

지능형 프록시 서버를 이용한 웹검색시간 개선

권영미*, 이극**

* 목원대학교 컴퓨터공학과

** 한남대학교 컴퓨터공학과

Improvement of Web-surfing Response Time using Intelligent Proxy Server

Youngmi Kwon*, Geuk Lee**

* Dept. of Computer Engineering, Mokwon University

** Dept. of Computer Engineering, Hannam University

요 약

프록시(proxy)는 보안과 캐쉬를 목적으로 설치하는 서버이다. 에이전트는 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결해 주는 소프트웨어이다. 에이전트의 인터넷 관련한 대표적 응용으로는 인터넷 정보검색과 온라인 전자상거래를 들 수 있다. 인터넷 정보검색은 사용자로부터 원하는 주제를 받아 계속적으로 새로운 웹 사이트를 찾아다니며 정보를 찾아주는 기능이다. 온라인 전자상거래는 사용자가 원하는 물품이 어디에 있는지 찾아주는 기능이다. 에이전트는 지능(intelligence)를 가지면서 사용자의 취향을 추출하고 이를 학습해 나간다. 본 논문에서는 이러한 에이전트의 기능과 프록시의 기능을 복합적으로 갖춰 사용자 특성 파악을 단순한 정도로만 하여 요구가 예상되는 문서들을 미리 가지고 옴으로써 사용자의 웹정보검색(web surfing)시간을 단축시킬 수 있는 지능형 프록시 서버(Intelligent Proxy Server)를 제안한다. 제안된 방법은 검색결과를 사용자에게 가공하여 제공함으로써 검색대기시간을 더욱 줄일 수 있게 한다.

1. 서 론

프록시(proxy) 서버는 대체적으로 보안 및 캐쉬 기능을 위해 설치하는 서버이다. 보안이라 함은 방화벽(firewall) 기능을 가지고 있어 외부로부터의 네트워크 접근을 선별적으로 차단한다는 것이며,

캐쉬기능이라 함은 한번 접속이 이루어진 문서를 프록시 서버의 캐쉬에 저장해 놓았다가 이후 이 문서에 대한 요구가 있을 때 해당 사이트까지 다시 접근하지 않고 프록시 서버에 저장되어 있는 내용을 보내주는 기능이다.

에이전트(agent) 시스템은 자율성(autonomy), 지

능(intelligence), 이동성(mobility), 사교성(social ability)등을 갖고 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결하여 주는 시스템이다. 통신분야에서는 인터넷 정보 검색, 전자상거래, 이동 컴퓨팅, 전자우편등의 메시지 응용분야에서 활발하게 연구되고 있다. 인터넷 정보 검색에서의 에이전트는 사용자가 필요로 하는 정보를 입력받고 스스로 사용자의 요구 패턴도 학습하여 사용자에게 필요한 정보를 찾아내 사용자에게 알려주는 기능을 갖는다. 전자상거래에서의 에이전트는 새로운 카탈로그에 대한 정보를 수시로 점검하여 많은 상거래 사이트중 사용자가 원하는 물품이 어디에 있는지 찾아내서 알려주는 기능을 한다.

본 논문에서는 프록시 서버가 갖는 캐쉬 기능과 에이전트 시스템이 갖는 스스로 알아서 사용자의 요구를 해결해 주는 기능을 통합할 수 있는 지능형 프록시 서버(intelligent proxy server)를 제안한다. 제안된 지능형 프록시 서버는 분산된 서버들의 집합으로 구성되며 에이전트 시스템의 기능중 특히 사용자들의 특성을 고려하여 미리 문서들을 가져오는 기능을 갖는다. 또한 가지고 온 문서를 사용자에게 가공하여 제공함으로써 검색대기시간을 줄이는 기능도 갖는다.

본 논문의 구성은 다음과 같다. 제 2장에서 지능형 프록시 서버의 목적 및 구조와 메시지 타입을 기술하고 제 3장에서 사용자들의 웹검색 특성을 보이며 제 4장에서 결론을 맺는다.

