

콘텐츠 분산 환경에서 서버 선택을 위한 리다이렉션 시스템의 설계

정회원 김 선 호*, 고 훈**, 신 용 태***

A Design of Redirection System for Server Selection on the Content Distribution Environment

Seonho Kim*, Hoon Ko**, Yongtae Shin*** *Regular Members*

요 약

웹(Web)이 급속히 발전함에 따라 웹상에서 단순한 텍스트 파일 뿐 아니라 이미지, 오디오, 비디오 등의 멀티미디어 서비스에 대한 사용자의 요구도 기하급수적으로 증가하고 네트워크 대역폭의 수요도 급격히 증가하고 있다. 이러한 상황에서 콘텐츠의 빠른 전송과 적은 트래픽 발생을 위하여 Caching, P2P, CDN 등의 기술이 대두되었다. 본 연구에서는 콘텐츠가 여러 복제 웹 서버에 분산 배치되어 있는 환경에서 클라이언트의 요청에 가장 적절한 서비스를 해 줄 수 있는 웹 서버를 선택하기 위한 리다이렉션 시스템을 설계하고 구현하였다. 본 리다이렉션 시스템은 클라이언트의 요청을 토폴로지 상 가깝고 부하가 적은 복제 웹 서버로 리다이렉트 함으로써 클라이언트가 기대하는 응답시간을 단축시켰다. 이러한 연구는 지속적으로 증가하고 있는 대규모 멀티미디어 콘텐츠의 안정적이고 빠른 전송에 기여할 것이다.

Key Words : Redirection; Replica Web Server; Server Load; AS.

ABSTRACT

With the development of Web technologies, the demand for multimedia contents and network bandwidth is increasing explosively. In this situation, Caching, P2P and CDN are using for fast transmission with low traffic. In this research, we designed and implemented a redirection system for best replica selection where contents are distributed to multiple replicated web servers. This redirection system selects a close and least loaded replica and redirect a client's request to the replica. As a result, this system improved user-perceived performance. It is expected that this redirection system will contribute to the faster and more stable transmission of massive multimedia contents.

I. 서 론

최근 인터넷 기술의 발달로 웹상에서 단순한 텍스트 파일 뿐 아니라 이미지, 오디오, 비디오 등의 멀티미디어 서비스에 대한 사용자의 요구가 급증하

고 있으나 네트워크 대역폭과 서버의 성능 향상이 콘텐츠의 증가를 능가하지 못하고 있어 서버와 네트워크의 부담이 가중되고 서비스의 질이 떨어지는 문제가 발생하게 되었다.

이러한 문제를 해결하기 위해 Caching, Mirroring 등의 기술들이 사용되었으나[1] 최근에는 서버로 접근

* 동덕여자대학교 정보과학대학 컴퓨터전공 강의전임교수(shkim98@dongduk.ac.kr),

** 대전대학교 컴퓨터공학과 초빙교수(skoh21@daejin.ac.kr), *** 숭실대학교 컴퓨터학과 부교수(shin@computing.ssu.ac.kr)

논문번호 : 030387-0903, 접수일자 : 2003년 9월 3일

하지 않고 서버의 정보를 갖고 있는 사용자 간에 서로 서비스를 주고 받는 P2P(Peer To Peer) 기술이 하나의 방법으로 제시되었으나 이것은 보안과 관리 측면에서 해결해야 할 많은 문제들이 있다[2]. 그러므로 공간적으로 멀리 떨어져 있는 서버의 콘텐츠를 인터넷상에 분산하여 설치한 다수 서버에 복제하여 사용자와 가까운 서버로부터 서비스하게 함으로써 서비스의 질을 높이도록 하는 CDN(Content Delivery Network) 기술이 적절한 대안으로 제시되고 있다[3]. CDN은 인터넷의 여러 장소에 복제 서버를 설치하고 사용자와 가까운 복제 서버가 사용자의 요청에 응답하게 함으로써 서버의 부하를 줄이고 응답 시간을 단축시켜 결과적으로 서비스의 질을 높이는 결과를 가져왔다.

그러나 DNS 등에 의한 가까운 복제 서버의 선택은 특정 복제 서버로의 부하 집중 등 또 다른 문제를 야기시킬 수 있다. 그러므로 이러한 콘텐츠 분산 환경에서는 변하는 인터넷 토폴로지와 복제 서버의 부하 상태에 따라 클라이언트의 요청을 적절한 복제 서버로 전달하여 서버와 네트워크의 부하를 분산할 수 있는 리다이렉션 메커니즘이 필요하다.

