계층적 프록시 캐싱을 이용한 인터넷 성능 향상

이효일, 김종현
연세대학교 전산학과
hyoil@magics.yonsei.ac.kr, jhkim@dragon.yonsei.ac.kr

Hierarchical Proxy Caching for Improving on Internet Performance

Hyoil Lee, Sooyoung Kim, Jonghyun Kim
Department of Computer Science, Yonsei University

초고속 정보통신망을 비롯한 정보 인프라의 구축이 확대되면서 인터넷을 비롯한 다양한 정보 서비스들이 활성화되고 있다. 최근 인터넷의 사용자 수가 크게 증가함에 따라 웹 서버에 걸리는 부하와 통신망의 트래픽이 급증하고 있으며, 이들은 응답시간을 지연시키는 주요 요인이 되고 있다. 이러한 문제를 해결하기 위하여 인기 있는 정보는 클라이언트에 가까이 위치한 프록시 서버에 캐싱함으로써 웹서버의 병목현상을 완화시키고, 통신망의 트래픽을 줄이며, 서비스 응답 시간을 줄일 수 있다. 또한 여러 프록시 캐쉬들에 저장된 정보들을 클라이언트들이 공유함으로써 인터넷 성능을 보다 향상시킬 수 있다. 이 논문에서는 실제 웹 트레이스를 이용한 시뮬레이션을 통하여, 프록시 캐쉬들을 계층적으로 접속한 인터넷 환경에서 캐쉬 크기에 대한 캐쉬 적중률을 분석하였다.

1. 서론

초고속 정보통신망을 비롯한 정보 인프라의 구축이 확대되면서 다양한 종류의 정보 서비스들이 활성화되고 있다. 그 중에서 가장 널리 사용되고 있는 것은 인터넷을 통한 World Wide Web(WWW; 이하 웹이라 함) 서비스이다. 최근 웹의 사용자 수가 크게 증가함에 따라 웹 서버(web server)에 걸리는 부하(load)와 통신망의 트래픽(traffic)이 급증하고 있다. 특히, 유용한 정보를 가진 웹 서버는 하루에 수십만 이상의 액세스 요구들을 받고 있으며, 그와 같은 높은 부하는 응답시간이 길어지게 하는 주요 요인이 되고 있다. 이와 같은 현상은 앞으로 인터넷 서비스가 더욱 활성화되고 사용자가 계속 늘어날 것으로 예상됨에 따라 시급히 해결해야 할 문제로 대두되고 있다. 본 논문에서는 실제 웹 트레이스를 이용한 시뮬레이션을 이용하여, 프록시 캐쉬들이 계층적으로 접속된 인터넷 환경에서 캐쉬 크기에 따른 캐쉬 적중률의 변화를 분석하였다.

2. 관련 연구

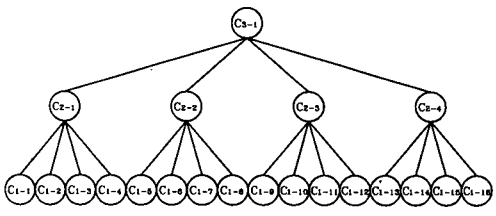
인터넷 정보를 액세스하는 속도를 향상시키는 방법으로서, 웹 서버와 통신망 트래픽을 줄여 서비스의 처리 속도를 높이는 웹 캐싱(web caching)에 관한 연구가 활발히 진행되고 있다. Harvest [1] 와 Squid [2]는 사용자 컴퓨터와 웹 서버 사이에 존재하는 여러 프록시 서버들에 캐쉬를 위치시킴으로써 계층적 캐쉬(hierarchical cache)를 구성하였다. 각 캐싱 서버는 상호 협력하면서, 원하는 문서를 가진 가장 가까운 노드로부터 전송 받도록 하고 있다. CRISP [3]는 프록시 서버들에 있는 캐쉬들에 어떤 문서들이 저장되어 있는지를 가리키는 정보를 가진 통합 디렉토리(global directory)를 이용하여 네트워크 트래픽을 줄였다. Gadde et al.[4]는 계층적 프록시 캐쉬 구조에서 푸쉬(Push) 알고리즘을 이용한 성능 향상을 분석하였다. E. P. Markatos[5]는 단일 웹서버에서 캐쉬 크기에 따른 캐쉬 적중률을 측정하였다. 대규모의 계층적 분산 캐싱을 이용하여 응답 시간과 트래픽을 줄이는 노력

외에도 웹 서버의 주기억장치를 이용하여 캐싱하는 방법[6, 7]과 캐싱 교체 알고리즘에 대한 연구들[8, 9]도 활발히 진행되고 있다. 최근 정보통신망이 확대되면서 여러 단계의 계층적 프록시 캐싱 환경이 구성되고 있다. 본 연구에서는 다른 연구들[1,2,4]에서도 고려된 3-레벨 4진 트리(3-level 4-ary tree) 구조의 계층적 프록시 캐싱 환경에서 캐쉬 크기에 따른 적중률의 변화를 분석하였다.

