

효과적인 미디어 스트리밍 서비스를 위한 요구 선호 버퍼 관리

방철석^o 정인범

강원대학교 컴퓨터정보통신공학과

csabang^o@snslab.kangwon.ac.kr, ibjung@kangwon.ac.kr

Request Preference Buffer Management for Effective Media Streaming Service

Cheolseok Bang^o Inbum Jung

Dept. Computer Information & Telecommunication Engineering, Kangwon National University

요 약

네트워크 환경이 발전하면서 사용자가 원하는 때에 서비스를 받을 수 있는 미디어 스트리밍 서비스에 대한 요구가 증가하고 있다. 미디어 스트리밍 서비스는 특정 시간대 선호되는 미디어들에 대한 서비스가 집중되는 현상을 발생한다. 특정한 미디어에 대한 집중적 요구를 하는 다수의 사용자들을 만족시키기 위해서는 미디어 서버내의 메인 메모리를 효과적으로 사용할 수 있는 버퍼 관리 기법이 필요하다. 본 논문에서는 미디어 서버내의 버퍼 성능 향상을 위하여 사용자 선호도를 반영한 정책을 제안하고 평가한다.

1. 서 론

컴퓨터 기술의 발달에 힘입어 사용자의 컴퓨팅 환경이 그래픽 환경으로 변하였다. 검은 화면에 텍스트로만 이루어진 환경은 일부 개발자들의 전유물이 되었다. 많은 사용자들은 좀더 멋진 그래픽 환경과 함께 고화질의 멀티미디어 데이터를 요구하고 이를 위한 애플리케이션도 많이 개발되고 있다. 최근에는 네트워크 환경이 발달함에 따라 사용자의 요구에 따라 서비스를 받을 수 있는 미디어 스트리밍 서비스의 관심도 증가하고 있다.

미디어 스트리밍 서비스는 원하는 시간에 보고 싶은 것을 볼 수 있다는 점에서 사용자들의 관심이 높다. 더 많은 사용자에게 보다 좋은 서비스를 하기 위해서는 스트리밍 서버의 효과적인 자원 관리가 반드시 필요하다. 이를 위해서 클러스터 형태로 시스템을 구성하거나 용량이 큰 미디어 데이터를 분산 저장하는 등 여러 가지 방법이 연구되고 있다.

효과적인 스트리밍 서비스를 하기 위해서 반드시 고려되어야 하는 것이 사용자의 선호도이다. 스트리밍 서비스에는 사람들의 요구가 대부분 비슷한 시간대에 이루어지고 그 대상 또한 한정되는 특징이 있다. 서버에 다수의 미디어 데이터를 저장하고 있더라도 대부분의 서비스는 현재 인기가 있는 몇 가지로 한정된다. 특정 시간대에 몇몇 미디어들에 대한 서비스가 집중되기 때문에 미디어 서버의 성능 향상을 위해서는 여러 가지 요인들을 고려해야 한다. 그 중에서도 디스크에 비해 그 가격이 비싸고 용량이 적은 메인 메모리는 대용량의 미디어 데이터를 다루는 미디어 서버의 성능에 중요한 역할을 하고 있다.

이러한 제한된 메인 메모리를 효과적으로 이용할 수 있는 방법으로 미디어 서버내의 버퍼의 히트율을 높일 수 있는 버퍼 관리 정책의 선정은 서버의 성능향상에 중요한 요인이다. 본 연구에서는 사용자의 요구 선호도에 따른 버퍼관리를 통해 버퍼들의 히트율을 높이고자 한다. 또한 기존에 사용되는 버퍼 관리 정책들과 성능 비교를 통하여 사용자 선호도가 반영된 정책이 서버 시스템의 성능 향상에 기여함을 보인다.

논문의 구성은 다음과 같다. 2장에서는 버퍼 관리의 문제점과 기존에 연구된 버퍼 대체 정책들에 대해 소개한다. 3장에서는 사용자 선호도가 반영된 버퍼정책을 설명한다. 4장에서는 소개된 버퍼 관리 정책들에 대한 실험과 성능 평가를 한다. 마지막으로 5장에서는 결론 및 향후 연구 과제에 대해 기술한다.

2. 관련연구

효과적인 버퍼 관리는 스트리밍 서비스에만 국한 되는 문제가 아니다. 운영체제, 데이터베이스, 웹 서비스 등 다른 많은 분야에서도 이는 중요한 문제이다. 효과적인 버퍼 관리를 위한 대체 정책 중 가장 많이 애용되는 것이 LRU(Least Recently Used)이다. LRU는 대부분의 응용 프로그램에서 나타나는 지역 국부성을 이용하는 정책으로 운영체제에서 많이 사용되고 좋은 성능을 보여주고 있다. 그러나 LRU 정책에는 가장 최근에 접근된 것이 항상 최우선순위를 가질 뿐 다른 사항은 고려하지 않고 있다. 이런 단점을 보완하기 위하여 LRU에 대한 많은 연구가 진행되어 왔다.

