

리눅스 클러스터 서버 상에서 RTSP 기반의 VOD 스트리밍 시스템의 설계 및 구현

김영진⁰ 홍기호 최황규
강원대학교 컴퓨터·정보통신공학과
{yjkim, red}@mirae.kangwon.ac.kr hkchoi@kangwon.ac.kr

Design and Implementation of the RTSP Based VOD Streaming System On Linux Cluster Servers

Young-Jin Kim⁰ Ki-Ho Hong Hwang-Kyu Choi
Department of Computer, Information and Communication Engineering, Kangwon National University

요 약

리눅스 클러스터 서버는 고성능 저가의 PC와 공개 소프트웨어인 리눅스를 활용함으로써 고가의 단일 서버에 비하여 확장성과 가용성이 높은 고성능 서버의 실현이 가능하다. 이에 따라 리눅스 클러스터 서버를 대용량 멀티미디어 스트리밍 서비스 등에 활용하고자 하는 연구가 활발히 이루어지고 있다. 본 논문에서는 리눅스 클러스터 상에서 스트리밍 표준 프로토콜인 RTSP/RTP를 기반으로 현재 가장 널리 활용되고 있는 Microsoft사의 ASF 파일 형식을 지원하는 스트리밍 서버와 클라이언트 플레이어를 설계 구현한다. 구현된 시스템은 Windows 시스템에 비하여 비교적 저가에 현재 인터넷상에서 스트리밍 서비스의 표준으로 자리잡고 있는 ASF 파일의 스트리밍 실현이 가능하다.

1. 서론

최근 통신 기술의 급속한 발전으로 통신망 서비스 사업이 다각화됨에 따라 초고속 통신망의 보급이 가정으로까지 빠르게 확대되고 있다. 이에 따라 인터넷을 기반으로 하는 웹 서비스 업체들이 기하 급수적으로 증가하고 있고, 인터넷을 기반으로 하는 서비스도 다양하게 변화하고 있으며, 이를 이용하려는 서비스 가입자들도 폭발적으로 증가하고 있는 추세이다. 인터넷 서비스 제공 업체들이 많은 가입자들에게 동시에 안정된 서비스를 제공하려면 비교적 고가의 고성능의 대용량 서버를 설치 운영하여야 한다.

비교적 저가의 고성능 PC 또는 워크스테이션을 빠른 네트워크로 연결한 클러스터 시스템은 강 결합의 단일 서버와 비교하여 확장성과 신뢰성이 높고, 가격 대 성능 면에서 우수한 것으로 나타났다. 이중 리눅스 기반의 PC 클러스터는 과학계산용의 슈퍼컴퓨팅 분야에 많이 활용될 뿐만 아니라 웹 서버, 게임 서버, VOD 서버 등 고부하와 고가용성 응용분야에 활용하고자 하는 많은 연구가 이루어지고 있다[1,2,3].

한편, 최근 인터넷상에서 멀티미디어 스트리밍 서비스는 인터넷 방송의 확산 등으로 그 활용범위가 점점 넓어지고 있다. 그러나 현재 스트리밍 서비스를 위한 몇몇 상용 제품들이 활용되고 있지만 서비스의 비용이 높거나 특정 미디어 타입만을 지원한다는 단점을 갖는다.

본 논문에서는 리눅스 클러스터 상에서 스트리밍 표준 프로토콜인 RTSP/RTP를 기반으로 현재 가장 널리 활용되고 있는 Microsoft사의 ASF 파일 형식을 지원하는 스트리밍 서버와 클라이언트 플레이어를 설계 구현한다. 구현된 시스템은 Windows 시스템에 비하여 비교적 저가에 현재 인터넷상에서 스트리밍 서비스의 표준으로 자리잡고 있는 ASF 파일의 스트리밍 실현이 가능하다.

본 논문은 2장에서 리눅스 가상 서버 기반의 리눅스 클러스터에 대하여 기술하고, 3장에서는 스트리밍 서비스의 종류와 폭 넓게 쓰이고 있는 상용의 스트리밍 서버에 대해서 기술

한다. 4장에서는 클러스터형 스트리밍 서비스에 대한 부하 분산 방법과 서버의 설계 및 구현에 대한 결과를 보인다.

2. 리눅스 클러스터

리눅스 가상 서버는 리눅스 운영체제를 기반으로 고속 네트워크에 연결된 서버들의 클러스터로 성능이 뛰어나고 가용성이 높은 서버를 쉽게 구축할 수 있다[1].

일반적인 가상 서버의 구조로 그림 1과 같이 하나의 부하 분산기와 여러 대의 실제 서버를 가진다. 실제 서버들은 동일한 서비스를 운영하도록 구성되며, 서비스 내용은 각 서버의 지역 디스크에 복제되거나, 분산 파일 시스템으로 공유됨으로써 제공된다.