따라서 본 연구의 목적은 대규모 인터넷상에 산재한 복제 서버 중에서 클라이언트의 요청 시 클라이언트와 토폴로지 상 가깝고 부하가 적은 서버가 응답할 수 있도록 하여 클라이언트가 인지하는 응답시간을 향상시킬 수 있는 리다이렉션 시스템을 설계하는 것이다.

II. 관련 연구

2.1 리다이렉션 메커니즘

2.1.1 DNS-based Redirection

기존 프로토콜이나 클라이언트 소프트웨어의 변경 없이 사용할 수 있는 가장 단순하고 일반적인 방법이다. 클라이언트의 DNS 질의에 대해 복제 서버 중 하나의 IP 주소를 반환한다. 클라이언트 측의 아무런 수정 없이 기존 인터넷상의 DNS 인프라를 그대로 사용할 수 있고 콘텐츠 제공자 입장에서도 단지 콘텐츠가 저장되어 있는 단말 서버들의 위치 목록만을 DNS에 추가하면 된다.

그러나 대부분의 경우 클라이언트는 TCP/IP에 설정된 지역 DNS에게 질의를 맡기게 되는데 지역

DNS에 의해 질의를 받은 요청 리다이렉터는 클라이언트의 정확한 위치를 알 수 없기 때문에 인접한 서버 선택을 보장할 수 없으며 하나의 지역 DNS에 의한 요청은 같은 서버로 리다이렉션 되므로 같은 지역 DNS에 접근한 클라이언트가 많은 경우 특정 서버에 부하가 집중될 수 있다. 또한 DNS 기반의 서버 선택은 가장 일반적인 방법이지만 이 방법은 결국 지역 DNS와 가까운 복제 서버를 선택하는 것이므로 기본적으로 클라이언트와 지역 DNS의 위치가 가깝다는 것을 전제하는 것이다. 그러나 실험 결과[4] 클라이언트와 지역 DNS가 같은 AS에 있는 경우는 64%, 같은 BGP 라우팅 정보를 기반으로 할 때 같은 네트워크 클러스터에 있는 경우는 16%에 불과하였다. 또한 [5]에서 웹 클라이언트와 지역 DNS 사이를 측정된 결과 15%만이 1 hop 거리에 있었고 30%는 적어도 8 hop 이상 떨어진 것으로 나타나 대부분의 웹 클라이언트는 그들의 지역 DNS와 토폴로지 상 떨어져 있는 것으로 나타났다. 이것은 다이얼 업 클라이언트의 경우도 같은 결과를 보였다.

2.1.2 HTTP Modification

HTTP와 같은 응용 수준의 프로토콜은 URL의 prefix나 suffix를 이용하여 요청을 전달 할 수 있다. HTTP 문서 안의 객체에 대한 URL을 재설정하거나 프로토콜의 302 응답코드(Moved Temporarily)를 사용하여 요청 전달을 수행한다[1]. 요청 리다이렉터는 웹 서버로 동작하여 클라이언트 요청시 302 응답코드와 함께 인접 서버의 위치를 반환하고 클라이언트는 302 응답 메시지에 포함된 서버 주소로 요청을 한다. 이 경우 클라이언트가 바로 요청 리다이렉터에게 요청을 보내기 때문에 요청한 클라이언트의 정확한 위치를 파악할 수 있다.

구현하기가 매우 간단하지만 모든 수신되는 패킷에 대해 적절한 서버로 라우팅 되도록 IP 주소를 변경해 주어야 하므로 부가적인 오버헤드와 지연이 발생한다. 또한 중앙 서버가 HTTP 요청을 받아 다른 서버로 요청을 전달하기 때문에 중앙 서버가 전체 시스템을 무력화시킬 수 있다. 또한 클라이언트가 리다이렉션된 사이트를 북마크하여 다음에 북마크를 통하여 서버에 접속하게 되면 중앙 서버로부터 최적의 서버를 할당받지 못하고 하나의 서버에 부하가 집중될 수 있다.

