

# 병렬 미디어 스트림 서버에서 저장노드수의 변화에 따른 스트라이핑 크기 결정에 관한 연구

김 서 균<sup>†</sup> · 남 지 승<sup>††</sup>

## 요 약

본 논문에서는 구현한 리눅스 기반의 병렬 미디어 스트림 서버를 기술하고, 최대 성능을 실현하기 위한 저장 파일들의 스트라이핑 정책을 제시하였다. 구현된 병렬 미디어 스트림 서버는 한계사용자 서비스 상황을 예측하고 이를 기반으로 시스템 RAID라는 저장장치 클러스터링 기법을 도입하여 최적의 스트라이핑 정책과 이를 위한 시스템 구성 요건을 제시함으로써 타 시스템에 비하여 성능을 크게 향상시킬 수 있었다. 이 시스템은 하나의 컨트롤 노드와 여러 대의 저장노드들로 구성되는데, 저장노드의 수는 성능요구에 따라 유연한 확장이 가능하다. 시스템의 성능은 저장노드의 성능과 수에 따라 결정되며, 저장노드들은 각 클라이언트에 대하여 균일한 부하분산을 통하여 서비스를 제공한다. 이 시스템의 특징은 서비스 하고자 하는 파일의 특성에 따라 조개고자 하는 최적의 스트라이핑 크기를 제시하여, 각 서비스 종류별, 저장노드들의 수에 따른 시스템 최적화를 이룬다.

## The study of striping size according to the amount of storage nodes in the Parallel Media Stream Server

Seo-gyun Kim<sup>†</sup> · Ji-seung Nam<sup>††</sup>

## ABSTRACT

In this paper, we proposed the striping policy for the storage nodes in the Linux-based parallel media stream server. We newly developed a storage clustering architecture, and named it as a system RAID architecture. In this system, many storage cluster nodes are grouped to operate as a single server. This system uses unique striping policy to distribute multimedia files into the parallel storage nodes. If a service request occurs, each storage cluster node transmits striped files concurrently to the clients. This scheme can provide the fair distribution of the preprocessing load in all storage cluster nodes. The feature of this system is a relative striping policy based on the file types, service types, and the number of storage nodes to provide the best service.

키워드: 리눅스(Linux), 병렬 미디어 스트림 서버(Parallel media stream server), 시스템 RAID(System RAID), 컨트롤 노드(control node), 저장노드(storage node), 스트라이핑(striping), 스트림(stream), 클러스터링(clustering)

### 1. 서 론

인터넷에 접속하여 서비스를 제공받는 사용자들은 해당 사이트의 서비스 초기의 대기 시간이 길면 바로 다른 사이트로 이동해 버린다. 그러므로 사용자가 접속했을 때의 대기 시간을 줄이기 위해 많은 노력이 이루어지고 있다. 그러나 VOD 시스템은 타 인터넷 서비스에 비하여 초기지연시간에 덜 민감하다. 대신 서비스의 질이 우수하여야만 하는데, 동영상의 화질이 만족스러워야 함은 물론 끊어짐이 없어야 한다. 또한 동시에 많은 사용자가 접속했을 때 서버가 이를 충분히 처리

할 수 있어야 한다. 현재 인터넷에서 제공하고 있는 VOD는 그 화질이 매우 떨어질 뿐만 아니라 많은 사용자가 동시에 접속하였을 경우 끊어짐 현상이 매우 빈번하여 동영상을 감상하는데 인내심을 가져야 한다[1].

