웹에서 프락시 서버의 유용성 분석

장우석°, 김경환*, 최용준*, 김종근*

*영남대학교 컴퓨터공학과

An Analysis of Proxy effect in WWW

WooSuck Chang°, KyungHwan Kim*, YongJun Choi*, ChongGun Kim*

*Department of Computer Engineering, Yeungnam University

요 약

웹 사용자의 수가 증가하면서 웹서버에 대한 접속 횟수가 급격히 증가하고 있다. 이것은 망의 부하와 서버의 부하를 증가시켜 웹 사용자가 원하는 정보의 전달을 지연시킨다. 프락시 서버는 이러한 문제를 해결하는 방법으로 서버 파일의 복사본을 클라이언트 가까이 위치시켜 망과 서버의 부하를 감소시킨다. 그러나 웹사용에서 프락시 서버의 도입효과에 대한 구체적인 연구결과가 없으므로 시스템 관리자가 프락시 서버의 도입여부를 결정하기가 힘들다. 본 논문에서는 웹사용 환경에서 프락시 서버의 성능을 평가할 수 있는 모델을 제시하고 시뮬레이션을 통해 이를 분석하고 평가한다.

1. 서론

웹 사용자의 수가 증가하면서 웹서버에 대한 접속 횟수가 급격히 증가하고 있다. 이것은 망의 부하와 서버의 부하를 증가시켜 웹 사용자에게 원하는 정보의 전달을 지연시킨다.

증가하는 사용자의 접속 요구를 수용하기 위해서는 망의 대역폭을 늘리고 서버 용량을 확장해야 하지만 이는 많은 비용이 요구된다. 보다 간단한 방법으로 프락시 캐싱(Proxy caching)기법이 있다. 사용자의 접속정보를 프락시 서버에 일시 보관하여 여러 사용자가 공유함으로서 망의 부하와 웹서버의 부하를 감소시키면서 사용자에게 빠르게 정보를 전달할 수 있다. 전자의 방법보다 비용 및 유지보수 측면에서 장점이 있어 널리 사용된다.[1][2]

프락시 캐시를 도입하기 위해서는 추가적인 시스템이 필요하고 프락시 캐시 도입에 따른 성능향상이나 도입 기준에 대한 연구가 필요하다.

본 논문에서는 웹에서 프락시 서버의 유용성을 분석하기 위해서 일반적인 인터넷 환경인 서버-WAN-프락시 서버-LAN-클라이언트 환경에서 프락시 캐싱의 성능을 시뮬레이션하여 프락시 캐싱의 성능을 평가하고 프락시 서버의 도입을 판단하기 위한 척도를 제시한다.

2. 웹 캐싱(Web caching)

캐싱(Caching)은 컴퓨터 메모리 시스템(computer memory system), 운영체제하에서의 가상 메모리 관리(virtual memory management), 파일 시스템(file system) 그리고, 데이터베이스(database)에서 오래 동안 연구된 분야이다. 웹에서의 캐싱은 DNS, Gopher, FTP와 같은 외부 네트웍 서비스의 성능향상을 위해 사용되었는데, 오늘날 웹 캐싱(web caching) 연구의 궁극적인 목적도 성능향상에 있다.

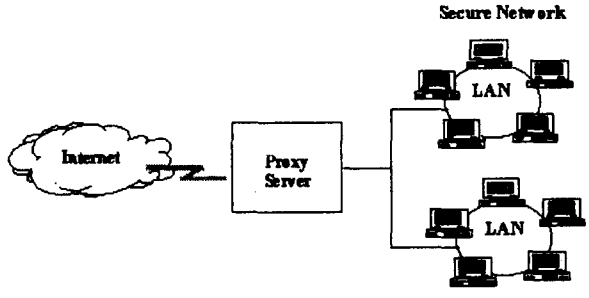
이 논문은 정보통신부의 대학 소프트웨어 연구센터 지원사업중 영남대학교 주관의 "차세대 인터넷/인트라넷의 QoS 보장형 네트워킹 시스템 개발" 연구과제의 지원으로 연구되었습니다.