2.2 서버 선택을 위한 정책

[6]에서는 클라이언트의 요청을 웹 서버들간에 돌아가며 분배하기 위하여 Round Robin DNS를 사용한다. 클라이언트의 요청에 대해 여러 대의 서버에 동일한 비율로 서비스 요구가 전송 될 수 있도록 웹 서버 IP 주소를 반환함으로써 서버의 부하를 분산하도록 하는 것이다. 구현이 간단하고 가장 일반적인 방법이다. 그러나, 이것은 DNS가 웹 서버의 상태는 알 수 없기 때문에 현재 웹 서버의 상태는 고려하지 않고 단순히 하나씩 차례로 선택을 해나감으로써 각 서버에 접속하는 것을 공평하게 하고자 하는 것이다. 동질의 웹 서버 시스템에서는 Round Robin을 사용하여 트래픽을 분산시킬 수 있지만 부분적인 부하 분산만 가능하다.

[7]에서는 사용자와 가장 가까운 서버를 선택하도록 하기 위해서 시간에 따라 변화하는 인터넷 토폴로지를 반영하여 AS Path 길이가 짧은 서버를 선택하였다. 그러나 AS를 기반 한 거리 측정은 간단하지만 AS의 범위가 매우 다양하므로 AS Path 만으로는 정확히 가까운 서버를 찾는 것이 어렵다.

[8]에서는 라우터가 클라이언트의 요청을 가로채어 가까운 복제 서버로 라우팅하는 SPREAD를 제시하였으나 이 시스템을 적용하기 위해서는 많은 라우터를 수정해야 하기 때문에 적용하기에는 너무나 많은 비용이 든다.

[9]에서는 AS간의 거리, AS 내부에서의 거리와 네트워크 지연을 고려하였다. 하나의 매트릭을 고려한 것보다는 더 정확하게 상태를 반영할 수 있지만 모든 복제 서버에 "agent"의 구현을 필요로 하며 네트워크 지연을 측정하기 위하여 추가적인 트래픽을 발생시킨다.

[10]에서는 Hops, RTT, 등 여러 서버 선택 매트릭 간에 비교 연구를 하여 특정 서버의 과거 지연에 대한 통계 데이터가 현재의 요청 지연을 예측할 수 있게 함으로써 가장 지연이 적은 서버를 선택할 수 있다는 것을 발견하였으나 이 연구는 응답 시간이 아닌 비용 함수로 HTTP 지연 시간을 비교하였으며 전체 응답시간은 추정하지 못하였다. 또한 HTTP에서 요청하는 파일 사이즈가 보통 10KB 이하로 작다는 것을 가정하였기 때문에 용량이 큰 파일의 경우는 결과가 다를 수 있다.

[11]의 연구에서는 사용 가능한 대역폭과 RTT를 측정하는 틀을 사용하여 파일 크기가 작은 경우는

RTT를 이용하여 서버를 선택하고 파일 크기가 큰 경우는 RTT와 사용가능한 대역폭을 함께 고려하여 동적인 서버 선택하는 알고리즘을 택하여 과거 정적인 접근 매트릭과 HTTP 전송 시간을 비교한 결과 50% 이상 향상되었다. 그러나 대역폭을 고려하는 경우 시간과 네트워크 트래픽 면에서 오버헤드가 크기 때문에 적당하지 않다.

III. 리다이렉션 시스템의 설계

3.1 시스템의 구조 및 개요

본 연구의 리다이렉션 시스템은 클라이언트의 요청시 동적으로 클라이언트와 인접하고 부하가 적은 복제 서버를 선택하여 해당 서버로 클라이언트의 요청을 리다이렉트한다. 이것은 기존의 정적인 서버 선택과는 달리, 하나의 복제 서버로 요청이 집중되는 문제점을 제거하여 클라이언트에게 안정적인 웹 서비스를 제공하는 것을 가능하게 한다.

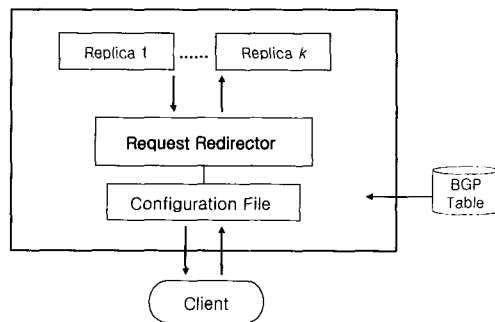


그림 1. 시스템의 구조

그림 1의 시스템 구조에 표시된 각 구성 요소들의 기능을 정의하면 다음과 같다.

