

다중 주문형 비디오 서버 시스템의 설계 및 구현[†]

최우성^o 장영배 김성완 오승택 김호중 맹승렬 조정완
한국과학기술원 전자전산학과 전산화전공
{wschoi,ybjang,swkim,stoh,hjkim,maeng,jwcho}@camars.kaist.ac.kr_

Design and Implementation of a Multiple Video-on-Demand Server Systems

Woo-Seong Choi^o Young-Bae Jang Sung-Wan Kim Seung-Taek Oh Ho-joong Kim
Seung-Ryol Maeng Jung-Wan Cho
Division of Computer Science, Dept. of EECS, KAIST

요 약

오늘날 네트워크의 발달로 인하여 사용자가 원하는 영상매체를 실시간으로 요구하는 멀티미디어 서비스에 대한 사용자의 요구가 날로 증가하고 있다. 때문에 영상매체를 제공하는 서버는 여러 사용자들이 동시에 접속하였을 때 서비스를 제공할 수 있는 능력을 가져야 한다. 본 논문에서는 멀티미디어 서비스를 위한 VOD(Video-On-Demand) 시스템을 설계 및 구현한다. VOD 시스템은 미디어를 분석하여 BES(Back-End Server)에 저장하는 영화 관리자(Movie Manager), 본산 저장된 데이터를 읽어 들여 사용자나 연결된 FES(Front-End Server)로 보내는 BES, 사용자에게 미디어 데이터를 전송하는 FES와 전송된 데이터를 받아서 재생하는 클라이언트(Client)로 구성 된다. 또한 다수의 서버를 연결하기 위하여 SCI(Scalable Coherent Interface)를 이용한 효율적인 메시지 전송방법을 구현한다.

1. 서론

ISDN과 ADSL 등의 고속 네트워크의 보급으로 인하여 사용자들은 영상매체를 실시간으로 제공 받기를 원하게 되었다. 사용자의 요구에 의한 주문형 비디오(VOD, Video-on-Demand)서버의 특징은 동시에 여러 사용자의 요구를 수용해야 하는데 있다. 그리고 사용자가 요구하는 멀티미디어 데이터는 그 형태가 다양하기 때문에 이를 고려한 서버의 개발이 필요하다. 이를 위해 다수의 미디어 서버로부터 다양한 대역폭을 요구하는 미디어 데이터를 동시성을 잃지 않고 가능한 한 많은 사용자들에게 제공해 주어야 한다. 병렬 연산(parallel computing)환경은 이러한 요구에 적합한 환경이라 할 수 있다.[1]

본 논문에서는 병렬적으로 동작하는 미디어 서버 시스템을 제안한다. 또한 미디어 서버 시스템의 성능을 최대한 활용할 수 있도록 미디어 서버들 사이의 부하 배분(load balancing) 정책과 입출력 및 프로세스 스케줄링 방법 등을 설계하고 구현한다. 사용자에게 효과적으로 다양한 미디어 데이터를 제공해주기 위하여 통신요소 시스템 사이의 효율적인 통합 메커니즘을 지원할 수 있도록 복합형 분산 공유메모리 시스템을 개발한다. 효과적인 통신 메커니즘의 설계를 위하여, 복합형 분산 공유메모리 시스템을 기반으로 한 통신 시스템을 개발하여, 서버들 사이의 통신이 원활하게 이루어 지도록 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 서버들 사이의 부하배분을 위하여 사용되는 공유메모리의 구성에 관하여 간략히 살펴 보고, 3 장에서는 시스템의 개요와 구조에 대하여 설명을 한다. 4 장에서는 구현 방법과 구현한 시스템의 성능을 평가하고, 5 장에서는 결론을 맺는다.

2. 서버사이의 효율적인 통신 모듈의 구현

2.1. 개요

서버 통신 모듈은 FES(Front-End Server)와 BES(Back-End Server), 영화 관리자 서버 사이의 SCI(Scalable Coherent Interface)를 이용한 통신을 담당하는 모듈이다. 서버 통신 모듈은 한 서버로부터 다른 서버로의 스트림(stream) 요청/응답/전송 명령을 SCI에서 사용하는 공유 메모리 읽기/쓰기, DMA 전송 명령으로 변환한다. 공유 메모리 읽기/쓰기는 작은 크기의 데이터를 전송하는 데 적합한 반면 DMA 전송은 커다란 크기의 데이터를 전송하는 데 적합하므로, 서버간의 요청/응답 메시지는 공유 메모리 읽기/쓰기로 처리하고 스트림 전송은 DMA로 처리한다.[2]

2.2. 구조

서버 통신 모듈은 SCI 공유 메모리를 관리하는 메모리 세그먼트 객

체와, 세그먼트 객체에서 지정하는 메모리 영역 상에서 동작하는 큐, 배열 객체로 구성된다.

