

VOD 서버와 자바 비디오 플레이어를 이용한 노래방 시스템 구현

○
문보석, 김인수, 최인욱, 박성순
안양대학교 컴퓨터학과

Implementation of Noraebang System by using VOD Sever and Java Video Player

○
Bo-Seok Moon, In-Soo Kim, In-Uk Choi and Sung-Soon Park
Dept. of Computer Science and Engineering, Anyang University

요 약

요즘 PC를 이용한 여러 가지 사업 아이템들이 상품화되어 쏟아져 나오고 있다. 대표적으로 PC방을 비롯해 가정에서도 고속 인터넷이 가능해지면서 게임뿐만이 아니라 전반적인 생활 자체가 컴퓨터를 통해 이루어져 가고 있다고 해도 과언이 아니다. 흔히 볼 수 있는 노래방 또한 그 예외가 될 수 없다. 현재 노래방에서의 한계점은 신곡이 추가될 때마다 대부분 칩 자체를 교환하거나 추가해줘야 하고 TV 브라운관에 보이는 영상 또한 비디오를 통해 따로 상영해 주어야 한다. 본 논문에서는 PC로 클라이언트-서버 모델을 이용해 노래방 시스템을 구성하여 기존 노래방 시스템의 단점을 극복하고 나아가서 여러 가지 부가 서비스를 지원해 줄 수 있는 인터넷 노래방 시스템의 구현에 대해서 기술한다.

1. 서론

초고속 인터넷 시대가 열리면서 사용자들의 요구 사항이 매우 다양해지고 단순한 웹서핑이나 게임이 아니라 쇼핑, 은행 업무, 티켓 구매, 멀티미디어 등의 수많은 아이템들이 창출되고 있다. 특히 멀티미디어 분야의 데이터 요구가 폭발적으로 증가하면서 인터넷 방송국은 물론이고 기타 여러 사이트에서도 멀티 캐스팅과 더불어 VOD(Video On Demand) 서비스를 함께 제공하고 있다. 이제 얼마 머지 않아서 우리 생활 자체가 모두 가정에서의 PC를 통해 이루어지는 날이 올 것이라는 사실은 누구나 인지하고 있을 것이다.

이러한 시대적 상황에 발 맞추어 가정에서의 PC

는 물론이고 냉장고, 세탁기, TV, 보일러 등의 가전 제품 또한 네트워크에 연결시켜 원격 조종이 가능한 시제품들이 나오고 있으며 우리가 흔히 볼 수 있는 노래방 역시 그 예외는 아닐 것이다.

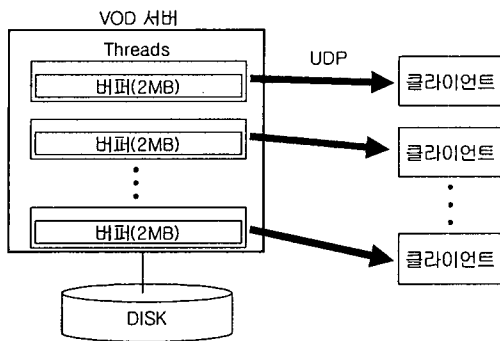
아직까지 노래방 기계는 신곡이 추가 될 때마다 칩 자체를 교환하거나 추가 하기 때문에 비용면에서 매우 불리한 단점을 가지고 있고 부가적인 서비스를 하기에는 그다지 효율적이지 못한 것이 사실이다. 이를 비용면에서 뿐만 아니라 부가적인 면에서도 경쟁력을 높이기 위해 지금의 독립적인 시스템이 아닌 클라이언트-서버 모델로 구성해 준다면 똑같은 곡이 들어있는 칩이 여러 기계에 중복되어 존재하기 때문에 발생하는 비용 부담을 줄이고 네트워크를 통해 갖가지 부가 서비스를 제공할 수 있을 것이다.

본 논문에서는 이러한 클라이언트-서버 모델의 PC 기반 노래방 시스템의 구현에 대해 기술하고 이를 위하여 서버는 CPU Pentium 166, RAM 48M, HDD 1.2G, Network 100Mbps, OS Linux 그리고 클라이언트의 개발환경은 CPU Pentium 350, RAM 64M, HDD 6.4G, Network 10Mbps, OS Windows 98 이지만 실제 실험은 여러 사양에서 실시되었고 또한 클라이언트는 OS의 종속을 피하기 위해 Java로 구현되었다.

