

# 차별화 서비스를 위한 퍼지기반 성능분리

박범주\*, 강명구\*\*, 박기진\*\*, 김성수\*\*\*

\*삼성전자 첨단기술연구소

\*\*아주대학교 산업정보시스템공학부

\*\*\*아주대학교 정보통신전문대학원

\*e-mail : bumjoo@samsung.com

## A Fuzzy Based Performance Isolation for Differentiated Service

Bumjoo Park\*, Myeongkoo Kang\*\*, Kiejin Park\*\*, Sungsoo Kim\*\*\*

\*Advanced Technology Training Institute, Samsung Electronics

\*\*Division of Industrial & Information Systems Engineering, Ajou University

\*\*\*Graduate School of Information and Communication, Ajou University

### 요 약

본 논문에서는 SLA(Service Level Agreement)와 같이 차별화 서비스를 지원하는 웹서버 시스템의 가동성 척도를 향상시키기 위해 기존의 동적 성능 분리 기법에 퍼지 기법을 접목하였다. 특히, 클러스터 기반 웹서버 시스템의 부하량에 대한 판단기준 혹은 사용자 요청률 및 동적요청 비율 변화시에 발생하는 애매모호한 상황을 효과적으로 반영하기 위해, 퍼지제어 기법에 기초한 부하분배 메커니즘을 제안하였다. 이를 통해, 기존의 퍼지 기법을 활용하지 않은 성능분리 기법과 퍼지기법을 활용한 경우에 대해 응답시간(95-percentile of response time) 성능 비교 평가를 통해 퍼지기반의 성능분리 기법이 차별화 서비스 시스템의 성능을 더욱 강건하고 효율적으로 향상시킬 수 있다는 점을 검증하였다.

### 1. 서론

클러스터 기반 웹 서버에서 고품질 차별화 서비스를 제공하기 위해서는 계층별 사용자 요청 처리 성능을 높이는 것이 중요하다. 이를 위해, 먼저 웹 서버를 구성하는 개별 컴퓨팅 노드의 부하(Load)를 정확히 파악해야 한다. 일반적으로 컴퓨팅 노드의 부하는 CPU 사용률, 메모리 사용량, 혹은 대기 작업의 수 등으로 판단하고 있으며, 노드의 부하가 매우 높은 상황(Heavily-loaded) 혹은 부하가 작게 걸린 상황(Lightly-loaded)에 대한 판단 기준 등이 응용 분야에 따라 매우 복잡할 수 있다. 컴퓨팅 노드 사이의 부하 조절을 위해 최근 많이 사용되고 있는 Layer-7 스위치는 컴퓨팅 노드를 결정하기 전에 HTTP 요청을 분석할 수 있는 측면 때문에 사용자 요청 내용에 대하여 상세한 정보를 고려한 분배가 가능하다.

한편, 계층별 사용자 요청에 따라 그에 알맞은 특정 컴퓨팅 노드들을 할당할 수 있어야 한다. 컴퓨팅 노드 분할 기법인 성능 분리(Performance Isolation) 기법을 통해, 성능 분리된 노드마다 서로 다른 계층의 요청 처리를 담당하게 하고, 상위 계층의 사용자 요청일수록 더 많은 컴퓨팅 노드를 할당해 줌으로써, 차별화 서비스를 보장할 수 있게 된다. 이때 부하분배기(Load Balancer)는 계층별 사용자의 요청을 처리할 컴퓨팅 노드를 결정하며, 웹 서버의 상태(요청률, 각 노드의 현재 부하)를 정확히 파악하고 있어야, 사용자 계층별 요청을 최적으로 분배할 수 있다. 따라서 부하분배기의 핵심적인 성능 요소는 웹 서버를 구성하는 각 노드의 부하 균형도 및 응답시간이라 볼 수 있다.

이 때, 사용자 요청률의 변화나 웹 서버 부하량 판단에 있어서 애매모호한 상황에 대한 대응이 요구된다[1]. 즉, 비록 노드에서 대기중인 사용자 요청수가 많아도, 동적 콘텐츠(Dynamic Contents) 보다 정적

“본 연구는 2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행되었음.”(KRF-2005-041-D00630).

