자바를 이용한 멀티캐스트 스트리밍 시스템 구현

지 일 구○ 차 호 정
광운대학교 전자계산학과

Implementation of a Java-based Multicast Streaming System

Ilgoo Jhi and Hojung Cha
Dept. of Computer Science, Kwangwoon University

요 약

본 논문에서는 자바를 사용한 고화질의 MPEG-1 데이타를 전송하는 멀티캐스트 스트리밍 시스템의 구현을 기술한다. 서버는 다 채널의 MPEG-1 스트림 전송을 지원하며 각 채널의 QoS를 보장하기 위한 스케쥴링 정책을 사용한다. 서버 구현에는 실시간을 지원하는 Java Real Time 패키지를 사용하였고, 클라이언트는 Sun사의 Java Media Framework 패키지를 사용하였다.

1 서론

최근 프로세서와 통신망 관련 기술의 발달로 인해 각종 멀티미디어에 대한 관심이 높아지고 있으며 이러한 추세에 따라 멀티미디어 서비스에 대한 요구와 필요성이 대두되고 있다. 멀티미디어 서비스의 방법에는 사용자마다 일정한 네트워크 대역폭을 차지하는 일대일의 형태와 단일 스트림의 데이타만으로 많은 사용자에게 서비스를 할 수 있는 일대다의 형태가 있다. 멀티미디어 데이타는 단순한 문자나 그림보다 그 데이타의 크기가 매우 크기 때문에 실제로 서비스할 때 필요로하는 네트워크 대역폭이 매우 커진다 따라서 광범위한 서비스를 위해서는 네트워크 대역폭을 차지하는 양이 작은 일대다의 형태가 더 적합하다. 이러한 일대다의 서비스를 제공하는 대표적인 예로 MBone[1]을 들 수 있다 MBone은 현재의 인터넷이 갖는 한계를 극복하고 영상과 음향정보 등을 포함한 멀티미디어 정보를 실시간으로 전송하기 위한 가상망이다. MBone에서는 멀티캐스팅을 사용하여 멀티미디어 정보를 다수의 전송대상에게 전송한다 멀티캐스팅을 이용해서 정보를 전송하는 경우에 정보전송에 필요한 대역폭의 축소, 통신망에서의 동시성 및 전송비용의 절감의 장점이 있다.

멀티미디어 서비스는 실시간에 서비스를 제공할 수 있도록 보장할 수 있는 QoS(Quality of Service)에 대한 고려가 반드시 필요하다. 예를들어, 화상회의와 같은 서비스에서 대화자들의 음성정보와 영상정보가 실시간에 모든 대화자들에게 전송되지 못한다면 회의가 이루어질 수 없다. 전송에 걸리는 시간뿐 아니라 서비스의 품질, 요구하는 네트워크 대역폭과 같은 QoS와 관련된 문제들이 있으며 이에 대한 고려가 포함되어야 한다. 따라서 멀티미디어 서비스를 위한 서버를 설계할 때 실시간에 QoS를 보장하면서도, 환경에 종속적이지 않은 모델을 제시하여야 한다.

본 논문에서는 자바를 사용한 고화질의 MPEG-1 데이타를 멀티캐스트 스트리밍 방식으로 전송하는 시스템의 구현을 기술한다. 시스템은 데이타를 클라이언트에 전송하는 서버와 전송받은 데이타를 재생하는 클라이언트로 구성된다. 서버에서는 다 채널의 스트림 전송을 지원하며, 각 채널의 QoS를 보장하기 위한 스케쥴링 정책을 제시한다 서버는 실시간에 QoS를 보장하기 위해서 자바를 확장시켜서 실시간을 보장하는 Young[2][3]의 Java Real Time 클래스를 사용하였다 클라이언트는 고속연산을 필요로 하는 멀티미디어를 지원하기 위하여 Sun사의 Java Media Framework[4] 멀티미디어 API 패키지를 사용하였다. [1]

논문의 구성은 다음과 같다. 2장에서 전체 시스템의 개요를 설명한다 3장에서는 서버와 클라이언트의 설계와 구현에 대하여 서술하고 실행결과를 보인다. 4장에서는 결론을 맺는다.

