

리눅스 클러스터 서버 상에서 RTSP기반의 실시간 및 VOD 스트리밍 시스템 개발

김종근^o 최황규

강원대학교 컴퓨터·정보통신공학과

godofdream@mail.kangwon.ac.kr^o, hkchoi@kangwon.ac.kr

Implementation of Real Time and VOD Streaming System Based on Linux Cluster Server

JongKeun-Kim^o HwangKyu-Choi

Dept. of Computer, Information and Communication Engineering, Kangwon National
University

요 약

최근 몇 년 동안 컴퓨터 사용자들의 멀티미디어 데이터에 대한 요구가 빠르게 증가하고 있다. 멀티미디어 데이터를 서비스하기 위해서는 대용량의 저장장치가 필요하며 연속재생이 가능하여야 한다. 이와 관련하여 데이터 압축기술과 스트리밍 기술과 고성능 PC를 고속의 네트워크로 연결하는 클러스터 기술에 대한 연구가 진행되고 있다. 클러스터 시스템은 비교적 저렴한 가격의 고성능 시스템으로 고가의 단일서버에 비해 확장성과 가격대 성능면에서 유리하다.

이에 본 연구에서는 리눅스 클러스터 기반의 오디오/비디오 실시간 및 VOD 스트리밍 서버와 일반적인 사용자에게 친숙한 윈도우 환경의 플레이어를 구현하였다. 또한 본 연구에서 구현한 시스템에서는 기존에 윈도우 NT계열에서만 스트리밍이 가능했던 ASF포맷의 멀티미디어 데이터를 리눅스 서버에서도 스트리밍 서비스가 가능하도록 하였다.

1. 서 론

현재 인터넷의 급속한 보급으로 인하여 컴퓨터 사용자는 빠르게 증가하고 있다. 또한 네트워크 기술의 발전과 압축 및 전송기술의 비약적인 발전은 텍스트 기반 데이터의 전송뿐만 아니라 화상회의 및 인터넷 폰, 실시간 오디오-비디오, VOD 등의 높은 대역폭을 요구하는 멀티미디어 데이터의 전송을 가능하게 하였다. 그러나 이러한 음성/화상 등의 멀티미디어 데이터에 대한 요구는 실시간 검색을 요구하며 대용량이라는 특징을 가지므로 기존의 기술을 이용하는 것은 상당한 무리가 따른다. 이러한 문제 해결을 위해 다양한 데이터 압축기술과 스트리밍에 대한 많은 연구가 연구되고 있다.

또한 컴퓨터 사용자의 급격한 증가로 인하여 빠르게 늘어나는 사용자의 요청을 처리하기 위해 클러스터 서버 구조가 제안되었다. 클러스터 서버는 여러 대의 PC나 워크스테이션을 고속의 네트워크로 연결한 구조이다. 이러한 클러스터 서버는 가용성이 뛰어나고 확장이 용이하다는 장점을 가지며 실제 구현되어 그 성능을 인정받고 있다.

이에 본 논문에서는 실시간 스트리밍의 표준 프로토콜인 RTP/RTSP을 통한 스트리밍 서버 및 클라이언트를 구현한다. 현재 MPEG과 ASF 포맷의 파일이 스트리밍에 널리 사용되고 있다. 특히, ASF 포맷은 널리 사용되고 있지만 MS사의 윈도우즈 서버에서만 스트리밍을 제공할 수 있다. 본 논문에서는 MPEG과 MOV 포맷의 스트리밍을 지원하는 애플사의 다윈 스트리밍 서버에 ASF 파일 포맷을 위한 스트리밍 모듈을 추가하고, 리눅스 환경의 클러스터 서버를 구성하였다. 클라이언트는 윈도우 기반으로 DirectShow를 이용하여 MPEG과 ASF 포맷의 VOD 기능에 LIVE 기능을 추가하였다.

본 논문의 구성은 2장에서는 관련 연구로서 RTP/RTSP, ASF, 다윈

스트리밍 서버 및 클러스터 구조 등에 대해 살펴보고 3장에서는 서버의 구성에 대해 설명한다. 그리고 4장에서는 클라이언트(플레이어)의 구성을 알아본다. 마지막으로 5장에서 결론 및 향후 발전 방향에 대해 설명하고 마치도록 한다.

