

## Ring구조를 이용한 DNS기반 부하분산

김소영<sup>o</sup> 홍영식

동국대학교 컴퓨터공학과

soylove<sup>o</sup>@dongguk.edu, hongys@dongguk.edu

### DNS-based Load Distribution Using Ring Topology

So-Young Kim<sup>o</sup>, Young-Sik Hong

Dept. of Computer Engineering, Dongguk University

#### 요 약

인터넷 사용자의 요청이 점차 증가함에 따라, 여러 웹 서버들이 하나의 클러스터를 이루어 사용자 요청을 더욱 효율적으로 처리하는데 많은 연구가 이루어지고 있다. 이러한 연구들 중에는 클러스터내의 최대 효율을 위하여 웹 서버들간의 부하를 분산시키는 기법들에 대한 다양한 기법들이 연구되고 있다.

이에, 본 논문에서는 DNS를 기반으로 하는 부하 분산 기법을 적용하되, 각 웹 서버들에 의해 수집된 부하정보를 바탕으로 추가적인 부하 분산을 하고자 한다. 이 기법은 실험에 의해 기존의 부하 분산 기법에 비해 웹 서버들간의 부하가 더욱 균등하게 분산됨을 보이고 있다.

#### 1. 서 론

인터넷 사용자 수가 증가함에 따라 웹 서버에 대한 요청이 급격히 증가하고 있다. 이에 웹 서버들이 하나의 클러스터를 이루어 급격히 증가되는 요청을 처리하는 방법을 사용하고 있다[1]. 또한 클러스터를 이루는 웹 서버들간의 부하를 분산시킴으로써, 기존 웹 서버의 활용도를 높일 수 있게 된다.

웹 서버들간의 부하를 분산시키기 위해서는 웹 서버들간의 부하 정보를 수집하고 분석해야 하는 작업이 우선이 되어야 하고, 그 후에 분석 정보를 바탕으로 요청에 대한 부하를 분산시키는 작업이 진행되어야 한다. 대부분의 부하 분산 기법에서는 분석과 분산을 하나의 시스템에서 행해지는데, 이는 시스템에 더 많은 오버헤드가 부가됨으로 시스템의 성능을 더욱 떨어뜨리는 요인이 된다.

이에 본 논문에서는 클러스터를 구성하는 웹 서버들의 부하 정보 분석을 각 서버에서 수행함으로 분석에 대한 오버헤드를 줄이고자 한다.

본 논문은 기존의 부하 분산 기법들의 문제점을 2장에서 알아보고, 3장에서는 본 논문의 제안 기법인 각 웹 서버들의 부하 분석의 수행 과정을 제시한다. 마지막으로 4장에서는 제안 기법이 실험을 통하여 각 웹 서버들의 부하분산에 대한 작업이 균등하게 이루어짐을 보이고 있다.

#### 2. 관련 연구

일반적으로 부하 분산은 클라이언트 측에서 수행하는 방법과 서버 측에서 수행하는 방법으로 나누어 볼 수 있다[1].

클라이언트 측에서 부하 분산을 행하는 경우에는 클라이언트가 직접 웹 서버의 정보를 가져와 요청을 균등하게 분산시킨다. 이 방법의 구현상 문제점은 클라이언트 측에 부하 분산을 위한 추가적인 프로세스가 필요하다는 점이다.

이에 일반적으로 부하 분산은 서버 측에서 담당하고 있다. 서버 측에서 수행되는 부하 분산 기법은 DNS 기반 방식, 디스패처 기반 방식, 서버 기반 방식이 있다.

디스패처 기반 방식은 부하 분산을 담당하는 디스패처가 각 웹 서버에서 처리중인 연결 개수를 고려하여 요청을 할당하는 기법이다. 이 기법은 디스패처가 부하 분석과 분산을 동시에 행하기 때문에 디스패처에 대한 오버헤드가 많다. 또한, 부하 분산을 위해서 각 웹 서버의 연결 개수 정보만을 고려하기 때문에 실제 각 웹 서버들이 가지는 부하 정도는 고려되지 않는다는 단점을 가진다[2].

DNS기반 방식은 DNS가 여러 웹서버의 IP정보를 유지하고 있으며, 클라이언트의 요청이 들어올 때마다 IP를 순차적으로 응답에 실어 보내어 여러 서버에 부하를 할당하는 방법이다[7]. DNS기반 방식은 IP분배 스케줄을 초기에 설정하고, 변경하지 않기 때문에 각 웹서버의 부하에 따라 스케줄을 변경하지 못한다는 단점을 가진다.