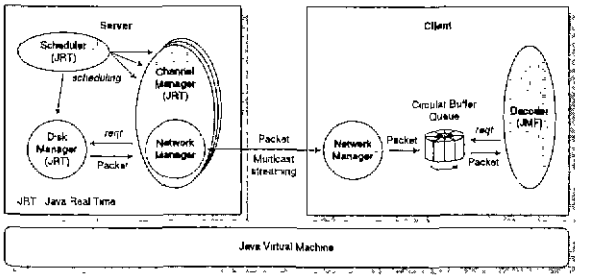

그림 1: JMS의 구조

2 시스템 개요

[그림1]은 JMS(Java Multicast System)의 구성을 나타낸다. JMS의 서버는 다 채널의 멀티캐스팅을 제공하는 서버로 각 채널의 QoS를 보장하기 위한 스케쥴러가 있다. 그리고 채널을 관리하는 채널 관리자(Channel Manager), 채널 관리자에 종속되어 멀티미디어 데이타를 전송하는 네트워크 관리자(Network Manager)가 있다 디스크 관리자(Disk Manager)는 네트워크 관리자의 디스크 읽기 요구를 받아 디스크를 읽어서 네트워크 관리자의 버퍼를 다시 채워주는 역할을 맡는다. 또한 현재 전송되고 있는 멀티미디어 데이타의 확인을 위한 바쥬얼 윈도우와 각 채널 관리자를 예약하여 생성할 수 있는 프로그램 예약 기능을 가지고 있다. 대상 멀티미디어 데이타로는 MPEG-1 System회일을 사용하였다

서버는 실시간 QoS를 보장하기 위해서 전역주기라고 불리우는 스케쥴러의 주기와 지역주기라고 불리우는 채널관리자의 주기에 의해서 동작한다 전역주기는 스케쥴러가 프로세서를 할당받고 스케쥴링을 한 후, 블럭되었다가 다시 프로세서를 할당 받을 때까지의 주기를 말한다. 지역주기는 각각의 채널 관리자들이 각각의 데이타를 전송해야 하는 데이타율에 따른 주기를 말한다 스케쥴러는 각 채널 관리자들의 지역주기보다 충분히 짧은 전역주기에 의해 동작하는데 이는 한 지역주기내에 동일한 채널 관리자의 요구가 다수개 들어오는 것을 방지 하기 위함이다. 동일한 채널 관리자의 요구가 다수개 들어오게 되면 서버시스템의

[1]본 연구는 한국과학재단의 특정기초연구사업(과제번호:97-01-00-12-01-5)에 의해 지원받았음

자원의 한계량인 네트워크 바운드나 디스크 바운드를 벗어나는 양의 요구가 들어오게 되어, 각 채널의 QoS를 보장할 수 없게 된다.

사용자가 채널 생성 대화상자를 통해서 멀티캐스트 IP, 포트 번호, 서비스 시작시간과 마감시간, 서비스 채널 번호, 제목등의 정보를 입력하고 MPEG-1 System데이타를 선택한다. 선택된 데이타의 헤더를 검사하여 평균비트율을 계산해 내고, 이 정보를 사용하여 채널 관리자를 생성한다. 채널 관리자는 고유의 네트워크 관리자를 함께 생성하여 데이타 전송에 사용하고, 데이타 전송이후에 디스크 관리자를 통해 디스크에서 데이타를 읽어 채널 관리자의 버퍼를 다시 채운다. 채널 관리자는 서비스 시작시간을 통한 예약으로 생성될 수도 있고 정보입력 즉시 생성될 수도 있다. 채널 관리자가 생성되면 스케줄러에 관련정보가 등록되어 스케줄링시에 사용된다.

JMS의 클라이언트는 서버에서 전송하는 멀티캐스트 스트림을 받아서 원형 버퍼 큐에 저장하는 네트워크 관리자와 이 버퍼 큐에서 데이타를 읽어 들여서 디코딩하는 디코더로 이루어져 있다. 네트워크 관리자는 블럭당하는 일이 없이 서버에서 전송하는 데이타를 받아서 버퍼에 넣는다. 디코더는 MPEG-1 System데이타를 디코딩하는 비율대로 버퍼에 데이타를 요구하고 버퍼에서 데이타를 읽어들인다. 네트워크 관리자를 통해서 전송받는 데이타의 양과 디코더에서 읽어서 디코딩한 데이타의 양을 가지고 버퍼큐의 관리에 사용한다.

