웹 캐싱을 위한 개선된 교체 정책 알고리즘

김 현 섭°, 홍 영 식
동국대학교 컴퓨터공학과
{khs7615°, hongys}@dongguk.edu

Advanced Replacement Policy Algorithm for Web Caching

Hyun Seob Kim°, Young Sik Hong
Dept. of Computer Engineering, Dongguk Univ.

요 약

인터넷 사용의 급격한 증가로 인해 웹 정보의 양적 팽창과 더불어 네트워크 병목현상과 웹 서버의 과부하 문제가 증가함에 따라 사용자의 접근 지연시간도 증가하게 되었다. 이러한 접근 지연 시간을 줄이는 방안으로 프록시 캐싱(Proxy Caching)이 사용되었고, 효율적인 프록시 서버 캐시 운영을 위한 캐시 교체 정책에 관한 연구가 많이 진행되어 지고 있다. 따라서, 본 논문에서는 서버의 오버헤드를 줄이면서, 캐시에 저장되어 있는 오브젝트를 요청한 클라이언트에게 짧은 지연시간에 전달하고자 하는 개선된 교체 정책 알고리즘을 제안하고 실험을 통해 성능을 평가한다.

1. 서 론

인터넷의 사용자수가 기하급수적으로 증가함에 따라 웹서버의 부하와 인터넷 트래픽(Traffic)도 현저하게 증가하였다. 이로 인하여 웹 서버와의 통신이 어려워지고, 통신이 가능해지더라도 높은 지연시간과 잦은 통신 불량으로 인해 사용자들의 요구에 취약점이 되고 있다. 이러한 취약점들을 해결하기 위해 많은 연구가 이루어져 왔고, 그 중 가장 대표적인 방안으로 프록시(Proxy) 서버에서 캐시를 사용하여 히트율(Hit Ratio)를 최대화함으로써 지연시간을 줄이고자 하였다. 하지만 제한된 캐시 사이즈만으로 이것을 해결할 수 없어서 캐시내에 교체 정책을 통해 이 문제를 해결하고 있다

기존의 연구 방향은 성능을 최적화하기 위해 접근 요구의 최신성(Recency) 또는 동일한 오브젝트에 대한 접근 요구의 발생 빈도(Frequency)[2][8] 중 어느 한쪽만을 고려한 연구가 대부분이었으나, 그 이후로는 지연시간을 줄이기 위해 Cost-to-Size 모델을 고려하여 연구들이 이루어지고 있다. 즉, 캐시에 오브젝트를 저장해서 프록시 서버와 원격 서버에서의 트래픽을 최소화하고 클라이언트의 히트율을 높이고자 하였다. 반면에 많은 연구들이 지연시간보다는 히트율에 많은 비중을 두고 연구를 하여 왔다. 최근에는 지연시간(Latency)[5]을 이용한 교체 정책도 발표되었지만, 지연 시간만 측정을 하고 히트율이나 바이트 히트율을 측정하지 않았다.

본 논문에서는 지연시간을 줄이기 위한 것을 목적으로 트레이스(Trace)에 기반한 클라이언트와 프록시 그리고 원격서버에서의 접근 요구의 최신성과 발생 빈도수, 지연시간을 모두 반영하여 프록시 캐시에서의 지연 시간을 측정하여 캐시 교체 정책을 통해 전체 히트율, 바이트 히트율(Byte Hit Ratio), 지연시간을 측정하여 성능을 평가하였다.

2. 관련연구

기존에 연구되어 왔던 전통적인(Traditional) 교체 정책에서 확장된 교체 정책들은 Cost-to-Size[7] 모델을 고려하여 다음과 같은 방향으로 연구되어져 왔다.

Size-Adjusted LRU[3][6] 교체 정책은 Cost-to-Size 모델에 따라 우선순위를 두어 사용하는 정책으로 비용을 고려하여 측정한 후 우선 순위가 작은 것부터 교체하는 방법이다.