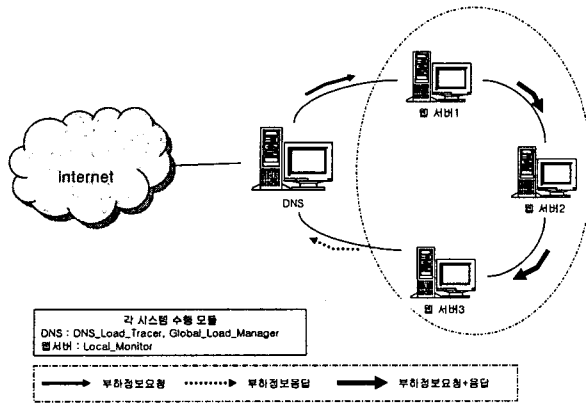
서버 기반 방식은 모든 웹 서버들이 부하 분산에 참여한다[4]. 각 서버가 클라이언트의 요청을 받고, 자신의 현재상태에서 부하가 기준 이상일 경우 다른 서버에게 재전송하는 방식으로, 다른 시스템에서 부하정보의 수집 및 분석을 수행할 필요가 없어 추가적인 부하는 줄여줄 수 있다. 그러나 요청을 재전송하는 과정에서 발생하는 메시지는 네트워크 부하를 가중시키는 단점을 가지고 있다.

#### 3. Ring기반 부하분산 기법

##### 3.1. Ring구조 클러스터

일반적인 부하 분산 기법은 하나의 프로세서 또는 하나의 시스템이 각 웹 서버의 부하 정보를 수집하고 분석한 후, 분석 정보를 바탕으로 분산을 하게 되므로, 해당 프로세서 또는 해당 시스템에 부하가 가중된다는 문제점이 있다[3].

이러한 문제점을 해결하기 위해서 본 논문에서는 부하 분산만을 행하는 시스템을 두고, 부하 분석은 각 웹 서버가 담당하도록 하는 [그림 1]과 같은 링 구조의 클러스터 방식을 제안한다.



[그림 1] Ring구조 클러스터

[그림 1]에서와 같이, DNS에는 DNS\_Load\_Tracer 모듈과 Global\_Load\_Manager 모듈이 있고, 각 웹 서버에는 Local\_Monitor 모듈을 설치한다. 각 모듈 별 기능은 아래의 [표 1]과 같다.

모듈 명	기능
DNS_Load_Tracer	부하 분산을 위한 작업 실행 여부 판단
Global_Load_Manager	클러스터 내의 웹 서버들의 부하 정보 갱신
Local_Monitor	해당 서버의 부하 정도를 이전 서버의 부하 정도와 비교

[표 1] 각 모듈별 기능 설명

### 3.2. 각 모듈 수행 과정

다음[그림 2]는 클러스터내의 시스템간 통신을 위한 메시지 구조이다.

```

struct msg{
    bool reqLoad_Tag; //부하분배 정보 갱신 여부를 표현
                    // 0 : 부하 정보 수집 요청
                    // 1 : zone파일 갱신을 위한 데이터 요구
    float loadInfo; // 부하정보
    int serverNum; //loadInfo를 가지고 있는 서버의 번호
}
    
```

[그림 2] 클러스터내 메시지 구조

각 서버는 메시지를 받을때 reqLoad\_Tag를 우선 확인한다. reqLoad\_Tag가 0일 경우에 이 메시지를 받은 서버는 자신의 부하를 측정하여 저장한 후 다음 서버에게 메시지를 전달한다. reqLoad\_Tag가 1일 경우에는 서버의 부하를 측정하고, 앞 서버와의 부하를 비교하여 부하가 적은 서버의 정보만을 메시지에 실어 전달한다.

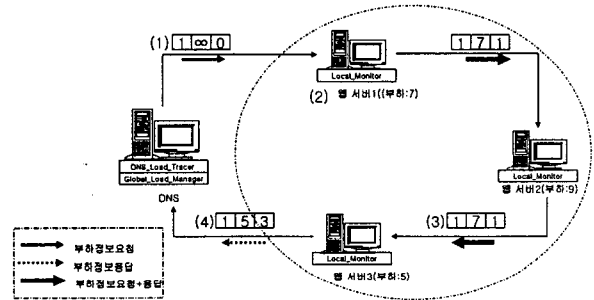
제안 기법에서의 구성 모듈 별 수행과정은 다음과 같다.