또한 통신망을 통하여 서비스를 제공해주는 서버의 성능이 일시에 많은 사용자를 감당하기가 어려우면 이를 해결하기 위해 많은 비용을 감수해야만 한다. 통신망의 발전과 더불어 이런 망에 어울리는 초고속 미디어제공 시스템이 지원되어야만 그 효과를 충분히 누릴 수가 있다. 미디어제공 시스템의 경우 저장장치에서의 데이터의 액세스 속도가 성능을 좌우하게 되는데, 이러한 문제를 해결하기 위해 RAID, SAN 등 많은 기술들이 도입되었으나, 많은 사용자에게 멀티미디어 스트림을 원활히 제공하는데 따른 막대한 비용부

<sup>†</sup> 준 회원: (주)포스트림 기술이사  
<sup>††</sup> 종신회원: 전남대학교 컴퓨터공학과 교수  
논문접수: 2001년 7월 27일, 심사완료: 2001년 10월 5일

담을 경감시키고 효과적인 부하분산을 위해서 아직도 활발한 연구가 계속 이루어지고 있다[2]. 이에 새로운 기술을 이용하여 균일한 부하분산 및 시스템간 저장 장치들을 유기적으로 결합하여 동시 연동이 가능하도록 한 새로운 데이터 전송기법을 개발하고 적용해야 할 필요가 있다.

본 논문에서는 새로운 멀티미디어 데이터 전송 기술을 통하여 개발하고 있는 시스템 RAID라는 새로운 개념의 병렬 미디어 스트림 서버(모비덕 시스템)를 기술하였다. 이 시스템은 동일한 성능을 갖는 다수의 단일 시스템 노드들이 병렬 클러스터링 그룹으로 구성되어 기능적으로 하나의 단위 시스템으로 동작하는 시스템으로서, 멀티미디어 데이터를 각 노드에 분산 저장하고 각 서버 저장 노드들이 미디어 데이터를 클라이언트에 병렬 전송함으로써 서버의 사용자 가용성과 시스템 확장성을 극대화한 시스템이다. 개발한 병렬미디어 스트림 서버는 인터넷 상에서 고품질의 영화를 제공할 수 있으며, 기존 VOD서버보다 3배 이상의 동시 접속자에게 고품격의 서비스가 가능하다. 이 연구에서는 모비덕 시스템의 최고 장점인 최대동시사용자 수를 제시하고, 이를 위한 최적의 스트라이핑 정책을 제시하였다. 본 논문에서 제시한 스트라이핑 정책은 지금까지 많은 논문에서 연구되었던 RAID기반 VOD 시스템에서의 절대적 스트라이핑 크기 결정 방법과 많은 차이를 보이는 결과를 도출하였다[3, 4].

## 2. 관련 연구

### 2.1 스트리밍 기술

스트리밍 기술은 실시간으로 동영상이나 오디오를 보고 듣게 하는데 필수적인 기술이다. 과거에는 음악이나 비디오 파일을 보기 위해서 엄청나게 큰 사이즈의 파일을 다운로드 받아야만 했다. 그러나 이제는 단순히 클릭만 하면 불과 1-2초 뒤에 곧장 동영상이나 음악이 움직인다. 스트리밍 기술은 파일을 다운로드 받으면서 동시에 재생해 줌으로써 실시간 효과를 극대화 해준다. 이러한 기술 덕분에 생방송도 가능해졌다. 다시 말하면 이 기술은 FTP처럼 네트워크 상에서 데이터를 전부 다운로드한 다음 이를 실행시키는 방법이 아니라 망을 통하여 연속적으로 전송되는 데이터를 클라이언트에서 곧바로 실행시킬 수 있도록 하는 기술로서 주로 인터넷 방송을 위주로 발전되어 왔다. 서버와 사용자간에는 최소한도의 Quality를 보장해 줄 수 있는 네트워크 대역폭이 확보되어야 함은 물론 동영상을 보기 위하여는 클라이언트의 성능 또한 최소 요구를 만족시켜야 한다.

이러한 데이터를 제공하는 서버는 클라이언트가 원하는 많은 수의 데이터를 다수의 사용자에게 일정한 전송률로 제공할 수 있어야만 한다. 따라서 스트리밍 기술은 네트워크 대역폭, 제공 서버의 성능이 그 성패를 좌우하게 된다.