GDS(Greedy-Dual Size)[1] 교체 정책은 Greedy-Dual 알고리즘에서 확장된 정책으로 다양한 캐시 사이즈와 비용을 고려하는 교체 방법이다. 이 알고리즘은 Greedy-Dual과 마찬가지로 유틸리티 함수(Utility Function)에 가장 작은 값을 얻은 오브젝트를 교체하게 된다. 키 값은 오브젝트의 사이즈와 비용면을 고려하여 결정된다. 이 교체 정책에서는 빈도수에 대한 고려를 하지 않았다. 자주 요청되는 것은 다시 요청될 가능성이 많다는 것을 고려하면 성능 향상에 커다란 영향을 미치게 된다.

GDSP(Popularity-Aware Greedy-Dual Size)[3] 교체 정책은 GDS에서 확장된 알고리즘으로 오랜 기간동안 웹 오브젝트의 접근이 빈번하게 일어나는 것을 고려하여 오브젝트의 키 값을 설정하는 방법이다. 이 교체 정책은 GDS와 달리 요청된 오브젝트외에 다른 오브젝트의 키 값에도 영향을 미치게 된다.

Segmented LRU[3] 교체 정책은 참조된 빈도수가 많은 것은 앞으로의 참조율도 많을 것이라 보고 고려하게 된다. 단, 오브젝트의 사이즈가 동일하다는 전제하에서 이루어지는 교체 정책이다. 하지만, 웹에서 오브젝트의 사이즈가 일정할 수 없다는 점을 고려해야 한다.

LRU-SP(Size-Adjusted and Popularity-Aware LRU)[3] 교체 정책은 Cost-to-Size 모델에서 힙(heap) 구조를 바탕으로 캐시를 관리하고 , 우선순위를 고려하여 교체할 오브젝트 설정하게 된다. 이 기법은 히트율과 바이트 히트율이 다른 정책에 비해 높은 성능을 보이고 있는 반면에 지연시간을 고려하지 않은 취약점을 가지고 있다.

3. 개선된 교체 정책

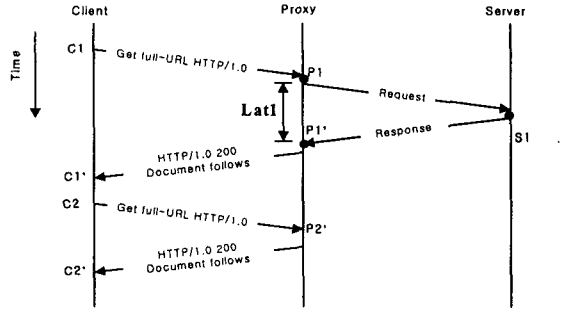
본 논문에서는 기존의 교체 정책을 개선한 방안으로 LRU-SP 교체 정책에서 확장한 교체 정책으로 네트워크 지연시간(프록시와 서버 사이의 지연시간)을 고려하여 LRU-SP-LAT 라는 교체 정책을 통해 프록시 서버와 원격 서버 사이에 요구되는 트래픽을 줄이고 클아이언트

가 요청한 오브젝트에 전체적인 지연시간을 줄이고자 하였다.

3.1 지연시간 측정

클라이언트가 요청한 오브젝트를 프록시 서버나 원격서버에서의 가져오는 지연시간을 측정하고자 포워드 캐시 구조를 사용하여 시간 측정을 하였다.

[그림1]에 클라이언트(C1)에서 요청한 것이 캐시에서 Miss가 됐을 경우 프록시 캐시(P1)에서 요청 메시지(Request Message)를 보내고, 이에 대한 서버(S1)에서 요청한 프록시 캐시에 위치를 확인하고, 프록시 캐시(P1')에게 요청한 오브젝트와 응답메시지(Response Message)를 보내게 된다.