2. 관련연구

2.1 서버에서의 효율적인 디스크 접근을 위한 버퍼 관리

VOD 서버에서는 <그림 1>과 같이 새로운 클라이언트가 접속해 오면 서비스를 제공할 클라이언트 세션 쓰레드를 생성하고 클라이언트가 요구하는 데이터를 버퍼링 하게 되는데 클라이언트 세션 쓰레드가 관리하는 내부 버퍼에 약 10초 분량(2MB)을 버퍼링한 후 UDP를 통해 전송하게 된다. 쓰레드는 일정 간격으로 Mpeg-1 데이터를 패킷 단위(16KB)로 전송하며 버퍼의 잔여 데이터량이 10%이하로 떨어지면 다시 버퍼의 나머지 공간에 버퍼링을 해서 한 클라이언트의 디스크 접근 시간(횟수)을 줄인다.

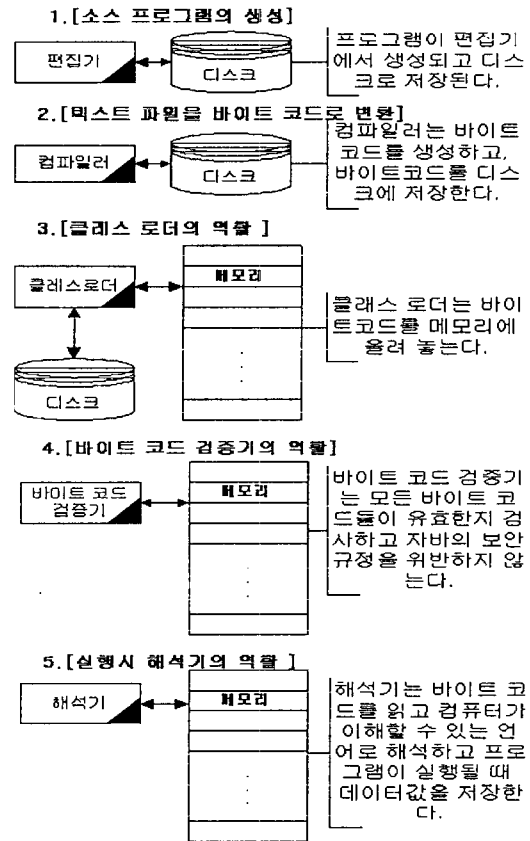


<그림 1> 클라이언트 세션 쓰레드의 서비스 형태

2.2 자바

자바란 선 마이크로 시스템에서 개발한 언어로 가전기기 이식을 위해 탄생되었으나 인터넷 환경에 강력한 성능을 보임으로 인해 현재 인터넷 언어로 많이 사용되고 있고 또한 기기 이식이 우수하도록 설계되었다. 자바에서 ByteCode 생성은 컴파일러 언어에 가까운 형태를 보여주고 있으며 VM(Virtual Machine)

의 ByteCode 해석은 인터프리터 언어의 형태를 취하고 있다. ByteCode 생성과 해석 과정은 <그림 2>와 같으며 본 논문에서는 윈도우용 JDK 1.3을 사용하였다.



<그림 2> ByteCode 생성과 해석 과정

2.3 JMF

JMF는 자바에서 이용할 수 있는 특수 클래스 라이브러리로 동영상을 JPG 형식으로 캡처하거나 반대로 캡처된 파일을 동영상으로 만들 수 있으며 Mpeg-1 과 더불어 Wav, Mp3 등 여러 종류 압축 화일의 디코더 역할을 할 수 있다. 또한, RTP(Real Time Protocol)를 지원하기 때문에 네트워크를 통한 동기화를 쉽게 지원하며 본 논문에서는 윈도우용 JMF 2.0을 사용하였다.

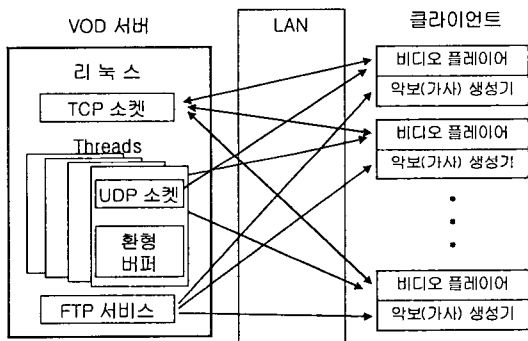
2.4 미디 파일

본 노래방 시스템에서는 실제 노래방의 기계 처럼 가사가 나오기도 하지만 가사 뿐만 아니라 악보까지

도 출력이 된다. 이 악보는 미리 준비되어 있는 것이 아니라 미디 파일을 읽어들이면서 동적으로 생성되는 것이다. 현존하는 미디 파일은 다양한 구조로 되어 있고 그 미디 파일의 악보를 생성 시키기는 매우 복잡하다. 이 문제를 보완하기 위해서 고려된 것은 이 복잡한 미디 파일 구조를 일련의 규정을 두어 미디 음파, 가사, 음표 등의 정보를 규칙처럼 만든 것이다. 이것은 보통 미디 파일과는 크게 다르지 않은 구조로 되어있고 미디 파일을 재생 하는데 있어서 아무런 지장없이 악보를 생성시킬 수 있다.