LRU 정책에서 가장 큰 문제로 지적되는 것은 접근빈도에 따른 배려가 없다는 것이다. 얼마 전까지 빈번한 접근이 이루어진 것보다 최근 한번의 접근이 우선순위가 더 높다. 이런 단점을 극복하고자 나온 정책들로는 LRU-K[1], LRFU[2] 등이 있다. 이 방법들의 주요 해결책은 버퍼에 카운터를 두거나 과거 참조 거리를 기반으로 대체를 정하는 것이다. LRU 대체 정

* 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12146-0) 지원으로 수행되었음

* 본 연구는 강원대학교 ITRC 지원을 받아 수행하였음

책에 접근 빈도라는 하나의 변수를 더 추가하여 데이터에 대한 접근 빈도를 고려하는 것이다.

EELRU 정책은 LRU 방식에서 과거 참조 거리가 버퍼보다 커서 발생하는 문제를 해결하고자 제안된 정책이다[3]. 버퍼의 크기가 주기적으로 접근하는 데이터보다 클 경우 계속해서 버퍼 미스가 발생하는 것을 막아주는 방법이다.

이런 방법들은 대체적으로 LRU 보다 좋은 성능을 나타내고 있다. 그러나 모든 경우에 적용되는 것은 아니다. 각각의 정책들은 그것이 최적의 성능을 내는 그런 환경에서만 좋은 성능을 낼 뿐이다.

사용자 수준에서 요구 미디어 스트림 당 이중 버퍼를 사용하여 사용자들에게 확실한 QoS를 보장하는 연구가 있었다[6]. 그러나 이런 시스템에서는 미디어 데이터를 저장하고 전송하는 노드에서 메모리 부족으로 인한 성능의 한계를 나타내며 시스템의 성능 확장성 제한 요인으로 되었다.

3. 사용자 선호도 기반 버퍼 관리 기법

미디어 스트리밍 서비스에서는 다수의 사용자들이 같은 미디어에 대한 요구를 동시에 그리고 산발적으로 발생한다. 그 결과 대용량의 데이터에 대한 순차적이고 반복적인 접근 형태를 나타내게 된다. 기존의 버퍼 관리 정책들은 이러한 미디어 데이터 접근 특성을 충분히 반영하지 못하므로 좋은 성능을 보장하지 못한다. 이렇게 대용량의 데이터에 순차적인 접근이 많은 시스템에서는 성능 개선을 위하여 지역 국부성을 고려하기 보다는 데이터에 대한 선호도나 프리패치와 같은 기법이 더 효과적일 수 있다.[4,5]

본 연구에서는 LRU(Least Recently Used), LFU(Least Frequently Used) 정책에 사용자 선호도를 추가하므로 버퍼에서 히트율을 높이는 선호도 기반 버퍼관리 기법을 제안한다. 선호도를 추가한 대체 정책은 기존의 LFU와 유사하지만 미디어 데이터의 선호도를 더해 주어 선호도가 높은 데이터가 대체되는 것을 한 번 더 보호할 수 있게 하였다. 사용 빈도에 선호도 수치를 더해 주어 최근 사용빈도가 낮더라도 선호도가 높은 것은 대체 선정에서 제외 시켰다. 성능 평가에서 사용한 선호도 수치는 다음 식에 의해 결정하였다.

$$\text{favor}(i) = \text{zipf}(i) \times N \quad (1)$$

zipf(i) : i 번째 영화에 대한 요청 확률
N : 전체 서비스 요청 수

식(1)에서도 알 수 있듯이 선호도 수치는 zipf[7] 분포와 N의 곱을 이용하였다. zipf 분포는 해당 인덱스에 대한 요청 확률을 나타내기 때문에 1 미만의 실수로 나타난다. 한 번의 참조가 이루어지면 카운트는 같은 1씩 증가하는데 선호도에 따른 차이가 1미만이라면 선호도 수치는 의미가 없어진다. 또 실수 수치를 그대로 이용한다면 계속되는 실수 연산으로 인해 성능 저하를 초래할 수도 있다. 이런 이유로 사용자의 요청에 유연하게 대응하면서 선호도 수치를 정수화 하는 방법으로 N을 곱하였다. 이 수치는 실제 서비스를 요청하는 수이기도 하다. 이렇게 하면 전체 사용자의 요청이 증가하면 자연스럽게 선호도 수치도 증가하고 반대의 경우에는 수치가 감소하여 전체 요청에 맞게 선호도 수치를 반영할 수 있다.

