

액세스 패턴을 이용한 프락시 캐싱

임재현*, 이준연**

*국립공주문화대학 컴퓨터정보과

**동명정보대학교 멀티미디어공학과

e-mail:nolboo@munhwa.kongju-c.ac.kr

Proxy Caching using Access Patterns

Jae-Hyun Lim*, Jun-Yeon Lee**

*Dept of Computer Information, Kongju National Culture College

**Dept of Multimedia Engineering,

Tongmyong University of Information Technology

요약

웹 프락시 통신량 특성은 캐싱, 용량 설계 및 모의 실험 연구에 영향을 미치는 매개 변수를 인식하는데 도움을 준다. 본 논문에서는 프락시 서버에 나타나는 통신량을 추적하여 사용자의 액세스 패턴을 분석하고 그에 따른 캐싱 모델을 수립하였다. 적중률과 가중치 적중률을 평가한 결과 캐싱의 유용성을 개선하였다.

1. 서론

인터넷에서 웹 사용의 대중성 증가는 수많은 요구를 발생하고 있다. 웹이 직면하고 있는 규모의 문제는 세 가지로 설명할 수 있다[2]. 첫째, 증가하는 사용자의 요구를 수용하기 위해 웹 서버의 규모를 증가시키는 것이다. 둘째, 인터넷 그 자체가 계속해서 용량을 증가시키고 새로운 망(network) 기술을 수용하는 것이다. 셋째, 클라이언트에 초점을 맞춰 웹 브라우저와 프락시 서버를 계층적으로 확장하는 것이다. 비용 면에서 인터넷의 규모를 확장하는 가장 좋은 정책은 클라이언트 가까이에서 데이터를 캐시해서, 액세스 지연을 개선시키고 망과 서버의 부하를 줄이는 것이다. 즉 캐싱은 서버에 요청되는 데이터의 양과 망을 통해 전달되는 데이터의 양을 감소시켜 클라이언트와 서버의 성능을 개선시킨다.

지금까지 대부분의 많은 웹 서버가 텍스트와 이미지 객체를 주로 서비스하였다. 하지만, 웹은 텍스트 기반의 정보 시스템에서 멀티미디어 정보 시스템으로 급격히 발전하고 있다. 최근의 연구에서 연속 미디어인 오디오와 비디오 객체가 웹 상에서 1998년 9개월 동안 3배 가까이 증가하였다고 한다. 현재 웹 서버 상에 저장되어있는 연속 미디어의 양이 상대적

으로 적지만, 2003년에는 50%이상으로 증가할 것이라고 한다[1]. 뿐만 아니라 이들 데이터를 액세스하는 클라이언트는 손에 들고 다니는 PDA부터 워크스테이션에 이르기까지 다양해지며, 이들 각각은 성능과 서비스 요구가 다르다. 현재의 웹 프락시는 텍스트와 이미지 데이터를 위해 설계되었기 때문에 연속 미디어를 캐시 하거나, 사용자의 다양한 서비스 요구를 효과적으로 처리하지 못한다.

이와 같은 환경의 변화를 수용하기 위해서는 프락시 서버에 대한 사용자의 액세스 패턴을 정확히 분석할 필요가 있으며 새로운 캐시 전략이 필요하다. 본 논문에서는 프락시 캐시의 성능에 영향을 미치는 액세스 패턴 특성을 파악하기 위해 서로 다른 작업 부하로 액세스하는 2가지의 프락시 서버를 대상으로 통계치를 수집하였다. 이들 수집된 데이터를 분석한 결과 동일한 크기의 객체를 캐시 하는 CPU 캐시 또는 가상기억장치와는 달리, 프락시 캐시 안의 문서는 몇 바이트의 텍스트 화일에서 몇 메가 바이트의 비디오에 이르기까지 다양한 크기를 갖는다[4]. 화일의 타입은 그 크기와 밀접한 관계가 있으며 비디오, 오디오 타입의 크기는 텍스트, 이미지 보다 상당히 크다. 예를 들어 1GB의 프락시 서버가 수만 개의

