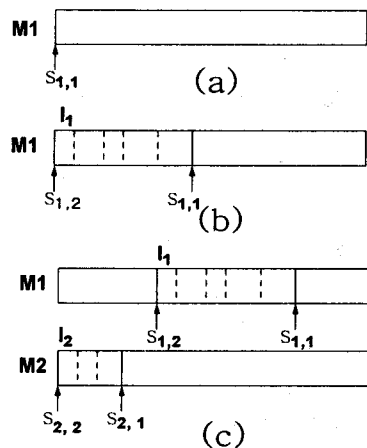


그림 2 Interval이 생성되는 예

- (a) M1이라는 멀티미디어 객체에 S_{1,1}이라는 스트림이 서비스를 개시한다.
- (b) 일정 시간이 지난 후 M1이라는 객체에 대하여 S_{1,2}라는 스트림이 서비스를 개시하면 하나의 간격 I₁이 생성된다.
- (c) I₁은 S_{1,2}가 상영을 종료할 때까지 존재한다. M₂에 대해서도 유사한 일이 발생할 수 있다.

이러한 상황에서 I₁과 I₂의 가치를 비교하여 보자. I₁은 I₂보다 크기가 크므로 당연히 I₂를 캐쉬에 보관하는 것이 더 많은 간격을 캐쉬에서 보관할 가능성이 있다. 그러나 한번 캐쉬에 저장이 되고 서비스가 개시되면 해당 간격이 없어질 때까지 캐쉬에서 지속적으로 서비스를 받게 함으로써 끊김 현상을 방지할 수가 있다. 결국 안정된 서비스를 하기 위하여 성능 저하가 생길 수 있다.

3.3 PSIC(Preemptive but Safe Interval Caching)

PSIC에서는 각 간격이 생명주기를 가지고 움직이게 된다. 각 간격의 상태변화는 다음의 그림과 같다.

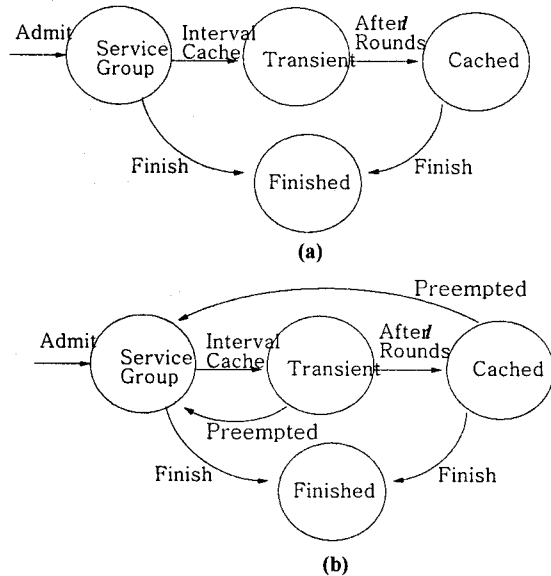


그림 3 각 간격(Interval)의 상태전이도

일단 하나의 스트림 서비스가 허용이 되면 하나의 서비스 그룹에 편제가 되고 하나의 간격이 만들어진다. 일정시간이 흐르고 나면 간격은 캐쉬에 편입될 수 있다. 이때 더 중요한 간격이 생기고 다른 간격이 캐쉬에 들어오기 위해 현재 캐쉬에 있던 간

격을 캐쉬에서 빼내야 하는 경우, 해당 간격이 캐쉬에서 빠져나오더라도 디스크 대역폭과 버퍼의 크기 등을 고려하여 끊김 현상이 발생하지 않겠다는 판단이 서면 더 중요한 간격으로 치환(Preempted)이 일어나게 되는 것이다. 그림 3에서 (a)는 양보가 일어나지 않는 NIC를 설명하는 그림이고 (b)는 양보가 가능한 PSIC에 대한 내용을 설명하고 있다.

PSIC에서 양보가 일어나더라도 문제가 발생하지 않는 지 판단하는 기법에 대하여서 본 논문의 범위를 벗어나므로 자세한 설명은 생략하도록 하였다. 하지만 양보가 가능하기 때문에 더욱 중요한 간격이 캐쉬에서 보관될 확률이 높아짐으로써 성능향상이 일어나게 된다.

4. 성능평가

성능평가를 위한 시뮬레이션을 위해서 [7]에서 제시된 MPEG-1 자료와 IBM Ultra9zx 하드 디스크의 특성이 사용되었다. 사용자의 행동 양식은 멀티미디어 성능분석에 많이 사용되는 Zipf 분포를 사용하였다.

