서버 클러스터 환경에서 에너지 절약을 위한 전력 정보 기반의 동적 서버 부하분산

김동준, 강남용, 권희웅, 곽후근, 김영종, 정규식
숭실대학교 정보통신전자공학부
e-mail: djkim@q.ssu.ac.kr

A Dynamic Server Load Balancing based on Power Information for Saving Energy in a Server Cluster Environment

Dongjun Kim, Namyong Kang, Huiung Kwon, Hukeun Kwak,
Youngjong Kim, Kyusik Chung
School of Electronics Engineering, Soongsil University

요 약

서버 클러스터에서 부하 분산기는 사용자의 요청을 각 서버로 분산시키는 역할을 한다. 리눅스 가상 서버(LVS: Linux Virtual Server)는 소프트웨어적으로 사용되는 부하 분산기로서 여러 가지 스케줄링 방식들을 가지고 있다. 그러나 부하 분산 시에 서버의 유동적인 부하 정보를 반영하지 못하는 단점이 있다. 이에 개선된 방식으로 서버의 동시 연결 개수에 따라 상한계(Upper Bound)와 하한계(Lower Bound)를 설정하고, 요청을 분산하는 동적 스케줄링(Dynamic Scheduling)이 존재한다. 그러나 서버의 상태에 따라 상한계와 하한계가 바뀔 수 있음에도 불구하고 이 값들이 고정되어 있다는 단점을 가진다.

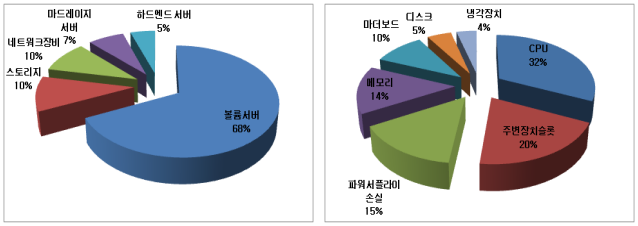
본 논문에서는 기존 부하 분산 방법의 단점을 극복하는 서버 전력 정보에 기반한 스케줄링 방식을 제안한다. 제안된 방식은 서버의 부하 정보를 기반으로 에너지를 추정하고 전력 수치를 기반으로 LVS의 가중치 테이블을 주기적으로 갱신한다. 그리고 부하 분산기는 클라이언트로부터 요청 받은 트래픽을 각 서버의 에너지 소모 상태에 따라 적용시킴으로써 에너지 소모가 최소화되도록 부하를 분산한다. 또한 서버의 상태에 따라 상한계와 하한계가 바뀔 수 있음을 고려하여 상한계와 하한계를 설정하지 않고 서버의 상태에 따라 적절하게 요청이 분배되도록 하였다. 15대의 PC를 사용하여 실험을 수행하였으며, 실험 결과는 기존 부하 분산 알고리즘 중 성능이 가장 좋은 알고리즘에 비해 서버의 성능이 동일한 경우 성능 및 소비전력 면에서 거의 동등하였고, 서버의 성능이 상이한 경우 50.2% 성능 향상 및 27.3% 소비 전력 절감을 확인하였다.

1. 서론

지난 수년 동안 IT의 급속화된 발전과 함께 인프라를 이루는 데이터 센터의 서버 장비들의 성능 또한 향상되고 있다. 서버의 성능이 향상됨에 따라 데이터 센터의 전력 소모 또한 꾸준히 증가하고, 이에 따라 배출되는 이산화탄소 또한 매년 증가하고 있다. 이러한 배경으로 최근 IT 분야에서 녹색 성장을 추구하는 그린 IT가 주목을 받고 있다. 2009년 5월 13일에 발표된 국가 녹색 성장 위원회의 "저탄소 녹색 성장을 위한 그린 IT 국가 전략"에서 정의한 그린 IT는 녹색(Green)과 정보통신기술(IT)의 합성어로 규정하고 "IT 부문 녹색화(Green of IT)", "IT 융합 경제사회 저탄소화(Green by IT)", "IT를 활용한 기후 변화 대응 역량 강화(IT for Green)" 분야들을 포함하는 포괄적 의미이다.