3 설계 및 구현

다음은 JMS의 스케줄링모델과 구현에 대해서 설명하고, 클라이언트의 구성과 구현을 기술한다.

3.1 서버

JMS는 클라이언트의 요구를 수용하여 멀티미디어 데이타를 전송하는 것이 아니라, 서버쪽에서 서비스할 채널의 갯수를 정하여 데이타를 전송하는 푸쉬(push) 시스템이다 따라서 몇개의 채널을 서비스할 것인가를 서버쪽에서 결정하여 채널을 생성해야 한다. 이에 대한 기준이 되는 것은 디스크 바운드와 네트워크 바운드이다. 한 전역주기에 처리할 요구의 데이타양이 디스크 바운드와 네트워크 바운드 중에 작은 값보다 커지게 되는 경우가 발생하게 되면 각 채널 관리자의 QoS를 보장할 수 없게 된다 따라서, 처음에 채널 관리자의 수를 몇개로 할 것이가를 결정할 때 디스크 바운드와 네트워크 바운드를 계산해서 최악의 경우의 처리요구를 수용할 수 있을 만큼의 채널 관리자를 생성해야 한다 최악의 경우의 처리요구는 한 전역주기내에 각 채널 관리자들이 모두 한번씩 요구를 한 경우이다.

다 채널의 멀티미디어 데이타를 전송하면서 각 채널의 QoS를 실시간에 보장해주기 위해서는 각 채널이 서버의 자원을 독점하지 않고, 데이타율에 따른 채널의 주기에 따라 전송할 수 있도록 해야한다 이를 위해서 스케줄러는 전역주기에 따라 동작하며, 각 주기마다 작업들의 순서를 조절하여 각각의 채널 관리자들의 데이타율에 따른 지역주기에 필요한 양의 데이타를 전송할 수 있도록 스케줄링을 한다

[그림2]는 스케줄러와 채널 관리자의 스케줄 모델을 나타낸다. 각 채널 관리자들은 각각의 지역주기에 따라서 요구큐에 요구를 보낸 후 블럭되고, 스케줄러는 요구큐를 검사하여 각각의 요구에 따라서 프로세서를 할당한다. 스케줄링은 채널 관리자와 네트워크 관리자, 디스크 관리자의 순서조정을 통해서 수행된다. 스케줄러의 전역주기내에 가장 많은 요구가 들어올 때라도 총 채널 관리자의 수와 같거나 작기 때문에 각 채널 관리자가 프로세서를 독점하지 않도록 하면 각 채널 관리자의 QoS를 보장할 수 있다. 따라서 한 사이클의 전역주기에 요구큐에 들어온 채널 관리자들의 요구들은 그 순서에 따라서 프로세서를 할당 받는다. 작업이 끝난 관리자들은 다음 지역주기에 의해 활성화 될 때까지 블럭된다.

JMS에서 네트워크 전송에 사용하는 데이타 패킷의 크기는

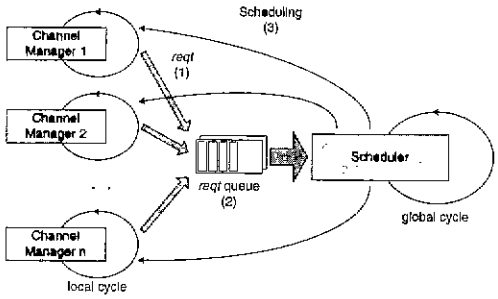

그림 2: 스케줄러와 채널 관리자의 스케줄 모델

$32KBytes$이다. 이 데이타 패킷의 크기를 기준으로 해서 주기를 계산할 수 있다. 전역주기는 최대 데이타율을 갖는 채널 관리자의 지역주기 보다 짧게 설정하여야 하므로, 실험데이타로 사용하는 MPEG-1 System데이타의 최대 평균비트율인 $192KBytes/sec$를 가지고 계산한다. 마찬가지로 각각의 채널 관리자의 지역주기도, 처음 채널 관리자를 생성할 때 MPEG-1 System데이타의 헤더를 분석해서 계산해 낸 평균비트율로 계산한다.