2. 지능형 프록시 서버(Intelligent Proxy Server)

2.1 목 적

지능형 프록시 서버의 목적은 사용자로 하여금 웹검색 시간을 줄일 수 있도록 하는 것이다. 이를 위해 기존의 프록시 서버가 수행하던 캐쉬 기능을 제공해야 한다. 즉 이전에 가져온 적이 있는 문서에 대해서는 일정 기간 동안 보관해 두고 원격 서버에 가지 않고도 사용자에게 제공할 수 있어야 한다. 또한 지능형 프록시 서버는 사용자들의 검색 특성이 현재 검색중인 문서에 하이퍼링크로 링크된 문서를 다음에 검색할 확률이 높은 것에 착안하여 현재 검색중인 문서에 링크되어 있는 문서들을 미리 가져오도록 하여 사용자들의 대기시간을

줄인다.

또한 가져온 문서를 선택적으로 증류하여 사용자에게 제공할 수 있다. 문서를 부분적으로만 보여준다면 이미지 데이터의 경우 네트워크 상황에 따라 크기를 조절한다든가 색깔 수를 줄여 전체적인 데이터 크기를 줄일 수 있다. 또한 PostScript 과 같이 특수 포맷으로 만들어진 문서의 경우 일반 텍스트(plain text) 문서로 변환함으로써 전체적인 통신량을 줄인다.

2.2 구 조

지능형 프록시 서버는 다음의 세 가지 서버들의 집합으로 구성된다.

① proxy server

기존의 캐쉬기능을 수행하는 서버이다.

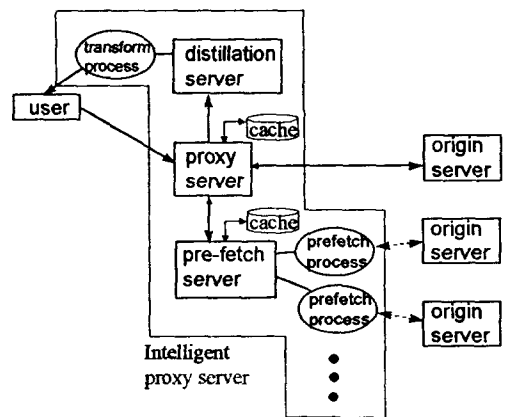
② pre-fetch server

사용자가 멀지 않은 미래에 가져오도록 요청할 문서를 예상하여 미리 가져오는 기능을 수행하는 서버이다.

③ distillation server

가지고 온 문서를 사용자에게 적당히 변환하여 사용자가 판단할 수 있는 정도의 질(quality)로 텍스트 및 이미지를 제공해 줌으로써 전체 통신량을 줄이는 기능의 서버이다.

이들의 전체적인 구성도는 [그림 1]과 같다.

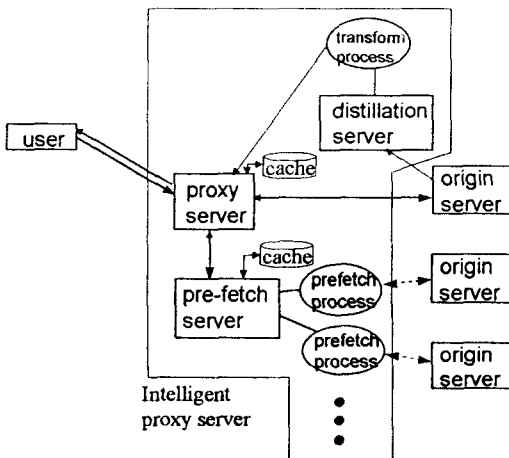


[그림 1] 지능형 프록시 서버의 구성도

Origin 서버는 사용자가 요구한 문서를 갖고 있는 로컬 또는 리모트에 위치한 http 서버이다.

Pre-fetch 서버는 간단한 기능으로 구현될 수도 있고 완전한 에이전트로서 에이전트 엔진, 영역지식(domain knowledge), 통신 모듈로 구성될 수도 있다. 본 논문에서는 가장 간단한 기능만을 갖도록 설계되었다. 추후 기능 확장 가능하다.

