

# 리눅스 가상 캐시 서버를 이용한 멀티미디어 스트리밍 시스템의 성능평가

윤상용<sup>0</sup> 윤창일 이용우  
서울시립대학교 전자전기컴퓨터공학부  
(racc<sup>0</sup>, cruise45)<sup>0</sup>@metalab.uos.ac.kr, ywlee@uos.ac.kr

## Performance Evaluation of Multimedia Streaming System with Linux Virtual Cache Server

Sang Yong Yoon<sup>0</sup> Chang Il Yun Yong Woo Lee  
Faculty of Electrical & Computer Engineering, The University of Seoul

### 요 약

고용량 멀티미디어 데이터의 사용 증가와 인터넷 사용 인구의 폭발적인 증가로 현재의 네트워크 인프라로는 사용자의 요구를 충분히 수용하지 못하고 있는 상황이다. 이러한 문제를 해결하기 위해 제시된 웹 캐싱 기술도 캐시 서버에 부하가 집중될 때에 지속적인 서비스를 해 주는 것이 불가능한 경우가 발생한다. 본 논문에서는 여러 대의 캐시 서버를 다양한 형태의 가상 캐시 서버로 구축하여 캐시 서버 한 대에 집중되는 부하를 분산하는 방안에 대해서 연구하였다. 이를 위하여 가상 캐시 서버 시스템을 다양한 구성 방식과 알고리즘에 따라 구축하고 성능을 평가하였다. 이런 과정을 거쳐서 멀티미디어 데이터의 스트리밍 서비스에 가장 적합한 구성 및 알고리즘을 제시하였다.

### 1. 서 론

최근 인터넷의 사용이 대중화되어가면서 고용량 멀티미디어 데이터의 사용이 늘고 있다. 이와 함께 인터넷 사용 인구의 폭발적인 증가로 현재의 네트워크 인프라로는 사용자의 요구를 충분히 수용하지 못하고 있는 상황이다.

이러한 문제를 해결하기 위해 제시되는 효과적인 해결 방법 중 하나가 바로 웹 캐싱 기술[1]이다. 웹 캐싱 기술은 네트워크 상에서 발생하는 병목 현상을 줄이고, 지연시간을 최소화시켜줄 뿐만 아니라 웹 서버의 부하도 경감시켜 준다.

그러나 이러한 웹 캐싱 기술도 캐시 서버에 부하가 집중될 때에 지속적인 서비스를 해 주는 것이 불가능한 경우가 발생한다.[2] 최근에 많은 사람들이 이와 관련되어서 리눅스 가상 서버[3]와 같은 부하 분산에 관련된 연구를 하고 있다.

본 논문에서는 여러 대의 캐시 서버를 하나의 가상 캐시 서버로 구성하여 캐시 서버 한 대에 집중되는 부하를 분산할 수 있도록 하였다. 그 결과 더 많은 클라이언트의 요청을 처리할 수 있도록 하는 가상 캐시 서버 시스템을 다양한 구성 방식과 알고리즘에 따라 구축하고 성능을 평가하였다. 이런 과정을 거침으로써 멀티미디어 데이터의 스트리밍 서비스에 가장 적합한 가상 캐시 서버의 구성 및 알고리즘을 찾아내었다.

### 2. 리눅스 가상 서버

가상 서버라는 말은 실제로 존재하는 서버가 아니라, 여러 대의 서버들을 하나로 모아서, 마치 하나의 서버처럼 동작하도록 한다는 개념에서 나온 것이다. 따라서 가상 서버 구성은 실제로 동작하는 여러 개의 독립적인 서버들을 일정한 방식에 의해 하나로 모은 후, 이들을 대표하면서 부하를 분산하는 로드 밸런서를 두는 방식으로 구성되어진다.

리눅스 가상 서버는 Direct Routing, IP Tunneling, NAT(Network Address Translation)등의 3가지 구성 방식과 RR, WRR, LC, WLC, LBLC, LBLCR, DH, SH등의 8가지 부하 분배 스케줄링 알고리즘[4]으로 구성할 수 있다. 따라서 리

눅스 가상 서버는 구성 방식과 스케줄링 알고리즘에 따라 총 24가지의 다른 방식으로 구성할 수 있다. 데이터 전송시의 성능은 구성 방식 중 NAT 방식이 가장 우수한 성능을 보이며, 스케줄링 알고리즘에서는 LBLC와 LBLCR의 경우 다른 알고리즘에 비해 좋은 성능을 보인다.[5] 본 논문에서는 3가지 구성 방식별로 LBLC와 LBLCR 알고리즘을 사용하여 시스템을 구성하고, 성능을 측정하였다.

