

연속 미디어 서버에서 Hiccup 발생을 최소화시키기 위한 완전 연결 버퍼링 알고리즘

홍재인*⁰, 박병수***, 최명렬*

*한양대학교 전자전기제어계측공학부

**상명대학교 컴퓨터정보통신 공학부

{monadic1979*⁰, choimy*}@asic.hanyang.ac.kr, bpark@smuc.ac.kr**

A Completely Connection Buffering Algorithm for Minimizing Hiccup in Continuous Media Server

Jae-In Hong*⁰, Byoung-Soo Park***, Myung-Ryul Choi*

*Dept. of EECI, Hanyang University

**Dept. of Computer, Information and Telecommunication, Sangmyung University

요 약

본 논문에서는 멀티미디어 어플리케이션의 핵심인 오디오나 비디오와 같은 연속 미디어에서 hiccup의 발생을 최소화시키기 위한 기존의 연속적인 재생, 재생 요구 이동, 복제, N 버퍼링 기술들의 단점을 보완한 완전 연결 버퍼링 알고리즘을 제안하였다. 제안한 알고리즘은 기존의 재생 요구 이동 기술에서 불균형적인 재생 요구 분포에 따라 야기되는 지연 시간을 최소화시키기 위해 그룹, 버퍼, 디스크 사이에 완전 연결을 통하여 하나의 재생 요구에 대해 전체 디스크를 검색할 수 있는 특징이 있다. 향후 제안된 알고리즘은 구체적인 통계적 모델을 제시하여 다양한 재생 요구의 빈도 수에 따른 검증이 필요하다.

1. 서 론

오늘날 컴퓨터와 통신망 기술이 급속도로 발달함에 따라, 정보 제공 형태는 기존의 문자 위주의 정보에서 그림, 사운드, 동영상 등의 다양한 멀티미디어 정보를 제공하는 방향으로 진행되고 있다. 이러한 멀티미디어 정보 응용의 중요한 핵심은 오디오나 비디오와 같은 연속 미디어(Continuous Media, CM)라고 할 수 있다.

연속 미디어의 검색과 재생에서 중요한 것은 hiccup이 발생하지 않도록 하는 것이다. 여기서 hiccup이란 오디오에서 발생하는 랜덤 노이즈, 비디오의 지터 현상과 같은 분열과 지연 현상 등을 뜻한다. 이러한 hiccup은 저장공간, 스케줄링, 데이터 전송, 자원의 공유 등의 실시간 제약조건들에 의해 좌우되며, 그러한 제약조건이 만족되지 않았을 때 발생한다[1]. 위에서 설명한 hiccup 이외에, 연속 미디어 서비스에서 중요한 것은 시작 지연 시간을 줄이는 것이다. 시작 지연 시간이란 사용자의 요구를 받아 재생을 시작하기까지 걸리는 시간을 의미하며, 동시에 연속적인 미디어를 재생하는 경우에 중요하게 고려되어야 할 요건이다[2, 3, 4].

본 논문에서는 기존의 hiccup의 발생을 줄이기 위한 방법과 시작 지연 시간을 최소화하기 위한 재생 요구 이동, 복제, N 버퍼링 기술들을 2장과 3장에서 각각 소개

하였다. 그리고 재생 요구의 불균형적인 분포로 인해 지연 시간이 발생할 수 있는 재생 요구 이동 기술의 지연 시간 문제 요인을 해소하기 위해, 버퍼와 디스크, 재생 요구 그룹간의 완전 연결을 통해 재생 요구의 불균형적인 분포를 해소하고, 지연시간을 줄이는 방법을 4장에서 제안하였으며, 마지막으로 5장에서 결론을 맺었다.

2. 연속 디스플레이를 위한 알고리즘

연속 디스플레이를 제공하기 위해서 모든 객체와 디스크는 각각 동일한 재생율, R_c 와 동일한 데이터 전송율, R_t 를 제공한다고 가정하면, 하나의 객체 A는 연속 디스플레이를 제공하기 위해 n 개 블록(A_0, A_1, \dots, A_{n-1})으로 분할되어 전송된다. 이때 n 은 블록 사이즈를 결정하는 함수가 되며, 객체 A의 사이즈가 된다. 분할된 블록들이 디스크에 연속적으로 놓여 있다고 가정하면 디스크에서 메모리로 전송되는 전송단위가 블록이 된다. 이 때 블록은 디스크에 라운드-로빈 방식으로 연속적으로 할당되며, 하나의 블록이 재생되기 위해 요구되는 시간을 시간주기($T_p = B/R_c$)로 정의한다. 연속적인 디스플레이가 이루어지는 과정은 다음과 같다. 사용자가 객체 A를 요구하면, 시스템은 디스크에서 처음 A_0 블록을 검색하여 메모리로 전송하여 디스플레이를 초기화한다. 이때, 시간주기가 완성되기 전에 다음 A_1 블록($A_{(i+1) \bmod A}$)을 검색하

여 연속적인 디스플레이가 이루어지도록 하며, 이러한 과정들은 디스플레이가 끝날 때까지 반복된다.

