

확장 DNS를 이용한 분산 웹 서버의 동적 부하분산 기술

김선영*, 문종배*, 김명호**

승실대학교 컴퓨터학과

e-mail:{ksy, comdoct}@ss.ssu.ac.kr, kmh@computing.ssu.ac.kr

Dynamic Load Balancing Technique For Distributed Web Server Using Extended DNS

Sun-Young Kim*, Jong-Bae Moon*, Myung-Ho Kim**

Department of Computer Science, Soongsil University

요 약

컴퓨터 네트워크 및 WWW의 발전으로 인터넷 사용자 및 웹 서버의 수는 기하급수적으로 증가하고 있다. 그러나 네트워크의 발달에도 불구하고 웹 서버에 접속하는 사용자의 수가 많아짐에 따라 웹 서버에 병목현상이 발생하고 있다. 이러한 문제를 해결하기 위해 클러스터 시스템이 사용된다. DNS기반 클러스터는 분산 웹 서버 환경에 적합하지만 웹 서버의 상태를 고려하지 않는다. 따라서 본 논문에서는 DNS의 단점을 보완한 부하분산 시스템을 제안한다. 이 시스템은 부하분산 시스템을 따로 설치하지 않고 기존 DNS 프로토콜의 수정 없이 부하분산 모듈을 추가하여 웹 서버의 부하를 분산한다. 본 논문에서 구현한 시스템을 이용하여 분산 웹 환경에서 서버 부하를 고려하여 클라이언트의 요청을 적절히 분배할 수 있다.

1. 서론

컴퓨터 네트워크 및 WWW의 발전으로 인터넷 사용자 및 웹 서버의 숫자는 기하급수적으로 증가하고 있다. 이런 상황에서 멀티미디어 서비스를 제공하는 사이트들은 실시간으로 네트워크와 서버의 부하를 감지하여 지능적으로 분산함으로써 사용자에게 가장 빠르고 안정적인 서비스를 제공할 수 있도록 해야 한다. 이를 위해 웹 서버 클러스터 시스템을 많이 사용하고 있다.

웹 서버 클러스터 시스템에서는 웹 서버의 과부하를 해결할 수 있도록 여러 형태의 부하분산 방법들이 사용하고 있다[1,2,4]. 부하분산 방법은 요청을 서버들에게 분배하는 주체에 따라 크게 DNS, 디스패처, 서버 기반 방식으로 나눌 수 있다. DNS 기반 방식은 요청을 DNS 서버가 분배하는 방식[2]으로 부하를 분산하는 디스패처가 없이 라운드로빈 방식으로 부하를 분배하고 서버의 부하를 고려하지 않는다. 디스패처 기반 방식은 클라이언트의 요청을 한

서버로 모아서 적절한 알고리즘을 사용하여 부하를 분산하는 방법으로써 서버의 부하 정보를 고려하지 않거나 현재 서비스 중인 TCP연결 수만을 고려하여 부하분산의 불균형이 나타난다[4]. 서버 기반의 방식[5]은 사용자 요청을 받은 서버가 스케줄링 알고리즘에 따라 부하를 분산하는 방법으로 전면 서버들이 부하를 분산하는 과정에서 전면 서버 사이에 부하 불균형이 생길 수 있으며 사용자는 항상 최초 접속 시에 서비스를 받기 위하여 두 차례의 접속 요청을 해야 한다는 단점이 있다.

본 논문에서는 라운드로빈 DNS 방식의 정적인 부하분산의 한계를 보완하고 지역적으로 분산된 서버환경에서 부하를 적절하게 분산하기 위한 확장 DNS를 이용한 부하분산 시스템을 제안한다. 또한 기존 BIND와의 호환성을 위해 DNS 프로토콜의 수정 없이 부하분산 모듈을 추가하고 각 웹 서버의 부하를 고려하여 적절히 부하를 분산할 수 있도록 한다.