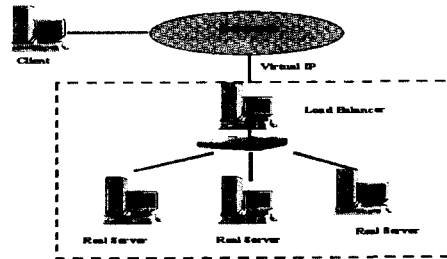


그림 1. 일반적인 클러스터 시스템 구조

부하 분산 서버는 실제 서버의 서비스 내용에 대하여 클라이언트의 요청이 있을 때마다, 적절한 스케줄링 방법(예, Round Robin)에 의해서 상호 연결되어 있는 실제 서버에 클라이언트의 요청을 발송한다. 점선의 직사각형으로 표시된 클러스터의 실제 서버들은 빠른 LAN/WAN으로 연결되며, 클라이언트에게는 하나의 단일 이미지의 시스템으로 보이도록 투명하게 설정된다. 즉, 실제 서버들의 외부 연결부로서의 부하

분산 서버는 클라이언트의 요구들을 각각 다른 노드에게 할당하고 클러스터의 병렬 서비스를 하나의 IP 주소상의 가상 서버로 보이도록 한다. 높은 가용성은 노드 또는 데몬의 장애나 시스템의 구조 변화를 적절히 감지하여, 노드를 제거하거나 대체하여 이루어진다.

3. 스트리밍 서버

이장에서는 현재 널리 사용되고 있는 인터넷 기반의 상용 스트리밍 시스템인 RealNetworks의 RealSystem, Microsoft의 WMT, Darwin Streaming Server 등을 비교 분석한다.

1) RealNetworks의 RealSystem

세계 최초의 스트리밍 솔루션이며 한때 전세계 시장 점유율 70% 이상이었다. 또 최근 버전에서는 브로드캐스팅 기능을 지원하여 웹 브라우저 없이도 디지털 인터넷 방송, 웹 프리젠테이션, 애니메이션, 스트리밍 비디오, 스트리밍 오디오를 모두 감상할 수 있다. 그러나 고가의 서버 프로그램을 구입해야 하고 동시 접속자수가 제한된다는 단점이 있다.

2) Microsoft의 Windows Media Technologies

Microsoft의 스트리밍 솔루션이며 최근들어 Real Network사의 RealSystem의 시장 점유율을 앞지르고 있다. 폭 넓은 스트림 대역폭, 여러 가지 플레이 백 미디어 포맷 지원하며 비디오 재생시, 화면의 사이즈가 픽셀 단위로 자유롭게 조절 가능하다. 또 고대역폭의 라이브 스트리밍을 지원하며 강력한 멀티캐스트 기능을 가지고 있다. 그러나 서버 운영 플랫폼이 MS Windows 시스템으로 한정되어 있다는 단점이 있다.

3) Darwin Streaming Server

비디오 스트리밍 서버로서는 최초로 오픈 소스 애플리케이션이라는 장점을 갖으며, 이런 오픈 소스 정책으로 인하여 사용자수가 급격히 증가하고 있다. 또 윈도우와 애플 OS를 모두 지원한다. 또 업계 표준 스트리밍 프로토콜인 RTP/RTSP를 지원하며, 음성과 이미지의 고품질로 인기가 높다[6].

4. 클러스터형 스트리밍 서버 설계 및 구현

이 장에서는 먼저 빠른 패킷 전송과 부하 분산기의 병목 현상을 제거하여 성능을 높이고 확장이 용이하도록 하기 위하여 하나의 가상 IP를 공유하는 다이렉트 라우팅에 의한 부하 분산 서버 설계 방법을 보인다. 그리고 이를 바탕으로 하여 스트리밍 서비스를 위한 서버 설계 방법과 클라이언트 재생기의 구현 결과를 기술한다.

4.1 부하 분산 서버 설계

그림 2에서와 같이 클러스터 서버는 리눅스 가상 서버의 다이렉트 라우팅을 사용하여 리눅스 상에 스트리밍 서비스를 위한 실제 서버로 구성된다.

리눅스 가상 서버의 부하 분산 방식 중, 네트워크 주소 번역(NAT)에 의한 방법은 요청의 작업 할당과 응답 패킷을 모두 부하 분산 서버에서 처리하기 때문에, 서버가 증가하는 경우, 부하 분산 서버가 병목 구간이 된다. IP 터널링과 다이렉트 라우팅에 의한 부하 분산 방식은 실제 서버가 클라이언트에게 직접 응답 패킷을 전송하므로, NAT에 의한 방법 부하 분산기의 병목 구간에 대한 문제를 해결할 수 있다. 그리고 다이렉트 라우팅은 터널링에 대한 오버헤드가 없으므로 보다 확장성이 좋은 클러스터 시스템을 구축할 수 있다.

