서버 클러스터 환경에서 에너지 절약을 위한 학습기반의 서버 전원 모드 제어

함치환, 김호연, 곽후근, 권희웅, 김영종, 정규식
숭실대학교 정보통신전자공학부
e-mail : hch@q.ssu.ac.kr

An Autonomous Learning based Server Power Mode Control for Saving Energy in a Server Cluster Environment

Chihwan Ham, Hoyeon Kim, Hukeun Kwak, Huiung Kwon,
Youngjong Kim, Kyusik Chung
School of Electronics Engineering, Soongsil University

요 약

서버 클러스터는 성능을 보장하기 위하여, 요청 수에 상관없이 미리 측정된 최대 부하 시점에서 처리 가능한 서버 수를 항상 운영하고 있다. 이것은 서비스의 품질은 보장할 수 있으나, 소비 전력 효율이 낮아 상당한 전력을 낭비하는 단점이 있다.

본 논문에서는 사용자 요청 수에 기반하여 동작시킬 서버의 수를 결정하는 알고리즘을 제안한다. 제안된 방식은 기존에 학습된 정보(과거 정보)와 현재의 정보에 근거하여 Off/Sleep 상태의 서버를 On 시키거나, On 상태의 서버를 Off/Sleep시켜, 현 시점에서 필요한 최적의 서버대수를 유지하도록 했다. 15대의 서버로 클러스터를 구성하고, SPECweb을 통해 실험을 수행하였다. 실험결과, 서버 모드를 제어할 경우에 제어하지 않는 경우에 비해 약 35%의 전력이 절감되고, 본 논문에서 제안하는 학습 방법을 추가로 적용할 경우 약 5%의 전력이 추가로 절감되었다.

1. 서론

데이터 센터는 '전기 먹는 하마'로 불릴 정도로 전력소비량이 많은 곳이다. 데이터 센터 IT 장비의 전력 소비 중 서버가 80% 차지하는데 이를 낮추는 게 그린 IT 실현을 위해 절실히 필요하다. 이를 해결하는 방법 중의 하나가 서버들 전력 소모를 최소화하도록, 필요하면 일부 서버를 일정시간 Off하는 부하 분산기를 구현하여 사용하는 것이다[1].

일반적으로 사용할 수 있는 부하 분산기로서 리눅스 가상 서버를 고려해 볼 수 있다. 리눅스 가상 서버(LVS: Linux Virtual Server)[2]는 리눅스를 기반으로 독립된 여러 서버들을 하나의 클러스터로 구성하여 뛰어난 확장성과 가용성을 제공한다. 가상 서버의 구조는 서버 외부에서는 마치 클러스터 서버가 하나의 고성능 서버인 것처럼 보이도록 하고, 실제 서버 내부에서는 사용자의 요청을 스케줄링을 이용하여 처리한다.

LVS는 사용자(Client)로부터 받은 요청을 처리하는 방식(Packet Forwarding Methods)에 따라 세 가지 방식(NAT: Network Address Translation, Direct Routing, IP Tunneling)이 존재한다. 이러한 방식은 서버가 처리한 요청을 LVS를 통해 사용자에게 전송하는 방식(NAT)과 서버가 직접 사용자에게 보내는 방식(Direct Routing, IP Tunneling)으로 나눌 수 있다.

