

웹 캐시에서 사이트의 인기도에 의한 도큐먼트 교체정책

유행석0 *장태무

*동국대학교 컴퓨터 공학과

(heehang0, jtm)@dongguk.edu

Hang Suk Yoo Tea-Mu Jang

*Dept. of Computer Engineering, Dongguk University

Document Replacement Policy by Site Popularity in Web Cache

요약

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

Abstract

Most web caches save documents temporarily into themselves on the basis of those documents. And when a corresponding document exists within the cache on user's request, web cache sends the document to corresponding user. On the contrary, when there is not any document within the cache, web cache requests a new document to the related server to copy the document into the cache and then turn it back to user.

Here, web cache uses a replacement policy to change existing document into a new one due to exceeded capacity of cache.

Typical replacement policy includes document-based LRU or LFU technique and other various replacement policies are used to replace the documents within cache effectively.

However, these replacement policies function only with regard to the time and frequency of document request, not considering the popularity of each web site.

Based on replacement policies with regard to documents on frequent requests and the popularity of each web site, this paper aims to present the document replacement policies with regard to the popularity of each web site, which are suitable for latest network environments to enhance the hit-ratio of cache and efficiently manage the contents of cache by effectively replacing documents on intermittent requests by new ones.

key word : replacement policy, considering of web site popularity, hit ratio

1. 서론

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 웹 캐시를 사용하였으며[2],

이로 인해 전체적인 네트워크 통신량의 감소 및 반응시간의 효과적인 개선을 가져왔다[3]. 웹 캐시는 사용자의 도큐먼트 요청이 있을 때 요청한 도큐먼트가 캐시 내에 존재하면 그 도큐먼트를 사용자에게 전송해 준다. 그리고 캐시 내에 존재하지 않으면 새로운 도큐먼트를 서버에게 요청하여 캐시 내에 복사를 하고 그 도큐먼트를 사용자에게 되돌려 준다. 이 때 캐시의 용량 초과로 인해 새로운 도큐먼트를 기존의 도큐먼트와 효율적으로 교체하기 위해서 교체정책(replacement policy)을 사용한다. 따라서, 도큐먼트 교체정책에 따라 성능의 차이를 보일 수 있다. 보편적으로 사용을 많이 하는 웹 캐시 교체정책으로는 LRU(Least Recently Used)기법과 LFU(Least Frequently Used)기법 등이 있다. 하지만 위의 교체정책은 사이트의 인기도를 고려하지 않고 캐시내의 도큐먼트의 요청 시간과 크기, 빈도수 만을 고려하여 교체정책을 수행한다. 요청이 빈번한 사이트를 고려하여 그 사이트의 객체를 캐시에 저장함으로써 캐시의 적중률(hit-ratio)을 높이고 캐시의 내용을 효율적으로 관리함과 동시에 요청이 빈번하지 않은 도큐먼트를 효과적으로 교체할 수 있다. 따라서 본 논문에서는 현대 네트워크 환경에 적합한 캐시 정책인 사이트의 인기도를 고려한 도큐먼트 교체 정책을 제시하고 빠른 응답시간과 서버로의 도큐먼트 요청을 줄여 네트워크 통신량을 감소시킨다.

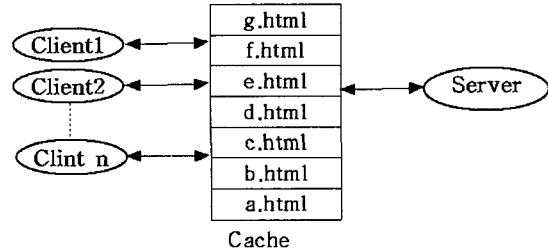
2. 기존 연구

웹 캐시 교체정책은 효율적인 페이지 교체를 통하여 웹서버와의 접속을 줄이고, 클라이언트의 페이지 요청 시 적중될 수 있는 확률을 높여 인터넷의 성능을 향상시킨다. 본 장에서는 기존의 교체정책에 대해 살펴보고, 더 효율적인 웹 캐시 교체정책을 모색한다.