3. 시뮬레이터의 구성과 작업 부하

2.1. 시뮬레이터의 구성

본 연구에서는 시뮬레이터를 AweSim v. 2.0 [10]을 이용하여 개발하였다. 계층적 프록시 캐쉬 구조는 [그림 1]과 같은 3-레벨 4진 트리 형태를 고려하였다. 각 프록시 서버에는 4개씩의 하위 프록시 캐쉬들이 접속되며, 하위 프록시 캐쉬에 대한 디렉토리를 저장하고 있다. 그리고 모든 클라이언트들은 최하위 레벨(레벨 1)에 연결되는 것으로 가정하였다. 클라이언트에서 웹 문서를 요구하게되면, 클라이언트가 직접 연결된 최하위 레벨의 프록시 캐쉬를 검색하게 된다. 캐쉬 적중이 되면 즉시 서비스를 하고, 그렇지 않으면 상위 프록시 서버로 요구를 보낸다. 상위 프록시 서버는 요구된 문서가 자신의 캐쉬 혹은 하부에 접속된 다른 프록시 서버에 있는지 디렉토리를 통하여 검사한다. 만약 하부 서버에 있다면 요구를 그 서버로 보낸다. 만약 하위 프록시 캐쉬들에도 그 웹 문서가 없다면 다시 상위 프록시 캐쉬로 요구를 보낸다. 최상위 캐쉬에서도 적중되지 않는다면 그 웹 문서가 있는 원래의 웹 서버로 요청하여 서비스를 받게 된다.



[그림 1] 계층적 프록시 캐쉬 구조

2.2 작업부하

웹 트레이스의 특성은 웹 캐싱과 시뮬레이션 결과에 큰 영향을 미친다. 본 연구에서는 UCB(University of California, Berkely)에서 수집된 웹 트레이스 [11]를 이용하여 시뮬레이션 하였다. [표 1]의 첫 번째 트레이스는 시뮬레이터 검증을 위해 사용하였으며, 실제 연구에서는 두 번째 트레이스를 주로 사용하였다. 이 트레이스는 40시간 동안 수집한 170만개의 요구들이 포함되어 있다.

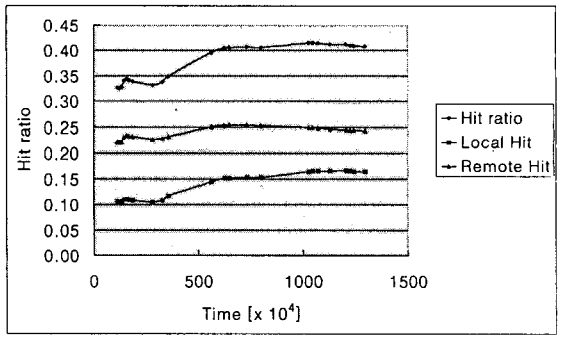
이 트레이스를 분석한 결과, 웹 문서 요구 발생 시간은 지수분포를 가지며, 전체 참조된 문서의 10% 정도가 전체 요구의 50%를 차지하였다. 또한 평균 전송 바이트는 약 6~8KB 이며, 중수(Median)는 약 2KB 이다. 다른 웹 트레이스에 대한 연구들에서 분석한 결과들도 유사하였다[12, 13].

[표 1] UCB 트레이스들

트레이스	수집일시	수집시간	총 요구수	총 전송 바이트
UCB(1)	1996.11.17	4시간	61,354 개	640,238,759
UCB(2)	1996.11. 6	40시간	1,703,835 개	12,573,824,026

4. 시뮬레이션 결과 및 분석

본 연구에서는 실제 웹 트레이스를 이용한 시뮬레이션을 이용하여 캐쉬 크기에 따른 캐쉬 적중률을 비교하였다. [그림 2]는 1500만 시뮬레이션 시간 동안의 캐쉬 적중률의 추이를 보여주고 있다. 이 분석에서 각 프록시 캐쉬의 용량은 32MB로 하였다.