컨텐츠(Static Contents)의 비중이 매우 높다면, 노드에 부하를 많이 주지는 못한다. 이와 같이, 컴퓨팅 노드의 부하량 판단에 있어서 애매모호한 상황을 반영하기 위해 본 논문에서는 사용자 계층별 요청률에 따라 웹 서버 컴퓨팅 노드들의 성능 분리를 동적으로 수행하는, 퍼지 이론을 적용한 웹 서버 성능 분할 기법을 도입하였다. 제안된 기법을 통해 응답시간 성능 분석 등의 시뮬레이션 평가를 통해 퍼지기법을 활용한 기법이, 이를 사용하지 않은 기법에 비해 우수한 성능을 보여주고 있음을 검증하였다.

## 2. 관련 연구

네트워크 단계의 QoS(Quality of Services)만으로는 인터넷 비즈니스 양단간 QoS 를 지원할 수 없는 문제점을 해결하기 위하여, 웹 서버 단계에서 클라이언트 요청의 우선 순위에 따라 차별화된 서비스를 제공할 수 있는 기법에 대한 연구가 수행된 바 있으며[2], Layer-4 부하 분배 방식은 TCP/IP 레벨에서 동작하며, HTTP 요구가 보내지기 전에 TCP/IP 연결 설정이 이루어지기 때문에 서비스 노드에 대한 선택이 요구된 내용에 의해 결정될 수 없다. 이에 비하여 Layer-7 부하 분배 알고리즘은 서비스 노드를 결정하기 전에 HTTP 요구를 분석할 수 있기 때문에 클라이언트 요구 내용에 대하여 상세한 정보를 고려한 분배가 가능하다[3].

기존에는 클러스터 웹 서버의 부하 조절(Load Balancing)을 위해서 Layer-4 기반 스위치가 사용되었으나, 최근 내용기반 분배가 가능한 Layer-7 스위치로 급속히 대체되고 있다.

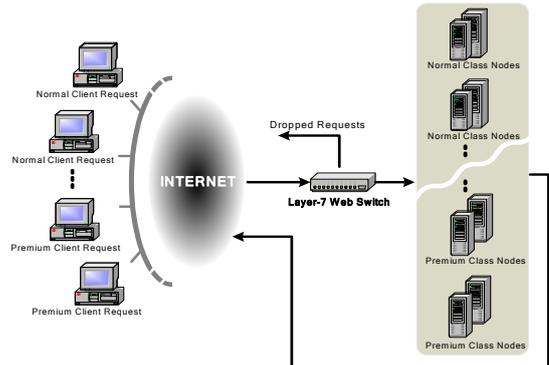
LARD(Locality-Aware Request Distribution)[4]는 웹 서버 클러스터에서 내용 기반 라우팅에 대한 초기의 연구로써, LARD 에서 부하 분배기는 각 웹 문서에 대하여 서비스 노드와 일대일 대응을 유지하며 주어진 문서에 대하여 첫 번째 요구가 도달하면 가장 적은 부하가 걸려 있는 서비스 노드에 우선적으로 할당한다. 한편, 동적 분할(Dynamic Partition) 기법은 사용자의 계층별 요청 수준 혹은 필요에 따라서 동적으로 서버 노드를 변경적으로 할당하며, DDS(Demand-driven Service Differentiation), Dynamic Partitioning 기법 등이 있다[5].

DDS 기법은 클라이언트 요청량에 따라 CPU 와 디스크 I/O 용량을 할당함으로써 차별화된 서비스를 제공하고, e-비즈니스와 관련된 웹 사이트 컨텐츠처럼 CPU 와 디스크 I/O 의 처리를 많이 요구하는 동적 요구가 많은 상황에 적합하다. 또한, Dynamic Partitioning 기법은 SLA 를 만족하는 서비스를 상위 계층 사용자에게 제공하기 위해 서버의 부하량에 따라 서버 노드를 동적으로 분할하는 기법이다. 즉, 상위 계층을 서비스하는 노드에 과부하가 걸렸을 경우, 이보다는 더 낮은 계층을 서비스하고 있는 서버를 추가적으로 이용하거나, 과부하 상황이 해소되면 다시 서버를 반납하는 것을 기본 개념으로 하고 있다. 이러한 동적 분할 개념은 부하 변화 따라 여러 사용자 계층에게 효율적이면서도 질 높은 서비스를 제공할 수 있지만, 정적 요청 비율이 증가할 경우 정적 분리 기법이 더 좋

은 시스템 성능을 제공하는 문제점도 있다.