2.1 LRU(Least Recently Used) 교체 정책

가장 일반적인 웹 캐시 교체정책인 LRU[4]는 웹 캐시의 용

량이 초과되어 도큐먼트의 교체가 필요할 때 기존의 도큐먼트들 중에 가장 오래 된 첫 번째 도큐먼트를 새로운 도큐먼트로 교체하는 작업을 수행한다.



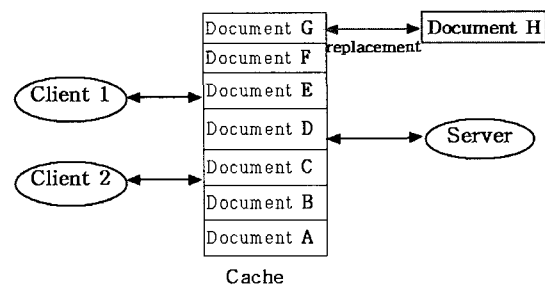
[그림 1] 웹 예서의 기본적인 캐시 구조

[그림 1]은 기본적인 캐시 구조를 보인다. 위의 구조에서 클라이언트가 새로운 도큐먼트를 요구하면 캐시 내에 요청된 도큐먼트가 없기 때문에 캐시는 서버에게 재요청을 하고 서버에서 새로운 도큐먼트를 가지고 와서 캐시 내에 복사를 하고 클라이언트에게 새로운 도큐먼트를 전송해 준다. 이때, 캐시내의 용량이 초과되었을 경우 도큐먼트 교체정책을 수행한다. 여기서 LRU 정책을 사용하기 위해 [표 1]

<Document>	<Referece Time>
Document G	↓
Document F	
Document E	
Document D	
Document C	
Document B	
Document A	

cache size = 7

[표 1] 도큐먼트별로 참조된 시간



[그림 2] LRU를 이용한 도큐먼트 교체 정책

과 같이 도큐먼트 A, B, C, D, E, F, G를 참조된 시간 별로 순서대로 캐시에 저장되었다고 가정하고 새로 요청된 페이지가 도큐먼트 H라고 하였을 때, [그림2]와 같이 도큐먼트 A, B, C, D, E, F, G 중에 참조된 시간이 가장 오래된 도큐먼트 G와 새로운 도큐먼트 H를 교체한다.

그러나, LRU 교체정책은 전체참조(total-reference)의 2/3가 도큐먼트를 한번만 접근(access) 하는 웹의 특성상 한번만 참조되는 도큐먼트들로 범람 될 우려가 있고, 또한 도큐먼트 부재에 관계없이 모든 페이지의 참조시간에 대한 정보를 유지해야 하므로 그에 따른 메모리 요구량이 높다.

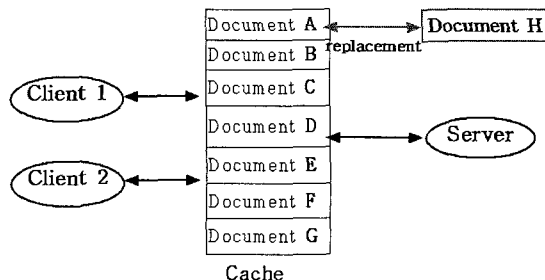
2.2 LFU(Least Frequently Used) 교체 정책

LFU 교체정책은 LRU와는 달리 도큐먼트 교체가 일어날 때 가장 사용 빈도가 낮은 도큐먼트와 새로 요청된 도큐먼트와 교체를 하는 웹 캐시 교체정책이다. 즉 참조 횟수가 가장 적은 도큐먼트와 새로운 도큐먼트의 교체를 의미한다. 새롭게 요청한 페이지를 도큐먼트 H라고 했을 때 [표 2]와 같이 도큐먼트 A, B, C, D, E, F, G를 참조된 횟수 별로 순서