2. 관련 연구

2.1 RTP/RTSP

일반적인 엄격한 타이밍 요구조건을 갖는 TCP기반의 HTTP, FTP와는 달리 RTP는 주로 미디어 스트림에 대한 전송을 목적으로 설계되었다. RTP는 실시간 멀티미디어 전송을 목적으로 만들어졌기 때문에 최소의 오버헤드 및 프리젠테이션의 데이터 품질에 대한 모니터링 능력이 없다[1]. 또한 session으로부터 participant 간의 추가/삭제에 대한 모니터링 기능역시 없다. 이러한 단점을 극복하기 위해서 RTCP 프로토콜이 개발되었고, RTCP는 프리젠테이션의 데이터 품질과 함께 RTP session에 대한 모니터링 기능을 제공한다[1].

RTSP는 네트워크 상에서 멀티미디어 데이터 스트리밍 방법(데이터의 전송 초기화 및 제어)에 대한 표준안으로써 RTP보다 상위 단계의 프로토콜로서 멀티미디어 스트림에 대한 command/control 기능을 제공한다. RTSP는 애플리케이션으로 하여금 서버로부터 데이터를 요청하게 하거나, 멀티미디어 회의에 참여하게 할 수 있다. RTSP는 비연결 지향 프로토콜이며, 각각의 스트림은 session ID에 의해 서로 구별이 된다. 그러나 RTSP의 control 요청은 연결이 보장된 TCP프로토콜로 전송된다. 실제로 control request는 멀티미디어 데이터를 포함하지 않기 때문에 안정된 연결지향의 프로토콜을 이용한다[1]. 다음의 [표 1]은 RTSP에 정의된 메소드들이다.

Method	설명
SETUP	Session 설정을 요구하며, Server는 필요한 Resource를 할당
DESCRIBE	Streaming 되는 media에 대한 description 정보를 요구
PLAY	Streaming Service의 시작을 요구
PAUSE	Streaming Service의 일시 중단을 요구
TEARDOWN	Streaming Service이 종료와 Session 종료 를 요구

표 1 RTSP Methods

2.2 ASF(Advanced Streaming Format)

Windows Media 콘텐츠를 정의하는 데이터 구조로서 네트워크를 통한 멀티미디어의 스트리밍에 적합한 구조이다. ASF 규정은 데이터를 스트리밍하거나 파일로 저장할 수 있도록 작은 패킷으로 정렬하도록 정하고 있다.

ASF 파일은 세 개의 레벨로 구성되어 있다. Header object는 파일 크기, 스트림의 개수와 같은 일반적인 파일에 대한 정보를 가지며, data object는 실제의 패킷타입 된 데이터를 가지고 있다. Index object는 꼭 필요한 정보는 아니지만 ASF 탐색이나 기타 타임스탬프에 대한 정보를 가진다[2].

2.3 다원 스트리밍 서버

다원 스트리밍 서버는 애플사에서 제작한 스트리밍 서버로서, 쿼터 임 타입의 멀티미디어 데이터를 서비스하도록 제작 되었다. 다원 스트리밍 서버는 스트리밍 프로토콜로서 RTP/RTSP를 사용하고 있다[3].

본 논문은 다원 스트리밍 서버에 ASF 포맷의 데이터에 대한 스트리밍 서비스가 가능하도록 모듈을 추가하였다.

2.4 클러스터 서버

클러스터 서버란 여러 대의 고성능 PC 또는 workstation을 고속의 네트워크로 결합하는 기술로서 확장성과 가격 대 성능이 우수한 서버 시스템을 제공하기 위해 연구되고 있다. 클라이언트가 서비스를 요청하면 front-end에서 로드밸런싱 알고리즘에 따라 real server에 각 클라이언트의 요청을 전송하고 요청을 받은 실제 서버는 해당 서비스를 처리하게 된다.

이러한 클러스터 시스템에서는 부하분산 기술에 따라 그 성능의 차이를 가져온다. 현재 연구된 부하분산 기술로서 주로 DSN 기반의 방법, 디스패처 기반의 방법, 서버 기반의 방법 등으로 나누어진다[4].