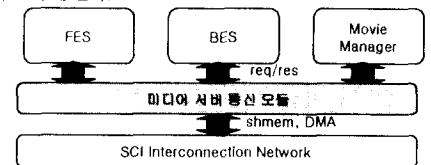


그림1. 서버 통신 모듈

메모리 세그먼트 객체는 공유 메모리 영역을 관리한다. 각 세그먼트 객체는 특정 메모리 영역을 유지하는데, 이 주소는 지역 노드의 물리적 메모리 영역을 가리키거나, 원격 노드의 메모리를 사상(mapping)한 가상 메모리 영역을 가리킨다. 전자는 지역 공유 메모리 읽기/쓰기, DMA 읽기/쓰기 용으로 생성되며, 후자는 원격 공유 메모리 읽기/쓰기 용으로 선언된다.

하나의 메모리 세그먼트는 다른 노드의 메모리 세그먼트와 통신하기 위하여 그 세그먼트에 접속하여야 한다. 접속 방식은 원본과 대상 세그먼트가 DMA 전송용인가 또는 공유 메모리 접근용인가에 따라 다르다. 공유 메모리 세그먼트의 경우 지역 노드에 존재하는 원본 세그먼트는 물리적 메모리를 가지지 않으며, 접속 후 원본 세그먼트로의 메모리 읽기/쓰기 접근은 SCI 인터페이스에 의하여 원격 노드의 대상 세그먼트에서 수행된다. DMA 세그먼트의 경우 접속하는 원본 세그먼트와 접속된 대상 세그먼트는 모두 물리적 메모리를 가지며, DMA 전송시 원본 세그먼트의 메모리 내용이 대상 세그먼트의 메모리 영역에 복사된다. SCI의 DMA 전송 명령은 대상 노드와 세그먼트의 정보를 필요로 하므로, 하나의 원본 세그먼트를 여러 대상 세그먼트에 접속한 후 필요에 따라 대상 세그먼트를 지정하여 DMA로 전송할 수 있다.

공유 큐와 배열 객체는 공유 메모리 세그먼트를 바탕으로 원격 노드 간에 큐와 배열을 구현한 객체이다. 공유 메모리를 이용한 통신은 단순히 메모리 읽기/쓰기로 이루어지므로 실제 메시지 전송을 제공하기 위해서는 상위 데이터 구조를 구현하여야 한다. 이러한 상위 수준 구현의 한 방법으로 큐와 배열을 공유 메모리 상에 구현 함으로서, 서버 프로세스 간의 요청/응답과 같이 고정된 형태의 메시지를 쉽게 전송할 수 있다. 큐는 메시지들의 순서를 유지하여 전송하므로 FES에서 BES로의 스트림 요청에 적합하며, 배열은 메시지 순서를 고려하지 않으므로 FES가 BES의 응답을 기다리며 폴링하는 경우에 적합하다.

3. 미디어 서버 및 클라이언트의 구현

[†] 이 연구는 인공지능연구센터(CAIR)의 지원을 받아 이루어졌다.

3.1. 영화 관리자(Movie Manager)

3.1.1. 개요 및 구조

영화 관리자는 여러 개의 노드(node)에 흩어져 있는 미디어 데이터의 통합적인 관리를 위하여 구현되었다. 영화 관리자는 서버 전체의 영화를 중앙 집중적인 관리를 위하여 서버의 미디어 데이터 목록 검색, 새로운 영화의 삽입, 기존 영화의 삭제 등의 기능을 담고 있어야 한다.

