

사용자 접근 패턴을 반영한 연속미디어의 재생량에 따른 proxy caching 기법

홍현옥⁰ 장신애 정기동
동의공업대학 사무자동화과, 부산대학교 전자계산학과
hhong@dit.ac.kr, {sajang, kdchung}@melon.cs.pusan.ac.kr

Quantity-based Proxy Caching Policy for Continuous Media Streams Using User's Access Pattern

Hyeon-Ok Hong Sin-Ae Jang Ki-Dong Chung
Dept. of Office Automation, DongEui Technical College
Dept. of Computer Science, Pusan National University

요 약

연속미디어 데이터에 적합한 프락시 캐싱 기법을 연구하기 위하여 인터넷 상에서 서비스 되고 있는 비디오 데이터에 대한 사용자 액세스 패턴을 분석하였다. 비디오 특성에 따른 인기도의 변화와 life cycle 을 분석하여 효율적인 프락시 캐싱 기법을 제안한다. 연속미디어 객체의 인기도에 비례하여 전체 또는 앞부분을 캐싱함으로써 클라이언트에 대한 초기지연시간을 감소시키고, 캐쉬 공간을 효율적으로 사용하면서 캐싱의 효과를 극대화 하고자 한다. 비디오의 aging을 고려하여 재생량에 기반한 인기도를 계산할 수 있는 지수분포 모형을 제시하고 주기적으로 인기가 낮은 오브젝트를 인기가 높은 오브젝트로 대체해 나가는 방법을 제시한다. 제안하는 기법의 성능을 평가하기 위하여 인터넷 방송국 VOD 서버의 로그 데이터를 이용한 트레이스 기반 실험을 수행하였다. 그 결과, 제안하는 기법이 BHR, DSR, 재배치의 측면에서 기존의 캐싱 기법에 비하여 전반적으로 우수한 성능을 보여주었다.

1. 서론

인터넷을 이용하는 사용자들과 인터넷 상에서 서비스 되는 VOD와 같은 대용량 연속미디어 파일들이 나날이 급증함에 따라서 시스템 효율이 떨어지고 있다. 특히 연속미디어 데이터는 텍스트나 이미지와 같은 데이터에 비하여 크기가 크고 높은 전송 대역폭을 요구하므로, 용량이 제한되어 있는 서버와 네트워크의 부하를 더욱 가중시킬 뿐만 아니라, 패킷 손실과 지터 현상 등으로 인하여 화질과 음질이 좋지 않은 서비스를 하게 된다.

이러한 문제점에 대한 해결책으로 프락시 캐쉬 (Proxy Cache)를 들 수 있다[2,3,4,5,6]. 프락시 캐쉬는 네트워크 상에 위치하는 캐쉬로서, 최근에 자주 요구된 데이터를 저장하였다가 클라이언트의 요구가 발생하였을 경우에 서버에 접근하지 않고 직접 데이터를 전송하여, 같은 데이터가 같은 링크를 통해 반복 전송되는 것을 방지한다[7].

지금까지 연구된 프락시 캐싱 기법은 텍스트나 이미지와 같은 이산 미디어를 위한 기법이 대부분이고, 이러한 기법들은 오디오나 비디오와 같은 연속미디어의 캐싱에 적용시키기에는 부적합하다. 연속미디어는 이산미디어에 비하여 대용량이고, 전송에서 높은 대역폭을 요구하며, 실시간으로 서비스 된다. 따라서, 연속미디어를 프락시 서버에 저장하기 위해서는 미디어의 특성을 고려한 새로운 캐싱 기법이 필요하다. 특히 연속미디어 응용에 적합하게 부합할 수 있도록 사용자 접근 패턴을 반영한 맞춤형 서비스를 고려한다면 시스템 효율을 획기적으로 향상시킬 수 있다.

본 논문에서는 연속미디어 데이터에 적합한 프락시 캐싱 기법을 연구하기 위하여 웹 상에서의 비디오 데이터에 대한 사용자 액세스 패턴을 분석하였다. 비디오 특성에 따른 인기도의 변화와 life cycle 을 분석하여 효율적인 프락시 캐싱 기법을 제안한다. 연속미디어 객체의 전체 또는 앞부분을 캐싱하는데, 연속미디어의 인기도 분포와 캐싱 되는 데이터 양의 분포를 같게 하는 기법이다. 이것은 객체들의 앞부분 데이터를 캐싱함으로써 클라이언트에 대한 초기지연시간을 감소시키고, 객체의 인기도에 따라 적정한 양만큼의 데이터를 캐싱함으로써 캐쉬 공간을 효율적으로 사용하면서 캐싱의 효과를 극대화 하고자 한다. 그리고 프락시는 저장용량이 제한되어 있으므로 주기적으로 인기가 낮은 오브젝트를 정리하여 인기가 높은 오브젝트로 대체해 나가는 방법을 제시한다.