2.1 웹 캐싱의 형태

웹 캐싱은 HTML문서와 같은 웹 객체(web object)의 일시적인 저장으로, 사용자가 요청한 객체가 캐시에 있는 경우 해당 서버에 접속하지 않고 캐시에 저장된 자료를 사용자에게 전송한다. 이것은 대역폭 소비(bandwidth consumption)의 절감, 서버 부하(server load) 감소, 그리고 지연시간(latency)의 감소라는 세 가지 중요한 장점을 가진다.

웹에서 구현되는 캐싱의 형태는 크게 다음의 세 가지로 구분될 수 있다.

첫 번째, 캐싱은 클라이언트 어플리케이션에서 실행될 수 있다. 웹브라우저에 가상적으로 구현되는 캐시가 여기에 해당된다. 이것은 웹 브라우징의 속도를 높일 수가 있지만 단지 한 사용자만을 위한 것으로 효율성은 높지 않다.



[그림 1] 프락시 서버

두 번째, 클라이언트와 서버의 사이에 그림1과 같이 프락시 서버를 두는 것이다. 대부분의 프락시 캐시가 이와 같은 형태를 이룬다. 프락시 캐시는 웹서버로부터 저장된 웹 페이지들을 다수의 클라이언트에게 제공한다.[6]

세 번째, 좀 더 나은 성능을 위해 많은 프락시 캐시가 분산처리의 형태를 이루는 것이다. 이 구조는 사용자가 서버에게 요구하고 있는 내용이 프락시 캐시 안에 없을 경우 서버로 직접 접속하는 대신에 이웃하는 프락시 캐시에 웹 문서를 요구하는 방식이다.

첫 번째를 제외한 프락시 캐시는 여러 클라이언트에게 서비스를 제공한다. 프락시 캐시가 웹 문서를 요청 받았을 때 메모리, 디스크 혹은 기타 다른 곳의 문서로 응답이 가능하다면 실제 서버로의 접속이 없이 클라이언트에게 응답한다. 만약 캐시에서의 응답이 불가능하다면 프락시 캐시는 직접 서버로 접속해서 문서를 가지고 온다. 프락시 캐시에 저장되어야 하는 문서의 양이 실제 저장가능한 양을 초과한다면 프락시 캐시는 캐시에서 문서를 대체(replace)해야 할 필요가 있다. 일반적으로 캐시 교체정책(cache replacement algorithm)은 미래에 요청되는 모든 문서 중에서 캐시에 의해 성공적으로 서비스되는 문서의 비(Hit ratio)를 최대화하려고 한다. [7][8]

2.2 캐시 성능 평가 파라메터

캐시 성능은 적중률(hit ratio:HR)과 가중치 적중률(weight hit ratio:WHR)로 평가한다.[4]

캐시 적중률은 사용자의 전체 요청 수에서, 캐시에 파일이 존재하여 적중된 수의 비율을 의미한다. 웹 캐시에서 캐시 적중률(HR)은 서버에 도착하는 사용자 요청의 감소 비율로 간주할 수 있다. 캐시 적중률은 식(1), (2)와 같이 계산할 수 있다.[4] 여기서 r_i는 캐시에 파일i의 요청 횟수를나타내며 n은 캐시 내의 전체 자료 수, i는 1과 n 사이 자연수이다.

$$HR = \frac{\sum_{i=1}^{n}(r_i - 1)}{\sum_{i=1}^{n} r_i} \quad \cdots\cdots (1)$$

웹 캐시는 고정된 크기의 페이지 단위로 캐시를 하는 것이 아니라 가변 길이의 파일 단위로 캐시를 하기 때문에 단순히 캐시 적중률만으로 정확한 성능 평가가 어렵다. 이를 위해 가중치 적중률을 이용한다. 가중치 적중률(WHR)은 사용자가 요청한 전체 바이트량에서, 캐시에서 적중된 바이트량의 비율을 나타낸다. 웹 캐시에서 가중치 적중률은 캐시를 사용함으로써 얻어지는 망 통신량 감소를 나타내는데 사용한다. s_i는 서버에 있는 파일 i의 크기를 나타낸다.

