

컨텐츠를 기반으로한 웹 서버 클러스터 기법

명원식⁰ 장태무
동국대학교 컴퓨터공학과
(wsmyung⁰, jtm)⁰@dongguk.edu

Content-Based Web Server Cluster Scheme

Won Shig Myung⁰, Tae Mu Chang
Dept. of Computer Engineering, Dongguk University

요 약

본 논문에서는 서버의 운영체제에 의존적이지 않은 디스패처(Dispatcher)방식과 클라이언트의 요청에 대한 서버의 응답을 클라이언트와 직접 통신하는 직접 라우팅(Direct Routing) 방식을 혼합하고, 클러스터로 구성된 웹 서버들은 각각 서로 다른 컨텐츠를 가지고 클라이언트 요청에 해당 서버에서 응답하는 컨텐츠를 기반으로한 웹 서버 클러스터 기법을 제안한다. 기존의 디스패처는 해당 디스패처에 관련된 서버들 간의 부하 분산을 효과적으로 처리 하는데 목적이지만, 제안한 컨텐츠를 기반으로한 웹 서버 클러스터 기법은 부하 분산 보다는 디스패처에 대한 오버헤드(Overhead)를 감소시키고 클라이언트의 서비스에 대한 응답시간(Response-time)을 최소화 하는데 그 목적을 둔다. 또한 본 논문에서는 서로 다른 컨텐츠로 구성되어 있는 서버들 중에 특정한 한 서버가 페일(Fail)이 발생했을 때 전체 컨텐츠로 구성되어 있는 또 하나의 서버를 가지고 페일 문제를 해결한다. 본 논문에서 제안한 컨텐츠를 기반으로한 웹 서버 클러스터 기법을 월드 와이드 웹(World Wide Web)에 적용시 폭발적으로 증가하는 서비스 요청과 이로 인한 디스패처의 오버헤드를 효과적으로 처리하여 보다 빠르고 신뢰적인 서비스가 가능할 것으로 기대된다.

1. 서론

월드 와이드 웹(World Wide Web)상에서 인터넷의 급속한 성장은 인터넷 트래픽을 증가시키고 있으며, 특히 트래픽의 상당한 부분이 웹 서비스에 관한 것으로 인터넷에서 웹의 비중은 점점 커지고 있다. 트래픽의 증가로 인해 웹 서버의 성능(Performance)과 가용성(Availability)을 높이는 방안이 더욱 중요시되고 있으며, 증가하는 웹 서비스 요구에 대처하기 위해서 확장성과 가격 대 성능비가 우수한 웹 서버 클러스터가 많이 연구되고 있다

클러스터(Cluster) 기법이란 병렬 처리 기법의 한계를 뛰어 넘어 시스템의 성능을 향상시키기 위한 방법으로 네트워크에 접속된 여러 개의 서버를 하나로 연결하는 방법을 말하며, 월드 와이드 웹 서비스 기반하의 클러스터 기법을 웹 서버 클러스터(Web Server Cluster)라 한다. 한편, 웹 클러스터를 구현하는 방법으로 구현 아키텍처(Architecture)에 따라 브로드캐스팅(Broadcasting) 방식[1], 라운드 로빈 DNS(Round-Robin DNS) 방식[2], 직접 라우팅(Direct Routing) 방식[3], 디스패처(Dispatcher) 방식[4] 등이 있다.

현재의 웹 서버 클러스터는 클라이언트의 요청을 적절한 웹 서버로 전달해주는 하나의 디스패처(분배기)와 동일한 컨텐츠(Content)를 가진 여러 대의 서버 노드로 구성되어 있는데 클라이언트들로부터의 서비스 요청을 디스패처에서 특정한 하나의 웹 서버에 집중되지 않도록 적절한 스케줄링

방식으로 전체 서버에게 전달하는 디스패처 방식으로 이루어지고 있다. 위 설명은 그림1과 같다.

