

# 멀티미디어 객체 처리 성능 향상을 위한 웹 캐싱 기법의 설계

A Design of Web Caching for Improvement of Processing  
Performance on a Multimedia Object

나윤지\*, 최승권\*\*, 신승수\*\*\*, 조용환\*  
충북대학교 컴퓨터공학과\*, 애니솔루션\*\*,  
시그마정보통신\*\*\*

Na Yun-Ji\*, Choi Seung-Kwon\*\*,  
Shin Seung-Soo\*\*\*, Cho Yong-Hwan\*  
Dept. of Commuter Engineering,  
Chungbuk National University\*  
ANY Solution Co., Ltd\*\*,  
Sigma Information Technology Co, Ltd\*\*\*

## 요약

본 연구에서는 멀티미디어 객체 처리의 성능 향상을 위한 웹 캐싱 기법을 제안한다. 제안 기법은 멀티미디어 객체의 이질성을 기반으로 하고 있으며, 멀티미디어 객체의 캐싱 효율을 높여 캐싱 시스템의 성능을 높이고자 하는 것이다. 또한 실험에서 제안 시스템의 성능 분석을 통해 시스템의 성능 향상을 검증하였다.

## Abstract

In this paper, We offer a new web caching technique for a performance improvement of a multimedia object process. The proposed method is based on a heterogeneity of Web object. And it increases performance of a caching system as increasing the caching efficiency of a multimedia object. In the experiment, We verified performance improvement through performance analysis of a proposed system.

## 1. 서론

웹 캐싱 기법의 성능과 관련된 이슈는 한정된 저장 공간의 효과적인 운영을 통한 웹 시스템 사용자의 응답속도 향상이라 할 수 있다[1,2,3,4]. 기본적으로 웹 캐시의 저장 공간은 한정되어 있으므로 사용 빈도가 높은 객체를 캐시의 저장영역에 저장하여야 사용자 응답속도를 향상시킬 수 있다. 하지만 한 가지 더 고려해야 할 사항은 객체가 웹 트래픽에 미치는 영향이다. 이런 관점에서 보면 멀티미디어 객체에 대한 관리 중요성이 부각된다. 일반적으로 다른 객체들에 비

해 멀티미디어 객체의 크기가 크다. 이에 따라 멀티미디어 객체의 사용 증가는 웹 트래픽의 증가를 가져오게 된다. 따라서 멀티미디어 객체 중심의 웹 캐싱 기법이 필요하다. 지금까지 많은 웹서버가 텍스트와 이미지 객체를 주로 서비스하였다. 하지만, 웹은 점차 멀티미디어 환경의 서비스로 발전하고 있으며 향후 웹 서비스의 50% 이상이 멀티미디어 서비스가 될 것이라 예측하고 있다[5].

기존의 연구 중에서 객체를 파일 타입별로 나누어 별도의 캐시 공간에 저장하고 관리하는 기법이 제안되었으나, 이 경우 캐시 관리자의 구성과 관리에 대

한 부담이 증가하게 된다. 일반적으로 사용빈도가 비슷하고 크기가 비슷한 객체를 각각의 구분된 영역에 저장함으로써 한정된 공간의 캐시 영역의 할당을 복잡하게 만드는 단점이 발생하게 된다.

본 연구에서는 멀티미디어 객체 처리 성능 향상을 위한 웹 캐싱 기법을 제안한다. 제안기법은 멀티미디어 객체를 위한 캐시의 저장 공간을 할당하고, 이것의 동적인 관리를 통하여 멀티미디어 객체의 처리 성능을 향상시키는 기법이다.

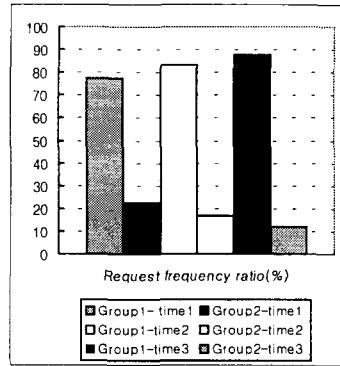
제안 기법에 대한 실험 결과 멀티미디어 객체의 참조가 많은 웹 시스템에 대한 적응률 향상으로 사용자 응답속도의 성능 개선이 가능함을 확인할 수 있었다. 또한 실험 결과를 통해 제안 시스템이 멀티미디어 객체의 처리 속도의 개선을 확인할 수 있었다.