2. 관련 연구

2.1 라운드로빈 DNS

NCSA에서 처음으로 분산된 서버들의 부하분산을 위하여 DNS에서 라운드로빈 방식을 사용하였다[3]. 라운드로빈 방식은 DNS에 하나의 이름으로 여러 IP를 연결하여 순차적으로 한 번씩 요청에 대한 응답을 할당하는 방식이다. 이 방법은 설치가 간단하고 부하를 분산하는 디스패처가 없기 때문에 한 서버가 장애로 서비스를 할 수 없게 되어도 전체 시스템에 영향을 미치지 않는다. 즉, SPOF(Single Point Of Failure)가 발생하지 않는 장점이 있다. 그러나 DNS의 특성상 클라이언트에서 캐싱으로 인한 부하의 불균형이 발생한다[2]. 또한 DNS는 서버들의 상태를 파악할 수 없기 때문에 장애가 발생한 서버에도 사용자 요청을 할당하도록 하는 단점이 있다.

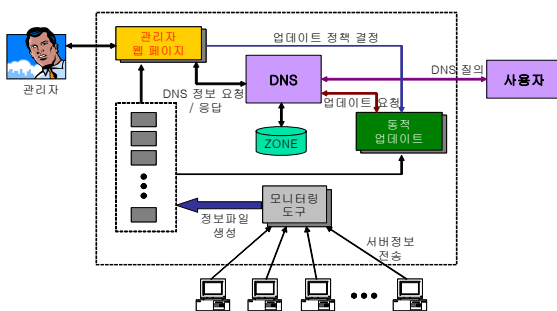
2.2 lbnamed

lbnamed는 스탠포드 대학에서 부하분산을 목적으로 펄(Perl)을 이용하여 만든 DNS이다[1]. lbnamed는 기존 DNS 프로토콜을 수정하지 않고 만들었으며, 기존 응용프로그램인 BIND와도 호환이 가능하도록 구현되었다. lbnamed는 서버들의 부하를 동적으로 파악하여 부하가 가장 작은 쪽으로 사용자 요청을 연결하는 방법을 사용한다. 그러나 펄로 작성되어 기존 BIND보다는 빠르지 못하고, 등록된 서버들의 다양한 부하를 고려하지 않아 사용자 요청이 다양하고 부하가 동적으로 변하는 환경에서는 적당하지 못하다.

3. 확장 DNS를 이용한 부하분산 시스템

3.1 전체 구조

확장 DNS를 이용한 부하분산 시스템의 구조는 그림 1에서와 같이 BIND, 동적 업데이트 모듈, 모니터링 도구, 관리도구로 구성되어 있다.



(그림 1) 확장 DNS를 이용한 부하분산 시스템 전체구조

모니터링 도구는 BIND에 등록된 지역적으로 분산되어 있는 서버들의 부하정보를 수집한다. 여기서 부하정보는 네트워크, CPU, 메모리 정보 등이 될 수 있다. 모니터링 도구는 모니터링 데몬과 모니터링 에이전트로 구성되어 있다. 동적 업데이트 모듈은 모니터링 데몬이 수집한 서버들의 부하 정보를 기반으로 최적의 서버를 선택하여 BIND의 zone 레코드 정보를 실시간 업데이트한다. 관리도구는 업데이트 정책을 정하고 이에 따라 최적의 서버 또는 그룹만을 등록하고 나머지는 리스트에서 삭제하도록 한다.

3.2 동적 업데이트 모듈의 설계 및 구현

동적 업데이트 모듈은 DNS에 등록된 서버들의 부하량을 측정하여 최적의 서버군을 유지하도록 한다. 서버들의 부하량은 CPU, 메모리, 네트워크의 사용량을 참조하여 사용한다. 최적의 서버군을 유지하는 기준은 분산된 서버들이 서비스하는 내용에 따라 달라진다. 대용량의 파일 전송을 하는 서비스일 경우 CPU와 메모리 사용률 증가보다 네트워크 사용률의 증가폭이 크며, 서버에서의 많은 처리를 요구하는 서비스일 경우는 네트워크 사용률 증가보다 CPU 사용률의 증가폭이 크다. 따라서 총부하량 계산에는 서비스마다 사용량의 증가폭이 큰 기준의 가중치를 높여서 계산한다. 서버들의 동적인 부하변화가 많은 환경에서는 zone파일 레코드의 수정이 많이 일어날 수 있다. 이 때 발생하는 부하를 줄이기 위하여 동적 업데이트 모듈을 BIND와 별개의 프로세스로 구성하며 기본적으로 최적의 서버군에서 라운드로빈 방식을 기반으로 레코드의 수정을 줄일 수 있도록 한다.