화일을 포함한다고 할 때, 텍스트나 이미지 화일만을 저장한다면 수백만 개의 화일을 저장할 수 있어 서버에 도착하는 요청 수는 감소할 것이다. 하지만 비디오, 오디오 타입을 많이 포함하는 경우 수많은 화일을 교체하며 프락시 서버의 디스크 공간을 많이 차지하지만, 이들을 캐시 하는 것은 망 통신량 감소에 많은 도움이 될 것이다. 따라서 본 논문에서는 서버에 도착하는 요청 수와 망 통신량을 감소시키기 위해 기존의 캐싱 전략과는 다르게 화일 타입에 따라 캐싱 하는 전략을 제안한다. 본 논문에서 제안한 방법의 성능을 평가하기 위해 모든 화일을 포함할 수 있는 무한 디스크 크기를 갖고 모의실험을 한다. 이것은 실제 프락시 캐시가 달성할 수 있는 적중률의 상한 값을 갖는다. 화일의 교체(replacement)가 필요 없는 디스크 크기를 조사한 후, 디스크 크기를 제한한 상태에서 LRU, LFU, SIZE 3가지 교체 정책을 이용하여 기존의 방법과 본 연구에서 제안한 타입별 캐싱의 성능을 비교한다. LRU(Least Recently Used)는 최근에 사용되지 않은 화일을 먼저 교체하는 정책이다. 시간적 지역성(temporary locality)에 기반한 정책으로 최근에 사용된 화일이 가까운 시간 내에 다시 사용될 가능성이 높다는 점을 이용한다. 인터넷 캐시에서는 다른 교체 정책에 비해 크게 우수한 성능을 가지지 못한다. LFU(Least Frequently Used)는 사용 빈도수가 가장 적은 화일을 먼저 교체하는 정책이다. 이 정책은 많이 사용된 화일은 계속해서 사용될 가능성이 높다는 점에 기반 한다. 그러나 처음에 많이 사용되고 나중에 사용되지 않는 화일을 제대로 교체하지 못하는 단점이 있다. SIZE는 크기가 가장 큰 화일을 먼저 교체하는 정책이다. 인터넷 캐시에서는 하드웨어 캐시나 화일 시스템 캐시와는 다르게 캐시의 단위가 화일 단위이기 때문에 캐시할 화일의 크기가 각기 다르다. 따라서 크기가 큰 하나의 화일이 크기가 작은 여러 개의 화일을 교체하는 경우가 발생한다. 이 정책은 캐시된 화일 중에서 크기가 가장 큰 화일을 먼저 교체함으로써 이러한 문제점을 개선한다. SIZE 교체 정책은 인터넷 캐시에서 우수한 성능을 가진 것으로 평가하고 있다 [3][7].

성능 평가는 서버에 도착하는 요청의 수를 기준으로 한 적중률(hit rate:HR)과 프락시가 반환하는 클라이언트-요청 바이트를 측정하는 가중치 적중률(weighted hit rate:WHR)을 사용한다[6].

2. 액세스 패턴 분석

2.1 작업부하

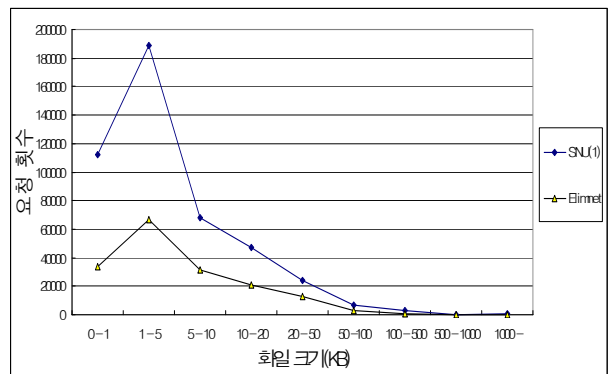
웹의 통신량 특성을 파악하기 위해 본 연구에서는 2개의 작업부하를 분석하였다. 이 중 두 가지는 서울대(SNU)에서 운영하고 있는 프락시 서버의 로그이고, 나머지 하나는 제이씨현의 엘림네트(elimnet)에서 운영하고 있는 프락시 서버의 로그이다. <표 1>에 알 수 있듯이 엘림네트보다는 외부에 개방된 서울대의 프락시 서버가 액세스 횟수 및 통신량이 3 배 이상 많다.