## 3. 퍼지 기반 웹서버 성능 분리 기법

그림 1 은 본 논문에서 대상으로 하는 Layer-7 스위치 기반 One-way 웹 서버 구조이다. 두 계층의 사용자(NC: Normal Class User, PC: Premium Class User) 를 가정할 경우 서비스 노드는 일반 사용자의 요청을 처리하는 일반 노드(이하 NC), 고급 사용자를 서비스하는 상위 노드(이하 PC)로 나눈다.



(그림 1) Layer-7 스위치 기반 One-way 웹 서버 구조

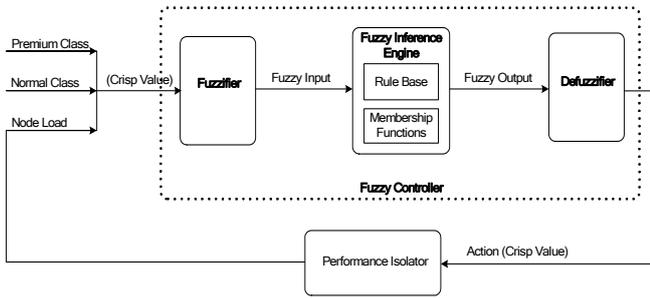
One-way 구조에서는 들어오는 패킷은 웹 스위치를 거쳐 웹 서버로 전달되는 반면에 나가는 패킷은 웹 스위치를 거치지 않고 직접적으로 사용자에게 전달되는 구조를 가지며, Two-way 구조는 클러스터 안의 각각의 서버는 유일한 하나의 IP 주소로 설정되어 들어오는 패킷과 나가는 패킷은 웹 스위치에 의해 재작성된다. 따라서 One-way 구조는 Two-way 구조에 비하여 스위치의 오버헤드를 대폭 줄일 수 있다.

웹 서버 구조에 설계에 적용된 가정은 다음과 같다.

- 계층별 사용자 요청은 일반 사용자 요청과 고급 사용자 요청으로 구분되며, 각각의 요청은 해당되는 컴퓨팅 노드에서만 처리된다.
- 노드의 부하, 계층별 사용자 요청률 변화에 따라 해당 계층을 서비스하는 컴퓨팅 노드의 임대(Borrowing) 및 대여(Lending) 동작이 발생한다. 예를 들어 고급 사용자의 요청이 증가하여, PC의 성능이 떨어질 경우, PC는 일반 사용자를 대상으로 서비스하는 노드(NC)를 임대하여 사용한다. 이와 같은 동적인 임대와 대여 동작을 통해, 계층별 서비스 노드간의 부하 균형 및 계층별 사용자 요청에 대한 차별화 서비스를 제공할 수 있다.

퍼지 기반 웹 서버 성능 분리 알고리즘을 구현하기 위해 설계하고자 하는 퍼지 제어기의 퍼지 변수(Fuzzy Variable)를 현재 재구성(임대 및 대여) 가능한 웹 서버의 부하량 및 사용자 계층별 요청률(Arrival Rate)로 설정하였다. 퍼지 변수를 도입한 퍼지 성능 분할 제어기의 구조는 그림 2 와 같다. 계층별 도착률과 서버 노드의 부하량은 Fuzzifier 에 의해 퍼지 변수 입력값

으로 변환된 후, 퍼지 추론 엔진에서는 퍼지 규칙 및 멤버십 함수를 이용하여, 최종 퍼지 출력값을 산출한다. 퍼지 출력값은 여전히 퍼지한 성질을 가지고 있기 때문에 이를 제거하기 위해 Defuzzifier 에서 처리된 후, 최종적인 제어 동작(즉 서버 노드의 임대 및 대여를 위한 Crisp 한 값)을 성능 분리기(Performance Isolator: e.g. 부하분배기)에 지시한다. 정리하면, 퍼지 성능 분할 제어기의 입력과 출력은 Crisp 한 이진 값을 가지며, 퍼지 제어기 내부에서만 애매모호한 성질을 가진 퍼지 변수로 처리된다. 이와 같은 제어 동작은 요청률 및 노드 부하량 변동에 따라 계속해서 반복하여, 동적인 성능 분할 기능을 수행한다.



