웹 캐싱 지역 프로토콜에서 요청에 대한 효과적인 라우팅

이용찬, 김성천
서강대학교 컴퓨터학과

Efficient Request Routing in Web Caching Neighborhood Protocol

Yongchan Lee, Sungchun Kim
Dept. of Computer Science, Sogang University

요 약

월드 와이드 웹(WWW)서비스는 상당한 지연이 발생할 것이라고 예상하는 수준까지 성장해 왔다. 이런 이유로 웹 지연시간을 줄이기 위한 기술들이 중요하게 되었다. 프록시 서버 캐시는 느린 응답시간과 네트워크 정체와 같은 문제들을 해결하기 위해서 제안되었는데 이런 프록시 서버들을 계층적으로 구성하는 방법에 대한 많은 연구가 있었다. 하지만 이런 일반적인 웹캐싱 계층 기법들은 프록시 서버들간의 통신 경로에 있어서 유연성을 제공하지 못하는 정적 구조의 기법이다. 이런 정적인 기법에서의 단점을 해결하기 위해서 캐싱 지역 프로토콜(Caching Neighborhood Protocol)이 제안되였고 이 CNP는 요청들을 협력적으로 처리하는 일련의 프록시 서버의 집합이 매 단일 요청에 대해서 바뀔 수 있고 이에 따라서 요청 처리 경로가 다양해 질 수 있기 때문에 정적 구조 기법과는 반대인 동적 구조 기법이다. 하지만 이러한 동적 웹 캐싱 계층 구조인 CNP에서는 각 웹서버를 대표하는 캐싱 대리인(Caching Representatives)에 대한 웹문서의 링크참조정보와 부하정보를 고려하지 않았기 때문에 사용자에게 빠른 응답시간을 제공하지 못하는 단점이 있다. 본 논문에서는 최근에 제안되었던 프록시 서버들을 동적 웹캐싱 계층으로 구성한 CNP와 새로 추가된 C-Rep의 상태정보를 이용해서 새로운 요청 라우팅 기법을 제안하였다. 제안한 기법은 이런 C-Rep의 상태정보를 이용하여 응답시간의 향상과 C-Rep의 부하를 분산시킬 수 있었다.

제 1 장 서 론

전세계적으로 인터넷 사용에 대한 체증이 가시화 되고 있다. 전송속도를 높이기 위해 네트워크 회선 증설이나 설계에 대한 최적화 외에 적은 비용으로 문제점을 해결할 수 있는 대안은 웹캐싱(Web caching)이 될 수 있다.

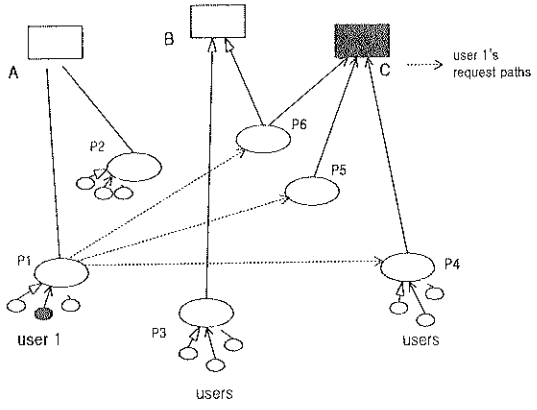
여러 개의 프록시 서버(proxy server)가 있을 때 웹캐싱의 성능을 높이기 위해서 프록시 서버들을 계층적으로 구성하는 것에 초점을 맞춘 연구들이 있었다[1][2]. 하지만 이런 대부분의 방법들은 고정된 프록시 서버들의 구성에 의해 처리되지 않은 모든 요청들의 전송에 있어서 충분한 유연성을 제공하지 못하는 정적 웹캐싱 계층(static Web caching hierarchies) 구조를 가지고 있다. 이에 대해 요청들을 서로간에 협력적으로 처리하는 프록시 서버들의 집합(set)이 매 단일 요청 때마다 변할 수 있고 그러한 요청들을 처리하는 경로가 다양할 수 있는 동적 웹캐싱 계층(dynamic Web caching hierarchies)구조인 캐싱지역 프로토콜(Caching Neighborhood Protocol : CNP)이 제안되었다[3].

