

중소형 사이트를 위한 고가용성 웹 서버 클러스터

문종배^U 김명호
승실대학교 컴퓨터학과

comdoct@ss.soongsil.ac.kr, kmh@computing.soongsil.ac.kr

High Availability Web Server Cluster for Middle Web Sites

Jong-Bae Moon^U Myung-Ho Kim
School of Computing, Soong-Sil University

요 약

최근에 리눅스 가상 서버 프로젝트를 이용한 클러스터를 많이 구축하고 있다. 리눅스 가상 서버는 확장성이 뛰어나지만 성능도 뛰어나다. 그러나 하나의 전면 서버가 사용자 요청을 받아서 후면 서버가 처리를 하여 사용자에게 응답을 보내는 구조이므로 하나의 전면 서버는 사용자 요청의 급격한 증가로 병목현상을 일으킬 수 있고, 이 때문에 다수의 후면 서버가 있음에도 불구하고 서비스를 할 수 없는 상황이 발생하게 된다. 따라서 본 논문에서는 다수의 전면 서버를 두어 병목현상을 근본적으로 없애는 방법을 제안하였다. 모든 클러스터 노드들은 전면 서버와 후면 서버의 역할을 모두 할 수 있도록 하였고 부하량이 많지 않으면 자기 노드에서 직접 처리를 하여 사용자에게 응답할 수 있도록 하고, 부하량이 많을 때에만 클러스터의 다른 노드들 중 부하가 가장 작은 노드를 선택하여 사용자 요청을 처리할 수 있도록 하였다. 그리고 클러스터 노드들의 부하량을 고려한 스케줄링 알고리즘을 개발하여 좀 더 효율적인 부하 분배방식을 제안하였다.

1. 서론

최근 인터넷 사용이 일반화되면서 인터넷 사용자의 수가 매년 증가하고, 인터넷은 새로운 산업의 형태로 발전하고 있다. 전자 상거래는 그 좋은 예가 될 수 있다. 현재 전자 상거래와 같은 서비스를 제공하는 사이트가 점차 늘어나고, 이 가운데 하루 평균 백만 건 이상의 접속이 이루어지고 있는 사이트도 생겨나고 있다. 이러한 사이트에서는 사용자의 요청을 처리하기 위해 많은 컴퓨팅 파워를 요구하고 있다[6]. 또한 이러한 사이트는 시스템이 다운되었을 경우 엄청난 경제적 손실을 입게 된다. 따라서 시스템의 다운 타임 최소화도 요구하고 있다. 이러한 측면에서 고성능과 고가용성을 제공하는 웹서비스의 필요성이 대두되었다.

단일 시스템으로 구성된 기존의 사이트들은 단일 시스템의 한계 때문에 고성능과 고가용성의 서비스를 제공하는 것이 이미 한계에 도달하였다[2,4]. 이러한 단일 시스템의 한계를 극복하고 더욱 강력한 컴퓨팅 파워와 시스템의 안정적 서비스를 제공하기 위해 클러스터링 기술이 웹사이트에 적용되고 있다[5].

최근 고성능과 고가용성 웹 서버 클러스터 시스템을 구축하기 위해 리눅스 가상 서버를 사용하는 사례가 늘고 있다[1]. 이러한 가상 웹 서버 클러스터 시스템의 일반적인 형태는 사용자의 요청을 받아 부하 분산을 하는 하나의 전면 서버와 사용자의 요청을 실제로 처리하는 다수의 후면 서버로 구성되는 것이 일반적이다. 그러나 이런 형태는 사용자의 요청을 후면 서버가 처리하여 직접 사용자에게 보낸다고 하여도 사용자의 요청이 급격히 증가하여 전면 서버에 병목현상을 불러

일으키는 상황이 발생할 수 있다. 또한 이 전면 서버가 작동을 하지 않는다면 다수의 후면 서버가 있음에도 불구하고 서비스가 전혀 이루어 질 수 없다.

본 논문에서는 고성능 가상 서버를 기반으로 하여 부하분산을 하는 전면 서버와 사용자의 요청을 처리하는 후면 서버의 역할을 동시에 할 수 있는 고가용성 웹 서버 클러스터 시스템을 설계하고 구현한다.