- 클라이언트(Client)
웹상에서 객체를 요청하는 주체로써 일반 사용자의 웹 브라우저를 의미한다.
- 환경 설정 파일(Configuration File)
AS 번호와 해당 AS의 복제 서버의 IP 주소 mapping을 유지한다.
- 리다이렉터(Redirector)
클라이언트로부터의 요청을 받아들여 BGP 라우팅 테이블과 환경설정 파일을 근거로 클라이언트와 가까운 k 개의 복제 서버를 찾아내고 그중 부하가

가장 적은 복제 서버의 IP주소를 클라이언트에게 응답하는 클라이언트와 복제 서버 사이의 중간 주체이다.

○ 복제 서버(Replica Server)

자신의 부하 정보를 리다이렉터에게 보내주고 클라이언트로부터 콘텐츠 요청을 받아 응답하는 주체이다.

○ BGP Table

BGP 라우팅 테이블을 저장하고 있는 파일

서버의 부하가 많이 걸리고 처리 시간이 상대적으로 많이 요구되는 복제 서버의 위치 추적을 위한 라우팅 정보, DB 관리 등은 리다이렉터가 담당하고 복제 서버는 전송만 담당하도록 기능을 분리함으로써 요청 처리 시간을 줄일 수 있도록 하였다.

3.2 서버 선택 알고리즘

본 논문에서는 인접하고 부하가 적은 서버를 선택하는데 있어서 리다이렉터가 클라이언트와 인접한 k 개의 후보 서버를 선택하고 후보 서버의 부하 정보를 받아 가장 부하가 적은 서버를 선택하도록 하는 알고리즘을 제안한다.

그림 2에 표시된 제안한 알고리즘의 서버 선택 과정은 다음과 같다.

- 1) 클라이언트는 지역 DNS에게 도메인 이름 해석을 요구한다.
- 2) 지역 DNS는 매핑 테이블에 의해 클라이언트의 요청에 대해 리다이렉터의 IP주소로 응답 메시지를 생성하여 클라이언트에게 전달한다.
- 3) 클라이언트는 응답 메시지를 분석하여 해당 리다이렉터에게 서비스를 요청한다.
- 4) 리다이렉터는 BGP 라우팅 테이블, 클라이언트의 IP주소, 복제 서버 리스트를 참조하여 클라이언트와 가까운 AS의 후보 서버 집합을 선택한 후 후보 서버의 부하 정보를 수집한다.
- 5) 리다이렉터는 부하가 가장 적은 서버를 선택하여 클라이언트의 요청이 해당 서버로 리다이렉트 되도록 한다.

3.2.1 AS 그래프에 의한 인접 서버 선택

리다이렉션 시스템이 클라이언트와 가까운 서버를 찾는 메트릭으로는 클라이언트와 서버 간의 AS Path를 사용한다. 이것은 AS(Autonomous System)

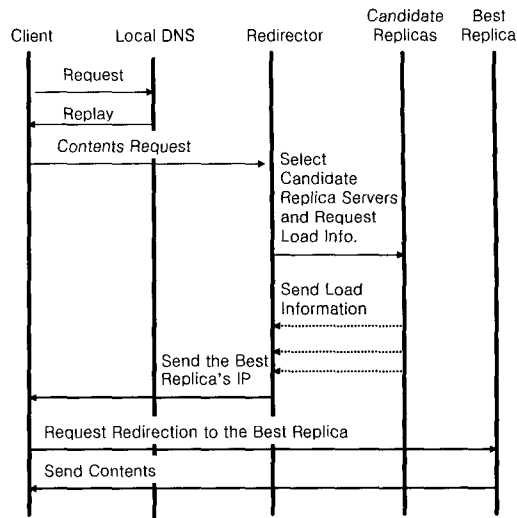


그림 2. 서버 선택 과정

내부에서의 지연이나 손실은 거의 없는 것을 가정하는 것이다[7].