몇몇 객체들을 동시에 디스플레이 하기 위해서 시간 주기는 디스크 드라이브에서 다중의 블록들이 검색되는 시간 슬롯들로 분할되며, 각 슬롯들은 고정된 크기를 갖는다. 하나의 시간 주기에서 블록을 검색할 수 있는 최대 숫자는 디스크에 의해 제공되는 디스플레이의 최대 숫자가 된다. 즉, 시스템이 D개 디스크로 구성되어 있고, 시간 주기가 N개의 슬롯으로 구성되어 있다면, 한 시간 주기 동안 동시에 디스플레이 할 수 있는 최대의 숫자는 $(N \times D)$ 이다. 이 때 한 시간 주기 동안에 제공되는 슬롯의 집합은 그룹으로 정의되고, 각 그룹들은 라운드-로빈 방식으로 디스크에 액세스하며, 시간 주기가 바뀔 때마다 다른 디스크를 액세스한다. 이때 블록을 검색하는데 소비되는 최대 지연 시간은 $(T_p \times D)$ 가 된다.

3. 기존의 시작 지연 시간을 줄이기 위한 알고리즘

3.1 재생 요구 이동 기술 (Request migration)

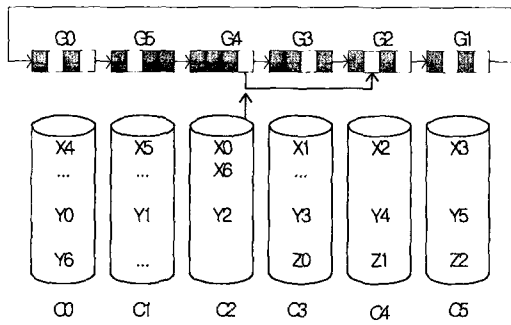


그림 1. 재생 요구 이동 기술

디스플레이의 부하가 라운드-로빈 방식으로 디스크에 할당되는데도 불구하고, 그림 1에서 도시된 G4와 G0의 경우처럼, 어떤 그룹의 부하는 다른 그룹에 할당되는 양보다 더 많이 할당될 수 있다. 이러한 부하의 불균형적인 분포는 미래에 요구되는 객체의 재생에서 시작 지연 시간을 증가시키는 결과를 가져온다.

재생 요구 이동 기술은 빈 슬롯이 없는 그룹에 재생 요구가 들어오면, 다른 빈 슬롯을 가진 그룹에 재생 요구를 넘겨주는 기술이다. 이 때 재생요구를 옮기기 위해서 임시 버퍼를 사용하게 되는데, 사용되는 버퍼의 개수는 재생 요구를 주고받는 두 그룹간의 거리 차이에 의해 결정된다.

재생 요구를 주고받는 과정을 예를 들어 설명하면, 위의 그림 1에서 시스템은 현재 C2를 검색하고 있는 G4가 전송하는 X 객체 재생 요구를, 앞으로 들어오게 될 Z 객체 재생 요구를 만족시키기 위해서 G2로 보내준다. 이때 재생 요구를 넘겨주는 자세한 과정은 다음과 같다. G4가 X0을 읽는 시간 주기 동안 G3과 G2는 각각 X1과 X2를 읽어 버퍼에 저장한다. 다음 시간 주기에는 G4는 Z0을 읽어 전송하고, G2는 X3을 읽어 버퍼에 저장하고, 버퍼에 저장된 X1이 재생된다. 또 다음 시간 주기에서

G2가 X4를 읽고, 버퍼에 있는 X2가 재생된다. 즉 X 객체의 재생 요구가 G4에서 G2로 옮겨졌으며, 새로 요구된 Z 객체의 시작 지연 시간이 $1T_p$ 로 감소하게 된다. 이 때 필요한 임시 버퍼의 개수는 3개이다. 이러한 과정을 통해 재생 요구 이동 기술은 시작 지연 시간을 줄일 수 있다. 그러나 재생 요구 이동 기술은 디스플레이를 위해 부가적인 메모리의 사용이 필요하며, 재생요구를 넘겨줄 그룹에 빈 슬롯이 존재하지 않는 경우 재생 요구 이동 기술을 사용할 수 없다. 또한 버퍼의 개수를 미리 예측할 수 없는 단점이 있다[5].

