

확장가능한 다목적 주문형 멀티미디어 시스템의 설계 및 구현

김태현^o, 조진성, 성민영, 정세욱, 김기한, 신현식
서울대학교 컴퓨터 공학과

Design and Implementation of a Scalable, Multi-purpose Multimedia-on-Demand System

Taehyoun Kim, Jinsung Cho, Minyoung Sung, Sewook Jung,
Keehan Kim, Heonshik Shin
Department of Computer Engineering, Seoul National University

요약

주문형 멀티미디어 시스템은 사용자의 요구에 따라 다양한 응용을 가지며, 이러한 응용에 대해 서비스를 제공하는 서버는 고객의 수에 따라 융통성 있게 확장될 수 있어야 한다. 본 논문에서는 사용자의 목적에 적합한 멀티미디어 응용 프로그램의 작성을 용이하도록 하는 다목적 고객 API와 통신망의 형태와 사용자의 규모에 따라 적절히 구축될 수 있는 분산 다중 서버 시스템을 설계, 구현한다. 설계된 분산 다중 서버는 전위 노드의 역할을 수행하는 서비스 조정자와 복수의 데이터 서버로 이루어진 서버 풀(pool)로 구성되며, 유닉스 호환의 플랫폼에 쉽게 이식, 확장될 수 있도록 설계, 구현되었다. 또한 구현된 프로토타입 서버 시스템과 고객 API를 이용하여 작성된 CinemaNet 응용 프로그램을 연동하여 WWW 환경에서 주문형 비디오 서비스를 개발한다.

1 서론

멀티미디어 응용들에서 취급되는 데이터들은 일반적으로 기존의 데이터에 비해 그 크기가 상당히 크므로 대용량의 저장 장치를 필요로 한다. 따라서 멀티미디어 서버는 대용량의 저장공간과 실시간 접근성을 보장할 수 있는 수준의 대역폭을 제공하여야 하는데 기존의 단일 서버 시스템으로는 다수의 사용자에게 보장된 서비스를 제공하는 데 한계가 있다[1]. 이러한 한계를 극복할 수 있는 대규모 서버는 분산 시스템이나 다중 프로세서를 이용한 병렬 시스템으로 구현할 수 있으며 통신망의 구조나 사용자의 규모에 따라 확장이 용이한 구조를 가져야 한다.

한편 멀티미디어 서버를 이용해 멀티미디어 데이터를 제공받는 고객 시스템은 사용자의 목적에 따라 전자 신문, 전자 도서관, 주문형 비디오 등 다양한 응용의 형태를 띄게 된다. 그러므로 여러 응용을 목적에 맞게 개발하기 위한 고객 시스템 API의 제공은 응용 프로그래머의 부담을 크게 덜어 줄 수 있다.

따라서 본 논문에서는 고객과 서버의 양 측면에서 다음의 두 가지 요건을 충족시킬 수 있는 멀티미디어 시스템을 개발한다. 첫째, 다양한 목적을 가진 고객 응용 프로그램의 지원을 들 수 있다. 새로운 응용 프로그램의 개발이 요구될 때, 이것은 서버에 대한 완벽한 이해를 필요로 하므로 응용 프로그래머의 부담이 크다. 본 논문에서는 쉽게 멀티미디어 응용 프로그램을 작성할 수 있는 고객 시스템 API를 설계, 구현함으로써 프로그래머에게 서버 구조의 투명성을 제공한다. 둘째, 다수의 고객이 요구하는 품질의 서비스를 보장할 수 있

는 확장가능한 서버 구조가 필요하다. 주문형 멀티미디어 시스템은 다양한 형태의 통신망으로 구성될 수 있으며, 사용자의 규모에 따라 확장이 용이해야 한다. 본 논문에서는 PC와 워크스테이션으로 구성된 분산 다중 서버 시스템 구조를 제안하고 프로토타입 시스템을 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 주문형 멀티미디어 서비스의 요구사항에 대해 언급한다. 3장에서는 확장가능한 다목적 주문형 멀티미디어 시스템을 위한 분산 다중 서버와 고객 API의 구조와 기능을 설명하고, 4장에서는 3장에서 구현된 프로토타입 서버를 기반으로 웹 환경에서 주문형 비디오 서비스를 개발한 예를 보인다. 마지막으로 5장에서는 결론과 향후 과제를 제시한다.

