

사이트의 인기를 고려한 문서 교체 정책

유행석⁰ 장태무
동국대학교 컴퓨터공학과
{heehang⁰, jtm}@dongguk.edu

Document Replacement Policy Considering Site Popularity

Hang Suk Yoo⁰, Tae Mu Chang
Dept. of Computer Engineering, Dongguk University

요 약

대부분의 웹 캐시는 문서를 기반으로 하여 캐시 내에 임시적으로 문서를 저장하고 사용자의 요청이 있을 때 그에 해당된 문서가 캐시 내에 존재하면 그 문서를 사용자에게 전송해 주고, 캐시 내에 존재하지 않을 때에는 새로운 문서를 서버에게 요청하여 캐시 내에 복사를 하고 사용자에게 되돌려 준다. 이때 캐시의 용량 초과로 인해 새로운 문서를 기존의 문서와 교체하기 위해 문서 교체정책(replacement policy)을 사용한다. 일반적인 교체정책에는 문서를 기반으로 한 LRU기법이나 LFU기법 등이 있고, 이러한 여러 가지 교체정책을 사용하여 캐시내의 문서를 효과적으로 교체한다. 하지만, 위의 교체정책은 사이트의 인기를 고려하지 않고 문서 요청 빈도수만을 고려하여 교체정책을 수행한다. 따라서 본 논문에서는 요청이 빈번한 문서와 사이트의 인기를 고려한 교체정책을 사용하여 요청이 빈번하지 않은 문서를 효과적으로 교체함으로써 캐시의 적중률(hit-rate)을 높이고, 캐시의 내용을 효과적으로 관리할 수 있는 현대적인 네트워크 환경에 적합한 문서 교체정책인 사이트의 인기를 고려한 문서 교체 정책을 제시한다.

1. 서 론

인터넷 사용이 계속적으로 증가함[1]과 동시에 서버의 부하 및 네트워크 통신량의 증가, 검색 지연 등의 여러 가지 문제가 대두되었다. 따라서 인터넷의 성능을 향상시키는 효율적인 방법으로 웹캐시를 사용하였으며[2], 이로 인해 전체적인 네트워크 통신량의 감소 및 반응시간을 효과적으로 개선시켰다[3]. 웹캐시는 사용자의 문서 요청이 있을 때 요청한 문서가 캐시 내에 존재하면 그 문서를 사용자에게 전송해 준다. 그리고 캐시내에 존재하지 않으면 새로운 문서를 서버에게 요청하여 캐시내에 복사를 하고 그 문서를 사용자에게 되돌려 준다. 이 때 캐시의 용량 초과로 인해 새로운 문서를 기존의 문서와 효율적으로 교체하기 위해서 교체정책(replacement policy)을 사용한다. 따라서, 문서 교체정책에 따라 성능의 차이를 보일 수 있다. 보편적으로 사용용 많이 하는 웹캐시 교체정책으로는 LRU(Least Recently Used)기법과 LFU(Least Recently Used)기법등이 많이 사용된다. 하지만 위의 교체정책은 사이트의 인기를 고려하지 않고 캐시내의 문서의 요청 빈도수만을 고려하여 교체정책을 수행한다. 즉 요청이 빈번한 사이트를 고려하여 그 사이트의 객체를 캐시에 저장함으로써 캐시의 적중률(hit-rate)을 높이고 캐시의 내용을 효과적으로 관리함과 동시에 요청이 빈번하지 않은 문서를 효과적으로 교체할 수 있다. 따라서 본 논문에서는 현대 네트워크 환경에 적합한 캐시 정책인 사이트의 인기를 고려한 문서 교체 정책을 제시하고 빠른 응답시간과 서버로의 문서 요청을 줄여 네트워크 통신량을 감소시킨다.