영화 관리자는 사용자 인터페이스, 미디어 데이터 베이스, 미디어 분석 및 분배기의 세 가지로 이루어져 있다. 사용자 인터페이스는 사용자로부터 입력을 받고 영화에 대한 정보를 출력하는 일을 담당한다. 미디어 데이터 베이스는 서버에 저장된 미디어 데이터에 대한 관리를 담당하면서 메인 윈도우나 미디어 분석 및 분배기가 미디어 데이터에 대한 질의를 한 경우 그에 해당하는 답을 한다. 본 서버의 미디어 데이터 베이스는 기존의 상용 데이터 베이스를 사용하지 않고 간단한 데이터 베이스를 구현하여 사용하였다. 따라서 미디어 데이터 베이스는 크기가 작고 영화 관리자에 최적화 되어 있어서 다른 모듈(module)의 질의에 빨리 동작할 수 있도록 구성하였다. 미디어 분석 및 분배기는 새로운 영화를 삽입할 때 이용한다. 사용자로부터 영화 삽입에 관한 요청이 들어오면 미디어 분석 및 분배기는 지정된 파일을 열고 BES와 연결한 후에 미디어 데이터를 분석하여 보내주는 역할을 한다. BES와의 연결은 SCI를 이용한 원격 메모리 접근(remote memory access)방식을 택하고 있다.

3.1.2. 미디어 데이터의 분석

미디어 분석 및 분배기의 동작 중 미디어 데이터의 분석 방법에 대하여 살펴본다. 본 미디어 서버는 MPEG-1과 MPEG-2로 암호화(encoding)된 미디어 데이터를 기반으로 한다. 따라서 영화 관리자는 사용자의 요청에 따라 MPEG-1과 MPEG-2로 암호화 된 데이터를 분석하고 적당한 형식으로 다시 만드는 작업을 할 수 있어야 한다.

MPEG 데이터는 프레임(frame, 한 장의 그림)을 기본 단위로 하며, 프레임들 여러 개가 모여 하나의 GOP(group of pictures)를 형성하고 있다. GOP는 하나 이상의 I 프레임이 있어서 독립적으로 복호가 가능한 그림들의 모임이다. 하나의 GOP가 몇 개의 프레임으로 구성되었는지에 대하여 정해진 바는 없지만 MPEG-1에서는 15개의 프레임을 모아서 하나의 GOP를 만드는 것이 가장 일반적인 방식이다. 또한 MPEG-1에서는 1초를 30프레임으로 구성하는 것이 가장 일반적인 방법이므로 보통 1 GOP는 0.5초 분량의 영상을 담고 있다고 할 수 있다.

본 미디어 서버에서는 이러한 구조의 MPEG 미디어 데이터를 1초 분량으로 나누어서 각각의 BES에 돌아가며 저장하는 방식을 취하고 있다. 따라서 미디어 분석 및 분배기는 2개의 GOP를 하나의 세그먼트(segment)의 단위로 하여 조개는 작업을 수행하여야 한다. 또한 이렇게 세그먼트 단위로 조개진 데이터들은 광역 스트라이핑(striping) 방법으로 여러 개의 BES에 보내져서 저장된다.[3]

3.2. BES

3.2.1. 개요

BES는 영화 관리자로부터 분석된 미디어 스트림을 분산 저장하고, FES에서 영화 프레임 요구를 받아들여 원하는 프레임을 FES에 전달하는 기능을 수행한다. 각 서버와의 통신은 공유 메모리를 기반으로 한 공유 큐(Shared Queue)를 사용하여 이루어진다. 그리고 미디어의 프레임은 대용량의 자료로 이루어 지므로 원격 DMA 전송을 통하여 전송된다.

3.2.2. 구조

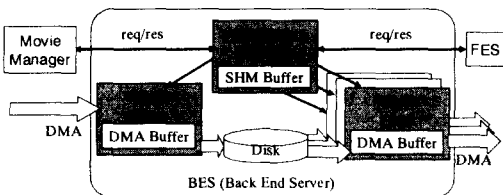


그림2. Back-End 서버의 전체 구조도

BES에는 크게 세 가지의 쓰레드(Thread)가 동작한다. 각 서버에서 전

송된 요구를 분석하고 적절한 명령을 내리는 메시지 처리 쓰레드(Message Handler Thread), 영화 관리자와 연결되어 미디어 스트림을 분산 저장하는 영화 저장 쓰레드(Movie Store Thread), 그리고 FES와 연결되어 요청된 프레임을 디스크에서 읽어 DMA를 이용하여 FES에 전달하는 영화 읽기 쓰레드(Movie Load Thread)가 BES를 구성한다.