<Document>	<Number of Referece>
Document G	11
Document E	9
Document F	7
Document D	6
Document C	4
Document B	2
Document A	1

cache size = 7

[표 2] 도큐먼트별로 참조된 횟수



[그림 3] LFU를 이용한 도큐먼트 교체 정책

대로 캐시에 저장되었다고 가정하면, [그림 3]과 같이 도큐먼트 A, B, C, D, E, F, G 중에 가장 사용빈도수가 낮은, 즉 참조된 횟수가 가장 적은 도큐먼트 A와 새로운 도큐먼트 H를 교체한다.

하지만, LFU 교체정책은 가장 적은 참조횟수를 가진 도큐먼트만을 교체하기 때문에 캐시의 오염(pollution)을 초래할 수 있다.

따라서 이러한 현상을 피하기 위해서는 일정기간 동안의 참조된 기준값(threshold parameter)을 설정하고 그 값 보다 적은 페이지를 제거해야 한다.

2.3 그 밖의 다른 도큐먼트 교체 정책

일반적인 LRU나 LFU 교체정책을 보완하기 위해 여러가지 교체정책들이 등장하였다. 가장 용량이 큰 도큐먼트를 교체하는 SIZE 교체정책과 SIZE와 LRU의 혼합형인 GD-Size(Greedy Dual-Size) 교체정책[5], LRU의 장점과 LFU의 장점을 혼합시킨 LRFU 교체정책[6], 도큐먼트의 크기별로 교체할 때 사이즈가 크고 요청이 빈번한 도큐먼트가 제거를 최소화하기 위해 참조되었던 빈도수로 나누어 보정해주는 SLRU(Size-adjusted LRU) 교체정책[7]과 모든 도큐먼트를 사이트 별로 저장하는 교체정책[8]등이 있다. 사이트를 기반으로 한 교체정책은 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 교체 될 수 있는 확률이 높아, 결국 캐시의 적중률이 감소시킬 수 있다.

3. 결론

웹 캐시 교체정책의 목표는 효율적인 페이지 교체를 통하여 웹 서버와의 접속을 줄이고, 사용자의 도큐먼트 요청 시 적중될 수 있는 확률을 높여 인터넷의 성능을 향상시키는데 있다[9]. 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

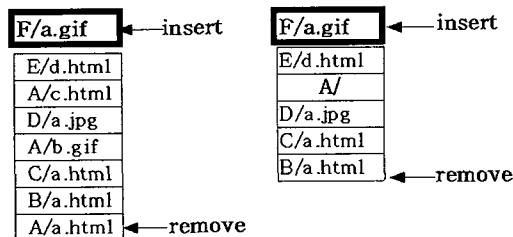
기존의 교체정책은 객체를 기준으로 교체를 한다. 하지만 본 논문에서는 자주 요청되는 사이트를 고려하기 위해 기준값(threshold)을 두어서 기준값을 만족하는 사이트가 캐시 내에 존재하면 그 사이트 내에 포함된 객체들을 사이트별로 저장을 한다.

즉, 클라이언트가 여러 번 요청한 객체들을 포함하고 있는 사이트는 다음에 접속 될 확률이 높다는 것이다. 기준값을 만족하지 않은 사이트가 캐시 내에 존재할 때에는 클라이언트가 요청한 객체만을 저장한다. 따라서 이러한 사이트를 캐시에 저장함으로써 적중률을 높일 수 있다. 캐시의 용량초과로 인해 교체정책을 수행할 때에는 기준값을 만족

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 3] 요청한 도큐먼트



[그림 4] 기존 LRU 교체정책 [그림 5] 제안한 교체정책

하는 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 여유 공간을 확보하여 자주 일어날 수 있는 교체정책 수행을 줄임으로써, 이때 발생하는 지연시간(latency)을 감소시킬 수 있다.