2 주문형 멀티미디어 서버

2.1 요구사항

주문형 멀티미디어 서비스는 사용자의 선택에 따라 원하는 시간, 원하는 품질(QoS)로 멀티미디어 데이터 서비스를 제공함을 의미한다. 일반적으로 멀티미디어 정보가 대용량인 반면 실시간 재생을 요구한다는 점을 감안하면 주문형 멀티미디어 서비스를 제공하는 서버는 다음과 같은 기능적 요구사항을 갖는다고 할 수 있다.

- 대용량의 멀티미디어 데이터를 저장할 수 있는 충분한 용량의 저장장치를 가져야 하며, 저장장치와 통신망에 충분한 대역폭

이 확보되어야 한다.

- 데이터의 용도, 성격에 따라 적절하게 분류, 배치되어야 한다.
- 동시에 다수의 사용자에게 실시간 서비스를 제공할 수 있는 효율적 저장, 검색 메카니즘을 가져야 한다.

2.2 대규모 주문형 멀티미디어 서비스

대규모 주문형 멀티미디어 서버는 다수의 사용자에게 서비스를 제공하기 위해 2.1 절에서 언급한 것 외에 다음과 같은 요구 조건을 추가로 가진다.

- 서버는 데이터 가용성과 성능 향상을 위해 각 저장 노드에 적절히 멀티미디어 데이터를 복제, 저장하여야 한다.
- 데이터가 분산, 저장되어 있음을 사용자에게 감출 수 있도록 투명성을 제공하여야 한다.
- 사용자가 늘어남에 따라 확장이 용이한 하드웨어, 소프트웨어 구조를 가져야 한다.

이러한 요구조건을 만족시키기 위해 대규모 주문형 멀티미디어 서비스를 제공하는 서버는 일반적으로 다수의 노드로 이루어진 분산 시스템[2] [4]이나 다중프로세서를 이용한 대용량 병렬 시스템[5]의 형태로 구현된다. 분산 시스템의 형태로 구축되는 멀티미디어 서버의 경우 PC와 워크스테이션을 이용해 적은 비용으로 확장성이 뛰어난 서버를 구축할 수 있는 장점이 있다. 따라서 본 논문에서 개발하는 주문형 멀티미디어 서버는 PC와 워크스테이션으로 구성되는 분산 다중 서버 구조로 설계되었다.

3 주문형 멀티미디어 시스템의 설계 및 구현

3.1 다중서버의 설계 및 구현

본 연구에서 개발하는 확장가능한 멀티미디어 서버는 서비스 조정자(service coordinator)와 데이터 서버 풀(pool)로 구성된다. 서비스 조정자는 통신망을 통해 각 서버에 분산된 데이터 자원을 효율적으로 관리하며, 사용자에게 투명성을 제공한다. 데이터 서버 풀은 기본적으로 디스크 배열을 저장장치로 가지는 PC 서버들로 구성된다. 서버 시스템을 구성하는 각 시스템 요소들은 그림 1과 같이 지역망(LAN)으로 연결되며, 고객과는 고속 통신망을 통해 연결된다.¹

서버 풀을 구성하는 단일 서버의 기본 구조는 그림 2와 같다. 단일 서버는 PC에 기반하며 QNX 마이크로커널 실시간 운영체제 상에서 동작한다². 서버는 [7]에서 구현된 구조에 기반하여 디스크 배열을 저장장치로 가지며 서비스 관리자(MoDman), 디스크 배열 관리자(DAman), 통신 관리자(NETman)로 구성된다. 서비스 관리자는 고객의 요청과 그에 대한 응답을 각각 디스크 배열 관리자와 통