Distillation 서버는 [그림 1]에서 로컬쪽에 위치하여 프록시 서버와 메시지를 주고 받고 사용자에게 결과를 가공하여 주는 일을 한다. 그러나 distillation 서버는 [그림 2]에서와 같이 리모트의 origin 서버쪽에 위치하여 transform process의 결과를 로컬쪽 프록시 서버에게 결과를 넘겨줄 수 있다. Distillation 서버는 여러 프록시 서버에게 또는 여러 origin 서버에게 공유될 수 있다.



[그림 2] 리모트 사이트에 위치한 distillation 서버

Distillation 서버의 위치를 어디로 할 것인가는 웹검색의 전체 부하를 어떻게 나눌 것인가 하는 기본에 의해 정해져야 한다. 리모트쪽에 위치시키는 것은 검색결과와 통신량을 줄이고자 하는 목적에 맞으며 로컬쪽에 위치시키는 것은 사용자에게 검색 대기시간을 짧게 보장해 주기 위한 목적에 맞다. 또한 distillation 서버는 로컬이든 리모트든 여러 개로 구성될 수도 있는데 그런 경우 각 distillation 서버는 자기가 지원해야될 프록시 서버나 origin 서버의 영역을 나누어 담당하게 된다.

Pre-fetch 서버가 어떤 문서들을 미리 가지고 올 것인가에 대한 여러 기준이 있을 수 있다. 첫 째, 현재 검색중인 문서에 하이퍼링크된 모든 문서들을 미리 가져다 놓을 수 있다. 이것은 아주 많은 prefetch 프로세스를 발생시키게 되고 시스템의 성능을 저하시키며 대량의 통신데이터를 유발할 수 있다. 둘째, 제일 앞쪽에 하이퍼링크된 문서부터 유한개의 문서를 미리 가지고 올 수 있다. 셋 째, 랜덤하게 아무거나 몇 개를 가지고 오는 방법이 있을 수 있다. 넷 째, 그 동안의 히스토리(history)에 근거한 pre-fetch를 할 수 있다. 즉, 현재 검색중인 문서를 요구한 후 사용자들이 다음으로 가기를 원했던 하이퍼링크의 횟수를 카운트해두었다가 많이 방문되어지는 특성을 갖는 문서들을 미리 가져다 놓는 것이다. 이 방법은 많이 방문되는 문서의 경우 사용되어질 수 있지만 사용자들이 별로 사용하지 않는 문서의 경우는 카운트의 빈도가 높지 않고 따라서 히스토리에에만 의한 판단이 신뢰성이 낮다. 따라서 히스토리에 근거한 방법과 다른 방법을 병행하는 것이 좋다. 사용자의 특성을 파악할 수 있는 에이전트 기능을 갖게 될 때 보다 높은 hit-ratio를 갖는 pre-fetch가 가능할 것이다.

프록시 서버가 갖고 있는 캐쉬는 저장 기간이 일단위 또는 주단위가 되지만 pre-fetch 서버가 갖고 있는 캐쉬의 저장 단위는 분 단위로 갱신된다. pre-fetch 서버의 캐쉬에 있는 내용을 사용자가 요구한 경우 프록시 서버는 리모트에 있는 origin 서버가 아닌 pre-fetch 서버로부터 내용을 받고 자신의 캐쉬에 기록해 둔다.

2.3 메시지

1) HTTP GET request 메시지

이 메시지는 사용자가 웹문서를 브라우저에 요청하였을 때 발생한다. 사용자와 프록시 서버간, 프록시 서버와 origin 서버사이에서 발생된다.

2) HTTP GET response 메시지

이 메시지는 HTTP GET request 메시지에 대한 응답으로 발생한다.