그린 IT 기술이 저탄소 녹색 사회 구현에 핵심적인 이유는 IT 부문이 차지하는 온실 가스 방출량은 2%에 불가하나 기존 타 산업 분야의 85%에 이를 적용 및 활용하여 상대적인 이산화탄소 방출량의 감축이 가능하기 때문이다. 또 다른 이유는 세계적으로 폭발적인 IT 장비 확산으로 인터넷에서 활용되는 트래픽 량이 2006년 637 Gbps에서 2025년은 190배인 121 Tbps로 예측됨에 따라 2025년경에 전 세계 전기 사용량의 15% 이상이 IT 장비에 의하여 소비될 것이라는 진단 예측에 기인한다. 특히 서버, 스토리지 및 네트워크 장비들을 직접 운용하는 데이터 센터는 "전기 먹는 하마"로 불릴 정도로 전력 소비량이 많은 곳으로 그린 IT를 실현하는 데 있어 우선적인 고려 대상이다. 데이터 센터는 크게 IT 장비와 이를 안정적으로 운용하기 위한 기반 설비로 나눌 수 있다. IT 장비는 데이터 센터마다 약간의 차이는 있으나 데이터 센터에서 사용하는 전체 에너지 소비량의 약 50% 이상을 차지한다.

또한, (그림 1)에서처럼 데이터 센터 IT 장비의 전력 소비 중 서버가 차지하는 비중은 80%에 달하며 서버 전력 소비 중 CPU가 가장 많은 32%의 비중을 차지한다[1]. 이를 종합해보면 데이터 센터 내 서버, 특히 CPU 전력 소비를 줄이는 게 그린 IT 실현을 위해 매우 중요한 일임을 알 수 있다.

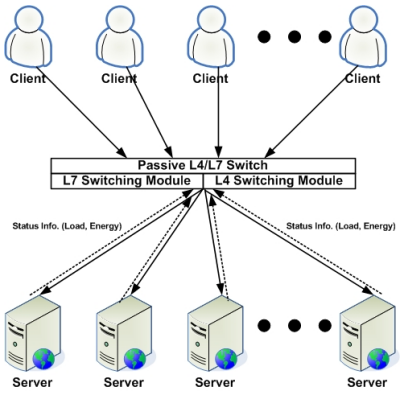


(a) 데이터 센터 전력 소비 비중 (b) 서버 전력 소비 비중

(그림 1) 데이터 센터 IT 장비/서버 전력 소비 비중

데이터 센터에는 규모에 따라 서버 수백/수천/수만대로 구성되어 있고, 이 서버들이 담당하는 서비스별로 그룹으로 나뉘는데 이 그룹을 각각 하나의 서버 클러스터라고 부른다. 서버 클러스터 규모에 따라 하나의 부하 분산기가 하나의 서버 클러스터 또는 여러 서버 클러스터의 부하 분산을 담당한다. 한 대의 부하 분산기가 수십/수백/수천대의 서버를 담당한다는 점을 고려할 때 부하 분산기에 의한 에너지 절감 효과가 매우 클 것이다.

본 논문에서는 이러한 부하 분산기에 적용시킬 수 있는 서버의 전력 소모 상태에 따라 트래픽을 분산 하는 스케줄링 방식을 제안한다. 제안 시스템은 각 서버의 전력 정보를 추정하고 이에 따라 클라이언트로부터 들어오는 트래픽의 부하를 분산하여 상이한 처리 능력의 서버에 능동적으로 대처하여 성능을 높이고, 이에 따라 전체적인 서버 클러스터의 전력을 절약하는 부하 분산 방식을 의미한다. (그림 2)는 전력 기반 부하 분산기의 전체적인 구조를 나타낸다.



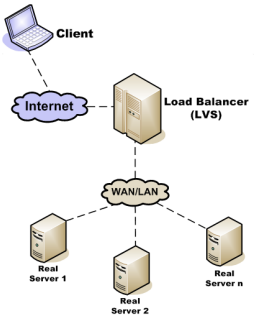
(그림 2) 전력 기반 부하 분산기

본 논문의 구성은 다음과 같다. 2장은 LVS와 동적 스케줄링에 대한 특징 및 단점에 대해 설명하고, 3장에선 서버 전력에 기반한 동적 스케줄링(DS-SPB) 알고리즘에 대해 기술한다. 4장은 실험 환경, 실험 결과 및 토론을, 5장은 결론 및 향후 연구 방향을 기술한다.

