

# 분산 웹서버에서의 효과적인 부하 균등 정책

노진홍<sup>0</sup> · 홍영식  
동국대학교 컴퓨터공학과  
{jhno, hongys}@dgu.edu

## Efficient Load Balancing Policy in Distributed Web Server Systems

Jin-Hong Lho<sup>0</sup>, Young-Sik Hong  
Dept. of Computer Engineering, Dongguk Univ.

### 요 약

인터넷의 급성장은 웹서버의 요청율을 급격하게 증가시키고 심각한 정체를 일으키고 있다. 이를 해결하기 위하여 클라이언트의 요청을 분산된 여러 서버들에게로 전달시키는 방법이 많이 사용되고 있다. 이 방법은 웹서버의 처리량을 향상시키고 확장성과 가용성을 제공하지만 서버들의 부하를 균등하게 하는 방법이 필요하게 된다. 이러한 부하 균등을 위해 크게 DNS(Domain Name Server)를 활용한 정책과 IP 단계에서 분배기(Dispatcher)를 사용한 정책이 제안되었다. 본 논문에서는 전자와 후자의 단점을 보완하고 성능을 향상시킨 다중 분배기를 제안하고 실험하였다.

### 1. 서론

WWW의 폭발적인 통신 증가는 유명한 웹사이트로의 요청율을 급격하게 증가시켜 심각한 정체를 일으킨다. 이런 유명한 사이트의 관리자는 사용자의 요구를 만족시켜 주기 위해서 웹서버의 성능을 향상시킬 필요를 접하게 된다.

이를 위한 해결방법은 사용자에게 투명성을 제공하면서 여러 서버들에게로 요청을 분산시킬 수 있는 분산구조를 가지는 것이다.[1] 하지만 하나의 서버에 많은 양의 요청들이 대기하고 있는데 반하여 다른 서버들은 요청이 없는 상태가 된다면 전체적인 시스템의 성능이 저하될 수 있다. 다시 말하면 부하 균등 방법의 효율성에 따라 가용성과 확장성이 결정되므로 클라이언트의 요청을 웹서버에게로 균등하게 할당해주는 방법이 필요하다. 이러한 방법으로는 URL 이름을 서버 중 하나의 IP 주소로 매핑 시켜주는 DNS를 사용하는 방법과 IP 단계에서 패킷의 주소를 변경하는 방법이 있다. IP 분배기를 사용한 시스템은 모든 요청을 제어할 수 있는 반면 지역적으로 인접한 클러스터링 웹서버에게만 적용 가능하다. 또한 잦은 요청으로 인하여 부하가 많을 경우에는 패킷의 주소를 변경하는 작업으로 인해 IP 분배기가 병목현상을 일으키게 된다. 이에 반해 DNS를 사용한 방법은 병목현상을 일으키지 않고 멀리 떨어진 분산웹서버에도 쉽게 적용시킬 수 있다. 하지만 문제점은 IP-주소 캐싱으로 인하여 제한된 요청만을 제어할 수 있다는 것이다.

본 논문에서는 여러 개의 IP 분배기를 사용하고 이에 맞는

알고리즘을 사용함으로써 DNS와 IP분배기의 문제점들을 보완한 다중 IP 분배기에 대해 제안하고자 한다. 이 방법은 사용자들에게 투명하고 빠른 응답시간을 제공하며, 요청에 대한 제어도 효율적으로 할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 기존의 DNS 분배 기법과 IP 분배 기법에 대해 알아본 후, 3장에서는 시스템 모델을 설명하고 다중 IP 분배기 시스템을 제안하며, 4장에서는 실험결과를 바탕으로 기존방법과 비교한다. 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