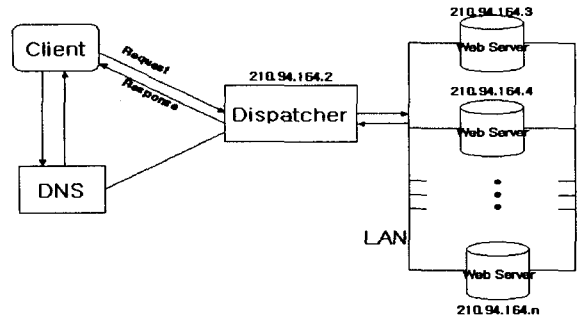


그림 1 기존의 디스패처 방식 구조

이러한 디스패처 방식은 모든 클라이언트의 요청을 제어할 수 있지만, 디스패처의 처리 작업을 통해서 웹 서버에게 전달된다. 디스패처로 도착하는 모든 패킷들을 관리하고 변경하는 작업들은 많은 처리비용이 들고 클라이언트의 요청이 많아지면 병목현상을 일으킬 수 있으며[5], 또한 문자(text) 정보 같은 작은 데이터만 요구하는 패킷(HTML)을 받는 경우는 별 문제가 되지는 않지만 요즘과 같이 기본적인 문자 뿐만 아니라 동영상 파일, 음악 파일, 이미지 파일과 같은 많은 양의 멀티미디어 데이터(MPEG, MP3)를 요구하는 패킷을 받는 경우엔 웹 서버간 부하가 균등하더라도 요청된 데이터의 크기가 크면 특정 웹 서버의 부하는 증가되고 전

채 웹 서버의 부하는 불균등하게 되어 패킷을 처리하는데 많은 시간이 걸리거나 심각한 경우 더 이상 서비스를 할 수 없게 될 수도 있다. 이러한 문제점을 개선하기 위해서 본 논문에서 제안한 웹 서버 클러스터 기법을 소개한다.

2. 본 논문에서 제안하는 웹 서버 클러스터 기법

본 논문에서 제안하는 웹 서버 클러스터 기법은 디스패처에서 각 웹 서버에 대한 효과적인 부하분산을 하는데 중점을 두는 것이 아니고, 디스패처의 트래픽에 대한 오버헤드를 감소시키고, 클라이언트의 요청에 대한 서비스 응답 시간을 최소화하는데 목적을 둔다. 제안하는 웹 서버 클러스터 구조도는 그림 2와 같다.

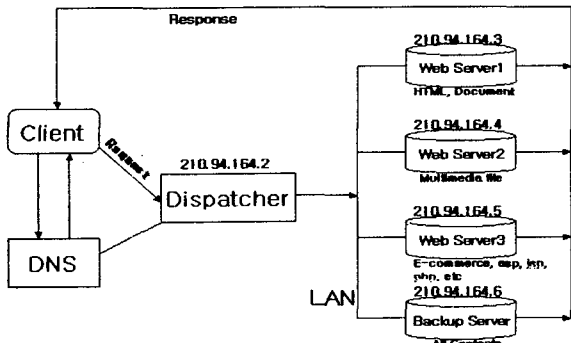


그림 2 제안한 웹 서버 클러스터 전체 구조도

제안하는 웹 서버 클러스터 기법의 구성은 들어오는 모든 클라이언트의 요청에 대해 각각의 콘텐츠를 가지고 있는 웹 서버로 전달하는 하나의 디스패처와 서비스에 대한 응답을 하는 세 개의 웹 서버 그리고 세 개의 웹 서버들 중 파일에 대처하기 위한 하나의 백업 서버를 가지고 있다. 각 서버에 대한 IP 주소가 있는데, 클라이언트 입장에서는 디스패처를 서버로 인식하여 모든 클라이언트는 디스패처의 단일 IP 주소만으로 접근 하며, 웹 서버 자체는 감추어지게 된다. 디스패처는 IP 패킷 단위로 처리를 하며, 클라이언트에서 송신된 IP 패킷을 해당 웹 서버로 전달하는데 이때 클라이언트 IP 패킷 헤더 정보의 목적지 주소를 실제 웹 서버의 IP 주소로 바꾸는 작업을 통해 실제 웹 서버에게 요청을 전달한다. 요청을 전달받은 웹 서버는 클라이언트에게 응답할 때 자신의 주소를 디스패처 서버의 주소로 변환하여 응답을 한다. 이 IP 패킷 헤더 정보 수정(Rewriting) 과정을 도식화하면 그림 3과 같다.