DNS\_Load\_Tracer: DNS에서 수행되는 모듈로서 각 웹 서버의

부하 정보를 수집하기 위해서 reqLoad\_Tag를 1로 설정하여 가장 첫 번째 웹 서버의 Local\_Monitor에게 메시지를 송신한다.

Load\_Monitor: 각 웹 서버에서 수행되는 모듈로 이전 서버에서 받은 메시지의 reqLoad\_Tag에 따라 수행된다. reqLoad\_Tag가 0인 경우에는 서버의 부하 정보를 수집하고, reqLoad\_Tag가 1인 경우에는 서버의 부하를 측정한 후, 메시지의 부하정보와 비교하여 부하가 적은 서버의 정보를 메시지에 실어 다음 웹 서버의 Local\_Monitor에게 전달한다

Global\_Load\_Manager: DNS\_Load\_Tracer가 보낸 메시지 가진 모든 웹 서버를 거쳐 다시 DNS로 돌아올 때 수행되며, 보내온 메시지 중에서 reqLoad\_Tag의 값에 따라 다른 수행을 한다. reqLoad\_Tag가 0이면 각 웹 서버가 부하 정보의 수집을 완료했음을 나타내고, reqLoad\_Tag가 1이면 전체 웹 서버들 중에서 부하가 가장 적은 웹 서버의 정보를 받는 경우로 DNS의 zone 파일을 수정하여 부하 분산에 필요한 정보를 갱신한다.

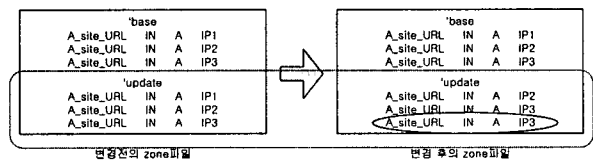


[그림 3] reqLoad\_Tag가 1일 경우 전송 메시지 형태

앞의 [그림 3]은 reqLoad\_Tag가 1인 경우에 각 웹 서버들간의 전달되는 메시지를 보이고 있다.

그림과 같이 DNS\_Load\_Tracer가 reqLoad\_Tag와 loadInfo값을 각각 1과 임의의 큰 수를 설정하여 웹 서버1에게 보내면 웹 서버1은 loadInfo와 자신의 부하 정보를 비교하여 웹 서버1의 부하가 적으면 메시지의 loadInfo와 serverNum를 변경하여 다음 서버로 전달한다. 이러한 과정을 거쳐 웹 서버3은 메시지를 Global\_Load\_Manager에게 보낸다.

### 3.3. zone파일 갱신



[그림 4] zone파일 갱신

[그림 4]는 Global\_Load\_Manager에서 받은 reqLoad\_Tag의 값이 1인 경우에 zone파일의 변경 전과 후를 보인다. 변경 'update'항목에서도 아랫부분으로 갈수록 최신 정보로 부하가 가장 적음을 나타낸다.

이와 같이, zone파일을 변경 후 가장 부하가 적은 웹 서버를 찾아 더 많은 요청을 처리하도록 요청을 분산시킨다.

4. 실험 및 결과 분석

4.1. 실험 환경 및 환경 변수

제안한 방법은 한 시스템에서 부하 정보를 분석하는 것을 클러스터를 구성하는 각 서버가 나누어 수행함으로써 DNS의 부하를 줄일 수 있도록 웹 클러스터의 구조를 변형한 것이다.

[표 2]는 본 논문의 실험을 위한 환경 변수를 나타낸다.

환경 변수	환경 변수 값
사용자 수	1000명
요청 시간	6시간
부하 생성	㈜뱅크타운의 LoadCube 부하 생성
부하 정보 요청 간격	1분
클러스터 파일 갱신 주기	1,3,5,10(분)

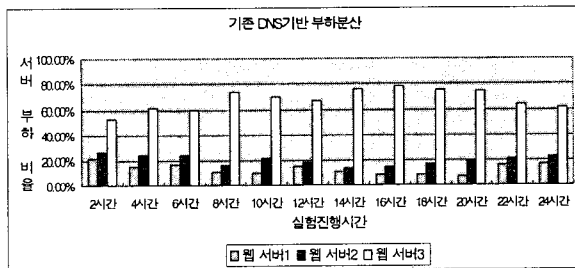
[표 2] 실험환경 변수

[표 2]에서 보는 바와 같이, 부하 생성을 위하여 사용자의 요청은 1000명의 사용자가 6시간동안 임의로 웹 페이지에 대한 요청을 하도록 하였고 이 실험을 위하여 (주)뱅크타운의 LoadCube 부하생성기를 이용하였다[6].

