

확장성 있는 클러스터링 기반의 무선 인터넷 프록시 서버

곽후근[○] 한경식[○] 정규식[○]
 송실대학교 정보통신전자공학부
 {gobarian, ihanks[○], kchung}@q.ssu.ac.kr

A Scalable Clustering based Wireless Internet Proxy Server

Hukeun Kwak Kyungsik Han[○] Kyusik Chung
 School of Electronics Engineering, Soongsil University

요 약

TranSend는 클러스터링 기반의 무선 프록시 서버로 제안된 것이나 시스템적인(Systematic) 방법으로 확장성을 보장하지 못하고 불필요한 모듈간의 통신구조로 인해 복잡하다는 단점을 가진다. 기존 연구에서 시스템적인 방법으로 확장성을 보장하고 모듈간의 통신 구조를 단순화(Simplification)한 CD-A라는 구조를 제안하였으나 이 역시 프록시 서버가 증가하면 클러스터링을 위해 사용된 LVS-NAT(Network Address Translation) 방식으로 인해 LVS(Linux Virtual Server)가 병목이 되는 단점을 가진다. 이에 본 논문에서는 프록시 서버가 증가해도 확장성을 보장하는 LVS-DR(Direct Routing) 방식을 사용한 클러스터링 기반의 무선 인터넷 프록시 서버를 제안한다. 16대의 컴퓨터를 사용하여 실험을 수행하였고 실험 결과 클러스터링을 위해 사용된 LVS-NAT 방식에 비해 LVS-DR 방식이 평균 30.85%, 최고 78.50%의 성능 향상을 보였다.

는 방법으로 동작한다.

1. 서 론

현재 정보화 사회는 인터넷과 무선 이동 통신이라는 커다란 두 개의 축에 의해 선도되고 있으며 인터넷과 무선 이동 통신은 서로 조화를 이루면서 급속도로 성장하고 있다. 무선 인터넷이란 언제 어디서나 인터넷에 접속하여 다양한 정보검색과 전자상거래 등을 하는 것으로 기존 인터넷 환경의 공간적 제약을 극복하여 서비스하는 방식이며, 좀 더 넓은 의미로는 무선 랜등 고정 무선 인터넷 서비스를 포함하여 무선을 통해 인터넷에 접속하는 것을 뜻한다. 급격한 인터넷 사용의 증가와 핸드폰, 노트북, PDA 등 휴대용 단말기의 사용 급증으로 사람들이 이동하면서 언제든지 인터넷망에 접속하여 정보를 얻고, 통신할 수 있는 무선 인터넷은 더욱 더 새로운 사회적 관심거리가 되고 있다.

기존 유선 인터넷과 달리 무선 인터넷은 낮은 대역폭, 빈번하게 접속이 끊기는 현상, 단말기 내의 낮은 컴퓨팅 파워 및 작은 화면, 사용자의 이동성, 네트워크 프로토콜, 보안 등의 특성에 따른 많은 제약점들을 갖고 있다. 또한 무선 인터넷 서버는 급증하는 사용자에 따른 대용량 트래픽을 처리할 수 있도록 확장성이 있어야 한다. 이에 위의 문제를 캐싱(Caching)과 압축(Transcoding, Distillation)으로 해결하는 방법으로 무선 프록시 서버[1-3]를 사용한다.

1.1. TranSend[2]

그림 1은 TranSend 프록시 시스템의 전체적인 구조를 나타낸다. FE는 Client 요청에 대한 외부 인터페이스를 담당하며, User Profile DB는 사용자와 관련된 정보(Preference)를 저장한다. Cache는 Client의 요청을 처리하며, Worker는 데이터에 대한 압축을 수행한다. Manager는 Distiller를 관리하고, Graphical Monitor는 시스템 전체의 상태를 볼 수 있게 해준다.