## 2. 멀티미디어 객체 특성 분석

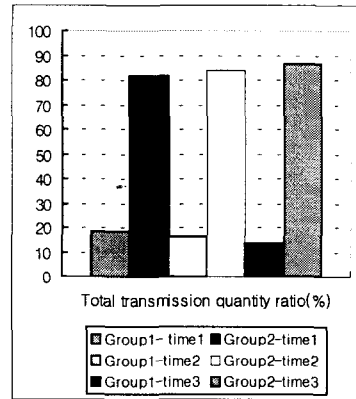
콘텐츠 제작 기술과 네트워크의 발전은 웹 상에서 더욱 다양한 콘텐츠의 서비스를 가능하게 하고 있다. 하지만, 이러한 발전은 한 편으로는 사용자의 멀티미디어 콘텐츠 요구를 증가시켜 웹 트래픽의 폭주를 가져오고 있다. 웹에서 다루고 있는 콘텐츠는 객체 단위로 이루어지고 있으며, 멀티미디어 콘텐츠의 경우 멀티미디어 객체 단위로 다루어진다. 표 1은 객체의 그룹을 특성을 중심으로 나눈 것이다.

[표 1] 객체 그룹 특성

구분	참조 특성	참조 빈도수 비율	총전송량 비율
그룹1	- 텍스트 위주의 HTML 문서 - 뉴스그룹, 전문가 그룹, 텍스트 위주의 커뮤니티·자료 공유 웹사이트의 객체	높다	낮다
그룹2	- 아이콘이나 웹 화면에 사용되는 GIF, JP(E)G, AVI 등의 비교적 작은 크기의 그래픽 및 멀티미디어 객체 - 웹 화면을 통해 전송 받는 크기가 큰 그래픽 객체 및 MPEG 등의 동영상 객체 - 젊은 연령의 사용자 주된 대상으로 하는 웹사이트의 객체	낮다	높다



▶▶ 그림 2.1 객체 그룹별 참조빈도비율



▶▶ 그림 2.2 객체 그룹별 총전송량 비율

그림 1은 세 개의 시간 구간에 대해 2개의 객체 그룹별 총 참조 빈도수 비율을 나타낸 것이고, 그림 2는 그림 1에 대한 객체 그룹별 총 전송량 비율을 나타낸 것이다.

그림2는 아이콘이나 웹 화면에 사용되는 GIF, JP(E)G, AVI 등의 비교적 작은 크기의 그래픽 및 멀티미디어 객체와 웹 화면을 통해 전송 받는 크기가 큰 그래픽 객체, MPEG 등의 동영상 객체와 같은 크기가 큰 멀티미디어 객체 그룹이다. 전체 웹 객체 참조 빈도수 면에서는 멀티미디어 객체가 차지하는 비율이 그룹1의 객체에 비해 작다. 하지만 총전송량 면에서는 80% 이상의 높은 비율을 차지한다. 그룹1은 텍스트 위주의 HTML 문서와 뉴스그룹, 전문가 그

를, 텍스트 위주의 커뮤니티·자료 공유 웹사이트의 객체와 같이 비교적 같은 웹 객체 위주의 객체 그룹이다. 이 객체는 그룹1의 객체에 비해 그 편차가 작고 전체 객체 참조빈도수 면에서 80% 이상의 높은 참조 특성을 나타낸다. 하지만 그룹 2의 객체에 비해 총전송량에서 차지하는 비율은 작다.

멀티미디어 객체의 경우 객체의 사용 빈도가 낮은 경우라도 객체의 크기에 의해 웹 트래픽의 급격한 증가와 작은 크기의 객체를 캐시 밖으로 한꺼번에 교체시킨다. 캐시 적중률 관점에서 보면 작은 크기의 객체를 캐시에 저장하는 것이 캐시의 적중률을 향상시킨다. 따라서 멀티미디어 객체의 캐시부재는 캐시의 효율을 떨어뜨리는 주된 원인이 된다.

웹 캐싱의 효율성의 판단할 수 있게 하는 또 하나의 관점은 네트워크 트래픽 관점이다. 웹 캐싱은 전통적인 캐싱 기법과는 다르게 웹이라는 환경을 기반으로 하고 있어, 네트워크의 트래픽은 사용자 응답 시간에 직접적인 영향을 미치게 된다. 즉, 트래픽 관점의 웹 캐싱 효율은 사용자 응답속도의 개선이라 할 수 있다. 일반적으로 그룹2의 멀티미디어 객체가 그룹1의 객체에 비해 네트워크 트래픽을 증가시킨다.