<표 1> 작업부하

	기간	액세스 횟수	총 전송 바이트량(MB)
SNU	1일	451,335	5,577.69
Elimnet	1일	170,512	1,711.75

2.2 화일 크기

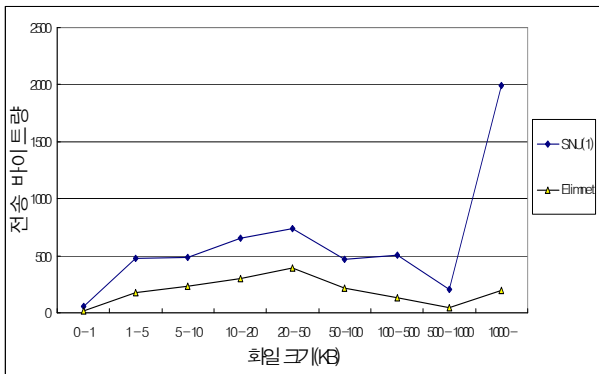
화일 크기에 대한 액세스 분포는 평균치보다 작은 것에 집중되어 있고, 몇몇의 아주 큰 화일들이 존재한다. 캐시에 크기가 큰 화일을 저장하지 않고 사용자가 많이 사용하는 크기가 작은 화일을 저장한다면 캐시 적중률은 상당히 증가할 것이다. 그러나 크기가 큰 화일이 캐시에서 적중되면 가중치 적중률이 많이 증가한다.



(그림 1) 화일 크기별 요청 분포도

(그림1)에서 화일의 크기별 사용자의 요청 횟수를 보면, 1K - 5K 크기를 가진 화일에 가장 많은 액세스가 집중되어, 평균치 이하의 크기를 가진 화일에 액세스가 집중함을 알 수 있다. 반면에 크기가 100K 이상인 화일에는 사용자의 액세스가 아주 적게 발생함을 알 수 있다. (그림1)에서 크기가 작은 화일에

대한 액세스가 가장 많이 발생했음에도 불구하고, (그림2)를 보면 크기가 큰 파일의 전송 바이트량이 크기가 작은 파일의 전송 바이트량과 비슷하게 나타난다. 이것은 작은 파일들을 많이 요청함으로써 망 통신량이 증가할 수도 있지만, 크기가 큰 파일을 몇 번만 액세스하더라도 망 통신량이 많이 증가함을 의미한다.



(그림 2) 파일 크기별 전송 바이트량 분포도

<표 2>는 작업부하별로 파일 타입에 따른 사용자 액세스 횟수와 비율을 나타낸 것이다. 텍스트 타입에는 “html, htm, txt”와 같은 확장자를 가진 파일이 포함되고, 이미지 타입에는 “gif, jpeg, xbm” 등의 확장자를 가진 파일이 포함된다. 오디오 타입에는 “au wav, mp3” 등이 포함되고 비디오 타입에는 “mpeg, mpg, qt, mov, avi, movie” 등이 포함된다. 알려지지 않은(unknown) 타입으로 분석된 파일은 대부분이 CGI에 의해 생성된 것이다. 사용자의 액세스는 이미지와 텍스트 파일에 집중되어, 95%이상의 액세스가 이미지와 텍스트 타입의 파일에 집중되었다.