제안하는 기법의 성능을 평가하기 위하여 인터넷 방송국 VOD 서버의

로그 데이터를 이용한 트레이스 기반 실험을 수행하였다. 그 결과, 제안하는 기법이 BHR, DSR, 재배치의 측면에서 기존의 캐싱 기법에 비하여 전반적으로 우수한 성능을 보여주었다. 논문의 구성은 다음과 같다. 2장에서 관련연구와 연구배경을 살펴보고, 3장에서는 제안하는 캐싱 기법과 사용자 접근 패턴을 반영한 인기도에 대하여 자세히 서술하며, 4장에서 실험을 통한 성능평가를 보이고, 5장에서 결론과 향후 연구과제를 기술한다.

2. 연속미디어 데이터의 프락시 캐싱에 관한 연구

최근 들어 웹 문서를 위한 캐싱 뿐만 아니라 연속미디어 데이터를 위한 프락시 캐싱에 관한 연구가 많이 진행되고 있다[4,5,6]

[4]은 멀티미디어 스트림의 앞부분 일정 양만을 캐싱하여 초기 지연을 줄이고 workahead smoothing을 수행하는 기법을 제안했고, [5]은 interval caching의 개념을 프락시 캐쉬에 도입하여 프락시의 저장 공간과 디스크 대역폭을 고려하는 RBC(resource-based caching)을 제안하였다. [6]은 이질적인 네트워크 환경을 고려한 video staging이라는 기법을 제안하였다.[8]에서는 연속미디어의 특성을 고려하여 클라이언트의 재생량에 기반한 인기도 측정 방법을 제시하고, 그 인기도에 따라 캐싱하는 PPC 기법을 제안하고 있다. 그러나 이 방법은 인기도의 변화가 아주 급격한 비디오 데이터일 경우에는, 새로운 비디오가 만들어져서 서비스가 시작되면 인기도가 급상승하기 때문에 그에 비례하여 재배치 효율이 급격히 떨어지므로 시스템 성능이 저하될 수 있다. 본 논문에서는 이러한 점을 개선하기 위하여 사용자 접근 패턴을 반영한 프락시 캐싱 기법을 제안한다.

3. 사용자 접근 패턴을 반영한 캐싱 정책

인터넷에서 서비스 되는 연속미디어 파일에 대한 사용자 접근패턴을 분석하고 연속미디어 데이터를 효율적으로 서비스 할 수 있는 캐싱 정책을 연구하기 위하여 hananet에서 제공한 인터넷 방송 VOD 서버에서 9일간 모은 로그데이터와 imbc에서 제공한 인터넷 방송 VOD 서버에서 20일간 모은 로그데이터를 사용하였다. 상세한 내용은 표1에 나타나 있다. 우선 연속성이 있는 드라마를 중심으로 사용자 접근 패턴을 분석해 보았다.

표 1 : 사용자 접근 패턴 분석에 사용한 로그데이터

Trace	# objects	# accesses	Date	# days
hananet	67	689526	2000/8/28- 2000/9/5	9
imbc	1,000	2758104	2001/5/12- 2001/5/31	20

3.1 요구분포와 재생량 분포와의 관계

드라마를 중심으로 연속미디어 데이터에서 사용자 요구분포와 재생량 분포가 어떤 관계에 있는지 조사해 보았다. 그림 1은 각 비디오 파일을 요구빈도수에 따라 정렬하고 요구분포와 재생량 분포를 함께 나타낸 것이다. 비디오 파일은 요구가 되어야 재생이 되므로 요구빈도수와 재생량은 근본적으로 무관할 수 없으나 연속미디어 데이터의 인기도 측정에 있어서 요구빈도를 사용하는 경우와 재생량을 사용하는 경우는 분명히 차이가 있고 캐싱 성능에도 영향을 미칠 것이라는 것을 알 수 있다. 비디오 데이터의 경우 항상 처음부터 끝까지 다 보는 것은 아니며, 접속한 해보고 곧 바로 빠져 나가는 사용자도 많은 점을 고려하면 재생량에 따른 인기도 측정이 접속빈도수에 따른 인기도 측정보다 훨씬 타당성이 있다.

