

비 공유 저장장치를 가지는 클러스터 기반 서버에서 다중 디스크립션 코딩을 이용한 스트리밍 방법

박 유 현[†] · 김 학 영^{**} · 김 명 준^{***} · 김 경 석^{****}

요 약

일반적인 비 공유 저장장치를 가진 시스템에서는 특정 콘텐츠에 대한 편향된(skewed) 요청 문제를 피할 수 없다. 본 논문에서는 비 공유 저장장치 모델로 설계/구현된 클러스터 기반의 스트리밍 서버에서 사용자의 편향된 콘텐츠 요청에 대처할 수 있고 스트리밍 서비스 중인 서버 또는 NS 카드에 문제가 생기더라도 스트리밍 서비스가 중단되지 않도록 다중 스트리밍 방법인 MDC(Multiple Description Coding)를 적용한 스트리밍 전송방법을 제안한다. 제안하는 방법은 클러스터 시스템에서 단일 세션으로 스트리밍 서비스를 제공하는 것이 아니라 MDC를 활용하여 다중 세션을 통한 스트리밍 서비스를 제공함으로써 모든 서버의 부하가 같아지고, 특정 서버에 문제가 발생하더라도 스트리밍 서비스가 끊어지지 않으면서 MDC의 특성인 적응형 서비스도 가능하게 되는 장점이 있다.

키워드 : 다중 디스크립션 코딩, 다중 스트리밍, 비 공유 저장장치, 부하분산, 적응형 스트리밍, VoD

The Streaming Method using Multiple Description Coding for cluster-based server with shared-nothing storage

BAK, Yuhyeon[†] · KIM, Hagyoung^{**} · KIM, Myungjoon^{***} · KIM, Kyongsok^{****}

ABSTRACT

The cluster system with shared-nothing storage cannot escape from the problem of skewed request toward specific contents. This paper, therefore, suggests streaming method using MDC (Multiple Description Coding) instead of using single original content; this method is able to cope with skewed request in shared-nothing storage server as well as to continue to provide services in case of the system failure. Also, the system can support adaptive streaming service according to user player type, network status, the load of server, and client.

Key Words : Multiple Description Coding, Multiple Streaming, Shared-Nothing Storage, Load Balancing, Adaptive Streaming, VoD

1. 서 론

전통적인 클라이언트/서버 구조의 컴퓨터 시스템은 클라이언트가 데이터를 요청하면 서버가 서비스를 제공한다. 이러한 구조에서 데이터를 요청하는 사용자 수가 증가하면 단일 서버만으로는 모든 사용자의 요청을 처리할 수 없기 때문에 클러스터 기반의 다중 서버로 서비스를 제공하는 방법들이 제안되었다.

일반적인 클러스터 기반의 다중 서버를 가지는 시스템은

비 공유 저장장치를 가지는 구조이다. 따라서, 모든 데이터에 대해서 균등한 사용자 요청이 들어오면 최적의 성능을 내지만, VoD(Video On Demand) 서비스와 같이 인기 많은 데이터에 대한 요구와 인기 없는 데이터에 대한 요구가 현저하게 다르면, 인기 많은 데이터를 가지고 있는 서버는 사용자 요청을 제대로 처리하지 못하는 반면에, 인기 없는 데이터만을 가지는 서버는 시스템 자원을 낭비하는 단점을 가진다. 이러한 사용자 요청의 불균형으로 인해 시스템 자원의 효율적인 사용을 위한 부하분산에 관한 많은 연구가 이루어졌다.

이러한 시스템에서 부하 불균형 문제를 해결할 수 있는 대표적인 방법은 콘텐츠 복제와 스트라이핑(striping) 방법이다[1, 2]. 콘텐츠 복제 방법은 자원의 낭비 문제와 동적으로 복제를 하는 경우에 처리해야 하는 시점에서 부하가 급

[†] 정 회 원 : 한국전자통신연구원 디지털홈연구단 인터넷서버그룹 선임연구원

^{**} 정 회 원 : 한국전자통신연구원 디지털홈연구단 인터넷서버그룹 책임연구원(팀장)

^{***} 중 심 회 원 : 한국전자통신연구원 디지털홈연구단 인터넷서버그룹 책임연구원(그룹장)

^{****} 정 회 원 : 부산대학교 전자전기정보컴퓨터공학부 교수
논문접수 : 2006년 3월 9일, 심사완료 : 2006년 5월 23일

증하는 문제점을 가진다. 스트라이핑은 여러 개의 저장장치에 동일한 부하를 주기 때문에 모든 저장 장치에 동일한 부하를 줄 수 있는 효과가 있지만, 여러 서버에서 여러 디스크를 공유하는 시스템에서는 SAN과 같은 고비용의 장비를 구축하지 않고서는 외부 통신비용으로 인한 오버헤드가 발생한다. 또한, 여러 디스크가 동일한 크기와 속도를 가지지 않는 경우에는 복잡성이 증가하며 새로운 디스크의 추가하면, 이전 데이터를 모두 재구성해야 하는데 재구성을 수행할 때 시스템의 부하가 급증하게 된다.

본 논문에서는 클러스터 기반의 비 공유 저장장치를 가지는 스트리밍 서버에서 사용자의 편향된 데이터 요청에도 모든 서버의 부하를 동일하게 하면서, 특정 서버에서 에러가 발생하는 경우에도 스트리밍 서비스가 끊어지는 현상을 방지하기 위하여 MDC(Multiple Description Coding)를 활용하는 방법을 제안한다. 제안하는 방법은 또한 사용자 단말 및 네트워크 상황에도 동적으로 적용할 수 있는 장점을 가진다. 본 논문의 구성은 다음과 같다. 2장에서는 스트리밍 서버의 유형, MDC에 대해 살펴보고 3장에서는 본 논문이 제안하는 시스템인 차세대 인터넷 서버에 대해서 설명한 후, 4장에서 차세대 인터넷 서버로 구성된 클러스터 시스템에서 MDC를

이용한 부하 분산, 서버의 고장 시 지속적인 스트리밍 서비스 방안 및 QoS 지원방안을 설명한다. 5장에서는 구현 결과 및 시뮬레이션 결과를 살펴보고 6장에서 결론을 맺는다.

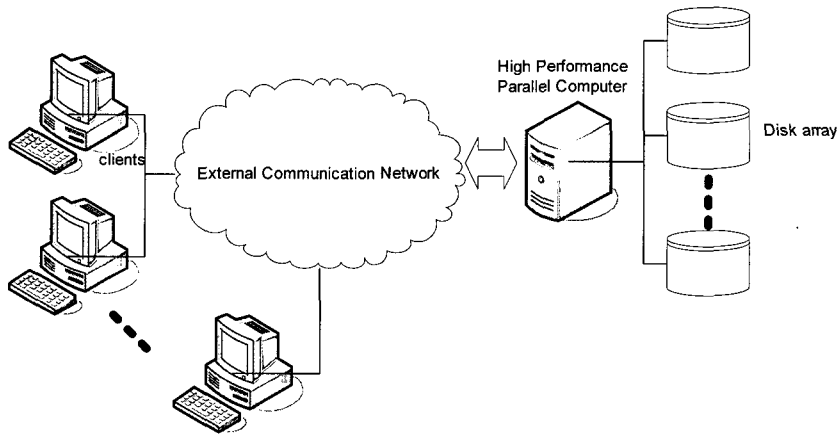
2. 관련 연구

2.1 스트리밍 서버의 유형

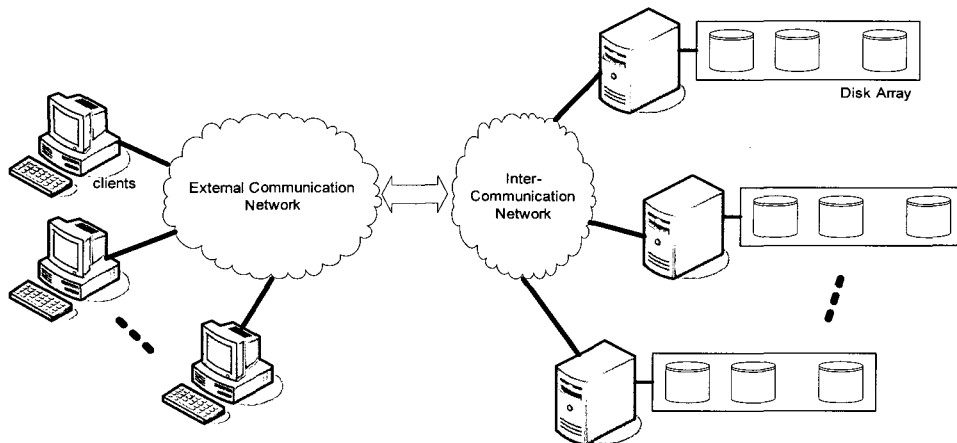
초기의 스트리밍 서버에 관한 연구는 주로 대용량 병렬 컴퓨터로 구성하였다. 하지만 경제성과 확장성을 위해서 클러스터 기반의 스트리밍 서버에 관한 연구들이 진행되고 있다[3-6]. 클러스터 기반의 스트리밍 서버는 비 공유/공유 저장장치를 가지는 스트리밍으로 구분할 수 있다.