한편, 디스패처는 웹 서버들의 콘텐츠에 대한 인덱스 파일을 가지고 있어서 클라이언트의 요청이 들어오면 해당 웹 서버에게 요청을 전달한다. 또한 디스패처 서버는 웹 서버들에 대한 모니터링을 하면서 특정한 웹 서버가 파일이 받

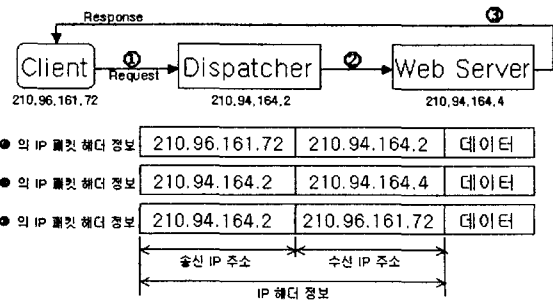


그림 3 IP 패킷의 헤더 정보 Rewriting

생할 경우 파일된 서버의 서비스 요청을 백업 서버에게 전달 한다.

세 개의 웹 서버는 각각 다른 콘텐츠의 형태(Type)를 가지고 있다. 그림 2에서 보는 것처럼 서버1은 정적 콘텐츠(HTML, Document, Image, etc.)를 가지고 있고, 서버2는 멀티미디어 콘텐츠(video, audio, etc.)를 가지고 있다. 서버3은 동적 콘텐츠(전자상거래, ASP, JSP, etc.)를 가지고 있다. 따라서 디스패처 서버에서 전달된 클라이언트 요청을 해당 웹 서버에서 응답하게 된다. 세 개의 웹 서버 중 특정한 한 개의 웹 서버가 고장이 발생했을때 백업 서버가 이를 대처한다. 백업 서버는 다른 웹 서버의 모든 콘텐츠를 가지고 있어서 어떤 웹 서버의 파일에도 대처할 수 있다. 클러스터로 구성된 웹 서버들은 LAN으로 구성되어져 있다. 각 웹 서버들은 해당 콘텐츠를 업그레이드 했을 경우 백업 서버에게 업그레이드한 콘텐츠를 복사 해야 한다.

2.1 동작 원리

제안한 웹 서버 클러스터의 동작은 그림 4와 같다. 클라이언트가 디스패처 서버 이름이 포함된 URL로 DNS에게 주소를 요청하고 DNS는 디스패처 서버의 주소를 응답한다. 요청 받은 디스패처 서버는 클라이언트의 요청에 맞는 콘텐츠를 가지고 있는 웹 서버에게 요청을 전달하게 된다. 요청을 처리한 웹 서버는 자신의 주소를 숨기고 디스패처 서버의 주소로 클라이언트에게 응답을 하고 클라이언트는 실제 웹 서버의 주소를 모른 채 디스패처에게 서비스 요청을 계속하게 된다.

3. 성능 평가

본 논문에서 제안한 웹 서버 클러스터 기법을 객관적으로 평가하기 위하여 1대의 디스패처와 3대의 웹 서버, 그리고 백업 서버 1대로 구성하였으며, 시스템 사양은 Pentium III 1.133MHz, 256 RAM, HDD 30GB로 구성하였다.

또한, WOW Linux 7.0 운영체제에서 UC Berkely 대학의

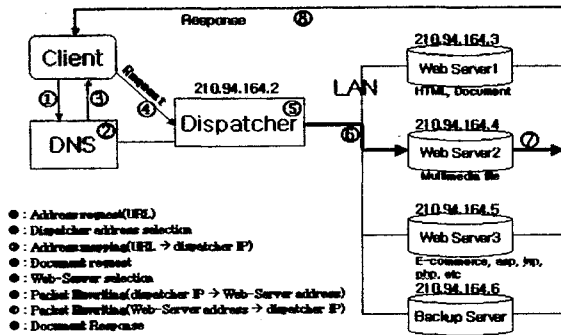


그림 4 제안한 웹 서버 클러스터 동작원리

네트워크 시뮬레이터인 ns-2.1b8a를 사용하여 시뮬레이션 하였다. 성능 평가를 위해 사용한 Data Set은 동국대학교 웹 서버 로그 파일을 표본 추출하여 Data Set으로 이용하였고, 성능 비교는 앞에서 언급한 디스패처 방식(부하가 적은 서버를 선택하는)과 제안한 웹 서버 클러스터 기법에 대해 실험하였다.