3. 설계 및 구현

3.1 전체 시스템 구성



<그림 3> 전체 시스템 구성

본 논문에서 구현된 노래방 시스템은 <그림 3>과 같이 클라이언트-서버 모델을 이용한 것으로 기존 노래방의 독립적인 기계와는 전혀 다른 형태이다. 서버의 운영체제는 리눅스이며 클라이언트는 자바로 만들어졌기 때문에 운영체제와는 독립적으로 동작하게 된다.

3.2 서버

3.2.1 TCP

클라이언트와 서비스 제공에 필요한 갖가지 메시지(정보)를 교환하며 해당 메시지에 대한 적절한 응답을 하게 되고 메시지는 TCP 패킷 데이터 부분의 상위 1Byte 값에 따라 구별된다. 새로운 클라이언트가 접속할 때마다 새로운 클라이언트 세션 스레드를 생성시키며 스레드는 해당 클라이언트에게 Mpeg-1 데

이터를 UDP 소켓을 통해 전송한다.

3.2.2 Thread

클라이언트가 접속한 후 서비스에 필요한 모든 정보를 얻게 되면 클라이언트 세션 스레드가 생성이 되는데 이 스레드는 클라이언트가 원하는 Mpeg-1 데이터를 환형 버퍼에 약 10초 분량(2M)을 버퍼링 한 후 클라이언트가 개설한 UDP 소켓에 일정 간격으로 전송한다. 이러한 버퍼링으로 하여금 클라이언트 세션 스레드의 디스크 접근을 효율적으로 줄일 수 있다. 본 논문에서 구현된 Push 방식은 클라이언트 버퍼의 오버플로우를 방지하기 위해 Mpeg-1 데이터 전송 간격을 클라이언트가 송신하는 메시지에 따라 조절하게 된다.

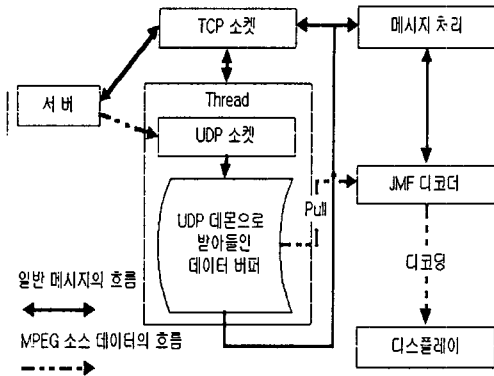
3.2.3 FTP 서비스

클라이언트는 VOD 서버의 TCP 소켓에 접속하기 전에 우선 FTP 접속을 하고 필요한 미디 파일을 다운 받는다. 다운 받은 미디 파일을 가지고 악보(가사) 생성기에서 악보와 가사를 생성하는데 이러한 작업을 위해 FTP 서비스를 해준다.

3.2.4 스트리밍 서비스 과정

- 1) 클라이언트의 FTP 접속
 - 클라이언트가 원하는 미디 파일의 다운로드
- 2) 클라이언트의 TCP 접속
 - 클라이언트의 여러 가지 정보를 저장할 객체를 생성시키고 클라이언트가 원하는 파일명을 수신하면 해당 파일을 열어 버퍼에 미리 버퍼링을 해둔 후 클라이언트에게 서비스 준비가 끝났다는 메시지 전송
- 3) 클라이언트의 준비 완료 메시지 수신
 - 클라이언트가 UDP 소켓을 생성한 후 준비 완료 신호를 서버가 수신하면 클라이언트 세션 스레드를 생성하고 UDP 소켓을 개설해 버퍼에 있는 데이터를 패킷(16KB) 단위로 전송
 - 패킷을 송신하면서 버퍼의 잔여 데이터량이 일정 비율(10%) 이하로 떨어지면 버퍼의 잔여 분량의 데이터를 다시 버퍼링
- 4) 클라이언트의 각종 메시지 수신
 - 클라이언트의 여러 가지 메시지를 수신하여 해당 작업을 수행(Play , Pause , Resume , Stop , Alive , Close , Buf_High , Buf_Low 등등)

3.3 비디오 플레이어



<그림 4> 비디오 플레이어의 구성

비디오 플레이어는 JDK1.3과 JMF 2.0을 사용하여 자바로 구현되었으며 서버에서 스트리밍 서비스 되는 Mpeg-1 데이터의 디스플레이 과정은 다음과 같다.