<표 2> 타입별 사용자 액세스 횟수

파일 타입	SNU(1) (비율%)	Elimnet (비율%)
image	302,798(67.1%)	123,970(72.8%)
text	130,619(28.9%)	42,599(25.0%)
video	512(0.1%)	89(0.0%)
audio	1,274(0.3%)	394(0.2%)
application	7,928(1.8%)	881(0.5%)
multipart	87(0.0%)	5(0.0%)
unknown	8,117(1.8%)	2,574(1.5%)
전체	451,335(100%)	170,512(100%)

<표 3>에서 타입별 전송 바이트를 살펴보면, 이미지와 텍스트가 액세스 횟수와는 다르게 절대적인 비율을 차지하지 않는다. 왜냐하면 텍스트 파일의 크기는 다른 타입의 크기보다 작기 때문에 액세스 횟수가 많더라도 전체 전송 바이트량은 크기가 큰 다른 타입의 파일에 비해 적게 나타난다. 이미지 타입이 전체 바이트 전송량의 45% - 70%을 차지하고, 텍스트는 10% 내외를 차지한다. 또한 타입별 액세스 횟수와는 달리, 오디오와 애플리케이션 타입을 가진 파일의 전송 바이트량이 상대적으로 큰 비율을 차지한다.

<표 3> 타입별 전송 바이트량 (단위:MB)

파일 타입	SNU(1) (비율%)	Elimnet (비율%)
image	2,514 (45.1%)	1,209 (70.7%)
text	628 (11.3%)	203 (11.9%)
video	293 (5.2%)	36 (2.0%)
audio	490 (8.8%)	15 (0.9%)
application	1,621 (29.1%)	235 (13.7%)
multipart	3 (0.0%)	4 (0.3%)
unknown	28 (0.5%)	9 (0.5%)
전체	5,577 (100%)	1,711 (100%)

타입별 파일의 평균 크기를 보면, 타입에 따라 많은 차이가 있다. 이미지 파일의 평균 크기는 8K - 9K 정도이고, 텍스트 파일은 5K 정도로 나타난다. 반면에 오디오 타입의 파일은 40K - 400K, 비디오 타입의 파일은 400K - 570K 의 평균 크기를 보인다. 이처럼 파일의 타입은 파일의 크기와 밀접한 관계가 있다. 비디오나 오디오의 파일의 크기는 텍스트나 이미지 파일의 크기보다 상당히 크다. 캐시의 크기가 제한되어 있을 때, 비디오나 오디오 파일 대신에 텍스트나 이미지 타입의 파일을 저장한다면 보다 많은 파일을 저장할 수 있을 것이다. 또한 하나의 큰 비디오 파일은 수많은 텍스트 파일을 제거할 수도 있다. 만약 캐시에 크기가 큰 몇몇 파일만 존재한다면 적중률은 상당히 감소할 것이다. 그러나 크기가 큰 파일이 캐시 내에서 적중된다면, 가중치 적중률은 상당히 증가한다. 캐시는 적중률과 가중치 적중률을 상호 보완하여 파일들을 저장할 필요가 있다.

2.3 캐시 성능

캐시 크기를 제한하지 않고 실험함으로써 실험 작업부하의 최적 적중률과 최적 가중치 적중률을 구하였다. <표 4>를 보면, 실험 작업부하의 적중률은 39% - 49%까지 나타나며 가중치 적중률은 30% - 46%이다. 적중률이 가중치 적중률보다 높게 나타나는 이유는 실제 캐시에서 적중된 화일의 평균 크기가 사용자가 액세스한 화일의 평균 크기보다 작기 때문이다. 즉 사용자는 평균 화일 크기보다 작은 화일을 더 많이 중복 액세스함을 나타낸다.

이 결과는 다른 연구 결과[2]와 마찬가지로 사용자의 액세스 횟수가 높으면 적중률과 가중치 적중률이 증가함을 알 수 있다.