(그림 2) 퍼지 성능 분할 제어기의 블록도

퍼지 제어기로 입력된 요청률과 부하량(Crisp)은 멤버십 함수에 의해 해당 집합에 속하는 정도인 퍼지 입력값으로 변환되어야 하며, 이때 Crisp 입력값이 여러 멤버십 함수 구간의 값을 동시에 가질 경우, 최대 퍼지 값을 갖는 멤버십 함수값(Max-Min 알고리즘)을 결과로 출력한다. 이와 같은 방식에 의거하여, 모든 퍼지 입력 변수의 애매모호함이 결정되면, 표 1 에 나타난 규칙에 의해 추론 엔진은 최종 퍼지 추론 결과를 출력한다.

표 1 퍼지 추론 규칙

요청률 부하량	Low	Medium	High
Very Low	Negative Large	Negative Moderate	Negative Small
Low	Negative Moderate	Negative Small	Approximately Zero
Medium	Negative Small	Approximately Zero	Positive Small
High	Approximately Zero	Positive Small	Positive Moderate
Very High	Positive Small	Positive Moderate	Positive Large

이와 같이 퍼지입력 변수의 단계별 조합에 따라 모두 15 개의 추론 규칙이 생성되며, 이들의 집합을 규칙 베이스라고 한다. 추론엔진은 규칙베이스와 앞 절에서 설명한 멤버십함수를 결합하여 최종 단일 퍼지 결론(MISO: Multi Input Single Output)을 유도해 낼 수 있다. 이 과정을 나타내면 다음과 같다.

1) Input: x is X\* and y is Y\*

2) 규칙

Rule 1: If x is X<sub>1</sub> And y is Y<sub>1</sub> Then z is C<sub>1</sub>

Rule 2: If x is X<sub>2</sub> And y is Y<sub>2</sub> Then z is C<sub>2</sub>

Rule n: If x is X<sub>n</sub> And y is Y<sub>n</sub> Then z is C<sub>n</sub>

3) Conclusion: z is C\*

퍼지 출력값은 비퍼지화 과정을 거쳐 최종 제어값(Crisp)으로 변환하는데 데, 이 경우 COA(Center of Area) 기법을 적용하여 출력값을 계산하였다(식 1). 여기서, W<sub>i</sub> 는 i 번째 추론 규칙의 퍼지값이고, B<sub>i</sub> 는 i 번째 추론 규칙의 중앙 값이다. 이렇게 함으로써 도착률의 변화에 신속하게 대처할 수 있게 된다.

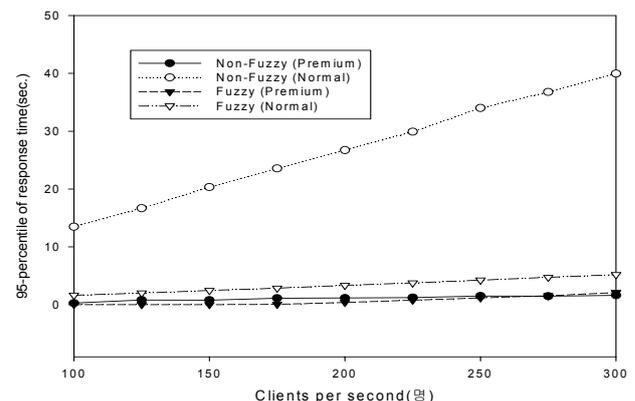
$$P_i^{COA} = \frac{\sum_{i=1}^n W_i * B_i}{\sum_{i=1}^n W_i} \quad (1)$$

퍼지 제어기의 규칙에 따른 추론 결과를 고려하여, 성능 분리기에 전달할 최종 결과를 결정하게 되는데, 우선권은 항상 고급 사용자에게 있다. 고급 사용자가 노드 임대를 요청할 경우 일반사용자 계층을 서비스하는 서버 노드 집합으로부터 노드를 임대 받게 된다.