본 논문에서는 동적 웹캐싱 구조의 CNP에서 기인한 비효율적인 요청전송의 문제점의 개선을 통해 비교적 더 적은 단계의 레벨 계층을 거치는 것으로 네트워크 트래픽과 응답시간(response time)을 줄이는데 있다. 이와 동시에 프록시 서버의 부하를 고려함으로써 프록시 서버들의 부하 불균형으로 야기되는 문제점을 개선하고자 한다. 웹 문서간의 링크참조에 관한 정보와 프록시 서버의 부하에 대한 정보를 토대로 요청에 대해 개선된 라우팅 기법을 이용하여 시뮬레이션을 통한 기존의 기법과의 비교를 통해서 평균응답시간을 약 8-14% 감소시킬 수 있었다.

제 2 장 CNP에서 요청에 대한 효과적인 라우팅

캐싱 지역에 대한 기본적인 개념은 [그림 1]에서 볼 수 있듯이 A, B 그리고 C로 이루어진 세 개의 origin 서버가 있고 각각의 origin 서버는 그 자신의 캐싱 지역을 가지고 있다. 승

인된 캐싱 지역의 C-Rep(Caching Representative)로부터 가져온 문서는 그 C-Rep를 대표하는 origin 서버로부터 직접적으로 가져왔다는 것으로 간주될 수 있다는 것을 의미한다.



[그림 1] Origin 서버 C의 문서를 요청한 경우

따라서 사용자 1 에서 origin 서버 C의 문서를 요청한다면 캐싱 지역 C에서 CSP(Client Side Proxy Server)인 P1에 비교적 근접한 P4, P5, P6의 C의 C-Rep들 중 다음의 요청 전송 테이블에 의해 선택된다.

요청 전송 테이블

[그림 2]는 각 프록시 서버에서 가지고 있는 제안된 요청 전송 테이블의 구조의 한 예를 보이고 있다.

For	Next proxy	Reference	Load
A	direct	...	...
B	P3	...	...
C	P4	a/e/.., c/..	7%
	P5	b/.., ...	-14%
	P6	a/d/f/..	-2%