<표 4> 최적 캐시 적중률 및 가중치 적중률

	적중횟수	적중률	총 적중 크기(MB)	가중치 적중률
SNU(1)	220,815	48.9%	2,554.08	45.8%
Elimnet	66,343	38.9%	504.53	29.5%

<표 5> 타입별 유효한 화일의 수와 크기

화일 타입	유효한 화일 수	비율	유효한 화일 크기(MB)	비율
image	39,355	69.8%	340.21	53.0%
text	14,607	26.0%	64.15	10.0%
video	88	0.2%	41.01	6.4%
audio	217	0.4%	67.70	10.5%
application	1,082	1.9%	125.92	19.6%
multipart	17	0.0%	0.78	0.1%
unknown	974	1.7%	2.41	0.4%
전체	56,340	100%	642.22	100%

<표 5>는 SNU(1) 서버에서 사용자에게 의해 두 번 이상 액세스된 화일의 수와 크기를 화일 타입별로 분석한 것이다. 화일이 사용자에게 의해 두 번 이상 액세스된다는 것은 첫 번째 액세스 때 화일이 캐시에 저장되고, 다음 번 액세스 때에는 캐시에서 적중됨을 의미한다. 캐시는 이러한 화일을 저장해야만 그 효과가 나타난다[5]. 본 논문에서는 캐시에 저장할 필요성이 있는 두 번 이상 참조된 화일의 총 크기를 구하여 캐시의 전체 크기로 결정하였으며, 앞으로 이 크기를 “유효 캐시 크기”라는 용어로 사용

한다. SNU(1)의 경우 유효 캐시 크기는 640MB이며, Elimnet의 경우 유효 캐시 크기는 200MB이다.

3. 캐싱 전략

사용자의 액세스 패턴을 분석한 결과 논 논문에서 적용하려는 캐싱 전략은 다음과 같다.

- 인터넷에는 다양한 타입의 화일이 존재하고, 각 타입은 캐시의 성능에 영향을 미치고 있다. 크기가 작은 텍스트, 이미지 화일 타입은 캐시의 적중률에 유리하게 작용하고, 상대적으로 크기가 큰 오디오, 비디오 화일 타입은 캐시의 가중치 적중률에 유리하게 작용한다. 지금까지의 인터넷 캐시 전략은 모든 화일을 동일하게 취급하였지만, 본 논문에서는 화일 타입별로 캐싱을 수행한다.

- <표 1>과 <표4>에서 알 수 있듯이 액세스 횟수가 높으면 적중률이 높음을 알 수 있다. 따라서 화일 타입별로 크기를 결정할 때 타입별 요청 횟수와 전송 바이트량을 중요한 팩터로 이용한다.

- <표 5>에서 나타나듯이 화일 타입별로 유효한 화일의 수와 크기 비율을 중요한 팩터로 이용한다.

화일 타입별로 캐시를 분리할 때 캐시의 크기는 4가지 팩터의 비율을 산술 평균하여 사용하였다. <표 6>은 타입별 캐시의 크기 비율이다. 캐시를 총 3개의 부분으로 구분하였는데, 이미지 캐시, 텍스트 캐시, 그리고 나머지(etc) 캐시로 분리하였다.

<표 6> 타입별 캐시의 크기 비율

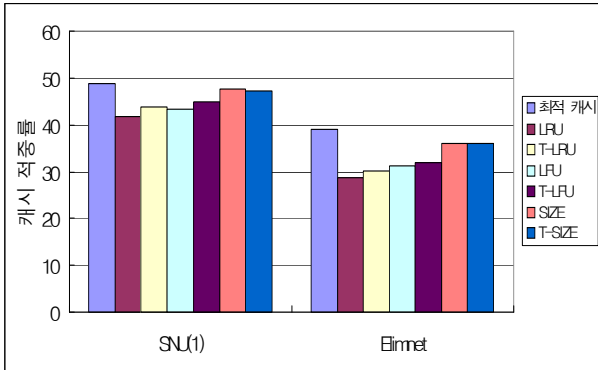
캐시 타입	SNU(1)	Elimnet
image cache	58.8%	73.9%
text cache	19.1%	18.5%
etc cache	22.1%	7.6%