대부분의 드라마에서 시간의 흐름에 따라서 변화하는 인기도의 경우 도 재생량으로 접근하였을 때 요구빈도로 접근하였을 때보다 훨씬 유사한 패턴을 보였으므로 본 논문에서는 이런 점에 착안하여 사용자 접근 패턴을 반영한 연속미디어 데이터의 재생량에 기반한 캐싱 기법을 제안한다.

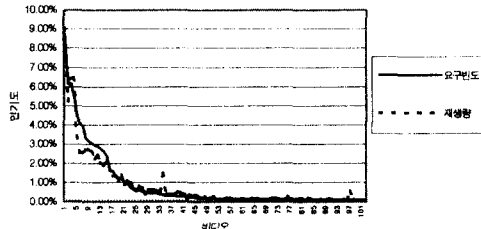


그림 1 : 요구분포와 재생량 분포

3.2 사용자 접근 패턴을 반영한 재생량에 따른 인기도

비디오 데이터에 대한 사용자 접근 패턴의 일반적인 특징을 살펴보기 위하여 생성된 후부터의 lifecycle을 살펴 보았다. 사용자들은 일반적으로 항상 새로 만들어진 비디오를 선호하므로 생명주기에 따른 인기도 변화의 패턴이 뚜렷하게 나타났다. 특히 연속성이 있는 드라마의 경우는 imbc에서 제공하고 있는 모든 드라마에서 일정한 패턴을 보여주고 있다. 즉, 그림 2에서 나타나 있는 것과 같이 가장 재생량이 많은 시기는 생성된 후 1일 동안으로 그 동안은 급격하게 증가하였다가 1일이 지난 후 부터는 지수분포 형태로 거의 일정하게 감소하였다. 따라서 드라마의 aging을 고려하여 재생량에 기반한 인기도를 계산할 수 있는 모형을 찾을 수 있다면 캐싱에도 활용할 수 있을 뿐만 아니라, 연속성이 있는 드라마이기 때문에 다음에 만들어 지는 비디오 오브젝트는 예측된 인기도로 프리패칭이 가능하다. 같은 드라마의 인기도는 횡수에 관계없이 거의 비슷하기 때문에 다음 회분의 비디오가 생성되면 이전 비디오의 인기도를 그대로 적용해서 프리패칭 할 수 있다. 프리패칭은 예상을 비교적 정확하게 할 수 있는 경우에는 아주 효과적인 방법이다. 사용자가 요구하기 전에 인기도가 급상승할 비디오 오브젝트를 캐시에 프리패칭 하게 되면 자주 접근되는 비디오에 대한 BRR가 향상되고, 전반적인 WWW 지연 시간을 단축시킬 수 있기 때문이다.

일반적으로 생존함수나 수명분포를 파악할 수 있는 대표적인 모형이 지수분포이다. 이 논문에서도 드라마 비디오 데이터에 대한 재생량에 따른 인기도 예측 모형을 찾기 위하여 지수 분포를 사용하였다. 찾은 모형은 위험율이 선형다항식인 지수함수로 다음과 같다.

$$f(t) = \alpha \exp\{\beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \beta_4 t^4 + \beta_5 t^5\} \quad \alpha > 0 \quad t \geq 0$$

α : 드라마가 처음 생성된 후 24시간 동안의 접속량(byte)

$\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$: 위험율 (선형다항식의 계수)

t : 비디오가 생성된 후 경과 일 수

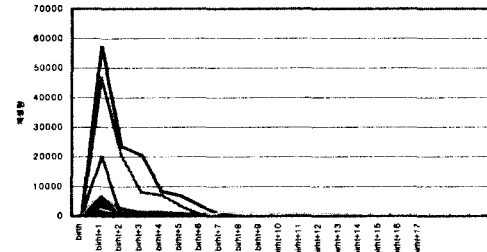


그림 2 : 생성된 후 재생량에 따른 인기도의 변화

모형을 추정하기 위하여 위험율인 다항식은 비선형회귀모형을 설정한 후 SAS(Statistical Analysis System)의 PROC NLIN(NonLinear regression)을 이용하여 최소제곱법으로 적합 시킨 추정결과는 표2와 같다.