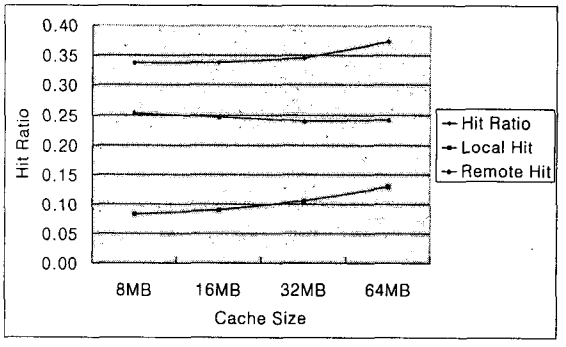


[그림 2] 시간 경과에 따른 캐쉬 적중률
(Cache size = 32 MB)

시뮬레이션 결과에 따르면, 캐쉬 적중률은 0.32~0.42 정도를 보였으며, 그 중 0.12~0.16은 지역 캐쉬 적중률(local cache hit ratio)이다. [그림 2]에서 500만 시뮬레이션 시간 이전의 낮은 적중률은 캐쉬의 워밍

업 시간(warming-up time)과 트레이스 특성에 기인한 것이다.

[그림 3]은 캐쉬 크기에 따른 캐쉬 적중률을 보여주고 있다. 캐쉬 크기가 증가함에 따라 지역 캐쉬 적중률은 0.08에서 0.13으로 62% 향상되지만, 원격 캐쉬 적중률(remote cache hit ratio)은 거의 변함이 없었다. 전체 캐쉬 적중률은 0.33에서 0.37 정도로 11% 높아졌는데, 이러한 향상은 주로 지역 캐쉬 적중률 향상에 기인한 것이다.



[그림 3] 캐쉬 크기에 따른 적중률

5. 결론

인터넷 사용자의 수가 급속히 늘어남에 따라, 웹 서버의 부하나 통신망의 트래픽도 급증하고 있다. 이들은 서비스 응답시간이 길어지게 하는 주요 요인이 되고 있다. 웹 캐싱은 이런 문제점을 해결하여, 서비스를 개선시키는 좋은 방안이다. 본 연구에서는 계층적 프록시 캐싱 환경에서 캐쉬 크기에 따른 적중률의 변화를 시뮬레이션을 통하여 분석하였다. 시뮬레이션 결과에 따르면, 캐쉬의 크기가 커짐에 따라 지역 캐쉬의 적중률이 크게 높아져서 성능 향상에 도움이 되는 것을 확인하였다. 그러나 원격 캐쉬는 여러 하부 프록시 서버들에 의하여 공유되기 때문에 캐쉬 크기에 별 영향을 받지 않는 것으로 나타났다.

참고문헌

[1] A. Chankhunthod and M. F. Schwartz, " A hierarchical Internet object cache," In Proceedings of USENIX 1996 Annual Technical Conference, January 1996.

[2] D. Wessels et al., "Squid Internet object cache," <URL: http://squid.nlanr.net/>

[3] S. Gadde, M. Rabinovich, and J. Chase, "Reduce, Reuse, Recycle:An approach to building large Internet caches," In Proceedings of 6th Work shop on Hot Topics in Operating Systems (HOTOS-VI), May 1997

[4] S. Gadde, J. Chase, and M. Rabinovich, "Directory structures for scalable Internet caches," Technical Report CS-1997-18, Duke University Department of Computer Science, November 1997

[5] E. P. Markatos, "Main Memory Caching of Web Documents," Computer Networks and ISDN Systems, 1996

[6] I. Tatarinov, V. Sooviev, and A. Rousskov, "Static caching in Web servers," In Proceedings of The Sixth International Conference on Computer Comunication and Networks, 1997

[7] J. Yang, W. Wang, and R.Muntz, "Dynamic Web caching," ncstrl.ucla_cs/980042, Department of Computer Science, University of California, LA, November 1998

[8] I. Tatarinov, "Performace Analysis of Cache Policies for Web Servers," In Proceedings of 9th International Conference on Computing and Information, ICCI'98 Winnipeg, June 1998

[9] B. Krishnamurthy and C. E. Wills, "Proxy cache coherency and replacement - Towards a more complete picture," 19th IEEE International Conference on Distributed Computing System, may 1999

[10] A. Alan B. Pritsker, J. J. O'Reilly, and D. K. LaVal, Simulation with Visual Slam and AweSim

[11] UC Berkeley Home IP Web Traces, <http://ita.ee.lbl.gov/html/contrib/UCB.home-IP-HTTP.html>

[12] M. F. Arlitt and C. L. Willamson, "Internet web servers: Workload characterization and Performance Implications", IEEE/ACM Trasactions on Networking, Vol. 5, NO. 5. October 1997

[13] G. Abdula, E. A. Fox, M. Abrams, and S. Williams, "WWW Proxy traffic characterization with Application to caching", TR-97-03, Virginia Polytechnic Inst. and State University