3.2 복제 기술 (Replication)

이 기술은 시스템의 시작 지연 시간을 줄이기 위해 디스크에 객체들을 복제해 두는 방식이다[3, 5].

복제 기술에서 객체 A의 원본이 주 사본 : A^p 이 되고, 복제된 객체들이 부 사본 : A^i ($1 \leq i \leq r$) 이 된다. 이 때 r은 복제된 부사본의 개수가 되어, 하나의 객체 A의 인스턴스의 수는 $(r+1)$ 이 된다. 각 복제본은 첫 번째 블록 검색의 이득을 최대화하기 위해 디스크의 수에 따라 위치가 결정된다. 객체 A의 주사본은 임의의 디스크 (A_0^p 를 갖는 d_i)에 할당이 되고, 부사본 A^i 의 첫 번째 블록은 $(d_i + jd/r + 1)$ 에서 시작된다. 이때, 객체 A의 인스턴스 개수가 2개라고 가정하면, A의 최대 시작 지연 시간은 복제를 하는 경우, 복제를 하지 않은 경우에 비하여 1/2로 줄어든다. 따라서 복제기술은 시스템에서 예상되는 시작 지연 시간을 줄일 수 있다[3].

복제 기술의 종류에는 객체를 균등하게 모두 복제하는 완전 복제, 재생 요구 빈도 수에 따라 복제하는 선택 복제, 객체의 앞부분만을 부분 복제 등의 방법들이 있다.

이러한 복제 기술은 복제할 수 있는 객체 수의 한계를 가지고 있으며 비용의 문제를 고려해야한다. 또한 부분 복제 기술에서는 부사본의 개수에 따라 hiccup이 발생할 수 있는 단점이 있다.

3.3 N 버퍼링 기술 (N Buffering)

N 버퍼링은 N개의 버퍼를 사용하여, 블록 검색 시간의 편차를 줄이는 기술이다. 이때 데드라인 스케줄링을 사용하여 블록의 검색 순서를 결정하며, 그림 2에 도시하였다.

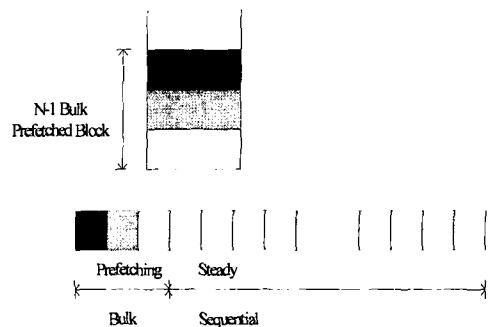


그림 2. N 버퍼링 기술

시스템은 선 인출 단계와 안정 단계로 이루어지는데, 선 인출 단계에서는 다량 선 인출 방법을 사용하여, N-1 블록을 한번에 버퍼에 저장하고, 안정 단계에서는 연차 방법을 사용하여, 버퍼가 비는 시간 동안 남은 블록을 차례대로 전송한다. 이 기술은 전송하고자 하는 객체의 처음 블록이 도착하였을 때, 디스플레이를 초기화하여 시작 지연 시간을 줄일 수 있고, 버퍼에 미리 저장해 두기 때문에 블록의 검색시간이 최대 $(N-1)T_p$ 보다 길지 않으면 hiccup이 발생하지 않는다. 그러나 버퍼의 개수를 최소화시키는 연구가 필요하며, 블록이 손실되거나 손상된 경우에 hiccup이 발생하여 지연 시간이 길어지는 단점이 있다.

4. 제안한 완전 연결 버퍼링 알고리즘

재생 요구 이동 기술은 그룹의 재생 요구가 불균형 분포를 이루는 것을 이용하여 재생 요구를 이동시킨다. 하지만 인접 그룹에 빈 슬롯이 없는 경우에는 시간 지연이 증가하는 경향이 있고, 재생 요구를 넘겨줄 수 있는 그룹이 없는 경우에는 이 기술을 사용할 수 없다. 또한 임의 버퍼의 크기가 고정되어 있지 않기 때문에 메모리 요구사항을 예측할 수 없는 단점이 있다. 이 장에서는 hiccup의 발생을 해결하기 위해 버퍼를 사용하는 방법을 제안하였으며, 그림 3에 도시하였다.