2. 가상 서버를 이용한 웹 서버 클러스터

웹 서버 클러스터는 여러 대의 서버 노드로 클러스터를 구성하여 하나의 웹사이트 서비스를 제공할 수 있도록 구성된 시스템이다. 최근 웹 서버 클러스터의 고성능과 확장성, 안정성을 확보하기 위하여 리눅스 가상 서버 프로젝트를 이용한 구조가 많이 사용되고 있다. 클러스터를 구성하는 서버들은 사용자의 요청을 받아들여 분배하는 전면 서버 부분과 사용자 요청을 실질적으로 처리하는 후면 서버 부분으로 구분되는 것이 일반적이다. 이러한 가상 서버를 기반으로 한 웹 서버 클러스터 시스템의 전체 구성 모습은 다음 그림 1과 같다.

2.1. 기존 가상 서버를 이용한 웹 서버 클러스터 시스템의 문제점

기존의 가상 서버를 이용한 일반적인 웹 서버 클러스터 시스템은 하나의 전면 서버와 다수의 후면 서버로 구성되어 있거나, 이러한 형태의 확장이기 때문에 전면 서버에 지나치게 많은 요청이 발생하여 전면 서버의 처리능력을 넘어서서 병목현상이 발생 할 수 있다.

또, 여러 서버들이 처리하는 양은 불균형적이고, 서버들의 시스템 부하

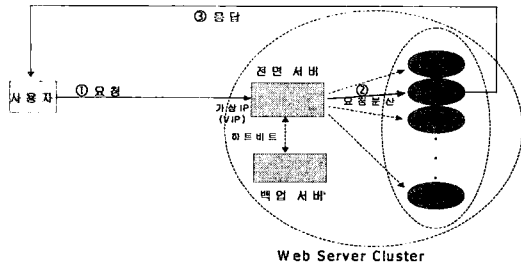


그림 1 가상 서버를 이용한 웹 서버 클러스터의 구조

량은 서버들마다 모두 다르기 때문에 부하량이 많은 서버로 사용자의 요청이 계속 할당되면 서버가 처리하는 속도가 느려지게 되어 사용자에게 만족할 만한 응답시간을 보장하지 못하게 된다. 리눅스 가상 서버의 경우 기본적으로 제공하는 스케줄링 알고리즘들은 서버들의 부하량을 고려하지 않는 방법이므로 부하의 불균형을 초래할 수 있다.

3. 새로운 고가용성 웹 서버 클러스터 시스템

3.1. 고가용성 웹 서버 클러스터 시스템의 전체 구조

본 논문에서 구현한 고가용성 웹 서버 클러스터 시스템은 그림 2에서 보는 것처럼 전면 서버와 후면 서버의 역할을 동시에 하는 다수의 노드들로 구성되어 있고, 각 노드들은 하트비트 데몬, 로드 모니터, 부하량에 따른 부하분배 방식의 스케줄링 모듈 등의 컴포넌트들로 구성되어 있다.

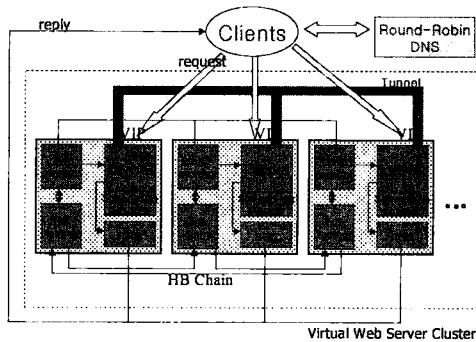


그림 2 전면 서버와 후면서버의 구분을 없앤 고가용성 웹 서버 클러스터 시스템의 구성도

여러 전면 서버들의 가상 IP들을 Round-Robin DNS에 하나의 이름으로 등록을 함으로써 사용자에게 고가용성 웹 서버 클러스터 시스템에 대한 투명성을 제공하도록 한다. 사용자의 요청은 일차적으로 Round-Robin DNS에 의해서 임의의 서버에 전달된다. 사용자 요청을 받은 서버는 자기 노드의 부하량의 변화를 측정하여 임계값 보다 작으면 자기 노드에서 사용자 요청을 처리하여 응답을 직접 사용자에게 보내도록 한다. 그 값이 임계값 보다 크면 로드 모니터와 스케줄링 모듈에 의해서 부하량이 가장 작은 노드를 선택하여 IP 터널을 통해 사용자 요청을 재전송한다.