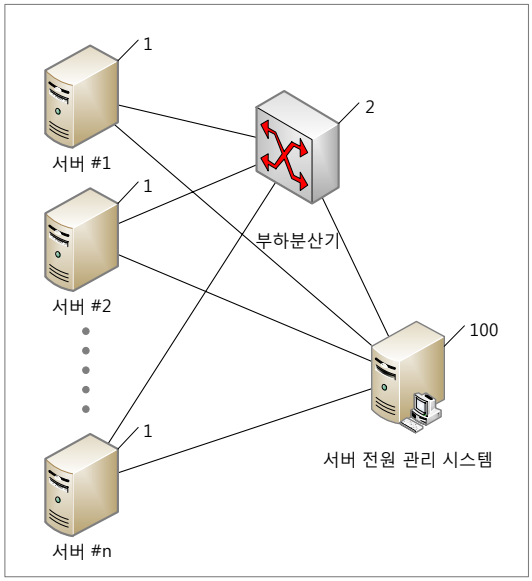
또한 LVS는 사용자 요청을 실제 서버들로 스케줄링 하는 방식에 따라 8가지 방식(RR: Round-Robin, WRR: Weighted Round-Robin, LC: Least Connection, WLC: Weighted Least Connection, LBLC: Locality-Based Least Connection, LBLCR: Locality-Based Least Connection with Replication, SH: Source Hash, DH: Destination Hash)이 존재한다. 스케줄링 방식은 실제 서버들로 요청을 분산할 때, 순서대로 분산하는 방식(RR, WRR), 가상 서버와 실제 서버와의 연결 개수를 이용해서 분산하는 방식(LC, WLC), 실제 서버들을 몇 개의 단위의 묶어서 분산하는 방식(LBLC, LBLCR) 및 해시를 이용하는 방식(SH, DH)으로 나눌 수 있다.

본 논문은 기존의 서버 전원 제어 시스템의 한계를 분석하고, 이를 개선할 새로운 학습 기반 서버 전원 제어 시스템을 제안한다. 본 논문의 구성은 다음과 같다. 제 2장에서는 기존의 서버 전원 제어 시스템의 한계를 소개한다. 3장에서는 기존의 서버 전원 제어 시스템의 문제점을 개선하는 학습에 기반한 서버 전원 제어 시스템을 제안한다. 4장에서는 실험 및 토론을, 5장에서는 결론을 제시한다.

2. 기존 연구

2.1 기존 서버 전원 제어 시스템

(그림 1)에 보이는 것과 같이 서버 전원 제어 시스템이 제안되었다[3]. 기존의 방법은 부하 분산기에 연결된 다수의 서버의 부하를 모니터링 하는 트래픽 모니터링 모듈과 다수의 서버에 인가되는 총부하가 미리 설정된 임계치 이하로 감소하면 부하 분산기의 설정을 변경함으로써 기동중인 서버 가운데 미리 정해진 순서에 따라 어느 하나의 서버에 대하여 부하의 분산을 막는 트래픽 제어 모듈이 존재한다. 또한 트래픽 제어 모듈에 의하여 부하의 분산이 막힌 서버의 트래픽이 미리 설정된 임계치 이하로 떨어지면 해당 서버에 대하여 서버 종료 신호를 발생시키는 서버 전원 제어 모듈로 구성된다.



(그림 1) 기존의 서버 전원 제어 시스템 구조

2.2 접근 방식

2.2.1 기존 서버 전원 제어 시스템의 문제점

기존 서버 전원 제어 시스템의 문제점은 다음과 같다.

(1) On/Off시 서버의 상황을 고려하지 않은 채 미리 정해진 순서로 동작시킨다.
(2) 서버가 On되었을 경우에 균등한 부하 분산을 보장할 수 없다.
(3) 현재의 상황만을 고려하여 상황에 대처하기 때문에 미래의 상황을 예측 및 대처할 수 없다.
(4) 갑작스런 요청의 급등(DDoS) 및 급감에 대처하기 위한 방안이 없다.

2.2.2 본 연구의 접근 방식

본 논문 연구의 접근방식은 다음과 같다.

(1) 학습된 내용을 기반으로 현 상황에 적합한 서버를 선정하고, 전환될 모드(Off/Sleep)를 결정한다.
(2) 기존 서버의 연결을 새로 켜진 서버에 마이그레이션하여 신속하게 균등한 부하 분산을 보장한다.
(3) 학습된 내용을 현재/미래의 상황에 대처 및 예측하기 위한 근거로 사용함으로써 효율적인 서버 운용을 보장한다.
(4) 트래픽 변화정도를 근거로 하여 요청의 급등/급감에 대처한다.