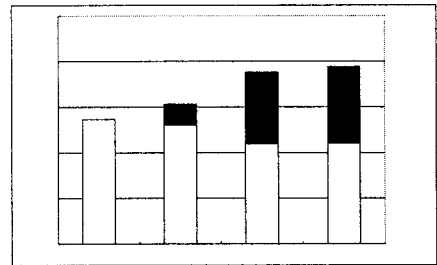


그림 4 각 기법별 성능 비교(대용량)

위의 실험에서는 Zipf 계수는 0.2가 사용되었고 버퍼의 크기는 256M, 하드디스크의 개수(RAID 스타일)는 10개로 가정하였다. 세로 축은 서비스된 스트림의 개수를 의미하고 제일 앞의 막대는 캐쉬 기법을 사용하지 않는 경우, 두 번째가 NIC, 세 번째가 PSIC, 네 번째가 STP-L이 된다. 캐쉬를 사용하지 않는 기법에 비해서 캐쉬를 사용하는 기법들은 각각 12%, 38%, 42%의 성능 향상이 있는 것을 확인할 수 있다. STP-L의 경우 성능이 가장 우수한 것으로 확인이 되고 있다. 하지만 STP-L의 경우 각 주기마다 각 논리적 블록의 중요도를 판단하는 과정이 발생되어야 하므로 상당히 많은 계산 부담이 따르는 문제가 있다. PSIC의 경우 성능도 우수하고 계

산 부담도 적기 때문에 실시간 멀티미디어 데이터를 서비스하는 경우 가장 적합하다고 할 수 있다.

그림 5에는 서버의 용량을 줄여서 디스크 1개 메모리 36M로 환경을 꾸민 후 실험을 하였다. 이 경우 그림 4의 결과와 유사한 성능이 보여지고 있다. STP-L이 가장 성능이 우수하고 PSIC이 다음이며 NIC가 캐싱 기법을 사용하지 않는 것보다는 조금 좋은 결과를 얻었다. 그러나 서버의 용량을 작게 하는 경우 캐쉬를 위한 용량이 부족하기 때문에 캐싱을 통해 얻는 이점이 줄어들다. 최근에는 그림 4의 성능을 갖는 서버는 충분히 구할 수 있는 상황이기 때문에 캐싱기법을 사용하여 실시간 멀티미디어 데이터를 서비스를 수행하는 것이 같은 가격의 서버에서 보다 높은 성능을 낼 수 있다는 실험 결과를 얻었다.

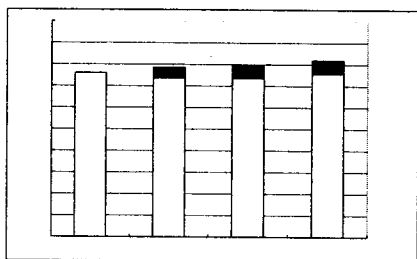


그림 5 각 기법별 성능 비교(소용량)

4. 결론

실시간 멀티미디어 데이터를 효과적으로 서비스하기 위해서는 여러 가지가 고려되어야 한다. 같은 비용을 가지고 끊임 현상이 발생하지 않으면서 더욱 더 많은 사용자에게 서비스를 제공하는 것은 멀티미디어 시스템을 설계하는데 있어서 매우 중요한 일이 될 것이다.

본 논문에서는 이러한 방안의 하나로 실시간 멀티미디어 데이터를 위한 캐싱 기법들에 대하여 살펴 보았다. 성능은 조금 떨어지지만 가장 간단한 NIC 기법과 성능은 가장 우수하지만 계산이 복잡한 STP-L 기법에 대하여 살펴보았다. PSIC 기법의 경우 안정된 서비스를 보장하면서도 성능도 우수하고 계산도 복잡하지 않은 장점을 가지고 있어 멀티미디어 서비스를 위한 캐싱에 가장 적합하다고 판단된다.

향후에는 사용자의 행동 패턴이 미리 알려진 경우 보다 성능향상을 꾀할 수 있는 캐싱 기법에 대한 연구가 진행될 것이다.

참고문헌

- [1] Scott A. Barnett and Gary J. Anido. Performability of disk-array-based video servers. *ACM Multimedia Systems Journal*, 6(1):60-74, January 1998.
- [2] Aist Dan, Rajat Mukherejee, Daniel M. Dias, Dinakr Sitaram. Buffering and caching in large-scal video servers. In *Compccon-Technologies for the Information Superhighway*, pages 217-224, Los Alamitos, CA, January 1995
- [3] KyungOh Lee, Heon Y. Yeom, "Deciding Round Length and Striping Unit Size for Large Scale Multimedia Storage Servers", *Lecture Notes in Computer Science*, Vol.1508, pp. 33-44, 1998
- [4] KyungOh Lee, Heon Y. Yeom, "An Effective Admission Control Mechanism for Variable-Bit-Rate Video Streaming", *Multimedia Systems Journal*, 7(1999) pp. 305-311
- [5] B. Ozden, R. Rastogi, and A. Silberschatz. Buffer replacement algorithms for multimedia storage systems. In *Proc. of IEEE International Conference on Multimeida Computing and Systems*, pages 172-180, Hiroshima, Japan, 1996
- [6] Jin B. Kwon, Heon Y. Yeom, "Distributed Multimedia Streaming Over Peer-to-Peer Networks", *International Conference on Parallel and Distributed Computing (Euro-Par 2003)*, August, 2003, Klagenfurt, Austria.
- [7] Jin B. Kwon, Heon Y. Yeom, "Multimedia Content Distribution Over Peer-to-Peer Networks", *IEEE International Conference on Distributed Computing Systems (Poster Session)*, May, 2003, Providence, Rhode Island, USA