2. 기존연구

2.1 리눅스 가상 서버(LVS: Linux Virtual Server)[2]

리눅스 가상 서버(LVS: Linux Virtual Server)는 리눅스를 기반으로 독립된 여러 서버들을 하나의 클러스터로 구성하여 뛰어난 확장성과 가용성을 제공한다. (그림 3)은 리눅스 가상 서버의 전체적인 구조를 나타낸다.



(그림 3) 리눅스 가상 서버의 구조

2.2 LVS 스케줄링[3]

LVS는 클라이언트로부터 받은 요청을 6가지 스케줄링 방식을 이용하여 분산시킨다. 각각의 스케줄링 방식은 다음과 같다.

(1) RR(Round-Robin) & WRR(Weighted Round-Robin)

RR은 클라이언트로부터 오는 요청을 순서대로 서로 다른 서버로 보내고, WRR은 서버의 처리 용량이 다를 때, 각 서버의 처리 용량에 비례하는 가중치를 두어 요청을 분산한다.

(2) LC(Least Connection) & WLC(Weight Least Connection)

LC는 연결된 개수를 측정하여 가장 작은 연결 개수를 가지는 서버로 요청을 분산하고, WLC는 각 서버의 처리 용량에 따라 가중치(Weight)를 할당하고 이 가중치에 따라 요청을 할당한다. 가중치가 큰 서버로 더 많은 요청을 할당 하게 된다.

2.3 동적 스케줄링(Dynamic Scheduling)[4]

동적 스케줄링은 기존 LVS 스케줄링의 단점을 보완하기 위해 제안된 스케줄링 알고리즘이다. 서버의 동시 연결 개수를 이용하여 부하를 분산하는 방식으로 Throttling Mechanism[5]을 스케줄링에 적용하였다. Throttling Mechanism은 상한계(Upper Bound)와 하한계(Lower Bound)를 두고 상한계를 넘어선 서버에게는 더 이상 작업을 할당하지 않고, 하한계로 떨어지면 다시 작업을 할당하는 방식으로 동작하게 된다.

2.4 접근 방식

(1) LVS 스케줄링의 단점

LVS 스케줄링 알고리즘이 가지는 단점을 정리하면 <표 1>과 같다.

<표 1> LVS 스케줄링 알고리즘의 단점

스케줄링	단 점
RR	서버의 처리 능력이나 요청 컨텐츠가 다른 경우 이를 반영하지 못한다.
LC	사용자의 요청 컨텐츠가 다른 경우 이를 반영하지 못한다.

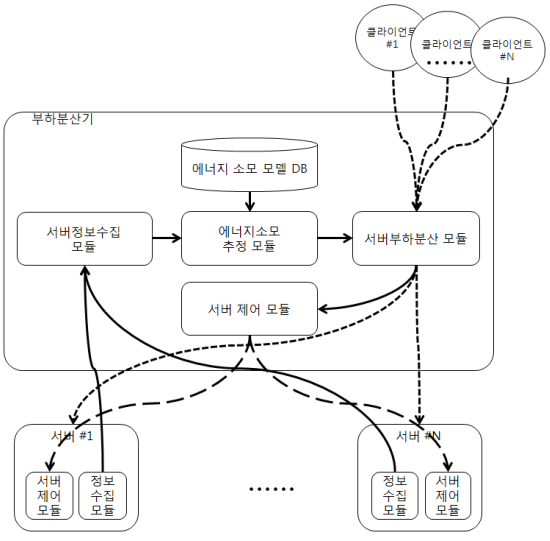
(2) 동적 스케줄링의 단점

- 사전 실험을 통하여 상한계와 하한계를 설정해주어야 한다.
- 상한계와 하한계는 사전 실험으로 고정되어 있어 서버의 처리 능력이 바뀔 때마다 이를 다시 수정해주어야 한다.