스트리밍 기술은 동영상 데이터를 인코딩하는 부분에서부터,

이를 전송하고 재생하는 부분까지를 모두 포함한다. 이 중 스트림 서버는 스트리밍 기술 구현의 핵심적인 부분으로 물리적인 면에서 범용 서버에 비하여 고성능의 중앙처리장치(프로세서), SCSI 버스, 고속 저장장치 등을 요구한다. 포괄적인 기술이 집약된 스트림 서버는 크게 데이터 전송, 데이터 재생, 망관리, QoS 관리, 접속사용자 관리 등을 위한 소프트웨어적인 부분과 데이터 저장, 데이터 복구, 프로세서 처리량 등을 고려한 하드웨어 부분으로 나눌 수 있다. 인터넷 방송에 적용되는 스트리밍 비디오 기술은 28K/56K의 저 대역 모뎀 인프라상에서도 비디오의 실시간 전송을 실현하고 있다. 현재 국내의 스트림 서버 시장은 주로 고성능 서버에 고가의 RAID나 SAN과 같은 광채널 저장장치를 수용한 장비를 사용하고 있으며, 대표적인 어플리케이션 서비스 기반으로 마이크로소프트의 Windows Media Technology(WMT)나 Real Networks사의 Real System G2가 그 주류를 이루고 있다 [5, 6]. 이와 더불어 양질의 인터넷 방송서비스를 위하여는 서비스 제공자와 이용자간의 충분한 네트워크 대역폭 확보가 필요하고 인코딩 시스템의 성능 및 압축기술에 많은 영향을 받는다.

### 2.2 스트림 서버 기술

스트림 서버의 광범위한 도입에 있어 가장 큰 장벽은 VOD에 대한 수요 부족이 아니라, 사운드와 이미지의 실시간 전송 및 전통적인 전자우편과 파일 전송 응용 운용 양쪽을 모두 만족시키면서 지역망이나 인트라넷에 스트림 기술을 도입하는 것이다. VOD를 잘 활용하려면 복수의 동시 흐름을 처리할 수 있는 전용 서버가 필요하며, 가격 또한 천차만별이다. 비디오 흐름 기능을 제어하는 소프트웨어는 또 다른 고가품이며 각사에서 제공하는 VOD소프트웨어는 사용자 컴퓨터에서 정지, 재생, 시간 선택 등과 같은 기능성을 이용할 수 있게 해준다[7].

인터넷 서비스 병목의 주원인이었던 통신망의 성능 향상으로 사용자의 서비스 요구는 높아진 반면, 이들의 요구 수용을 위한 멀티미디어 서버 시스템의 성능이 이에 미치지 못해, 서버 증설, 중계 시스템 도입, 캐쉬 서버 도입 등을 통하여 겨우 해결하고 있는 실정이다[8]. 이런 현상은 최근에 인터넷 방송, 인터넷 영화상영관 등이 늘어나면서 스트림 위주의 서비스가 늘어나면서 생겨났다.

현재 스트림 어플리케이션은 이미 대부분이 상당한 수준에 이르렀으며, 데이터 압축기술의 발전과 더불어 초고속 광통신망의 보급으로 VOD 스트림 서비스를 위한 기반은 1~2년 전에 비해 매우 발전되었다. 이제 이들 스트리밍 기술을 탑재하게 될 서버의 채택이 사용자들의 서비스의 질을 결정하는 중요한 요소로 떠오르게 되었다. 스트림 서버를 도입할 때에 최우선적으로 고려되는 사항으로는 안정성, 저장매체의 액세스 속도, 관리자의 편의성, 단일 시스템당 동시사용자 수

용능력 및 가격 등을 들 수 있다[9,10].

### 3 새로운 병렬 미디어 스트림 서버의 제안

구현한 병렬 미디어 스트림 서버는 RAID(Redundant Arrays of Inexpensive Disks) 레벨 0의 개념을 네트워크 시스템에 확장 적용한 구조를 가진다. RAID level 0는 하나의 큰 파일을 디스크 어레이 내의 모든 디스크에 스트라이핑(Stripping)이라는 처리과정을 거쳐서 여러 개의 작은 파일 크기로 분할하여 저장하는 방법이다.