표 2 : 추정 결과

Parameter	Estimate
α	4753.238265
β_1	-1.710952
β_2	1.065243
β_3	-0.351231
β_4	0.049573
β_5	-0.002487

추정한 지수함수가 데이터를 잘 적합하고 있는가를 다음 표3의 결과를 가지고 chi-square로 적합도 검정을 해보았다. k=7일 때 p-value는 0.001262 이었으며 k=8일 때 p-value는 0.002591 이었다. 따라서 p-value가 0.001보다 크기 때문에 유의수준 $\alpha = 0.1\%$ 에서 추정된 지수함수가 데이터를 잘 적합 시키고 있음을 알 수 있었다.

표 3 : 데이터의 인기 순위에 대한 캐싱되는 데이터 양

t	byte	추정된byte	chi
1	4754	4753.24	0.00
2	1827	1838.54	0.07
3	1390	1352.06	1.06
4	874	942.70	5.01
5	641	562.21	11.04
6	307	347.33	4.68
7	271	268.12	0.03
8	234	230.52	0.05

3.3 인기도에 따른 캐싱과 재배치 정책

연속미디어 데이터는 크기가 크기 때문에 객체 단위로 캐싱할 경우에 인기가 높은 소수의 객체들만 저장할 수 있다. 따라서 저장 공간을 비효율적으로 사용하게 되고 소수의 캐싱된 데이터를 제외한 나머지 데이터에 대한 전송의 초기 지연 시간이 크게 된다. 각 객체의 앞부분 일정량의 데이터만을 캐싱하는 prefix 캐싱의 경우에는 모든 데이터에 대한 전송의 초기 지연 시간은 감소되지만, 인기가 낮아서 거의 요구되지 않는 비디오에 대해서도 인기가 매우 높은 비디오와 동일한 저장공간을 할당함으로써 저장공간의 낭비를 초래한다. 이 논문에서 제안하고 있는 기법은 위 두 가지 기법의 단점을 보완한 것으로서, 비디오의 인기도 분포와 캐싱되는 데이터 양의 분포를 같게 한다. 이렇게 함으로써 가능한 많은 객체에 대한 전송의 초기 지연 시간을 감소시키면서, 서버로부터 전송되는 데이터 양을 최소화 할

수 있다.

각 연속미디어 객체의 인기도 분포와 객체의 캐싱되는 데이터 양의 분포를 같게 하기 위하여 계산된 인기도 값을 이용하여 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 많이 할당 받고, 인기가 낮은 객체들은 공간을 거의 할당 받지 못한다. 따라서 최적의 양을 결정할 때는 인기가 높은 객체부터 계산한다. 그리고 나머지 객체들에 대해서는 남은 객체들 사이에서의 인기도 비율과 남은 캐시 공간의 크기를 이용하여 최적의 양을 계산한다.

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

$$\hat{p}'_i = \frac{D'_i}{\sum_{k=i}^N D'_k}$$

그리고 각 객체의 최적의 캐싱할 데이터 양은 다음과 같이 계산된다.

$$\hat{s}_i = \min\{\hat{p}'_i \times (S - \sum_{k=0}^{i-1} \hat{s}_k), L_i\}$$

계산되어진 값과 실제 캐싱되어 있는 데이터 양을 비교하여 그 차이 값만큼 캐싱 또는 재배치를 수행한다.[8]

4. 실험

실험환경은 표 4와 같고, 모든 클라이언트는 재생도중에 임의의 시점에서 재생을 중단할 수 있다.

표 4 : 실험환경

Media object size	16MB ~ 47MB
Number of media object	39
Simulation duration	202385개의 사용자 요구를 처리하는 시간
프락시 서버와 클라이언트 사이의 전송 delay	100ms
서버과 프락시 서버 사이의 전송 delay	200ms

