

대용량 멀티미디어 메일을 위한 분산 멀티미디어 스트림 상의 데이터서버의 설계 및 구현

박용희⁰ 전성미⁰ 고희선⁰ 임영환⁰⁰

⁰ 송실대학교 대학원 컴퓨터학부

⁰⁰ 송실대학교 대학원 미디어학부

{⁰mirinae, smchon, ⁰⁰haft}@media.soongsil.ac.kr

⁰⁰ yhlim@media.soongsil.ac.kr

A design and implementation of data server on distributed multimedia streams for huge multimedia mail

Yong-Hee Park⁰

Sung-Mi Chon⁰

Hee-Seon Ko⁰

Young-Hwan Lim⁰⁰

⁰School of Computing, Soongsil Univ.

⁰⁰School of Media, Soongsil Univ.

요 약

기존의 멀티미디어 메일은 대용량 메일이라 개인 사용자에게 그 부담이 크지 않을 수가 없었다. 이에 인터넷 기반에서 멀티미디어 데이터서버에 대해서 많은 제품뿐만 아니라 서비스에 대한 QoS 만족을 위해 많은 연구실에서 연구해 왔었다. 특히나, VOD 서버나 인터넷방송을 위한 서버의 구축을 위해 많은 연구가 있어왔다. 본 연구는 인터넷기반이 아닌 본 연구실의 프로토타입인 VIP(Visual Interface Player)와 멀티미디어 스트림 엔진인 Essence 를 이용한 멀티미디어 동영상 메일 전송 기법과 Essence 를 이용한 동영상 멀티미디어 메일을 저장하는데 필요한 Essence 내에 데이터베이스를 지원하기 위한 DBMedium 개발과 데이터베이스 라이브러리를 개발하기 위한 일련의 과정을 제안하고 있다.

1. 서론

최근 컴퓨터와 통신 기술의 발달과 더불어 영상 및 비디오, 오디오 등을 중심으로 한 멀티미디어 정보 서비스에 대한 요구가 증가하고 있다. 그 중에서 일반인에게 가장 친숙한 멀티미디어 응용으로는 VOD, 화상회의, 원격진료 및 원격 교육 등이 있는데, 여기서는 멀티미디어 메일이라는 새로운 멀티미디어 서비스를 제공하고 있다. 인터넷상의 동영상 멀티미디어 메일은 동영상을 포함한 멀티미디어 연출 프로그램을 프리젠테이션 편집기로 작성하여 인터넷의 이메일로 첨부해서 보내면 받은 연출 프로그램을 실시간으로 수행해 볼 수 있는 메일이다. 즉, 송신자는 동영상이 포함된 멀티미디어 메일의 내용을 VIP(Visual Interface Player)라는 멀티미디어 연출 편집기로 작성하여 인터넷에서 기존에 사용되고 있는 이메일(e-mail)에 첨부해서 보내면 수신자는 일반 이메일을 통하여 받은 것을 두 번 클릭하여 "Open"하거나 "Play"하면 멀티미디어 연출 프로그램이 수행하면서 동영상 메일을 실시간으로 연출해 볼 수 있도록 하는 것이다. 동영상이 포함된 멀티미디어 메일과 관련된 기술은 많이 개발된 상태이나 다음과 같은 문제점 때문에 저속망의 인터넷에서 실용화가 되어 있지 못하다. 첫 번째로 멀티미디어 이메일은 그 데이터 자체가 대용량이라 개인 PC 사용자에게 있어서 대용량의 하드디스크 공간을 요구하게

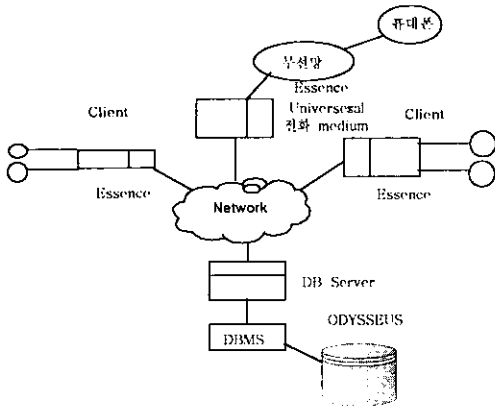
된다. 이를 위해 대용량 멀티미디어 메일서버가 필요하게 되었다. 본 연구에서는 서버시스템 NT 급으로 멀티미디어 데이터베이스로 한국과학기술원에서 개발한 '오딧세우스'를 이용하여 멀티미디어 데이터를 저장할 것이다. 두 번째로 대용량의 멀티미디어 파일을 어떻게 보내는가에 있다. 이는 VIP 라는 멀티미디어 편집기를 이용하여 프리젠테이션 파일(.prs 라는 확장자)로 만들어 프리젠테이션 정보만을 가지는 파일로서 실제 데이터의 크기는 1KB 정도만을 차지하게 된다. 실제 대용량의 데이터는 보내는 송신자의 PC에 존재하기 때문에 전송하는데 불편함이 없다. 세 번째로 송신자가 보낸 멀티미디어 메일을 열어 프리젠테이션 파일(.prs)을 실행시키면 항상 소스가 되는 멀티미디어 데이터가 존재하는 컴퓨터는 항상 On-line 상태로 존재해야 한다. 개인 컴퓨터 사용자일 경우에는 이런 점이 불편함이 아닐 수가 없다. 이에 앞에 언급하였듯이 On-Line 상태로 작동할 수 있는 메일 서버가 마찬가지로 필요하게 되었다. 이에 기본적으로 멀티미디어 스트림 엔진인 Essence 를 이용하여 멀티미디어 데이터를 전송하고 앞에서 언급한 멀티미디어 DBMS인 '오딧세우스'를 이용하여 Essence 에서 오딧세우스를 이용하여 멀티미디어 데이터를 저장하거나 질의문을 실행하여 결과 값을 받을 수 있는 Essence DBMS 구현과 Essence 에서 Database 에 관하여 모듈

화하여 라이브러리로 구현하여 멀티미디어 데이터를 실시간으로 데이터베이스에 저장하도록 하겠다.

