

클러스터 웹서버에서의 로드 밸런싱에 대한 퍼지접근

배원열*, 박종규*, 김학배*
*연세대학교 전기전자공학과
e-mail: anybuy@yonsei.ac.kr

A Fuzzy Approach to Load Balancing in a Cluster Web Server

Won-Youl Bae*, Jong-Gyu Park*, Hag-Bae Kim*

*Dept. of Electrical and Electronic Engineering, Yonsei University

요약

클러스터 웹서버는 실질적인 네트워크 컴퓨팅 어플리케이션을 구현하기 위한 중요한 플랫폼으로서 생각되어 왔으며, 그 필요성이 더해가고 있다. 이러한 환경 속에서 트래픽의 증가는 놀라울 정도로 증가하여 다중 호스트 사이의 부하분배 및 효과적인 관리를 위해서 로드밸런싱 기술을 적용할 수 있다. 그러나 네트워크 전역상태(global state)의 불확실성을 반영하고, 고유의 습성을 예상하여 정확한 작업 배치 결정을 하기 위한 적당한 수학적 모델이 많지 않다. 그러한 측면에서 퍼지론적은 수학적으로 모델링하기 어려운 문제를 해결하여 전역상태의 불확실성을 효과적으로 반영할 수 있는 강력한 패러다임의 하나이다. 본 논문은 이러한 퍼지론적 접근을 통하여 클러스터 웹서버 환경에서의 동적 부하분배를 실현하는 시스템설계 와 도구를 제안한다.

1. 서 론

인터넷 인구의 폭발적인 성장은 서버/클라이언트 환경에서 트래픽 혼잡을 야기 시켜 왔으며, 인기가 높은 웹사이트의 경우, 특히 가장 많이 사용하는 시간대에는 짧은 시간에도 금방 과부하가 걸리기 쉽다. 게다가 단순한 정보의 전달에 그쳤던 인터넷의 기능이 확장되어, 정보의 전송만이 아닌 저렴한 비용으로서 빠른 전송과 정확성, 안정성, 확장성에 대한 요구 또한 증대되고 있다. 따라서 이를 해결하기 위한 방법으로 여러 개의 서버에 같은 웹 서비스를 운영하며, 정보를 인터넷에 제공하는 웹 클러스터가 있다[1]. 이러한 환경 속에서 시스템 전체의 성능을 높이기 위해서는 서버들 사이에 동적 부하분배를 위한 효과적인 로드밸런싱 기술이 요구된다. 따라서 이러한 목표를 달성하기 위해 각 시스템 노드들은 상태정보를 교환하고 전역상태(global state)에서 지역상태(local state)의 예측정보를 바탕으로 제어를 결정하지 않으면 안 된다. 그러나, 전역상태의 불확실성으로 인한 정확한 작업 배치결정을 할 수 없어 부하불균형이 발생하고 결과적으로 클러스터 웹서버의 성능 향상에는 많은 기여를 하지 못하였다. 분산

시스템의 효율설계에 있어서 항상 주된 이슈가 되어 온 이러한 문제는 효과적으로 배치결정을 하기 위한 수학적 모델링을 하기가 어렵다는데 그 원인이 있다. 본 논문은 그러한 환경 속에서 부하분배 문제를 다루기 위하여 전역상태에서 불확실성을 야기시키는 소스들을 모델링 하는 방법으로 퍼지원리를 제안한다. 지금까지 퍼지론적을 사용하여 로드밸런싱을 효율적으로 하기 위한 많은 연구들이 있어왔다. Stancovic[2]은 전체 시스템 상태에서의 불확실성을 관리하는 작업 스케줄링을 위하여 Bayesian decision theory를 적용했으며, Kumar[3]는 로드밸런싱을 위해 fuzzy expert system을 제안했다. fuzzy expert system은 상태정보에서 불확실성을 반영하며, 퍼지론적을 바탕으로 스케줄링을 결정한다. 또한, [4]에서 제안된 fuzzy-based consistency model은 각 컴퓨팅 머신이 예측된 consistency relaxation을 바탕으로 탄력적인 스케줄링과 상태 업데이트 결정을 할 수 있도록 한 메커니즘이다. 그러나 지금까지의 접근 방법들이 리눅스 기반의 클러스터 웹서버에서 적용되기 위해서는 세밀한 수정이 되

어져야 한다. 본 논문에서 제안한 퍼지 로드밸런싱 서비스는 모든 가용 서버에게 클라이언트의 요청을 위임하며, 어떠한 퍼지규칙에 근거하여 클라이언트의 요청들을 배치한다. 본론에서는 전역상태의 불확실성을 명백히 반영할 수 있는 시스템의 파라미터에 대해서 논의하고 이러한 파라미터를 기반으로 만들어진 퍼지컨트롤 규칙의 설명과, 출력 값으로 정량화된 서비스 우선순위에 의해 로드밸런싱 하는 과정을 설명한다.