4. 성능 평가

4.1 성능 평가 방법

성능 평가는 AMD 2000+ MP Dual CPU와 2GByte의 메모리가 탑재된 리눅스 시스템에서 이루어 졌다. 총 3가지 LRU, LFU, LRU+가중치, 정책들에 대한 성능평가를 수행하였다. 기존의 버퍼관리 정책으로는 LRU와 LFU를 이용하였고 여기에 미디어 데이터의 선호도를 더하여 우선순위를 결정하는 LRU+가중치 방법을 추가하였다.

식 (1)에서의 전체 서비스 요청을 나타내는 N은 500으로 하였고 버퍼의 사이즈는 10KByte로 하였다. 버퍼 사이즈를 10KByte로 한 이유는 네트워크 상에 데이터를 전송할 때 10KByte가 가장 효과적인 결과를 반영한 것이다[6].

사용자의 요청에 대한 빈도는 포아송 분포를 이용하였다. zipf 분포의 선호도를 결정짓는 skew factor는 0.271을 기준으로 0.1355, 0.542 3가지 경우로 나누었다.[7] 각각의 경우에 따라 버퍼 수를 달리 하여 버퍼 히트율을 측정하였으며 포아송 분포의 λ는 2로 하여 초당 2개의 요청을 발생시켰다.

4.2 성능 평가 결과

성능측정에서 우선 수행한 것은 zipf 분포의 skew factor를 달리 하였을 경우 버퍼 수 변화에 따른 히트율이다. skew factor 값은 0에 가까울수록 선호도의 변화가 많이 나타나고 1에 가까울수록 고르게 나타난다. 본 실험에서 skew factor 0.271을 기준으로 선호도의 변화가 보다 심한 0.1355와 정도가 덜한 0.542를 이용하여 성능을 측정하였다[7]. 선호도 변화가 심한 경우인 skew factor 0.1355를 스트리밍 서비스에 비추어 보면 500 번의 사용자 요청 중 150 번이 한 두 개의 미디어 데이터에 집중되는 경우를 의미한다.

그림 1,2,3 은 3가지의 skew factor에 대하여 각각 성능을 측정한 결과이다. 대체적으로 3가지의 경우가 큰 차이를 보이지는 않는다. LRU와 LFU는 거의 비슷한 성능을 보이고 있으며 선호도에 따라 가중치를 부여한 정책이 약 4~5% 정도의 성능 향상을 보여주고 있다. 기존의 정책에 선호도만을 추가하여 이와 같은 성능 향상을 볼 수 있었다.

그림 4의 결과에 나타나듯이 선호도를 반영한 버퍼관리 정책들 간의 비교에서는 선호도 변화가 심한 skew factor 0.1355에서 가장 좋은 성능을 보였다. 이런 결과는 접근하는 데이터가 1,2개의 미디어 데이터에 집중하면서 버퍼의 재사용이 늘어나기 때문이다. 그러나 skew factor에 따라 많은 변화를 기대하였으나 실제 성능 평가에서는 약 2%의 성능 차이만을 보여 주었다. 이는 skew factor가 증가하는 만큼 사용자의 요구가 증가하지는 않기 때문이다. 예를 들어 skew factor가 0.271에서 0.1355 로 두 배 변하여도 전체 사용자 요구 500 회 중 특정 미디어에 대한 요구는 130회에서 150회로 그 변화의 폭이 크지 않다.

실험결과로 평가할 때 사용자 선호도를 미디어 서버 버퍼관리 기법들에 적용하므로 성능 향상을 관찰할 수 있었다. 선택된 선호도 수치들 사이에 성능은 큰 차이를 발생하지 않았지만, 실제 미디어 서버 환경은 본 연구에서 사용한 사용자 선호도 Zipf 분포 값들보다 특정한 미디어에 대한 사용자 요구 분포가 더욱 극단적일 것으로 예상되므로 본 연구에서 제안한 선호도 기반 버퍼 관리 기법이 효과적일 것이다.