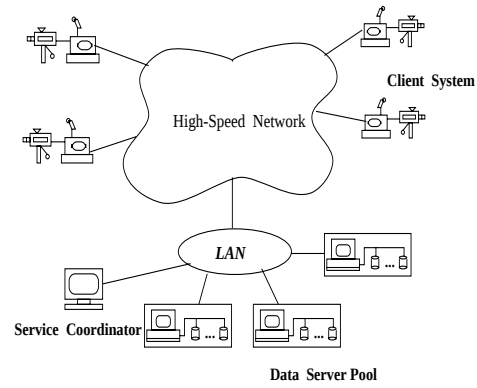


그림 1. 주문형 멀티미디어 서비스의 구조

신 관리자에게 전달, 중계하는 역할을 한다. 디스크 배열 관리자는 디스크 서브시스템과 멀티미디어 화일 시스템을 구현한 부분으로, 디스크 스트라이핑 기법을 사용하였으며 이를 위해 별도의 디스크 배열 화일 시스템(DAFS)을 가진다. 통신 관리자는 이더넷 상에서 TCP/IP 프로토콜을 이용하여 구현하였으며, ATM 네트워크로의 확장을 감안하여 설계되었다.

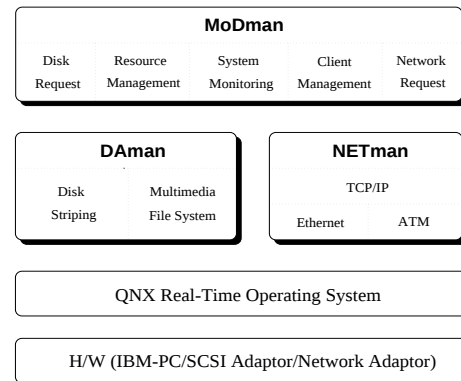


그림 2. 단일 멀티미디어 서버의 구조

다중 서버 시스템의 소프트웨어 구조와 각 시스템 요소들간의 상호작용은 그림 3과 같다. 서비스 조정자는 그림 3에서와 같이 서비스 스케줄링, 고객 수용 검사, 서버 상태 모니터링, 서버 지명 서비스, 전역 카탈로그 유지 및 관리 등의 역할을 수행한다. 데이터 전송 이외의 모든 제어 정보를 서비스 조정자가 관리함으로써 멀티미디어 서버는 고객에게 일정 수준의 투명성을 제공하게 된다.

서비스 조정자의 주요 기능은 다음과 같다. 첫째, 사용자 인증 기능을 수행한다. 사용자 인증 과정은 허가되지 않은 고객의 접근을 방지함으로써 시스템 자원이 낭비되지 않도록 하는 데 일차적인 목적이 있다. 그 외에도 사용자 계정 관리, 이용 패턴 로깅을 통해 좀더 지능적인 서비스를 가능하도록 한다. 둘째, 고객 수용 검사와 서버 모니터링이 있다. 고객 수용 검사 과정은 서버의 가용 능력을 감안하여 수용 여부를 결정함과 동시에 서버의 부하를 고려하여 고객이 서비스받을 서버를 지정한다. 또한 결함 탐지를 위해 고객이 특정 아이템을 요청하면 서버의 상태를 점검한다. 만약 관련된 서버가 장애 상태³이면 관련 서버 정보와 카탈로그를 동적으로 재구성한다.

¹ 현재 서버는 이더넷으로 연결되어 있으며 고객과는 ATM 네트워크와 같은 고속망으로 연결될 수 있다.

² 본 연구에서 구현된 서버 소프트웨어는 유닉스 환경에 쉽게 이식되도록 설계되어, 유닉스 화일 시스템을 이용한 워크스테이션 서버를 구성요소로 가질 수 있다.

³ 현재 구현된 서버는 저장장치 고장, 통신 장애 등의 원인에 의한 노드 장애만을 처리한다.