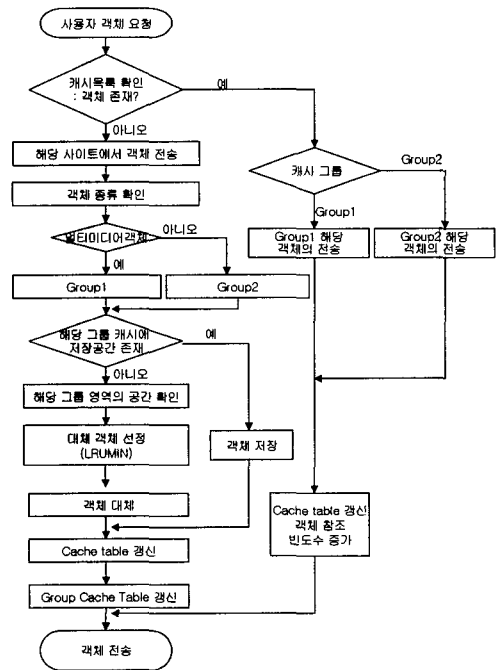
따라서 멀티미디어 웹 객체의 크기 편차는 매우 크다. 따라서 멀티미디어 객체를 효율적으로 관리할 수 있는 웹 캐싱 알고리즘이 필요하다. 또한 웹 객체는 사용자 개인의 참조 특성에 따라 가변적인 객체 참조 특성을 가진다. 이러한 웹 객체의 참조 특성은 객체의 특성을 반영한 웹 캐싱 알고리즘을 필요로 한다.

### 3. 제안 기법

웹 캐싱 기법의 성능과 관련된 이슈는 한정된 저장 공간의 효과적인 운영을 통한 웹 시스템 사용자의 응답속도 향상이라 할 수 있다. 또한 3장에서 살펴본 듯이 멀티미디어 객체가 웹에서 차지하는 비율이 점차 증가하고 있고, 웹 트래픽에 미치는 영향이 크기 때문에 본 연구에서는 멀티미디어 객체의 효과적인

관리를 통한 트래픽 감소로 인한 사용자 응답속도의 개선이 주안점이다.

그림 3은 캐시 대체 알고리즘을 나타낸 것이다. 클라이언트의 객체 요청이 들어오면 캐시 관리자는 해당 객체그룹의 목록에서 존재 여부를 확인한다. 이때 캐시의 저장 영역에 해당 객체가 있을 경우 사용자가 요구한 객체를 사용자의 IP로 전송하게 된다.



▶▶ 그림 2.3 캐시 대체 알고리즘

이때 해당 객체는 캐시 테이블에서 참조빈도수가 1 증가한다. 캐시 목록에서 요청된 객체를 찾을 수 없을 경우 해당 URL의 인터넷 서버에 객체 서비스를 요청하여 전송 받는다. 전송 받은 객체는 객체에 따라 그룹1 또는 그룹2로 분류되고 캐시 관리자는 해당 그룹의 캐시 영역에 이 객체를 저장할 공간이 있는지를 확인한다. 저장할 공간이 있을 경우 객체를 해당 캐시 영역에 저장하고 캐시 목록을 갱신하고, 없을 경우 해당 그룹의 캐시 영역에 대해 캐시 대체 기법

을 이용하여 객체를 대체한 후 캐시 목록을 갱신한다. 이때 사용할 수 있는 웹 캐시 대체 기법은 다양하지만 여기에서는 LRUMIN을 사용한다. LRUMIN 기법은 LRU 기법의 변형이며 SIZE 기법과 같이 LRUMIN은 크기가 큰 객체에 의해 크기가 작은 많은 객체가 저장 영역에서 제거되는 경우를 줄일 수 있다고 알려져 있다. LRUMIN의 동작은 다음과 같다. 먼저, 이 기법은 새롭게 들어오는 객체의 크기를 S로 설정한다. 다음으로, S보다 큰 여유 공간이 없으면 S/2 이상의 큰 객체들 중에서 LRU로 제거한다. 이때, S/2 이상의 큰 객체가 없으면 S/4 이상의 큰 객체들 중에서 LRU로 제거한다. LRUMIN은 이러한 방식으로 새로운 객체의 저장을 위한 여유 공간이 생길 때까지 이 과정을 계속해서 반복한다.