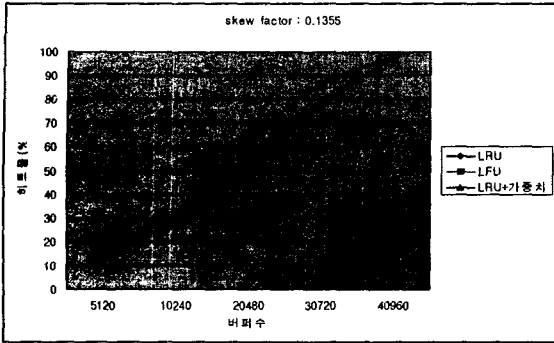


그림 1 : skew factor가 0.1355 인 경우

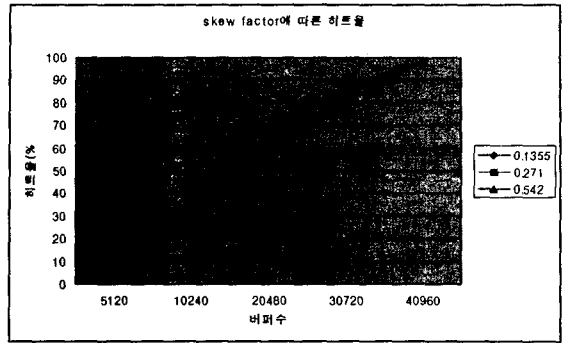


그림 4 : skew factor 변화에 따른 버퍼 히트율

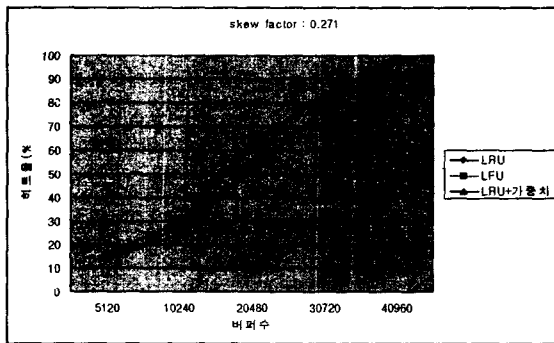


그림 2 : skew factor가 0.271 인 경우

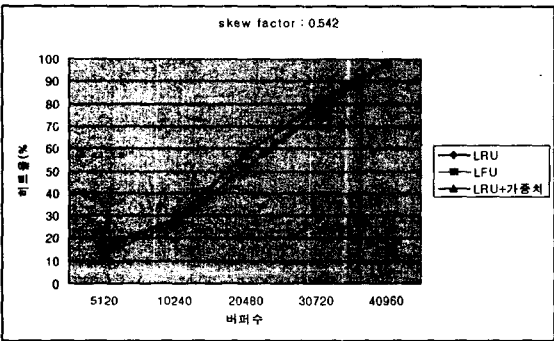


그림 3 : skew factor가 0.542 인 경우

4. 결론 및 향후 연구

미디어 스트리밍 서비스는 대용량의 미디어 데이터를 다루고 특정 미디어에 대한 사용자 선호가 집중되므로 시스템의 버퍼 관리가 서버 성능에 중요한 영향을 미친다. 본 연구에서는 이런 문제에 효과적으로 대처하고자 기존의 버퍼 기법들인 LRU, LFU 정책 외에 사용자들의 특정 미디어에 대한 선호도를 반영한 버퍼 정책을 제안하였다. 성능 평가 결과 사용자 선호도 기반 버퍼 관리 기법은 히트율을 개선하므로 서버의 성능 향상에 기여함을 관찰하였다.

향후에는 본 연구 결과를 바탕으로 미디어 데이터의 접근 특성을 고려한 프리패치 기법에 대한 연구도 이루어 질 것이

다. 또한 GOP 단위로 분산 저장된 미디어 데이터를 위하여 GOP 단위의 버퍼 관리 기법과 유사한 서비스들을 하나의 그룹으로 묶어 관리하는 그룹 단위 버퍼 관리 기법에 대한 연구도 진행 할 것이다.

5. 참고 문헌

- [1] E. J. O'Neil, P.E. O'Neil, G. Weikum, "The LRU-K Page Replacement Algorithm For Database Disk Buffering", Proc. Of the 1993 ACM SIGMOD Conference, pp 297-306, 1993
- [2] Donghee Lee, Jongmoo Choi, Jong-Hun Kim, Sam H. Noh, Sang Lyul Min, Yook Cho, Chong Sang Kim, "On the Existence of a Spectrum of Policies that Subsumes the Least Recently Used(LRU) and Least Frequently Used(LFU) Policies", Proc. Of the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp 134-143, 1999
- [3] Yannis Smaragdakis, Scott Kaplan, and Paul Wilson, "EELRU: Simple and Effective Adaptive Page Replacement", Proc. OF the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp 122-133, 1999
- [4] 김순철, 조유근, "가변 비트율을 이용하는 주문형 비디오 서버에서의 효율적인 버퍼 관리 기법", 정보과학회논문지 (A) 제25권 제2호, pp. 177-186, 1998
- [5] 이상호, 문양세, 황규영, 조완섭, "주문형 비디오 시스템에서의 동적 버퍼 할당 기법", 정보과학회논문지 : 시스템 및 이론 제28권 제9호, pp. 442-460, 2001
- [6] 서동만, 방철석, 이좌형, 김병길, 정인범, "QoS를 지원하기 위한 리눅스 클러스터 VOD 서버의 성능 분석," 정보과학회 제 30회 춘계학술발표회 논문집, 2003
- [7] C.C.Aggarwal, J.L.Wolf, and P.S.Yu, "On optimal batching policies for video-on-demand storage servers", Proc. of IEEE ICMS'96, pp.253-258, Hiroshima, Japan, June 1996