

퍼지-뉴럴 네트워크를 응용한 지능형 로드밸런싱 알고리즘 개발

추 교수*, 김 완 용**, 정 재 윤**, 김 학 배**

Development of Intelligent Load Balancing Algorithm in Application of Fuzzy-Neural Network

Gyo-soo Chu*, Wan-yong Kim**, Jae-yun Jung**, Hag-bae Kim** *Regular Members*

요 약

본 논문에서는 복잡하고 비선형적인 특성을 가진 웹 클러스터링 시스템의 모델링을 위해 퍼지-뉴럴 네트워크 구조를 응용하여 효율적인 최적의 부하분산 알고리즘 모델을 제안한다. 기본적으로 리눅스 환경의 웹 클러스터 시스템을 바탕으로 하였으며 이는 부하를 실제 서버(리얼서버)로 분배해주는 로드밸런서와 그 하단에 실제 부하를 처리하는 여러 대의 리얼서버로 구성된다. 기존의 부하분산 방법에서는 각각의 리얼서버들에 대한 접속수 등 단편적인 판단정보를 기준으로 부하분산을 결정하였다. 즉, 리얼서버의 네트워크 처리량 중 입력에 관련된 정보만으로 부하분산을 결정하였으므로 실제 시스템 성능 향상에 적당하지 않다. 이에 대한 대안으로 리얼서버의 시스템 상태 정보로 네트워크 부하량의 입·출력 두 가지 측면 모두를 기준으로 판단한다. 즉, 로드밸런서가 리얼서버에게 요청을 전달할 때의 리얼서버 상태(접속수) 및 그 요청에 대해 클라이언트에게 응답을 보낼 때의 리얼서버 상태 등을 종합적으로 고려한다. 또한 그에 따른 시스템의 CPU 상태로만 한정하여 부하분산을 결정하지 않고 각 리얼서버의 CPU 및 메모리 상태를 모두 고려하여 보다 효과적인 부하분산 스케줄링 방법을 제안한다. 본 연구에서 제안된 방법이 기존의 방법과 비교하여 좀 더 향상된 최적화 모델을 구축하며 보다 나은 지능형 로드밸런서 모델을 생성함을 시뮬레이션을 통하여 비교 분석하였다.

Key Words : 클러스터 웹 서버, 퍼지-뉴럴, 리얼서버, ANFIS scheduler, Fuzzy-neural analyzer

ABSTRACT

This paper suggests a method to effectively apply an application model of fuzzy-neural network to the optimal load distribution algorithm, considering the complication and non-linearity of the web server environment. We use the clustering web server in the linux system and it consists of a load balancer that distributes the network loads and some of real servers that processes the load and responses to the client. The previous works considered only with the scrappy decision information such as the connections. That is, since the distribution algorithm depends on the input of the whole network throughput, it was proved inefficient in terms of performance improvement of the web server. With the proposed algorithm, it monitors the whole states of both

* 연세대학교 전기전자공학과 디지털정보처리 연구실 (professor2000@yonsei.ac.kr),

** 연세대학교 전기전자공학과 디지털정보처리 연구실 (chsyong@yonsei.ac.kr),

** 연세대학교 전기전자공학과 디지털정보처리 연구실 (jaeyun@yonsei.ac.kr),

** 연세대학교 전기전자공학과 디지털정보처리 연구실 (hbkim@yonsei.ac.kr)

논문번호 : KICS2004-08-174, 접수일자 : 2004년 8월 30일

network input and output. Then, it infers CPU and memory states of each real server and effectively distributes the requests of the clients. In this paper, the proposed model is compared with the previous method through simulations and we analysis the results to develop the optimal and intelligent load balancing model.

I. 서론

웹 서비스 사용자가 늘어가고 네트워크 처리 요구량이 늘어남에 따라 웹서버는 과부하 상태에 놓이게 되는 경우가 빈번하게 발생한다. 이런 상황에서는 서비스의 질은 물론이고 기본적인 웹서버의 기능마저 상실하게 될 수 있다. 따라서 서비스 기능 마비현상을 줄이고 서비스 질을 높이기 위해서 웹서버 클러스터링은 필수적이다. 이러한 웹서버 클러스터링 시스템의 성능을 높이기 위해서 부하를 효과적으로 분산하기 위한 로드밸런싱 기술이 중요하다^[1].

리눅스 클러스터링 웹서버 환경에서 로드밸런싱의 기본 구조는 부하를 분산하는 부하분산 서버(이하 로드밸런서)와 그 밑단에서 실제 클라이언트의 요청을 처리하는 다수의 실제 서버(이하 리얼서버)로 이루어진다. 각 리얼서버는 상태정보를 로드밸런서에 전달하고 로드밸런서는 전달된 상태정보를 바탕으로 부하분산을 수행하게 된다^{[2],[3]}. 그러나 기존의 부하분산 방법에서는 부하정보의 비선형성과 리얼서버 시스템의 자원정보를 네트워크 입력, 즉 클라이언트의 요청인 접속수에만 의존하였다. 그래서 리얼서버의 성능을 제대로 반영하지 못하여 클러스터링 웹서버의 성능 향상에 비효율적이었다.