3. 학습에 기반한 서버 전원 제어 시스템

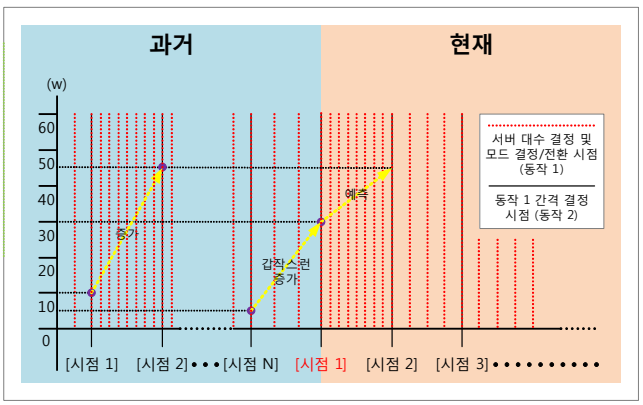
3.1 각 시점에서의 동작 과정

크게 두 가지 동작의 반복 수행으로 이루어진다. 하나는 현재 상황을 판단하여 최적의 서버 대수를 결정하여 그에 따라 일부 서버의 모드 변화를 결정하는 동작인데 이 동작이 일정 주기로 반복 수행된다(동작1). 다른 하나는 과거와 현재의 소모 전력을 판단하여 학습기반으로 위 동작의 주기를 조정하는 동작인데 이 동작 역시 일정 주기로 반복 수행한다(동작2). 동작2의 수행주기는 한번 결정되면 정적으로 운영되며, 동작1의 수행주기는 동작2에 의하여 결정되어 동적으로 운영된다.

동작1은 다음과 같다.
(1) 소모 전력에 관한 현재 상황을 고려하여 최적의 서버 대수를 결정하며 그에 따라 서버 모드 조절한다.

동작2는 다음과 같다.
(1) 현재의 소모 전력 정보를 학습 테이블에 갱신한다.
(2) 과거와 현재의 소모 전력을 비교하여 변화의 추이를 보아 (동작1)이 적용되는 서버 모드 조절의 주기가 전력 소모 상황변화에 알맞게 대처하는 지를 판단하여 필요하면 서버 모드의 조절의 간격을 조정한다.
(3) DDoS와 같은 아주 급격한 상황 변화가 감지될 경우 (동작1)을 바로 수행하게 만든다.

(그림 2)는 제안된 방법을 설명하는 그림이다. 과거 전력소모에 대한 정보를 유지하고 있으며(처음 나오는 [시점1]~[시점N]) 현재 시점이 [시점 1]부터 시작된다. 그림 2에서 세로 방향의 실선이 동작 2가 적용되는 시점이고 세로 방향의 점선이 동작 1이 적용되는 시점이다. 구체적으로 설명하면, 1번째 시점의 동작에서 현재 [시점 1]의 소모 전력 값은 30W이고, 과거의 소모 전력은 [시점 1]에서 [시점 2]로 증가했기 때문에, 서버가 증가되어야 함을 예측하고 동작 1의 간격을 조절한다. 그리고 과거 [시점 N]에서 현재 [시점 1]까지 갑작스런 증가로 DDoS와 같은 상황에 대비하여 전환 시점의 간격을 빠르게 하여, 모드 전환이 빨리 이루어질 수 있도록 대처한다. 동작 2가 끝나면 동작 1이 적용되고 동작 1에서는 서버의 대수를 판단하고 서버 모드 조절(On/Off)을 수행한다.



(그림 2) 각 시점에서의 동작 과정

3.2 서버 모드 전환 수행 과정

(그림 3)은 서버 모드를 전환하는 동작을 수행하는 과정을 나타내며, 그림의 단계별 설명은 다음과 같다.

단계 (1) : 클라이언트가 서버에 요청함으로써 트래픽이 발생하고, 이를 사용량 모니터링 모듈로 측정하여 에너지 소모 추정 모듈을 통해 소모 전력을 획득한다.

단계 (2) : 서버대수 조정 판단 모듈은 학습 테이블을 통하여, a) 서버 대수를 결정하는 시점인가, b) 서버 모드를 결정 및 전환하는 시점인가를 판단한다.

단계 (3-a) : 현재 시점에서 학습 테이블의 과거 동일 시점의 값과 현재의 값을 비교하여, 현재 시점에서 필요한 최소 서버대수를 결정하고, 학습 테이블에 최소 서버대수를 저장하며, 서버 모드의 결정 및 전환 간격을 결정한다.

