

복제 웹 서버 환경에서 적응력 있는 서버 선택 메커니즘

(Adaptive Server Selection Mechanism in the
Replicated Web Server Environment)

김 선 호 [†] 신 용 태 ^{**}
(Seonho Kim) (Yongtae Shin)

요 약 최근 인터넷 사용자와 대용량 콘텐츠의 급증으로 서버와 네트워크의 부하가 가중되고 서비스의 질이 떨어지는 문제가 발생하였다. 이러한 문제의 해결로 콘텐츠를 다수의 지역 서버에 복제하여 서비스하고자 하는 기술들이 대두되고 있고, 이런 환경에서는 클라이언트의 요청에 대해 서비스할 적절한 서버를 선택하는 것이 매우 중요한 문제이다. 그러므로 본 연구에서는 클라이언트의 요청 시점에서 인터넷 토폴로지 상 가깝고 부하가 적은 서버가 서비스할 수 있도록 하는 서버 선택 메커니즘을 설계하여 제안한다. 본 연구는 계속적으로 증가하는 대용량, 실시간 멀티미디어 콘텐츠의 빠르고 안정적인 서비스를 가능하게 할 것이며 이로 인해 디지털 콘텐츠 서비스의 새로운 비즈니스 모델 창출에 크게 기여할 수 있을 것이다.

키워드 : 분산, 복제 서버, 리다이렉션

Abstract A rapid growth of the Internet user and rich media content cause an excessive server load and high network traffic, and thus it decreases the quality of service. A solution to this problem is to distribute the content on multiple replicated servers. However, in this circumstance, clients face additional task of selecting the best server to provide proper service which clients request.

This paper is intended to propose an adaptive server selection mechanism for a client request based on distance and server load. This will offer a fast and scalable service in response to the increase of massive and realtime multimedia content and contribute to creating a new business model of digital content service.

Key words : Distribution, Replicated Web Server, Redirection

1. 서 론

최근 인터넷 사용자 수와 대용량 디지털 콘텐츠의 급증으로 서버와 클라이언트 간의 네트워크 상에 상당한 부하가 발생하고 있으며 이러한 현상은 서버와 네트워크에 병목 현상을 야기 시켜 클라이언트에 대한 서비스 질을 현격하게 저하시키는 중요한 요인으로 작용하고 있다.

이와 같은 서버의 부하와 네트워크 지연을 인터넷상에 콘텐츠를 분산시킴으로써 해결하기 위한 방안으로

캐싱(Caching), P2P(Peer to Peer), CDN(Content Distribution Network)과 같은 방법들이 제안되어 사용되고 있다. 데이터 저장 비용이 네트워크 회선 비용보다 저렴한 현실을 감안할 때 이러한 콘텐츠 분산 기술은 고품질의 인터넷 서비스를 위해 매우 현실적인 방법이라 할 수 있다.

분산된 지역 서버에 콘텐츠를 복제하여 다수의 복제 서버를 사용하는 환경에 있어서는 클라이언트가 어떤 복제 서버에 접속하는가에 따라 응답시간 등 서비스의 질이 달라질 수 있다. 그러므로 이러한 환경에서 가장 중요한 이슈는 복제 서버 중 가장 적절한 서버로 클라이언트의 요청을 전달하여 빠르고 안정적인 서비스를 받도록 하는 것이다[1].

적절한 복제 서버 선택을 위한 메트릭으로 인접도, 서버 부하, 네트워크 트래픽 등을 고려한 여러 연구들이 진행되어 왔으나[2-8] 수시로 변하는 웹 서버의 상태나

· 본 논문은 한국과학재단 특장기초(과제번호 :R01-2001-000-00362-0) 연구비 지원에 의해 수행되었음

† 정 회 원 : 동덕여자대학교 정보과학대학 교수
shkim98@dongduk.ac.kr

** 총신위원 : 숭실대학교 컴퓨터학부 교수
shin@comp.ssu.ac.kr

논문접수 : 2003년 9월 9일

심사완료 : 2004년 4월 1일

토폴로지의 변화를 적절히 반영하지 못하고 있다. 그러므로 본 연구에서는 클라이언트의 요청에 대해 인터넷 토폴로지와 서버의 상태 변화에도 적응력 있는 서버 선택 메커니즘을 제안하고 구현하였다.