메시지 처리 쓰레드의 목적은 다른 서버(영화 관리자, FES)에서 요청되는 메시지를 받아들이고, 그 내용에 따라 적절한 쓰레드에게 전달하는 것이다. 이를 위하여 메시지를 저장하고 꺼낼 수 있도록 원격 메모리를 이용한 공유 큐를 각 서버별로 설정한다. 공유 큐에 저장된 메시지는 메시지 분석기(Message Analyzer)를 통하여 적절한 쓰레드의 지역 큐에 저장한다. 이를 구현하기 위하여 원격 메모리 쓰기가 가능한 공유 큐를 서버의 개수만큼 SCI 명령을 이용하여 설정한다. 또한 각 쓰레드에게 명령을 전달하기 위하여 각 쓰레드 별로 메시지 큐를 갖게 하고 메시지 처리 쓰레드는 메시지 큐에 분석된 메시지를 저장하는 것으로 명령 전달을 끝내게 된다.

영화 저장 쓰레드는 미디어 분석기에서 전송되는 메시지를 분석하고 그에 따른 영화 프레임 정보를 디스크에 분산 저장하는 역할을 수행한다. 영화 관리자와의 통신은 메시지를 받기 위한 공유 큐와 영화 프레임 큐를 위한 DMA 버퍼를 이용하여 이루어진다. 하나의 영화는 한 서버로의 부하 집중을 방지하기 위하여 각각의 BES에 순차적으로 분산되어 전달되고, 하나의 BES에서 받아들인 영화 프레임도 하나의 디스크에 부하가 집중되지 않도록 보유하고 있는 SCSI 디스크에 순차적으로 분산하여 저장한다. 이러한 방법은 RAID와 유사한 형태로서 부하 분산의 관점에서 시스템의 자원을 고르게 활용할 수 있게 한다.

영화 읽기 쓰레드는 FES로부터 전송된 사용자의 요구에 의한 영화 프레임을 서비스하기 위해 FES의 메시지를 분석하고 그에 따른 적절한 영화를 전송하는 역할을 수행한다. 미디어 분석기로부터 받은 영화 프레임들은 각 BES별로 적절히 분산되어 있기 때문에 FES에서 사용자가 요구하는 프레임이 존재하는 BES에게 프레임 전송을 요청하게 된다. 공유 큐를 이용하여 프레임 요구를 받은 BES는 자신의 여러 디스크 중 어느 디스크에 저장되어 있는지를 계산하고, 영화 저장 쓰레드에서 만들어 두었던 프레임 정보를 사용하여 원하는 프레임의 정확한 위치를 구하여 디스크에서 읽어 DMA 전송을 통해 FES에게 전달한다.

3.3. FES

3.3.1. 개요

부하 분산을 위해 미디어 데이터를 여러 BES에 분산하여 저장한 데이터를 사용자에게 하나의 서버로 투명하게 보이기 위해 사용자와의 인터페이스(interface) 역할을 담당하는 FES가 필요하게 된다. FES는 사용자 서비스 요구 수용 및 처리, 사용자 스케줄링, BES로의 미디어 데이터 요구, 클라이언트로 미디어 전송의 역할을 담당한다.

3.3.2. 구조

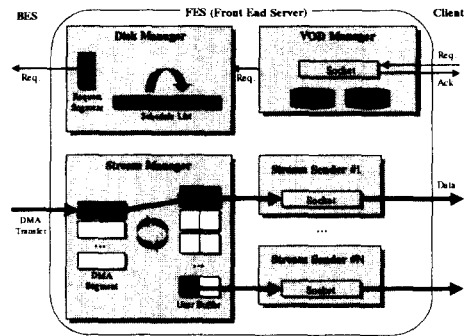


그림3. FES의 구조도

FES는 전체적으로 VOD Manager, Disk Manager, Stream Manager, Stream Sender의 크게 네 부분으로 구성된다. VOD Manager는 사용자의 각종 서비스 요구를 네트워크로부터 실시간으로 받아들여 적절한 응답을 한다. 새로운 사용자에 대해 인증(authorization) 과정을 거쳐 유효한 사용자에 대해 현재 미디어 서버가 유지하고 있는 미디어들의 리스트를 사용자에게 전달해주고 사용자와 데이터 채널(channel)을 구성한다. 사용자가 이 리스트 중에서 관심 있는 미디어를 선택하여 그 정보를 보내주면