2. 기존 연구

웹캐시는 서버와 클라이언트 사이에 위치하며, 클라이언트의 페이지 요청시 웹캐시 내에 임시적으로 저장된 페이지를 탐색하여 클라이언트가 요청한 페이지의 유무를 검사하고 웹캐시 내에 저장된 페이지가 존재한다면 그 페이지를 돌려주거나, 웹캐시 내에 존재하지 않을 때 서버에게 요청하여 클라이언트가 요구하는 페이지를 웹캐시내에 복사를 하고 그 페이지를 클라이언트에게 돌려준다. 이때 웹캐시의 용량이 초과되어 서버에게 요청한 페이지를 웹캐시 내에 삽입하고, 기존의 페이지를 삭제하는 작업을 수행하게 된다. 이러한 일련의 과정을 웹캐시 교체정책이라고 한다. 일반적으로 LRU 기법과 LFU 기법이 많이 사용된다. 또한, 최근에는 사이트를 기반으로 한 교체기법도 제시되었다. LRU는 웹캐시의 용량이 초과로 인해 문서의 교체가 필요할 때 기존의 문서들 중에 최근에 가장 적게 사용된 문서를 새로운 문서와 교체하는 교체정책이다. 즉 참조시간이 가장 오래된 문서를 새로운 문서와 교체를 의미한다. 하지만, LRU 교체정책은 전체참조(total-reference)의 2/3가 문서를 한번만 접근(access) 하기 때문에 한번만 참조되는 문서들로 범람 될 우려가 있고, 또한 문서 부재에 관계없이 모든 페이지의 참조시간에 대한 정보를 유지해야 하므로 그에 따른 메모리 요구량이

높다. LFU 기법은 LRU와는 달리 도큐먼트 교체가 일어날 때 가장 사용 빈도가 낮은 도큐먼트와 새로 요청된 도큐먼트와 교체를 하는 교체정책이다. 즉 참조 횟수가 가장 적은 도큐먼트와 새로운 도큐먼트의 교체를 의미한다. 하지만, LFU 교체정책은 가장 적은 참조횟수를 가진 도큐먼트만을 교체하기 때문에 캐시의 오염(pollution)을 초래할 수 있다. 즉, 빈도수가 적은 도큐먼트들이 캐시 내에 존재하기 때문에 이러한 현상이 일어날 수 있다. 사이트를 기반으로 한 교체정책[4]은 모든 도큐먼트를 사이트 별로 저장하는 것이다. 하지만 한번 요청된 객체도 사이트별로 저장하므로 캐시용량의 낭비를 초래할 수 있다. 또한, 사이트 내에 요청이 빈번한 객체가 하나만 존재한다고 할 때 교체정책에 있어서 그 객체는 다른 사이트 내의 객체들 보다 교체될 수 있는 확률이 높을 것이다. 결국 캐시의 적중률이 감소될 수 있다는 것이다.

3. 본론

웹 캐시 교체정책은 효율적인 페이지 교체를 통하여 웹 서버와의 접속을 줄이고, 사용자의 도큐먼트 요청 시 적중될 수 있는 확률을 높여 인터넷의 성능을 향상시킨다[5]. 본 논문에서는 교체정책에 있어서 적중률을 높이고, 반응시간을 줄이는데 목적을 두었다.

3.1 제안하는 웹 캐시 구조

일반적인 웹 캐시의 구조는 캐시 내에 각각의 객체들을 저장한다. 본 논문에서 제안하는 캐시의 구조는 사이트와 객체가 혼합된 구조를 보인다. 사이트의 인기도를 고려하여 자주 요청되는 사이트는 그 사이트 내에 요청된 객체들을 사이트별로 저장하고, 자주 요청되지 않은 사이트는 그 사이트 내에 요청된 객체만을 저장한다. 또한, 자주 요청된 사이트라 할지라도 그 사이트 내에 특정 객체만을 클라이언트가 요청하면 그 객체는 사이트별로 저장하지 않고 해당된 객체만을 저장한다.