본 논문에서는 그림 2와 같이 스트리밍 서버가 수행되는 펜티엄 II 233MHz, 리눅스 환경의 서버 세대와 하나의 펜티엄 II 400MHz의 다이렉트 라우팅 방식의 부하 분산 서버로

클러스터형 스트리밍 서버를 구현하였다.

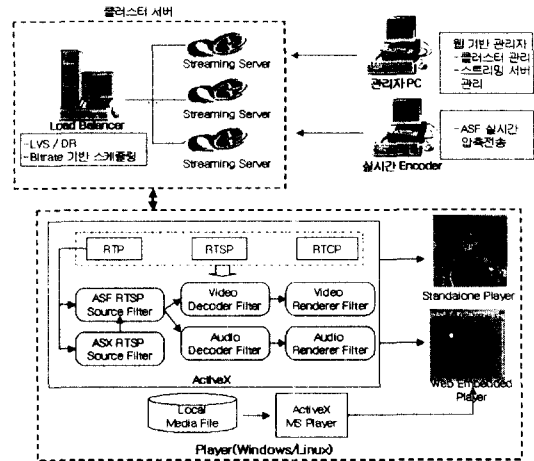


그림 2. 클러스터형 스트리밍 서버 구조

4.2 스트리밍 서버 설계

스트리밍 서버는 리눅스 상에서 구현되었으며, Microsoft의 ASF 파일 형식을 지원하며 RTSP/RTP 프로토콜을 사용하여 구현하였다. ASF(Advanced Streaming Format)는 윈도우즈 계열의 스트리밍 서비스를 위하여 Microsoft사에 의해 독자적으로 개발된 스트리밍 전용 포맷이다. 따라서 현재는 Windows NT, Windows 2000같은 운영체제가 있어야만 서비스가 가능하다. 하지만 본 논문에서는 리눅스 상에서 ASF 파일 형식을 스트리밍 할 수 있는 서버를 자체적으로 개발하였다. 따라서 WMT를 이용하는 스트리밍 서비스를 할 경우 요구되는 높은 비용(운영체제의 구입비용)을 대폭 절감할 수 있도록 하였다. 또 기존의 윈도 계열의 스트리밍 서버는 자체 프로토콜을 사용하였지만, 본 연구에서는 표준으로 자리잡은 RTP/RTSP를 사용하여 범용성을 높였다.

1) RTSP 메소드

RTSP는 제어와 실시간 전송을 위해서 RTP의 상위 계층으로 설계되었다. 따라서 RTP가 수정되거나 기능이 추가되어도 RTSP에서 계속적으로 사용할 수 있다는 장점이 있다. RTSP는 기능적으로 HTTP와 매우 유사하다. 구문과 작동방법이 유사하고 대부분의 경우 HTTP확장은 RTSP에 추가된다. RTSP는 네트워크상에서 아래와 같은 메소드들을 정의하며, 이것은 플레이어가 서버에게 특정 역할을 요청하는 Remote-Controller의 기능을 가능하게 한다. 따라서 RTSP에서 제공되는 메소드를 이용하여 다양한 사용자 피드백이 가능하다.

표 1. RTSP Method

METHOD	기 능
SETUP	세션 설정을 요구하며, 서버는 필요한 리소스를 할당
DESCRIBE	미디어에 대한 description 정보를 요구
PLAY	스트리밍 서비스의 시작을 요구
PAUSE	스트리밍 서비스의 일시 중단을 요구
TEARDOWN	스트리밍 서비스의 종료와 세션 종료를 요구

2) RTP의 이용

RTP는 스트리밍이 이루어지는 네트워크 상의 실제 패킷 단위이며, 간단한 헤더와 미디어 데이터로 이루어지며 UDP를 이용하여 전달된다. RTP 패킷은 12Byte의 헤더와 Payload로 구성된다. Payload는 또한 encoding-sepecific layer에 의해 싸여진다. 본 논문에서는 RTP의 Payload 부분에 ASF 데이터를 넣어 RTP 패킷을 구성하였다. 구성된 패킷 정보는 그림 3과 같다. 그림과 같이 패킷을 구성함으로써 ASF 파일을 RTP 형식으로 전송이 가능하게 된다. 패킷을 받은 클라이언트 측에서는 RTP 헤더를 검사하여 시간 정보 등을 추출하여 재생에 이용하게 된다. 이런 일련의 작업은 RTP Service Thread에서 처리된다.