본 논문에서는 이러한 부하정보의 비선형성을 고려하여 부하분산에 효과적으로 적용시키기 위한 방법으로 퍼지-뉴럴을 응용한 새로운 모델을 제시하며 그것을 통한 리얼 서버의 시스템 자원정보 모델링 및 스케줄링 알고리즘을 제안한다. 또한 CPU의 상태로만 한정하여 부하분산을 결정하지 않고 각 리얼서버의 CPU 및 메모리 상태에 대한 두가지 측면을 모두 고려하여 보다 효과적인 부하분산을 수행하고자 한다. 즉 본 연구에서 제안하는 모델에서는 네트워크 입력뿐만 아니라 클라이언트의 응답인 네트워크 출력도 함께 고려하여 각각의 리얼서버의 CPU와 메모리의 상태를 추론한다. 리얼서버 시스템의 상태에 대해 CPU 및 메모리 상태를 다 고려하는 이유는 기존 연구처럼 CPU만을 고려했을 때 각각의 리얼서버의 CPU의 상태가 같을 경우에 대한 올바른 정책이 세워지지 않을 수 있기 때문이다.

본 논문의 구성은 다음과 같다. 2장에서는 제안

된 모델에서 퍼지 추론시 기본이 되는 Yamakawa에 의해 제안된 퍼지-뉴럴 네트워크를 설명하고 3장에서는 퍼지-뉴럴 네트워크 기반의 부하분산 알고리즘 모델을 제안한다. 4장에서는 모델링에 대한 시뮬레이션 결과를 설명하고 5장은 결론 및 향후 연구 과제에 대한 설명을 한다.

II. Yamakawa 퍼지-뉴럴 네트워크

Yamakawa에 의해 제안된 퍼지-뉴럴 네트워크(Yamakawa's FNNs)는 퍼지 추론과 학습이 결합된 형태를 가진다. Yamakawa's FNNs의 구조는 그림 1과 같다^[4].

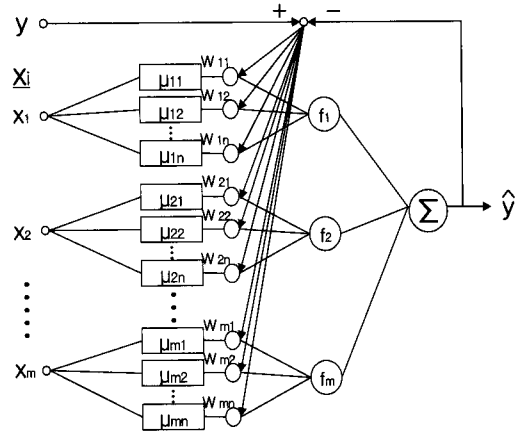


그림 1. Yamakawa's FNNs

$$\hat{y} = f_1(x_1) + f_2(x_2) + \dots + f_m(x_m) = \sum_{i=1}^m f_i(x_i) \quad (m = \text{퍼지-뉴런의 수})$$

Yamakawa's FNNs에서는 규칙의 형태가 식(1)과 같은 보수적 소속 함수를 가지는 간략추론법(식(2))이 퍼지 추론부에 사용되었다. 또한 신경망의 학습은 오류역전파알고리즘(Error Back-propagation Algorithm)에 의해 수행되게 된다. Yamakawa's FNNs는 기존의 신경망에 비해 학습속도가 빠르고 항상 수렴한다는 장점을 가지고 있다^[5].

$$R \text{ if } x \text{ is } A, \text{ then } y = w \quad (1)$$

$$f_i(x_i) = \frac{\sum_{j=1}^n \mu_{ij}(x_i)w_{ij}}{\sum_{j=1}^n \mu_{ij}(x_i)} \quad (2)$$

III. 제안하는 지능형 로드밸런서 구조

그림 2은 본 논문에서 제안하는 퍼지-뉴럴 모델링을 응용한 부하분산을 위한 지능형 로드밸런서 및 리얼 서버의 구조를 나타낸다. 전체적인 구조 및 내용은 다음과 같다. 서버 상태 리포터(Server Status Reporter)는 리얼서버의 네트워크 상태를 모니터링 하여 그 상태 정보를 수집, 저장, 보고하는 역할을 한다. 퍼지-뉴럴 분석기(Fuzzy-Neural Analyzer)는 서버 상태 리포터에 의해서 보고된 정보(네트워크 입·출력)를 이용하여 리얼 서버의 시스템 자원 상태인 CPU 부하 평균(CPU Load Average) 및 메모리 사용량(Memory Usage)을 추론한다. 지능형 로드밸런서에서 부하분산을 담당하는 ANFIS 스케줄러(ANFIS Scheduler)는 퍼지-뉴럴 분석기에서 추론된 시스템 자원을 바탕으로 클라이언트가 요청한 서비스를 할당할 가장 적절한 리얼 서버를 선택한다.