#### 4. 성능 평가

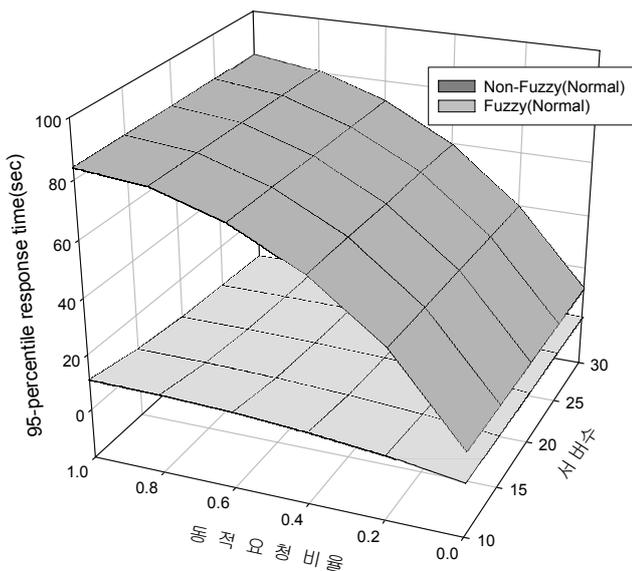
퍼지기반 웹 서버 클러스터 시스템 성능 분리 기법의 성능 평가를 위해, 응답시간(95-percentile of response time)에 대한 비퍼지 기법과의 비교분석을 진행하였다. 이 때, 퍼지기법을 사용하지 않은 성능 분리 방법은 특정 계층의 클라이언트 요청률이 높아져 과부하가 걸릴 경우, 과부하가 걸린 서버보다 한 단계 낮은 우선 순위의 계층에서 서버를 가져가고, 클라이언트의 요청이 적어지면 다시 한 단계 낮은 우선 순위의 계층으로 반납하는 방식을 적용하였다.

95-percentile of response time 은 들어온 요청 중 95% 이상은 서비스 제공자와 고객간의 계약된 시간 안에 응답시간을 만족한다는 것을 의미한다. 웹 서버는 각 클라이언트 계층으로부터 정적 요청과 동적 요청을 받아들이며, 클라이언트 계층을 premium · normal 2 개 수준으로 나누어 성능을 분석했다.



(그림 3) 사용자 요청률 변화에 따른 응답시간 비교

그림 3에서는 차별화 서비스를 위한 2개 계층을 대상으로, 동적 분할에 따른 퍼지 및 비퍼지 기법간의 응답 시간을 사용자의 요청률의 변동에 따라 표시하였다. 퍼지 기반 성능 분리 기법과 비퍼지기법 모두 Premium 계층이 Normal 계층에 대해 응답시간 성능이 우수하게 나타나고 있다. 한편, 퍼지 기반 기법이 비퍼지기법에 비해 Normal 계층에서 월등한 응답시간 성능을 보이고 있다. 이는 퍼지기법이 사용자 요청률 및 서버 부하량 증가에 따라 서버 노드수를 보다 탄력적으로 조정할 결과 때문이라 판단된다. Premium 계층에서는 퍼지기법 기법이 비퍼지기법에 비해 사용자 요청률이 적은 구간에서는 약간 우수한 성능을 보이고 있으나, 요청률이 커질수록 응답시간 성능이 거의 동일한 수준을 보이고 있다. 결국, 퍼지기법이 Normal 계층 중심으로 비퍼지기법에 비해 우수한 성능을 보이고 있음을 알 수 있으며, Premium 계층에서의 성능 차별화는 퍼지 규칙 및 멤버십 함수의 변화를 통해 분석될 필요가 있다.