[그림 1] 네트워크 지연시간

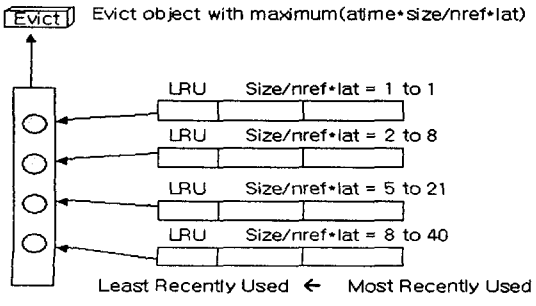
측정 되어지는 지연시간은 프록시에서 오브젝트를 요청하고자 P1 시점에서 Request Message를 보낼때의 시간을 체크하고, 서버에서의 응답으로 Response Message를 받은 P1' 시점까지의 시간을 체크하여 지연시간을 측정하게 된다.

$$Lat1(p) = P1'end - P1start \quad .....(1)$$

Lat1(P1과 P1')사이에 시간을 측정하여 교체 정책에 반영하게 되고 빈도수, 최근 사용 시간, 지연시간, 사이즈 등을 고려하여 캐시에서 교체 정책이 이루어진다.

3.2 교체 정책 알고리즘

프록시 서버와 원격 서버 사이의 지연시간을 계산해 교체 정책을 하기 위한 본 논문의 구조도는 [그림 2]와 같다.



[그림 2] LRU-SP-LAT 교체 정책 구조도

여기서 LRU(atime)은 오브젝트가 최근에 참조된 시간, size는 오브젝트의 사이즈, nref는 사용 빈도수, lat는 프록시 서버에서 요청한 오브젝트를 원격 서버에서 프록시 서버까지 가져오는데 걸리는 지연시간을 나타낸다.

이것은 제한된 프록시 캐시상에서 웹에 존재하는 오브젝트를 클라이언트가 요청을 하게 되면 실제적으로 오브젝트의 사이즈는 가변적이기 때문에 전체 지연시간은 물론 처음 요청할 때 프록시 서버에서 요청한 오브젝트를 원격 서버에서 가져오는데 있어 지연시간에 영향을 주게 된다.

[그림 2]의 구조도에서 Cost-to-Size 모델을 적용하여 측정한 각 오브젝트 사이즈, 참조빈도수, 지연시간, 최근에 참조된 시간을 측정해서 캐시내에서 우선순위 키 값을 계산하여 키 값이 큰 것을 교체하는 방식으로 이루어진다.

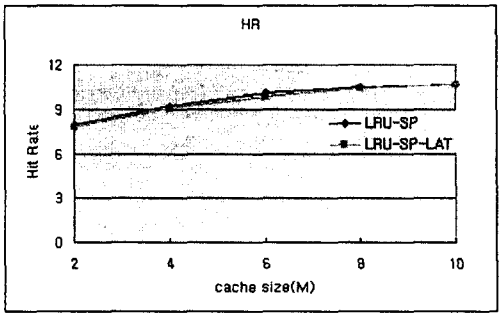
4. 성능분석

실험 환경은 Pentium III PC 2대와 윈도우 2000, 리눅스 7.1, squid 프록시 서버를 사용하여 실험을 하였다. 여기서 클라이언트에서 요청하는 오브젝트로는 NLANR(National Laboratory for Applied Network Research)에서 제공하는 트레이스 파일을 사용하였다. 본 실험에서 사용한 매개변수는 다음과 같다.

매개변수	설정값
메시지 요청 개수	17889개
메시지 요청 Byte	77M
프록시 서버	1개

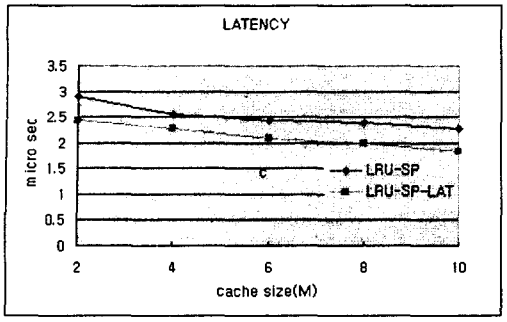
[표 1] 실험 매개 변수

위의 [표 1]에서 설정한 매개변수를 통해서 비교모델 LRU-SP 교체 정책을 실험한 결과와 제안하는 LRU-SP-LAT 교체 정책을 통한 지연시간과 히트율을 비교 실험 하였다.



