

재생량에 기반한 연속미디어 데이터 프록시 캐싱 기법

홍 현 옥[†] · 임 은 지^{††} · 박 성 호^{†††} · 정 기 동^{††††}

요 약

연속미디어 데이터를 인터넷을 통해 서비스하기 위해서는 서버의 과부하, 네트워크의 혼잡, 사용자에 대한 응답시간 지연 등의 문제를 해결하여야 한다. 본 논문에서는 연속미디어 프록시 캐싱에서 데이터를 관리하는 PPC(Popularity-based Prefix Caching) 기법을 제안한다. 사용자 접근패턴을 반영하는 PPC 기법은 객체들의 앞부분 데이터를 그 객체의 인기에 비례하여 캐칭함으로써 사용자에 대한 초기지연시간을 감소시키고, 캐쉬 공간을 효율적으로 활용하고자 한다. 연속미디어에 대한 사용자 접근의 특성이 이산미디어와는 상이하다는 점을 고려하여, 연속미디어의 인기도를 더욱 정확하게 나타내기 위해서 각 연속미디어에 대한 사용자의 재생 데이터 양과 접근의 최근성을 반영하는 인기도를 제안한다. 그리고 제안하는 기법의 성능을 평가하기 위하여 인터넷 방송국 VOD 서버의 로그 데이터를 이용한 트레이스 기반 모의 실험을 수행하였다. 그 결과, PPC 기법은 BHR, 초기 지연 시간, 재배치의 측면에서 기존의 LRU, LFU캐싱 기법들에 비하여 성능이 향상되었고, 특히, 초기 지연시간과 재배치 오버헤드의 감소 측면에서 매우 우수한 성능을 보였다.

Playback Quantity-based Proxy Caching Scheme for Continuous Media Data

Hyeon-Ok Hong[†] · Eun-Ji Lim^{††} · Seong-Ho Park^{†††} · Ki-Dong Chung^{††††}

ABSTRACT

In this paper, we propose a proxy caching scheme that stores a portion of a continuous media object or an entire object on the Internet. The proxy stores the initial fraction of a continuous media object and determines the optimal size of the initial fraction of the continuous media object to be cached based on the object popularity. Under the proposed scheme, the initial latency of most clients and the data transferred from a remote server can be reduced and limited cache storage space can be utilized efficiently. Considering the characteristics of continuous media, we also propose the novel popularity for the continuous media objects based on the amount of the data of each object played by the clients. Finally, we have performed trace-driven simulations to evaluate our caching scheme and the popularity for the continuous media objects. Through these simulations, we have verified that our caching scheme, PPC outperforms other well-known caching schemes in terms of BHR, DSR and replacement and that popularity for the continuous media objects based on the amount of the playback data can enhance the performance of caching scheme.

키워드 : PPC, 연속미디어(Continuous Media), 프록시(Proxy), 캐싱(Caching)

1. 서 론

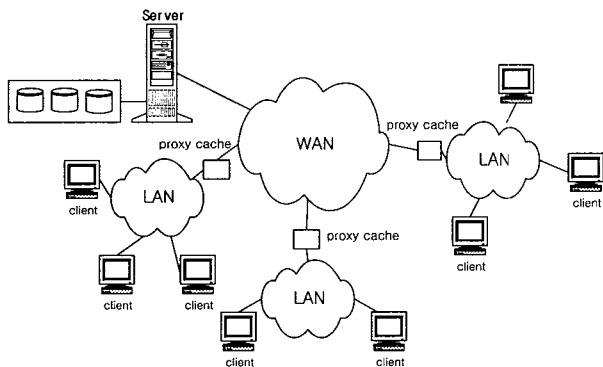
인터넷을 이용하는 사용자들과 인터넷 상에서 서비스 되는 VOD와 같은 대용량 연속미디어 파일들의 증가는 네트워크의 대역폭의 확장 속도에 비해 매우 빠른 속도로 증가하고 있다. 특히 연속미디어 데이터는 텍스트나 이미지와 같은 데이터에 비하여 크기가 크고 높은 전송 대역폭을 요구한다. 그러므로 용량이 제한되어 있는 서버와 네트워크의

부하를 더욱 가중시키게 되어 패킷 손실과 지터 현상 등을 발생시킴으로써 일정 수준 이상의 서비스의 품질을 유지시키기가 매우 어렵다.

이러한 문제점에 대한 해결책으로 프록시 캐쉬 (Proxy Cache)를 들 수 있다[2-6]. 프록시 캐쉬는 네트워크 상에 위치하는 캐쉬로서, 최근에 자주 요구된 데이터를 저장하였다가 사용자의 요구가 발생하였을 경우에 서버에 접근하지 않고 직접 데이터를 전송하여, 같은 데이터가 같은 링크를 통해 반복 전송되는 것을 방지한다[7]. 일반적으로 프록시 캐쉬는 (그림 1)에서처럼 서버와 사용자 사이의 경로상에서 사용자와 인접한 곳에 위치한다. 다시 말해서, 프록시는 LAN상에서 WAN으로 연결되는 라우터에 위치하여 데이터