실제 인터넷의 토폴로지를 이용하기 위해 NLAN R에 의해 제공되는 BGP 라우팅 테이블을 이용한다. 이것은 RouteView 프로젝트[12]에 의해 구축되었고 <http://routeviews.org> 사이트로부터 주기적으로 BGP 라우팅 테이블을 읽어와 AS 탐색 트리를 생성함으로써 AS Path를 구할 수 있다. 이 데이터는 인터넷 토폴로지를 생성하거나 이용하고자 하는 많은 연구들에 의해서 사용되어져 왔고 검증되어 왔다[13-15]. 원하는 수의 복제 서버를 발견할 때까지, 또는 전체 그래프가 모두 읽혀질 때까지 그래프를 탐색한다. BGP 라우팅 테이블은 하루에 한번씩 읽어 오도록 하며 테이블을 읽어 AS 그래프를 만드는 시간 동안의 지연을 줄이기 위해 중전의 그래프를 그대로 유지하고 있다가 최신 그래프가 다 형성된 후에 업데이트한다. BGP(Border Gateway Protocol)는 AS 번호가 다른 네트워크간의 라우팅 정보를 주고받는 프로토콜이다[16]. AS Path는 해당 목적지까지 갈 때 경유하는 AS 번호들로 구성된 것이다.

예를 들어, 그림 3과 같이 네트워크가 구성되어 있을 경우 AS100의 입장에서 네트워크 1.1.1.1에 대한 AS Path는 200 300 400 500이 된다. AS500에 있는 라우터가 AS400에 있는 라우터에게 203.2.53.25.2에 대한 정보를 전달할 때 AS Path는 500으로 하여 전달하므로 AS400에는 1.1.1.1 500이 저

장되고 AS400에 있는 라우터가 AS300에 있는 라우터에게 203.253.25.2 에 대한 정보를 전달할 때 자신의 AS 번호를 AS Path에 Prepend 시킨 400 500을 전달하여 AS300에는 1.1.1.1 400 500이 저장된다. 이런 식으로 AS100에는 1.1.1.1 200 300 400 500이 저장되게 된다. 동일한 목적지에 대한 AS Path의 길이가 짧을 경우 가까운 곳에 있다고 판단하여 짧은 AS Path를 가진 경로를 선택한다. AS Path 400 500의 길이는 2가 되고, AS Path 200 300 400 500의 길이는 4가 된다.

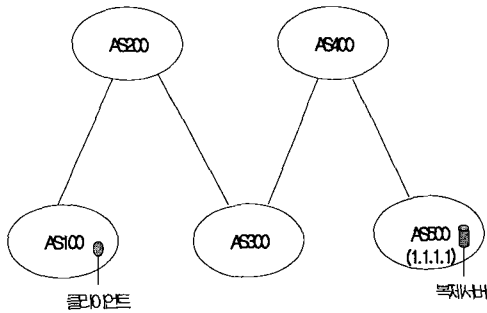


그림 3. AS 그래프에 의한 경로

리다이렉터에서 BGP 라우팅 정보를 근거로 AS 그래프를 생성하여 서버를 선택하는 과정은 다음과 같다.

- 1) 클라이언트의 요청 메시지가 리다이렉터로 전달
- 2) BGP 테이블에서 네트워크 prefix와 AS 번호를 mapping한 이진 트리를 만들어 클라이언트의 IP 주소를 AS 번호로 변환
- 3) AS Path 그래프로 AS 연결 리스트를 만들고 복제 서버 리스트를 참조하여 클라이언트와 가까운 AS의 k개의 복제 서버들을 후보 서버로 선택
- 4) 후보 서버들에게 부하 정보를 전송하도록 요구

3.2.2 부하가 적은 복제 서버의 선택

리다이렉터는 후보 서버 중 부하가 가장 적은 복제 서버의 IP주소로 응답한다. 이때, 리다이렉터가 고려하는 서버의 부하 메트릭으로는 서버 자체의 성능을 나타내는 성능계수와 현재 서버가 연결하여 수행 중인 작업 수를 고려한다.

- 성능 계수 : 성능 계수는 클라이언트의 요청 (request)을 처리하는 시간(second) 으로 정의한다. 성능 계수를 구하기 위하여 웹 서버 성능 분석 툴인 httpperf[17]를 이용하여 현재 서비스 중

인 각 복제 서버의 성능을 주기적으로 체크하여 그 평균값을 성능 계수로 한다.

- 현재의 부하 : 현재의 부하는 현재 복제 서버에 연결되어 있는 서비스 수로 정의한다.