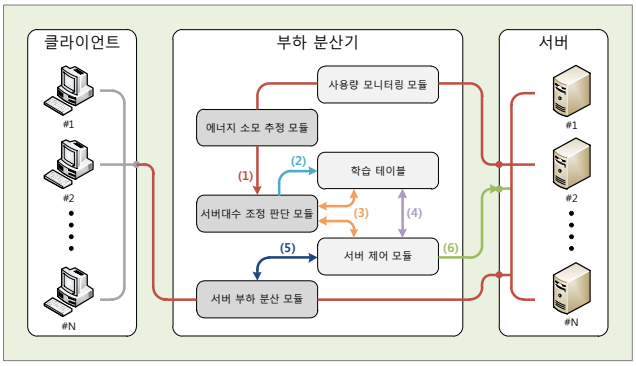
단계 (3-b) : 결정된 간격마다 서버 제어 모듈은 서버의 모드를 결정하고, 모드 전환을 지시하며, 모든 전환이 끝나면 학습 테이블에 모드가 전환된 서버를 저장한다.

단계 (4) : 서버 제어 모듈은 학습 테이블의 최소 서버대수와 서버의 모드 상태를 확인하고, 모드를 전환할 서버를 결정하여 모드 전환 a) On, b) Off 을 실행한다.

단계 (5-a) : 서버 모드를 On모드로 전환해야 할 때, 전환 될 서버를 켜고, On상태가 된 서버의 부하 분산을 다른 서버와 균등하게 한다. 이때 추가 기능으로 빠른 안정화를 위해서 다른 서버의 연결을 전환된 서버로 마이그레이션을 한다.

단계 (5-b) : 서버 모드를 Off모드로 전환해야 할 때, 전환될 서버를 끄기 전에 부하 분산 목록에서 제거시킴으로써, 서버의 연결이 완전히 종료되었을 때 서버의 모드 전환을 한다. 이때 추가 기능으로 빠른 연결 종료를 위해 전환될 서버의 연결을 다른 서버로 마이그레이션을 한다.

단계 (6) : 학습 테이블의 정보에 따라 서버의 Off 방법은 On 방법을 고려하여 (Sleep 모드)와 (일반 Off 모드)중에 결정된다. 학습 테이블의 정보에서 모드 결정 및 전환 간격이 작다면 빠른 부팅을 필요로 하는 (Sleep 모드)로 Off 방법을 결정하고, 간격이 크다면 (일반 Off 모드)로 Off 방법을 결정한다.

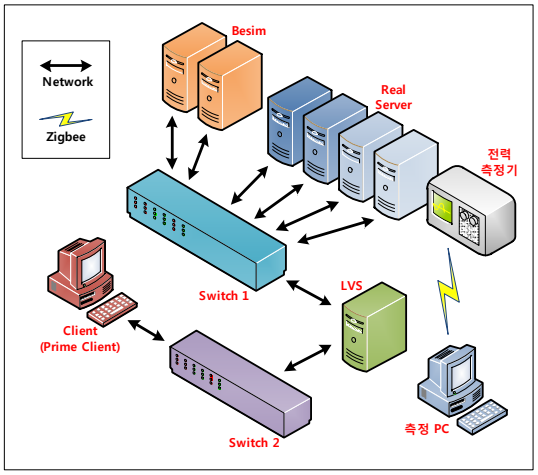


(그림 3) 서버 모드 전환 수행 과정

4. 실험 및 토론

4.1 실험 환경

(그림 4)는 실험 전체 구성을 나타낸다. 전력 측정기는 Real Server의 전력 값을 측정하여, Zigbee를 통해 측정 PC로 전송한다. Switch 2대를 이용하여 Real Server와 Client를 각각 분리된 네트워크를 구성하였다.



(그림 4) 실험 전체 구성도

<표 1>은 실험에 사용된 하드웨어와 소프트웨어를 나타낸다.