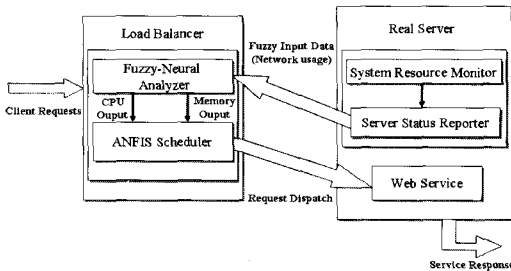


그림 2. 지능형 로드밸런서 부하분산 구조 제안

표 1은 기존의 부하분산 연구와 제안하는 모델의 차이점을 보여준다.

표 1. 기존 부하분산과 제안하는 모델 비교

구분	기존 부하분산 방식	제안하는 모델
리얼서버 정보	접속수 (네트워크 입력)	접속수, 요청에 대한 응답 (네트워크 입·출력)
추론하는 정보	CPU 부하 평균	CPU 부하 평균 및 메모리 사용량
부하분산 스케줄	접속수에 따른 스케줄링	추론된 정보에 따른 ANFIS 스케줄링

3.1 퍼지-뉴럴 분석기 모델

기존의 Yamakawa's FNNs에서는 한 번의 단계만 거쳐 입력 데이터에 대한 학습을 진행하게 된다. 이 경우 학습 데이터에서 비슷한 패턴이 많을 경우 부분적인 최적화가 이루어 질 수도 있다. 이것은 네트워크가 학습의 결과 가중치를 조절하는 행위가 더 이상 네트워크의 성능을 향상시키지 못할 경우 발생한다^[6]. 이러한 부분 최적화를 방지하기 위하여 본 논문에서 제안한 알고리즘은 첫 번째 단계에서의 출력 데이터 값을 이용하여 한 번 더 학습을 진행한다. 이 경우 두 번의 학습을 통하여 출력값을 판단함으로써 한 단계의 학습을 거치는 것보다 더 좋은 성능을 보인다. 각 단계에서의 학습은 모멘텀 오류역전파 알고리즘^[7]을 이용하며 제안된 모델은 그림 3과 같다.

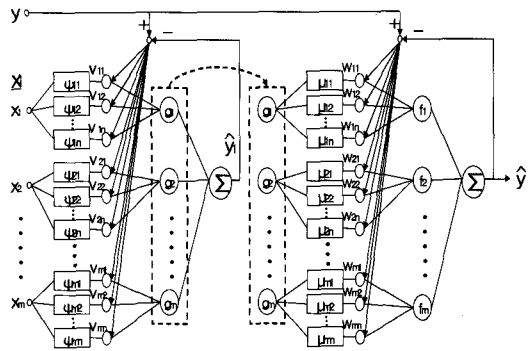


그림 3. 퍼지-뉴럴 네트워크를 응용한 학습 모델

입력 데이터 x_i 에 대하여 첫 번째 학습 단계에서의 퍼지-뉴런의 출력 $g_i(x_i)$ 및 출력 \hat{y}_1 은 각각 식 (3), 식(4)와 같다. v_{ij} 는 연결강도, ϕ_{ij} 는 소속 함수를 나타낸다.

$$g_i(x_i) = \frac{\sum_{j=1}^n \phi_{ij}(x_i)v_{ij}}{\sum_{j=1}^n \phi_{ij}(x_i)} \quad (3)$$

$$\hat{y}_1 = g_1(x_1) + g_2(x_2) + \dots + g_m(x_m) = \sum_{j=1}^m g_j(x_j) \quad (4)$$

(m = 퍼지-뉴런의 수)

두 번째 학습 단계에서는 첫 번째 학습 단계의 퍼지-뉴런 출력 데이터를 입력 데이터로 사용하여 학습을 진행한다. w_{ij} 는 연결강도, μ_{ij} 는 소속 함수를 나타내고 퍼지-뉴런의 출력은 $f_i(g_i(x_i))$ 이다.

$$f_i(g_i(x_i)) = \frac{\sum_{j=1}^n \mu_{ij}(g_i(x_i))w_{ij}}{\sum_{j=1}^n \mu_{ij}(g_i(x_i))}$$

모델의 최종 출력은 식(5)와 같이 두 번째 학습 단계의 퍼지-뉴런 출력의 합으로 표현된다.

$$\begin{aligned} \hat{y} &= f_1(g_1(x_1)) + f_2(g_2(x_2)) + \dots + f_m(g_m(x_m)) \\ &= \sum_{i=1}^m f_i(g_i(x_i)) \end{aligned} \quad (5)$$

퍼지-뉴런 분석기는 제안된 학습 모델을 이용하여 일정기간의 학습을 진행하며 학습정보를 결정된 후 이를 이용하여 각 리얼서버에서 전달된 네트워크 상태정보를 바탕으로 리얼서버의 시스템 자원을 추론한다⁸⁾.