이 시스템은 동일한 성능을 갖는 다수의 단일 시스템 노드들이 병렬 클러스터링 그룹으로 구성되어 기능적으로 하나의 단위 시스템으로 동작하는 시스템으로서, 멀티미디어 데이터를 각 노드에 분산 저장하고 각 서버 저장 노드들이 미디어 데이터를 클라이언트에 병렬 전송함으로써 서버의 사용자 수용성과 시스템 확장성을 확대하였다. LINUX를 채용하여 시스템의 안정성을 확보하였고, 또한 웹 기반의 편리하고 다양한 관리도구를 개발하여 서버관리자의 편의성을 증가시켰다.

(그림 1)에 병렬 미디어 스트림 서버의 개요를 묘사하고 서비스 순서를 간략히 나타내었다. 그림을 보면 먼저 사용자가 웹을 통하여 병렬미디어 스트림 서버에 접속하여 동영상 서비스 요청을 한다. 관리노드는 로그인 과정을 통한 인증으로 허가된 사용자 여부를 확인하고, 사용자에게 저장노드와 요청 데이터의 정보가 담긴 메타파일(Meta File)을 전송함과 동시에 저장 노드들에게 해당 사용자에게 서비스 지시를 내린다. 이때부터 저장 서버들은 동시에 각 사용자에게 접속되어 미리 스트라이핑되어 저장되어 있는 스트림 데이터를 전송한다. 사용자측에서는 각 저장 서버들로부터 도착된 데

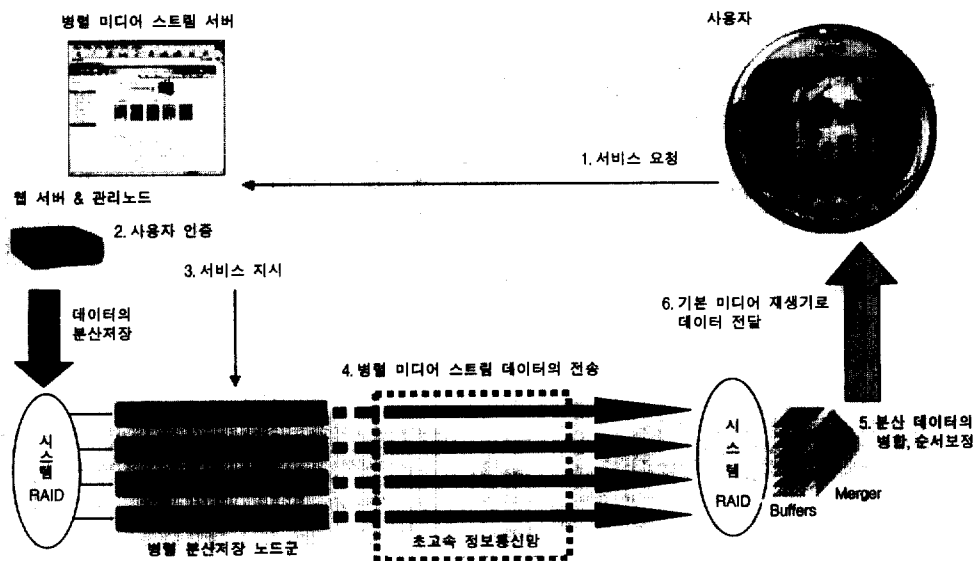
이터를 필터를 통하여 데이터를 재조합하고, 미디어 재생기로 보낸다.