세제, 지명 서비스와 전역 카탈로그 정보⁴의 유지를 위해 데이터베이스를 관리한다. 서비스 수용 과정에서 서비스 조정자는 관련 서버와 고객간의 연결을 위한 매개 역할을 담당하고 실제 데이터 전송은 서버와 고객 사이에 직접적으로 이루어진다. 서비스 조정자를 데이터 전송 과정에서 배제한 이유는 서버 시스템 내의 통신 성능이 따라 시스템의 성능이 좌우되므로 서비스 조정자가 성능의 병목 지점이 되는 것을 방지하기 위한 것이다. 또한, 서비스 조정자는 데이터 서버의 상태에 따라 전역 카탈로그에 멀티미디어 화일의 메타 정보를 동적으로 추가, 삭제, 수정할 수 있도록 한다. 이러한 정보의 관리를 위해 mSQL(Mini SQL) 데이터베이스 엔진과 C 언어 API를 이용하였다.

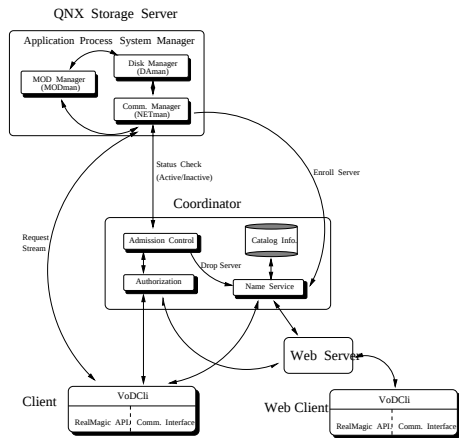


그림 3. 다중 서버의 동작

분산 시스템을 갖는 멀티미디어 서버는 데이터 저장 구조에 따라 크게 두 가지 형태로 나눌 수 있다. 예를 들어, Bellcore의 Calliope[3]의 경우 각 노드가 독립적으로 서비스를 수행하며 이 저장 노드들을 관리하는 별도의 노드가 존재한다. 이 방법은 노드 간의 부하 불균형이 일어날 가능성이 있지만 VOD와 같은 응용에는 사용자 요청의 편중 현상이 있음을 감안하여 적절한 중복 메카니즘으로 부하 균배를 얻을 수 있다. Microsoft의 Tiger File Server[2]의 경우 모든 노드에 데이터를 스트라이핑된 형태로 저장함으로써 보다 높은 디스크 대역폭과 부하 균배 효과를 얻을 수 있다. 그러나, 이 방법의 단점은 전역 스트라이핑(wide striping)을 적용함으로써 시스템의 확장성에 제약을 받는다는 점이다.

따라서 본 논문에서 구현된 데이터 서버는 확장성을 감안하여 각기 자율성을 가지고 서로 독립적으로 동작하도록 설계되었다. 각 서버 노드가 독립적으로 동작하므로 시스템의 성능과 사용자가 요구하는 QoS 수준에 따라 적절히 부하가 분배되어야 한다.

3.2 고객 시스템 API의 설계 및 구현

본 절에서는 주문형 멀티미디어 고객 응용 프로그램(MoDcli)의 개발을 돕기 위한 API의 설계와 구현에 대해 언급한다. 일반적으로 주문형 멀티미디어 서비스를 위한 고객 응용 프로그램은 네트워크 통신을 통한 서버와의 데이터 교환, 멀티미디어 장치의 제어 및 미디어 재생, 사용자의 상호작용 등의 작업을 수행하여야 한다. MoDcli

API는 이러한 고객 응용 프로그램의 개발을 돕기 위한 인터페이스로서 설계되었으며, 다음과 같은 기능을 제공하는 것을 목적으로 한다. 첫째, 사용자로 하여금 데이터가 원격지에 저장되어 있음을 감출 수 있도록 투명성을 제공한다. 둘째, 멀티미디어 장치에 의존적인 부분은 API 함수 내에 감추어 장치 독립성을 구현한다. 세째, 새로운 멀티미디어 장치에 대한 확장이 용이한 구조로 설계된다. 네째, 모든 미디어를 다룰 수 있는 일관된 인터페이스를 제공한다.