VOD Manager는 Disk Manager에게 이 사용자에 대하여 스케줄링을 하도록 사용자 정보를 전달한다. Disk Manager는 현재 미디어 서버에 접속하여 서비스를 받고 있는 사용자들에 대하여 미디어의 실시간 서비스 요구를 만족하면서 균등하게 서비스될 수 있도록 스케줄링을 한다. 현재 서비스해야 할 사용자들 스케줄링 알고리즘을 이용해 선정된 후, BES에 이 사용자에게 보내야 할 미디어의 프레임 인덱스(frame index) 정보를 SCI기반의 링(ring) 네트워크를 통해 전달한다. Stream Manager는 각 사용자들에 대해 버퍼 블록을 유지하고 BES에서 SCI 네트워크를 통해 전달되는 미디어 블록을 해당 사용자의 버퍼 블록으로 옮긴다. 버퍼 블록이 모두 옮겨지면 해당 사용자에게 미디어 데이터를 전송하는 Stream Sender에게 데이터를 전송할 것을 알린다. Stream Sender는 사용자와 구성된 데이터 채널을 통해 미디어 데이터를 전송하게 된다.

FES와 BES는 공유 메모리의 형태로 서로의 데이터를 교환하게 된다. 각 BES로 미디어 블록의 요구(request)를 위한 요구 세그먼트(segment)와 BES에서의 응답을 위한 응답 세그먼트, 그리고 BES에서 FES로의 미디어 블록 전송을 위한 데이터 세그먼트 영역을 공유 메모리로 지정한다.

본 연구에서 미디어 서버가 지원하는 미디어 형식은 대역폭이 일정한 MPEG-1, MPEG-2이다. 미디어 서버가 유지하는 미디어의 대역폭이 일정하면 모든 사용자에 대해 우선순위를 동등하게 유지 할 수 있고 이런 스케줄링을 지원하는 가장 직관적이고 효율적인 방법이 라운드 로빈(round robin) 알고리즘이다. 현재 미디어 서버에 접속한 각 사용자들을 스케줄 리스트에 입력하고 차례대로 한 사용자씩 번갈아 가면서 서비스하는 방법이다. BES에 저장된 미디어의 단위 블록들은 1초를 기준으로 하여 때문에 한 라운드를 서비스하는 데 걸리는 시간은 최대 1초로 이다.

미디어 서버는 각각의 사용자에 대해 실시간 서비스 조건을 모두 만족 해야 하므로 이를 위해 기존에 이미 서비스되고 있는 사용자에 대한 실시간 조건을 만족시키면서 새로운 사용자의 서비스 요청을 수용할 수 있는가를 결정하는 접근 제어 알고리즘이 필요하다. 본 연구에서는 미디어 데이터의 비트 속도와 디스크의 대역폭 등을 계산하여 사용자가 데이터의 손실 없이 서비스 받을 수 있는 최대의 경우를 미리 결정하는 결정적(deterministic) 접근 제어 방법을 이용하고 있다.[4]

본 연구에서는 각 사용자들의 버퍼를 2개를 두는 2-way 버퍼링을 이용한다.[5,6] BES는 FES가 요구하는 미디어 블록이 디스크를 찾아 읽어 들여 FES로 데이터 세그먼트를 통해 DMA 전송을 한다. Stream Manager는 BES로부터 미디어 블록이 도착했는지 실시간으로 조사하여 이 블록을 필요로 하는 사용자의 버퍼 영역으로 데이터를 이동시킨다. 데이터의 이동이 완료되면 해당 사용자의 Stream Sender에 전송 신호를 주고 이 신호를 받은 Stream Sender는 2개의 버퍼 중 현재 보내야 할 버퍼 블록을 알아내어 해당 사용자에게 데이터를 전송한다.

3.4. 클라이언트

3.4.1. 개요와 구성

클라이언트는 실제로 사용자에게 영화를 보여 주는 역할을 한다. 클라이언트는 FES에서 전달되는 미디어 데이터의 헤더를 분석하여 이를 윈도우의 MCI(Multimedia Control Interface)의 API(Application Program Interface)를 이용하여 윈도우 시스템에 등록되어 있는 복호화 기기(decoding device)에 전달하는 역할을 한다.

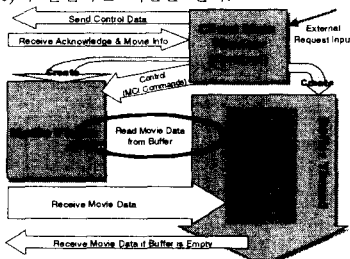


그림4. 클라이언트 구조도

클라이언트는 크게 사용자와 통신을 하는 주 윈도우(main window), 서버로부터 받은 데이터를 임시로 저장하는 버퍼, 실제로 미디어 데이터를 재생하는 매체 재생기(media player)로 나뉜다. 주 윈도우에서는 사용자들의 입력을 받아 들어서 매체 재생기에 MCI명령을 보낸다. 버퍼는 바이트 형의 배열로 패킷을 통하여 전달된 자료들을 안정적으로 재생하기 위하여 여분의 공간에 저장해 두는 것이다. 여기에 저장된 데

이터를 MCI 장치가 읽어 간다. 매체재생기에서는 전달되는 데이터를 받아들여서 복호화 기기를 이용하여 사용자에게 영상과 소리를 보여주는 역할을 한다.