<표 1> 실험용 하드웨어 & 소프트웨어

		하드웨어		소프트웨어	개수
		CPU (Hz)	RAM (MB)		
LVS		Quad Q9450 2.66G	2048	RR/WLC	1
SPEC web	Client (Prime Client)	Quad Q6600 2.4G	4096	SPECweb	1
	Besim	P-IV 2.4G	512	apache	2
	Real Server	P-IV 2.4G	512	apache	15

4.2 실험 방법

각 실험은 <표 2>와 같이 진행되었으며, 알고리즘의 적용 및 비적용을 반복하였고, 요청 패턴은 SPECweb[4]의 Banking Design[5]을 이용하였으며, 실험 3은 (예외적인 요청 패턴)으로 실험하였다. (예외적인 요청 패턴)은 서버로의 요청의 급등이나 급감이 많은 패턴을 말한다. 실험 1은 LVS의 RR 부하분산방법이 적용되었으며 실험 2와 3은 LVS의 WLC 부하분산방법이 적용되었다.

<표 2> 각 실험 진행 방법

	적용 알고리즘	요청 패턴	시간
실험 1	전원 제어 알고리즘	SPECweb Banking Design (일반적인 요청 패턴)	90분
실험 2	학습 알고리즘	SPECweb Banking Design (일반적인 요청 패턴)	40분
실험 3	학습 알고리즘	SPECweb Banking Design (예외적인 요청 패턴)	40분

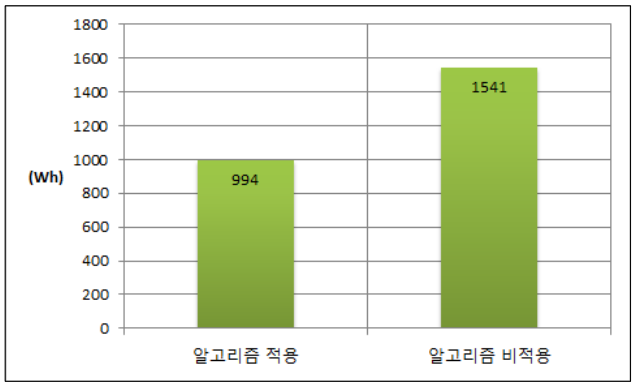
4.3 실험 결과

(그림 5)는 서버 전원 제어 알고리즘이 적용 및 비적용한 총 소비 전력의 결과이다. 서버 전원 제어 알고리즘 적용 아래, 학습 알고리즘을 적용 및 비적용한 (일반적인 요청 패턴)에서의 총 소비 전력의 결과는 (그림 6)이고, (그림 7)은 (예외적인 요청 패턴)에서의 총 소비 전력의 결과이다.

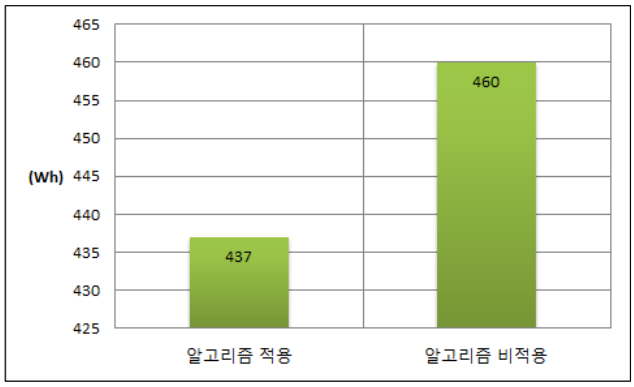
이 결과는 전력 측정기를 사용하여 측정한 값으로 하드웨어적으로 실제 소비되는 전력을 의미한다.

서버 전원 제어 알고리즘을 적용할 경우와 그렇지 않을 경우의 차이는 약 550Wh 차이를 보였다. 서버 전원 제어 알고리즘을 적용할 경우 약 35%의 소비전력이 절감되었음을 확인 하였다.