† 정 회 원 : 동의공업대학 컴퓨터정보계열 교수
 †† 정 회 원 : 한국전자통신연구원 컴퓨터 및 소프트웨어 연구소 연구원
 ††† 준 회 원 : 부산대학교 대학원 전자계산학과
 †††† 종신회원 : 부산대학교 전자계산학과 교수
 논문접수 : 2002년 3월 20일, 심사완료 : 2002년 6월 3일

를 캐싱하여, 서버로의 접근을 줄여줌으로써 서버의 부하, WAN의 트래픽 그리고 사용자의 초기 지연을 감소시킨다. 그러나, 지금까지 연구된 프록시 캐싱 기법은 대부분 텍스트나 이미지와 같은 이산미디어를 대상으로한 기법이었다. 연속미디어 데이터는 일반적으로 스트리밍 모드로 서비스됨으로써, 사용자는 객체의 전체가 아니라 일부분만을 접근할 수 있다. 그러므로 접근의 빈도보다 더 세밀하게 사용자의 접근 패턴을 반영할 수 있는 척도에 관한 연구가 필수적이다.



(그림 1) 인터넷 상의 프록시 캐쉬

본 논문에서는 연속미디어 데이터를 대상으로 한 프록시 캐쉬에 연속미디어 데이터의 특성을 반영한 PPC(Popularity-based Prefix Caching) 기법을 제안한다. 이 기법은 객체들의 앞부분 데이터를 그 객체의 인기에 비례하여 캐싱함으로써 사용자의 초기 지연시간을 감소시키고, 캐쉬 공간을 효율적으로 활용한다. 연속미디어에 대한 접근 패턴을 더욱 정확하게 나타내기 위해서 각 연속미디어에 대한 사용자의 재생 데이터 양과 접근의 최근성을 반영하는 인기도를 정의한다. 그리고 제안하는 기법의 성능을 평가하기 위하여 인터넷 방송국 VOD 서버의 로그 데이터를 이용한 트레이스 기반 모의 실험을 수행하였다. 그 결과, PPC가 BHR, 초기 지연시간, 재배포 오버헤드의 측면에서 기존의 캐싱 기법인 LFU, LRU 등에 비하여 우수한 성능을 보였으며, 특히 재배포 오버헤드가 매우 큰 폭으로 감소하였다.

본 논문의 구성은 다음과 같다. 2장에서 현재까지 진행되어 온 관련연구에 대하여 고찰하고, 3장에서는 제안하는 캐싱 기법과 인기도에 대해 자세히 서술하며, 4장에서는 실험을 통한 성능평가를 보이고, 5장에서는 결론과 향후 연구과제를 기술한다.

2. 관련 연구

2.1 웹 캐싱

웹 캐싱이란 텍스트나 이미지와 같은 데이터를 포함하는 웹 문서를 위한 캐싱을 말한다. 이것은 문서 단위로 캐싱이

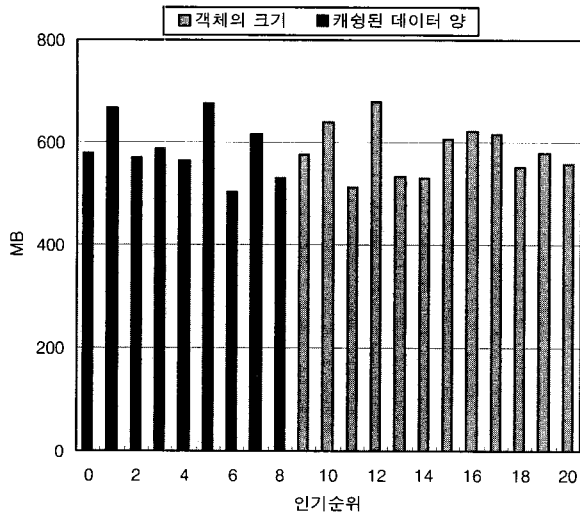
수행되므로 캐싱 단위의 크기가 동일하지 않다는 점에서 동일 크기의 블록이나 페이지 단위로 수행되는 파일시스템 캐싱과 구별된다. 현재의 웹 캐싱에 관한 연구들은 주로 웹 캐쉬의 재배포 알고리즘에 초점을 맞추고 있다. 즉, 웹 문서들에 대한 과거의 접근 정보를 이용하여 가장 저장효율이 낮은 문서를 캐쉬에서 제거하고 저장효율이 높은 문서를 저장하여 웹 캐쉬의 제한된 자원을 효과적으로 사용하도록 한다. 보편적으로 널리 사용되는 접근 정보로는 접근의 빈도(Frequency), 접근의 최근성(Recency), 문서의 크기를 들 수 있고[13-15], 대표적인 알고리즘은 LRU, LFU, SIZE, LRU-SIZE, LRU-MIN, LRFU, GD-SIZE 등을 예로 들 수 있다[5, 12]. 그러나 이러한 기법들은 텍스트나 이미지와 같이 비교적 크기가 작은 전통적인 데이터를 위한 객체 단위의 캐싱 기법이므로, 상대적으로 대용량인 연속미디어 데이터에 적용하기는 부적합하다.