두 가지 방식에 대한 처리량(Throughput)을 보면 제안한 웹 서버 클러스터가 기존의 디스패처 방식보다 정적 콘텐츠(HTML, Document, Image, etc.)가 69%, 멀티미디어 콘텐츠(video, audio, MPEG, etc.)는 56%, 동적 콘텐츠(전자상거래, ASP, JSP, PHP, etc.)는 53%가 각각 증가하였음을 알 수 있었고, 응답 시간에서도 지수 분포 모델로 실험했을 때 기존의 디스패처 방식은 평균 2.7초에 반해 제안한 웹 서버 클러스터는 응답 시간이 평균 2.3초로 16%정도 더 짧다는 것을 보았다. 전체적으로 제안한 콘텐츠를 기반으로한 웹 서버 클러스터 방식이 기존의 디스패처 방식보다 성능면에서 월등히 향상되었음을 알 수 있었는데, 이러한 이유는 오래 걸리는 클라이언트 요청으로 인해 짧은 요청이 기다려야 하는 기존의 방식을 제안한 콘텐츠를 기반으로한 웹 서버 클러스터 기법으로 해결했기 때문이다.

4. 결론 및 향후 과제

인터넷 사용자의 기하급수적 증가는 정보 제공 서버의 과부하를 초래하고 있다. 또한 웹을 포함한 인터넷 서비스가 전자 상거래와 같은 사업의 중요한 도구가 됨에 따라 인터넷의 중단되지 않고 연속적으로 서비스 하는 것과 클라이언트 요청에 대한 서비스 응답 시간이 매우 중요한 문제로 대두되고 있다. 이를 해결하기 위해 기존의 방법들은 부하분산을 중심으로 많은 연구가 되고 있는데, 이는 분산 웹 서버 시스템이 잘 계획되지 않는다면 WWW의 복잡성과 불규칙성으로 인해 전체적인 시스템 성능이 저하될 수 있다.

본 논문에서는 클러스터로 구성된 웹 서버들은 각각 다른

컨텐츠를 가지고 있으므로 부하분산하는 알고리즘을 적용하지 않고 디스패처는 클라이언트 요청에 맞는 콘텐츠를 가지고 있는 웹 서버에게 전달하는 모델을 제안 하였다. 제안한 웹 서버 클러스터는 기존의 솔루션에 비해 다음과 같은 장점이 있다.

첫 번째로 제안한 웹 서버 클러스터는 부하 분산 기능이 없으므로 디스패처에 대한 오버헤드가 적어 전체 웹 서버의 성능을 효율적으로 운용한다. 두 번째로 기존의 디스패처 방식의 서비스 응답이 디스패처를 거치지 않고 클라이언트에게 직접 전달되기 때문에 응답 시간이 감소된다. 세 번째로 특정한 웹 서버의 패일이 발생하더라도 백업 서버에서 쉽게 대처할 수 있다.

제안한 웹 서버 클러스터의 단점으로는 디스패처가 다운되었을때 서비스가 마비될 수 있고, 인기있는 특정한 웹 서버에 많은 트래픽이 발생할 수 있다. 따라서 앞으로의 연구 과제로는 주 디스패처가 다운되었을 때 전환 가능한 보조 디스패처를 구현하는 방법과 특정한 웹 서버에 몰리는 트래픽을 분배 처리하는 연구도 진행되어야 할 것이다.

5. 참고 문헌

- [1] Kangashafju, J. Ross, K. W., "A clustering structure for reliable multicasting," Computer Communications and Networks, 1999. Proceedings. Eight International Conference, pp.378-383, 1999.
- [2] Kangasharju, J., Ross, K. W., "A replicated architecture for the Domain Name Syster," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE computer and Communications Societies. Proceedings. IEEE, Vol.2, pp.660-669, 2000
- [3] Dongeun Kim, Cheol Ho Park, Deayeon Park, "Request rate adaptive dispatching architecture for scalable Internet server," Cluster Computing, 2000. Proceedings. IEEE International Conference. pp.289-296, 2000.
- [4] Canal, R., Parcerisa, J. M., Conzalez, A., "Dynamic cluster assignment mechanisms," High-Performance Computer Architecture, 2000. HPCA-6. Proceedings. Sixth International Symposium, pp.133-142, 1999.
- [5] M. Colajanni, P.S Yu, V. Cardellini, "Dynamic load balancing in geographically distributed heterogeneous Web-servers." Proc. of 18th IEEE Int'l. Conf. on Distributed Computing Systems(ICDCS'98), pp.295-302, Amsterdam, The Netherlands, May 1998.