### 3. 실험 환경 및 방식

#### 3.1. 실험 환경

실측에 사용되는 시스템은 그림 1에서 보듯이 하나의 웹 서버가 존재하고, 그 밑에 실제 캐시 서버들을 하나의 가상 캐시 서버로 구성해서 웹 캐싱 서비스를 제공하는 구조이다. 캐시 서버를 웹 서버 측면에 배치하는 이러한 방식을 후방배치캐싱(Remote Proxy Caching)이라고 한다. 웹 서버에 대한 모든 요청은 가상 캐시 서버를 통하게 되며, 가상 캐시 서버에서는 로드밸런서가 클라이언트들의 요청을 적절히 분배해서 실제 캐시 서버로 보내게 된다. 실제 캐시 서버에서는 이 요청을 받아서 캐싱 서비스를 하도록 하였다.

실험에 사용된 클라이언트 컴퓨터는 38대이며, 로드밸런서와 웹 서버는 각각 1대, 실제 캐시 서버는 총 3대로 구성하였다. 실험에 사용된 컴퓨터의 사양들은 표 1과 같다.

#### 3.2. 실험 방식

실측의 정확도를 높이기 위해서 우선 캐시 서버 내에 캐싱되어 있는 모든 데이터를 제거하고, 실측에 사용될 실제 데이터를 가지고 있는 웹 서버에 클라이언트가 멀티미디어 스트리밍 데이터를 요청하게 한다. 이때 클라이언트는 컴퓨터 하나당 3개의 요청을 하도록 하였다. 실측에 사용된 멀티미디어 스트리밍 데이터는 표 2와 같다.

캐시 서버에 큰 부하를 줄 목적으로 표 2와 같은 고용량의 멀티미디어 스트리밍 데이터를 가지고 실험을 하였다. 초기에 8대의 클라이언트가 웹 서버에 멀티미디어 스트리밍 데이터를

요청한다. 그리고 10분마다 추가로 6대의 클라이언트가 웹 서버에 멀티미디어 스트리밍 데이터를 요청하게 하였다. 이렇게 되면 50분 뒤에는 38대의 클라이언트가 웹 서버에 데이터를 요청하게 된다. 캐시서버에 대한 실험은 90분을 하였으며, 웹 서버, 로드밸런서, 실제 캐시 서버들에 대한 CPU 사용률과 네트워크 사용률을 측정하였다.

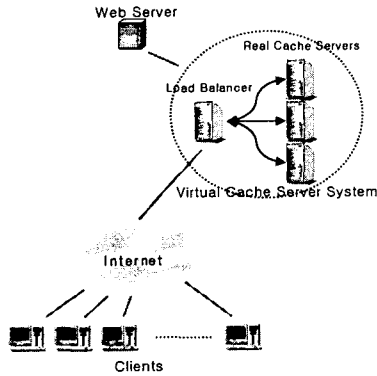


그림 1. 시스템 구성도

표 1. 실험에 사용된 시스템 사양

용도	CPU	RAM	Network
Web Server	400Mhz	196Mb	100M
Load Balancer	866Mhz	128Mb	100M
Real Cache Server	866Mhz	128Mb	100M
Client	1.7Ghz	512Mb	1G/100M

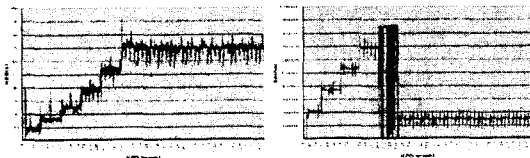
표 2. 측정에 사용된 동영상 정보

파일 크기	대역폭	플레이 시간
25Mb	703Kbps	5분

#### 4. 성능평가

##### 4.1. 단일 캐시 서버 시스템의 성능평가

가상 캐시 서버 시스템과의 성능 비교를 위해 우선 캐시 서버를 한 대만 두는 웹 캐싱 시스템을 구축하여 위의 실험 방식과 동일한 방법으로 동일한 사양의 캐시 서버에 걸리는 부하를 측정하였다. 그림 2는 캐시 서버의 CPU 사용률과 네트워크 사용률을 측정한 결과이다. 측정 결과 클라이언트의 수가 26대까지는 원활한 네트워크 상태를 보이지만, 6대의 클라이언트 추가 후에는 네트워크 상태가 상당히 불안해 지면서 네트워크 사용률이 매우 저조하게 나타나는 것을 볼 수 있다.



(a) CPU 사용률 (b) 네트워크 사용률

그림 2. 웹 캐싱 시스템에서 캐시 서버의 CPU 및 네트워크 사용률

#### 4.2. 가상 캐시 서버 시스템의 성능평가

##### 4.2.1. CPU 사용률 분석

그림 3은 Direct Routing 방식으로 구성되고 LBLC 스케줄링 알고리즘을 사용했을 때, 각 노드의 CPU 사용률을 나타낸 것이다. (a)는 웹 서버, (b)는 로드밸런서, (c)~(e)는 각 실제 캐시 서버의 CPU 사용률을 나타내었다.