### 2. 관련연구

클라이언트는 웹서버에게 문서를 요청하기 위해 URL에 포함된 호스트의 이름을 IP 주소로 변환하는 DNS를 사용하고, DNS는 서버의 주소와 주소의 유효시간(TTL)을 명시한다. TTL 기간 동안에는 중간에 있는 네임 서버들(intermediate name server)이 주소변환을 담당하지만 TTL이 만기되면 클라이언트는 DNS에게 다시 주소변환을 요청한다. 그러므로 TTL값이 작을수록 DNS가 서버들을 선택할 수 있고, TTL값이 클수록 클라이언트의 요청을 제어할 수 없게 된다. 그러므로 적절한 TTL값의 선택과 서버를 선택하는 DNS의 알고리즘에 따라 부하를 균등하게 할 수 있다.[7] 그러나 주소 변환(address translation)으로 인한 통신 부하를 줄이기 위해 중간 네임 서버를 비롯한 여러 개체들이 서버의 주소를 캐싱하기 때문에 DNS는 아주 적은 양의 요청들만 자신의 정책대로 제

어할 수 있다. 일반적으로 DNS는 클라이언트가 요청을 할 때마다 라운드 로빈(Round Robin)방식으로 서버를 선택[8]하지만 클라이언트와 서버들의 상태 정보에 따라 서버와 TTL값을 적응적으로 선택하는 알고리즘이 성능이 더 좋은 것으로 알려져 있다.[5]

DNS를 사용한 정책의 대안은 클라이언트의 요청을 모두 제어하면서 여러 서버들 중 하나의 서버로 연결시켜 주는 분배기를 사용하는 것이다. 이 방법은 투명성을 제공하기 위해 웹 서버들에게 단일 가상 IP 주소(Single Virtual IP Address, 이하 IP-SVA)를 제공한다. IP-SVA는 중앙 집중식 스케줄러인 분배기의 주소이고, 분배기만이 해당 서버들의 주소를 유일하게 알기 때문에 클라이언트는 분배기의 주소만 알 수 있다. 그러므로 클라이언트가 IP-SVA를 가진 분배기에게 요청을 하면 분배기는 적절한 서버에게 요청을 전달하고, 실제로는 서버가 작업을 처리한다. 분배기에서 실제 서버에게 요청을 전달하는 방법에 따라 크게 패킷주소변환 방법[4]과 HTTP 방향변환 (HTTP redirection)방법[6]이 있다. 하지만 모든 요청을 관리하는 분배기가 병목현상을 발생할 수 있기 때문에 분배기를 사용한 정책들은 서버를 선택하는데 간단한 알고리즘을 사용한다는 단점이 있다.

3. 다중 IP 분배기 시스템

한 개의 분배기를 가진 시스템이 부하를 균등하게 하는 데는 성능이 좋겠지만, 실제 웹 환경과 같은 부하를 가진 헤비테일(heavy-tail)모델에서는 클라이언트의 요청을 변화가 심하고 서버가 과부하되는 경우가 많기 때문에 한 개의 분배기를 가진 시스템에서는 병목현상이 발생할 수밖에 없다.[3] 다중 분배기는 기존 모델처럼 모든 서버들을 관리하지 않고 자신에게 주어진 서버들만을 관리하므로 기존의 한 개의 분배기가 모든 서버를 관리하는 것보다 처리할 작업이 줄어들게 되어 병목현상을 제거할 수 있고 응답시간도 줄일 수 있다. 또한 다중 분배기는 자신이 관리하는 서버들의 부하를 계산하고 일정시간마다 다른 분배기들과 정보를 교환하므로 특정 분배기에 부하가 걸릴 수 있는 점을 방지하고 한 개의 분배기만 있을 경우만큼 부하를 균등하게 할 수 있다.

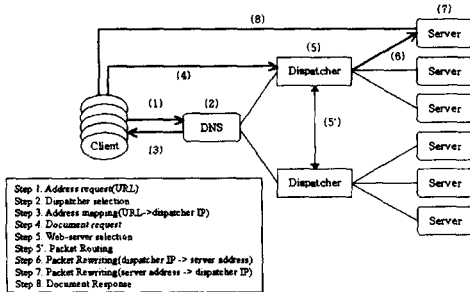


그림 1 다중 분배기 구조

본 논문의 목적은 클라이언트의 모든 요청을 제어하여 서버의 부하를 균등하게 하고, 응답시간을 줄일 수 있는 시스템

을 제안하는 것이다. 그림 1은 본 논문에서 제시하는 다중 IP 분배기 구조이다.