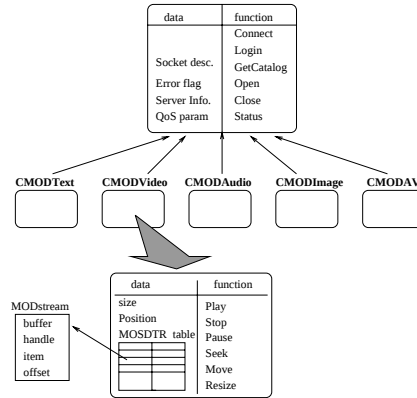


그림 4. MoDcli API의 구조

그림 4는 MoDcli API의 구조를 나타낸 것이다. 각 미디어 클래스는 미디어 데이터를 그 데이터 멤버로 하며 미디어 데이터의 재생, 멀티미디어 장치 제어 등의 제반 작업을 수행한다. 미디어 클래스는 Generic Media Class(CMODGeneric)를 기반으로 하며, CMODGeneric은 주로 네트워크 관련 정보와 사용자 정보를 유지/관리한다. MoDcli API는 마이크로소프트 윈도우즈 멀티미디어 확장에서 제공하는 MMSYSTEM 라이브러리, Winsock 등을 이용하여 Visual C++로 구현되었다. 본 연구에서는 구현된 MoDcli API를 이용하여 웹 환경을 고려한 주문형 비디오 서비스 응용 프로그램 CinemaNet을 구현하였으며 4장에서 자세히 설명하기로 한다.

4 WWW에서의 주문형 비디오 서비스 구현

본 연구에서는 구현된 프로토타입 서버 시스템과 다목적 고객 API를 이용해 작성된 CinemaNet 응용 프로그램을 이용해 웹 환경에서 주문형 비디오 서비스(VOD; Video-on-Demand)를 구축하였다.

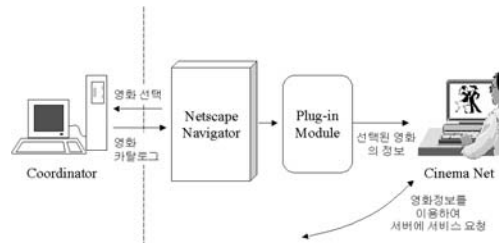


그림 5. VOD 실행 과정

⁴전역 카탈로그란 서비스가능한 모든 멀티미디어 데이터 아이템에 대한 정보(데이터 보유 서버의 주소, 미디어 타입, 장르, 설명)를 말한다.

주문형 비디오 서비스의 구성요소는 다음과 같다. 서버 시스템은 3.1절에서 설계, 구현된 프로토타입 서버와 웹 서버를 CGI (Common Gateway Interface) 프로그램을 통해 연동, 확장한 것이다. 고객 시스템은 웹 브라우저, 고객 응용 프로그램 CinemaNet, 웹 브라우저와 고객 응용 프로그램 간의 통신을 담당하는 플러그인(Plug-In) 프로그램[9], 하드웨어 MPEG 디코더로 구성된다. 고객 응용 프로그램인 CinemaNet은 Visual C++의 MFC 라이브러리와 고객 API의 CMODVideo 클래스를 이용해 Windows 95 환경에서 동작하도록 구현되었으며, 기본적인 VCR 기능과 음량 조절 기능 등 다양한 기능을 가진다.

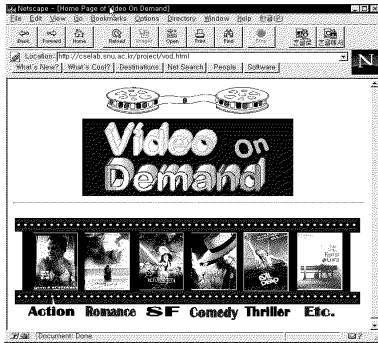


그림 6. 주문형 비디오 서비스 홈 페이지

주문형 비디오 서비스의 실행 과정은 그림 5와 같다. 먼저 사용자는 그림 6의 주문형 비디오 서비스 홈페이지를 통해 장르별로 영화를 선택하게 된다. 사용자가 영화를 선택하면, 선택된 영화의 카탈로그 정보를 화일 형태로 저장하고 플러그인 프로그램을 동적으로 로드하는 CGI 프로그램이 실행된다. 이렇게 구동된 플러그인 프로그램은 영화 정보⁵를 분석한 뒤 CinemaNet 응용 프로그램을 호출한다. 플러그인으로부터 선택된 영화의 정보를 넘겨 받은 CinemaNet 프로그램은 해당 영화를 저장하고 있는 데이터 서버와 직접 연결을 설정한 뒤 영화를 서비스하게 된다. 그림 7은 CinemaNet의 실행 화면이다.