웹 서버의 경우 간혹 크게 사용률이 올라가는 것을 제외하고는 거의 CPU를 사용하지 않는 것을 볼 수 있다. 이때 CPU 사용률이 크게 증가하는 것은 캐시 서버에서 웹 서버에 멀티미디어 스트리밍 데이터를 요구할 때, 그 요청을 처리해 주기 때문이다. 로드밸런서를 보면 주기적으로 CPU 사용률이 피크를 나타낼 때가 있다. 이는 클라이언트의 요청이 있을 때 실제 캐시 서버를 부하 분배 스케줄링 알고리즘에 따라 분배할 때 나타나는 것으로 실험시에 10분마다 클라이언트의 수를 늘린 것과 관계가 있다. 실제 캐시 서버들의 실험시 CPU 사용률을 보면, 캐시 서버를 하나만 두었을 때보다 CPU 부하가 많이 감소했음을 볼 수 있다.

IP Tunneling과 NAT 구성 방식의 경우도 각 노드에 걸리는 CPU의 부하가 많이 개선되었다. IP Tunneling의 경우 Direct Routing과 거의 비슷한 CPU 사용률을 보인다. 그러나 NAT 구성 방식은 그림 4의 (a)에서 보듯이 로드밸런서의 CPU 사용률이 50%까지 증가함을 볼 수 있는데, 이는 NAT의 특징과 관계가 깊다. NAT는 클라이언트와 서버 사이에 오고가는 모든 패킷들의 IP 헤더 정보가 로드밸런서를 통해서 변경되어야 한다. 바로 이 과정에서 많은 부하가 걸리기 때문에 CPU 사용률이 다른 구성 방식보다 많이 나타난 것으로 보인다.

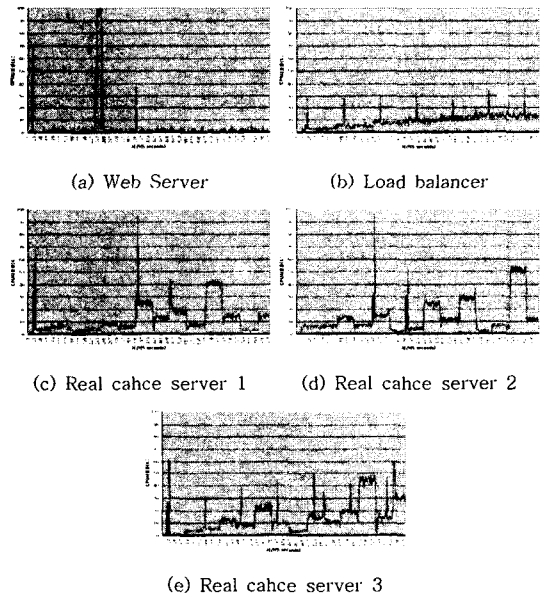


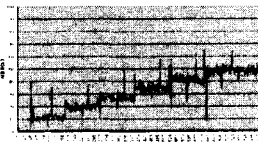
그림 3. Direct Routing 구성 방식과 LBLC 스케줄링 방식으로 구성된 가상 캐시 서버와 웹 서버의 CPU 사용률(%)

##### 4.2.2. 네트워크 사용률 분석

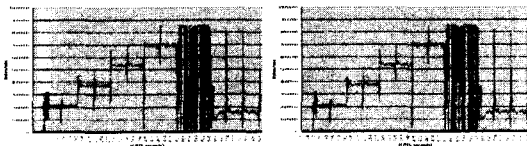
그림 5는 각 노드의 네트워크 사용률을 나타낸 것이다. 실제 각 노드들이 서비스를 하는지를 알아보기 위해 각 노드들이 보낸 패킷들을 나타내었다. 웹 서버에서 갑자기 패킷을 많이 보내는 경우가 있는데, 이는 로드밸런서에 의해서 선택된 실제

캐시 서버들이 클라이언트에서 요청된 멀티미디어 스트리밍 데이터를 웹 서버에 요청해서 캐싱 서비스를 하기 위한 과정에서 나타나는 것이다. 로드밸런서에서는 캐싱 서비스를 하지 않기 때문에 보낸 패킷이 거의 없음을 볼 수 있으며, 각 실제 캐시 서버들도 안정적으로 웹 캐싱 서비스를 하는 것을 볼 수 있다. 점차적으로 보내는 패킷들이 많아지는 것은 10분마다 클라이언트의 숫자를 늘리는 것과 연관이 있는 것으로 캐시 서버가 한 대일 경우보다 더 많은 수인 38대의 클라이언트의 요청을 안정적으로 서비스 할 수 있었다.