[그림 3] 캐시 사이즈 변화에 따른 히트율

[그림 3]에서 캐시 사이즈가 증가함에 따른 히트율을 LRU-SP와 제안한 교체 정책 LRU-SP-LAT를 비교 실험한 결과이다. [그림 3]에서 보는 바와 같이 두 정책이 유사한 히트율을 보인다.



[그림 4] 캐시 사이즈 변화에 따른 지연시간

[그림 4]는 프록시와 서버 사이의 네트워크 지연시간이 감소함으로써 클라이언트에서의 전체적인 지연시간이 감소한 것을 알 수 있다. 클라이언트에서 오브젝트를 요청한 후, 응답이 오기까지의 지연시간을 측정한 결과 15.23%의 성능 향상을 가져왔다. 이것은 클라이언트가 요청한 오브젝트가 제안한 교체정책에 의해 프록시 캐시내에 요청한 오브젝트의 높은 히트율이 측정되었다.

5. 결론 및 향후과제

현재 기하급수적으로 증가하는 웹 사용자에 비례하여 요청을 감당하기 위해 네트워크의 대역폭을 늘리거나 고성능의 서버를 도입해야 하는데 한계가 있다. 이러한 한계점을 완화하기 위한 방안으로 본 논문에서는 프록시 캐시 교체 정책을 제안하였다. 클라이언트에서 요구하는 오브젝트를 원격서버에서 가져오지 않고 프록시 캐시에서 가져옴으로써 전체 지연시간을 줄일 수 있다. 이에 지연시간을 고려하여 응답시간을 측정한 결과 최근에 참조된 시간, 사이즈, 빈도수만을 가지고 측정한 것보다 전체 지연시간이 감소한 것을 확인 할 수 있었다.

향후 과제로는 실험 환경을 확장하여, 클라이언트와 프록시 서버의 수를 늘려서 CNP(Cache Neighbor Protocol)를 고려한 연구를 진행할 것이다.

6. 참고문헌

[1] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications" Proceedings of the IEEE Infocom '99 Conference, New York, NY , March 1999.

[2] Pei Cao, Sandy Irani, "Cost-Aware WWW Proxy Caching Algorithms" Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, Dec 1997.

[3] Kai Cheng, Yahiko Kambayashi, "LRU-SP: A Size-Adjusted and Popularity-Aware LRU Replacement Algorithm for Web Caching" Proc. 24th IEEE Computer Society International Computer Software and Applications Conference (Compsac'2000), pp.48-53, IEEE Computer Society Press, Oct. 2000.

[4] Ludmila Cherkasova, Gianfranco Ciardo, "Role of Aging, Frequency, and Size in Web Cache Replacement Policies" In Proceedings of the Sixth International Symposium on Computers and Communications (ISCC'01), Hammamet, Tunisia, July 3-5, Page(s):64-71, 2001.

[5] Dan Foygel, Dennis Strelow, "Reducing Web Latency with Hierarchical Cache-based Prefetching" 2000 International Workshop on Parallel Processing August 21 - 24, 2000 Toronto, Canada.

[6] Hohn Dilley, Martin Arlitt, "Improving Proxy Cache Performance-Analyzing Three Cache Replacement Policies" IEEE Internet Computing, Vol. 3, No. 6, pp. 44-50, November/December 1999.

[7] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol", IEEE/ACM Transactions on Networking 8 (2000), no. 3, 281-293.

[8] Shudong Jin, Azer Bestavros, "Popularity-Aware GreedyDual-Size Web Proxy Caching Algorithms" In Proceedings of the 20th International Conference on Distributed Computing Systems, Taipei, Taiwan, Republic of China ,IEEE, 2000.