제안하는 메커니즘에서는 리다이렉터가 클라이언트와 인터넷 토폴로지 상 가깝고 부하가 적은 웹 서버를 선택하는 역할을 한다. 클라이언트의 요청을 받은 리다이렉터는 클라이언트가 속한 AS¹⁾ 또는 클라이언트와 가까운 AS에 복제 서버가 있는지를 탐색한다. 복제 서버가 있는 AS에서는 대표 서버를 두어 AS내의 복제 서버들의 부하 정보를 기록하도록 하고 리다이렉터의 요청에 부하가 가장 적은 복제 서버의 주소로 응답한다. 가까운 AS의 기준은 BGP(Border Gateway Protocol)[12] 라우팅 테이블을 이용하여 AS Path가 짧은 것으로 하며 서버의 부하는 서버의 초기 성능과 가용 메모리 양, 현재 연결된 커넥션 수를 기준으로 한다.

이러한 메커니즘은 클라이언트의 요청에 대해 가깝고 부하가 적은 복제 서버가 응답함으로써 서버의 부하를 줄이고 네트워크의 전송율과 서비스 속도를 향상시킬 수 있다.

2. 관련연구

복제 웹 서버 환경에서 서버 선택을 위한 기존 연구들은 다음과 같다.

[2]에서는 Round Robin DNS를 사용하여 클라이언트의 요청이 대해 여러 대의 서버에 동일한 비율로 전송될 수 있도록 차례로 웹 서버 IP 주소를 반환하는 것이다. 구현이 간단하고 가장 일반적인 방법이다. 그러나 이것은 현재 웹 서버의 상태는 고려하지 않고 단순히 하나씩 차례로 선택을 함으로써 각 서버에 접속하는 것을 공평하게 하고자 하는 것이다. 동질의 웹 서버 시스템에서는 Round Robin을 사용하여 트래픽을 분산시킬 수 있지만 부분적인 부하 분산만 가능하다.

[3]에서는 동적으로 변하는 네트워크 상태를 고려하기 위해 클라이언트가 대상 서버들에게 ping을 하여 일정 시간 기다린 후 응답이 온 서버들 중 클라이언트와의 대역폭이 가장 좋은 서버를 선택하였다. 이것은 기존의 과거 전송 데이터에 의한 정적인 대역폭만을 고려한 것에 비해 좋은 성능을 보였으나 동적인 데이터 수집을 위한 오버헤드가 큰 단점이 있다.

[4]에서는 사용자와 가장 가까운 서버를 선택하도록 하기 위해서 시간에 따라 변하는 인터넷 토폴로지를 반영하여 AS Path 길이가 짧은 서버를 선택하였다. 그러

나 AS의 범위가 매우 다양하며 거리만을 고려한 서버 선택은 서비스의 질을 향상하기 위한 좋은 방법이라고 할 수 없다.

[5]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.

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

[7]에서는 Hops, RTT 등 여러 서버 선택 메트릭 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 응답시간이 아닌 비용 합수로 HTTP 지연시간을 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.

[8]의 연구에서는 사용 가능한 대역폭과 RTT를 측정하는 툴을 사용하여 파일 크기가 작은 경우는 RTT를 측정하여 서버를 선택하고 파일 크기가 큰 경우는 RTT와 사용가능한 대역폭을 함께 고려한 동적인 서버 선택 알고리즘을 택하여 과거 정적인 접근 메트릭과 HTTP 전송시간을 비교한 결과 50% 이상 향상되었다. 그러나 대역폭을 고려하는 경우 시간과 네트워크 트래픽 면에서 오버헤드가 크기 때문에 적당하지 않다.

3. 리다이렉션 시스템의 설계

3.1 설계 목표 및 시스템 모델

적절한 서버를 선택하기 위한 리다이렉션 시스템은 다음 조건을 만족해야 한다.

- 네트워크 계층의 어떠한 수정이나 지원 없이 응용 계층의 프로그램 등에 의해 동작해야 한다.
- 과도한 네트워크나 서버의 부하를 생성해서는 안 된다.
- 다양한 조건과 시간의 변화에 대해서도 안전하게 동작해야 한다.
- 확장 가능하고 쉽게 적용할 수 있어야 한다.

이러한 조건을 만족하기 위해 제안하는 시스템은 다음을 가정한다.

- 각 AS별로 AS의 크기, 클라이언트 수 등에 따라 복제 서버의 수와 위치가 적절히 배치되어 있다.
- 각 AS에는 0~n 개의 복제 서버가 있을 수 있으며

1) Autonomous System. 관리적 측면에서 한 단체에 속하여 관리되고 제어됨으로써 동일한 라우팅 정책을 사용하는 네트워크 또는 네트워크 그룹

한 AS 내에 복제 서버가 2개 이상일 때에는 대표 서버를 둔다.