2. 클러스터 웹서버의 환경 및 퍼지 로드 밸런싱

본 논문에서는 리눅스 클러스터 웹서버의 플랫폼으로서 다이렉트 라우팅을 이용한 가상서버를 고려하며, 스케줄링 알고리즘으로는 퍼지 스케줄링 로드밸런싱 알고리즘을 고려한다. 또한 네트워크에서 각각의 프로세싱 요소들은 네트워크를 통하여 다른 것들과 직접적인 통신과 명령을 수행하기 위한 자신의 메모리를 가지고 있다고 전제한다. 동적 로드밸런싱을 실행하는 퍼지 로직 모델은 입력값으로 부정확한 값을 갖는다. 그리고 이 입력값들은 서로 다른 우선순위를 가지는 서비스들을 분석하고 할당하는 퍼지 추론 엔진으로 전달된다. 그러면 퍼지 로직 로드 밸런싱 서비스는 클라이언트의 요청을 적절한 서버에 라우팅 할 수 있게 된다. 본 제안에서는 로드 밸런서가 서비스의 유용성을 나타내는 서비스 우선순위를 바탕으로 라우팅 하는 것을 고려한다. 이론적으로 퍼지 로직은 아날로그 프로세스 혹은 수학적으로 모델링 하기 어려운 자연현상을 나타내기 위한 방법이다[5]. 이러한 아날로그 프로세스들은 사실상 연속적이고 이산적인 세그먼트들로서, 쉽게 분류되어 질 수 없는 것들이다. 일반적 퍼지시스템의 구성 요소는 1)입력변수의 퍼지화, 2)퍼지추론 규칙의 적용, 3)결과 값들의 비퍼지화의 세 가지 단계를 포함한다. 그림 1은 본 논문에서 적용한 퍼지로직 컨트롤의 구조를 나타낸 그림이다.

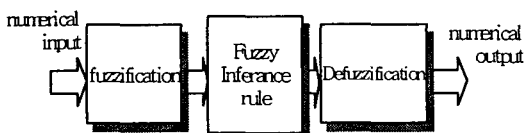


그림 1. 퍼지 로직 추론엔진의 구조

청들을 가장 적합한 서버에게 라우팅을 할 수 있으며 로드밸런서는 리얼서버의 현재의 상태를 바탕으로 하여 스케줄링 결정을 할 수 있다. 그러나 그러한 상태정보들은 서버에 도착한 클라이언트의 요청으로서 갱신하거나 신뢰할 수 없다. 왜냐하면 이러한 요청들은 네트워크의 지연 등으로 인한 서버들의 상태를 정확하게 반영할 수 없기 때문이다. 따라서 로드밸런서는 시스템효율을 만들기 위해서 퍼지정보를 다루는 적당한 이론을 적용하는 것이 필요하다. 정확한 라우팅결정 하기 위한 퍼지로직 알고리즘에 사용 될 언어변수들에 대한 설명은 다음과 같다.

2.1 서버의 부하

서버들의 상태를 나타내며, 퍼지 추론 엔진의 입력값으로 사용될 언어적 변수로서 서버부하 및 서버의 응답성으로 고려한다. SL(Server Load)로 표기되는 서버 부하는 server가 얼마나 바쁜 상태에 있는지를 나타내는 것으로서 많은 요소들에 의해서 결정되어 질 수 있다. 예를 들면 서버에서의 작업의 도착율, 작업 실행시간, 프로세스의 속도, 메모리 가용성 등으로서 임의의 시간 T에서 측정된 상태값이다. 본 제안에서는 이러한 서버부하의 fuzzy set을 (low(L), medium(M), high(H))로 정의한다. 시스템에 있어서의 정확한 부하계산은 컴퓨터 조作的 많은 오버헤드가 필요하기 때문에, 우리는 서버부하를 측정하는 방법으로서 각 프로세스의 실행시간을 직접적으로 측정하지 않고 어플리케이션 벤치마킹 프로그램 실행시간을 측정하는 간접적인 방법을 사용한다. 이 어플리케이션 벤치마킹 프로그램은 백그라운드 프로세스로서 시스템에서 쉘 없이 동작하며 이 프로그램의 동작시간을 측정함으로써 시스템에서의 순간적인 부하레벨을 추론 할 수 있다. 그림 2는 SL에 대한 멤버십 그래프를 나타낸 것으로서 u는 멤버십의 정도를 나타낸다.

