

사용자 요청을 고려한 Hierarchical LVS 시스템의 설계 및 성능분석

오경호, 조진성, 정병수

경희대학교 전자정보학부 컴퓨터공학과

okh76@dblab.khu.ac.kr, {chosjs, jeong}@khu.ac.kr

Design and Performance Evaluation of Content-aware Hierarchical LVS

Kyung-ho Oh, Jinsung Cho, Byeong-Soo Jeong
Dept. of Computer Engineering, Kyung Hee University

요 약

클러스터링을 이용하여 웹 서버 구축 시 가장 중요하게 고려해야 할 점은 전면에서 부하 분산을 수행하는 부하 분산기에 걸리는 부하를 최소화 하는 것과 실제 사용자의 요청을 처리할 노드들 사이에 적절한 부하 분산이 이루어지도록 하는 것이다. 부하 분산 알고리즘은 서버의 성능에 많은 영향을 미치며 부하 분산에 이용되는 기준은 빨리 계산될 수 있어야 한다. 본 논문에서는 사용자 요청에 따라 노드들을 그룹핑하여 그룹별로 전용 부하 분산기를 가지고 사용자 요청의 종류에 따라 최적의 스케줄링 알고리즘을 적용할 수 있는 Hierarchical LVS(Linux Virtual Server) 시스템을 설계하고 단일 부하 분산기만을 사용하는 기존의 LVS와의 차이점을 실험을 통하여 제시하고자 한다.

1. 서 론

갈수록 비대해져 가는 인터넷 환경에서 클러스터링 기법이 고성능과 고가용도를 요구하는 응용 분야에서 하나의 대안으로 등장하고 있다. 클러스터형의 분산 서버 시스템은 확장성이 뛰어나고 장애에 대한 안정성도 높으면서 저가라는 점에서 인터넷 비즈니스 서비스용 서버로 적당하지만, 공유 메모리가 없는 독립적인 이기종의 분산 서버들로 구성되어 있어서 분산 어플리케이션의 개발이 힘들고 분산된 자원의 효과적인 운영이 시스템의 성능에 큰 영향을 미친다.

클러스터형 웹 클러스터에서 고려해야 할 중요한 이슈는 알맞은 라우팅 알고리즘을 선택하는 것과 효율적인 성능을 보이는 구조를 만드는 것이다[1].

Content-aware routing mechanism은 웹 서버 클러스터에 들어오는 패킷들을 요청되는 콘텐츠 타입 정보를 이용하여 분류한 뒤 실제 요청을 처리하는 서버로 보내는 방식이다. 이것은 웹 환경의 특성을 잘 살릴 수 있는 방식으로, 인터넷의 사용이 증가하면서 더욱 중요해지게 되고 많은 웹 서버 시스템에서 사용하게 되었다[2, 8].

웹 클러스터의 구조는 크게 Cluster-based

Web system, Virtual Web 클러스터 그리고 Distributed Web system으로 나뉜다[2].

Cluster-based Web system은 하나의 부하 분산기와 하나 이상의 서버노드들로 이루어지며, 분산기는 클라이언트의 요청이 웹 서비스를 호스팅 해주는 클러스터 시스템으로 들어오면 실제 이 요청을 처리할 적절한 노드로 분산시켜 주는 역할을 한다.

이러한 역할을 하는 대표적인 소프트웨어로는 LVS(Linux Virtual Server)가 있고 하드웨어로는 MagicRouter, LocalDirector 그리고 TCP Router 등이 있다[3, 4]. Centralized IP 클러스터는 다시 부하 분산기가 OSI 7 계층 중 어느 계층에서 동작하느냐에 따라 Layer-4 분산기와 Layer-7 분산기로 나누어진다[2]. Virtual Web 클러스터와 Distributed Web system은 부하 분산기를 따로 두지 않는 방법인데, 이렇게 함으로서 Load Balancer가 병목지점이 될 수 있는 문제점을 원천적으로 봉쇄할 수가 있다.

본 논문에서는 LVS를 사용하여 구성된 Cluster-based Web system 구조를 기반으로 하며 Layer-4 분산기를 사용한다.

현재 대부분의 웹 사이트는 여러 가지 콘텐츠 타입을 복합적으로 다루고 있다. 따라서 사용자가 어떠한 콘텐츠를 요청하느냐에 따라 서버에 걸리는 부하는 많은 차이가 있다. 이러한 환경에서 사용자의 요청을 처리할 노드를 선택할 때 주로 노드에 걸리는 부하를 주기적으로 수집하여 적절한 노드를 선택하는 동적인 방법을 사용하고 있다. 여기서 가장 문제가 되는 것이 부하 정보를 수집할 때 너무 많은 비용이 들고 부하 분산기의 부하가 커진다는 것이다. 이러한 문제점을 해결하기 위해서 본 논문에서는 서버의 부하정보 보다는 사용자의 요청 정보에 따라 분산을 수행하되 콘텐츠 타입에 따라 노드들을 그룹핑하고 그룹별로 적절한 스케줄링 알고리즘을 적용하는 Hierarchical LVS 시스템 구조를 제시하고자 한다.