3.2 ANFIS 스케줄링 알고리즘

로드밸런서 단에서 실제 부하분산을 담당하는 ANFIS 스케줄러는 퍼지-뉴런 분석기에서 추론된 CPU 및 메모리 사용 상태를 입력값으로 하여 부하 분산 정책을 결정한다. 기본적인 ANFIS 스케줄러 모델 형식은 식(6)과 같다.

$$\begin{aligned} R_i : & \text{IF } x \text{ is } A_i \text{ AND } y \text{ is } B_i \\ \text{THEN } & z = f_i = px + qy + r_i \end{aligned} \quad (6)$$

x , y 는 입력 값으로 CPU 부하 평균 및 메모리 사용량을 나타낸다. z 는 ANFIS 추론 출력으로 리얼서버를 배정하게 된다.

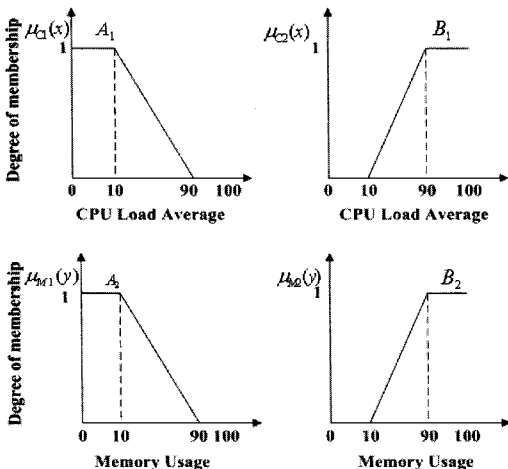


그림 4. 입력 값의 소속 함수

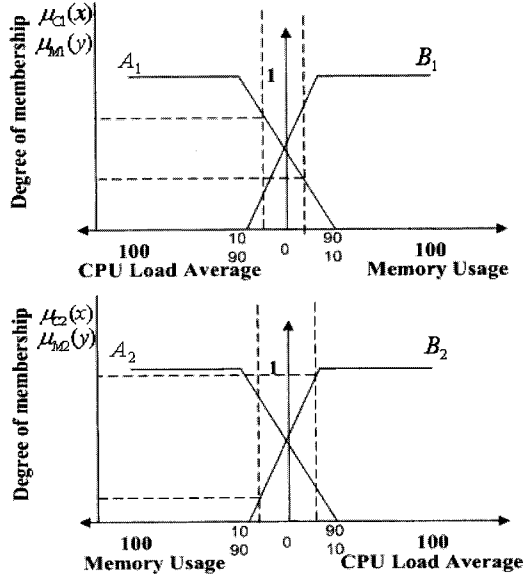


그림 5. 퍼지 추론 시스템

각 입력의 소속 함수는 그림 4와 같으며 그래프의 꺾이는 지점은 테스트 환경의 실험에 의해 얻어진 실험적 데이터 값이다. CPU나 메모리가 10미만 일 때는 부하를 처리하는데 전혀 문제가 없음을 나타내고 90이상일 때는 더 이상 부하를 처리하지 못함을 나타낸다. ANFIS 스케줄러의 퍼지 추론 시스템은 그림 5와 같이 CPU와 메모리의 사용 상태를 비교하여 리얼서버를 선택하게 된다. ANFIS 추론 시스템에서는 입력값인 CPU와 메모리 상태값이 주어지면 입력에 대한 소속 함수 값 $\mu_{C1}(x)$, $\mu_{C2}(x)$, $\mu_{M1}(y)$, $\mu_{M2}(y)$ 이 결정된다. 이 값으로 적합도 w_i 와 가중치 \overline{w}_i 의 값이 식(7), 식(8)에 의해 계산되어지고 식(9)에 의해 출력값인 리얼서버를 결정하게 된다.

$$w_i = \mu_{C_i}(x) \times \mu_{M_i}(y) \quad i=1, 2 \quad (7)$$

$$\overline{w}_i = \frac{w_i}{w_1 + w_2} \quad i=1, 2 \quad (8)$$

$$z_i^* = \sum_{j=1}^2 \overline{w}_j f_j = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \quad i=1, 2 \quad (9)$$

그림 6은 ANFIS 스케줄러가 부하를 분산할 리얼서버를 추론하는 구조를 보여준다.