2.2 연속미디어 데이터의 프록시 캐싱

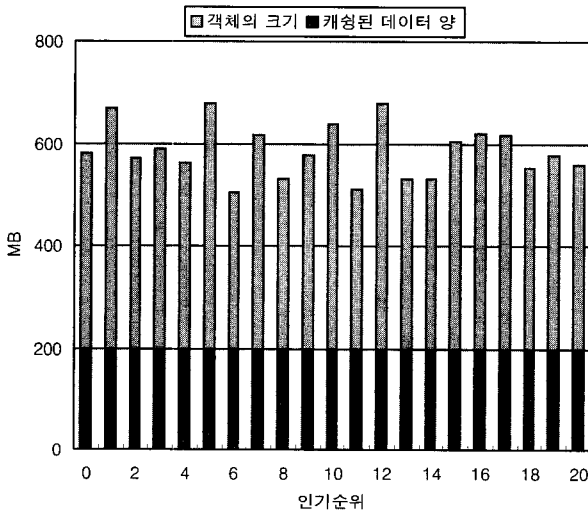
최근 들어 웹 문서를 위한 캐싱 뿐만 아니라 연속미디어 데이터를 위한 프록시 캐싱에 관한 연구가 많이 진행되고 있다[16-21]. 이런 연구들은 전통적인 웹 프록시가 가진 결함을 보완하고 연속미디어를 효율적으로 서비스 하기 위한 특별한 기술들을 제공한다. [16]은 연속미디어 데이터에 대한 접근의 연속성을 고려하는 Interval Caching의 개념을 프록시 캐쉬에 도입하였으며, 프록시의 캐쉬 공간과 대역폭을 프록시의 자원으로 규정하고 각 객체의 자원 요구량에 기반하는 캐싱 기법을 제안하였다. [17]은 멀티미디어 스트림의 앞부분 일정 양만을 캐싱하여 사용자의 초기 지연을 감소시키면서 workahead smoothing을 수행하는 Prefix Caching을 제안하였다. [18]은 Layered-encoded stream을 이용하여 변화하는 네트워크 대역폭에 적응적으로 전송하며 layer 단위로 캐싱할 수 있는 기법을 제안하였고, [19]는 특정 양의 비디오 데이터를 미리 프록시에 프리캐칭하여 서버로부터 WAN을 통해 전송되는 데이터의 양을 일정하게 만들고, video smoothing을 수행하여 이질적인 네트워크 환경에서 양질의 비디오 서비스를 제공할 수 있도록 하는 기법을 제안하였다. [20]은 연속미디어 데이터와 전통적인 미디어 데이터를 모두 효과적으로 처리하기 위한 프록시 서버의 고려 사항들을 나열하고, 여러 가지 다양한 기술들의 동적 결합을 제안하였다. 그러나, 이러한 기존의 연구들은 대부분 프록시 서버를 이용하여 다양한 기법을 제공할 수 있도록 하는 데에 초점을 맞추고 있으며 인터넷 환경에서 적용할 수 있는 연속미디어 데이터를 위한 캐싱 기법에 관한 연구는 미흡한 상태이다.

3. Popularity-based Prefix Caching(PPC)

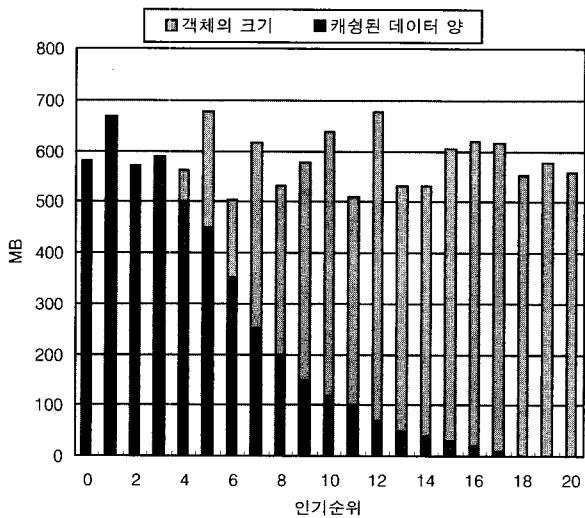
본 장에서는 연속미디어 데이터를 위한 프록시 캐싱 기법



(a) 객체 단위 캐싱



(b) Prefix 캐싱



(c) PPC (Popularity-based Prefix Caching)

(그림 2) 객체의 인기 순위에 따른 캐싱되는 데이터 양

인 Popularity-based Prefix Caching(PPC) 기법에 대해 설명하고, PPC의 성능 향상을 위하여 제안하는 연속미디어의 재생량에 기반한 인기도의 측정 방법을 기술하며, 측정된 인기도를 기반으로 수행되는 캐싱과 재배포에 대하여 설명한다.