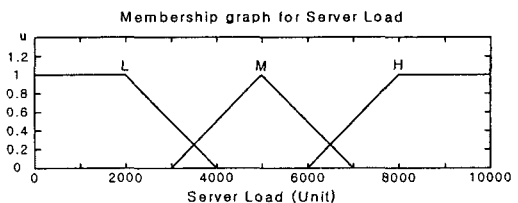


그림 2. SL에 대한 멤버십 그래프

일반적으로 라우터나 로드밸런서는 클라이언트의 요

2.2 서버의 응답성

퍼지 추론 엔진의 다른 또 하나의 입력값으로서, SR(Service Rank)로 표기되는 서버의 응답성을 고려한다 서버의 응답성을 측정하는 방법으로서는 RMI 시간을 측정하는 방법[6]등이 있으나 본 제안에서는 클라이언트 요청에 대한 서버의 응답이 클라이언트로 전달되는 대역폭을 측정한다. 서버의 응답성은 컴퓨팅 서버에서 대역폭은 매우 정확하게 측정될 수 있다. 이러한 대역폭의 단위는 시간당 킬로바이트로 나타내며, fuzzy set은 {small(S), medium(M), wide(W)}로 정의한다. 그림 3은 서버의 응답성에 대한 멤버십 그래프를 나타내고 있다.

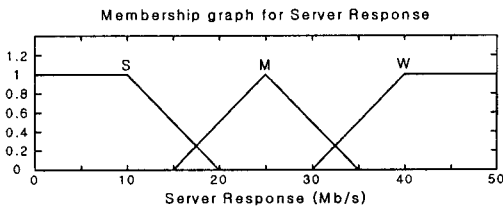


그림 3. 서버의 응답성에 대한 멤버십 그래프

2.3 서비스 우선순위

본 논문에서 로드밸런서가 서버에게 클라이언트의 요청을 적절히 배치결정을 하기 위한 파라미터로 서비스 우선순위를 고려한다. 서비스 우선순위는 서버의 현재 상태를 나타내는 최종 파라미터로서 서버부하와 서버의 부하와 서버의 응답성을 바탕으로 만들어진다. SP(service priority)로 표기되는 서비스 우선순위의 퍼지 셋은 다음의 6개로 정의된다. {very low(VL), low(L), medium low(ML), medium(M), medium high(MH), high(H)}. 여기서 서비스 우선순위는 서비스 우선순위가 높으면 높을수록 클라이언트의 요청을 보다 적절히 수용할 수 있다는 것을 의미한다. 그림 4는 SP에 대한 멤버십 그래프를 나타낸다.

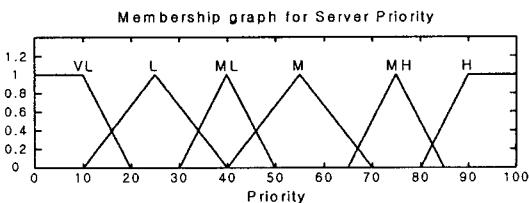


그림 4. SP에 대한 멤버십 그래프

2.4 퍼지추론규칙

앞에서 설명한 퍼지 변수들을 정의한 후 추론규칙

의 셋(set)을 정의하면 (표 1)과 같다.