대용량 병렬 스트리밍 서버는 (그림 1)과 같이 대용량 병렬 컴퓨터와 대용량 저장 시스템 기술을 이용하여 구성하고, 중앙 집중식 처리 방식이기 때문에 시스템 구성 및 구현이 용이한 장점을 가지나 저장 공간의 확장이나 처리 용량 증대에 융통성이 적고 단일점 고장이 가능하기 때문에 가용성이 적으며 구현 비용이 높은 단점을 가진다.

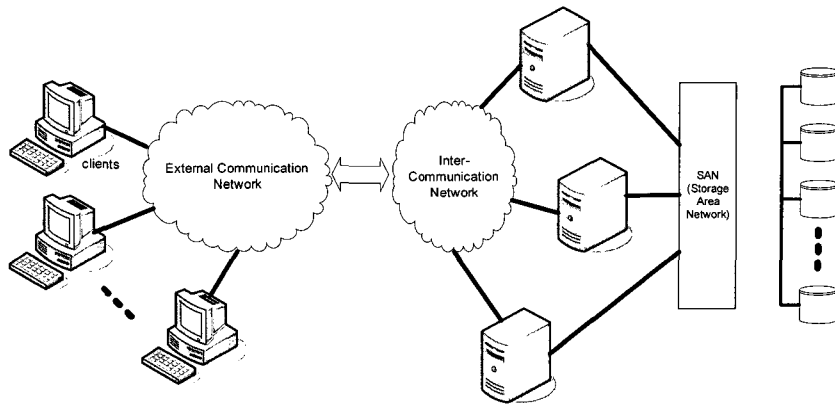
비 공유 저장장치를 가지는 스트리밍 서버는 다수의 중저가형 시스템을 상호 연결하여 구성하며, 관리 서버는 사용



(그림 1) 대용량 병렬 스트리밍 서버



(그림 2) 비 공유 저장장치를 가지는 스트리밍 서버



(그림 3) 공유 저장장치를 가지는 스트리밍 서버

자 요청을 받아 최적의 서버로 분배하는 방법이다. 저장 공간 및 처리 용량의 확장성이 있으나 저장공간을 공유하지 않기 때문에 특정 콘텐츠에 대한 접근이 불가능할 경우가 존재한다.

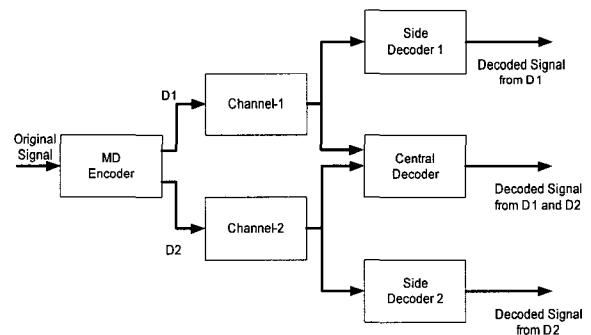
공유 저장장치를 가지는 스트리밍 서버는 초기 연구에서 주로 RAID 시스템으로 구현하였다[13]. 이러한 시스템은 관리 및 구성하기 쉽고, 저장장치 비용이 비교적 저렴하였으나, 확장성과 신뢰성에 제약이 있으며, 부가 관리 비용과 고장 가능성이 높은 단점을 가졌다. 최근에는 다수의 중저가형 시스템을 연결하고 SAN(Storage Area Network)과 같은 고속의 네트워크로 연결된 저장장치로 구성하는 방법이 제안되고 있다. 이러한 구성은 저장 공간 및 처리 용량의 확장성이 있고, 전체 서버에 문제가 생기지 않는 한 어떤 콘텐츠라도 접근이 가능하다. 하지만 이 방법 역시 SAN을 지원하기 위한 고가의 장비가 추가적으로 필요한 단점을 가진다.

2.2 MDC(Multiple Description Coding)

일반적인 미디어 코더가 하나의 비트 스트림을 생성하는 것과 반대로 계층화 코딩(LC:Layered Coding)과 MDC는 하나의 미디어 소스를 두 개 이상의 비트 스트림으로 만들어 피할 수 없는 전송에러에 강인하게 하는 소스 코딩 방법이다[24].

MDC는 벨 연구소에서 최초로 개발된 방법이다[7]. 그 후에 소스 신호에 대한 MDC는 정보 이론 분야와 신호 처리 분야에서 널리 연구되어 왔다[8, 9]. 신호 처리 분야에서 MDC의 개념은 음성 압축에 차례로 적용되었으며[10] 그 후에 영상과 동영상 압축에 차례로 적용되었다[11, 12]. MDC는 하나의 스트림을 여러 개의 분리된 비트 스트림으로 코딩 하는 것을 의미한다. 즉, 하나의 스트림을 여러 개의 스트림으로 분할하고 각 스트림에는 자신의 정보와 다른 스트림에 관한 부가 정보를 추가하여 어느 하나의 스트림을 잃더라도 나머지 스트림으로 복구할 수 있고 모두를 수신하게 되면 송신 시와 똑 같은 품질의 콘텐츠를 생성할 수 있는 코딩 기술이다[13]. 이것은 Layered Coding과는 달리 각 비트 스트림은 서로 독립적이며 어느 것이 더 중요한 것이 없이 각각 똑 같은 중요도를 가지며 어떤 스트림을 수신하든 재생이 가능하고 수신되는 스트림이 많을수록 재생 품질이 좋아지는 장점이 있다[16].

(그림 4)는 2개의 디스크립션으로 코딩되는 시스템을 나타낸다. MDC 인코더는 2개의 디스크립션을 생성하고 이들을 두개의 독립적인 채널을 통해서 전송한다. 만일 두 채널 모두에서 에러가 발생하지 않은 경우 중앙 디코더(central decoder)는 손상되지 않은 두 개의 디스크립션을 전송받고 이를 하나의 고품질 신호로 재구성하게 된다. 만일 하나의 채널만이 에러가 발생하지 않은 경우에는 주변 디코더(side decoder)가 하나의 저품질이지만 수용가능한 신호로 재구성하게 된다[24].



(그림 4) 2개의 디스크립션으로 구성된 MDC 시스템[24]

3. 시스템 구성

3.1 시스템 개요[17]

차세대 인터넷 서버는 고성능 인터넷 환경, 고품질 무선 통신 환경, 높은 인터넷 사용율을 보유한 한국에서 HDTV 급 고품질, 실시간 스트리밍 서비스를 제공하는 고성능 인터넷 서버이다. 고품질의 서비스를 제공하기 위해서는 고대역폭이 필요하고 그 대역폭이 전역적, 지역적으로 고르게 분산 되어야 한다. 몇 개의 서버가 서비스를 제공하면 백본 네트워크의 사용량이 늘어나서 백본 네트워크 증설 비용도 증가할 뿐만 아니라 서비스 요구도 집중되어 서버에 과부하가 걸릴 수 있다.