JMS를 구현하기 위하여 컴파일러로는 JDK1.1.6을 사용하였고, 유저인터페이스는 Swing[5]패키지를, 그리고 서비스하고 있는 데이타 확인을 위한 디코더로 JMF를 사용하였다. 스케줄러와 각 관리자들의 실시간을 보장하기 위해서 Java Real Time 패키지를 사용하여 구현하였다.

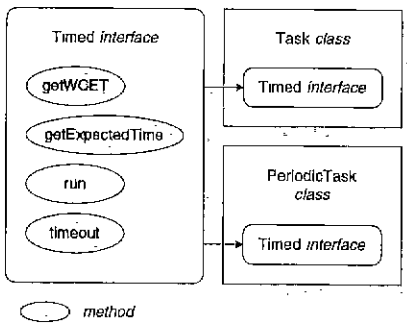

그림 3: Java Real Time 패키지의 구성

[그림3]은 Java Real Time의 구성을 나타낸다. Java Real Time 패키지는 Task, PeriodicTask라는 두가지 클래스와 Timed라는 인터페이스로 구성되어 있고, Timed인터페이스 안에는 시간 제약을 설정하기 위한 getWCET(), getExpectedTime(), timeout(), run()이라는 메소드를 제공한다 스케줄러와 각 관리자들은 주기적인 작업에 해당하므로 PeriodicTask클래스를 사용하였다

3.2 클라이언트

JMS의 클라이언트는 대상 실험데이타로 사용한 MPEG-1 System 데이타를 디코딩할 수 있는 디코더와 멀티캐스트 스트림을 전송받는 네트워크 관리자로 구성된다

클라이언트는 네트워크 관리자를 통해서 서버에서 멀티캐스트 스트림으로 전송하는 MPEG-1데이타를 받아서 원형 버퍼큐에 넣고, 디코더는 원형 버퍼큐의 데이타를 읽어 들여서 디코딩하여 화면에 보여준다. 네트워크 관리자는 주기적으로 전송되어 오는 멀티캐스트 스트림을 받아서 버퍼큐에 넣는 역할을

하게 되는데, 이때에 버퍼큐의 시작과 끝의 정보를 갱신할 때 디코더와 네트워크 관리자간의 동기가 필요하다. 클라이언트의 구현을 위해서 자바언어를 사용하였다. JDK1.1.6을 사용하여 컴파일을 하였고, Swing을 사용하여 유저인터페이스를, MPEG-1 System데이타의 디코딩을 위해서 JMF를 사용하였다. 클라이언트 구현에 사용된 JMF는 미디어 데이타를 자바 어플리케이션이나 애플릿에 통합하기 위한 API이다 JMF는 Java Media Player, Capture, Conferencing의 세부분으로 나누어지는데, 클라이언트는 Java Media Player를 사용하여 구현하였다

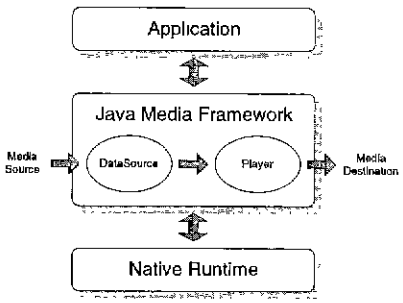

그림 4: JMF의 구조

[그림4]는 JMF의 구조를 보인다. Java Media Player는 DataSource와 Player로 구성된다. 데이타소스는 미디어의 위치나 전송에 사용되는 프로토콜등을 정의하고 있으며, 플레이어에서는 데이타소스를 통해 전송받은 데이타를 랜더링해 사용자에게 보여준다. 데이타소스 부분을 수정하여 설계한 원형 버퍼큐에서 데이타를 읽어 랜더링하도록 하였다. 클라이언트의 플레이와 포즈등의 디코더의 콘트롤은 JMF에서 제공해 주는 메소드를 사용하였다.