$$WHR = \frac{\sum_{i=1}^{n} s_i(r_i - 1)}{\sum_{i=1}^{n} s_i r_i} \quad \cdots\cdots (2)$$

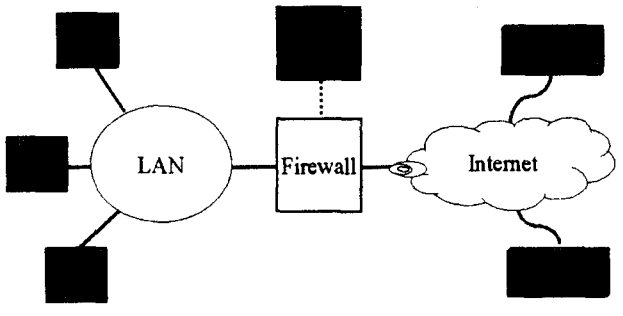
3. 모델링 및 성능평가

3.1 클라이언트-프락시 서버-웹서버 통신형태

네트워크 환경은 그림 2과 같이 LAN의 대역폭이 WAN의 대역폭보다 큰 형태로 학교나 기업 등에서 많이 사용되고 있는 모델을 기반으로 한다. 이 경우 프락시 서버는 방화벽 기능을 같이 하기도 한다.

3.2 클라이언트-프락시 서버-웹서버 통신 프로시져

시뮬레이션을 위한 클라이언트-프락시 서버-웹서버 사이에서 HTTP 요청(Request)에 따른 이벤트의 흐름을 시간값으로 나타낸 것이 그림3과 같다.



[그림2] 네트워크 환경

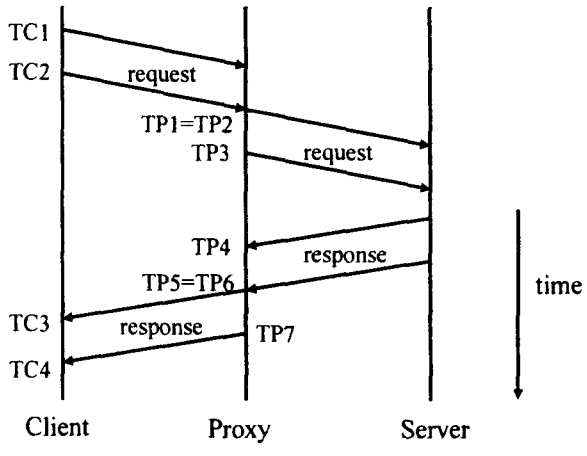
그림3에 나오는 각 파라미터가 나타내는 의미는 다음과 같다.

- TC1 : 클라이언트가 요청메시지로 첫 번째 바이트를 보내는 시간
- TC2 : 클라이언트가 요청메시지로 마지막 바이트를 보내는 시간
- TP1 : 프락시 서버가 요청메시지의 마지막 바이트를 받는 시간
- TP2 : 프락시 서버가 웹서버로 접속하는 시간
- TP3 : 프락시 서버가 웹서버로 요청메시지의 마지막 바이트를 보내는 시간
- TP4 : 웹서버에서 프락시 서버가 응답의 첫 번째 바이트를 받는 시간
- TP5 : 프락시 서버가 클라이언트로 응답의 첫 번째 바이트를 보내는 시간
- TP6 : 웹서버에서 프락시 서버가 응답의 마지막 바이트를 받는 시간
- TC3 : 클라이언트가 프락시 서버에서 응답의 첫 번째 바이트를 받는 시간
- TP7 : 프락시 서버가 클라이언트로 응답의 마지막 바이트를 보내는 시간
- TC4 : 클라이언트가 프락시 서버에서 응답의 마지막 바이트를 받는 시간

그림3의 파라미터를 이용하여 다음의 계산 값이 나올수 있다.

- 클라이언트와 프락시 서버사이의 전송시간
  - LCi=TC3-TC2-(TP6-TP1)
- 클라이언트와 프락시 서버사이의 대역폭
  - BCi=Si/(TC4-TC3)
- 프락시 서버와 웹서버사이의 전송시간
  - LPi=TP4-TP3
- 프락시 서버와 웹서버사이의 대역폭
  - BPi=Si/(TP5-TP4)

여기서 Si는 요청하는 웹페이지의 크기이다.