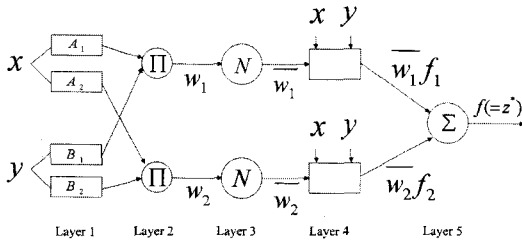


그림 6. ANFIS 스케줄러 추론 시스템

IV. 시뮬레이션 및 결과고찰

실제 필드에서 운영 중인 L4 스위치와 8대의 리얼서버로 이루어진 리눅스 클러스터링 웹서버 중 Application을 담당하는 3대의 리얼서버에서 수집된 1년간의 데이터 일부를 이용하여 시뮬레이션을 수행하였다. 각 리얼서버에는 로드밸런서로부터 요청된 부하를 받는 네트워크 카드가 두 개씩 존재한다. 따라서 메모리의 사용 상태 추론 값은 이 두 네트워크 카드의 사용량에 영향을 받는다. 수집된 데이터의 내용은 네트워크의 입·출력, CPU 부하 평균(Load Average), 메모리 및 디스크 입·출력이다. 그 중 연속적인 시간에 따른 600개의 데이터만을 사용하였다.

또한 모델링을 위한 값으로 학습 데이터 성능지수(PI)와 테스트 데이터 성능지수(E_PI)를 구하기 위하여 사용되어지는 데이터를 두 그룹으로 분할할 때 한쪽에는 입력변수의 값이 적은 데이터가 많고 다른 편에는 큰 데이터가 많은 등의 경우가 발생치 않도록 주의하기 위하여 학습 및 테스트 데이터를 짝수와 홀수로 구분하여 구성하였다⁹⁾.

평가를 위한 출력오차 E는 식(10)과 같다. N은 학습에 사용되는 입력 데이터 개수이며 y_p 및 \hat{y}_p 는 각각 출력 교사 신호 및 퍼지-뉴럴 모델의 최종 출력을 나타낸다.

$$E = \frac{1}{N} \sum_{p=1}^N (y_p - \hat{y}_p)^2 \quad (10)$$

4.1 CPU 부하 평균 모델링 및 해석

표 2는 리얼서버의 CPU 부하 평균을 모델링하였을 경우 기존의 모델링과 본 논문의 퍼지-뉴럴 네트워크를 응용한 제안된 모델링과의 동정오차에서 그 성능을 비교하여 보여준다. 제안된 모델은 기존 모델의 학습 성능지수보다 우수할 뿐만 아니라 테스트 데이터를 고려한 테스트 성능 즉 예측 성능

면에서도 우수한 특성을 보인다. 즉 제안된 모델이 기존 모델보다 근사화 및 일반화 능력 모두 우수한 것으로 나타난다. 또한 기존의 모델링 방법과 제안된 모델링 방법에서 네트워크 입력만을 고려한 경우보다는 입·출력을 모두 고려하였을 때 학습 성능지수가 우수한 특성을 보인다. 이 경우 기존의 모델에서는 일반화, 즉 예측능력은 다소 떨어지나 제안된 모델에서는 예측 성능 면에서도 우수한 특성을 보이므로 실 시스템 적용에서는 제안된 모델이 더 우수한 성능을 나타냄을 알 수 있다.

표 2. CPU 부하평균에 대한 모델링의 성능지수

Model	입력만 고려		입·출력을 고려	
	PI	E_PI	PI	E_PI
Yamakawa's FNNs Model	10.281203	25.944948	7.973634	27.742937
Proposed Model	9.447036	25.933981	6.559269	22.554228

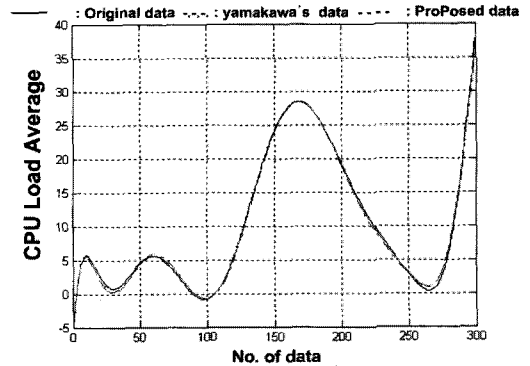


그림 7. 학습 데이터 성능비교

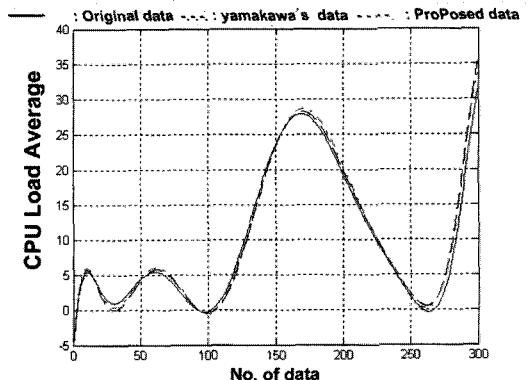


그림 8. 테스트 데이터 성능비교

퍼지동정 모델의 기존 모델과 제안된 모델의 출력과 실 데이터와의 비교는 그림 7, 8, 9와 같으며 실제 데이터 값을 잘 추종함을 알 수 있다.