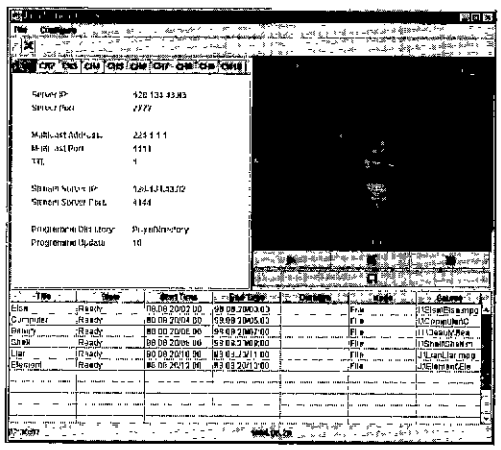

그림 5: 서버의 실행모습(Windows NT)

3.3 실행결과

[그림5]는 서버의 실행모습을 보여준다 서버는 툴바와 각 채널 관리자의 예약을 표시하는 표, 그리고 각 채널의 정보를 보여주는 패널과 실제 각 채널에서 전송되고 있는 멀티미디어 데이타를 디코딩해서 보여주는 비쥬얼 윈도우를 가지고 있다. 서버의 툴바에는 예약된 프로그램 채널을 취소할 수 있는 버튼이 있고, 메뉴바에는 File, Configure 메뉴가 있다 File메뉴에는 시스템을

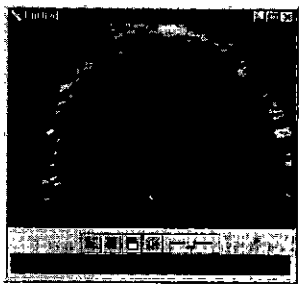

그림 6 클라이언트의 실행모습(Solaris 2 5.1)

종료하는 메뉴와 멀티미디어 데이타를 선택하는 메뉴가 있고, Configure에는 시스템의 환경을 셋팅하는 메뉴와 프로그램 채널을 예약하는 메뉴가 있다. 서버왼쪽의 채널의 정보를 보여주는 패널을 선택하면 현재 예약되거나 서비스중인 프로그램 채널의 정보를 보여준다

[그림6]은 클라이언트를 Solaris 2.5.1환경에서 실행시킨 모습을 보인다. 클라이언트는 랜더링된 멀티미디어 데이타를 보여주는 비쥬얼 화면과 클라이언트의 동작을 선택하는 버튼들, 그리고 클라이언트의 상태를 보여주는 상태패널로 구성된다. 동작을 선택하는 버튼은 왼쪽부터 재생, 일시멈춤, 클라이언트를 항상위로, 스피커 볼륨조절 버튼으로 구성되어 있다 클라이언트의 아래쪽의 패널은 현재 클라이언트의 상태, 클라이언트가 재생하고 있는 시간, 네트워크를 통해 받는 데이타율을 표시한다.

4 결론

본 논문에서는 특정 플랫폼에 종속되지 않는 모델을 설계하고, 다양한 환경에서 수정없이 사용할 수 있는 자바언어를 사용하여 다 채널을 갖는 멀티캐스트 스트리밍 서버를 구현하였다. 각 채널은 고유의 데이타 주기를 갖기 때문에 각 채널들의 QoS를 보장하기 위하여 스케쥴링 정책을 제시하고, 실시간 기능이 추가된 Java Real Time 패키지를 사용하여 구현하였다 다양한 환경에서 수행할 수 있는 클라이언트를 함께 구현하여 실험실에서 뿐만 아니라 인트라넷 환경에서 사용가능성을 보였다. 다 채널을 지원하기 때문에 동시에 여러 멀티미디어 데이타를 전송할 수 있으며, 각 채널의 스케쥴링 정책에 의해 QoS를 보장 받을 수 있다.

향후 연구 과제는 네트워크의 트래픽시에도 멀티캐스트 데이타를 지연없이 실시간에 전송할 수 있도록 QoS를 보장하는 것이다. 이를 위해서 실시간 통신 프로토콜에 관한 연구가 필요하다

참고 문헌

[1] MBone Korea Home Page, *http://paro.etri.re.kr*

[2] James Shin Young, "Jim's Java Joint", *http:// cad-ntws11.eecs. berkeley.edu/ java/ index.html*

[3] James Shin Young, 'Integration of a Real-time Programming System for Dynamic Reactive Systems in an Object-Oriented Language', *Berkeley Univ. Master of Science Report*, 1996

[4] Sun Java Media Framework API Home Page, *http:// java.sun.com/ products/ java-media/ jmf/ index.html*

[5] The Swing Connection, *http:// java.sun.com/ products/ jfc/ tsc/ index.html*