3.1 PPC 기법의 설계

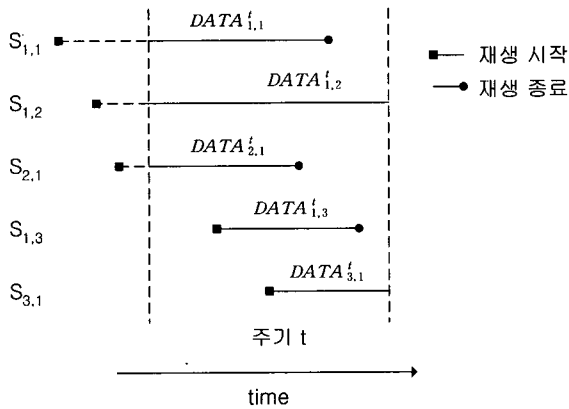
(그림 2)는 연속미디어 데이터에 객체 단위 캐싱과 Prefix 캐싱, 그리고 제안하는 PPC 기법을 각각 적용했을 경우에, 각 객체의 인기 순위에 따라 캐싱되는 데이터의 양을 예를 들어서 나타낸 것이다. (a)는 텍스트나 이미지 데이터에 일반적으로 사용되는 객체 단위 캐싱 기법을 크기가 큰 연속미디어 데이터에 적용한 경우로 인기가 높은 소수의 객체들만 저장할 수 있다. 그러므로 소수의 캐싱된 데이터를 제외한 나머지 대부분의 데이터에 대한 초기 전송 지연 시간이 커지게 된다. (b)는 모든 객체의 앞부분 일정량만을 캐싱하는 Prefix 캐싱의 경우를 나타낸다. 이것은 모든 객체에 대한 서비스 초기 지연시간은 감소하지만, 인기가 낮아서 거의 요구되지 않는 객체에 대해서도 인기가 매우 높은 객체와 동일한 캐싱 공간을 할당함으로써 캐싱 공간의 낭비를 초래한다. (c)는 본 논문의 PPC 기법을 적용한 경우를 나타낸다. 이것은 위의 두 가지 기법의 단점을 보완한 것으로서, 연속미디어 객체의 인기도 분포와 캐싱되는 데이터 양의 분포를 같게 한다. 이렇게 함으로써 가능한 많은 객체에 대한 전송의 초기 지연 시간을 감소시키면서, 서버로부터 전송되어야 하는 데이터 양을 최소화 할 수 있으므로 네트워크 트래픽과 서버의 부하를 감소시킬 수 있다.

3.2 재생량에 기반한 인기도

프록시는 어떤 데이터를 얼마나 캐싱할지를 결정하기 위하여 각 데이터들에 대한 과거의 접근 정보를 이용한다. 텍스트나 이미지와 같은 미디어의 캐싱에서 일반적으로 고려되는 사항은 접근의 빈도와 접근의 최근성, 객체의 크기이다. 그러나 연속미디어 객체는 일반적으로 스트리밍 모드로 전송되므로, 사용자가 한번 접근할 때 객체의 전체를 접근하는 것이 아니라 일부분만을 접근 할 수 있다. 따라서 접근의 빈도를 사용하는 것은 적합하지 않으며 더욱 정확하고 세밀하게 사용자의 접근 정도를 나타낼 수 있는 정보가 필요하다. 예를 들어 두 비디오가 단위 시간동안에 한 번씩 요구 되었는데, 한 비디오는 1분 동안 재생되었고 다른 것은 10분 동안 재생되었다면 두 비디오의 접근 빈도는 같지만 접근 정도는 다르다. 그러므로 후자의 인기도가 전자보다 더 높은 인기도를 가져야 한다. 그러므로 본 논문의 PPC 기법은 연속미디어 데이터의 재생량과 접근의 최근성을 반영하는 인기도를 제안한다.

(그림 3)은 t 번째 주기에서 각 연속미디어 객체들의 구간 재생량을 측정하는 것을 나타낸 그림이다. 구간 재생량이란 주어진 구간에서 사용자에게 의해 재생된 총 데이터의 양을 말한다. $S_{i,j}$ 는 주기 t 에 재생이 진행 중이거나, 주기 t 의 구간에서 재생이 종료된 스트림으로서, 객체 i ($i[1 \dots N]$)를 사용자 j 가 재생한 스트림이다. 실선으로 표시된 부분은 각 $S_{i,j}$ 에서 주기 t 의 구간에 포함되는 부분만을 나타낸 것이고, $DATA_{i,j}^t$ 는 실선 구간동안에 사용자 j 가 객체 i 에 대해 재생한 데이터의 양이다. 주기 t 에서 객체 i 를 재생한 총 사용자의 수를 c_i 라 할 때, 주기 t 에서 객체 i 의 구간 재생량 $DATA_i^t$ 는 식 (1)과 같다.

$$DATA_i^t = \sum_{j=0}^{c_i} DATA_{i,j}^t \quad (1)$$



(그림 3) 주기 t 에서 각 연속미디어 객체의 구간 재생량의 측정

프록시는 계산된 각 객체의 구간 재생량을 이용하여 가중치를 둔 평균 재생량인 D_i^t 를 식 (2)와 같이 계산한다.

$$D_i^t = \alpha DATA_i^t + (1 - \alpha) D_i^{t-1}, \quad 0 < \alpha \leq 1 \quad (2)$$

여기에서 α 는 최근 정보에 대한 가중치 값으로서, NOD (News On Demand) 데이터와 같이 생명주기가 짧은 데이터에 대해서는 α 값을 크게 하고, 상대적으로 생명주기가 긴 데이터에 대해서는 α 값을 작게 한다. 이렇게 최근정보에 대해 가중치를 둬으로써 접근의 최근성을 반영한 인기도를 구할 수 있다.