성능 계수는 미리 측정하여 리다이렉터가 보관하고 있으며 클라이언트의 요청 시에 복제 서버의 연결 서비스 수를 성능 계수와 곱하여 가장 작은 값을 갖는 서버가 서비스하도록 한다.

IV. 구현

4.1 구현 환경 및 모델

리다이렉션 시스템은 C언어와 PHP언어를 사용하여 Linux 상의 Gnu C 기반에서 구현되었다. 리다이렉터와 복제 서버 간에 Load Collector 와 Load Sender 의 통신을 위하여 프로토콜 확장성이 높은 XML을 사용하였다. 구현된 모듈은 다음과 같다.

- Redirect Processor

Redirect Processor는 클라이언트의 요청이 오면 IP-AS translation을 통해 클라이언트의 AS 번호를 알아내고 AS linked list를 탐색하여 가장 가까운 AS 번호를 알아내고 복제 서버 리스트에 등록되었는지 확인하여 k 개의 복제 서버를 찾을 때까지 반복한다. k 개의 후보 서버에게 부하 정보를 요청하고 부하가 가장 적은 복제 서버의 IP주소를 클라이언트에게 보내어 요청을 해당 서버로 리다이렉트하도록 한다. 복제 서버의 목록 관리는 컴파일 필요 없고 웹 서버 상에서 뛰어난 성능을 가진 PHP로 구현한다. 리다이렉터는 Load Collector를 사용하기 위해 시스템 호출을 하며 개개의 복제 서버의 connection 수(c)와 서버 성능 계수(r)를 사용한 수치는 Current Load 라고 정의하며 $r * (c+1)$ 로 계산하여 가장 작은 값을 갖는 복제 서버의 IP 주소를 반환한다.

- IP to ASN Translation

BGP 라우팅 테이블을 근거로 prefix를 이진 트리로 구성하여 IP 주소의 prefix에 해당하는 AS 번호를 탐색한다.

- AS Linked List

AS 연결 리스트를 구성하여 해당 AS와 가장 가까운 AS를 BFS(Breadth First Algorithm)으로 탐색한다.

- Load Collector

Load Collector는 후보 서버들의 IP 주소로 Load Sender에 소켓 접속하여 Load Sender에게 현재의 부하 상태를 요청하여 리다이렉터가 Best Replica Server 를 판단하는 정보를 제공한다.

○ Load Sender

Load Collector로부터 부하 정보를 원하는 커백션이 발생하면 현재의 부하 정보를 측정하여 Load Collector에 전달한다. Load Sender는 상주 서버여야 하며 동시에 여러 사용자를 수용할 수 있어야 하는 서버 기술이 필요하다. Load Sender는 서버 프로그램과 커맨드 프로그램으로 나뉜다. 서버 프로그램은 입출력 다중화 방식의 멀티 프로세스 프로그램으로 동시 접속 사용자를 고려하여 설계되었다. 서버 프로그램은 각각의 Load Collector의 요청에 새로운 프로세스를 생성하여 커맨드를 실행시키며 각각의 새로운 프로세스들은 Load Collector와의 통신 후 사라진다. Load Sender의 커맨드 프로그램은 Load Collector와 통신을 위해 Xml_parser를 가지고 있으며 부하 정보를 구하기 위해 파이프를 이용하여 현재 연결되어 있는 connection 수를 구하여 이 결과 값을 XML 형태로 바꾸어 Load Collector에 전달하고 끝난다.

다음 그림 4는 복제 서버의 Load Sender가 자신의 부하 정보를 전송하는 상태를 보여준다.

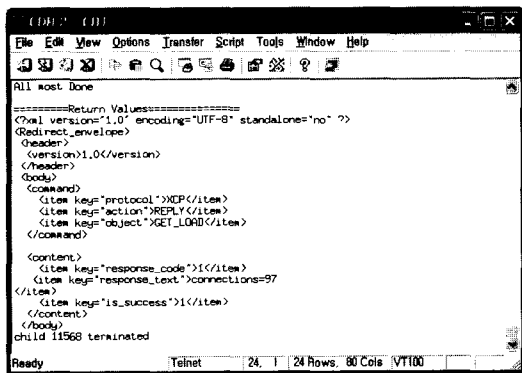


그림 4. Load_Sender의 처리 내용

V. 실험 및 분석

5.1 실험환경

실험은 리눅스 운영체제의 아파치 웹 서버 상에서 하였으며 실험에서 사용한 서버들의 사양과 http

erf를 이용한 복제 서버들의 성능계수는 표 1과 같다. 각 서버들은 100 Mbps의 LAN으로 연결되어 있다.