3) PRE_GET request 메시지

이 메시지는 프록시 서버로부터 pre-fetch 서버

로 요구된다. 프록시 서버는 pre-fetch 서버에게 사용자로부터의 요구가 예상되는 문서를 미리 가져오게 할 것인가 아닌가를 결정할 수 있다. 이것은 사용자가 프록시 서버를 사용하게 할 것인가 아닌가를 결정해 줄 수 있듯이 사용자측에서 옵션으로 선택할 수 있게도 할 수 있다. PRE_GET request 메시지에는 현재 사용자가 요청한 문서 내용이 들어가 있는 타입 또는 사용자가 새로 요구한 문서의 URL이 들어 있는 타입이 있다. pre-fetch 서버는 URL이 들어 있는 타입의 메시지를 받은 경우 해당 URL 문서가 자신의 캐쉬에 있는가 살펴보고 있는 경우 미리 fetch되어 있는 문서를 프록시 서버로 넘겨주며 캐쉬에 없는 경우는 프록시 서버에게 자신에게 pre-fetch되어 있지 않다는 사실을 알린다. 이런 경우는 pre-fetch 서버가 해당 문서를 가져오는 일을 하지 않고 프록시 서버가 가져오도록 하는데, 그 이유는 통신량이 많은 경우 프록시 서버로부터의 통신요구가 더 높은 우선순위를 갖도록 하는데, pre-fetch 요구보다 사용자로부터 발생된 문서검색 요구가 더 높은 우선순위로 처리될 수 있도록 하기 위해서이다. 현재 사용자가 요청한 문서 내용이 들어가 있는 타입의 PRE_GET request 메시지를 받은 경우 pre-fetch 서버는 그 문서의 링크들을 분석하여 자신의 규칙에 따라 어떤 URL 문서들을 prefetch할 것인가 결정하고 프록시 서버의 캐쉬에 없는 경우, 또는 프록시 서버의 캐쉬에 있더라도 If-Modified-Since 메시지에 의해 일정기간 이후 변경된 사항이 있는 문서인 경우 prefetch 프로세스들을 만들어낸다. 각 prefetch 프로세스들은 해당 origin 서버로 HTTP GET request 메시지를 보낸다.

Prefetch 프로세스들이 계속 캐쉬로 문서를 가져오고 있는 중에 pre-fetch 서버가 프록시 서버로부터 사용자 요구 URL이 들어있는 타입의 PRE_GET request 메시지를 받게 되는 경우가 있다. 이 때 pre-fetch 서버의 캐쉬에 들어있지 않은 문서가 요구된 경우, 현재 생성되어 있는 prefetch 프로세스들을 kill하고 새로운 prefetch 프로세스들을 생성함으로써 너무 많은 prefetch 프로세스들이 생겨나지 않도록 한다.

4) PRE_GET response 메시지

이 메시지는 PRE_GET request 메시지의 응답으

로 발생한다.

5) DISTILLATION request 메시지

Origin 서버에서 리모트에 위치한 distillation 서버에게 또는 프록시 서버에서 로컬에 위치한 distillation 서버에게 요청되는 메시지이다. 원래의 문서 내용이 들어가 있으며 distillation 서버는 transform 프로세스를 통해 각 사용자에게 제공될 정보로 가공한다. Distillation 서버는 여러 프록시에 의해 여러 사용자에게 공유될 수 있으므로 각각의 문서에 대해 하나의 transform 프로세스가 생성되어 문서를 가공한다.

6) DISTILLATION response 메시지

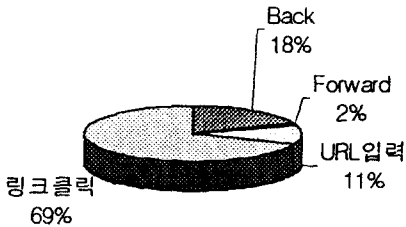
DISTILLATION request 메시지의 응답으로 발생하는 메시지이다. 이미지 데이터의 칼라수를 줄이든가 사이즈를 줄이든가 복잡한 형식의 텍스트가 단순 텍스트로 변화하는 등의 증류작용을 거친 결과 문서 내용을 포함한다.

3. 사용자 특성

사용자들이 브라우저를 통해 월드와이드웹 문서를 검색해 나갈 때 전혀 새로운 문서의 URL을 입력하여 이동하는 것 보다는 현재 검색중인 문서에서 하이퍼링크로 연결된 문서로 이동하거나 Back 버튼, Forward 버튼을 이용하여 이동하는 경우가 많다. [표 1]과 [그림 3]은 36명의 사용자가 144회에 걸쳐 인터넷 쇼핑물을 이용하면서 물건을 선택하기까지 브라우저에서 사용했던 검색 특성을 보여준다. Back 버튼을 사용한 경우가 18%, Forward 버튼을 사용한 경우가 2%, URL을 직접 입력한 경우가 11%, 하이퍼링크를 따라 이동한 경우가 69%이다.