(3) 본 연구의 접근 방식

제안된 알고리즘은 서버의 전력 정보를 이용하여 사전 실험 없이 서버의 상태가 다른 경우를 능동적으로 부하 분산에 반영하는 방식이다.

3. 호스트 에너지에 기반한 동적 스케줄링 (DS-SPB)



(그림 4) 제안된 스케줄링 알고리즘

(그림 4)는 2.3절에서 언급한 기존 LVS 스케줄링과 동적 스케줄링의 문제점을 기반으로 이를 해결할 수 있도록 제안된 알고리즘을 나타낸다. 제안된 알고리즘은 서버의 처리 능력이 바뀔 때마다 수정할 필요 없이, 전력 정보를 통해 서버의 처리 능력을 LVS가 스스로 파악하여 능동적으로 적용할 수 있고, 서버의 처리 능력에 따라 부하를 분산하는 방식이므로 상한계/하한계를 지정할 필요가 없다.

1 단계 : 서버들은 정보 수집 모듈을 통해 주기적으로 장치별 사용량(Utilization) 정보를 측정한다.

2 단계 : LVS는 서버들로부터 서버 정보 수집 모듈을 통해 주기적으로 정보를 수집한다.

3 단계 : 에너지 소모 추정 모듈은 수집된 정보를 바탕으로 아래의 수식을 통해 전력 소모량을 추정한다[6]. 각 장치의 전력 정보는 소모 모델 데이터베이스를 통해 얻는다.

$$P_i = B_i + \sum_r M_{r,i} \times \frac{R_{r,i}}{C_{r,i}}$$

P_i = i번째 서버의 소비 전력

B_i = i번째 서버의 유휴 상태 소비 전력

$M_{r,i}$ = i번째 서버 r장치의 최대 사용량에서의 소비 전력

$R_{r,i}$ = i번째 서버 r장치의 사용량

$C_{r,i}$ = i번째 서버 r장치의 용량

4 단계 : LVS는 3단계에서 추정된 전력 정보를 통해 각 서버에 대한 가중치 테이블을 주기적으로 갱신한다.

5 단계 : 사용자 요청을 받은 LVS는 현재 각각의 서버들에 대한 가중치 테이블을 검사하여 가중치에 비례하는 방식으로 요청을 분산한다.

4. 실험 및 토론

4.1 실험 환경

<표 2>는 실험에 사용된 하드웨어와 소프트웨어를 나타낸다.

<표 2> 실험용 하드웨어 및 소프트웨어

	하드웨어		소프트웨어	개수
	CPU (Hz)	RAM (MB)		
사용자	Quad Q6600 2.4 G	4096	AB[7]	1
LVS	Quad Q9450 2.66 G	2048	DR[8]	1
서버	P-4 2.4 G	512	Fedora	15

4.2 실험 방법

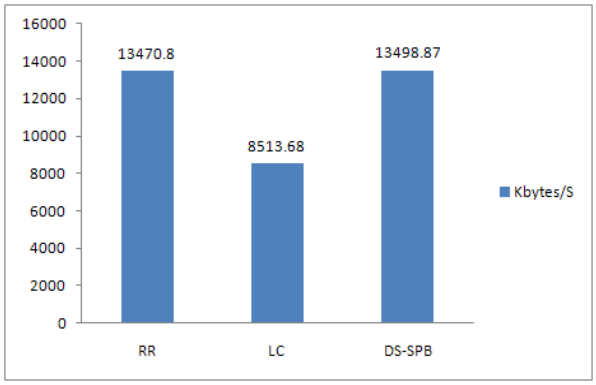
실험은 아래의 7 단계를 적용하였다.

(1) 실험에 사용할 스케줄링 알고리즘을 LVS에 세팅한다.
(2) 서버의 소비전력을 측정하기 위해 전력계 연결 후 초기화한다.
(3) AB(Apache Bench)의 concurrency를 500, request를 4,000,000 으로 설정하여 LVS에 웹페이지를 요청한다.
(4) 초당 요청 개수를 측정한다.
(5) 소비 전력을 전력계(HPM-100A)를 통해 측정한다.
(6) 다음 스케줄링 알고리즘으로 변경한다.
(7) 초기화 후 1 ~ 6 과정을 반복한다.