비교 대상으로는 인접 메트릭으로는 AS를, 부하 메트릭으로는 Round Robin을 사용한 방법을 택하여 AS와 서버의 부하를 고려한 제안하는 방법의 성능을 비교 검증하였다. 클라이언트가 가공의 요청 횟수를 10에서부터 500으로 증가시켰을 때 평균 응답 시간을 측정하였다.

본 실험에서는 실험의 편의를 위하여 후보 서버 k의 개수를 3으로 하고 BGP 테이블을 이용하여 임의로 클라이언트를 선택하여 가까운 AS의 복제 서버 k 개를 찾는 시간을 따로 측정하고 실험에서 사용하는 복제 서버에서 부하가 적은 서버로부터 응답을 받는데 걸리는 시간을 별도로 측정하여 더하였다.

표 1. 실험 서버 사양

	CPU	RAM	성능계수
Redirector	Pentium IV 1.6 GHz	512MB	
Replica1	Pentium IV 1.8GHz	8.5GB	3.9
Replica2	Xeon 2.4GHz	8.5GB	4.3
Replica3	Pentium IV 1.8GHz	256MB	4.8

5.2 실험결과

실험은 복제 서버1에는 100개의 커백션이, 복제 서버2에는 50개의 커백션이 지속적으로 연결되도록 하고 복제 서버3은 서비스 중인 커백션이 하나도 없도록 한 상태에서 클라이언트의 요청을 10에서 500으로 증가시킬 경우 1KB의 파일을 다운 받는데 걸리는 시간을 측정하였다.

제한된 리다이렉션 시스템에서는 클라이언트의 요청이 복제 서버의 성능과 현재 커백션 수를 곱하여 가장 작은 값을 갖는 웹 서버로 전달되어 서비스를 받는다. 반면 Round Robin에 의한 경우 클라이언트의 요청이 AS에 의한 거리에 의해 선택된 후보 서버들에게 차례로 전달되어 부하가 가장 큰 복제 서버로 전달된 경우를 가정하여 응답시간을 측정하였다.

측정 결과 그림 5에서와 같이 제안된 리다이렉션

시스템에서는 현 상태의 인접도와 서버 부하를 고려하여 여전히 가깝고 성능이 좋은 복제 서버로 요청이 전달되므로 향상된 응답시간을 나타내었다. 실험의 한계 상 검증이 어려웠으나 대용량의 파일의 경우는 기존에 비하여 더 좋은 응답시간을 나타낼 것이다.

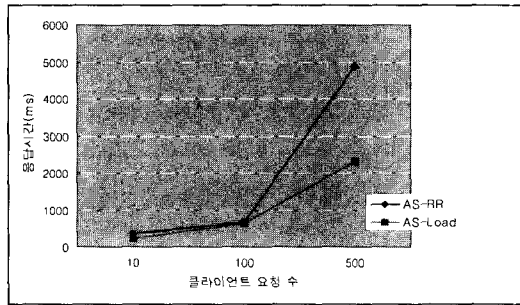


그림 5. 요청 수에 따른 응답시간

VI. 결론 및 향후 연구과제

본 리다이렉션 시스템은 다수의 복제 웹 서버 중에서 클라이언트가 요청한 시점에서 가장 적절한 서버를 동적으로 선택하는 것을 가능하게 하였다. 이것은 클라이언트가 인지하는 응답시간을 단축시켰으며 대용량의 파일일 경우 더욱 좋은 성능을 보일 것이다.

그러나 클라이언트의 요청이 특정 시점에서 집중될 경우 리다이렉터의 과부하 문제가 발생할 수 있다. 그러므로 향후 AS레벨의 토폴로지 하에서 트래픽과 지연 시간을 고려한 적절한 리다이렉터의 수와 위치 선정에 관한 연구가 있어야 할 것이다.