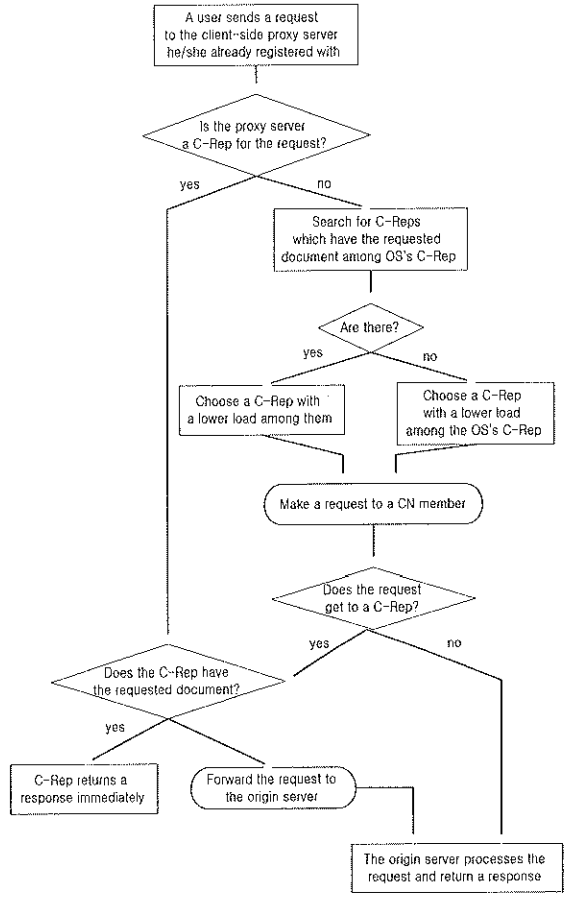
[그림 2] 요청 전송 테이블 구조

처음의 For 필드는 각 목적지의 origin 서버를 나타내며 다음의 프록시(next proxy) 필드는 각 origin 서버를 대표하는 C-Rep를 나타낸다. 다음의 필드는 자주 참조되는 문서에 대한 링크참조(reference)필드인데 각각의 C-Rep에서 자주 참조되고 있는 웹 문서간의 링크참조에 관한 정보를 나타낸다. 링크참조정보 다음에는 각 C-Rep의 부하상태를 나타내는 부하(load)필드가 존재한다. 각 C-Rep의 가중치 값과 실제적으로 연결된 상태를 비교해서 부하상태를 판단한다.

프록시 서버에서의 라우팅 결정

[그림 3]은 CSP의 요청에 대한 전송 단계를 설명한다.

CSP가 요청을 받았을 때 일단 자신이 요청된 문서를 소유하고 있는 origin 서버에 대한 C-Rep인지를 확인할 필요가 있다. CSP가 요청에 해당하는 origin 서버의 C-Rep일 경우 그리고 그 요청된 문서를 가지고 있다면 그 문서를 요청한 사용자에게 보내준다. 만일 가지고 있지 않다면 C-Rep는 그 문서를 가져오기 위해서 origin 서버에게 요청을 해야 하고 받은 문서를 결국 요청한 사용자에게 보내고 그 문서를 캐시한다.



[그림 3] CSP의 요청 전송 단계

CSP가 C-Rep가 아닐 경우 요청을 처리할 수 있는 C-Rep라고 생각하는 다른 프록시 서버로 전송해야 한다. 일련의 프록시 서버 후보자들의 집합을 포함하고 있는 origin 서버의 캐싱 지역 정보를 얻기 위해서 요청한 origin 서버에 해당하는 요청 전송 테이블을 검색하고 요청을 보내야 할 집합으로부터 하나의 C-Rep를 선택해서 전송한다.

C-Rep의 선택은 origin 서버가 처리하는 것보다도 CSP에 더 근접해 있어서 오는 시간에 대한 이득뿐만 아니라 그 요청에 해당하는 문서를 캐시하고 있거나 링크참조정보를 통해서 그 문서와의 링크관계에 대한 정보 그리고 각 C-Rep들의 부하정보를 가지고 적절한 C-Rep로 요청을 전송한다. 만일 그 테이블에 요청된 origin 서버에 대한 엔트리가 없다면, 그

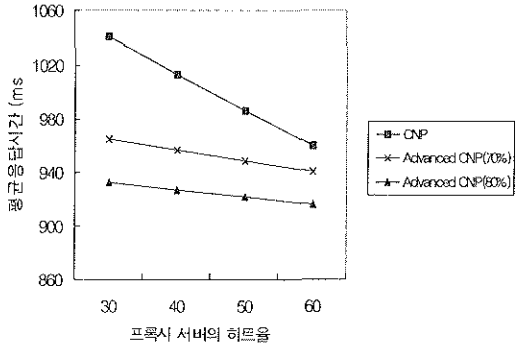
요청은 origin 서버에 전송된다. 이 때 CSP는 응답과 함께 서버의 캐싱 지역 정보를 받는다.

프록시 서버들은 인터넷 라우터들이 동작하는 것처럼 그들의 상태 정보와 프록시들간에 측정된 시간에 관해서 다른 프록시 서버들을 위해 그들이 가지고 있는 정보를 교환함으로서로 협력할 수 있다.