4.3 실험 결과

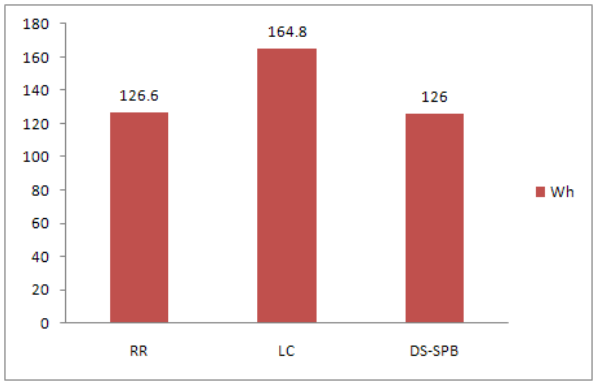
(1) 서버의 성능(상태)이 모두 동일한 경우

(그림 5)는 스케줄링 별 초당 요청수를 나타낸다. 같은 크기(1.4K)의 HTML 문서를 요청하는 방식으로 실험하였다.



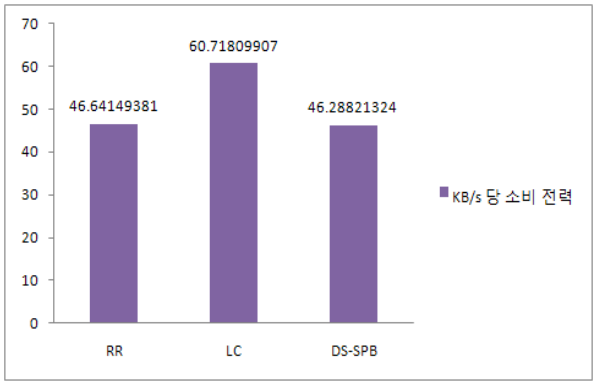
(그림 5) 스케줄링 별 초당 요청 수

(그림 6)은 스케줄링 별 누적 소비 전력을 나타낸다. AB(Apache Bench)를 실행함과 동시에 서버의 소비 전력을 측정하는 방식으로 실험하였다.



(그림 6) 스케줄링 별 누적 소비전력

(그림 7)은 스케줄링 별 KB/S 당 소비전력을 나타낸다. 서버의 평균 소비전력에 초당 요청수를 나누어 계산된다.

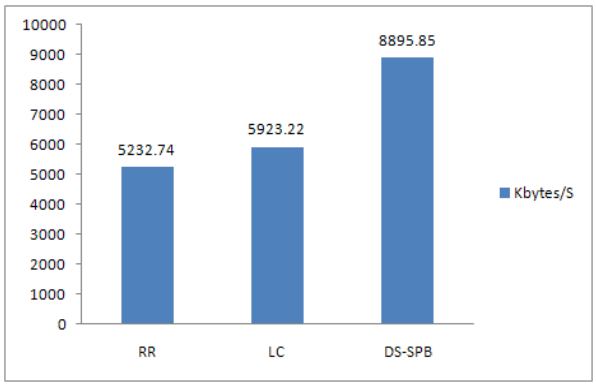


(그림 7) 스케줄링 별 KB/S 당 소비전력

(2) 서버의 성능(상태)이 상이한 경우

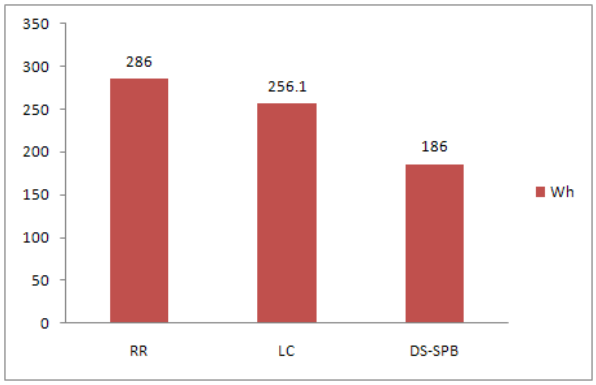
15대의 실험 서버 중 임의로 하나의 서버를 선택하여 외부 서버와 AB를 통해 지속적으로 트래픽을 요청하여 CPU의 40% 부하와 Network I/O 장치에 24% 부하를 가하여 서버의 성능을 상이하게 설정하였다. 이하 설정은 '서버의 성능이 동일한 경우' 와 같다.