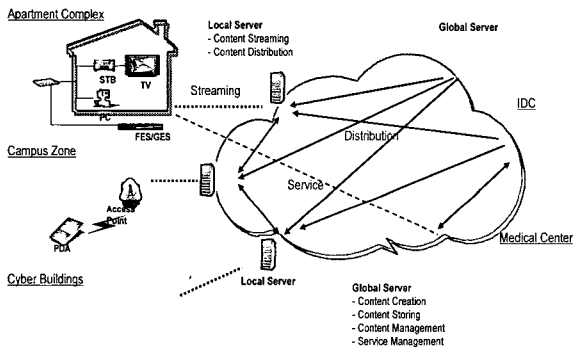
차세대 인터넷 서버는 이러한 문제점을 해결하기 위하여 설계 되었다. 차세대 인터넷 서버는 네트워크 대역폭을 효

울적으로 사용하기 위해서 서비스를 광역과 지역으로 구분한다. 넓은 지역에 서비스를 제공하는 서버는 광역서버(global server)이고, IDC, 전자정부, 디지털 도서관, 새로운 산업을 위한 응용 서비스를 제공한다. 좁은 지역에 서비스를 제공하는 지역서버(local server)는 회사, 대학교, 아파트 단지와 같이 지역 네트워크에 바탕을 둔 지역 공동체를 위한 서비스를 제공한다. 지역서버는 일종의 서비스 캐쉬와 같은 역할을 담당하는데 최종 사용자가 인식하진 못하지만 전역서버와 협업하여 서비스를 제공한다.

차세대 인터넷 서버는 많은 사용자에게 스트리밍 서비스를 제공하기 위해서 다음과 같은 기술을 적용하였다.

- 스트리밍 서비스에 최적화된 서버를 구현하기 위해 스트리밍 가속 기술을 가진 NS(Network Storage) 카드를 개발하였다.
- NS 카드를 지원하는 대용량 멀티미디어 파일 관리를 위해 표준 인터페이스를 가지는 파일시스템을 개발하였다.
- 네트워크 경로상의 전달지연을 최소화하기 위해 스트리밍 캐싱 기법을 도입하였다.

(그림 5)는 차세대 인터넷 서버가 운영되는 환경의 모습이다.



(그림 5) 차세대 인터넷 모델

3.2 하드웨어 구성

차세대 인터넷 서버는 두 개의 프로세서(processor)와 1개 이상의 NS(Network/Storage) 카드로 구성된다.

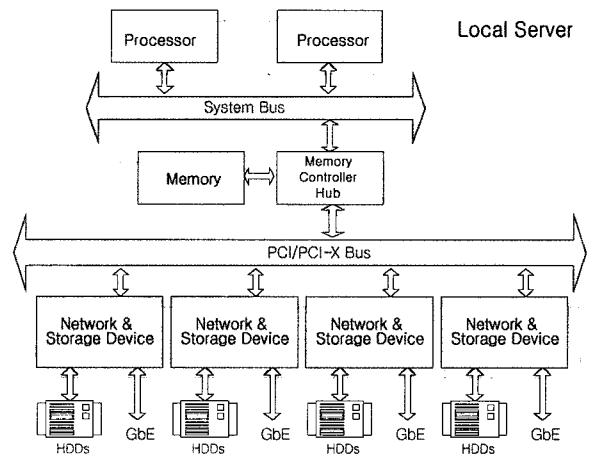
NS 카드[18]는 SCSI 저장장치, 이더넷 네트워크 컨트롤러, 메모리를 하나의 카드에 장착하여 PCI 인터페이스를 통하여 호스트와 접속되는 PCI 카드이다. NS 카드는 저장장치로부터 읽은 데이터를 메인 메모리에 로드시키지 않고 PMEM이라고 하는 NS 카드 내에 장착된 메모리로 데이터를 읽었다가 네트워크 인터페이스를 통해 바로 네트워크로 전송하는 기능을 가진다. NS 카드에서는 TCP/UDP 스택을 처리하는 TOE(TCP Offload Engine) 칩을 사용하고 있으며, 이로 인하여 불필요한 메모리 복사 없이 데이터 스트림을 저장장치에서 네트워크로 전송시킬 수 있다. 따라서 NS 카드는 저장장치와 네트워크 성능에서 기존의 I/O 가속 장치

보다 적은 CPU 사용율과 높은 성능을 가진다.

각각의 NS카드의 고유의 IP를 할당 받아 서비스를 제공하기 때문에 복수개의 NS 카드가 장착된 서버는 전통적인 클러스터 시스템의 기능을 수행할 수 있다. 즉, 전통적인 클러스터 시스템이 3대의 서버로 구성된다면 같은 성능을 한대의 서버에 NS 카드 3장을 장착시켜 구현할 수 있다. 따라서, NS 카드를 사용하게 되면 저비용으로 효율적인 스트리밍 서버를 구성할 수 있는 장점을 가진다.

또한, NS 카드는 다음과 같은 이유 때문에 일반 시스템에 비해 빠른 속도를 낼 수 있다.

- Zero-Copy : 컴퓨터가 처음 만들어질 때부터 지금까지 폰 노이만 방식의 컴퓨터 구조는 매우 광범위하게 사용되었다. 이 구조를 가지는 컴퓨터는 범용적 사용을 위해서 디스크에 저장된 내용을 네트워크로 보내기 위해서는 커널과 사용자 메모리 사이에서 많은 내부 복사 과정을 가지게 된다. 하지만 스트리밍 서비스만을 제공하는 시스템에서는 이러한 과정은 불필요하다.
- 큰 블록 사이즈 : 일반적인 리눅스 시스템은 페이지 크기의 한계 때문에 4K byte 이상의 블록을 지원하지 않는다. 따라서 스트리밍 데이터와 같이 매우 큰 데이터는 4K byte 이하의 블록 단위로 나누어져 관리하게 되고 이는 큰 블록으로 관리하는 것에 비해서 성능이 떨어지게 된다. 따라서 스트리밍 데이터와 같이 큰 용량을 가지는 데이터는 관리하게 되는 블록의 크기도 커야만 성능 향상을 도모할 수 있다. NS 카드는 128, 256, 512, 1024, 2048 KB의 블록 크기를 지원한다.
- Block Split : 한 NS 카드에는 여러 개의 디스크가 연결되어 있고, n 번째 블록에 대한 요청이 오면 이를 N (디스크 어레이의 디스크 수)개의 조각 블록들로 나누고 이들을 디스크 어레이의 n 번째 블록에 각각 나누어 저장되며, 따라서 k번째 조각 블록은 디스크 어레이의 k번째 디스크의 n번째 조각 블록으로 I/O를 하게 된다. 이와 같이, 파일 시스템의 조각화된 블록 단위로



(그림 6) 하드웨어 구성

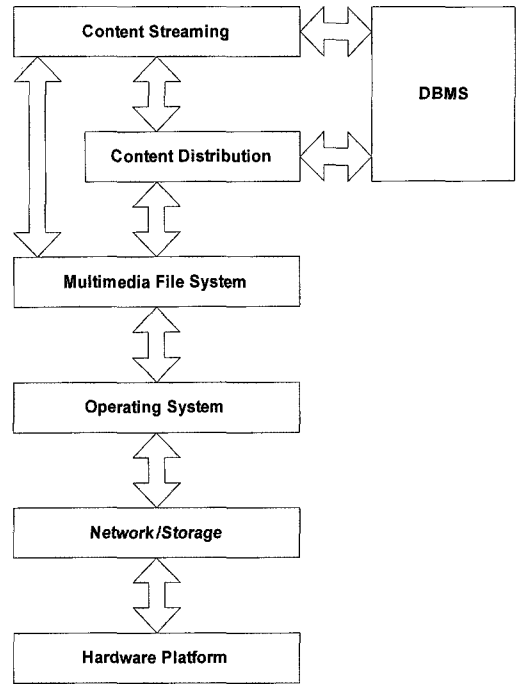
디스크 어레이에 균등하게 분산되어 파일이 저장되므로, 임의의 블록을 접근하고자 할 때, 디스크 어레이의 모든 디스크가 동시에 병렬 처리될 수 있다.