(그림 4) 서버수/동적요청 비율 변화에 따른 응답시간 비교(Normal 계층)

그림 4에서는 Normal 계층에 대해 서버수 및 동적요청 비율 변화에 따른 응답시간을 비교분석하였다. 기본적으로, 퍼지기법과 비퍼지 기법 모두에서 서버수가 증가할수록 응답시간이 감소하는 경향을 보이고 있으며, 동적요청 비율이 증가하면 응답시간이 증가하는 경향을 보이고 있다. 하지만, 퍼지기법의 경우 증가폭이 완만한 경향을 보이게 되므로 유의할만한 시스템 성능저하가 일어나지 않지만, 비퍼지기법의 경우 동적요청 비율 변화에 응답시간이 민감하게 반응하고 있음을 알 수 있다. 이러한 경우, 서버수 변화보다 동적 요청 비율이 급격히 증가할 때 시스템 가동성능 저하가 예상된다. 상기의 성능평가 실험을 통해, 퍼지기법 성능 분리

기법이 퍼지기법을 사용하지 않은 경우에 비하여 사용자 요청률 및 시스템 환경 변화에 따른 변수에 훨씬 강건하게 대응하고 있음을 확인할 수 있다.

## 5. 결론

본 논문에서는 퍼지 이론을 적용한 웹 서버 성능 분할 기법에 관하여 논하였다. 제안된 기법은 컴퓨팅 노드의 현재 부하량, 사용자 계층별 요청률을 퍼지 입력 변수로 하여, 애매모호한 노드의 정량적 부하를 정성적으로 표현할 수 있게 함으로써 이를 통해 계층별 요청률이 급격한 변화에 대응하여, 계층별 요청을 처리하는 담당 노드의 수를 동적으로 조절할 수 있게 하였다. 다양한 성능분석을 통해 제안된 퍼지 기반 성능 분리 방식의 웹 서버 응답시간이 차별화 서비스를 위해 상위 계층을 중심으로 비퍼지기법에 비해 개선되는 것을 확인할 수 있었다.

향후, 사용자 요청률 및 서버노드 부하량 특성을 보다 정밀하게 반영한 퍼지 멤버십 함수 구성을 통해 성능 개선 효과를 확대할 예정이다. Triangle 형태 이외의 다양한 함수 형태를 참조하여 멤버십 함수를 재구성하고자 한다. 또한, 95-percentile of response time 이외의 다양한 성능 파라미터에 대해서도 비퍼지기법과 비교분석 연구를 수행할 예정이다. 한편, 웹서버는 과부하 상황에서 우선순위가 높은 사용자의 요청이 거절되는 경우를 발생시킬 수 있다. 이러한 경우 사용자의 우선순위에 따라 상위 계층의 사용자 요청을 받아들이고 하위계층의 사용자 요청을 거절하는 승인제어(Admission Control)가 향상된 차별화 서비스를 위해 중요한 요소라 할 수 있다. 따라서, 퍼지기법을 적용한 승인제어 알고리즘을 제안하고 해당 알고리즘에 대한 퍼지 기법과 비퍼지기법의 성능 비교분석을 통해 차별화 서비스를 보다 체계적으로 지원하기 위한 연구를 수행할 예정이다.

## 참고문헌

- [1] A. Shaout, P. McAuliffe, "Job scheduling using fuzzy load balancing in distributed system," *Electronics Letter*, 34(20): pp. 1983-1985, 1998.
- [2] V. Kumar, T. Lakshman, and D. Stiliadis, "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet", *IEEE Communication Magazine*, Vol. 36, pp. 152-64, May 1998.
- [3] M. Aron, et al., "Scalable Content-aware Request Distribution in Cluster-based Network Servers", *Proceedings of the 2000 Annual Usenix Technical Conference*, 2000.
- [4] L. Cherkasova and M. Karlsson, "Scalable Web Server Cluster Design with Workload-aware Request Distribution Strategy WARD", *3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, pp. 212 -221, 2001.
- [5] M. Andreolini, et al., "A cluster-based web system providing differentiated and guaranteed services," *Cluster Computing*, 7(1): pp. 7-19, 2004.