- 1) 서버에 TCP 접속
 - 원하는 파일명을 서버에 송신하고 서버가 해당 파일을 열고 버퍼링한 다음 준비 완료 신호를 보낼 때 까지 대기
- 2) 쓰레드와 UDP 소켓 생성
 - 서버에서 송신하는 Mpeg-1 데이터를 수신하기 위해 쓰레드를 생성하고 그 내부에서 UDP 소켓을 개설한 후 서버에게 준비 완료 신호를 송신
- 3) 패킷 수신, 버퍼링 및 디코딩
 - UDP 소켓을 통해 수신된 패킷은 환형 버퍼에 저장되며 90% 이상 데이터가 채워졌을 때 JMF 디코더에서 디코딩을 시작하게 된다.
- 4) 서버에게 각종 메시지 송신
 - 클라이언트의 버퍼 상태, 버튼 조작등의 메시지를 서버에 송신

3.4 악보(가사) 생성기

기술된바 있지만 현존하는 미디 파일은 여러 형태의 구조를 가지고 있기 때문에 모든 종류의 미디 파일을 가지고 악보를 생성시키기는 쉬운 일이 아니다. 이러한 문제를 해결하기 위해 규칙을 나름대로 정의했으며 그 규칙이 미디 파일을 재생하는 데는 아무런 영향을 미치지 않는다. 악보 생성 루틴은 다음과 같다.

- 1) 미디 파일의 Header Chunk 부분을 읽어 Track 수와 한 박자의 길이를 얻는다.

- 2) 각각의 이벤트를 읽어서 음표, 가사, 템포, 박자를 구별해 낸다.
- 3) 음표를 정규 형태로 처리한다.
- 4) 각각의 이벤트로 구성된 정보를 이용하여 음표 이미지와 비교해 악보를 생성한다.

4. 결론 및 향후 과제

이제까지 VOD 서버와 자바 비디오 플레이어를 이용한 노래방 시스템 구현에 대해서 살펴 보았다. 이 시스템을 구현하면서 문제점이 드러났던 것은 비디오 플레이어와 악보(가사) 생성기가 자바로 만들어졌기 때문에 CPU와 메모리 점유율이 매우 높다는 것이다. 이 문제의 해결을 향후 과제로 남겨둔다.

본 논문에서 구현된 노래방 시스템은 단지 노래방 시스템만을 구축하기 위해 개발된 것이 아니다. VOD 서버는 실시간 스트리밍 서버로써 이를 호텔등의 객실 또는 랜이 구축되어 있는 아파트 단지등의 비디오 서버로 활용할 수 있으며 자바 플레이어 또한 가정의 PC 뿐만이 아니라 셋톱 박스(Set-Top Box)에서도 실행될 수 있다는 장점을 가지고 있다. 멀티미디어 데이터 요구가 폭증하고 있는 요즘 본 노래방 시스템에서의 VOD 서버와 자바 비디오 플레이어를 활용하여 많은 분야에 적용시킬 수 있을 것이다.

[참고문헌]

- [1] 김화중, "컴퓨터 네트워크 프로그래밍", 홍릉출판사, 1998.
- [2] 신재호의 2인, "Network Bible 2nd ED.", 영진출판사, 1998.
- [3] Richard Stones & Neil Matthew 저 / 이태용 역, "Linux Programmin 2nd Edition", 정보문화사, 2000
- [4] Terrence Chan 저 / 백정현 역, "UNIX 시스템 네트워크 프로그래밍", 이한출판사, 1999
- [5] H.M.Deitel, P.J. Deitel 저 / 손진욱 편저, "JAVA2 Programming Bible", 정보문화사, 2000
- [6] Craig Hunt 저 / 박창민 역, "TCP/IP 네트워크 관리 2nd Edition", 한빛미디어, 1999
- [7] Sean C. Sullivan, Loren Winzeler, Jeannie Deagen, Deanna Brown, "Programming with the Java Media Framework", John Wiley & Sons, 1998