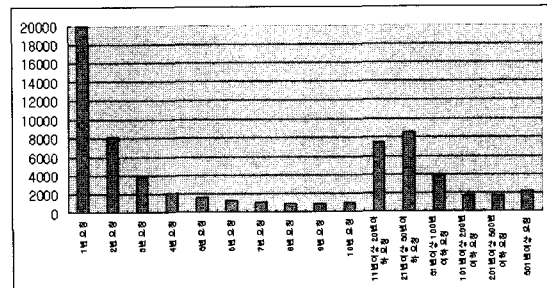
[표 3]은 요청한 도큐먼트를 나타내고, 캐시 크기를 7로 제한하였을 때 [그림 4]는 기존 LRU 교체정책, [그림 5]는 본문에서 제안한 교체정책을 나타내고 있다.

[표 3]을 보면 A 사이트의 도큐먼트들을 빈번하게 요청하는 것을 알 수 있다.

캐시의 용량이 7이고 [표 3]과 같이 도큐먼트 요청이 일어날 때 [그림 4]는 request 1, 2, 3, 4, 5, 6, 7을 캐시에 저장한다. 하지만 클라이언트가 request 8을 요청할 때에는 캐시의 용량 초과로 인해 request 1에 요청된 A/a.html을 삭제하고, 새로 요청한 request 8의 F/a.html 도큐먼트를 삽입한다. 클라이언트가 빈번하게 요청을 하는 도큐먼트들을 가지고 있는 인기도가 높은 A 사이트의 도큐먼트를 삭제한다. 하지만 [그림 5]는 캐시의 용량초과로 인해 도큐먼트를 교체 할 때 B 사이트의 a.html을 삭제한다. 이는 사이트별로 도큐먼트를 저장하고 인기도가 낮은 사이트의 도큐먼트를 삭제하기 때문이다.

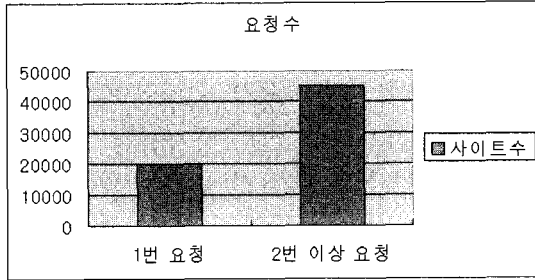
4. 구현 및 성능평가

본 논문에서 제안하는 사이트의 인기도를 고려한 교체정책을 구현하기 위해 동국대학교 웹서버 로그 파일을 분석

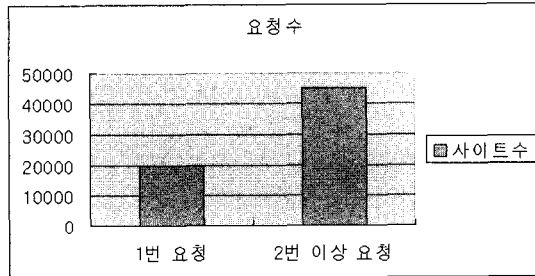


[그림 6] 동국대학교 웹 서버 로그분석

하였다. 총 요청수는 64,993개이고, 사이트별로 요청수 통계는 [그림 6]과 같다.



[그림 7] 동국대학교 웹 서버의 요청 통계(1)



[그림 8] 동국대학교 웹 서버의 요청 통계(2)

[그림 7]과 [그림 8]의 사이트 패턴을 분석하여 500개 1000개 2000개의 입력 데이터를 구성하였다.

동국대학교 웹 서버 로그 통계에 의한 입력 데이터는 [그림 9]와 같이 site 와 object 필드로 구성하였다.