클라이언트가 DNS에게 서버의 이름인 URL주소로 요청하면 DNS는 실제 서버가 아닌 분배기의 주소와 TTL값을 응답한다. DNS는 라운드 로빈 방식으로 분배기를 선택하고 TTL은 무한대로 설정한다. TTL값을 무한대로 설정하는 이유는 실제 웹 환경에서는 클라이언트들의 요청이 끝고루 분배되어 있지 않고 특정한 클라이언트의 요청이 대부분을 차지[2]하므로 TTL값의 의미가 크지 않기 때문이다. 그러므로 클라이언트는 DNS로부터 처음에 배정받은 분배기만으로 요청을 하게 되어 분배기로의 요청률 변화가 심하지 않게 되고, 그로 인해 분배기의 상태정보를 계산하여 교환하는 작업이 빈번하지 않아도 되므로 분배기의 부담이 줄어든다.

만약 한 분배기의 요청률이 다른 분배기보다 임계값 이상으로 높아지게 되면 분배기는 요청률이 가장 낮은 분배기로 요청 방향을 바꾸어 준다. 요청 방향을 바꾸려면 분배기가 클라이언트에게 HTTP의 301번 상태 코드를 응답하면 된다. 301번은 요청한 서버가 영구히 다른 곳에 존재한다는 뜻을 의미하는 코드로서, 응답 메시지의 헤더 필드에 다른 분배기의 IP주소를 가진 'Location'이 포함된다. 이후로 클라이언트는 이전한 분배기로 요청을 계속하게 되어 역시 요청률의 변화에 별로 영향을 주지 않는다.

분배기는 서버의 부하 정도를 알아내어 클라이언트의 요청을 자신이 관리하고 있는 서버로 전달시키고 요청에 포함되어 있는 객체의 수(hit)를 계산한다. 이때 분배기는 서버 사용률을 처리해야 하는 hit수로 측정하고, hit 수가 가장 적은 서버에게로 클라이언트의 요청을 전달한다.

4. 실험 및 분석

특정 웹서버만 과부하 되지 않고 부하 균등하게 작동하도록 하는 것을 평가하려면 서버 사용률의 표준편차가 중요한 평가 척도가 될 것이다. 시스템내의 모든 서버들 중에서 주어진 시간내 최고의 서버 사용률을 시스템 최고 사용률(System Maximum Utilization)이라고 정의할 때, 특별히 중요한 평가 척도는 시스템 최고 사용률의 누적빈도(Cumulative Frequency)이다. 즉 시스템 최고 사용률이 어떤 값 이하일 확률이 누적빈도이다. 특정 기간동안의 시스템 최고 사용률을 추적함으로써 부하 균등 알고리즘들의 성능을 평가할 수 있다.

시뮬레이터는 UCB/LBNL/VINT Network Simulator (version 2)를 사용하였고, 실험은 지수분포 모델과 헤비테일 모델을 사용하였다.

지수 분포 모델에서는 부하가 지수 분포되었다고 가정하였다. 헤비테일 모델에서는 중요한 특징들이 모두 실제 웹 부하와 비슷하고, 부하의 변화율이 매우 크다. 자세한 인수들은 다음과 같다.

인수	지수 분포 모델	헤비 테일 모델
세션당 웹페이지 수	Exponential (mean 12)	Inverse Gaussian ( $\mu=3.86, \lambda=9.46$ )
페이지사이 시간	Exponential (mean 1)	Pareto ( $\alpha=1.4, \kappa=2$ )
페이지당 포함된 hit 수	Uniform in {5-15}	Pareto ( $\alpha=1.4, \kappa=2$ )
hit사이의 시간	Exponential (mean 0.25)	Weibull ( $\alpha=1.4, \beta=2$ )
hit에 포함된 객체 크기	Exponential	Pareto ( $\alpha=1.383, \kappa=2$ )

그림 2와 3에서 각각의 X축은 시스템 최고 사용률, Y축은 누적 빈도를 나타낸다.