- Pipelined I/O : 비디오/오디오 스트림 데이터가 저장된 디스크 어레이가 쉬고 있는 상태에서 읽기 명령이 발생하면, 첫 번째 블록 조각 읽기 명령과 두 번째 블록 조각 읽기 명령들은 순서대로 디스크 어레이의 각 디스크에 전달된다. 그러면, 읽기 명령을 받은 디스크 어레이의 각각의 디스크들은 첫 번째 블록조각의 위치까지 헤드를 움직인 후 읽어 들여 내부전송하고, 이를 디스크 어레이 저장 장치에 외부 전송하게 된다. 이때, 각각의 디스크에서 각각의 블록조각을 전송하는 순서는 순차적인 필요는 없다. 디스크 어레이 제어장치에서는 전송 받은 N개의 블록 조각을 모아 파일 시스템 단위의 할당 단위인 하나의 논리적인 블록을 만든다. 즉, 임의의 디스크가 n번째 블록 조각 읽기가 완료되면, 즉시 디스크에 n+2번째 블록 조각 읽기 명령을 내린다. 그러면 n+2 번째 블록 조각 읽기 명령을 받은 디스크는 내부 전송과 외부전송 과정을 처리하게 된다. 이와 같이 디스크 어레이 제어 장치가 디스크 어레이에 블록 조각 읽기 명령을 큐잉하여 전달함으로써 모든 외부 전송을 하나의 시간 축으로 모았을 때 외부 전송이 쉽 없이 일어나는 것을 알 수 있고, 그만큼 디스크 어레이의 읽기 성능이 높아지게 된다.

3.3 소프트웨어 구성

차세대 인터넷 서버는 (그림 7)과 같은 소프트웨어 구조 [19]를 가진다.

- 멀티미디어 파일 시스템[22] : NS 카드를 지원하기 위한 파일 시스템으로 초기에는 전용 파일 시스템으로 개발되었다가 EXT3 파일 시스템 인터페이스를 가지도록 개발되었다. 개발된 파일 시스템은 EXT3NS 파일 시스템으로 불린다. EXT3NS는 표준 읽기/쓰기 인터페이스를 통하여 NS 카드로 빠른 I/O를 할 수 있으며 기존의 VFS 인터페이스도 지원한다. 또한 NS 카드의 디스크 스트라이핑 단위로 결정되는 큰 블록 크기(128, 256, 512, 1024, 2048 KB)를 지원한다. EXT3NS는 기존의 EXT3가 bitmap을 모두 동일한 블록 크기(1K~4K)를 표현하는 것과는 달리, 메타데이터 영역은 기존의 EXT3와 같이 1K~4K 블록으로 표현하고 데이터 영역은 대용량 데이터를 위한 대용량 블록(128, 256, 512, 1024, 2048 KB)으로 표현하는 방식으로 구성하였다.
- 콘텐츠 스트리밍 모듈 : 실제 콘텐츠의 스트리밍 서비스를 담당하는 모듈이다. 스트리밍 컨트롤 전송 프로토콜인 RTSP를 기반으로 다양한 사용자 커넥션을 관리, 분배하는 주 프로세스와 공유메모리를 사용하여 내부 IPC 메시지를 통해 내부 컨트롤을 생성하여 실제 동영상 서비스를 수행하는 NS 프로세스로 크게 나누어 진다.

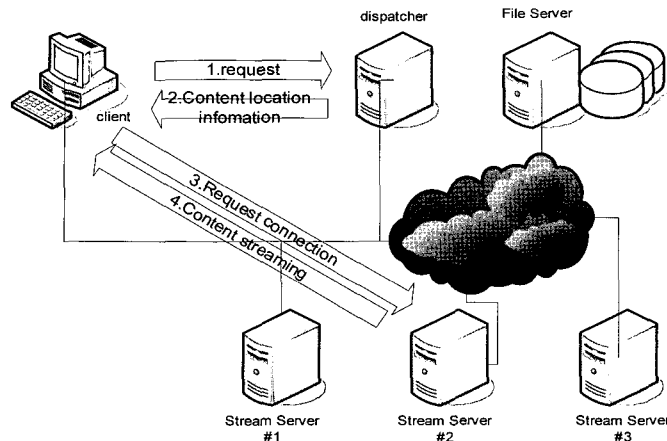


(그림 7) 소프트웨어 계층

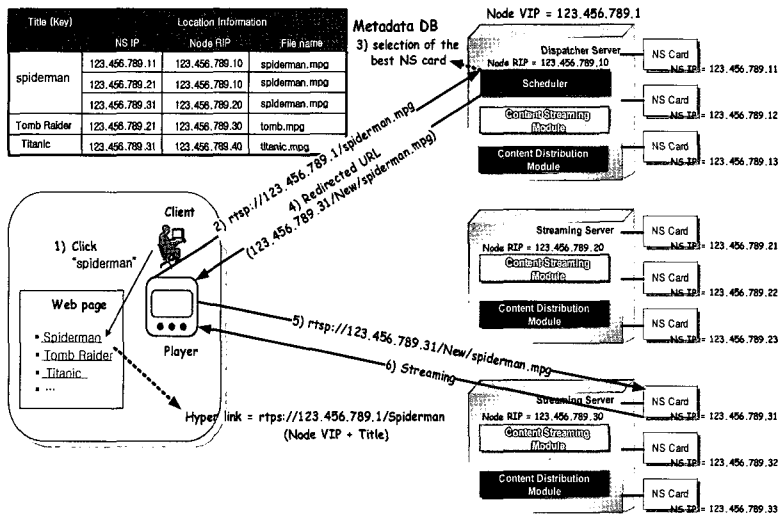
- 콘텐츠 분배 모듈 : 클러스터 내에서의 콘텐츠 배치를 담당하는 모듈이다. 파일 서버로부터 일반 스트리밍 서버로의 콘텐츠 배치와 일반 서버 사이의 콘텐츠 복제(replication), 콘텐츠 삭제, 시스템 부하 정보 수집 등의 역할을 수행한다.
- 운영체제 : 차세대 인터넷 서버는 리눅스를 기반으로 하고 있으며 현재 커널 버전은 2.6.10이다.
- 데이터베이스 : 시스템의 메타데이터를 관리하는 모듈로 ODBC를 제공하는 어떤 데이터베이스도 사용할 수 있다.

3.4 서비스 시나리오

스트리밍 서비스를 하기 위해서는 MPEG-2, MPEG-4와 같은 비디오 데이터를 설치하여야 한다. 콘텐츠 설치 과정을 통해 일반 속도로의 재생뿐만 아니라 고속 재생, 뒤로 재생, 뒤로 고속 재생과 같은 서비스를 제공하기 위한 보조 파일을 생성하고, 콘텐츠의 저장 위치 등을 지정하기 위한 메타 데이터를 생성한다. 사용자 요청으로부터 사용자에게 콘텐츠가 전송되는 과정은 (그림 8)와 같다. 이를 세부적으로 살펴보면, (그림 9)과 같이 사용자는 웹 화면에 서비스를 받을 수 있는 콘텐츠의 목록을 볼 수 있으며 이 중에서 특정 콘텐츠를 선택하면 선택한 콘텐츠 정보가 디스패처 서버로 가게 되고 디스패처 서버는 요청을 받은 콘텐츠가 저장되어 있는 NS 카드를 찾아서 위치 정보를 사용자의 플레이어에게 전송한다. 만일 선택된 콘텐츠가 서버 클러스터 내에 여러 개 존재할 경우 가장 부하가 적은 NS 카드를 선택해서 전송한다. 플레이어는 위치 정보를 통해 NS 카드로 접속하여 실제 콘텐츠를 전송 받아 서비스한다.



(그림 8) 스트리밍 서비스 전체 단계



(그림 9) 스트리밍 서비스 세부 단계

차세대 인터넷 서버의 지역서버 클러스터는 (그림 8)와 같이 파일서버, 디스패처와 다수의 스트리밍 서버들로 구성된다. 각 스트리밍 서버는 1~3개의 NS 카드를 장착하여 스트리밍 서비스를 제공한다.

3.5 시스템의 제한점

차세대 인터넷 서버는 저비용의 서버 구성을 위해 비공유 저장장치를 가지는 모델로 개발되었다. 따라서 다음과 같은 문제점이 발생할 수 있다.

- 특정 콘텐츠에 대한 집중적인 요청이 발생하면 해당 콘텐츠를 저장하고 있는 특정 NS 카드 또는 특정 서버에 부하가 집중된다.
- 스트리밍 서비스를 하는 중에 서비스를 제공하는 NS 카드 또는 서버가 문제가 발생하면 사용자 측에서는 스트리밍 서비스가 중단된다.