계산된 D_i^t 값을 이용하여 주기 t 에서 각 객체 i , $i[1 \dots N]$,의 인기도 P_i^t 는 식 (3)과 같이 계산할 수 있다.

$$P_i^t = \frac{D_i^t}{\sum_{k=1}^N D_k^t} \quad (3)$$

P_i^t 값은 $0 \leq P_i^t \leq 1$ 의 범위를 가지며, 이것은 N개의 연속

미디어 객체들 사이에서 객체 i 의 인기 비율을 나타내는 값이 된다.

3.3 캐싱과 재배포의 수행

PPC기법은 각 연속미디어 객체의 인기도 분포와 각 객체의 캐싱되는 데이터 양의 분포를 동일하게 하기 위하여 계산된 인기도 값을 사용하여 각 객체의 캐싱할 최적의 데이터 양을 계산한다. 그러나, 단순히 재생량에 따른 인기도의 비율로 캐싱할 데이터 양을 계산할 경우, 아주 인기가 높은 것은 그 객체의 길이보다 더 많은 공간을 할당 받고, 인기가 낮은 객체들은 공간을 거의 할당받지 못한다. 그러므로 캐싱할 최적의 양을 결정할 때는 인기가 높은 객체부터 계산하며, 나머지 객체들에 대해서는 남은 캐쉬 공간의 크기를 고려하여 계산한다.

객체 i 의 전체 길이를 L_i , i 의 캐싱될 최적의 크기를 \hat{S}_i , 캐쉬의 전체 크기를 S 라 하자. 모든 객체를 D_i^t 값에 따라 내림차순으로 정렬한 후, 다음 수식을 이용하여 수정된 인기도 값을 계산한다. \hat{P}_i^t 는 $i, i+1, \dots, N$ 의 객체 중에서 객체 i 의 인기도 비율을 나타내는 값이다.

$$\hat{P}_i^t = \frac{D_i^t}{\sum_{k=i}^N D_k^t} \quad (4)$$

그리고 각 객체의 최적의 캐싱할 데이터 양은 식 (5)과 같이 계산된다.

$$\hat{s}_1 = \min\{\hat{p}_1 \times S, L_1\}$$

$$\hat{s}_i = \min\{\hat{p}_i \times (S - \sum_{k=1}^{i-1} s_k), L_i\}, \quad 2 \leq i \leq N \quad (5)$$

프록시는 계산 오버헤드를 감소시키기 위하여 각 객체의 인기도와 캐싱할 최적의 데이터 양을 주기적으로 계산해서 테이블로 유지하며, 특정 객체에 대한 사용자의 요구가 발생할 때, 이미 계산되어 있는 값과 실제 캐싱되어 있는 데이터의 양을 비교하여 그 차이 값만큼 데이터를 캐싱 또는 재배포를 수행한다.

캐싱과 재배포의 수행을 위한 알고리즘은 (그림 4)에 나타내었다.

```

cache_data ( object i )
{
    if (  $s_i \geq \hat{s}_i$  ) return ;
     $w = s_i - \hat{s}_i$  ;
    if ( free space in cache <  $w$  ) {
         $E = \{j: s_j > \hat{s}_j, \quad 1 \leq j \leq N\}$  ;
        while ( free space in cache <  $w$  ) {
    
```

```

    j ∈ E 인 임의의 객체 j를 선택;
    j의 캐싱된 부분 중 뒤쪽 (si - ŝi) 데이터를 캐
    쉬에서 삭제;
    E = E(j);
}
object i의 뒷부분 w의 데이터를 추가로 더 캐싱;
return;
}
    
```

(그림 4) 재배치 알고리즘

4. 성능 평가

본 장에서는 트레이스 기반 모의 실험을 통하여 PPC 기법과 연속미디어 데이터의 재생량에 기반한 인기도의 성능을 평가하였다. 실험에 사용한 트레이스는 hananet¹⁾에서 제공하는 인터넷 방송 VOD 서버에서 9일 동안 모아진 로그 데이터를 사용하였다. 트레이스의 특성은 <표 1>과 같다.

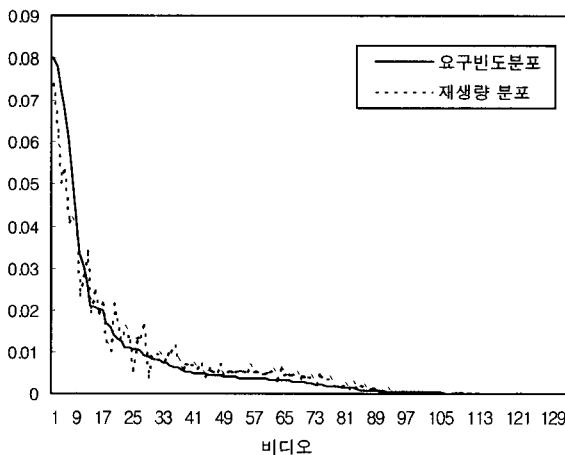
<표 1> 실험에 사용한 트레이스의 특성

Trace	# objects	# accesses	Date	# days
Vod1	108	689526	2000/8/28~2000/9/5	9

평균 비디오의 크기는 110MB (약 50분)이고, 평균 요구 도착률은 1회/초이다. 프록시에서 각 객체의 캐싱할 최적의 데이터 양을 계산하는 주기는 10분으로 하였고, 가중치의 값은 0.2로 하였다.