Client 요청을 FE가 받고 Cache 서버에 요청하여 존재하면 해당 데이터를 받고, 존재하지 않으면 Cache 서버가 웹 서버로부터 요청하여 받아온다. FE는 그 데이터를 Worker(Distiller)에게 보내 압축을 요청하여 압축된 데이터를 Client에게 보내

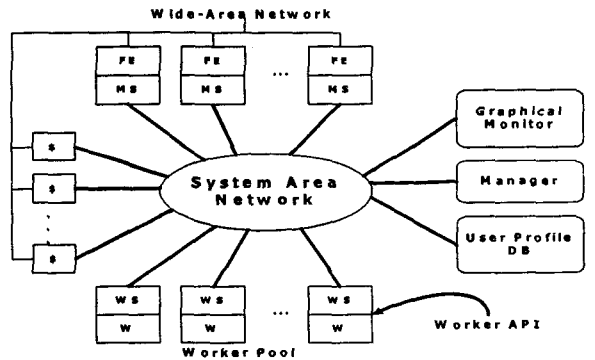


그림 1 TranSend 프록시 시스템 구조

1.2 CD-A[4]

CD-A 구조는 TranSend에 사용된 기본 모듈(FE, Cache, Distiller)에서 Distiller를 없애고 Cache에 압축(Distillation) 기능을 추가(이하 CD: Cache & Distiller)한 것이다. 그리고 이 모듈들(FE, Cache)을 하나의 호스트에 넣고 LVS(Linux Virtual Server)를 사용하여 부하 분산을 한 것(이하 CD-A: CD & All-in-one)이다. 그림 2는 CD-A 구조를 나타낸다.

CD-A 구조에서 Cache에 압축 기능을 추가한 이유는 전체 구조에서 Distiller를 제거하여 불필요하고 복잡한 통신 구조를 단순화하기 위해서이다. 그리고 모듈들을 하나의 호스트에 넣은 이유는 시스템적으로 확장하는 구조를 만들기 위해서이다. 즉, TranSend는 새로운 모듈을 추가시에 동작과정중의 병목을 찾아 그 모듈을 추가해야하는(No Systematic) 반면에, CD-A는 병목에 상관없이 새로운 호스트를 추가하면(Systematic) 그 호스트내에 모듈중에 필요한 모듈이 상대적으로 많이 사용된

다.

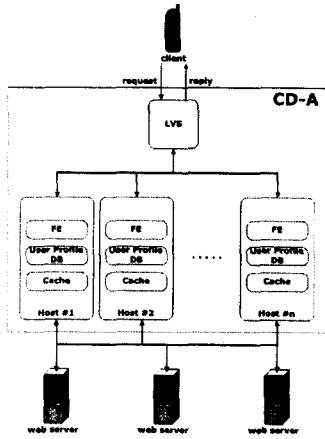


그림 2 CD-A 구조

1.3 접근 방식

TranSend 무선 프록시 서버 및 이의 개선 구조인 CD-A의 문제점을 정리하면 표 1과 같다.

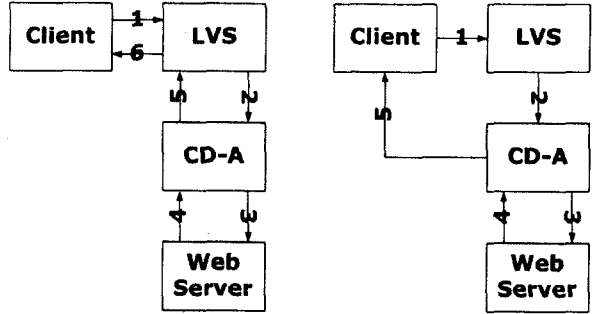
표 1 기존 구조의 문제점

기존 구조	문제점	설명
TranSend	확장성 (Scalability)	FE를 클러스터링 하기위해 사용된 LVS-NAT[5] 방식으로 인해 FE가 증가하면 LVS가 병목이 된다.
	복잡성 (Complexity)	TranSend는 FE, Cache, Distiller로 구성되어 FE를 중심으로 서로간에 통신 (Communication)을 하도록 구성되어 있다. 이러한 구조는 FE로 모든 통신이 편중되어 있고 Cache와 Distiller가 분리되어 있어 불필요한 통신을 하는 단점을 가진다.
CD-A	확장성 (Scalability)	CD-A Host를 클러스터링 하기위해 사용된 LVS-NAT 방식으로 인해 CD-A Host가 증가하면 LVS가 병목이 된다.

본 논문에서는 TranSend와 CD-A 구조의 단점인 LVS-NAT 방식 대신 LVS-DR[6] 방식을 사용하여 확장성 있는 클러스터링 기반의 무선 인터넷 프록시 구조를 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 기존 무선 프록시가 가지는 문제점을 해결하는 새로운 구조를 설명하고, 3장에서는 실험 및 결과를, 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 제안된 프록시 서버

그림 3은 LVS-NAT 방식과 LVS-DR 방식을 나타낸 것이고, 각 방식의 사용자 요청 처리 순서를 요약하면 표 2와 같다. LVS-NAT 방식은 클라이언트의 요청과 이에 대한 응답이 모두 LVS를 통과하게 된다. 즉, LVS는 프록시 서버를 증가하는 것에 비례하여 부하가 가중되어 결국 병목이 된다. 이에 비해 LVS-DR 방식은 클라이언트에 대한 응답을 프록시 서버가 직접함으로써 LVS의 부하를 최소화하여 전체 시스템의 확장성을 보장한다.