- 복제 서버들은 웹 서비스를 하는 서버이며 서로 같은 콘텐츠를 저장하고 있고 원본 콘텐츠와 동기화를 유지하고 있다.

이런 가정 하에 제안하는 시스템 모델의 개요는 다음과 같다.

- AS를 하나의 그룹으로 보고 인터넷상에 분산되어 있는 복제 서버들을 AS별로 그룹화 한다.
- 각 AS의 대표 서버는 복제 서버들의 초기 성능을 기억하고 부하를 주기적으로 점검한다.
- 리다이렉터는 클라이언트의 요청을 가까운 AS의 부하가 적은 복제 서버로 리다이렉션 한다.

이러한 시스템은 가까운 AS 내의 복제 서버들 중에서 현 상태의 부하가 가장 적은 서버를 동적으로 선택함으로써 네트워크의 상태를 반영한 적응력 있는 웹 서버 선택이 가능하여 네트워크 인프라의 수정 없이 클라이언트 수의 증가에도 안정적이고 확장성 있는 서비스를 제공할 수 있다.

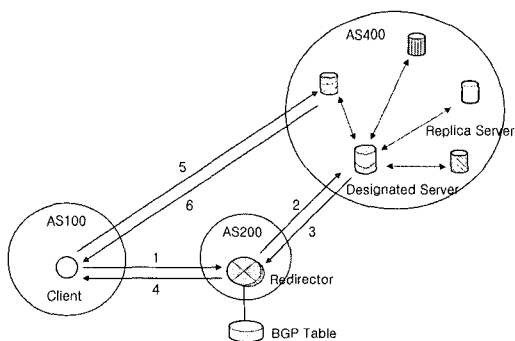


그림 1 시스템 모델

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

- 클라이언트(Client)
웹상에서 객체를 요청하는 주체로서 일반 사용자의 웹 브라우저를 의미한다.
- 리다이렉터(Redirector)
클라이언트로부터의 요청을 받아들여 BGP 라우팅 테이블을 이용하여 클라이언트와 가까운 AS를 찾아내고 AS의 대표에게 부하가 적은 서버의 IP주소를 요청하여 클라이언트에게 응답하는 클라이언트와 대표 서버 사이의 중간 주체이다.
- 대표 서버(Designated Server)
AS의 대표 서버로서 내부 서버들의 부하 상태를 점검하여 리다이렉터의 요청에 응답하는 주체이다.

- 복제 서버(Replica Server)
원본 서버의 콘텐츠를 복사하고 있으며 클라이언트로부터 콘텐츠 요청을 받아 응답을 보내주는 주체이다.

- BGP Table
BGP 라우팅 테이블을 저장하고 있는 파일이다.

3.2 서버 선택 알고리즘

제안하는 리다이렉션 시스템의 서버 선택은 다음 2단계의 정책에 의해 결정된다.

- 1단계 : 클라이언트와 토폴로지 상 가까우며 복제 서버를 포함하고 있는 AS의 선택
- 2단계 : AS내의 복제 서버들 중 성능이 좋고 현재의 부하가 적은 서버 선택

제안한 리다이렉션 시스템의 서버 선택 과정은 다음과 같다.

- 1) 클라이언트는 지역 DNS에게 도메인 이름 해석을 요구한다.
- 2) 지역 DNS는 매핑 테이블에 의해 리다이렉션 시스템의 리다이렉터의 IP주소로 응답 메시지를 생성하여 클라이언트에게 전달한다.
- 3) 클라이언트는 응답 메시지를 분석하여 해당 리다이렉터에게 서비스를 요청한다.
- 4) 리다이렉터는 클라이언트의 IP 주소를 AS 번호로 변환한다.
- 5) 리다이렉터는 복제 서버 리스트를 검색하여 해당 AS 번호의 대표 서버가 있는지 확인한다.
 - 5-1) 있으면, 대표 서버에게 서비스 받을 웹 서버의 IP주소를 요청한다.
 - 5-2) 없으면, AS 그래프를 탐색하여 클라이언트와 가까운 AS를 찾은 후 5)번부터 반복한다.
- 6) 대표 서버는 내부 서버들의 성능계수와 가용 메모리와 현재 연결된 커넥션 수를 고려하여 부하가 가장 적은 웹 서버를 선택하여 IP 주소를 반환한다.
- 7) 리다이렉터는 해당 IP주소를 클라이언트에게 전송한다.
- 8) 클라이언트는 선택된 서버와 연결을 설정하여 콘텐츠 전송을 요청한다.