이러한 문제점을 해결하기 위해서 미리 콘텐츠를 복수 개

의 NS 카드에 중복해서 저장하거나, 동적으로 콘텐츠를 복제(replication) 방법이 사용되었다. 하지만, 콘텐츠 중복으로 인한 저장공간의 낭비, 동적 복제의 경우 정확한 시점을 선택하는 오버헤드와 복제로 인한 서비스의 지연 문제와 콘텐츠를 모든 NS 카드에 복제하지 않을 경우에는 콘텐츠 배치 문제가 발생하게 된다.

4. MDC를 이용한 차세대 인터넷 서버 설계

3.5에서 언급한 시스템 제한점을 해결하기 위해 MDC(Multiple Description Coding)를 적용하여 설계하였다. MDC는 하나의 스트림을 여러 개의 스트림으로 분할하고 각 스트림에는 자신의 정보와 다른 스트림에 관한 부가 정보를 추가하여 어느 하나의 스트림을 잃더라도 나머지 스트림으로 복구할 수 있고 모두를 수신하게 되면 송신 시와 똑같은 품질의 콘텐츠를 생성할 수 있는 코딩 기술이다. 지금까지 MDC는 콘텐츠를 제공하는 시스템이 각기 다른 디스크립션을 클라이언트로 제공하는 모델에서 고려되어 왔다. 하지만, 차세대

인터넷 서버와 같이 비 공유 저장장치를 가진 시스템에 이를 적용하면 3.5에서 언급된 시스템의 제한점을 해결할 수 있다. 또한 MDC를 활용하게 되면 단말기, 네트워크 상태, 서버 또는 클라이언트의 부하 상태에 따라 적응적으로 스트리밍 서비스를 제공할 수 있다.

4.1 콘텐츠 배치 및 메타 데이터 관리

스트리밍 서비스를 하기 위해서는 원본 콘텐츠를 어셋(asset) 설치 과정을 통해서 ff_file, fr_file, index_file, s_e_file 등의 파일을 생성하고 bit rate, 상영 시간 등의 메타데이터를 뽑아낸다. MDC를 적용하게 되면 어셋 설치과정에서 콘텐츠를 MDC 코딩 수준에 따라 n개의 디스크립션을 생성하는 과정이 추가된다. 이렇게 생성된 n개의 디스크립션을 클러스터 시스템의 n개의 NS 카드의 저장장치에 저장된다. 모든 NS 카드가 각자 저장하고 있는 콘텐츠의 디스크립션을 사용자에게 전송하게 되면 RAID의 스트라이핑과 비슷한 효과를 가지게 되어 전체 NS 카드가 균일하게 사용된다.

원본 콘텐츠와 MDC로 인코딩 된 후의 디스크립션의 질적 관계는 다음과 같이 표현될 수 있다. 즉, 콘텐츠 C의 품질은 각 디스크립션의 모두 전송 받아 디코딩 한 후의 질과 같다.

$$QC = QD_1 \cup QD_2 \cup \dots \cup QD_n$$

(QC : 원본 콘텐츠의 품질, QD₁~QD_n : 디스크립션의 품질)

원본 콘텐츠로부터 인코딩 된 n개의 디스크립션은 파일로 관리된다. 따라서 각각의 디스크립션은 각각 다른 이름으로 저장되고 다음과 같은 메타데이터들이 관리 대상이 된다.

- GCT(Global Content Table) : 콘텐츠의 대략적인 정보, 콘텐츠 제목, 간단한 설명, 콘텐츠에 관한 간단한 설명, 상영 시간
- LCT(Local Content Table) : MDC의 디스크립션의 정

보(콘텐츠의 ID, 파일의 크기, 비트율, 전송율, 사용자 수 등)

- ABNORMAL_CT : 특정 NS 카드 또는 특정 서버에 문제가 생겨서 더 이상 서비스 해 줄 수 없는 디스크립션의 목록

n개의 NS 카드로 구성된 클러스터 시스템에서 CID에 대한 실제 파일 이름은 CID+NSID로 구성된다. 즉, CID가 "damo"라면 실제 파일이름은 "damo_1", "damo_2", ..., "damo_n"이 된다.

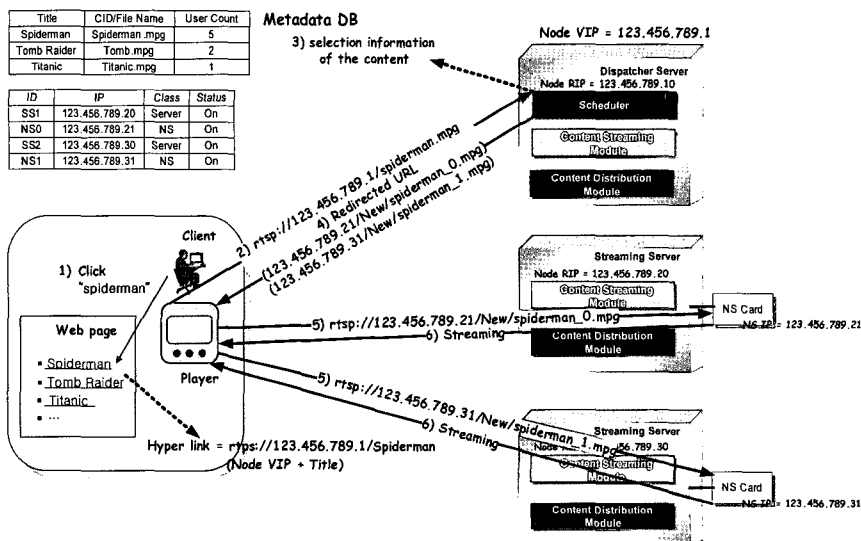
4.2 스트리밍 서비스 단계

MDC를 이용한 스트리밍 서비스는 다음과 같은 단계를 거쳐 제공된다.

- 1) 사용자는 서비스 받을 콘텐츠를 선택한다.
- 2) 사용자의 요청은 디스패처 서버로 전송된다.
- 3) 디스패처 서버는 사용자의 요청에 해당하는 디스크립션의 이름과 위치를 사용자에게 전송한다.
- 4) 사용자는 디스패처 서버로부터 전달받은 NS 정보를 바탕으로 각 디스크립션에 대해 해당 위치로 RTSP 요청을 수행한다.
- 5) 요청 받은 NS 카드들은 스트리밍 서비스를 수행한다.

4.3 부하 분산 및 서버 에러 대응 방법

이와 같이 차세대 인터넷 서버에 MDC를 적용하게 되면 요청 받은 콘텐츠에 대한 서비스를 모든 NS 카드가 제공하게 되어 모든 NS 카드의 부하가 균일하게 된다. 이는 RAID에서 스트라이핑(striping)과 비슷한 효과를 가지게 된다. n개의 NS 카드로 구성된 스트리밍 시스템은 사용자에게 동시에 n개의 스트리밍을 제공하고 사용자는 이를 받아서 디코딩 한 후 플레이 한다.



(그림 10) MDC를 이용한 스트리밍 서비스 단계

만일 스트리밍 서비스를 하고 있는 중에 특정 NS 카드 혹은 특정 서버에 문제가 생겨서 더 이상 스트리밍 서비스를 제공할 수 없는 경우에도 사용자 측에서는 스트리밍 서비스가 중단되는 것이 아니라 처음 서비스를 받을 때에 비해서는 콘텐츠의 질은 떨어지지만 정상적으로 서비스를 제공하는 n-1개의 NS 카드들로부터 전송 받는 디스크립션을 통해 서비스를 계속 받을 수 있다. 또한 정상적인 NS 카드들은 비정상적인 NS 카드들이 저장하고 있는 디스크립션을 파일 서버로부터 전송 받은 후에는 사용자에게 고품질의 스트리밍 서비스를 제공할 수 있다. MDC를 적용하지 않았을 때에 비해서 파일서버로부터 전송 받아야 하는 콘텐츠의 크기가 작아지기 때문에 사용자의 대기시간은 짧아지게 된다.

즉, 다음과 같이 원본 콘텐츠를 SD(Single Description), 이 콘텐츠를 MDC로 인코딩 했을 때 각각의 디스크립션을 D(각 디스크립션의 bit rate, file size는 동일로 가정)로 나타낼 수 있다.