[그림3] HTTP 요청에 따른 Event의 흐름

본 시뮬레이션 모델에서는 LAN 환경에서 LAN/WAN의 전송시간 차(LPi와 BPi를 변화), 프락시 캐시에 대한 클라이언트의 요구발생빈도, 클라이언트 요청에 대한 응답 자료의 바이트 양에 따른 프락시 캐시의 성능 변화를 비교 분석한다.

3.3 시뮬레이션 모델링

시뮬레이션을 위한 기본 모델은 WAN(T1)의 전송시간을 1, LAN(10Mbps)의 전송시간을 0.2로 가정하고 WAN의 속도를 점점 악화시켜 LAN과 WAN의 전송시간차에 따른 적정적중률을 찾아보았다. 적정적중률은 프락시 캐시를 도입할 때, 캐

시가 없을 때보다 자료의 평균전송시간이 적게 걸리는 최소 적중률이다.

적정적중률은 캐시 적중률을 20%에서 60%까지 5%씩 변화시켜 나오는 전송시간을 비교해서 적정 적중률을 계산했다.

또한 요청 바이트(Request Byte)와 응답 바이트(Response Byte)의 비는 실험대상으로 분석한 nety.yeungnam.ac.kr 웹서버의 로그파일을 분석한 결과 그 양의 비가 1대1에서 1대50까지 분포하고 있었고, 평균은 1대25로 정도인 것을 알 수 있었다. 시뮬레이션에서는 1대1, 1대25, 1대50 세 가지 경우에 따른 적정 적중률을 계산했다.

그리고 프락시 캐시에 동시 접속 가능한 클라이언트 수를 1000명으로 하고 클라이언트의 프락시 캐시에 대한 접근 시간을 5, 10, 20 세 가지 지수분포로 발생해서 프락시 캐시의 부하에 따른 성능을 분석했다.

성능분석을 위한 시뮬레이터로는 GPSS/H를 이용했다. 아래는 시뮬레이션을 위한 소스코드 일부분이다.

```
         SIMULATE
*
*        LAN ENVIRONMENT WITH PROXY
*
*        Request와 Response의 전송시간비          1:25
*        Lan과 Wan 사이의 전송시간비              1:5
*
*        Define Ampervariables
*
         INTEGER          &P
*
*        Storage Declaration Segment
*
         STORAGE                    S(PROXY),1000
*
         GENERATE         RVEXPO(1,5.0)
         QUEUE            LAN
*
*        transfer time from client to proxy
*
         ADVANCE                    0.8, 0.2
*
*        Go to Proxy Server
*
         BLET             &P=5
         TRANSFER         ,PROSER
*
TOCLI    TRANSFER         .45,MIS,HIT
```

<GPSS/H 소스 코드 일부분>

3.4 시뮬레이션 결과 및 분석

Hit Ratio	평균전송시간
20%	8.56(A1)
25%	7.93(A2)
30%	7.35(A3)
35%	6.80(A4)
40%	6.27(A5)
45%	5.76(A6)
50%	5.30(A7)
55%	4.85(A8)
60%	4.40(A9)
no proxy	4.91(AA)
적정적중률	**54.26%**

[표 1] 평균전송시간

(클라이언트 평균접근시간 : 10, 요청 대 응답비 : 1 대 25)

표 1은 랜 환경에서 클라이언트의 평균접근시간이 10이고 요구에 대한 응답의 양에 대한 비가 1대25일 때 요청자료의 평균전송시간을 나타낸다. 여기서 캐시적중률을 20%에서 60%까지 5%씩 변화해가며 나온 전송시간과 프락시 캐시가 없는 경우에 나온 전송시간을 비교해서 적정 적중률을 계산했다. 적정적중률은 프락시 캐시가 없는 경우 평균전송시간(AA), 프락시 캐시가 있을 때 각 적중률에 따라 구한 평균전송시간에서 AA보다 큰 값들 중 제일 작은 값(A7)과 AA보다 작은 값들 중 제일 큰 값(A8)으로 식(3)을 이용해 구할 수 있다.