3.2.1 1단계 : 인접한 AS의 선택

서버 선택의 1단계로 클라이언트와 가까운 서버를 찾는다. 인접 메트릭으로는 클라이언트와 서버 간의 AS Path를 사용한다. 목적지에 대한 AS Path의 길이가 짧은 경우 가까운 곳에 있다고 판단하여 클라이언트와 Path가 가장 짧은 AS를 선택한다. 이것은 AS 내부에서의 지연이나 손실은 거의 없는 것을 가정하는 것이다 [4].

AS Path 탐색은 실제 인터넷의 토폴로지를 이용하기 위해 NLNR에 의해 공개적으로 제공되는 BGP 라우

팅 테이블을 이용한다. 이것은 RouteViews 프로젝트[9]에 의해 구축되었는데 Oregon 라우터 서버는 인터넷상의 37개 BGP 라우터로부터 BGP 라우팅 테이블 정보를 가져오며 매 2시간 마다 갱신된다. 이 테이블은 인터넷 토폴로지를 생성하거나 이용하고자 하는 많은 연구들에 의해서 사용되고 검증되어 왔다[10,11].

본 연구에서도 <http://archive.routeviews.org>로부터 주기적으로 BGP 라우팅 정보를 업데이트 하여 수집한다. BGP 라우팅 테이블은 하루에 한번씩 읽어 오도록 하며 테이블을 읽어 AS 그래프를 만드는 시간 동안의 지연을 줄이기 위해 종전의 그래프를 그대로 유지하고 있다가 최신 그래프가 다 형성된 후에 업데이트한다.

AS path로 AS 그래프를 구성하여 연결 리스트로 저장한 후 BFS(Breadth First Search) 알고리즘을 이용하여 복제 서버가 있는 AS를 탐색한다. 클라이언트의 AS번호도 BGP 테이블을 이용하여 prefix 기반의 트리를 만들고 이진 탐색을 이용하여 탐색할 수 있다[13]. AS 그래프 G는 (V, E)로 구성되며 V는 노드 집합, E는 링크 집합이다. 각 링크에는 가중치를 두지 않고 루프를 방지하기 위해 자신의 AS 노드를 만나면 탐색을 중지한다. 두 노드 $v, w \in V$ 를 연결하는 링크 $e(v, w)$ 는 오직 한개만 존재하며 $e(v, w) \neq e(w, v)$ 인 방향성을 갖는 그래프이다.

AS 연결 리스트로부터 인접한 AS를 탐색하는 과정은 다음과 같다.

AS Graph Search Algorithm

```

start from Client's AS, ASc ;
if ASc has a replica break;
mark ASc as visited ;
for (all AS nodes adjacent to ASc)
    Enqueue if ASi is not visited yet ;
    if ASi has Replica or all nodes are visited
        break ;
    Dequeue(ASi) ;
    mark ASi as visited ;
endfor
    
```

3.2.2 단계 : 부하가 적은 복제 서버 선택

리다이렉터에 의해 가까운 AS의 복제 서버들이 선택되었으나 그 중 하나의 서버로 접속이 집중되는 경우는 가깝지만 신속한 서비스를 할 수 없을 것이다. 그러므로 서버의 부하 정도와 사용 가능한지(active)에 대한 정보를 알 필요가 있다. 웹 서버의 부하 정보로 CPU와 메모리 등의 시스템 자원과 서비스 중인 작업 수를 고려할 수 있는데 웹 서비스의 경우 요청을 처리하는데 있어서는 CPU의 속도 보다는 가용 메모리에 따라 성능이 크게 좌우되므로[14], 가용 메모리를 현재 연결된 커넥

션 수로 나눈 값을 현재의 서버 부하로 측정한다. 각 AS의 대표 서버는 해당 AS내의 복제 서버들의 초기 성능계수를 기억하고 서버들의 현재 부하를 주기적으로 점검하여 리다이렉터의 요청 시 식 (1)과 같이 초기 성능계수와 현재 부하를 고려하여 부하가 가장 적고 성능이 좋은 서버를 선택하여 응답한다.

$$Server\ Performance = P_{coefficient} * L_{current} \quad (1)$$

- 성능계수($P_{coefficient}$) : 성능계수는 초기 성능과 서비스 양이 각기 다른 웹 서버가 시간 당 처리하는 평균 클라이언트의 요청 수로 정의한다. 성능계수를 구하기 위하여 웹 서버 성능 분석 툴인 [httpperf\[18\]](http://perf[18])를 이용하여 각 복제 서버의 요청 처리 능력을 측정한다.
- 현재의 부하($L_{current}$) : 현재의 부하는 현재 웹 서버의 가용 메모리 양(free memory)을 서버에 연결되어 있는 서비스 수(connection)로 나눈 값으로 정의한다.