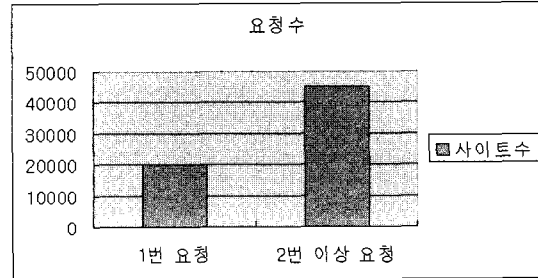
210.94.164.66/a.html	→	<table border="1"> <thead> <tr> <th>site(ip)</th> <th>object</th> </tr> </thead> <tbody> <tr> <td>210.94.164.66</td> <td>a.html</td> </tr> </tbody> </table>	site(ip)	object	210.94.164.66	a.html
site(ip)	object					
210.94.164.66	a.html					

[그림 9] 입력 데이터

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 도큐먼트의 크기는 각각 동일하다고 가정하였다. 캐시를 구현하여 입력 데이터를 로그로 하여 캐시 내에서 LRU 교체정책을 사용했을 때의 적중률과 본 논문에서 제안한 사이트의 인기도를 고려한 도큐먼트 교체정책을 사용했을 때의 적중률을 비교하였다. 객관성을 유지하기 위

해 요청수가 500개일 때와 요청수가 1000개일 때, 2000개일 때 LRU 교체정책과 사이트의 인기도를 고려한 도큐먼트 교체정책에 대한 적중률의 추이를 살펴보았다.



[그림 10] LRU 교체 정책과 제안한 교체정책의 비교

5. 결론 및 향후과제

[그림 10]에서 기존의 LRU 정책 보다 사이트의 인기도에 의한 교체정책이 약 6% 정도 적중률이 향상 되는 것을 알 수 있다. 즉, 요청이 빈번한 객체들을 포함하고 있는 인기도가 높은 사이트를 캐시에 저장함으로써, 캐시의 오염을 줄이고, 사이트별로 저장되어있기 때문에 교체정책이 일어나면 인기도가 높은 사이트내의 객체들은 캐시에 오래 남게 되므로 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 사이트내의 객체들과 서버에 있는 객체들과의 일관성문제는 Client Polling Method나 Invalidation Protocol Method, TTL, Active Refresh Method[10], 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] Exact analysis of Bernoulli superposition of streams into a least recently used cache Levy, H.; Morris, R.J.T.; Software Engineering, IEEE Transactions on , Volume: 21 Issue: 8 , Aug 1995 Page(s): 682 -688
- [5] Using performance maps to understand the behavior of Web caching policies Murta, C.D.; Almeida, V.A.F.; Internet Applications, 2001. WIAPP 2001. Proceedings. The Second IEEE Workshop on , 2001 Page(s): 50 -56
- [6] D. Lee et al., "LRFU: A Spectrum of Policies that Subsumes the Least Recently Used and Least Frequently Used Policies," IEEE Trans. Computers, vol.50, no. 12, 2001, pp. 1352-1361
- [7] LRU-SP: a size-adjusted and popularity-aware LRU replacement algorithm for web caching Kai Cheng; Kambayashi, Y.; Computer Software and Applications Conference, 2000. COMPSAC 2000. The 24th Annual International , 2000 Page(s): 48 -53
- [8] Site-based approach to Web cache design Kin Yeung Wong; Kai Hau Yeung IEEE Internet Computing , Volume: 5 2001
- [9] 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
- [10] 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



유행석

2000 중부대학교 전자계산학 전공 (공학사)
 2003 동국대학교 컴퓨터공학과 석사과정
 관심분야: 온라인 게임, 웹 캐시, 병렬컴퓨터구조, 분산처리, Grid, 게
 임보안



장태무

1977 서울대학교 전자공학과 (학사)
 1979 KAIST 전산학과 (석사)
 1996 서울대학교 컴퓨터공학과 (박사)
 1981 - 현재 동국대학교 컴퓨터공학과 교수
 1998 미국 University of South West Louisiana 교환교수
 관심분야: 병렬컴퓨터구조, 분산처리, Embedded System, Grid