(a) LVS-NAT (b) LVS-DR
그림 3 LVS-NAT 방식과 LVS-DR 방식

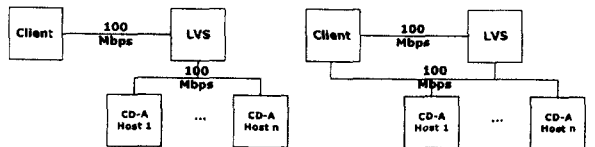
표 2 LVS-NAT와 LVS-DR 방식의 사용자 요청 처리 순서

Step	LVS-NAT	LVS-DR
1	Client가 LVS에게 데이터를 요청한다.	Client가 LVS에게 데이터를 요청한다.
2	LVS는 CD-A에게 데이터를 요청한다.	LVS는 CD-A에게 데이터를 요청한다.
3	CD-A는 Web Server에게 데이터를 요청한다.	CD-A는 Web Server에게 데이터를 요청한다.
4	Web Server는 CD-A로 데이터를 보낸다.	Web Server는 CD-A로 데이터를 보낸다.
5	CD-A는 LVS로 데이터를 보낸다.	CD-A는 Client에게 직접 데이터를 요청에 대한 응답을 한다.
6	LVS는 Client에게 데이터 요청에 대한 응답을 한다.	

3. 실험 및 토론

3.1 실험 환경

실험 환경은 Client 1대, LVS 1대, 동일 사양의 Host 16대로 구성되어 있고 부하 분산을 위하여 LVS라는 Load Balancer를 사용하였다. Client는 AB(Apache Bench), Cache는 Squid, Distiller는 JPEG-6b 라이브러리를 사용하였다. 클라이언트의 요청 개수는 약 200초 동안 프록시를 처리할 수 있는 최대 개수이고, 요청 이미지는 JPEG, 요청 크기는 300 bytes, 1 K, 10 K, 100 Kbytes, Variation(1~10 Kbytes)이다. 그림 4는 실험에 사용된 구조들의 구성도를 나타낸다.



(a) LVS-NAT 방식 (b) LVS-DR 방식
그림 4 실험에 사용된 구성도

3.2 실험 방법

표 3은 실험에 사용된 변수들을 정리한 것이고 각 구조의 실험 방법을 정리하면 표 4과 같다.

표 3 실험에 사용된 변수

사용자의 요청 개수	• 약 200초 동안 프록시가 처리할 수 있는 최대 개수
요청 이미지	• JPEG
요청 크기	• 300 bytes, 1 K, 10 K, 100 Kbytes, Variation
사용자 정보 (Preference)	• 이미지 Quality = 중간
웹 서버	• Cache 서버 자체에 동 (프록시내의 성능 평가에 초점을 맞춤)

표 4 실험 방법

step	설명
1	기본 CD-A 시스템을 구성한다. (Host 1)
2	AB(Apache Bench)를 이용하여 CD-A Host로 JPEG 이미지를 약 200초 동안 프록시가 처리할 수 있는 최대 개수로 요청한다.
3	초당 요청 개수 및 Host 안의 각 모듈의 CPU 점유율을 측정한다.
4	CD-A Host를 추가한다.
5	실험에 사용된 Host 수(16대)만큼 2~4 과정을 반복한다.

3.3 실험 결과

(1) LVS-NAT

그림 5는 이미지 크기를 다르게 요청했을 경우 호스트 개수에 따른 초당 요청수를 나타낸다.

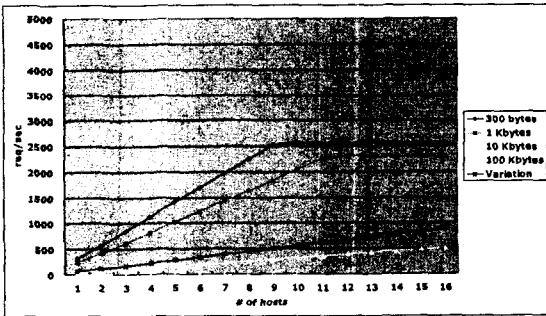


그림 5 호스트 개수에 따른 초당 요청수 (LVS-NAT)

(2) LVS-DR

그림 6는 이미지 크기를 다르게 요청했을 경우 호스트 개수에 따른 초당 요청수를 나타낸다.