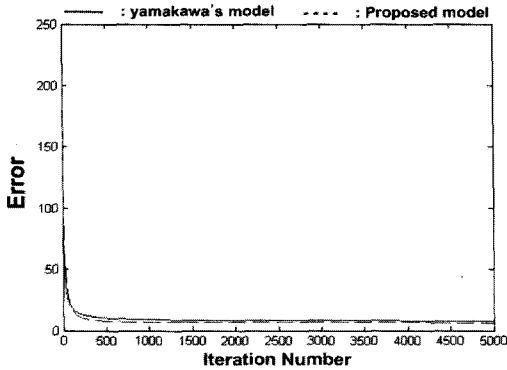


그림 9. CPU 부하 평균 출력오차

4.2 메모리 사용량 모델링 및 해석

표3은 메모리 사용량 모델링에 있어서 CPU 부하평균의 경우와 마찬가지로 기존의 모델링보다 본 논문에서 제안된 모델링이 동정오차 성능에서 우수한 특성을 나타냄을 보인다. 또한 입력만을 고려하였을 때보다 입·출력을 모두 고려하였을 경우 기존모델보다 근사화 능력이 월등이 우수함을 알 수 있다. 하지만 일반화, 즉 예측능력에서는 제안된 모델이 다소 떨어짐을 볼 수 있다. 이는 메모리 사용량은 네트워크 입·출력이 그리 많지 않더라도 캐싱, 기본 응용 프로그램 등으로 어느 정도의 자원이 항상 소모되는 상태에 있기 때문이다. 또한 각각의 리얼서버에는 두 개의 네트워크 카드가 존재하므로 각각의 요청에 대해서도 시스템 메모리의 사용량에 영향을 준다. 따라서 메모리의 추론 값은 네트워크 카드가 한 개일 때보다 정확한 추론이 어렵게 되어 CPU 부하 평균만큼 네트워크 입·출력에 대해 정확하게 측정되지 못한다.

퍼지동정 모델의 기존 모델과 제안된 모델의 출력과 실 데이터와의 비교는 그림 10, 11, 12와 같다.

표 3. 메모리 사용량에 대한 성능지수 결과

Model	입력만 고려		입·출력을 고려	
	PI	E_PI	PI	E_PI
Yamakawa's FNNs Model	34.5636	41.0543	28.0392	39.1521
Proposed Model	28.1251	35.5571	22.9497	40.6729