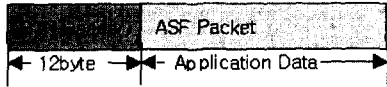


그림 3. RTP 패킷 구조

3) 서버 구조

그림 4는 스트리밍 서버의 개략적인 구조를 나타낸다. TCP Streaming Server는 클라이언트의 접속을 기다리는 모듈이다. 클라이언트가 접속하면 TCP Streaming Server는 클라이언트와 연결을 설정한 후 클라이언트의 요청을 RTSP Parser에게 전달한다. RTSP Parser는 클라이언트가 보내온 RTSP 메소드들을 판단하며, 분석된 내용에 맞는 응답을 클라이언트에 보내고 세션을 설정한다. 세션이 설정되면, Session Manager가 클라이언트와의 고유 세션을 관리하게 되며, 실제 비디오/오디오 데이터를 RTP Service Thread 모듈을 통해 클라이언트에 전달한다. RTP Service Thread는 서버에 데이터를 실제로 클라이언트에 전달하는 모듈이며, RTP 프로토콜을 이용하여 데이터를 전송하게 된다.

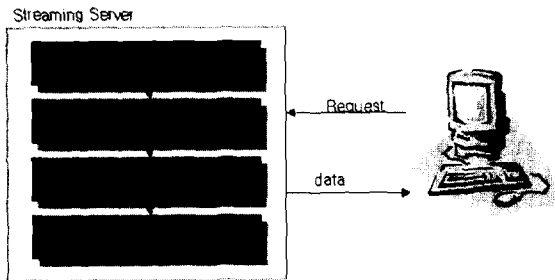


그림 4. 스트리밍 서버 구조

4.3 클라이언트 플레이어 설계

클라이언트 플레이어의 설계는 DirectShow의 Filter와 ActiveX Controller를 기본으로 설계되었으며, 서버로부터 들어오는 RTP 패킷 정보를 분석하여 ASF 패킷을 분리해내고, 분리된 ASF 패킷을 다시 오디오 데이터와 비디오 데이터로 분리하는 ASF/ASX RTSP Source Filter와 소스 필터로부터 출력된 압축된 비디오/오디오 데이터를 압축 해제하여 Video/Audio Render에 전달하는 Video/Audio Decoder Filter, 그리고 Video/Audio Renderer Filter는 Video/Audio Decoder로부터 압축이 풀린 비디오/오디오 데이터를 렌더링하여 실제로 사용자에게 보여준다. 또한 RTSP를 담당하는 ActiveX Controller가 클라이언트 재생기 내의 모든 모듈을 통합 관리한다. 클라이언트 플레이어가 시작되면 먼저 콘트롤

러가 시작되며, 사용자가 스트리밍 할 파일을 선택하면 콘트롤러가 서버와 통신하여 세션을 설정한 후, Source Filter, Video/Audio Decoder, Video/Audio Renderer를 메모리에 로딩하도록 구현하였다. 그림 5는 클라이언트 플레이어의 모습이다.



그림 5. 클라이언트 재생기 실행 화면

5. 결론

리눅스 클러스터 서버는 고성능 저가의 PC와 공개 소프트웨어인 리눅스를 활용함으로써 고가의 단일 서버에 비하여 확장성과 가용성이 높은 고성능 서버의 실현이 가능하다. 이에 따라 리눅스 클러스터 서버를 대용량 멀티미디어 스트리밍 서비스 등에 활용하고자 하는 연구가 활발히 이루어지고 있다.

본 논문에서는 리눅스 클러스터 상에서 스트리밍 표준 프로토콜인 RTSP/RTP를 기반으로 현재 가장 널리 활용되고 있는 Microsoft사의 ASF 파일 형식을 지원하는 스트리밍 서버와 클라이언트 플레이어를 설계 구현하였다. 구현된 시스템은 Windows 시스템에 비하여 비교적 저가에 현재 인터넷상에서 스트리밍 서비스의 표준으로 자리잡고 있는 ASF 파일의 스트리밍 실현이 가능하다.

참 고 문 헌

- [1] W. Zhang, "Linux Virtual Server for Scalable Network Services", Linux virtual server project document.
- [2] 권세오, 김상식, 김동승, "리눅스 클러스터형 웹서버 설계", 정보과학회지, vol. 18, 2000.
- [3] 최재영, 최종명, 김은희, 김민석, "고가용성 리눅스", 정보처리학회지, vol. 6, 1999.
- [4] 서대화, "리눅스 클러스터 환경에서의 분산 파일 시스템", 정보처리학회지, vol. 6, 1999.
- [5] W. Zhang and et al., "Linux virtual server project, <http://www.linuxvirtualserver.org>, 1998.
- [6] Darwin Streaming Server, <http://www.publicsourceapple.com/projects/streaming>.