4. 성능

성능 평가를 위한 환경은 다음과 같다.

	Intel Pentium III 450MHz
	128Mbytes
	4 * Quantum Fireball SE 2.1S (SCSI)
	Microsoft Windows NT workstation 4.00.1381 Service pack 3
	Dolphin Interconnect D310
	3

위의 환경에서 측정된 본 시스템의 성능은 다음과 같다.

	36 bytes	24.71 MB/sec	4.66 μsec
	256 kbytes	16.69 MB/sec	17.56 msec
	--	19.50 MB/sec	--

위의 결과에서 본 시스템의 성능을 제한하는 부분은 서버간의 영화 프레임 데이터 전송을 위한 대역폭 및 지연시간 이다. 위 수치상 한 서버가 최대한 수용할 수 있는 사용자 수(MPEG-1, 128KB/sec)는 대역폭의 경우 66.7명, 지연시간의 경우 56.9명이다. 이것은 100M Fast Ethernet에서 수용할 수 있는 최대 사용자 수인 40명(10MB/sec)을 상회하는 수치이다. 또한 영화 데이터를 재생하지 않고 소비만 하는 가상 사용자 프로그램을 작성하여 FES와 BES에 접속하여 수행해 본 결과 40명의 사용자에서 아무런 문제 없이 서비스를 제공함을 확인할 수 있었다.

5. 결론 및 전망

본 연구에서는 다중 서버 주문형 비디오에 적합하도록 서버를 분리하여 설계 하였다. 서버는 크게 영화 관리자, BES, FES로 나뉘어져 있다. 영화 관리자는 미디어를 분석하고 분할하여 BES로 전달한다. BES는 이들 데이터를 나누어서 여러 디스크에 분산시켜 저장하고 FES의 요구에 따라 자료를 FES로 보낸다. FES는 사용자에게 마치 하나의 컴퓨터를 사용하고 있는 것과 같은 환경을 제공하며, 사용자의 요구에 따라서 BES에 미디어 데이터를 요구하고 이를 클라이언트에게 보낸다. 본 연구에서 구현된 클라이언트는 MPEG-1과 MPEG-2를 지원한다.

본 연구에서 구현한 주문형 비디오 서버는 가상 클라이언트를 이용한 실험을 통하여 MPEG-1을 기준으로 40 명의 사용자를 수용 할 수 있음을 보였다. 또한 40 명의 사용자 상에서도 대역폭의 여유를 보여 이론적으로 더 많은 사용자의 요구도 수용 할 수 있는 가능성을 보였다. 전체 서버 시스템의 성능에 결정적인 영향을 미치는 부분은 서버 사이에 미디어 데이터를 전송 할 때에 생기는 대역폭의 한계와 지연 시간임을 알 수 있었다.

앞으로 서버사이의 대역폭을 개선하고 지연 시간에 대한 개선을 이룬다면 보다 나은 성능을 가진 VOD시스템을 구현 할 수 있을 것이다.

참고 문헌

[1] The Architectural Costs of Streaming I/O: A Comparison of Workstations, Clusters, and SMPs, Remzi. H. Arpaci-Dusseau, Andrea. C. Arpaci-Dusseau et al., HPCA '98
 [2] Low-Level SCI Software: Requirements, Analysis, and Functional Specification, F. Giacomini et al., SISCI, May, 1998
 [3] 후지와라 히로시, "그림으로 보는 최신 MPEG", 교보문고, 1995
 [4] T. Chiueh, M. Vernick, C. Venkatramani, "Performance Evaluation of Stony Brook Video Server", ECLSTR-24, February 1997.
 [5] J. Gafsi and E. W. Biersack, "Impact of Buffer Sharing in Multiple Disk Video Server Architecture", In Proceedings in the 6th Open Workshop on High Speed Networks, October 1997.
 [6] P. Shenoy, P. Goyal, M. Vin, "Issues in Multimedia Server Design", ACM Computing Surveys, Vol. 27, No. 4, December 1995.