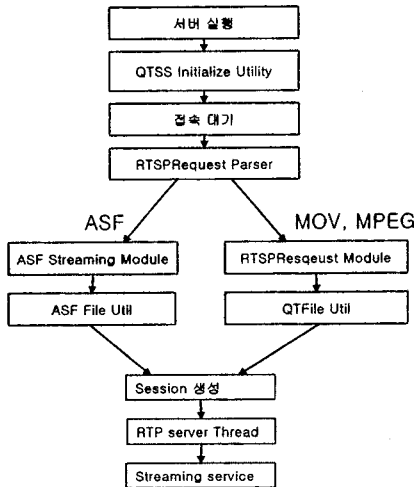


그림 1 서버의 구성

3. 서버의 구성

3.1 스트리밍 모듈

서버는 크게 두 가지 모듈로 구성된다. 즉, 기존의 다원 스트리밍 서버에서 서비스하는 MOV 및 MPEG 파일의 스트리밍 모듈과 ASF 파일의 스트리밍 서비스 모듈로 구성된다.

위의 [그림1]에서 세션을 생성하기 전에 각각의 포맷에 대한 Streaming Module과 File util이 실행된다. 클라이언트는 [표1]에 있는 RTSP method를 통하여 서버에 사용을 요청하고, 할당된 세션에 의해 서비스를 제공받는다. 반대로 서버는 클라이언트가 요구하는 각각의 메소드에 대한 분석을 통하여, 서비스 허락 여부를 응답한다.

3.2 클러스터 구조

본 논문에 적용된 클러스터 구조는 [그림2]에 나타난 구조를 가진다[5].

- 부하 분산기 : 사용자가 요구한 정보를 서비스 노드로 분배하는 노드로서 라운드 로빈 방식을 따른다.
- 서비스 노드 : 부하 분산기로부터 분배된 요구를 처리하는 노드로서 파일서버 노드로부터 데이터를 선인출하여 버퍼에 저장하며, 저장된 데이터를 사용자에게 전송한다.
- 파일서버 노드 : 파일서버 노드는 영화 파일을 저장하는 노드이다.

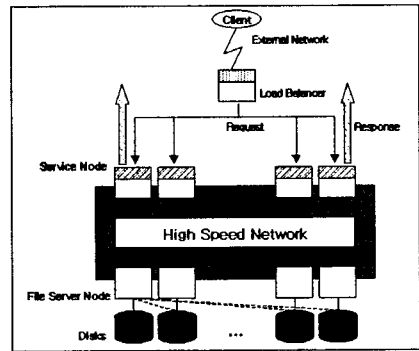


그림 2 사용된 클러스터 서버 구조

4. 클라이언트의 구성

4.1 플레이어

클라이언트(플레이어)는 DirectShow의 Filter 방식을 이용하여 구현하였다. 먼저 RTP 소스필터를 제작하였다. RTP RECEIVER Filter는 소스필터로서 RTSP 세션을 통하여 관리되며, 서비스가 이루어지면 서버로부터 RTP를 통하여 데이터를 전송받는다. 전송된 데이터는 각각의 타입에 따라 오디오와 비디오로 분리된 후 다음 단계의 필터로 보내진다. 디코더 필터는 미디어의 압축 포맷에 따라 기존의 것을 선택하거나 따로 제작하여 사용할 수 있다. 그 밖의 렌더링 필터는 Directshow에서 제공하는 방식을 사용하였다.

클라이언트는 원하는 파일의 URL이나 해당 정보를 가진 SDP 파일의 URL을 이용하여 서비스를 요청한다. 주로 라이브 서비스의 경우에 해당 파일이 존재하지 않으므로 SDP 파일의 정보를 분석하여 재생에 필요한 세션 정보를 얻어낸다. 즉, SDP에 정의된 내용을 바탕으로 재생에 필요한 필터(RTP session, decoder, renderer등)를 로드한다.

[그림3]은 플레이어에 사용된 필터들의 연결 모습이다. 특히 비디오 디코더 필터와 렌더링 필터 사이에는 압축 상태나 영상 효과 등에 의하여 다른 필터가 들어가는 경우가 빈번하다.

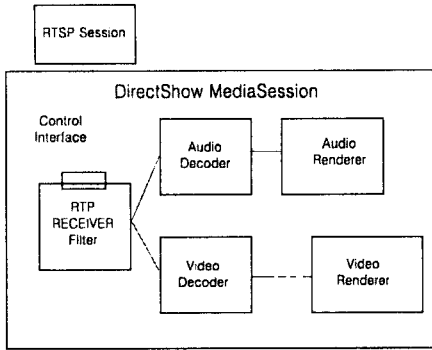


그림 3 DirectShow 기반의 Player 구성도

아래 그림[4]은 플레이어의 실행 모습이다. 동영상이 재생되는 부분과 그 아래 플레이어 컨트롤 부분으로 구성되어 있다.