Best Server Selection Algorithm

```

for (i = 1 ; i <= Number of Replica Servers ; i++) {
    Current_Performance =
        Performance_Coefficient*(Free_Memory/Connections);
    if i == 1
        Better_Replica = Replica ;
    if Current_Performance > Last_Performance
        Better_Replica = Replica ;
    Last_Performance = Current_Performance;
}
Best_Replica = Better_Replica ;
    
```

3.3 예상되는 응답시간

제한한 2단계 서버 선택 알고리즘으로 서비스 할 경우 클라이언트의 콘텐츠 요청으로부터 응답이 완료될 때까지의 전체 시간은 식 (2)와 같이 나타낼 수 있다. 여기에서 HTTP/1.1 프로토콜을 사용하는 경우 연결 지연시간은 매우 미미한 것으로 보고 별도로 구분하여 고려하지 않았다.

$$T_{Total} = T_{Distance} + T_{Load} + T_{Redirection} + T_{Download} \quad (2)$$

T_{Total}	: 전체 전송 시간
$T_{Distance}$: 가까운 AS의 대표 서버를 탐색하는데 걸리는 시간
T_{Load}	: 부하가 적은 복제 서버를 선별하는데 걸리는 시간
$T_{Redirection}$: 클라이언트가 요청을 리다이렉션 하는데 걸리는 시간
$T_{Download}$: 클라이언트가 서버로부터 패킷을 모두 전송 받는데 걸린 시간

4. 구현

4.1 구현 환경 및 모델

Linux 상에서 C언어와 PHP언어를 사용하여 개발하였다. 그리고 Load Collector와 Load Sender의 통신을 위하여 확장성이 높은 XML을 사용하였다.

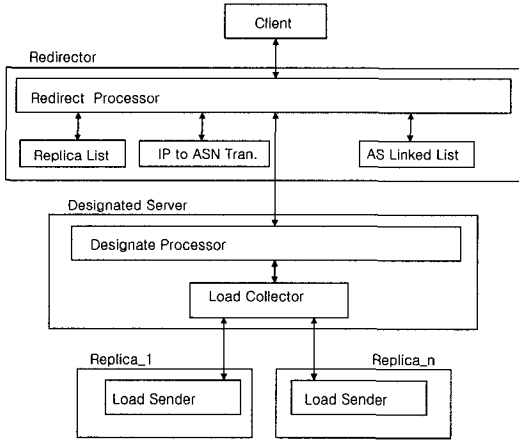


그림 2 구현 모델

그림 2에 표시된 구현 모델의 구성 요소들과 구현 방법을 기술한다.

- Redirect Processor

Redirect Processor는 클라이언트의 요청이 오면 IP-ASN Translation을 통해 클라이언트의 AS 번호를 알아내고 Replica List에서 해당 AS의 대표 서버 주소가 등록되어 있는지 확인하고 없으면 AS Linked List를 탐색하여 가장 가까운 AS 번호를 알아낸다. 가까운 AS의 대표 서버에게 해당 그룹 서버들의 부하 정보를 요청하여 부하가 적은 복제 서버의 IP주소를 받아 Client에게 보내어 요청을 해당 복제 서버로 리다이렉트하도록 한다.

- Replica List

AS 번호와 각 AS의 대표 서버의 IP 주소 매핑을 유지한다.

- IP to ASN Translation

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

- AS Linked List

BGP 라우팅 테이블로 AS Linked List를 구성하여 해당 AS와 가장 가까운 AS를 BFS(Breadth First Search) 알고리즘을 이용하여 탐색한다.

- Designated Processor

Designated Processor는 내부 복제 서버의 목록과 각 서버의 성능계수를 기억한다. Designated Processor는 Load Collector를 사용하기 위해 시스템 호출을 하며 복제 서버의 부하가 가장 작은 복제 서버의 IP 주소를 반환한다.