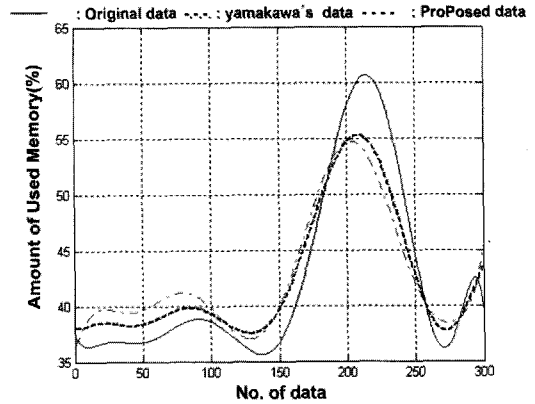


그림 10. 학습 데이터 성능비교

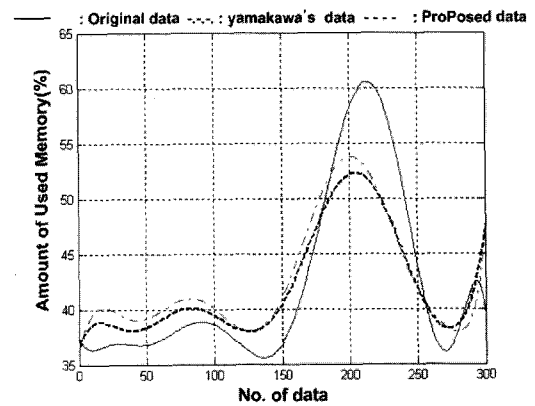


그림 11. 테스트 데이터 성능비교

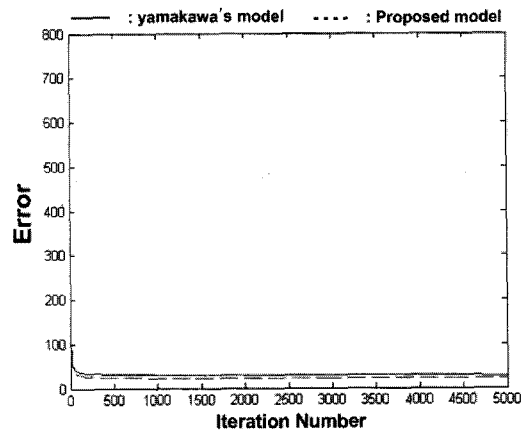


그림 12. 메모리 사용량 출력에러

4.3 부하분산 시뮬레이션 결과

접속수 등 단편적인 정보를 이용하여 부하분산을 수행하는 경우 및 리얼서버의 네트워크 입·출력

정보에 따라 시스템 자원을 추론하여 ANFIS 스케줄러에서 부하분산을 수행하는 경우의 각각의 리얼 서버 CPU 상태를 시뮬레이션을 통하여 비교분석해보았다. 시뮬레이션에서는 퍼지 추론에서 사용하였던 테스트 데이터를 이용하였다.

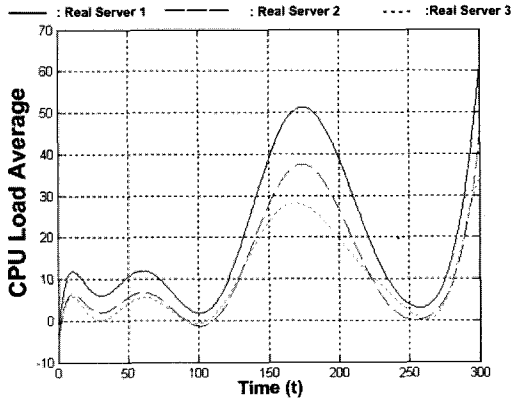


그림 13. 접속수에 의한 CPU 부하 평균

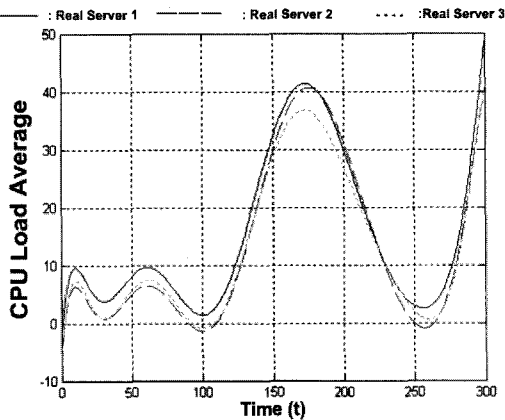


그림 14. 추론된 정보에 의한 CPU 부하 평균

접속수 등 단편적인 정보를 기준으로 부하분산을 하였을 경우 각 리얼서버의 CPU 부하 평균은 그림 13과 같다. 결과를 살펴보면 첫번째 리얼서버의 CPU 사용이 다른 리얼서버들에 비해 많음을 볼 수 있다. 이는 클라이언트의 요청에 대해 로드밸런서가 리얼서버에게 그 요청에 대해 스케줄링을 함에 있어서 부하분산을 효율적으로 하지 못함을 보인다. 즉 리얼서버의 시스템 사용 효율 측면에서 다소 떨어지게 된다.

이와는 달리 리얼서버의 입·출력 정보를 바탕으로 리얼서버의 시스템 자원 상태를 추론하고 CPU 와 메모리 두 가지 측면을 다 고려하여 부하분산을

수행하였을 경우 각 리얼서버의 CPU 부하 평균은 그림 14와 같이 나타난다. 시뮬레이션 결과는 각 리얼서버의 CPU 사용률이 시간에 따라 거의 비슷함을 보여준다. 이는 로드밸런서가 클라이언트의 요청에 대해 리얼서버에게 스케줄링을 함에 있어서 부하분산을 효율적으로 함으로써 부하분산에 있어 성능이 우수함을 보인다. 즉 접속수 등의 단편적인 정보를 기준으로 부하분산을 하였을 경우보다 리얼서버의 시스템 사용 효율 측면에서 좀 더 나은 결과를 보여준다고 할 수 있다.