#### 4. 실험 및 분석

표 2는 이 실험에 사용된 그룹1과 그룹2의 참조에 의한 총전송량을 기준으로 7.7:2.3, 8.3:1.7, 8.8:1.2의 세 가지 모델에 대해 이루어졌다.

[표 2] 실험모델 특성

구분	그룹별 비율	
	그룹1 참조빈도 비율 (총전송량비율)	그룹2 참조빈도 비율 (총전송량비율)
실험1	7.7(1.8)	2.3(8.2)
실험2	8.3(1.6)	1.7(8.4)
실험2	8.8(1.4)	1.2(8.6)

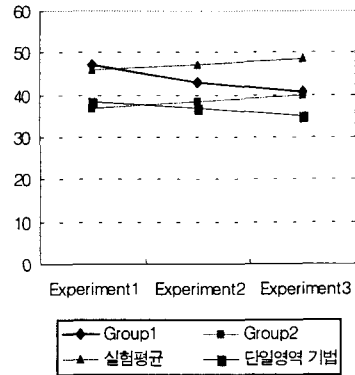


그림 2.4 객체 크기 반영한 적중률

그림 4는 객체의 크기를 고려한 적중률을 나타낸 것으로 각 객체의 크기 평균값을 각 객체의 적중률에 곱한 값의 평균을 낸 것이다. 그룹1에 대한 실험결과, 그룹1의 비율이 가장 높은 실험1의 경우 앞의 객체적중률 실험보다 다소 높은 적중률을 보인다. 하지만 그룹1 객체가 감소할수록 앞의 객체적중률보다 적중률이 낮아짐을 알 수 있다. 그룹2의 경우는 이 실험과는 다소 다른 점을 찾을 수 있다. 전체적으로는 앞의 실험에 비해 적중률이 높아졌으며, 특히 그룹2 객체의 비중이 높을수록 적중률이 더욱 높아졌음을 알 수 있다. 결과적으로 객체의 크기를 반영할 경우 제안 알고리즘 전체의 객체적중률이 높아졌다. 하지만 단일 영역 기법의 경우 제안 기법보다 그룹2 객체의 적중률이 낮기 때문에 전체적으로 적중률이 감소했다.

객체의 크기는 웹 기반 시스템의 트래픽에 영향을 미치는 요인이 된다. 크기가 큰 객체에 대한 캐싱 능력 향상은 웹 트래픽을 감소시켜 사용자 응답 시간을 감소시킬 수 있기 때문에 이 결과는 웹 캐싱 시스템에서 앞의 객체 적중률에 비해 더욱 큰 의미를 갖는다. 또한 제안 기법이 적중률 향상을 통해 멀티미디어 객체의 처리 성능을 개선할 수 있음을 기대할 수 있다.

## 5. 결 론

제안 기법에 대한 실험 결과 멀티미디어 객체 비중이 높아질수록 제안 기법의 적중률 향상을 확인 할 수 있었고, 이에 따라 제안 알고리즘이 멀티미디어 객체 처리 성능 향상으로 사용자 응답속도를 개선함을 기대할 수 있었다.

### ■ 참고문헌 ■

- [1] G. Barish, K. Obraczka, World Wide Web Caching: Trends and Techniques. IEEE Communications, Internet Technology Series, May 2000.
- [2] H. Bahn, S. Noh, S. L. Min, and K. Koh, "Efficient Replacement of Nonuniform Objects in Web Caches," IEEE Computer, Vol.35, No.6, pp.65-73, June 2002.
- [3] N. Niclausse, Z. Liu, P. Nain, "A New and Efficient Caching Policy for the World Wide Web," Proc. Workshop on Internet Server Performance(WISP 98), pp.94-107, 1998.
- [4] C. Aggarwal, J. Wolf and P. Yu, "Caching on the World Wide Web," IEEE Trans. Knowledge and Data Engineering, vol 11, no.1, pp.94-107, 1999.
- [5] S. Sahu, P. Shenoy, D. Towsley, "Design Considerations for Integrated Proxy Servers," Proc. of IEEE NOSSDAV'99, pp.247-250, June, 1999.