- Load Collector

Load Collector는 그룹의 내부 서버들에 접속하여 Load Sender로부터 현재의 부하 상태를 수집하여 대표 서버가 최적의 서버를 판단하는 정보를 제공한다.

- Load Sender

Load Collector로부터 부하 정보를 원하는 커넥션이 발생하면 현재의 서버 부하 정보를 측정하여 Load Collector에 전달한다. Load Sender는 서버 프로그램과 커맨드 프로그램으로 구성된다. 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 파일을 이용하여 현재 연결되어 있는 커넥션 수와 가용 메모리 양을 구하여 이 결과 값을 XML 형태로 바꾸어 Load Collector에 전달하고 작업을 끝낸다.

4.2 대표 서버와 멤버 서버간의 메시지 교환

Designated Processor에서 멤버 서버로 소켓 통신을 하여 해당 서버의 부하를 출력하기 위한 프로세스는 소켓 통신 프로토콜에 XML을 이용한 C 프로그램으로 한다. Load Collector는 받은 정보 중에서 response_text를 파싱하여 리다이렉터에게 보낸다. Load Collector와 Load Sender의 메시지 교환 방식은 다음과 같다.

Message of Load Collector

```
<Redirect_envelope>
  <header><version>1.0</version>
</header>
  <body>
    <command>
      <item key="protocol">XCP</item>
      <item key="action">SEND</item>
      <item key="object">GET_LOAD</item>
    </command>
  </body>
</Redirect_envelope>
```

Message of Load Sender

```
<Redirect_envelope>
  <header> <version>1.0</version>
</header>
  <body>
    <command>
```

```

<item key="protocol">XCP</item>
<item key="action">REPLY</item>
<item key="object">GET_LOAD</item>
</command>
<content>
<item key="response_code">1</item>
<item key="response_text">connections=39</item>
<item key="response_text">memory=378940</item>
<item key="is_success">1</item>
</content>
</body>
</Redirect_envelope>
    
```

5. 실험 및 분석

본 실험에서는 실험의 편의를 위하여 BGP 테이블을 이용하여 임의로 클라이언트와 대상 서버를 선택하여 서버 선택의 1단계인 가까운 AS 탐색 시간을 측정하고, 2단계 서버 선택과 전송 시간을 따로 측정하여 더하였다. 2단계 실험에서는 클라이언트와 복제 서버들을 같은 AS에 두고 AS내의 복제 서버의 수는 3대로 하여 주말, 주중, 주간, 야간 시간대로 나누어 각 10회씩 반복하여 측정하고 평균값을 구하여 반영하였다.

비교 대상으로는 인접 메트릭으로는 AS를, 부하 메트릭으로는 Round Robin을 사용한 방법을 택하여 AS와 서버의 성능을 고려한 제안하는 방법과 성능을 비교하였다. 특정 서버에 클라이언트의 연결이 집중되어 있는 경우 부하가 분산되는 정도와 전송할 파일의 크기를 1KB에서 1MB로 증가시켰을 경우와 동시 접속자수를 10에서 1000으로 증가시킬 경우 평균 응답시간을 측정하여 비교하였다.

5.1 httpperf를 이용한 복제 서버들의 성능계수 측정

웹 서버 성능 측정 툴인 httpperf를 사용하여 복제 서버들의 성능계수를 측정한다. 본 연구에서는 표 1의 서버들을 대상으로 실시하였다. 성능계수는 하나의 요청을 처리하는데 걸리는 시간으로 한다. 실험을 위해 send-buffer는 4096byte, receive-buffers는 16384byte, 커넥션 수는 1000개로 정의하여 처리하는데 걸리는 시간을 측정하였다. 다른 값을 주어 여러 번 측정하여도 비슷한 비율의 값을 보였다.

표 1 실험 서버 및 성능계수

	CPU	RAM	성능계수 ($P_{coefficient}$)
Replica_1	Pentium III 433KHz	512MB	0.23
Replica_2	Pentium III 1333KHz	512MB	0.71
Replica_3	Pentium II 333KHz	128MB	0.37

5.2 실험결과

실험을 위하여 복제 서버_1에는 100개의 커넥션이, 복제 서버_2에는 50개 커넥션이 지속적으로 연결되도록 하고 복제 서버_3은 서비스 중인 커넥션이 하나도 없도록 한 상태에서 새로운 1000개의 요청을 할 경우 각 요청들이 어떤 복제 서버로 전달되는지를 체크하여 서버 간에 부하가 분산되는지 알아보았다.