#### 3.1 시스템 RAID

서버 데이터의 대량 전송시 가장 큰 병목 현상을 나타내는 부분은 바로 저장장치의 액세스이다. RAID는 원래 데이터 손실을 방지하기 위하여 개발되었으나 액세스 속도가 늦은 Hard Disk들을 병렬로 배열하여 하나의 파일을 배열된 디스크들에 쪼개어 저장하였다가 파일전송 요구시 동시에 동작함으로써 시스템 I/O버스를 최대한 이용할 수 있도록 개선되었다. 그러나 그 구축 비용이 만만치 않을 뿐 아니라 가격 대 성능비가 충분하지가 않다. 그러므로 RAID 시스템 구축 비용으로 시스템을 여러 대 구입하여 이들 시스템별 저장 장치들을 네트워크로 연결하여 그 구조가 마치 RAID LEVEL0의 구조와 같이 동작하도록 하여 이를 시스템 RAID라 하였다. 시스템들이 여러 대가 네트워크 클러스터군으로 연결된 이 저장 구조체는 고가의 단일 저장서버장비에 비하여 디스크 액세스 속도가 뛰어나며, 성능면에서도 각 시스템별로 파일전송작업이 완전히 분산되어 처리되므로 스루풋이 단일 대형시스템보다 매우 뛰어나다.

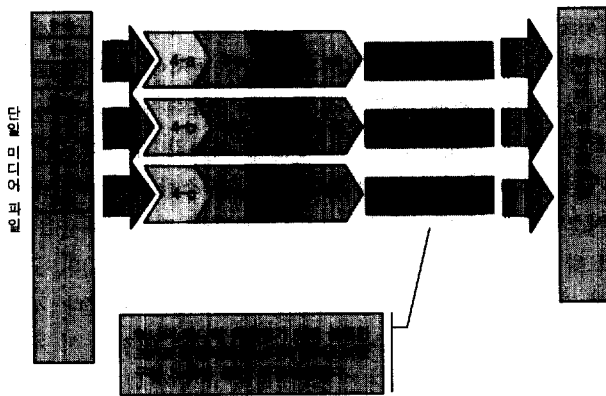
(그림 2)는 한 파일이 각 시스템에 스트라이핑되어 분산 저장되는 시스템 RAID의 모습을 그림으로 나타낸 것이다. 이 그림의 좌측은 하나의 큰 미디어 파일을 스트라이핑 정책에 의해 결정된 파일 조각들로 나누는 것을 보여준다. 나뉘어진 파일들은 각 저장 서버들에 순차적으로 저장되고, 각 저장 서버에서는 스트라이핑된 조각들이 하나의 파일로 재구성되어 같은 이름으로 저장된다. 바로 이점이 RAID LEVEL 0와 다른 점이다. 이것의 최고 장점은 하나의 파일이 각기 다른 시

병렬 미디어 스트림 시스템 구조



(그림 1) 병렬미디어 스트림 서버의 개요도

스텝에 스트라이핑되어 분산 저장되어 데이터 전송 요구시 각 시스템이 자신의 I/O 버스를 각각 최대한 차지하고 파일을 전송하게 되므로, 파일 액세스 속도가 RAID 구성 제품보다 월등하다. 또한 각 시스템마다 동일한 액세스 요구에 대해 동시에 동작하므로 별도의 부하분산 기능이 없어도 균일한 시스템 부하분산을 이룰 수 있다.



(그림 2) 시스템 RAID 개념도

### 3.2 스트리밍 프로그램

이 시스템은 LINUX 기반의 병렬미디어 스트리밍 기술을 적용하여, OS의 안정성 및 가격면에서 타제품보다 우수하다. 본 논문에서 개발된 병렬미디어 스트리밍 서버를 구동하는 프로그램(일명 MIRACLUS)은 웹 기반의 다양한 관리자 툴과 사용자 인터페이스를 제공한다. MIRACLUS의 서버와 클라이언트 프로그램의 기본 구성도는 (그림 3)과 같다. MIRACLUS 서버 프로그램에서는 데이터의 분산저장, 삭제, 저장시스템의 구성 및 관리, 사용자 관리 등의 관리기능을 제공한다. MIRACLUS 클라이언트 프로그램은 TCP를 통하여 전송되는 스트리밍 데이터의 병렬 수신 결합, 병렬 미디어 파일 포맷의 자동 인식 기능을 수행하고, 기존 미디어 재생기와 결합

된 구조를 제공한다.