2. 관련연구

Content-aware routing mechanism은 사용자의 요청 정보에 기반 하여 해당 요청을 처리할 노드를 선택하는 기술이다. 사용자의 요청 정보를 어느 정도까지 알아내느냐에 따라 Layer-4 분산기와 Layer-7 분산기로 구분한다. Layer-7 분산기가 더 많은 정보를 알아낼 수 있기 때문에 훨씬 정교한 부하 분산이 가능하지만 상대적으로 부하 분산기에 과중한 부하가 발생하게 되므로 일반적으로 Layer-4 분산기를 사용한다.

2.1 Distributed system

2.1.1 Cluster-based Web system (Web Cluster)

Web cluster는 동일한 지역에서 고속의 네트워크로 연결되어 있고 외부에는 단일 시스템 이미지로 보여진다. 각 서버노드는 일반적으로 자신만의 저장장치와 OS를 가지고 단일 시스템 자원으로 집합적으로 동작한다. 전면에 위치한 부하 분산기는 단일 Virtual IP(VIP)를 가지고 외부와는 VIP만을 통하여 교신하고 이것을 통해서 서버노드들을 외부에 연결하게 된다. 부하 분산기와 N개의 서버노드들로 이루어진 기본 Web Cluster의 구조는 그림 1과 같다.

그림 1에서 서버노드가 사용자의 요청 결과를 건네주는 경로는 나타나지 않았는데, 왜냐하면 요청의 결과는 부하 분산기를 다시 통과하여 사용자에게 건네지는 방식과 서버노드에서 직접 사용자에게 건네지는 방식이 있기 때문이다. 본 논문에서는 직접 사용자에게 건네지는 방식을 사용한다.

2.1.2 Virtual Web 클러스터

Virtual Web 클러스터는 전면에 부하 분산기를 사용하지 않고 외부에 단일 VIP 주소로 외부에 보여지는 구조이다. Web Cluster와의 주된 차이점은 VIP 주

소로 들어오는 사용자의 요청을 분산시킬 때 부하 분산기를 사용하지 않는다는 점이다. VIP 주소는 클러스터내 모든 서버노드들에 의해 공유되고, 따라서 각

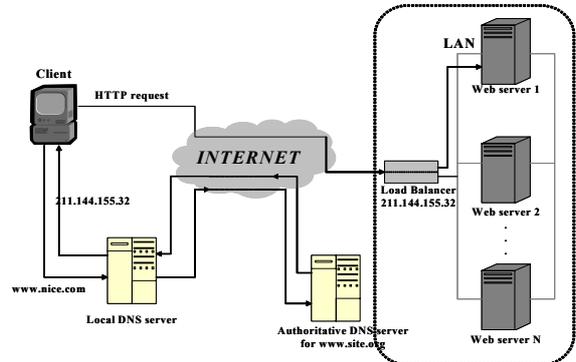


그림 1 Cluster-based Web system의 구조
서버노드들은 모든 들어오는 패킷들을 받고 그것들을 처리할지 거부할지를 적절한 해쉬 함수를 통하여 판단하게 된다. Virtual Web Cluster의 구조는 그림 2와 같다.

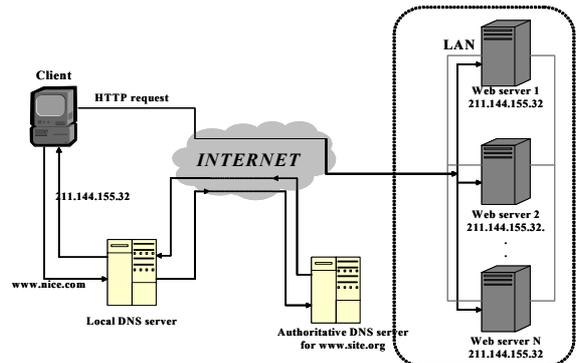


그림 2 Virtual Web 클러스터의 구조

2.1.3 Distributed Web system

Distributed Web system은 단일의 웹 사이트 주소에 다중의 IP 주소를 가지고 외부에 노출된다. Virtual Web 클러스터와 마찬가지로 부하 분산기에 의존하지 않는 구조를 가지며, 사용자의 요청을 분산하기 위해서 전적으로 DNS 메카니즘에 의해 웹 사이트명의 주소 해석 동안에 수행된다. Distributed Web system의 구조는 그림 3과 같다.