4.1 사용자 접근 패턴 분석

연속미디어에 대한 사용자 요구의 특성을 분석하기 위하여 요구 빈도 분포와 재생량의 분포가 어떤 관계에 있는지



(그림 5) 요구 분포와 재생량의 분포

를 조사하였다. (그림 5)는 각 비디오 파일을 요구 횟수에 따라 정렬하고 요구 분포와 재생량의 분포를 함께 나타낸 것이다. 비디오 파일은 요구가 되어야 재생되므로 요구 횟수와 재생량은 근본적으로 무관할 수 없다. 그러나, 그림에서 보듯이 두 분포는 약간의 차이를 나타낸다. 따라서 연속미디어의 인기도 측정에 있어서 요구 빈도를 사용하는 경우와 재생량을 사용하는 경우는 캐싱의 성능에도 영향을 미칠 것이란 것을 알 수 있다.

4.2 PPC 기법의 성능 평가

(그림 6)~(그림 8)은 PPC와 기존의 캐싱 기법인 LRU, LFU, LRU-MIN의 성능을 비교 측정된 것이다. 성능의 척도로는 BHR(byte hit rate), DSR(delay savings rate) 그리고 총 재배치 데이터 양으로 하였다. 각각의 그래프에서 캐쉬 크기는 전체 데이터 양에 대한 상대적인 크기로 표시하였다.

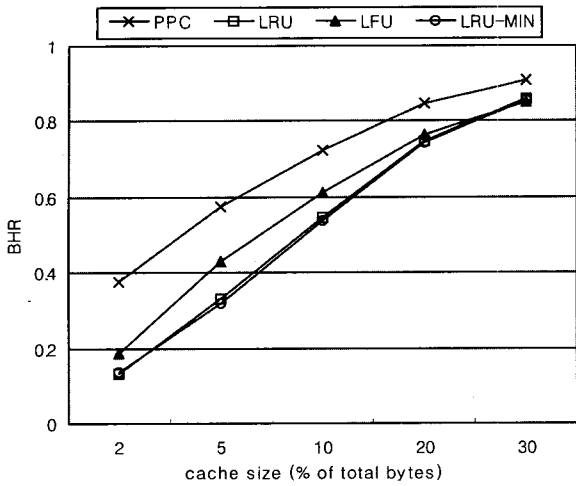
먼저, (그림 6)은 BHR을 측정된 그래프이다. PPC는 크기가 큰 연속미디어 데이터를 대상으로 하고 객체를 부분적으로 캐싱하는 기법이므로 HR(hit rate)을 측정하는 것은 부적합하다고 판단되어 BHR을 측정하였다. PPC의 BHR값이 다른 캐싱 기법들보다 최고 2배정도까지 높게 나타났다. 특히, 캐쉬 크기가 작을 때 차이가 더 크게 나타남을 알 수 있다.

(그림 7)은 DSR을 측정된 것이다. DSR은 캐쉬로부터 데이터를 서비스함으로써 감소시킬 수 있는 초기 지연시간의 비율을 나타내는 값이다. 서버와 프록시 사이의 전송지연은 200ms, 프록시와 사용자 사이의 전송지연은 100ms로 가정하고 실험을 하였다. 그림에서 보듯이, 다른 캐싱 기법들은 객체 단위로 수행되기 때문에 캐쉬 크기에 따라 지연시간의 차이가 크게 나타난다. 그에 반하여, PPC는 가능한 많은 객체들의 앞부분을 캐싱하므로 캐쉬크기에 상관없이 DSR 값이 거의 일정하게 높게 나타난다.

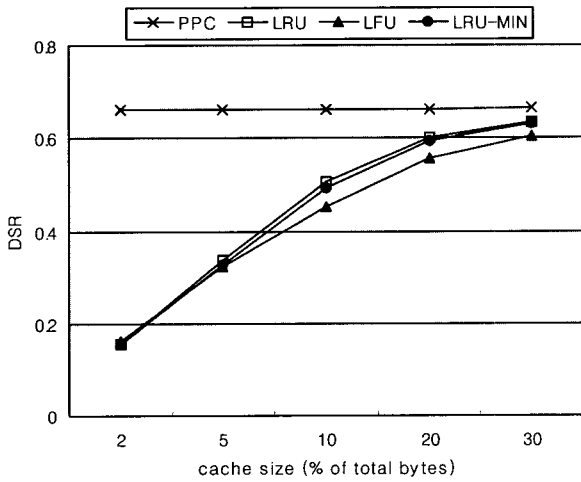
프록시 캐싱은 대역폭이 제한된 디스크 기반의 캐싱이므로 재배치 비용을 감소시키는 것이 매우 중요하다. (그림 8)은 전체 실험 기간동안 발생한 총 재배치 데이터 양을 측정된 것이다. PPC는 기존의 프록시 캐싱 기법들이 객체 전체를 단위로 재배치 되는데 반하여 PPC는 객체의 일부분만 재배치함으로써 기존의 캐싱 기법들과 비교하여 매우 높은 성능을 보인다.

PPC는 전반적으로 다른 기법들보다 우수한 성능을 나타내었으며, 특히 캐쉬 크기가 작은 경우에 더욱 큰 성능의 차이를 보였다. 이것은 PPC가 제한된 캐쉬 공간을 효율적으로 활용할 수 있도록 함을 나타내는 결과이다. 특히, PPC는 지연시간과 재배치 오버헤드의 감소면에서 많은 성능 향상을 보인다.