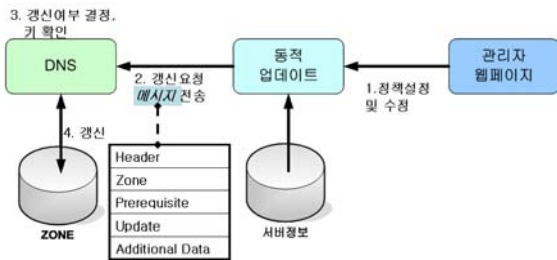
본 논문에서는 최적의 서버군 유지를 위해서 라운드로빈 방식을 기반으로 하고 방식은 다음과 같다.

- CPU 기반 방식: CPU 부하량이 최대 임계값을 넘는 서버들은 리스트에서 삭제하고 최소 임계값보다 작아지면 리스트에 추가하는 방식
- 네트워크 기반 방식: 네트워크 부하량이 최대 임계값을 넘는 서버들은 리스트에서 삭제하고 최소 임계값보다 작아지면 리스트에 추가하는 방식
- 메모리 기반 방식: 메모리 부하량이 최대 임계값을 넘는 서버들은 리스트에서 삭제하고 최소 임계값보다 작아지면 리스트에 추가하는 방식
- 복합 방식: CPU, 메모리, 네트워크의 부하의 비율의 합이 총 100이라 가정하고 관리자에 의해 그 비율이 설정되고 최대, 최소 임계값을 정하여 업

데이트하는 방식.

동적 업데이트 모듈은 관리자 웹 페이지에서 설정한 업데이트 정책에 의해 DNS 정보를 모니터링 도구에 의해 생성된 서버들의 정보 파일을 참조하여 실시간 업데이트한다.

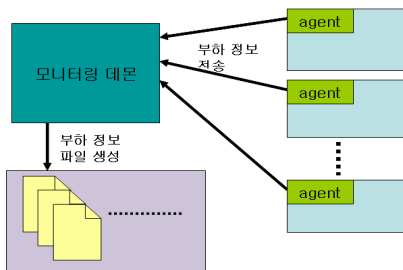
그림 2에서처럼 동적 업데이트 모듈은 DNS의 zone 파일을 직접 수정하지 않고 레코드를 갱신할 수 있도록 한다. 이를 위해 BIND의 동적 업데이트를 이용한다. 동적 업데이트 메시지는 보안을 위해 DNS 메시지의 추가 정보부분(additional data)에 RFC2845에 정의된 TSIG 레코드를 추가하여 보낸다. DNS는 갱신 요청 메시지를 받으면 TSIG Key를 확인하여 DNS에 대한 접근 권한이 있는지를 확인하고 권한이 있으면 Prerequisite 부분을 확인하여 요청한 메시지의 요구가 DNS에서 가능한지 판단한다. 예를 들면, DNS에 grid.or.kr이라는 도메인이 존재하지 않는데 www.grid.or.kr이라는 레코드를 추가하라는 메시지가 오면 갱신을 수행하지 않는다.



(그림 2) 동적 업데이트 모듈

3.3 모니터링 도구의 설계 및 구현

그림 3과 같이 모니터링 도구는 DNS에 등록된 서버들의 부하 상태 정보를 수집하는 역할을 한다. 모니터링 도구는 모니터링 데몬과 모니터링 에이전트로 구성된다.