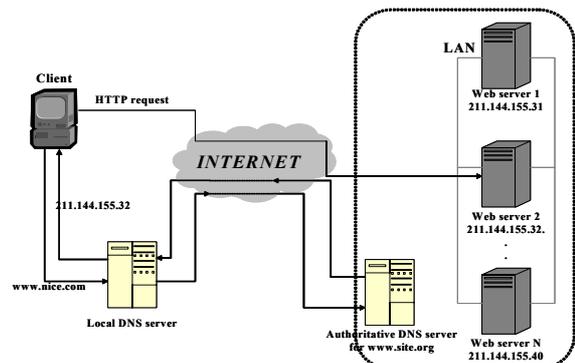


그림 3 Distributed Web system의 구조

2.2 LVS (Linux Virtual Server)

LVS는 Web Cluster 형태로 가장 대표적인 부하 분산 소프트웨어이다. 부하 분산을 수행하는 방식에 따라 NAT, IP Tunneling 그리고 Direct Routing 방식으로 나누어진다[3].

2.2.1 NAT

특정한 IP 주소를 한 그룹에서 다른 그룹으로 매핑하는 기술로 N-to-N의 정적 NAT, M-to-N(M>N)의 동적 NAT가 있는데 LVS 구성시에는 M-to-1의 방식이 사용된다. 서버노드가 사용자 요청에 응답할 때 부하 분산기를 다시 거쳐야 하므로 상대적으로 부하 분산기에 많은 부하가 걸리게 된다.

2.2.2 IP Tunneling

IP 데이터그램안에 IP 데이터그램을 넣는 기술로 특정 IP 주소로 향하는 데이터그램을 감싸 다른 IP 주소로 재지향할 수 있다. NAT와는 달리 단일 지역에 서버 노드들이 위치할 필요가 없고 서버노드에서 직접 사용자에게 응답이 가므로 부하 분산기의 부담은 줄어들지만 데이터그램을 감싸고 푸는 과정에서 분산기와 서버노드 모두에 부하가 발생하게 된다.

2.2.3 Direct Routing

NAT방식처럼 분산기와 서버노드들은 동일 지역에 위치해야 하지만 요청에 대한 응답은 서버노드에서 사용자에게 직접 전달된다. MAC 주소를 이용하여 서버를 선택하는 방식으로 현재 가장 널리 쓰이고 있다.

2.2.4 스케줄링 알고리즘

부하 분산기가 사용자 요청을 직접 처리할 서버노드를 선택하기 위한 알고리즘으로 LVS에서 사용 가능한 알고리즘은 표 1과 같다[3].

표 1 스케줄링 알고리즘

	동 작 방 식
RR	순서대로 한번씩 선택하여 할당
WRR	가중치를 고려하여 할당
LC	활성 접속수를 고려하여 할당
WLC	활성 접속수와 가중치 모두를 고려하여 할당

*RAN:Random, RR:Round Robin, WRR:Weight RR

LC:Least Connection, WLC:Weight LC

3. Hierarchical LVS 시스템의 설계

본 논문에서 제안하는 구조는 Web Cluster의 형태를 띤 기존 LVS에서 1단 으로 수행하던 부하 분산기를 N단으로 확장한 구조이다. 사용자가 요청하는 콘텐츠타입에 따라 부하 분산기를 K개로 확장하고

각각이 M개의 서버노드들을 관리하게 된다. 분산 단계는 N단으로 확장하되 분산기는 해당 그룹의 서버노드 역할도 수행하게 하여 활용도를 높인다. 그렇게 함으로서 부하 분산기의 부하를 최소화 할 수 있고 특정 콘텐츠타입은 특정 그룹 내에서만 서비스가 되므로 서버노드들의 주 메모리에서 캐쉬 적중률을 향상 시킬 수 있다. 또한 그룹별로 스케줄링 알고리즘을 적용할 수 있으므로 콘텐츠의 형태에 따라 최적의 성능을 보이는 알고리즘을 선택할 수 있다. 그리고 하나의 부하 분산기가 관리할 수 있는 서버노드의 개수에는 한계가 있는데 분산기를 계층형으로 설계함으로써 단일 시스템이 수용 가능한 서버노드의 개수도 무한대에 가깝게 확장할 수 있다. 계층이 늘어날수록 사용자 요청이 서버노드까지 가는데 거쳐야 할 경로가 많아지지만 일단 클러스터 시스템 내부에 도착하게 되면 경로 간 이동 소요 시간은 무시할 정도의 수준이다. 하지만 서버노드가 결과를 사용자에게 건네줄 때 왔던 경로를 되돌아가야 한다면 거쳐야 하는 경로도 두배로 늘어나게 되므로 결과를 사용자에게 직접 전달하는 Direct Routing 방식을 사용한다. Hierarchical LVS 시스템의 구조는 그림 4와 같다.