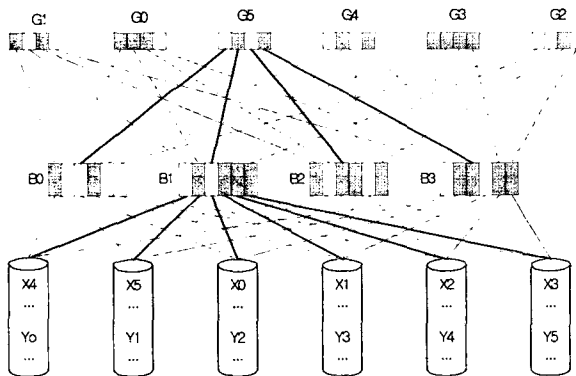


그림 3. 제안한 완전 연결 버퍼링 알고리즘

제안한 완전 연결 버퍼링 알고리즘에서 버퍼는 고정된 크기의 버퍼로 구성된 버퍼그룹(B0, B1...BN-1)의 형태로 구성된다고 가정하며, 사용되는 버퍼의 개수는 재생 요구가 저장되는 슬롯 그룹(G0, G1, ...GN-1)과 시스템의 디스크 숫자에 의해 결정된다. 즉, 버퍼그룹은 슬롯의 수와 같고, 버퍼 그룹 당 버퍼블록의 수는 디스크의 숫자와 동일하다. 또한 그림 3에 도시된 것처럼 슬롯, 버퍼, 디스크간의 연결은 완전 연결로 이루어지며, 크로스바의 형태를 취한다.

이 알고리즘에서 객체 X에 관한 재생 요구가 그룹으로 들어오면, 그룹은 버퍼에 재생 요구를 보내주고, 버퍼는 모든 디스크를 검색하고 있으므로 그에 해당하는 블록을 바로 찾아 재생한다. 이 때, 버퍼가 모든 디스크에 연결되어 있으므로 재생 요구가 어느 버퍼로 전송되는지에

관계없이 연속적으로, 요구된 객체의 블록들을 재생해줄 수 있다. 또한 재생요구는 그룹의 빈 슬롯 어디에나 할당될 수 있고, 그룹이 버퍼에 완전 연결되어 있으므로 재생 요구 이동을 하지 않아도, 요구된 객체의 전송이 가능하다. 또한 한번에 처리할 수 있는 최대 처리량은 그룹의 시간 슬롯수가 N이고 디스크의 수가 D일 때, 연속적인 재생에서와 마찬가지로 $N \times D$ 로 정의된다. 블록을 디스크에 할당하는 방법은 기존의 라운드-로빈 방식과, 임의 배치 방식 모두를 사용할 수 있으며 하나의 객체에 많은 재생 요구가 일어나는 경우를 고려하여 복제 기술을 사용할 수 있다.

향후 이 알고리즘은 연속 미디어의 통계적 모델을 제시하여 시뮬레이션을 통한 검증이 요구되며, 다양한 재생 요구의 빈도 수에 따른 효율적인 버퍼링 관리 연구가 필요하다.

4. 결론

본 논문에서는 연속 미디어에 대해서 hiccup의 발생을 최소화시키기 위한 기존의 연속 디스플레이를 소개하고, 재생 요구를 받아 재생을 시작하기까지 소비되는 시작 지연 시간을 최소화하기 위한 재생 요구 이동, 복제, N 버퍼링 기술 등을 언급하였다. 그리고 재생 요구 이동 기술에서 발생할 수 있는 지연 시간 문제를 해소하기 위해 그룹, 버퍼, 디스크사이를 크로스바의 형태로 완전 연결하여, 재생 요구를 버퍼가 즉시 처리할 수 있는 완전 연결 버퍼링 알고리즘을 제안하였다. 제안한 알고리즘은 향후 재생 요구를 균등하게 분배해주는 재생 요구 분배기를 추가하거나, 디스플레이 초기화 정보를 탐색 관리 모듈을 이용하여 저장함으로써 알고리즘의 성능을 향상시킬 수 있다.

참고 문헌

- [1] Dinkar Sitaram and Asit Dan "Multimedia Servers Applications, Environments, and Design" Morgan Kaufmann Publishers, San Francisco California.
- [2] Seon Ho Kim. " Bulk Prefetching with Deadline Driven Scheduling to Minimize Startup Latency of Continuous Media Servers." *To appear in IEEE ICME 2001*, Aug. 2001
- [3] Seon Ho Kim. "Replication Techniques to Minimize the Startup Latency of Continuous Media Servers." *To appear in SCI 2001*, July 2001.
- [4] S. Ghanaecharizadeh, S.H. Kim. "Design of Multi-user Editing Servers for Continuous Media." *Multimedia Tools and Application Journal*, 11(1), May 2000.
- [5] S. Ghanaecharizadeh, S.H. Kim, W. Shi, and R. Zimmermann, "On Minimizing Startup latency in Scalable Continuous Media Servers" *In Proceedings of Multimedia Computing and Networking*, pages 144-155. Proc. SPIE 3020, Feb. 1997.