3.1.1 하트비트(Heartbeat) 데몬

하트비트 데몬은 웹 서버 클러스터 시스템의 고가용성을 위한 컴포넌트이다. 하트비트 데몬은 웹 서버 클러스터 시스템에서 중요한 역할을 담당한다. 하트비트 데몬은 클러스터 시스템의 안정성과 고가용성을 위해 하트비트 체인을 구성하고 유지한다. 클러스터 시스템의 임의의 노드가 다운된 경우 하트 비트 체인에 의해서 결합 노드를 발견하게 된다. 결합 노드의 작업은 fake를 사용하여 나머지 노드중 임의의 노드에 의해서 수행하게 된다

3.1.2 로드 모니터(Load Monitor)

로드 모니터는 주기적으로 자기 자신의 부하 정보를 측정하고 정규화 한다. 이 때 정규화를 하는 기준은 각 부하정보에 가중치를 달리 함으로써 다른 값을 가질 수 있다. 이 정규화된 값을 커널에 보내어 유지하고, 부하의 변화가 있으면 다른 노드와 자신 노드에게 부하 정보를 알린다. 부하의 변화가 크지 않을 경우에도 부하 정보를 교환하는 것은 오히려 로드 모니터간의 네트워크 오버헤드를 늘릴 수 있다. 따라서 로드 모니터는 임계값을 설정해서 부하의 변화량이 임계값을 초과하는 경우에만 자신의 부하 정보를 다른 노드들에 보내도록 하도록 하였다.

3.1.3 서버들의 성능과 부하를 고려한 스케줄링 모듈

각 노드의 스케줄링 모듈은 사용자의 요청이 도착하면 로드 모니터에 의해 정규화 된 값을 기준으로 부하가 가장 작은 노드를 선택한다. 그리고 선택된 노드의 IP를 사용자 요청 패킷에 덧붙여 IP 터널을 통해 실행 서버로 전송하게 된다.

4. 실험 결과

웹 서버의 성능을 평가하기 위해서 SURGE라는 공개소스 프로그램은 사용하였다[3]. SURGE는 실제 사용자의 행동을 대신하는 쓰레드를 발생해서 웹 서버에 접속해 주어진 테스트 시간동안 사용자 요청을 발생한다. 본 논문에서는 SURGE를 일정시간 동안 수행한 후 결과로 얻은 로그파일에서 서버들의 총 처리량을 구하여 클러스터의 성능을 평가하였다.

4.1. 전면 서버의 증가에 따른 성능 평가

기존의 가상 서버를 이용하여 구성한 하나의 전면 서버와 다수의 후면 서버가 있는 형태의 클러스터링은 다수의 후면 서버가 일을 처리하기 때문에 후면 서버의 확장으로 성능을 높일 수 있지만, 하나의 전면 서버가 처리량의 한계에 도달하면 병목현상으로 인하여 서비스의 성능 향상이 없어지므로 어느 정도 이상의 서버 확장은 비용 부담만 더해질 것이다. 이를 확인하기 위해 하나의 전면 서버와 다수의 후면 서버가 있는 클러스터 시스템과 본 논문에서 설계 및 구현한 가상 서버를 이용한 고가용성 웹 서버 클러스터 시스템을 비교하는 실험을 하였다.

실험 결과를 나타낸 그림 3에서 보는 것처럼 기존의 클러스터 시스템

템에서는 후면 서버의 증가에 따라 서버들의 총 처리량이 증가하지만 어느 정도 이후에는 전면 서버의 처리 능력이 포화가 되어서 성능의 향상이 줄어들고 있음을 볼 수 있다. 그러나 새로운 클러스터 시스템에서는 전면 서버와 후면 서버의 수가 증가함에 따라 처리 건수가 선형적으로 증가하는 것을 볼 수 있다. 이로써 전면 서버와 후면 서버의 구분을 없애 뛰어난 확장성을 보장하고 병목현상을 없앴다고 볼 수 있다.

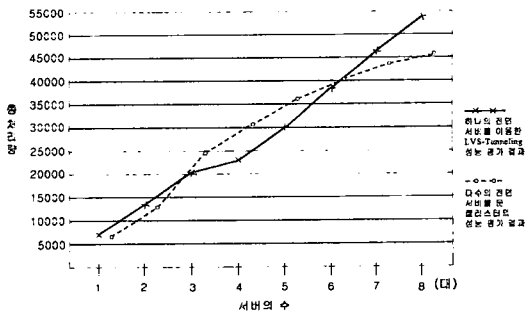


그림 3 기존 클러스터와 시로운 클러스터의 성능평가 결과

4.2. 부하량에 따른 부하분배 방법의 평가

본 논문에서는 부하량에 따른 부하분배 방법의 성능을 측정하기 위해 서버들의 부하량이 1:1:1로 똑같을 때와 1:6:1로 하나의 노드에 부하가 많이 걸려있을 때의 실험을 하였다.