그림 4는 제안하는 기법과 다른 캐싱 알고리즘의 성능을 비교 측정 한 것이다. 프락시의 성능을 측정하기 위한 패러미터로 Hit Rate와 Byte Hit Rate를 일반적으로 사용한다. 제안하는 기법은 연속미디어의 객체를 부분적으로 캐싱하는 기법이므로 BHR로 성능을 비교한 결과 그림 4의 (a)에서 보듯이, 특히 캐시 크기가 적을 경우에는 다른 기법들에 비하여 월등하게 우수하였다. 또 다른 성능 척도로는 DSR(delay savings ratio)와 재배치 byte수를 사용하였다. DSR은 캐시로부터 서비스 해 줌으로써 줄일 수 있는 지연 시간의 비율을 나타내는 값이다. 그림 4의 (b)를 보면 DSR면에서도 제안하는 기법과 PPC가 다른 알고리즘보다 월등히 우수함을 알 수 있다. 특히, 캐시 크기가 적을 때는 기존의 기법들 보다 2배 이상 성능이 향상되었다. 이것은 제안하는 기법과 PPC를 제외한 나머지 알고리즘들이 객체 단위의 캐싱을 수행하므로 캐시크기가 적을 때는 인기가 높은 몇 개의 객체만 저장할 수 있어서 평균 지연시간이 높은 반면, 제안하는 기법과 PPC는 대부분의 객체의 앞부분을 약간씩 저장하므로 지연시간이 현저히 감소한다. (c)는 전체 실험 기간동안 발생한 재배치 횟수를 측정 한 것이다. 캐시 크기가 5%일 때 재배치 byte수의 차이는 대단히 많이 나지만 캐시 크기가 15% 이상 일 때 재배치 byte수의 차이는 그래프 상에서 비슷하게 보일 지라도 극히 소량이라는 것은 세로축 값을 보면 알 수 있다. 뿐만 아니라 제안하는 기법은 사용자 접근 패턴을 반영한 모형을 의한 인기도이기 때문에 캐싱 구간을 줄이게 되며, 즉 시간 단위로 캐싱을 수행하더라도 재배치 byte 수는 크게 증가하지

않는다. 따라서 많은 오브젝트를 대상으로 한다면 제안하는 기법의 성능이 더욱 우수한 것으로 나타날 것이다. 앞으로 많은 인터넷 방송국을 대상으로 수많은 비디오 오브젝트를 효율적으로 서비스 할 수 있도록 하려면 가장 적절한 t값을 찾아내는 것이 중요한 연구 과제이다.

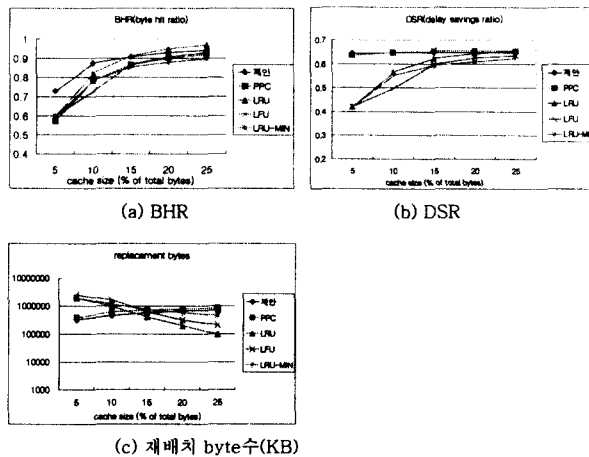


그림 3 : 다른 캐싱 알고리즘과의 비교

5. 결론 및 향후 연구 과제

연속미디어의 특성을 고려한 캐싱 기법을 제안하고 실험을 통하여 성능을 평가하였다. 그 결과 사용자 접근 패턴을 반영한 연속미디어의 재생량에 따른 캐싱 방법이 인기도를 민감하게 반영하는 미디어 VOD 데이터를 서비스 하는데 있어서는 다른 시스템 보다 성능이 좋을 수 있었다. 비디오의 aging을 고려하여 재생량에 기반한 지수분포 모형에 의한 인기도를 사용함으로써 데이터의 캐시 참조율을 최대화 하고 서버로의 접근율을 감소시켜 시스템 효율을 향상시킬 수 있다. 향후에는 인기도에 의한 프리패칭이 네트워크 상황에 미치는 영향에 대한 연구가 필요하다.

6. 참고문헌

- [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] J. Wang, "A Survey of Web Caching Schemes for the Internet", Technical Report TR99-1747, Cornell University Department of Computer Science
- [4] S. Sen, J. Rexford and D. Towsley, "Proxy Prefix Caching for Multimedia Streams," In Proc. IEEE Infocom, March 1999.
- [5] R. Tewari, H. M. Vin, A. Dan, D. Sitarum, "Resource-based Caching for Web servers", In Proc. SPIC/ACM Conference on Multimedia Computing and Networking, January 1998.
- [6] 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.
- [7] 박성호, 임은지, 최태욱, 정기동, "인터넷상에서 NOD 서비스를 위한 연속미디어 전송 및 푸쉬-캐싱 기법", 한국정보처리학회 논문지 제7권 제6호.
- [8] 임은지, 최태욱, 박성호, 정기동, "연속미디어 스트림의 재생량에 기반한 Proxy Caching 기법", 한국정보과학회 2000추계학술발표논문집 제 27권 2호, p588-590