	Back	Forward	링크클릭	URL 입력
횟수	202	19	753	120
퍼센티지	18%	2%	69%	11%

[표 1] 사용자특성 실험데이터



[그림 3] 인터넷 검색 사용자 특성

제안된 지능형 프록시 서버가 현재 검색중인 문서에 링크되어 있는 문서의 URL을 바탕으로 미리 가져다 놓는 작업을 하므로 하이퍼링크를 따라 검색해나가는 특성이 강한 사용자일수록 더 좋은 성능을 기대할 수 있다.

4. 결론

프록시 서버는 보안 및 캐쉬 기능을 제공하는 서버이며 에이전트는 자율성(autonomy), 지능(intelligence), 이동성(mobility), 사교성(social ability)등을 갖고 사용자를 대신해서 사용자가 원하는 작업을 자동적으로 해결하여 주는 시스템이다. 통신분야에서는 인터넷 정보 검색, 전자상거래 등에 응용되고 있다.

본 논문에서는 사용자의 요구를 미리 예상하여 요구될 문서를 미리 가져옴으로써 사용자의 웹검색 대기시간을 줄일 수 있는 pre-fetch 서버와, 전체적인 통신량을 줄이고 사용자에게 적당한 정도의 문서 질(質)을 제공해주는 기능의 distillation 서버, 그리고 기존의 프록시 서버가 연계되어 동작하는 지능형 프록시 서버를 제안하고 프록시 서버와 pre-fetch 서버, distillation 서버 사이에 오고가는 메시지를 정의하였다.

Distillation 서버와 pre-fetch 서버의 기능은 점차 보장되어 높은 수준의 에이전트 기능을 제공할수록 업그레이드될 수 있다.

Pre-fetch로 인한 통신량의 증가가 문제될 수 있으나 pre-fetch 서버로부터의 통신 요구보다 프록시 서버로부터의 통신 요구 우선순위를 높게 해주고 사용자로부터의 직접적인 검색 요구는 프록시 서버에서 담당하게 함으로써 pre-fetch로 인한 사

용자의 웹검색 대기시간이 늘어나는 것을 방지하였다.

새로운 URL을 입력하여 검색하는 것 보다는 현재 검색중인 문서에 하이퍼링크된 다른 문서를 따라가며 새로 검색을 요구하는 특성이 높게 나타나므로 전체적인 사용자의 웹검색 대기시간을 줄일 수 있다.

참 고 문 헌

- [1] 최중민, "에이전트의 개요와 연구방향", 한국정보과학회 정보과학회지 제 15권 제 3호, pp. 7-16, 1997
- [2] 이은석, 이진구, 강재연, "인터넷 상에서의 전자상거래를 위한 멀티에이전트 시스템," 한국정보처리학회 정보처리학회지 Vol.4, No. 5, pp. 55-66, 1997
- [3] D. Wessels, K. Claffy, "Internet Cache Protocol(ICP), version 2," Internet-Draft, 1996
- [4] Ingrid Melve, "Web caching System Design Checklist," <http://www.uninett.no/prosjekt/desire/arneberg>, 1997
- [5] Duane Wessels, "Configuring Hierarchical Squid Caches," <http://squid.nlanr.net/Squid/Hierarchy-Tutorial>, 1997
- [6] G. V. Dias, G. Cope, R. Wijayarathne, "A Smart Internet Caching System," http://info.isoc.org:80/isoc/whatis/conferences/inet/96/proceedings/a4/a4_3.htm, 1996
- [7] Yatin Chawathe, "A Load Balancing Resource Locator for Proxies," <http://HTTP.CS.Berkeley.EDU/~yatin/ptm/main.html>
- [8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "RFC2068: Hypertext Transfer Protocol - HTTP/1.1," Internet Community, 1997