- C : 원본 콘텐츠(SD)
- C' : MDC로 인코딩 된 콘텐츠
- N : MDC coding level (number of description)
- D_n^c : 콘텐츠 c 의 n 번째 디스크립션
- SD_b^c : 콘텐츠 c 의 SD bit rate
- SD_s^c : 콘텐츠 c 의 SD file size
- Db_n^c : 콘텐츠 c 의 MDC n 번째 디스크립션의 bit rate
- Ds_n^c : 콘텐츠 c 의 MDC n 번째 디스크립션의 file size

MDC로 인코딩 된 콘텐츠는 부가정보로 인해 SD와 비교하면 다음과 같이 된다.

$$Sdb^c < \sum Db_n^c, SD_s^c < \sum Ds_n^c (n = 1, 2, \dots, N)$$

$$Sdb^c > Db_n^c, SD_s^c > Ds_n^c (n = 1, 2, \dots, N)$$

즉, 모든 디스크립션의 크기의 합은 SD 콘텐츠에 비해 크지만 하나의 디스크립션의 크기는 SD 콘텐츠에 비해 작고, 대역폭 역시 마찬가지다.

또한 디스크립션의 저장 위치는 다음과 같이 표현할 수 있다.

$$L(D_n^c) = (LastLoc + n) \% NS_n$$

$L(D_n)$ 은 저장위치를 나타내는 함수로, n 번째 디스크립션이 저장된 위치를 나타낸다. 즉, c '콘텐츠의 n 번째 디스크립션은 n 번째 NS 카드에 저장하게 된다.

NS 카드는 네트워크와 스토리지를 결합한 장치이기 때문에 네트워크 문제 또는 디스크 문제가 발생하면 콘텐츠를 전송하지 못하게 된다. 특히 디스크는 비교적 고장이 많은 장치 중의 하나이기 때문에 고장에 대비할 수 있는 능력이 필요하다.

이러한 NS 카드의 예러가 발생하면 차세대 인터넷 서버는 다음과 같이 예러를 회복한다.

- 1) 예러가 발생하면 나머지 n-1개의 NS 카드는 n-1개의

디스크립션을 사용자에게 전송하게 된다. 이때, 사용자 측에서 볼 수 있는 화질은 정상적인 경우에 비해 떨어지지만 스트리밍 서비스를 계속해서 받을 수 있다.

- 2) 서버가 NS 카드 k 의 고장을 인식하면 NS 카드 k 에 저장되어 있는 디스크립션 중에서 현재 서비스 중인 디스크립션들을 선택한다.
- 3) 선택된 디스크립션의 bit rate, file size 등을 고려하여 n-1개의 NS 카드 중에서 저장 위치를 결정한다.
- 4) 파일서버로부터 각 디스크립션을 3)에서 결정한 NS 카드에 저장한다.
- 5) AbnormalCT에 해당 디스크립션 정보를 저장한다.
- 6) 각 사용자에게 복구된 디스크립션을 추가로 전송한다. 이 시점에서 사용자는 정상적인 스트리밍 서비스를 받을 수 있다.
- 7) NS 카드 k 가 정상적으로 복구되면, ABNORMAL_CT의 정보를 통하여 임시로 저장된 디스크립션을 삭제한다. 이때 ABNORMAL_CT의 해당 디스크립션의 Usage Count가 0이 아닌 디스크립션은 0이 될 때까지 기다렸다가 0이 되면 삭제한다. NS 카드 k 가 정상적으로 복구된 시점 이후의 사용자 요청은 NS1~NSn을 통해서 스트리밍 서비스를 받는다.

4.4 단말 적응 및 네트워크 적응 스트리밍 서비스

스트리밍 서버 클러스터의 부하가 크지 않고 사용자-스트리밍 서버간 네트워크 상태가 양호할 경우에는 모든 NS 카드가 각자 저장하고 있는 콘텐츠의 디스크립션을 사용자에게 전송하게 된다. 이는 RAID의 스트라이핑과 비슷한 효과를 가지게 되어 전체 NS 카드가 균일하게 사용된다.

하지만, 사용자의 스트리밍 요청 시 단말기의 성능이 떨어지거나 네트워크의 상태가 불량 또는 서버의 부하로 인해 요청 콘텐츠의 디스크립션 모두를 보낼 수 없거나 보낼 필요가 없는 경우가 발생한다. 이러한 경우에는 그 상태에 맞는 적당한 디스크립션의 수를 결정하여 스트리밍 서비스를 하며 이는 스트리밍 서비스 중에서도 동적으로 변화될 수 있다.

서버의 부하 정보를 제외한 모든 서비스 적응에 관한 부분은 클라이언트가 판단한다. 즉, 클라이언트는 초기에 서버로부터 n 개의 디스크립션을 받고 있다가 전송 지연이 생기거나 클라이언트의 부하로 인해 n 개의 디스크립션을 재생시키지 못하게 되면 디스패처 서버에 디스크립션을 줄이는 명령을 보낸다. 디스패처 서버는 n 개의 NS 카드 중에서 부하가 가장 큰 NS 카드를 선택하여 해당 클라이언트에 대해 더 이상 스트리밍 서비스를 수행하지 않도록 명령한다. 이때 해당 클라이언트에 대해 스트리밍 서비스를 수행하는 NS 카드는 n-1개가 된다. 반대로 클라이언트가 현재 스트리밍 서비스를 받고 있는 n-1개의 디스크립션 보다 더 많은 디스크립션을 수행하고자 하는 경우는 디스패처에 디스크립션의 추가를 요청하게 되고 디스패처는 전체 NS 카드 중에서 가장 부하가 작은 NS 카드를 선택하거나 해당 클라이언

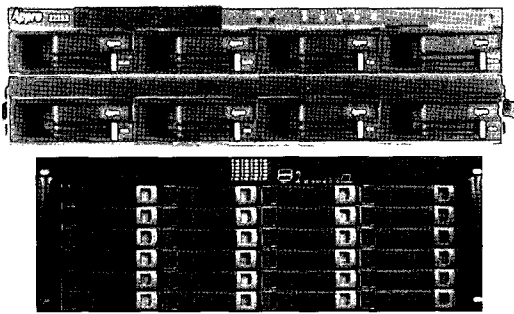
트에게 서비스를 수행하지 않고 있는 NS 카드를 선택하여 클라이언트의 현재 재생 시점부터 스트리밍 서비스를 시작한다.

5. 성능 평가

(그림 11)은 구현된 차세대 인터넷 서버의 모습이다. 컴팩트 모델과 표준 모델은 SCSI 디스크의 수, 최대 NS 카드 장착 수, 메모리가 다르다. 자세한 하드웨어 플랫폼은 <표 1>과 같다.

차세대 인터넷 서버는 일반 서버에 다수의 NS 카드를 장착한다. 각 NS 카드는 PCI 카드와 PCI-to-PCI bridge, 다수의 디스크, PMEM과 TOE로 구성된다. (그림 12)는 NS 카드의 실제 모습이다.

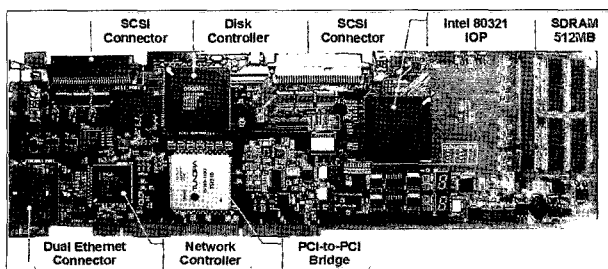
(그림 13)은 NS 카드를 사용하지 않았을 때와 사용할 때의 스트리밍 성능을 나타낸다. 테스트 대상 콘텐츠는 MPEG-2 10Mbps의 속성을 가진다. NS 카드를 사용하지 않을 경우에는 NIC 카드를 통해서 스트리밍을 하였으며 각 카드는



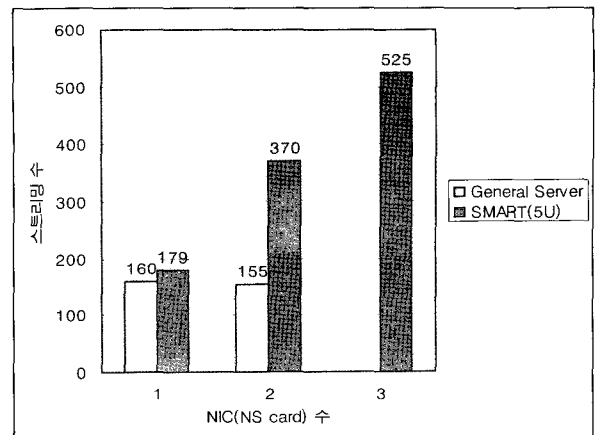
(그림 11) 차세대인터넷 서버(Compact 모델-상, 표준모델-하)