3.2 도큐먼트 교체정책

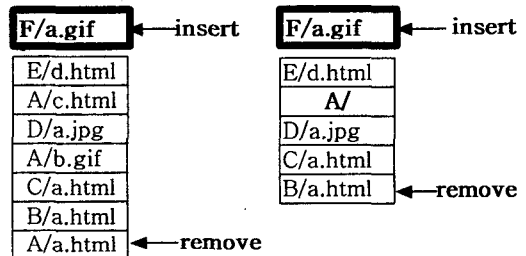
기존의 교체정책은 객체를 기준으로 교체를 한다. 하지만 본 논문에서는 자주 요청되는 사이트를 고려하기 위해 기준값(threshold)을 두어서 기준값을 만족하는 사이트가 캐시 내에 존재하면 그 사이트 내에 포함된 객체들을 사이트별로 저장한다. 즉, 클라이언트가 여러 번 요청한 객체들을 포함하고 있는 사이트는 다음에 접속될 확률이 높다는 것이다. 기준값을 만족하지 않은 사이트가 캐시 내에 존재할 때에는 클라이언트가 요청한 객체만을 저장한다. 따라서 이러한 사이트를 캐시에 저장

함으로써 적중률을 높일 수 있다. 캐시의 용량초과로 인해 교체정책을 수행할 때에는 기준값을 만족하는 사이트 내에 객체들 중에 요청 빈도수가 가장 낮은 것이나 캐시 내에 사이트별로 저장되지 않은 객체들 중에 요청 빈도수가 가장 낮은 것이 삭제되고, 새로 요청한 객체가 삽입된다. 또한 캐시 내에 저장된 객체보다 사이트별로 저장된 객체들의 요청 빈도수가 낮다면 사이트별로 저장된 객체들 모두를 제거하여 캐시의 오염을 줄이고, 캐시의 여유공간을 확보하여 자주 일어날 수 있는 교체정책 수행을 줄임으로써, 이때 발생하는 지연시간(latency)을 감소시킬 수 있다.

request.	requested document
request 1	http://A/a.html
request 2	http://B/a.html
request 3	http://C/a.html
request 4	http://A/b.gif
request 5	http://D/a.jpg
request 6	http://A/c.html
request 7	http://E/d.html
request 8	http://F/a.html

cache size = 7

[표-1] 요청한 도큐먼트

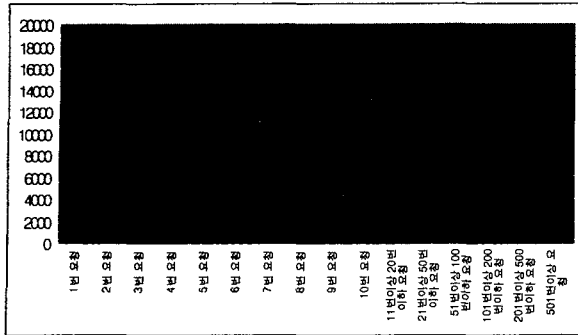


[그림1] 기존 교체정책 [그림2] 제안한 교체정책

[표1]은 요청한 도큐먼트를 나타내고 캐시 크기를 7로 제한하였을 때 [그림1]은 기존의 교체정책 [그림2]는 제안한 교체정책을 나타내고 있다.

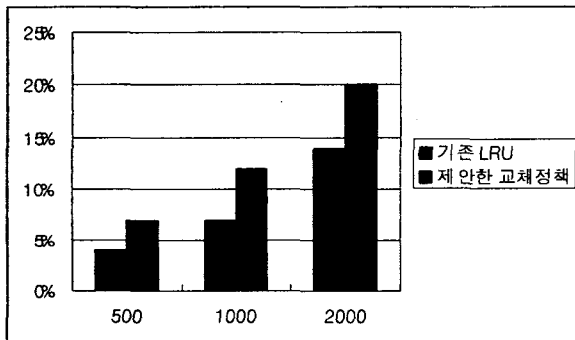
4. 구현 및 성능평가

본 논문에서 제안하는 사이트의 인기도를 고려한 교체정책을 구현하기 위해 동국대학교 웹서버 로그 파일 분석하였다. 총 요청수는 64,993개이고, 사이트별로의 요청수 통계는 [그림3]과 같다.