2. 시스템 요구사항

2.1. 전체적 기능에 대한 요구사항

본 시스템은 멀티미디어 스트림엔진(Essence)을 기반으로 하여 멀티미디어 편집기를 이용하여 멀티미디어 메일을 작성하여 데이터베이스인 '오딧세우스'에 저장하도록 한다.



<그림 1> 전체 시스템

비디오 파일이나 Prs 파일이 여러 개 있다고 가정했을 때 본 시스템은 다음과 같은 가정을 설정해 보자.

Vi: 중복, Prs i: 중복 $1 \leq i \leq n$.

첫째, Prs i 를 연출할 때, $1 \leq i \leq n$. 예를 들어 홍보물인 경우, 극단적인 경우 특정 Prs 파일 k 를 모든 클라이언트가 연출할 때, 또는 Prs 파일이 전부 다를 경우, 이를 만족해야 한다.

둘째, 서비스의 타입과 재생되는 시간에 따른 QoS 를 만족해야 한다. 만약 Live 이면 1 개의 실제 데이터를 멀티캐스팅(multicasting)기술이 요구되며 재생되는 시간차를 극복해야 한다. 이미 연출되어진 이전의 것은 보이지 않는다. On-Demand 인 경우 재생시간은 처음부터 보여지게 된다. 그리고 자료 검색을 할 수 있는 인터페이스가 요구되어 진다.

위와 같은 조건을 만족하기 위해 아래와 같은 요구사항을 만족해야 한다.

첫째, 연속된 멀티미디어 데이터의 저장이 가능해야 한다. 이미 저장된 멀티미디어 데이터의 저장뿐만 아니라 실시간으로 생성되는 멀티미디어 데이터의 저장이 가능해야 한다.

둘째, 특정 프리젠테이션 파일을 복수의 클라이언트가 연결하여 재생 시 QoS 를 만족해야 한다.

셋째, 복수의 프리젠테이션 파일을 복수의 클라이언트가 연결하여 재생 시 QoS 를 만족해야 한다.

넷째, 클라이언트 수 증가에 따른 시스템의 성능을 만족해야 한다.

다섯째, 인터넷기반이 아닌 어플리케이션 기반에서 로

컬 PC 의 파일을 원격지 서버에 저장할 수 있는 Upload 기능이 있어야 한다.

여섯째, On-Demand 서비스 경우를 위해 자료 검색 기능이 지원해야 한다.

3. 설계 및 구현

멀티미디어 편집기인 VIP 를 이용하여 멀티미디어 스트림을 구성하여 연출 데이터를 '오디세우스'를 이용하여 대용량 멀티미디어 데이터를 저장하고자 한다. 이를 위해 개발방향은 첫째, Essence 에 Database 에 대한 정보를 가지는 미디움을 구현해야 한다. Essence 는 실시간으로 멀티미디어 데이터를 전송해 주는 스트림엔진으로 데이터베이스에 대한 정보를 지원해주는 모듈이 아직 없다. 이 모듈을 스트림으로 구성될 수 있는 부분을 미디움으로 제작해야 한다. 둘째, Essence 에서 스트림으로 구성이 안 되는 질의문에 대하여 라이브러리로 모듈화 하여 Essence 가 데이터베이스를 동작하게끔 지원할 수 있는 라이브러리를 구현해야 한다. 다음은 미디움에서 지원하는 Essence API 와 DBManger API 를 정의해 보았다.

3.1. DBMedium API

int DBMedium::Open();

데이터베이스에 대한 초기화 단계와 재생하기 전의 준비과정 단계이다.

int DBMedium::Play();

실제 스트림의 데이터를 가져와 데이터 베이스에 저장하거나 데이터베이스에 있는 데이터를 읽어오는 일을 수행한다.

int DBMedium::Stop();

수행 중이든 일을 멈춰 데이터베이스의 트랜잭션 (Transaction)을 멈추는 단계이다.

int DBMedium::Close();

DBMedium 의 Close()는 데이터베이스의 트랜잭션을 정상적으로 종료시키고 데이터베이스를 디스마운트 시킨다.

3.2. DBManager API

void DBManager::SetDBName(char* pstrDatabase);

데이터베이스의 이름을 지정할 때 호출한다.

void DBManager::SetFileName(char* pFilename);

파일 이름을 지정할 때 호출한다.

void DBManager::SetValidTerm(char* pValidTerm);

유효기간을 설정할 때 지정한다.

void DBManager::SetFileDescription(char* pFiledescription);

파일에 대한 설명을 지정할 때 호출한다.

void DBManager::SetProtection(char* pstrProtection);

파일에 대한 사용자 권한을 설정할 때 호출한다.

void DBManager::SetFindMode(char* pfindmode, char* pkeyword);

Find 시 Find 의 모드 설정과 키워드를 지정할 때 호출한다. 모드 값은 만든지 중심으로 찾을 때는 "USERID"값을 파일 이름으로 찾을 때는 "FILENAME"값을 지정하면 된다. 키워드 값은 찾고자 하는 키워드를 입력하면 된다.