4. 성능 평가 및 결론

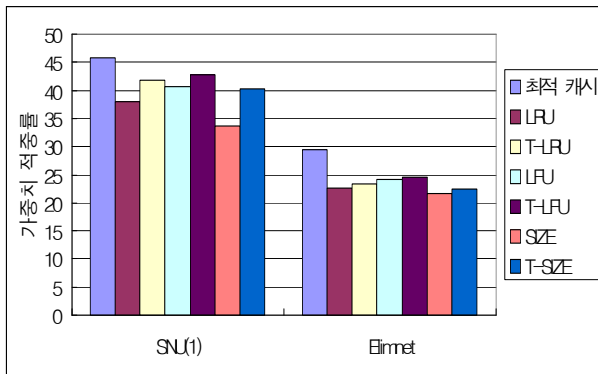
각 작업부하별로 LRU, LFU, SIZE 캐시 교체 정책을 기존 방법대로 캐시에 적용했을 때의 성능과, 타입별로 구분한 캐시에 적용했을 때의 성능을 비교하였다.

(그림 3)과 (그림 4)를 보면 하나의 통합된 캐시를 사용하는 것보다 화일 타입별로 캐시를 분류하여 캐싱 하는 것이 성능을 향상시키는 것으로 나타났다. 적중률 면에서 LFU, LRU 교체 정책을 사용할 때

타입별 캐시가 3% 가까이 성능 증가를 보였다. 그러나 SIZE 교체 정책을 적용하면 약간 감소하거나 유사한 성능을 보였다. 또한 SIZE 교체 정책이 캐시 적중률 면에서는 가장 우수한 성능을 보였다.



(그림 3) 교체 정책에 따른 캐시 적중률



(그림 4) 교체 정책에 따른 가중치 적중률

타입별 캐시는 적중률보다 가중치 적중률 면에서 보다 우수한 성능 향상을 보이는데, 작업 부하에 따라 1% - 10% 정도 성능이 증가하였다. 가중치 적중률은 모든 캐시 교체 정책에서 성능 향상을 보였으며, SIZE 교체 정책의 경우 가장 높은 성능 증가가 나타났다. 작업 부하에 따라 성능 향상 비율에 차이가 있는데, 비디오나 오디오와 같이 크기가 큰 화일의 비율이 적은 Elimnet의 경우 낮은 성능 증가를 보였고, 크기가 큰 화일이 많은 비율을 차지하는 SNU(1)의 경우 높은 성능 증가를 나타냈다. 즉, 타입별 캐시는 비디오나 오디오처럼 크기가 큰 화일의 액세스가 많은 경우 더욱 우수한 성능 향상을 나타낸다.

참고문헌

[1] Sambit Sahu, Prashant Shenoy, Don Towsley, "Design Considerations for Integrated Proxy

Servers", Proc. of IEEE NOSSDAV'99, pp. 247-250, June 1999.

[2] B. M. Duska, D. Marwood and M. J. Feeley, "The Measured Access Characteristics of World-Wide-Web Client Proxy Caches", Proc. of the USENIX Symposium on Internet Tech. and System", DEC. 1997.

[3] S. Williams, M. Abrams, C. R. Standrige, G. Abdulla, and E. A. Fox, "Removal policies in network caches for world-wide web document", In Proceedings of ACM SIGCOMM '96, Aug. 1996.

[4] M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Riber, and E. A. Fox, "Multimedia Traffic Analysis Using CHITRA95", ACM Multimedia 95-Electronic Proceedings, Nov. 1996.

[5] S. Williams, M. Abrams, C. R. Standrige, G. Abdulla, and E. A. Fox, "Caching proxies : Limitations and potentials", Proc. of 4th International WWW conference, Dec. 1995.

[6] A. Bestavros, R. L. Carter, M. E. Crovella, C. R. Cunha, A. Heddays, S. A. Mirdad, "Application-Level Document Caching in the Internet", Proc. of SDNE '95 2nd International Workshop on Services in Distributed and Networked Environments, Jun. 1995.

[7] Roland Peter Wooster, "Optimizing Response Time Rather than Hit Rates of WWW Proxy Caches", Master Thesis, Virginia Polytechnic Institute and State University, Dec. 1996.