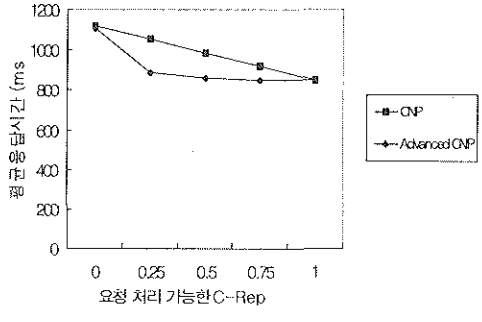
제 3 장 성능 평가

본 논문에서는 캐싱 지역 프로토콜에서의 클라이언트는 펜티엄 III(Redhat Linux 6.1)로 구성된 시스템에서 Compiler gcc 2.95.1을 이용하였고 프록시 서버는 SUN ULTRA 60(Solaris 2.6) 시스템에서 SUN WorkShop Compilers C/C++ 4.2를 이용하여 수행하였다.

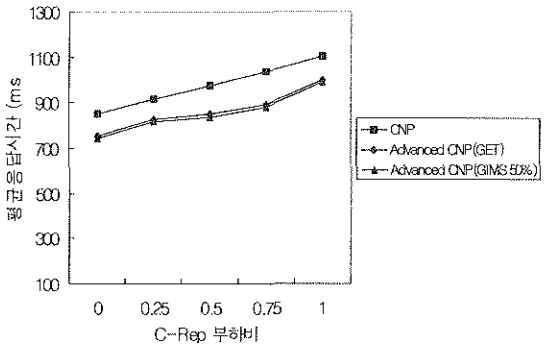
요청에 대한 응답시간을 성능 평가의 기준으로 삼았고 시뮬레이션에서 Get요청만 고려하고 문서의 크기는 평균 2-4Kbyte의 크기를 가진다. 그리고 만일 요청이 origin 서버까지 전송된다면 HTTP timeout이 일어나는 경우가 없다는 가정 하에서 수행되었다. 이런 가정 하에 부하에 대한 영향을 고려하지 않았을 때의 평균응답시간은 [그림 4], [그림 5]이고 고려했을 경우의 평균응답시간은 [그림 6], [그림 7]이다.



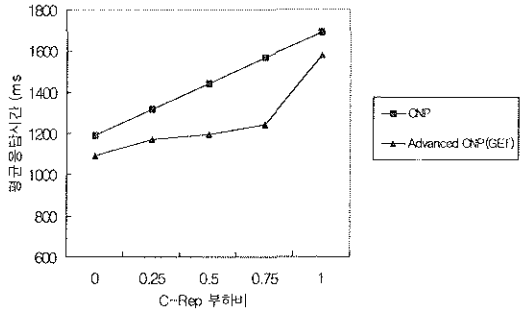
[그림 4] 히트율에 따른 평균응답시간



[그림 5] 적절한 C-Rep 선택에 의한 평균응답시간



[그림 6] 요청처리 가능한 C-Rep의 평균응답시간



[그림 7] 요청처리 불가능한 C-Rep의 평균응답시간

제 4 장 결 론

본 논문에서는 각 프록시 서버들에서 사용자들의 웹문서 링크참조에 관한 정보와 부하상태를 고려한 개선된 요청 전송 테이블을 통해 요청을 위한 보다 효율적인 라우팅 기법을 제안하였다. 이러한 기법을 통해 적절한 C-Rep를 선택하게 되고 이로써 요청에 대한 응답시간을 줄일 수 있다. 실험결과는 그러한 성능을 보여주고 있는데 기존의 기법에 비해 평균응답시간을 평균 8%-14%정도 감소시킬 수 있었다.

참 고 문 헌

[1] T.Berners-Lee, R. Calliau, A. Luotonen, H. Nielsen, and A. Secret, "The World-Wide Web", *Communications of the ACM*, 37(8), pp. 76-82, 1994.

[2] Squid Internet Object Cache. *(http://squid.nlanr.net/Squid)*

[3] C.-Y. Chiang, M. Liu and M. Muller, "Caching Neighborhood Protocol : A Foundation for Building Dynamic Caching Hierarchies with WWW Proxy Servers" to appear in *International Conference on Parallel Processing*, Aizu, Japan, September, 1999.