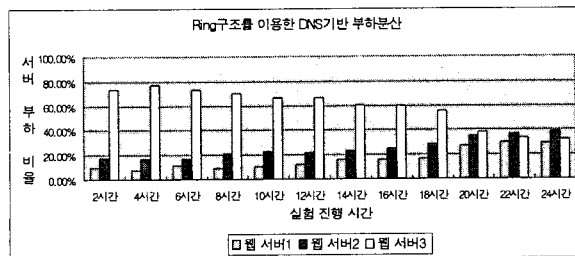
부하의 기준은 리눅스의 /proc/loadavg에서 얻을 수 있는 서버의 평균 시스템 부하를 이용하였다.

4.2. 실험 결과 및 분석

본 실험에 사용된 부하는 부하 생성기에서 임의적으로 생성되었기 때문에 측정된 부하의 총합을 100%라고 가정할 때, 각 웹 서버가 담당한 비율을 그래프에 반영하였다.



[그림 5] 기존 DNS기반의 부하분산 비율



[그림 6] Ring구조 이용시 부하분산 비율

[그림 5]는 기존의 DNS기반 부하분산 방식에서의 부하분산 정도를 측정한 것이다. 계속적으로 웹 서버3이 가장 많은 부하를 담당하고 있는 결과가 나타났다.

제안 방법을 실험 한 결과는 [그림 6]에서 보듯이 시간이 갈수록 부하가 각 웹 서버에 균등하게 분산됨을 확인할 수가 있다. 초기에는 웹 서버3에 부하가 집중됨을 보이나, 시간이 갈수록 점차 균등한 분산 비율을 보인다. 대략 20시간이 경과한 후부터는 각 웹 서버가 1/3가량의 부하를 나누어 수행되는 결과를 볼 수가 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 클러스터의 일반적인 구성 방법을 변형하여 DNS에서 부하 정보를 수집하여 평가하는 데에 추가적으로 발생하는 부하를 줄이면서 부하 분산을 수행하는 시스템을 제안하였다.

향후 연구 과제로는 부하 분석을 위한 메시지 전달형태가 링 구조이기 때문에 중간 서버의 고장에 대한 대처가 미흡하므로 이 점을 보완하기 위한 연구가 추가적으로 이루어져야 한다. 또한 본 논문에서 제안한 구성을 바탕으로 콘텐츠별 부하를 측정하여, 요청되는 콘텐츠에 따른 부하 분산을 수행하는 방법에 대한 연구가 필요하다.

6. 참고 문헌

[1]Valeria. Cardellini, Emiliano. Casalicchio, Michele. Colajanni. "The State of the Art in Locally Distributed Web-server Systems", ACM Computing Surveys Vol. 32, No. 2, pp. 263-311, June 2002

[2]The Linux Virtual Server Project, <http://www.linuxvirtualserver.org>

[3]김성영, 문종배, 김명호, "확장 DNS를 이용한 분산 웹 서버의 동적 부하분산 기술", 제21회 한국정보처리학회 추계학술발표대회 논문집 제11권 제1호, 2004.05

[4]Valeria Cardellini, Michele Colajanni, Philip S. Yu, "Redirection Algorithms for Load Sharing in Distributed Web-server Systems", Proc. IEEE 19th Int'l Conf. on Distributed Computing Systems (ICDCS'99), Austin, TX pp.528-535, June 1999

[5]Hiroshi Yokota, Shigetomo Kimura, Yoshihiko Ebihara, "A Proposal of DNS-Based Adaptive Load Balancing Method for Mirror Server Systems and Its Implementation", Proc. 18th Int'l Conf on Advanced Information Networking and Application(AINA'04), 2004. IEEE

[6]u-Biz Partner BANKTOWN, <http://www.banktown.com>

[7]Thomas T. Kwan, Robert E. McGrath, Daniel A.Reed. "NCSA's World Wide Web Server:Design and Performance", IEEE Computing, Volume 28, Issue 11, Nov 1995. Page(s):68-74