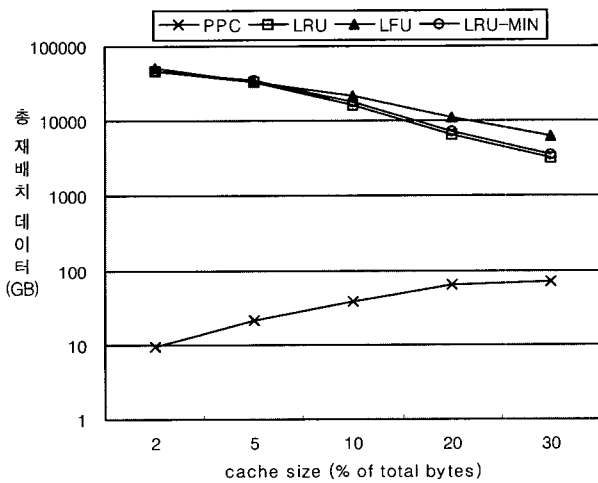
1) ㈜하나로 통신의 인터넷 포털 사이트(http://www.hananet.net)



(그림 6) BHR (byte hit rate)



(그림 7) DSR (delay savings rate)

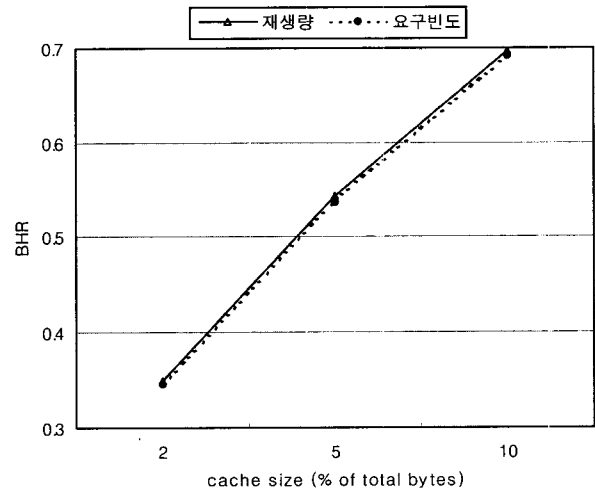


(그림 8) 총 재배치 데이터

4.3 인기도 측정 요소에 따른 성능 평가

본 절에서는 PPC의 캐싱 기법을 그대로 사용하되 데이

터의 인기도 측정에 있어서 재생량과 요구 빈도를 각각 사용하여 캐싱의 성능을 평가하였다. 그 결과로, 재생량을 사용하는 것이 PPC의 성능 향상에 미치는 영향을 측정하였다. 실험의 간단화를 위하여 캐싱의 성능을 나타내는 요소로 BHR만을 측정하였다. (그림 9)를 보면 재생량을 사용한 경우의 캐쉬 BHR값이 요구 빈도를 사용한 경우보다 더 높음을 알 수 있다. 이것은 연속미디어 데이터에 있어서 재생량을 사용하는 것이 데이터의 인기도를 더욱 정확하게 측정할 수 있다는 것을 나타낸다.



(그림 9) 인기도 측정 요소에 따른 캐쉬 BHR

5. 결 론

인터넷 환경에서의 멀티미디어 응용 시스템의 성능을 향상시키기 위하여 프록시 캐시를 사용할 수 있다. 이것은 자주 접근되는 데이터를 캐싱하여 사용자의 지연과 서버의 부하, 네트워크 트래픽을 감소시킬 수 있다.

본 논문에서는 대용량이고 실시간성을 요구하는 연속미디어 데이터를 위한 프록시 캐싱 기법인 PPC(Popularity-based Prefix Caching)를 제안하였다. 그리고 PPC의 성능 향상을 위한 방법으로 연속미디어에 대한 사용자 접근의 특성을 고려하여 데이터의 재생량에 기반한 인기도를 제안하였다. 제안하는 기법의 성능을 평가하기 위하여 인터넷 방송국 VOD 서버의 로그 데이터를 이용한 트레이스 기반 모의 실험을 수행하여 재생량에 기반한 인기도를 이용함으로써 PPC 기법의 성능을 더욱 향상시킬 수 있음을 증명하였다.

향후에는 연속미디어의 특성을 고려하여 새롭게 생성되는 객체에 대한 프리패칭에 대한 연구를 수행함으로써 서버의 부하를 분산시키고 사용자에게 보다 우수한 품질의 연속미디어 데이터를 서비스할 수 있는 기술에 관한 연구가 요구된다.

참 고 문 헌

[1] G. A. Gibson, J. Witter, and J. Wilkes, "Storage and I/O Issues in Large-Scale Computing," ACM Workshop on Strategic Directions in Computing Research, ACM Computing Surveys, 1996.

[2] A. Chankhunthod, P. B. Danzig, C. Neerdaels, M. F. Schwartz, K. J. Worrell, "A Hierarchical Internet Object Cache," In Proc. of 1996 Usenix Technical Conference, January, 1996.

[3] A. Luotonen, K. Altis, "World Wide Web Proxies," In Proc. of the First International Conference on the WWW, May, 1994.