그림 4 Hierarchical LVS 시스템의 구조

4. 실험

4.1 실험 환경

부하 분산기와 서버노드 모두 커널에 LVS 패치를 하고 서버노드 LAN 카드의 MAC(Media Access Control) 주소가 캐쉬되는 것을 막기 위해 hidden 패치를 적용하였다. hidden 패치를 하지 않을 경우 캐쉬 되어 있는 MAC 주소를 통해 분산기를 통과하지 않고 직접 사용자 요청이 서버 노드로 가는 경우가 발생하여 적절한 부하 분산이 이루어지지 않는다.

표 2 클러스터 환경

Real Server	CPU (PII-333) Memory (128MB) OS (Linux Red hat 6.2) Kernel (2.2.14)
Client	Dual CPU (Zeon-2G *2) Memory (1G) OS (windows 2000 server)
Load Balancer	CPU (PII-333) Memory (128MB) OS (Linux Red hat 6.2) Kernel (2.2.14)
Network bandwidth	100MB
Test tool	Web server Stress

실험은 모든 노드가 mp3 스트리밍 서비스와 일반 web page를 모두 서비스하는 기존 LVS 시스템과 콘텐츠 형태에 따라 일반 웹 서비스를 하는 그룹과 mp3 스트리밍 서비스를 하는 그룹으로 나누어진 H-LVS 시스템으로 실험을 하였다.

4.2 실험 결과

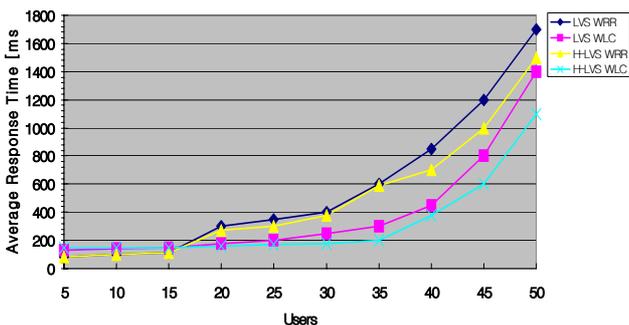


그림 5 평균 응답 시간

그림 5의 응답 시간은 web page에 대한 평균 응답 시간으로 기본적으로 LVS와 H-LVS(Hierarchical LVS) 모두 WLC방식이 WRR 방식 보다는 좋은 성능을 보이고 있다. 동일한 스케줄링 알고리즘을 사용했을 때에는 기존 LVS 구조와 비교하여 H-LVS 구조가 완만한 경사를 가지며 좋은 성능을 보여주었는데 이는 콘텐츠에 기반한 부하분산으로 서버 노드에서의 캐쉬 적중률이 높아졌기 때문이다.

5. 결론 및 향후계획

클러스터 시스템은 갈수록 높은 서비스 품질을 요구하는 현 인터넷 환경의 여러 문제점을 해결할 수 있다. 클러스터 시스템에서 가장 중요한 문제인 부하 분산을 효과적으로 하기 위해 본 논문에서는 기존의 부하 분산기를 N단으로 확장하는 Hierarchical LVS 구조를 제안하였다.

실험결과 기존의 LVS 보다 좋은 성능을 보였으며 스케줄링 방식으로는 WLC 방식을 사용했을 때 가장 좋은 성능을 보여주었다.

본 논문에서는 부하 분산기를 2단계로만 했지만 최적의 성능을 보이는 단계를 실험을 통해 찾아보는 것도 향후 연구의 가치가 있다고 생각한다.

참고 문헌

[1] V. Cardellini, M. Colajanni and P.S. Yu, "Dynamic Load Balancing on Web-server Systems" IEEE Internet Computing, pp. 28-39, May 1999

[2] VALERIA CARDELLINI AND EMILIANO CASALICCHIO, MICHELE COLAJANNI, AND PHILIP S. YU "The State of the Art in Locally Distributed Web-Server Systems" ACM Computing Surveys, Vol. 34, No. 2, June 2002. pp. 263-311.

[3] Wensong Zhang, Shiyao jin and Qanyuan Wu "Creating Linux Virtual Servers" LinuxExpo 1999 Conference

[4] Cisco System, Cisco Local Director (1998), <http://www.cisco.com/warp/public/cc/pd/cxsr/400/index.shtml>

[5] Eric Anderson, Dave Patterson, and Eric Brewer, "The Magicrouter: an application of Fast packet Interposing May. 1996 <http://www.cs.berkeley.edu>

[6] Vivek S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, E. Nahum, "Locality-aware request distribution in cluster-based network servers", In Proc. of 8th ACM Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, Oct. 1998.