$$\text{적정적중률} = A7\text{의적중률} + \frac{A7-AA}{A7-A8} \times 5\% \quad \cdots (3)$$

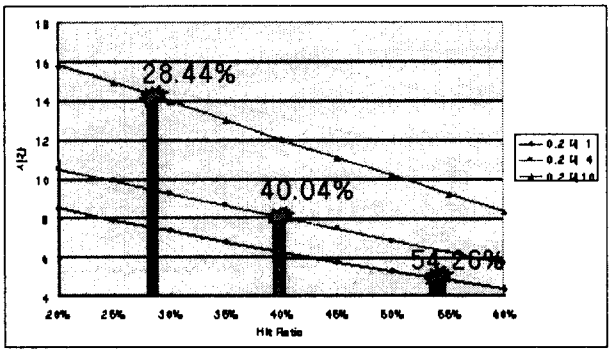
$$= 50\% + \frac{5.30-4.91}{5.30-4.85} \times 5\% = 54.26\%$$

그림 4는 WAN의 전송시간 변화에 따른 적정적중률 변화를 보여주고 있다. WAN의 전송시간과 LAN의 전송시간의 차가 클수록 적정적중률이 작게 나오고 이는 전송시간차가 클수록 캐시 적중률이 낮아도 프락시 캐싱의 효과가 있다는 것을 의미한다.

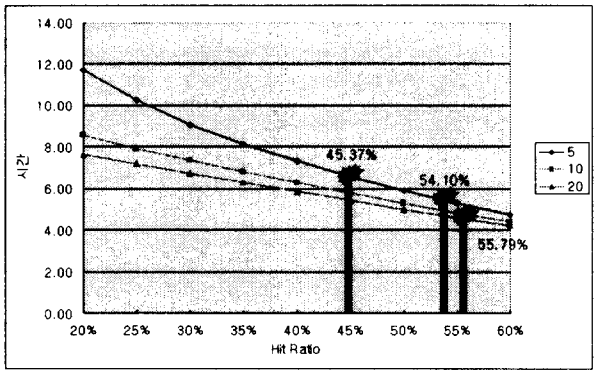
그림 5는 랜환경에서 프락시 캐시로 접근하는 클라이언트 수에 따른 적정적중률 변화를 보여주고 있다. 그림에 보는 바와 같이 클라이언트가 프락시

캐시로의 접근간격이 짧을 때 적정적중률은 낮게 나타났다. 이것은 프락시 캐시로 접속하는 사용자가 많을 때 효과가 있음을 보여주는 것이다.

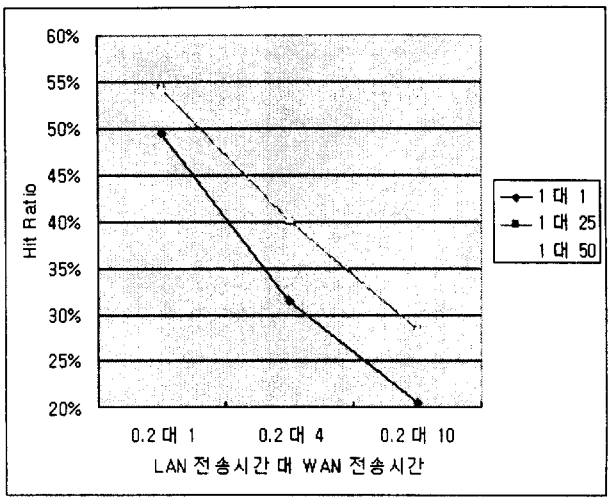
그림 6은 랜환경에서 요청 바이트에 대한 응답 바이트의 비에 따른 적정적중률 변화를 보여주고 있다. 두 환경 모두 요청 대 응답의 비가 작을 때, 즉 사용자가 용량이 큰 그림파일이나 멀티미디어 자료를 요구하는 것보다는 용량이 작은 텍스트 파일을 많이 요구할 때 효과가 있는 것을 알 수 있다. 한편 요청 대 응답비가 1대25과 1대50 일 때의 적정적중률의 차이는 거의 없는 것으로 볼 수 있다.