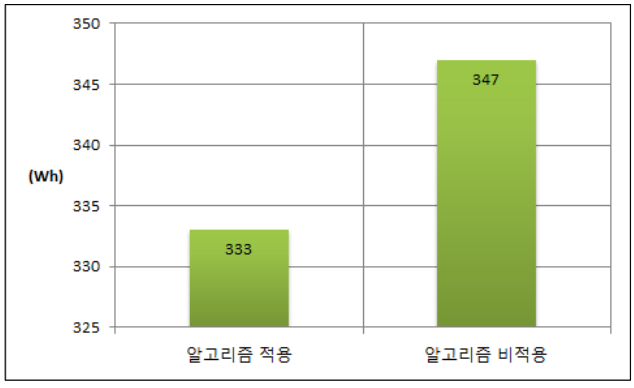
추가적으로 학습 알고리즘을 적용하였을 경우, (일반적인 요청 패턴)에서는 알고리즘을 적용할 경우와 그렇지 않을 경우의 차이가 약 23Wh 발생했으며, 약 5%의 전력이 절감되었음을 확인하였다. (예외적인 요청 패턴)에서는 약 14Wh 차이를 보였으며, 약 4%의 전력이 절감되었음을 확인할 수 있었다.



(그림 5) 서버 전원 제어 알고리즘 소비 전력 비교



(그림 6) 학습 알고리즘 (일반적인 요청 패턴) 소비 전력 비교



(그림 7) 학습 알고리즘 (예외적인 요청 패턴) 소비 전력 비교

4.4 토론

서버 전원 제어 알고리즘을 적용함으로써 35%의 소비 전력 절감을 확인할 수 있었다. 이것은 부하량에 따라 필요한 서버만을 동작시키고 나머지 서버를 Off시킴으로써 얻어진 결과이다.

서버 전원 제어 알고리즘에 더하여 학습 알고리즘을 적용할 경우, 경우에 따라 추가적으로 평균 4.5%의 소비 전력 절감이 가능함을 확인 하였다. 이것은 과거의 기록을 근거로 한 학습을 통하여 미래의 부하량을 예측하고, 서버의 모드를 전환하는 간격을 조절함으로써 보다 빠르게 상황에 대처할 수 있었기 때문이다. 학습 알고리즘을 적용한 (일반적인 요청 패턴)과 (예외적인 요청 패턴)의 소비 전력 차이는 요청 수가 (일반적인 요청 패턴)이 더 많기 때문이다.

정리하면, 학습에 기반한 서버 전원 제어 알고리즘을 적용하였을 경우, 서버 전원 제어 알고리즘을 전혀 사용하지 않은 경우에 비해 약 39.5%의 전력 절감이 가능하다.

5. 결론

부하 분산 시스템에 서버 전원 제어 시스템 및 알고리즘을 적용한다면, 소비 전력을 절감 시킨다는 것이 명백해졌다. 기존에 제안된 방법은 서버의 특성을 고려하지 않기 때문에, 상황에 맞는 효율적인 서버 모드 전환을 하지 못하며, 이것은 예외적인 상황에서 성능 저하와 소비 전력 증가를 야기할 수 있다. 또한, DDoS 같은 요청의 급등이나 급감에 대처하기 위한 방법을 제공하지 않았기 때문에, 예외의 상황에서는 성능의 신뢰가 떨어진다.

반면, 본 논문에서 제안한 알고리즘은 학습을 기반하고 서버의 특성을 고려함으로써, 상황에 맞는 최적의 서버 모드 전환을 제공한다. 이것은 성능 저하를 막고, 추가적인 소비 전력 절감이 가능함을 의미한다. 또한, 학습된 내용과 현재 상황을 기반으로 하여, 미래를 예측함으로써, 요청의 급증 및 급감에 대비할 수 있다. 이는 어떠한 상황에서도 성능의 신뢰를 지킬 수 있음을 의미한다.

참고문헌

[1] 이상학, 문성준, 김진환, 신상용, 서용원, 최영진, “공공부문의 그린 데이터 센터 구현 방안에 관한 연구”, 한국정보과학회지, 제 27권, 제 11호, 2009.

[2] LVS(Linux Vitual Server), http://www.linuxvirtualserver.org

[3] 서버 전원 관리 시스템 및 그 방법, 엔씨소프트, 공개특허 10-2010-0113383

[4] SPECweb. http://www.spec.org/

[5] SPECweb BankingDesign. http://www.spec.org/web2005/docs/BankingDesign.html