<표 1> 하드웨어 플랫폼

	Compact Model (2U)	Standard Model (4U)
CPU	Dual Xeon 3.06	Dual Xeon 3.06
Memory	1GB	2GB
OS HDD	36GB 15Krpm x 2	36GB 15Krpm x 2
HDD	MAX 6 Hot swap	MAX 24 Hot Swap
Capacity	876GB	3.5TB
Cooling	Total 4 Hot-swap	Total 12 Hot-swap
SCSI Backplane	2Ch <2Ch X 3disk>	6Ch <6Ch X 4disk>
Power	500W Single	1050W + 350W (3+1)



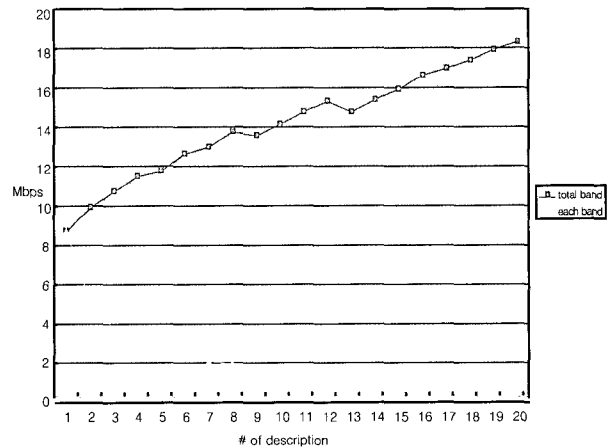
(그림 12) NS(Network / Storage) 카드



(그림 13) 일반서버 vs. NGIS

두 개의 1 GB 네트워크 포트를 가진다. 그림에서와 같이 NS 카드를 사용하지 않은 경우에는 CPU 부하와 네트워크 부하 때문에 NS 카드를 사용하는 경우보다 스트리밍 수가 적음을 알 수 있다.

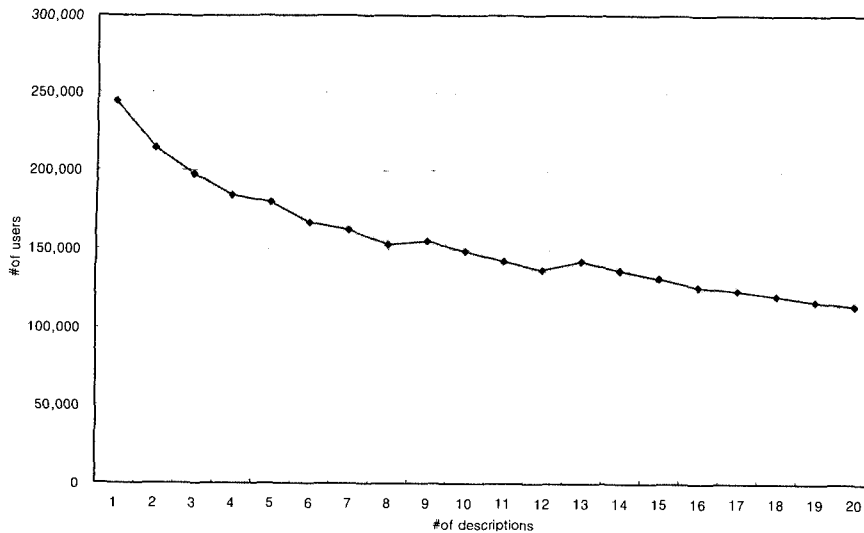
본 논문에서는 차세대 인터넷 서버에서 MDC를 활용했을 경우의 성능 측정을 위해서 아리조나 대학의 MDC 자료[23]를 참조하여 시뮬레이션 하였다. 참조한 비디오의 이름은 mobile이며, 파일 크기는 약 12.9MB이고, 상영시간은 12초, 평균 bit rate는 8.36Mbps이다. <표 2>는 mobile 비디오의 자세한 속성을 나타낸다. (그림 14)는 이 데이터를 기반으로 디스크립션의 수에 따른 대역폭의 변화를 나타낸다. 1개의 디스크립션의 대역폭은 8.6Mbps이지만, 20개의 디스크립션으로 인코딩 한 경우에는 각 디스크립션 1개의 대역폭은 0.9Mbps, 20개 전체의 대역폭 합은 약 18.3Mbps가 된다.



(그림 14) 디스크립션에 따른 bandwidth의 변화

<표 2> mobile(CIF Video Sequence)

Video name	Mobile	Number of frames	300
File size(Byte)	13,503,432	Min Frame size(Byte)	16,360
Time(s)	12	Max Frame size(Byte)	196,944
Mean bit rate	9,002,288	peak bit rate	63,022,080



(그림 15) 디스크립션 수에 따른 동시 사용자 수의 변화

(그림 15)는 디스크립션 수에 따른 동시 사용자 수의 변화를 보여 주기 위한 시뮬레이션 결과이다. 즉, 디스크립션이 하나일 경우 최대 244명의 사용자에게 스트리밍 서비스를 제공할 수 있는 반면에 디스크립션 20개로 서비스를 하는 경우에는 114명의 사용자에게 동시 서비스를 제공할 수 있다. (그림 14)와 (그림 15)를 볼 때, 디스크립션의 수가 증가할수록 시스템의 자원 사용량은 증가함을 볼 수 있다. 하지만, 부하 분산 및 에러 강인 서비스는 디스크립션의 수가 증가할수록 향상된다. 또한 4.3에서 설명한 서비스 중인 NS 카드 중에서 하나가 문제를 발생한 경우, 1개의 디스크립션인 경우는 최소 8.6Mbps의 속도 이상으로 파일 서버로부터 데이터를 받아와야 에러 복구가 가능하지만, 20개의 디스크립션의 경우는 최소 0.9Mbps의 속도만 보장되어도 파일 서버로부터 콘텐츠를 받아오면서 스트리밍 서비스를 할 수 있기 때문에 네트워크 자원을 효율적으로 사용할 수 있는 장점을 가진다.

6. 결론 및 향후 연구

차세대 인터넷 서버는 비 공유 저장장치를 가지는 서버이다. 각 서버는 디스크-네트워크 전송의 효율을 위한 NS 카드가 장착되어 있으며, 이 카드에는 2개의 기가비트 네트워크 포트가 있으며 여러 개의 SCSI 디스크가 연결되어 있다. 따라서, 차세대 인터넷 서버는 NS 카드의 도움으로 일반 서버에 비해 적은 자원을 사용하면서도 많은 사용자에게 스트리밍 서비스를 제공할 수 있다. 하지만, 단일 스트림 서비스를 제공하기 때문에 비 공유 저장장치 구조를 가지는 시스템이 가지는 부하 불균형 문제를 피할 수 없다. 부하 불균형을 해결하기 위해서 콘텐츠 복제를 사용할 수 있지만 복제로 인한 저장장치의 낭비되고 이를 위한 오버헤드가 존재한다. 따라서 본 논문에서는 MDC를 이용하여 비 공유 저장장치 구조를 가지는 시스템에서 부하분산을 효율적으로 수

행하면서도 적응적으로 스트리밍 서비스를 제공할 수 있는 방법을 제안하였다.

차세대 인터넷 서버에 MDC로 인코딩 된 콘텐츠를 사용하게 되면 RAID의 스트라이핑과 비슷한 효과를 얻을 수 있어 모든 서버 및 NS 카드의 부하를 동일하게 만들 수 있다. 또한 스트리밍 서비스 중에 이를 제공하는 서버 및 NS 카드의 문제가 발생하더라도 스트리밍 서비스가 중단되지 않고 QoS에 맞게 스트리밍 서비스를 계속할 수 있다. 문제 발생 이후에 파일 서버로부터 문제가 되는 디스크립션을 정상 NS 카드에 복구시키면 스트리밍 서비스 중에 양질의 스트리밍 서비스를 계속해서 제공할 수 있다.