(그림 3) 모니터링 도구의 구성

모니터링 데몬은 확장 DNS가 설치된 서버에서 실행되며, DNS에 등록된 각 서버들의 부하정보를 수집한다. 각 서버들의 부하정보는 5초마다 받아들이고, 각 서버의 IP주소에 해당하는 파일을 생성하

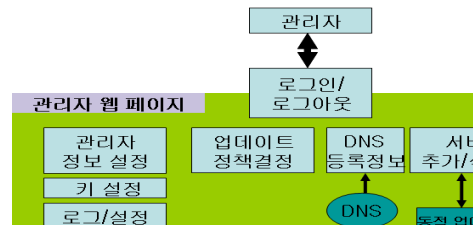
여 CPU, 메모리, 네트워크 정보를 저장한다. 에이전트에서 보내는 패킷이 UDP 패킷임을 고려하여 3번의 UDP 패킷이 도착하지 않았을 경우, 즉 15초 동안 서버의 정보 패킷이 오지 않았을 때에는 생성하였던 정보 파일을 삭제하도록 한다. 삭제된 서버는 관리도구와 동적 업데이트 모듈에 의해서 DNS 정보에서 자동으로 삭제되도록 한다.

모니터링 에이전트는 각 서버에 실행되며 서버들의 부하정보(CPU, 메모리, 네트워크)와 자신의 IP 주소를 포함한 다음과 같은 UDP 패킷을 모니터링 때문에 보낸다.

CPU Usage	MEM Usage	IP Address	CPU Count	Net Rate
-----------	-----------	------------	-----------	----------

3.4 확장 DNS 웹 관리도구의 설계 및 구현

확장 DNS는 리눅스 환경에서 수행되며, 여러 텍스트 기반의 명령어들을 이용하여 관리하도록 구성되어 있다. 이런 불편함을 덜기 위하여 관리자가 웹을 통하여 확장 DNS를 손쉽게 관리할 수 있도록 관리 도구를 제공할 필요가 있다. 따라서 웹 관리 도구는 확장 DNS를 웹을 통하여 손쉽게 관리할 수 있도록 하는 역할을 하며 구조는 그림 4와 같다.



(그림 4) 웹 관리도구 구조

관리자 웹 페이지에서는 로그인/로그아웃 기능, DNS 정보 보여주기 기능, 서버 상태 보여주기 기능, 로그 보기 기능, 관리자 정보 수정 기능, 업데이트 정책 설정 기능, 서버 추가/삭제 요청 기능, 설정 정보 기능, 키 설정 기능을 관리자가 쉽게 사용할 수 있도록 구성되어 있다.

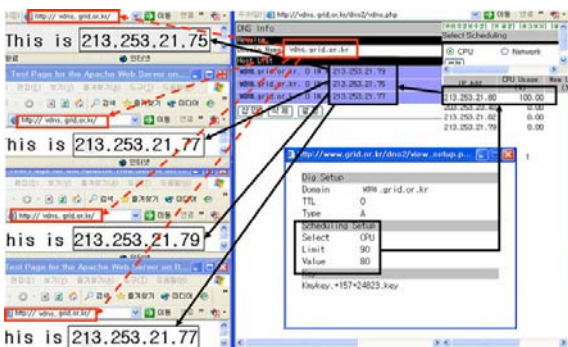
로그인/로그아웃 기능은 확장 DNS 관리를 하기 위한 인증을 받는다. DNS 정보 보여주기 기능은 현재 DNS에 등록되어 있는 서버들의 리스트를 보여주는 기능이다. DNS 정보는 BIND에서 제공하는 'dig' 명령어로 출력되는 부분을 편집하여 화면에 보여주도록 하였다. 서버 상태 보여주기 기능은 현재 리스트에 보이는 서버들의 부하정보를 실시간으로 보여주는 기능이다. 서버들의 상태에 관한 정보는 모니터링 도구에서 생성한 서버 정보 파일을 읽어

들어 화면에 보여주도록 하였다. 로그 보기 기능은 확장 DNS의 추가/삭제에 대한 로그를 볼 수 있는 기능이다. 동적 업데이트 모듈에서 생성되는 로그 파일을 열어 그 내용을 보여주게 된다. 이러한 로그 내용을 바탕으로 업데이트 정책을 정하는데 참조할 수 있다. 업데이트 정책 설정 기능은 확장 DNS의 스케줄링 정책을 설정할 수 있다. 서버 추가/삭제 요청 기능은 새로운 서버의 추가 또는 수정에 관한 폼을 제공하며 동적 업데이트 모듈에 실시간 업데이트를 요청하는 기능이다. 관리자 정보 수정 기능은 관리자의 아이디와 비밀번호를 변경할 수 있는 기능이다. 설정 보기 기능은 현재 확장 DNS의 모든 설정 정보를 보여줌으로써 변경된 설정들이 적용되었는지 확인을 위한 기능을 한다. 키 설정 기능은 DNS 보안을 위한 TSIG key를 설정한다.