[4] R. Tewari, M. Dahlin, Harrick M. Vin, J. S. Kay, "Beyond Hierarchies : Design Considerations for Distributed Caching on the Internet," In Proc. ICDCS '99, Austin, May, 1999.

[5] J. Wang, "A Survey of Web Caching Schemes for the Internet," Technical Report TR99-1747, Cornell University Department of Computer Science.

[6] J. Gwertzman, M. Seltzer, "The Case for Geographical Push-Caching," In Proc. of the 1995 Workshop on Hot Operating Systems, 1995.

[7] M. A. Goulde, "Network Caching Guide-Optimizing Web Content Delivery," White Paper, Inktomi Corp. http://www.inktom.com/products/network/tech_resources/tech/cachingguide.pdf, March, 1999.

[8] I. Tatarinov, V. Soloviev, A. Rousskov, "Static Caching in Web Servers," In Proc. of the 6th International Conference on Computer Communications and Networks (IC3N 97), Sept., 1997.

[9] A. Bestavros, R. L. Carter, M. E. Crovella, "Application-Level Document Caching in the Internet", In Proc. of Workshop on Services and Distributed and Networked Environments, June, 1995.

[10] A. Dan, D. Dias, R. Mukherjee, D. Sitaram, R. Tewari, "Buffering and Caching in Large-Scale Video Servers," In Proc. of IEEE COMPCON, March, 1995.

[11] B. Ozden, R. Rastogi, A. Silberschatz, "Buffer replacement algorithms for multimedia storage systems," In Proc. of the International Conference on Multimedia Computing and Systems, June, 1996.

[12] P. Cao, S. Irani, "Cost-Aware WWW Proxy Caching Algorithms," In Proc. Of the USENIX Symposium on Internet Technology and Systems, 1997.

[13] S. Williams, M. Abrams, C. Standridge, G. Abdulla, E. Fox, "Removal Policies in Network Caches for World-Wide Web Documents," In Proc. Of the ACM SIGCOMM '96, 1996.

[14] J. T. Robinson, N. V. Devarakonda, "Data Cache Management Using Frequency-based Replacement," in Proc. of ACM SIGMETRICS Conference, 1990.

[15] E. J. O'Neil, P. E. O'Neil, and G. Weikum, "The LRU-k page replacement algorithm for database disk buffering," in Proc. of International Conference on Management of Data, 1993.

[16] R. Tewari, H. M. Vin, A. Dan, D. Sitaram, "Resource-based Caching for Web Servers," In Proc. SPIC/ACM Conference on Multimedia Computing and Networking, January, 1998.

[17] S. Sen, J. Rexford and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," In Proc. IEEE Infocom, March 1999.

[18] Reza Rejaie, Haobo Yu, Mark Handely, Deborah Estrin, "Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet," In Proc. of IEEE Infocom'2000, Tel-Aviv, Israel, March, 2000.

[19] Y. Wang, Z.-L. Zhang, D. Du, and D. Su, "A Network-Conscious Approach to End-to-End Video Delivery over Wide Area Networks Using Proxy Servers," In Proc. IEEE Infocom, April, 1998.

[20] S. Sahu, P. Shenoy and D. Towsley, "Design Considerations for Integrated Proxy Servers," In Proc. IEEE NOSSDAV'99, June, 1998.

[21] M. Reisslein, F. Hartanto, K. W. Ross, "Interactive Video Streaming with Proxy Servers," In Proc. IEEE Infocom, April, 2000.



홍 현 옥

e-mail : hohong@dit.ac.kr
 1983년 부산대학교 영문학과 학사
 1986년 미 텍사스 주립대학 전산학 석사
 1999년 부산대학교 전자계산학과 박사
 과정 수료
 1995년~현재 동의공업대학 컴퓨터정보계
 열 조교수

관심분야 : 병렬처리 및 멀티미디어, 멀티미디어 콘텐츠



임 은 지

e-mail : ejlim@etri.ne.kr
 1999년 부산대학교 전자계산학과 졸업
 (이학사)
 2001년 부산대학교 대학원 전자계산학과
 졸업(이학 석사)
 2001년~현재 한국전자통신연구원 컴퓨터
 및 소프트웨어 연구소 연구원

관심분야 : 컴퓨터 네트워크, 캐쉬 서버, 운영체제, 멀티미디어



박 성 호

e-mail : shpark@melon.cs.pusan.ac.kr

1996년 부산대학교 전자계산학과 졸업
(이학사)

1998년 부산대학교 대학원 전자계산학과
졸업(이학 석사)

2002년 부산대학교 대학원 전자계산학과
졸업(이학 박사)

1998년~현재 부산대학교 전자계산소 강사

관심분야 : VOD 시스템, 멀티미디어 통신, 인터넷 캐싱



정 기 동

e-mail : kdchung@melon.cs.pusan.ac.kr

1973년 서울대학교 졸업(학사)

1975년 서울대학교 대학원 졸업(석사)

1986년 서울대학교 대학원 계산통계학과
졸업(이학박사)

1990년~1991년 MIT, South Carolina 대학
교환 교수

1995년~1997년 부산대학교 전자계산소 소장

1978년~현재 부산대학교 전자계산학과 교수

관심분야 : 병렬처리, 멀티미디어