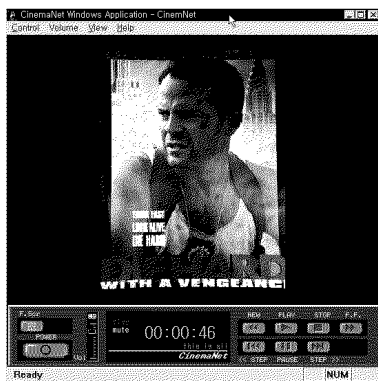


그림 7. CinemaNet의 실행 화면

⁵ 영화 정보는 해당 영화를 보유하고 있는 서버의 주소, 아이템 식별자 등의 정보를 포함한다.

5 결론 및 향후과제

본 논문에서는 확장성을 감안한 분산 다중 서버를 설계·구현하고 사용자의 목적에 맞는 멀티미디어 응용을 쉽게 구현할 수 있도록 설계된 다목적 고객 API를 구현한다.

구현된 분산 다중 서버는 서비스 조정자와 데이터 서버 풀로 구성되어 사용자의 규모에 따라 쉽게 확장될 수 있는 구조로 설계되었다. 구현된 다중 서버의 성능은 분석적 모델링[8]과 시뮬레이션[6]을 통해 검증되었다. 다목적 고객 API MoDcli는 객체 지향 개념을 도입하여 다양한 멀티미디어 응용 프로그램의 작성을 손쉽게 할 수 있도록 설계, 구현되었으며 본 연구에서는 MoDcli를 이용하여 주문형 비디오 서비스를 위한 CinemaNet 응용 프로그램을 작성하였다.

향후 과제로는 구현된 분산 서버를 이용한 계층적 다중 서버의 구현과 현재 하드웨어의 지원이 필요한 MPEG 재생 기능을 소프트웨어적으로 구현하는 것이다.

참고 문헌

- [1] R. Tewari, R. Mukherjee, D. M. Dias and H. M. Vin, "Design and Performance Tradeoffs in Clustered Video Servers," *Proc. of International Conference on Multimedia Computing and Systems*, pp. 144-150, 1996.
- [2] W. J. Bolosky, et al., "The Tiger Video File server," *Proc. of the 6th International Workshop on Network and OS Support for Digital Audio and Video*, pp. 97-104, 1996.
- [3] A. Heybey, M. Sullivan and P. England, "Calliope: A Distributed, Scalable Multimedia Server," *Proc. of the USENIX Annual Technical Conference*, 1996.
- [4] C. S. Freedman and D. J. DeWitt, "The SPIFFI Scalable Video-on-Demand System," *Proc. of SIGMOD '95*, pp. 352-363, 1995.
- [5] A. Laursen et al., "Oracle Media Server : Providing Consumer Based Interactive Access to Multimedia Data," *Proc. the 1994 International Conference on the Management of Data*, pp. 470-477, 1994.
- [6] 조진성, 김태현, 김영구, 성민영, 신현식, "멀티미디어 서버의 저장 구조를 위한 디스크 배열의 성능 분석," '95 가을 학술발표논문집, 한국정보과학회, pp. 823-826, 1995.
- [7] 조진성, 김태현, 김영구, 성민영, 신현식, "마이크로커널 환경에서의 비디오 서버를 위한 디스크 배열 관리자," '96 봄 학술발표논문집, 한국정보과학회, pp. 823-826, 1996.
- [8] 조진성, 신현식, "대규모 주문형 비디오 서버의 큐잉 모델," '97 봄 학술발표논문집, 한국정보과학회, pp. 383-386, 1997.
- [9] Z. Oliphant, "Programming Netscape Plug-Ins," Sams net, 1996.