[그림4] 랜환경에서의 평균전송시간
(클라이언트 평균접근시간 : 10, 요청 대 응답비 : 1 대 25)



[그림5] 랜환경에서 접근시간에 따른 적정적중률
(요청 대 응답비 : 1:25, LAN 대 WAN = 1:10)



[그림6] 랜환경에서 전송시간 비에 따른 적정적중률 비교
(클라이언트 도착간격 10)

시뮬레이션 결과 LAN 환경에서 LAN/WAN의 전송시간차이가 크고 동시에 이용하는 사용자가 많고 용량이 큰 파일보다는 작을 파일들을 많이 요구할 때 프락시 서버의 도입이 효과가 있는 것으로 판단된다.

4 결론 및 향후연구과제

최근 인터넷 사용자가 급격히 증가함으로써 서버와 망의 과부하 현상이 발생하고, 전송요구에 대한 전송지연시간이 증가하게 되었다. 웹 캐시는 서버의 부하, 망 통신량을 감소시키는 한가지 방안으로 가장 널리 사용된다. 웹 캐시의 가장 일반적인 형태가 프락시 캐시이다. 프락시 캐시는 망의 여러 사용자가 하나의 서버를 공유하여 사용함으로써 캐시의 성능 증가, 디스크 공간을 절약할 수 있다. 본 논문에서는 LAN 환경에서 LAN/WAN의 전송시간차이와 전송파일의 크기, 사용자수에 따른 프락시 캐시의 적정적중률을 비교 분석함으로써 프락시 서버의 유용성을 분석해 보았다. 그 결과 LAN/WAN의 전송시간차이가 크고 동시에 접속하는 사용자가 많고, 요청하는 자료가 텍스트나 작은 그림과 같은 용량이 작은 파일일 때 프락시 서버의 도입이 효과가 있는 것을 알 수 있었다.

본 논문은 프락시 서버의 도입을 결정하는 척도를 제시했다는 점에서 그 의미를 가진다. 향후 연

구 과제로는 무선환경, 인공위성통신, 케이블모뎀 등을 대상으로 한 프락시 캐시를 시뮬레이션 해보고 수학적으로도 분석해야 할 것이다.

참고문헌

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

[2] Mari Korkea-aho, "Scalability in Distributed Multimedia System", Master Thesis, Helsinki University of Technology, Nov. 1995.

[3] Peter B. Danzig, Richard S. Hall and Michael F. Schwartz, "A Case for Caching File Objects Inside Internetwork", Tech. Rep. CU-CS-642-93, University of Colorado Department of Computer Science, March 1993.

[4] 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 Distributed and Networked Environments, jun. 1995.

[5] Robert Buff, Arthur Goldberg, and Ilya Pevzner. "Rapid, Trace-Driven Simulation of the Performance of Web Caching Proxies", Submitted to WISP'98, March 1998.

[6] Anawat Chankhunthod, Peter B. Danzig, Chuck Neerdaels, Micheal F. Schwartz, and Kurt J. Worrell. "A Hierarchical Internet Object Cache", In Proceedings of the USENIX Technical Conference, San Diego, CA, January 1996.

[7] Junho Shim, Peter Scheuermann, and Rad다 Vingralek. "A Unified Algorithm for Cache Replacement and Consistency in Web Proxy Servers", In Preceedings of the Workshop on the Web and Data Bases(WebDB98), 1998. Extended version will appear in Lecture Notes in Computer Science, Springer Verlag, 1998.

[8] Junho Shim, Peter Scheuermann, and Rad다 Vingralek. "Proxy Cache Design: Algorithms, Implementation and Performance", IEEE Transaction on Knowledge and Data Engineering, 1999.

[9] Apache Group. Apache HTTP server documentation. http://www.apache.org/, 1999

[10] World Wide Web Consortium. Jigsaw HTTP server documentation. http://www.w3c.org/Jigsaw/, 1999

[11] Microsoft Corporation. Microsoft Proxy Server. http://www.microsoft.com/proxy/, 1999