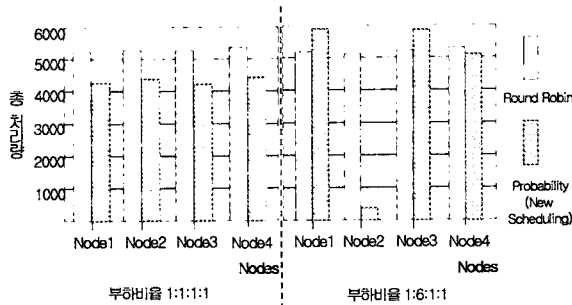


그림 4 부하량에 따른 부하분배 방식의 성능 평가

그림 4에서 보듯이 부하량이 1:1:1일 때에는 두 방법 모두 비슷한 결과를 나타내었다. 그러나 부하량이 1:6:1로 특정 노드에 많은 경우에서 Round-Robin 방식의 부하분배 방법은 부하량에 상관없이 비슷한 요청을 할당하고 있는 것을 볼 수 있고, 부하량에 따른 부하분배 방식은 부하량이 많은 노드에는 상대적으로 적은 양의 사용자의 요청을 할당하고 있는 것을 볼 수 있다. 부하량에 따른 부하분배 방법은 임계값을 정해서 부하량이 그 값보다 작으면 자기 노드가 처리를 하기 때문에 Round-Robin 방법과 비슷하게 동작하는 것처럼 보이지만 서버의 부하량이 임계값을 넘을 때에는 부하량이 가장 작은 서버를 선택하여 IP 터널을 통해 그 요청을 보낸다. 이렇게 부하량이

많은 노드에 적은 사용자 요청을 할당함으로써 서버 부하에 따른 처리시간의 지연을 막을 수 있다. 그러나 Round-Robin 부하분배 방법은 노드에 부하량이 많을 지라도 사용자의 요청이 할당될 수 있다. 이렇게 되면 그 노드의 처리 속도가 늦어지게 하는 원인이 된다.

5. 결론 및 향후 과제

본 논문에서는 가상 서버를 이용하여 전면 서버와 후면 서버의 역할을 동시에 할 수 있도록 고가용성 웹 서버 클러스터를 설치 및 구현하였다. 하나의 노드는 자신의 노드와 클러스터내 다른 노드의 부하를 계산 유지하는 로드 모니터 그리고 클러스터내 각 노드들의 상태 정보를 유지하고 결함 노드에 대한 결함 허용을 제공하는 하트비트 데몬, 서버의 성능과 부하량을 고려한 스케줄링 모듈 등으로 구성되어 있다. 그리고 리눅스 가상 서버를 이용하여 전면 서버와 후면 서버의 구분을 없애 다수의 전면 서버를 갖는 고가용성 클러스터의 손쉬운 확장성을 보장하면서 전면 서버와 후면 서버의 고가용성을 동시에 보장한다. 또 본 논문에서는 서버들의 성능과 부하량을 고려한 동적인 부하 분배방법을 개발하여 부하가 많은 서버에 사용자 요청을 적게 할당하도록 하였다. 이렇게 함으로써 각 서버들 간의 부하 불균형 문제를 좀 더 효율적으로 해결하였다.

웹 서버 클러스터는 고가용성을 위한 하트비트 기능에서 장애 노드의 판단과 Fake 할 때의 시간적 지연 동안에 사용자 요청 패킷들이 손실될 수 있다. 그래서 향후에는 장애 노드와 서비스 가능 서버들에 대한 판단 지연과 Fake에 따른 사용자 요청 패킷들의 손실을 줄이는 연구가 좀 더 필요하다.

6. 참고 문헌

- [1] W. Zhang, S. Jin, and Q. Wu, "Creating Linux Virtual Server", *The 5th Annual Linux Expo Conference* (May, 1999).
- [2] Huican Zhu, Ben Smith, Tao Yang, "A Scheduling sFramework for Web Server Clusters with Intensive Dynamic Content Processing", *Department of Computer Science University of California Santa Barbara, CA 93106*.
- [3] Paul Barford and Mark Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation", *Computer Science Department, Boston University*.
- [4] RADWARE Web Server Director Pro, <http://www.radware.com/support/paper/What to Look For in an IP Load Balancer>.
- [5] T. Brisco, DNS Support for Load Balancing, RFC 1794. <http://www.internic.net/ds/>
- [6] Raykumar Buyya, "High Performance Cluster Computing vol.1." Chap 36. *A Scalable and Highly Available Clustered Web Server*, Prentice Hall (1999).