IP Tunneling 방식도 네트워크에 걸리는 부하가 분산되기 때문에 캐시 서버를 한 대만 사용했을 경우보다 더 많은 클라이언트의 요청을 처리할 수 있었다. 그러나 NAT의 경우 그림 4의 (b), (c)에서처럼 네트워크 사용률이 매우 불안정하게 나타나다가 매우 저조해 지는 것을 볼 수가 있다. 이것은 바로 NAT의 구성 방식에서 원인을 찾을 수 있다. NAT는 모든 패킷들이 로드밸런서를 통하기 때문에 로드밸런서에 걸리는 네트워크 부하가 큰 데, 이 부하가 로드밸런서가 처리할 수 있는 패킷의 한계를 넘기 때문에 제대로 서비스를 할 수 없는 것이다.



(a) Load balancer의 CPU 사용률(%)



(b) Load balancer에서 보낸 패킷(byte/second) (c) Load balancer에서 받은 패킷(byte/second)

그림 4. NAT 구성 방식과 LBLC 스케줄링 방식으로 구성된 가상 캐시 서버의 로드밸런서의 CPU와 네트워크 사용률

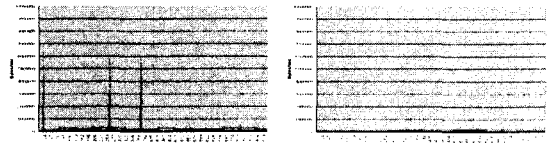
### 5. 결론 및 추후 연구 방향

본 논문에서는 인터넷 방송 등에서 사용되는 멀티미디어 스트리밍 데이터의 원활한 캐싱 서비스를 위하여 리눅스 가상 서버를 이용한 다양한 가상 캐시 서버 시스템을 구축하고 그 성능을 평가하였다.

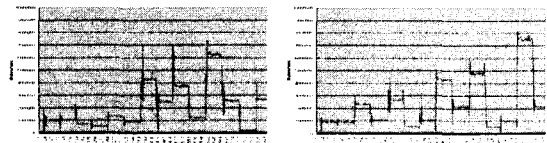
가상 캐시 서버 시스템을 사용함으로써 기존의 캐시 서버 한 대에 집중되는 부하를 적절히 분배하는 과정을 통해 더 많은 클라이언트들의 요청을 처리할 수 있었다. 그러나 NAT 구성 방식의 경우 모든 패킷들이 로드밸런서를 거쳐야 한다는 특징으로 인해서 패킷들의 부하 분산 효과를 거의 얻지 못 하였다. 이는 고용량 멀티미디어 스트리밍 데이터의 캐싱 서비스의 구축에 있어서 NAT 구성 방식은 알맞지 않다는 것을 의미한다.

본 실험에서는 여러 대의 클라이언트를 주기적으로 늘리면서 하나의 동일한 파일에 대한 캐싱 서비스에 대해서 제한적인 성능평가를 하였다. 그러나 실제적인 캐싱 시스템의 경우 다양한 멀티미디어 스트리밍 데이터를 다양한 클라이언트들에게 서비스를 해야 하기 때문에, 추후 보다 다양한 멀티미디어 스트

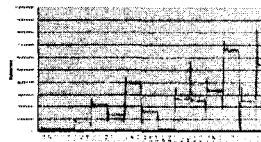
리밍 데이터 및 클라이언트들을 대상으로 하는 성능 측정을 실시할 필요가 있다. 이 경우 가변 요인들에 대한 파악과 해결 방안에 대한 연구도 병행되어야 할 것이다.



(a) 웹 서버에서 보낸 패킷(byte/second) (b) Load balancer에서 보낸 패킷(byte/second)



(c) Real cache server 1에서 보낸 패킷(byte/second) (d) Real cache server 2에서 보낸 패킷(byte/second)



(e) Real cache server 3에서 보낸 패킷(byte/second)

그림 5. Direct Routing 구성 방식과 LBLC 스케줄링 방식으로 구성된 가상 캐시 서버와 웹 서버의 네트워크 사용률

### 참고문헌

- [1] Barish, G.; Obraczke, K. "World Wide Web caching: trends and techniques", IEEE Communications Magazine, Vol.38 Issue.5, pp.1045 -1052, 2000
- [2] 이영석, 윤창일, 이용우, "멀티미디어 콘텐츠 전달 및 분산 시스템의 성능평가", 한국정보처리학회 봄 학술발표논문집, Vol.9 No.1, pp.1277-1280, 2002
- [3] 리눅스 가상 서버 <http://www.linuxvirtualserver.org>
- [4] <http://www.linuxvirtualserver.org/docs/scheduling.htm>
- [5] 박동국, 이용우, "CDD 시스템에서의 리눅스 가상 서버의 성능에 관한 연구", 한국정보과학회 봄 학술발표논문집, Vol.29 No.1, pp.613-615, 2002
- [6] squid 프로젝트 <http://www.squid-cache.org>