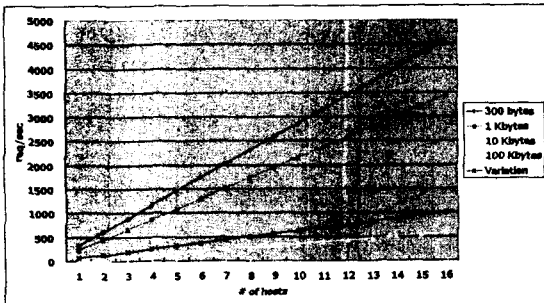


그림 6 호스트 개수에 따른 초당 요청수 (LVS-DR)

(3) LVS-NAT vs. LVS-DR

표 5는 16개의 호스트를 기준으로 LVS-NAT에 대한 LVS-DR의 성능 향상률을 나타낸 것이다. 평균 향상률은 30.85%이고, 최고 향상률은 300 bytes의 경우에 78.50%이다. 이미지 크기가 커짐에 따라 향상률이 낮아지는 이유는 이미지를 압축하는 시간이 상대적으로 길기 때문이다. 즉, LVS를 통과하는 패킷의 수가 적으므로 LVS에서 병목이 발생하지 않았기 때문이다.

표 5 LVS-DR의 성능 향상률 (16개의 호스트 기준)

%	300 bytes	1 Kbytes	10 Kbytes	100 Kbytes	Variation	Average
LVS-NAT	2567	2292	516	37.87	899	
LVS-DR	4582	3443	577	38.30	1012	
LVS-NAT vs. LVS-DR	78.50	50.22	11.82	1.14	12.57	30.85

4. 결론

본 논문에서는 무선 인터넷의 근본적인 문제점 중 일부를 해결할 수 있도록 제안된 TranSend 및 이의 개선 구조인 CD-A 프록시 서버의 문제점을 클러스터링을 위해 사용된 LVS-NAT 방식 관점에서 지적하고, 이의 개선 구조인 LVS-DR 방식을 제안하였다. 실험을 통해 제안된 구조가 성능 향상에 기여했음을 확인하였다.

향후 연구 방향을 요약하면 다음과 같다.

- (1) 부하 분산 시에 라운드 로빈 대신에 호스트의 처리 능력에 따른 부하 분산 방식으로 변환 : 부하 분산에서 라운드 로빈을 적용할 경우 부하가 동일하게 분산됨으로 최종 결과는 처리 능력이 가장 적은 호스트에 집중되는 단점을 가진다.
- (2) 다양한 레벨의 압축 및 저장 : 이미지 요청의 경우 요청 크기에 비례하여 압축 시간이 증가하므로 사용자에게 요청 이미지를 전송한 후 사용자가 지정할 수 있는 다양한 압축 레벨로 모두 압축하여 저장한다. 이렇게 미리 압축본을 저장해놓으면 사용자가 해당 이미지를 요청할 때 압축을 하지 않고 저장된 압축 이미지를 보냄으로서 사용자 요청에 대한 응답 시간을 최대한 줄일 수 있다.

참고문헌

[1] B. Housel, G. Samaras and D. Lindquist, "WebExpress: A client/intercept based system for optimizing web browsing in a wireless environment", Mobile Networks and Applications, ACM, pp. 419-431, 1998.
 [2] A. Fox, "A Framework For Separating Server Scalability and Availability From Internet Application Functionality", Ph. D. dissertation, U. C. Berkeley, 1998.
 [3] A. Maheshwari, A. Sharma, K. Ramamritham and P. Shenoy, "TranSquid: transcoding and caching proxy for heterogeneous e-commerce environments", Proceedings of 12th International Workshop on RIDE-2EC, IEEE, pp. 50-59, 2002.
 [4] 곽후근, 한경식, 정규식, "클러스터링 기반의 무선 인터넷 프록시 서버 성능 개선", 제31회 춘계학술발표회, 한국정보과학회, 2004.
 [5] Virtual Server via NAT, <http://www.linuxvirtualserver.org/VS-NAT.html>
 [6] Virtual Server via DR, <http://www.linuxvirtualserver.org/VS-DR.html>