향후 연구로는 MDC 코딩 수준을 결정하는 것과 MDC로 서비스를 받는 클라이언트의 부하를 줄이기 위한 방법이 있다. 즉, 디스크립션이 많을수록 서버 및 클라이언트에 부하를 주기 때문에 부하 분산 및 적응형 서비스를 위한 적절한 MDC 코딩 수준을 결정해야 하고, 클라이언트에서 다중 스트림으로 데이터를 전송 받아 디코딩을 해야 하기 때문에 단일 스트리밍 서비스에 비해 시스템 자원을 많이 사용하는 문제를 해결해야 한다.

참고 문헌

- [1] 한광록, 이승원, "데이터 복제 서버를 이용한 학사 관리 시스템의 부하 분산에 관한 연구", 한국정보처리학회 논문지 D, Vol.8-D, No.5, 2001. 10.
- [2] 김서균, 남지송, "병렬 미디어 스트림 서버에서 저장 노드 수의 변화에 따른 스트라이핑 크기 결정에 관한 연구", 한국정보처리학회 논문지 C, Vol.8-C, No.6, 2001. 12.
- [3] W. Bolosky, J. Barrera, R. Draves, G. Fitzgerald, Gibson, M. Jones, S. Levi, N. Myhrvold, and Rashid R, "The Tiger video filesaver", In Proc. NOSSDAV'96, 1996.
- [4] J. Gafsi and E. W. Biersack, "Modeling and performance

- comparison of reliability strategies for distributed video servers", IEEE Trans. On Parallel and Distributed Systems, 11(4), 2000.
- [5] L.Golubchik, R.R. Muntz, C.Chou, and S. Berson, "Design of fault-tolerant large-scale VoD servers:with emphasis on high-performance and low-cost", IEEE Trans. On Parallel and Distributed Systems, 12(4), 2001.
- [6] N. Venkatasubramanian and S. Ramanathan, "Load management in distributed video servers", In Proc. IEEE ICDCS'97, 1997.
- [7] Michael Zink, Andreas Mauthe, "P2P Streaming using Multiple Description Coded Video", Proceedings of EuroMicro2004, Sep., 2004.
- [8] L. Ozarow, "On a source coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, Vol.59, pp.1909-1921, Dec., 1980.
- [9] V. A. Vaishampayan, "Design of multiple description scalar quantizer," *IEEE Trans. Inform. Theory*, Vol.39, pp.821-834, May, 1993.
- [10] A. Ingle and V. A. Vaishampayan, DPCM system design for diversity systems with applications to packetized speech, *IEEE Trans. Speech and Audio Processing*, Vol.3, pp. 48-57, Jan., 1995
- [11] M. Orchard, Y. Wang, V. A. Vaishampayan, and A. Reibman, "Redundancy rate distortion analysis of multiple description image coding using pairwise correlating transforms," *Proc. Int. Conf. Image Processing*, pp.608-611, Oct., 1997.
- [12] Y. Wang, M. Orchard, V. A. Vaishampayan, and A. Reibman, "Multiple description coding using pairwise correlating transforms," *IEEE Trans. Image Processing*, vol. 10, pp.351-366, Mar., 2001.
- [13] Apostolopoulos, J.G., "Error-resilient Video Compression via Multiple State Streams", In Processings of VLBV'99. Oct., 1999.
- [14] 김일구, 조남익, "다중 표현을 이용한 에러에 강인한 동영상 부호화 방법", 방송공회논문지 2004년 제 9권 제 1호.
- [15] 김일구, 조남익, "비트율-왜곡 최적화된 DCT 계수 부호화를 이용한 다중 표현 동영상 압축 방법", 방송공학회 논문지, Vol.9, No.1, 2004. 3.
- [16] 김선호, 송병호, "멀티미디어 콘텐츠의 QoS를 개선한 전송 메커니즘", 정보처리학회 논문지 B, 제12-B권 제2호, 2005. 4.
- [17] "Next Generation Internet Server Requirement Specifications V1.0", ETRI, 2002. 9.
- [18] Chong-Won Park, Myung-Joon Kim, Jin-Won Park, "Design of NGIS: The Next Generation Internet Server for Future E-society", GCC2003, Shanghai, China, 2004.
- [19] Byung Kwon Jung, June Kim, Myung Joon Kim, Sung-Ho Hahm, "Implementation of high speed multimedia streaming card using the zero-copy mechanism for streaming server", Proceedings of International Technical Conference on Circuits/Systems, Computer and Communications(ITC-CSCC 2005), Vol.2, 2005.
- [20] Byung-Kwon Jung, Sung Woo-Seok, Seong-Woon Kim, Myung-Joon Kim, "Performance analysis of high speed multimedia streaming card for streaming server", International Symposium of Signals, Circuits and Systems (ISSCS 2005), Vol.2, 2005.
- [21] Chang-Soo Kim, Hag-Young Kim, Myung-Joon Kim, Jae-Soo Yoo, "Design of System for Multimedia Streaming Service", EUC05, Nagasaki, Japan, 2005.
- [22] Baik-Song Ahn, Sung-Hoon Sohn, Chei-Yol Kim, Gyu-Il Cha, Yun-Cheol Baek, Sung-In Jung, Myung-Joon Kim, "Implementation and Evaluation of EXT3NS Multimedia File System", Proceedings of the 12th annual ACM international conference on Multimedia, New York, USA, Oct. 2004
- [23] Video Traces Research Group, <http://trace.eas.asu.edu/tracemain.html>
- [24] Yen-Chi Lee, Joohee Kim, Yucel Altunbasak, Russell M. Mercereau, "Layered coded vs. multiple description coded video over error-prone networks", EURASIP signal processing : image communication, Vol.18, pp.337-356, May, 2003.



박유현

e-mail : bakyh@etri.re.kr

1996년 부산대학교 전자계산학과(학사)

1998년 부산대학교 대학원 전자계산학과 (이학석사)

2000년 부산대학교 대학원 전자계산학과 박사 수료

2000년 한국국방연구원(KIDA) 자원관리연구부 연구원

2001년~현재 한국전자통신연구원 디지털홈연구단 인터넷서버그룹 선임연구원

관심분야: 시스템소프트웨어, 클러스터 시스템, VoD, 다중 세션 스트리밍



김학영

e-mail : h0kim@etri.re.kr

1983년 경북대학교 전자공학과

전자계산전공(학사)

1985년 경북대학교 대학원 전자공학과

전자계산전공(공학석사)

2003년 충남대학교 대학원 컴퓨터공학과 (공학박사)

1988년~현재 한국전자통신연구원 디지털홈연구단 인터넷서버그룹 책임연구원(팀장)

관심분야: 인터넷 서버, 콘텐츠 스트리밍, 콘텐츠 분배, 시스템 소프트웨어, 분산파일시스템, GRID



김명준

e-mail : joonkim@etri.re.kr
 1978년 서울대학교 계산통계학과(학사)
 1980년 한국과학기술원 전산학과(이학석사)
 1986년 프랑스 Nancy 제1대학교 응용수학
 및 전산학과 이학박사
 1980년~1986년 아주대학교 종합연구소
 연구원

1981년~1986년 프랑스 Nancy 전산학 연구소(CRIN) 연구원
 1993년 프랑스 Univ. of Nice Sophia-Antipolis 방문교수
 1986년~현재 한국전자통신연구원 디지털홈연구단 인터넷서버그룹
 책임연구원(그룹장)
 관심분야: 리눅스시스템, 데이터베이스, 분산시스템, 인터넷서비스



김경석

e-mail : gimgs@asadal.cs.pusan.ac.kr
 1977년 서울대학교 무역학과(경제학사)
 1979년 서울대학교 전자계산학과(이학석사)
 1988년 일리노이 주립대(어바나-샴페인)
 전자계산학(박사)
 1988년~1992년 미국 노스다코타 주립대학교
 전자계산학과 조교수

1992년~현재 부산대학교 전자전기정보컴퓨터공학부 교수
 1993년~현재 국내 문자 코드 전문위원회 위원장(Korea JTCL/SC2)
 1997년~현재 국내 문헌 정보 및 로마자 전문 위원회(Korea TC46)
 위원
 1999년~현재 국내 인터넷 네임 위원회 및 한글 네임 WG 위원
 관심분야: 데이터베이스, 멀티미디어, 한글/한말 정보처리,
 인터넷 컴퓨팅