V. 결론

본 논문에서는 비선형적인 요소가 많고 언어적 규칙을 찾기 어려운 웹 클러스터링 시스템을 모델링하기 위하여 퍼지 추론에 의한 시스템과 신경회로망이 가지는 장점을 결합하여 보다 나은 시스템 성능 특성을 가지는 퍼지-뉴럴 네트워크를 응용한 새로운 모델을 제시하였다. 새로운 모델에서는 리얼서버의 네트워크 입·출력 정보를 이용하여 시스템 자원 상태를 추론하고 이 정보를 이용하여 부하분산을 수행하게 된다. 이 경우 리얼서버의 성능을 제대로 반영하기 위해서는 리얼서버의 네트워크 입력을 기준으로 하기보다는 네트워크의 입·출력 두 가지 측면을 다 고려했을 때의 성능지수가 크게 향상되었고 실제 데이터 값을 잘 추종함을 볼 수 있었다. 따라서 리얼서버의 네트워크 입력과 출력에 대한 정보를 가지고 시스템 자원을 추론하는 모델이 더 타당함을 알 수 있었고 본 논문에서 제안한 두 번의 학습 과정을 거치는 모델이 시스템 자원 상태를 추론함에 있어 더 좋은 결과를 보이는 것을 확인하였다. 또한 접속수 등 단편적인 판단정보만으로 부하분산을 수행하는 것보다 제안한 퍼지 추론 시스템에서 추론된 시스템 자원정보를 이용하여 스케줄링을 수행하는 것이 각 리얼서버들의 평균적인 CPU 사용률이 비슷하게 되어 시스템 사용 효율 측면에서 더 좋은 성능을 보이는 것을 시뮬레이션을 통하여 확인하였다.

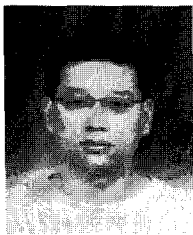
추후 연구로는 이러한 지능형 로드밸런싱 알고리즘을 부하분산 프로그램화하여 기존의 부하 분산 알고리즘과 함께 리눅스에 모듈형태로 실제 탑재함으로써 기존의 방법과 제안된 모델을 비교분석하여 부하분산에 있어서 지능형 로드밸런서의 부하분산 알고리즘의 효용성에 대해 입증하려 한다.

참 고 문 헌

- [1] Bryhni H, Klovning E, Kure O. A, "comparison of load balancing techniques for scalable web servers.", *IEEE Network*, pp.58-64, 2000.
- [2] "http://www.linuxvirtualserver.org/w-hatis.htm", *Linux Virtual Server Project*, 1998.
- [3] 배원열, "퍼지 모델 기반 최적 로드밸런싱 알고리즘 개발", 2002.12
- [4] 노석범, 오성권, "퍼지-뉴럴네트워크에 의한 비선형 공정 모델의 최적화," *대한전자공학회 학술회의 논문집, 1권0호*, pp.873-873, 1994.
- [5] 박호성, 오성권, 윤양웅, "퍼지 뉴럴 네트워크 구조로의 새로운 모델링 연구", *제어·자동화·시스템공학 논문지 제7권 제8호*, 2001.8.
- [6] 오성권, 노석범, 남궁문, "퍼지-뉴럴네트워크 구조에 의한 비선형 공정시스템의 지능형 모델링", *한국퍼지 및 지능시스템학회 논문집 제5권 제4호*, pp.41-55, 1995.
- [7] 함원일행, "뉴로 퍼지 카오스: 신세대 아날로그컴퓨팅 입문", *대광서림*, 1994.
- [8] 오성권, "프로그래밍에 의한 컴퓨터지능" *내하출판사*
- [9] 오성권, "퍼지모델 및 제어이론과 프로그램", *기다리출판사*, 1993.3

추 교 수(Gyo-soo Chu)

정회원

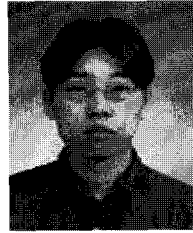


2002년 8월 원광대학교 전기 전자공학과 졸업
 2003년 3월~현재 연세대학교 전기전자공학과 석사과정
 <관심분야> Clustering web server, Load balancing, 지능형 홈네트워크, 퍼지이론

및 신경회로망 응용

김 완 용(Wan-yong Kim)

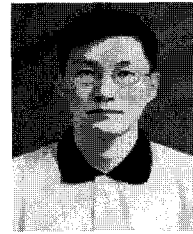
정회원



2003년 8월 연세대학교 전기 전자공학과 졸업
 2003년 9월~현재 연세대학교 전기전자공학과 석사과정
 <관심분야> Clustering web server, Load balancing, RTOS 및 Embedded Linux

정 재 윤(Jae-yun Jung)

정회원



2004년 2월 연세대학교 전기전자공학과 졸업
 2004년 3월~현재 연세대학교 전기전자공학과 석사과정
 <관심분야> Clustering web server, Load balancing, Intelligent Home Networks

김 학 배(Hag-bae Kim)

정회원



1988년 서울대학교 전자공학과 졸업
 1990년 미국 미시간대학교 전기 및 컴퓨터공학과 석사
 1994년 미국 미시간대학교 전기 및 컴퓨터공학과 박사
 1996년~현재 연세대학교 전기

전자공학과 부교수

<관심분야> 실시간 시스템, 인터넷 웹서버 기술, 디지털 시스템 고장포용 및 신뢰도 평가분야