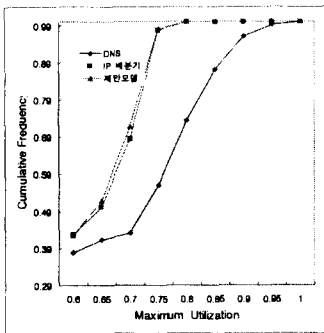


그림 2 지수 분포 모델

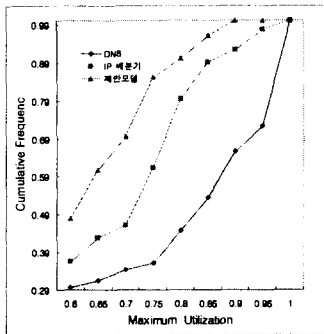


그림 3 헤비 테일 모델

지수 분포 모델로 실험한 그림 2를 보면 DNS를 사용한 시스템의 성능이 가장 좋지 않은 것을 알 수 있고, 한 개의 분배기를 가진 시스템과 제안한 시스템의 성능이 비슷한 것을 알 수 있다. 이 그래프를 보면 분배기 시스템과 제안된 시스템은 실험 기간동안 80%이하의 사용률만을 기록하였다. 즉 병목 현상을 일으킬 정도로 과부하가 되지 않았기 때문에 두 모델의 성능이 비슷하다. 하지만 실제 웹 환경과 비슷한 헤비 테일 모델을 실험한 그림 3을 보면 한 개의 분배기를 가진 시스템이 병목현상을 일으켜서 제안된 시스템의 성능보다 좋지

않고, DNS를 사용한 시스템이 가장 좋지 않은 것을 알 수 있다.

### 5. 결론 및 향후 연구 방향

웹 서버 시스템이 좋은 성능을 발휘하기 위해서는 부하 균등이 아주 중요하다. DNS에 기반한 알고리즘은 요청에 대한 제어를 할 수 없고, 분배기에 기반한 알고리즘은 병목현상을 일으킨다는 단점을 가지고 있었다. 본 논문에서는 기존의 DNS에 기반한 방법과 분배기에 기반한 방법의 문제점을 해결한 다중 분배기 시스템을 제안하였고, 부하 균등이 효과적으로 되었음을 알 수 있었다.

향후에는 서버의 부하정보에 대한 척도를 다양화하여 웹서버를 선택하는 알고리즘을 좀더 개선하고 다중 분배기 시스템의 고장 감내 능력을 향상시킬 것이다.

### 6. 참고문헌

- [1] D. Andresen, T. Yang, V. Holmedahl, O. H. Jbarra, "SWEB: Toward a scalable World Wide Web server on multicomputers", Proc. of IPPS'96, Honolulu, April 1996.
- [2] M. F. Arlitt, C. L. Williamson, "Web sever workload characterization: The search for invariants", IEEE/ACM Trans. on Networking vol. 5, no. 5, pp. 631-645. Oct. 1997.
- [3] P. Barford, M. Crovella, "Generating representative Web workloads for network and server performance evaluation", Proc. of ACM Sigmetrics '98, Madison, WI, pp.151-160, June 1998.
- [4] A. Bestavros, M. E. Crovella, J. Lie, D. Martin, "Distributed Packet Rewriting and its application to scalable Web server architectures", Proc. of 6th IEEE Int'l. Conf. on Network Protocols(ICNP'98), Austin, TX, Oct. 1998.
- [5] V. Cardellini, M. Colajanni, P.S. Yu, "Dynamic load balancing on Web-server systems", IEEE Internet Computing, Vol.3, No. 3, pp. 28-39, May-June 1999.
- [6] V. Cardellini, M. Colajanni, P.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.
- [7] M. Colajanni, P.S. Yu, D.M. Dias, "Analysis of task assignment policies in scalable distributed Web-server systems", IEEE Transaction on Parallel and Distributed Systems, vol. 9, no. 6, June, 1998.
- [8] T. T. Kwan, R. E. McGrath, D. A. Reed, "NCSA's World Wide Web server: Design and performance", IEEE Computer, vol. 28, no. 11, pp.68-74, Nov.1995.