제안한 리다이렉션 시스템에서 클라이언트의 요청은 성능계수와 현재 부하를 곱하여 가장 작은 값을 갖는 웹 서버로 전달된다. 반면 AS에 의한 거리만 고려하는 경우는 최악의 경우 1000개의 모든 요청이 커넥션이 가장 많은 복제 서버_1로 전달될 수 있으며, 좀 더 나은 방법으로 Round Robin 사용하는 경우에도 서버의 성능이나 현재의 부하에 상관없이 차례로 요청을 전달할 것이다. 본 실험에서는 해당 AS내에서 Round Robin으로 요청을 전달할 경우를 가정하여 부하 분산 정도와 응답 시간을 측정하여 비교하였다.

비교한 결과 그림 3에서와 같이 Round Robin으로 요청을 전달한 경우는 복제 서버의 성능이나 부하에도 불구하고 요청이 일정한 비율로 각 서버에게 전달된 반면 제안된 리다이렉션 시스템에서는 성능과 현재의 부하를 고려하여 여전히 성능이 좋은 복제 서버_2로 주로 요청이 전달되는 것을 알 수 있다. 그러므로 제안하는 시스템은 이질적인 서버의 성능과 동적인 부하를 반영하여 요청을 분산한다는 것을 알 수 있다.

파일 크기와 동시 접속자수에 따른 응답시간에 있어서는 그림 4, 그림 5에서와 같이 AS내에서 Round Robin에 의해 선택된 복제 서버로부터 전송을 받는 경우와 제안한 리다이렉션 시스템에 의한 전송을 비교한 결과 파일의 크기가 클수록, 동시 접속자수가 많을수록 제안한 시스템이 좋은 성과를 나타내었다.

제안한 시스템은 특정 서버에 장애가 발생한 경우에도 문제없이 동작하지만 Round Robin의 경우 장애가 있는 서버가 선택되어 문제를 감지하여 다른 서버를 다

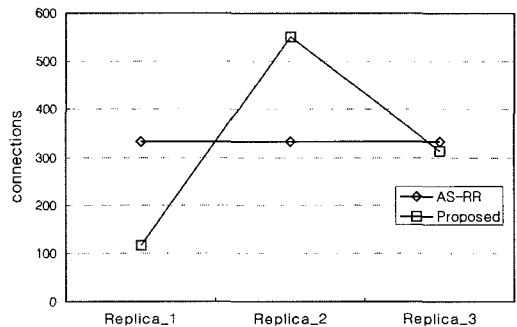


그림 3 부하 분산 정도

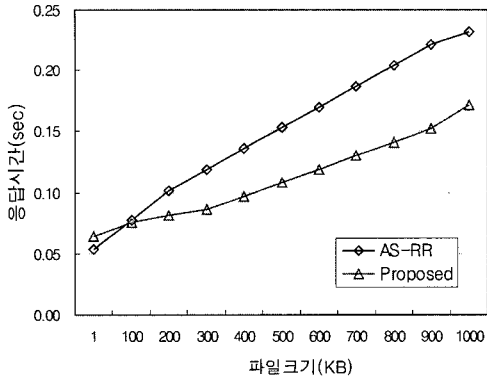


그림 4 파일크기에 따른 응답시간

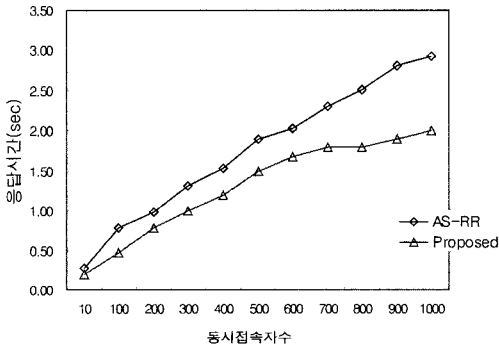


그림 5 동시 접속자 수에 따른 응답시간

시 선택하게 되는 경우 더욱 좋지 못한 결과가 나올 것이다.

6. 결론 및 향후 연구 계획

본 연구에서는 분산된 복제 웹 서버 환경에서 적절한 서버를 선택하기 위한 리다이렉션 시스템을 설계하고 제안하였다. 클라이언트의 요청에 대해 먼저 시스템의 리다이렉터가 BGP 라우팅 테이블을 이용하여 사용자와 거리가 가까운 AS의 대표 서버를 선택하고 대표 서버를 통해 AS내에서 부하가 가장 적은 복제 서버를 최종적으로 선택하여 요청을 리다이렉션 할 수 있도록 하였다.