[그림3] 동국대학교 로그 분석

[그림3]에서 사이트의 패턴을 분석하여 통계를 내어 그 통계에 맞게 입력 데이터를 만들었다. 구현 환경은 CPU P-III 900MHZ, RAM이 256MB, HDD 40GB에서 실험하였다. 운영체제는 windows 2000 professional을 사용하였다. TTL(Time To Live)값은 일정하다고 가정하고, 캐시의 크기와 요청된 문서의 크기는 각각 동일하다고 가정하였다. 캐시를 구현하여 입력 데이터를 로그로 하여 캐시 내에서 LRU 교체정책을 사용했을 때의 적중률과 본 논문에서 제안한 사이트의 인기도를 고려한 문서 교체정책을 사용했을 때의 적중률을 비교하였다. 객관성을 유지하기 위해 요청수가 500개일 때와 요청수가 1000개일 때, 2000개일 때 LRU 교체정책과 사이트의 인기도를 고려한 문서 교체정책에 대한 적중률의 추이를 살펴보았다.



[그림4] LRU 교체 정책과 제안한 교체정책의 비교

5. 결론 및 향후과제

[그림4]에서 기존의 LRU 정책 보다 사이트의 인기도를 고려한 교체정책이 약 6%로의 성능향상이 되는 것을 알 수 있다. 즉, 요청이 빈번한 사이트 내의 여러 개의 객체

들을 사이트 별로 캐시에 저장하면, 요청이 빈번한 사이트이기 때문에 다음에 그 사이트에서 접속할 확률이 높고, 클라이언트는 그 인기도가 높은 사이트에 존재하는 객체를 요청할 확률이 높다 라는 것이다. 이로 인해 기존의 교체정책 보다 높은 적중률을 보일 수 있다. 본 논문에서 제안한 사이트의 인기도를 고려한 교체정책을 사용할 때 적중률이 높다는 것은 서버로의 요청을 감소시키는 것은 물론 이에 수반된 네트워크 통신량의 감소 및 반응시간을 효과적으로 개선 할 수 있다는 것이다. 하지만, 본 논문에서 제안한 교체정책에 있어서도 캐시의 일관성 문제가 고려되지 않을 수 없다. 인기도가 높은 사이트 내의 객체들과 서버에 있는 객체들과의 일관성 문제는 Client Polling Method나 Invalidation Protocol Method, TTL, Active Refresh Method[6], Bloom Filter로 해결을 모색해야 할 것이다.

참고문헌

[1] Time Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, "The World-Wide-Web" Communications of the ACM, vol. 37, no. 8 pp. 76-82, Aug

[2] Ari Lutotonen and Kevin Altis, "World-Wide-Web Proxies" 1st Inter. Conf. on the WWW, Geneva, May 1994

[3] P. B. Danzig, R. S. Hall, M. F. Schwartz "A Case for Caching File Objects Inside Internetworks" in Proceeding of the ACM SIGCOMM' 93 September, 1993

[4] Stephen Williams, Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, Edward A. Fox, "Removal Policy in Network Caches for World-Wide-Web Documents", Dept. of Computer Science, Virginia Tech, Blacksburg, Proceedings, ACM SIGCOMM, Stanford, CA, revised August 1997

[5] Site-based approach to Web cache design
Kin Yeung Wong; Kai Hau Yeung. IEEE Internet Computing ; Volume: 5 2001

[6] An active refresh method in Web caching
Gang Zhang; Yantai Shu; Zheng Zhao; Zhijie Guan; Yang, O.W.W. Electrical and Computer Engineering, 2001. Canadian Conference on , Volume: 1 , 2001