4. 실험

본 장에서는 확장 DNS에서 사용할 수 있는 기능들을 수행하고, DNS에 등록된 라운드 robin DNS 정보를 관리자 웹 페이지를 통하여 여러 가지 동적 업데이트 정책에 의해서 실시간으로 동적 변경되는 실험을 수행하여 확장 DNS를 이용한 부하분산 시스템의 유용성을 살펴본다.

실험에 사용된 DNS 서버는 실제 네임서버 ns.grid.or.kr을 서비스하고 있는 시스템이며, DNS 응용프로그램을 위해 BIND 9에 TSIG 키를 설정하였다. 실험에 사용한 도메인은 초기에 4대의 클러스터를 라운드 robin 방식으로 설정된 vdns.grid.or.kr을 이용하였다.



(그림 5) CPU기반 정책 실험

그림 5에서와 같이 관리 도구에서 CPU사용률의 최대 임계값을 90%, 최소 임계값을 80%로 설정하였을 때 서버(213.253.21.80)의 CPU사용률이 90% 초과 이후 서버 리스트에서 삭제되며, 동시에 해당 도메인(vdns.grid.or.kr)으로 접속에 해당되지 않음을 볼

수 있다. CPU사용률이 최소 임계값(80%)보다 작아졌을 경우 삭제된 서버는 서버 리스트에 추가되며 다시 접속이 이루어지는 것을 볼 수 있다.

5. 결론

DNS 기반 웹 서버 클러스터는 구성하기가 쉽고 분산 환경에 적합한 이점이 있지만 각 서버의 부하를 고려하지 않는다는 단점이 있다. 본 논문에서는 기존의 라운드 robin DNS 방식의 단점을 보완하고 G 분산 웹 서버 부하분산을 위하여 확장 DNS를 이용한 부하분산 시스템을 제안한다. 이 시스템은 동적인 부하분산을 하기 위하여 zone 파일에 등록된 서버들의 부하 정보를 실시간으로 감지하여 한 서버로 집중될 수 있는 클라이언트의 요청을 효율적으로 분산한다. 그리고 동적 업데이트 모듈을 사용함으로써 zone 파일을 직접 수정하지 않고 관리 도구에서 DNS 레코드 정보를 업데이트 할 수 있다. 또한 이 시스템은 DNS 프로토콜을 수정하지 않고 모듈을 추가함으로써 기존의 DNS와 완벽한 호환을 할 수 있도록 한다. 웹으로 구성된 관리자 도구는 관리자가 손쉽게 확장 DNS를 관리할 수 있도록 인터페이스를 제공한다.

참고문헌

- [1] R. J. Schemers, "lbnamed: A Load Balancing Name Server in Perl," Proc. 9th systems Administration Conference, Monterey, CA, September 1995.
- [2] T. Kwan, R. McGrath, and A. Reed, "NCSA's World Wide Web Server: Design and Performance," IEEE Computing, Vol. 8, No. 2, pp. 17-26, March-April 2000.
- [3] M. Baentsch, L. Baum, and G. Molter, "Enhancing the Web's infrastructure: From caching to replication," IEEE Concurrency, Vol. 5, No. 1, pp. 56-67, January-March 1997.
- [4] Y.M. Teo, R. Ayani, "Comparison of Load Balancing Strategies on Cluster-based Web Servers," Transactions of the Society for Modeling and Simulation, 2000.
- [5] A. Bestavros, "Distributed Packet Rewriting and its Application to Scalable Web Server Architectures," Proc. 6th IEEE International Conference Network Protocols, October 1998.