(그림 8)는 스케줄링 별 초당 요청수를 나타낸다. 실험 방법은 '서버의 성능이 동일한 경우' 와 같다.



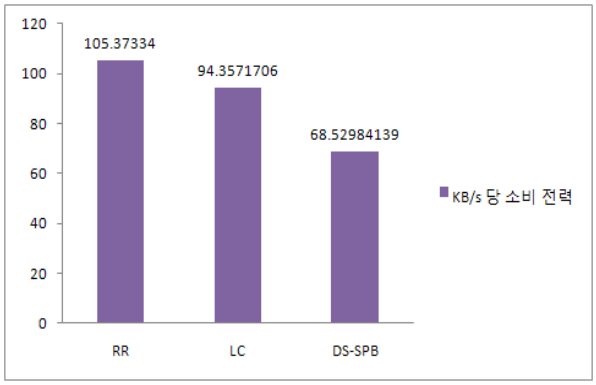
(그림 8) 스케줄링 별 초당 요청 수

(그림 9)은 스케줄링 별 누적 소비 전력을 나타낸다. 실험 방법은 '서버의 성능이 동일한 경우' 와 같다.



(그림 9) 스케줄링 별 누적 소비전력

(그림 10)은 스케줄링 별 KB/S 당 소비전력을 나타낸다. 계산 방법은 '서버의 성능이 동일한 경우' 와 같다.



(그림 10) 스케줄링 별 KB/S 당 소비전력

4.4 토론

제안된 DS-SPB 스케줄링은 다른 스케줄링에 비해 서버의 성능(혹은 상태)이 상이한 경우 높은 성능 향상과 소비전력 감소가 나타났다. 이는 기존 스케줄링 방식은 서버의 상태를 고려하지 않거나 가중치가 고정적이기 때문이다. 그리하여 서버의 유효자원을 효율적으로 사용하지 못하고, 누적 소비 전력 또한 증가한다. 하지만, 제안된 DS-SPB 스케줄링은 각 서버의 CPU, Memory 그리고 HDD의 사용량을 이용한 서버 전력 추정 모델에 따라 동적으로 부하를 분산함으로 유효 자원이 더 많은 서버에게 부하를 더 분산하여 더 좋은 성능을 보인다. 더욱이, 어느 특정한 서버에 과부하 없이 서버의 자원을 효율적으로 사용하기 때문에 서버 누적 소비 전력이 감소됨을 볼 수 있다.

5. 결 론

기존 방법이 서버의 처리 능력에 무관한 방법인 반면 제안된 방법은 서버의 전력 소모 상태를 이용하여 동적으로 부하를 분산하도록 하였다. 또한 실험을 통해 제안된 방법이 서버의 성능(혹은 상태)이 상이한 경우 기존 스케줄링 방법에 비해 높은 성능 향상과 전력 소비 절감을 확인하였다.

참고문헌

[1] 박성수, 박선택, "그린 IT 제품 동향", 한국통신학회지:정보와 통신, 제26권, 제9호, 2009.

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

[3] LVS Scheduling Algorithms, http://www.linuxvirtualserver.org/docs/scheduling.html.

[4] S. Hwang and N. Jung, "Dynamic Scheduling of Web Server Cluster", Proceedings of the 9th International Conference on Parallel and Distributed Systems, IEEE, 2002.

[5] C. A. Ruggiero and J. Sargeant, "Control of Parallelism in the Manchester Dataflow Machine", In Functional Programming Language and Computer Architecture, LNCS 274, Springer-Verlag, pp. 1-15, 1987.

[6] Nidh Singh and Shrisha Rao, "Energy Optimization Policies for Server Clusters", In IEEE Conference on Automation Science and Engineering, Toronto, Ontario, Canada, August 2010.

[7] AB(Apache Bench), http://www.apache.org.

[8] Virtual Server via Direct Routing, http://www.linuxvirtualserver.org/VS-DRouting.html