그림 4 클라이언트의 재생 화면

4.2 실시간 전송

동영상 데이터의 실시간 전송을 위해 DirectShow Filter를 이용하여 오디오/비디오를 캡처하고 이를 압축한다. 캡처장치는 WDM(Windows Driver Model)을 따르는 장치 드라이버를 사용하였다.

오디오/비디오의 실시간 전송을 위한 RTP Sender Filter는 서버로 실시간 RTP 전송을 위해 따로 제작 되었으며, AVI Mux Filter와 File Writer는 전송한 내용을 따로 파일로 저장하기 위해 사용한 필터로 DirectShow에서 제공하는 방식을 사용하였다.

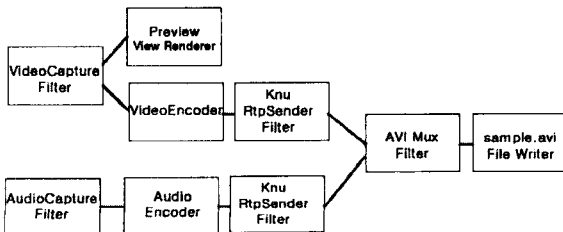


그림 5 캡처 및 전송 프로그램의 필터 연결

또, 라이브 서비스를 위한 실시간 전송 프로그램은 스트리밍이나 재생과 관련된 정보를 클라이언트에게 제공하기 위하여 미디어 정보를 담은 SDP 파일을 생성하는 기능도 제공한다.

5. 결론

본 논문에서는 빠르게 증가하는 사용자의 요청을 저렴한 비용으로 효과적으로 처리할 수 있는 리눅스 클러스터 서버 구조를 기반으로 하는 실시간 및 VOD 스트리밍 서버와 윈도우 기반의 플레이어를 개발하였다. 멀티미디어 데이터 스트리밍의 표준 프로토콜인 RTP/RTSP를 사용하여 실시간 및 VOD 스트리밍이 가능하며 또한 현재 윈도우 NT계열에서만 스트리밍 서비스가 가능한 ASF 포맷의 동영상 데이터를 리눅스 환경에서도 스트리밍 서비스가 가능한 모듈을 개발하였다.

향후 발전 방향으로 현재 ASF 스트리밍을 위하여 다윈 서버 위에 별도의 모듈로 삽입하여 프로세스가 증가되는 문제를 개선하고, 본 논문에서 적용한 클러스터 시스템에 멀티미디어 데이터의 특성을 고려한 동적버퍼분할을 이용한 클러스터내의 부하분산 알고리즘을 적용하여 보다 효율적인 서버의 구현을 목표로 하고 있다.

참고 문헌

- [1] RFC1889, RFC1890, RFC2326, RFC2327
- [2] "www.microsoft.com"
- [3] "developer.apple.com/darwin/projects/streaming/"
- [4] Valeria Cardellini, Michele Colajanni, Philip S. Yu. "Dynamic Load Balancing on Web Server Systems," IEEE Internet Computing 3(3). 28-39 1999.
- [5] W. J. Bolosky and J. S. Barrera, "The Tiger Video Fileserver", Proc. NOSSDAV '96, Apr. 1996.
- [6] D.M.Dias, W. Kish, R. Mukherjee, and R. Tewari, "A Scalable Highly Available Web Server," In Proceedings of the IEEE International Computer Conference, San Jose, CA, Feb. 1996.
- [7] IBM Corporation, "The IBM Interactive Network Dispatcher" 1998.
- [8] M.Aron, D.Sanders, P Druschel and W.Zwaenepoel. "Scalable Content-aware Request Distribution in Cluster-based Network Servers," In Proceedings of USENIX2000 Technical Conference, June 2000.
- [9] Microsoft, DirectX SDK, "www.microsoft.com/windows/directx/"
- [10] W. Zhang, "Linux Virtual Server for Scalable Network Services", Linux virtual server project
- [11] 권세오, 김상식, 김동승, "리눅스 클러스터형 웹 서버 설계", 정보과학회지, vol. 18, 2000.
- [12] 최재영, 최종명, 김은희, 김민석, "고가용성 리눅스", 정보처리학회지, vol. 6, 1999.
- [13] 서대화, "리눅스 클러스터 환경에서의 분산 파일 시스템", 정보처리학회지, vol. 6, 1999.
- [14] W. Zhang and et al., "Linux virtual server project", http://www.linuxvirtualserver.org, 1998