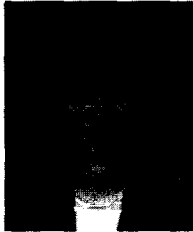
참 고 문 헌

[1] M. Rabinovich and O. Spatscheck, "Web Caching and Replication", Addison Wesley, 2002.
 [2] S. Saroiu, P.K. Gummadi, and S.D. Gribble. "A Measurement Study of Peer-to-Peer File Sharing Systems", In *Proceedings of Multimedia Computing and Networking*, 2002.
 [3] G. Peng, "CDN: Content Distribution Network", in *Stony Brook University Tech. Reports*, TR-125, Sc

hool of Computer Science Department, Stony Brook University, 2003.
 [4] Z. M. Mao, C. Cranor, F. Douglis, M Rabinovich, O. Spatscheck, and J. Wang, "A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers", In *Proceedings of USENIX Annual Technical Conference*, 2002.
 [5] A. Shaikh, R. Tewari, and M. Agrawal, "On the Effectiveness of DNS-based Server Selection". In *Proceedings of IEEE INFOCOM*, 2001.
 [6] T. Brisco, "RFC1794: DNS Support for Load Balancing", 1995.
 [7] P. R. McManus, "A Passive System for Server Selection within Mirrored Resource Environments Using AS Path Length Heuristics", *Applied Theory Communications, Inc.*, 1999.
 [8] P. Rodriguez, and S. Sibal. "SPREAD: Scalable Platform for Reliable and Efficient Automated Distribution", *Computer Networks*, pp33-46, 2000.
 [9] K. Delgadillo, "Cisco Distributed Director", White Paper, *Cisco Systems, Inc.*, 1999.
 [10] M. Sayal, Y. Breitbart, P. Scheuermann, and R. Vingralek, "Selection Algorithms for Replicated Web Servers", In *Proceedings of the Workshop on Internet Server Performance*, 1998.
 [11] RL Carter and ME Crovella, "Server Selection Using Dynamic Path Characterization in Wide-Area Networks", In *Proceedings of IEEE INFOCOM*, 1997.
 [12] The RouteViews Project, "http://www.routeviews.org".
 [13] H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger, "On Inferring AS-Level Connectivity from BGP Routing Tables", In *Proceedings of ACM Internet Measurement Workshop*, 2001.
 [14] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The Impact of Routing Policy on Internet Paths", In *Proceedings of IEEE INFOCOM*, 2001.
 [15] L. Subramanian, S. Agarwal, J.Rexford, and RH Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points", In *Proceedings of IEEE INFOCOM*, 2002.
 [16] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (RFC1771)", 1995.

[17] D. Mosberger and T. Jin, "httpperf--A Tool for Measuring Web Server Performance", *Hewlett-Packard Research Labs*, 1998.

김 선 호(Seon-ho Kim)



1987년 2월 : 이화여자대학교 수학교육전공 학사

1992년 9월 : 이화여자대학교 교육대학원 전자계산교육전공 석사

1987년 1월~1989년 12월 : 대우전자부품(주) 전산실

1990년 1월~1993년 10월 :

한국생산성본부 정보화사업부

1999년 3월~현재, 숭실대학교 컴퓨터학과 박사과정

1998년 3월~현재, 동덕여자대학교 정보과학대학 강의전임교수

<관심분야> Internet Protocol, Mobile IP, CDN, DRM

고 훈 (Hoon Ko)



1998년 2월 : 호원대학교 컴퓨터학과 졸업 학사

2000년 2월 : 숭실대학교 컴퓨터학과 통신연구실 석사

2002년 2월 : 숭실대학교 컴퓨터학과 통신연구실 박사수료

2000년 5월 - 2002년 7월 (주) 지오나스 선임연구원

2002년 9월~현재 : 대진대학교 컴퓨터공학과 강의 교수

2003년 1월~현재 : 한국정보보호학회 편집위원

<관심분야> 멀티캐스트, Mobile IP, 인터넷 보안, PKI.

신 용 태(Yong-tae Shin)



1985년 2월 : 한양대학교 산업공학과 학사

1990년 : Univ. of Iowa 컴퓨터학과 석사

1994년 : Univ. of Iowa 컴퓨터학과 박사

1994년~1995년:Michigan State Univ. 전산학과 객원 교수

1995년 3월~현재 : 숭실대학교 컴퓨터학부 부교수

<관심분야> 멀티캐스트, 그룹통신, 인터넷 보안, 이동 인터넷 통신