구현 및 분석 결과 제안한 서버 선택 메커니즘이 거리와 Round Robin에 의한 부하 분산만을 적용한 메커니즘에 비해 서버의 상태에 따라 적응력 있게 부하를 분산하며 응답시간에 있어서도 파일의 크기가 클수록, 동시 접속자수가 많을수록 좋은 성능을 나타내었다.

본 연구는 클라이언트 프로그램의 수정 없이 클라이언트 입장에서 동적으로 가장 좋은 서버를 선택할 수 있는 모델을 제안한 것에 큰 의미가 있다고 본다. 그러나 이러한 메커니즘은 리다이렉터가 모든 요청과 응답

을 제어하므로 복제 서버의 상태 파악이 용이하고 네트워크 운영체제에 독립적으로 동작할 수 있는 장점이 있으나 리다이렉터가 모든 요청을 처리하므로 리다이렉터로의 병목현상이 발생할 수 있다. 그러므로 향후 적절한 수의 리다이렉터 선택과 배치 문제가 연구되어야 할 것이며 리다이렉션 시스템에 보안 메커니즘을 추가하기 위한 연구가 이루어져야 할 것이다.

참고 문헌

- [1] Peng G., "CDN: Content Distribution Network," In Stony Brook University Tech. Reports, 2003.
- [2] Brisco, P.T. "RFC1794: DNS support for Load Balancing," 1995.
- [3] Dykes, SG., Robbins, KA., and Jeffery, CL., "An Empirical Evaluation of Client-side Server Selection Algorithms," In Proceedings of INFOCOM, Vol.3, pp. 1361-1370, 2000.
- [4] McManus, PR. "A Passive System for Server Selection within Mirrored Resource Environments Using AS Path Length Heuristics," Applied Theory Communications Inc., 1999.
- [5] Rodriguez, P. and Sibal, S., "SPREAD: Scalable Platform for Reliable and Efficient Automated Distribution," pp. 33-46, Computer Networks, 2000.
- [6] Delgadillo, K. "Cisco Distributed Director," White Paper, Cisco Systems Inc., 1999.
- [7] Sayal, M., Breitbart, Y., Scheuermann, P., and Vingralek, R., "Selection Algorithms for Replicated Web Servers," In Proceedings of the Workshop on Internet Server Performance, 1998.
- [8] Carter, RL. and Crovella, ME., "Server Selection Using Dynamic Path Characterization in Wide-Area Networks," In Proceedings of INFOCOM, 1997.
- [9] The RouteViews Project, <http://www.routeviews.org>.
- [10] Chang, H., Govindan, R., Jamin, S., Shenker, SJ., and Willinger, W., "On Inferring AS-Level Connectivity from BGP Routing Tables," In Proceedings of ACM Internet Measurement Workshop, 2001.
- [11] Tangmunarunkit, H., Govindan, R., Shenker, S., and Estrin, D., "The Impact of Routing Policy on Internet Paths," In Proceedings of INFOCOM, 2001.
- [12] Rekhter, Y. and Li, T., "A Border Gateway Protocol 4 (RFC1771)," 1995.
- [13] Tzeng, H., "Longest Prefix Search using Compressed Trees," In Proceedings of Globecom, 1998.
- [14] 권원상 역, "웹 성능 최적화", 한빛미디어, 2003.



김 선 호

1987년 이화여자대학교 사범대학 수학교육전공 학사. 1992년 이화여자대학교 교육대학원 전자계산교육전공 석사. 2004년 숭실대학교 컴퓨터학과 컴퓨터통신전공 박사. 1987년~1989년 대우전자부품(주) 전산실. 1990년~1993년 한국생산성본부 정보화사업부. 1998년~현재 동덕여자대학교 정보과학대학 강의전임교수. 관심분야는 Internet Protocol, Mobile IP, CDN, DRM



신 용 태

1985년 한양대학교 산업공학과 학사. 1990년 Univ. of Iowa 컴퓨터학과 석사. 1994년 Univ. of Iowa 컴퓨터학과 박사. 1994년~1995년 Michigan State Univ. 전산학과 객원교수. 1995년 3월~현재 숭실대학교 컴퓨터학부 부교수. 관심분야는 멀티캐스트, 그룹통신, 인터넷 보안, 이동 인터넷 통신