표 1. 퍼지 추론 규칙

SR \ SL	L	M	H
S	H	MH	ML
M	MH	M	L
W	M	L	VL

(표 1)에서 퍼지추론을 적용함으로써 배치결정을 만들 수 있다. 즉 SR이 wide 이고 SL가 low이면 SP는 high가 된다. 이 퍼지추론규칙 과 멤버십그래프를 가지고 퍼지화와 비퍼지화의 과정은 다음과 같이 실행될 수 있다. 먼저 SL의 입력값은 그들의 멤버십 그래프에 있는 각각의 멤버십 범위값으로 사상된다. 이러한 범위값들은 비교되어지고 둘 중의 최소값은 그들의 결과값의 멤버십 함수에 투영된다. 사다리꼴 형상에서 출력 그래프는 추론규칙의 출력을 나타낸다. 출력그래프가 만들어지고 나서 퍼지출력을 수치화 시키는 비퍼지화가 실행될 수 있다. 본 제안에서는 출력을 퍼지화 하기 위해서 Centroid Method[4]를 이용한다. $i=1,2,3..N$ 에 대하여 N개의 중첩된 면적 A_i 의 전체중심은 다음 식 1에 의해 주어진다

$$\bar{x} = \frac{\sum_{i=1}^N \bar{x}_i A_i}{\sum_{i=1}^N A_i} \tag{1}$$

식(1)에서 A_i 와 \bar{x}_i 는 i 번째 규칙에서 얻어진 삼각형과 사다리꼴로부터 각각의 중첩된 영역과 중심이다. 중심과 영역은 삼각형과 사다리꼴에 대하여 계산되어진다. 이 과정은 다른 추론규칙에 대해서 반복이 되며 여기서 입력값들은 중첩된 사다리꼴의 영역을 얻기 위하여 적용된다. 비퍼지화 과정은 서비스의 우선순위를 나타내는 중심값을 생성한다. 그러면 퍼지 로직 로드밸런서 서비스는 얻어진 우선순위를 바탕으로 하여 클라이언트의 요청을 최고의 서비스로 라우팅 할 수 있다. 여기서 만약에 로드밸런스가 가장 높은 우선순위를 가지는 서버에게 모든 요청들을 즉시 전송하면 Thundering Herd Effect[6]가 발생할 수 있는 것에 주의하여야 한다. 이러한 효과를 최소화하기 위하여 퍼지로직 로드밸런서 서비스는 우선순위 라운드 로빈 알고리즘을 바탕으로 하

여 클라이언트의 요청을 서버들에게 스케줄링 한다 [6].

3. 결 론

본 논문에서 리눅스 기반의 클러스터 웹서버 환경에서 퍼지 로드밸런싱에 대해서 논의를 했다. 전역상태의 불확실성을 고려하지 않는 기존의 알고리즘에 비해 불확실성의 효과를 명백히 반영할 수 있도록 퍼지센스에서 시스템의 파라미터를 모으는 방법들을 정의했다. 이러한 파라미터를 바탕으로 하여 퍼지 추론 규칙의 셋이 만들어졌다. 퍼지 컨트롤 규칙은 언어적 변수에 의해서 세워지기 때문에 직관적인 지식과 관찰은 제어 메카니즘으로 쉽게 도입할 수 있다. 퍼지 로드 밸런싱 알고리즘은 클러스터 웹서버 환경에서 전역상태의 불확실성을 명백히 반영하기 때문에 기존의 라운드 로빈과 분산 접근에 비해 중요한 성능 개선을 가져올 수 있다. 하지만 보다 다양한 토폴로지에서 네트워크 유용성을 반영하는 파라미터들 즉 각 서버에 도착하는 부하도착율, 실행시간, H/W, S/W 등을 고려한 부하특징에 대해서 보다 구체적이고 정확한 고찰이 필요 할 것으로 본다.

[참 고 문 헌]

[1] http://kldp.org/Translation/Virtual_Server-KLDP/index.html

[2] J. A. Stankovic, "An application of bayesian decision theory to decentralized control of job scheduling," IEEE Trans. Compue., vol. 36, pp. 117-130, Feb. 1985

[3] A. Kumar, M. Singhal, M. Liu, "A model for distributed decision making: An expert system for load balancing in distributed system," COMPSAC, pp. 507-513, 1987

[4] P. Chulhye and J. Kuhl, "A Fuzzy-Based Distributed Load Balancing Algorithm for Large Distributed System," Proc. 2nd Int'l Sym. Autonomous Decentralised Systems, pp. 266-273, Apr. 1995

[5] B. Losko, "Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence", Prentice Hall, New Jersey, 1992

[6] Lap-Sun Cheung, "A Fuzzy Approach to Load Balancing in a Distributed Object Computing

Network" Cluster Computing and the Grid, 2001. Proc. First IEEE/ACM Int'l Symposium on, pp. 694-699, 2001

[7] M. Mitzenmacher, "How Useful is Old Information," Proc. ACM Annual Sym. Principles of Distributed Computing, pp. 83-97, 1997