### 4. 제안한 스트리밍 서버의 디스크 스트라이핑 크기 조절을 통한 성능 분석

이 절에서는 병렬미디어 스트리밍 서버의 최적 스트라이핑 크기를 결정하기 위하여 클라이언트에 대한 서비스 증가시 하드디스크에서 초당 읽어야 하는 데이터의 수와 양을 계산식으로 나타내고, 이를 이용하여 최적 스트라이핑 크기를 계산하고, 수식에 따른 데이터를 제시하여 본 시스템의 성능을 구체적으로 평가하였다. 또한 각 최대사용자수, 최적 스트라이핑 크기, 최적 저장노드수의 상관관계를 분석하고 기술하였다.

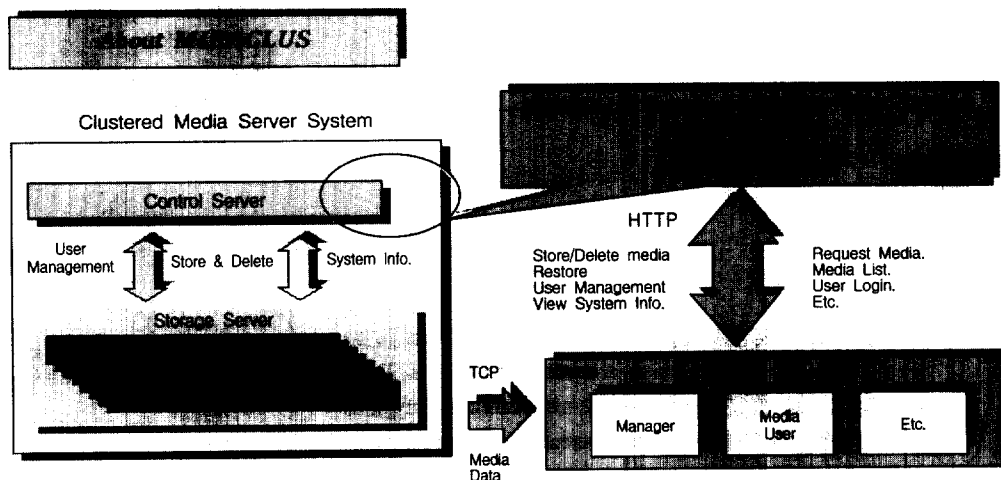
Seagate의 Data sheet를 참고한 하드디스크의 Average seek time과 Formatted transfer rate는 다음과 같다.

- Average seek Time : 5.2msec
- Formatted transfer rate(Max) : 40Mbytes/sec

Average seek Time은 HDD 디스크가 하나의 파일을 읽고자할 때 디스크 헤드가 실린더와 트랙을 이동하여 원하는 파일이 있는 장소까지 이동한 평균 시간을 나타낸 것이다. Formatted transfer rate(Max)는 포맷된 하드디스크가 연속적인 섹터에 저장된 데이터를 1초 동안 전송할 수 있는 최대량을 말한다.

이 데이터를 기준으로 전체 시스템이 동시에 수용 가능한 최대 허용 클라이언트수를 다음과 같이 유도하였다. 이를 구하기 위하여 시스템 환경을 다음과 같이 가정하였다.

- (1) 시스템의 CPU 처리능력은 단위시간당 데이터 전송량과 사용자 접속량을 충분히 지원한다.
- (2) 시스템의 메모리는 1Gbyte 이상으로 부족하지 않다.
- (3) 시스템의 네트워크 카드는 1Gbps를 지원한다.
- (4) 스트라이핑된 하나의 데이터는 HDD에 연속적으로 저



(그림 3) MIRACLUS 서버 및 클라이언트 프로그램의 구성

장되어 있다.

- (5) Average seek time에 HDD 읽기 지연의 모든 요소가 포함된 것으로 간주한다.
- (6) 비디오 데이터 업데이트를 위한 쓰기, 관리를 위한 모니터링 등의 부하는 고려하지 않는다.

하드디스크가 초당 40MB/sec의 읽기를 지원할 경우, 320Mbps의 데이터전송이 가능하다.

그러나 여러 파일을 읽어 들일 때 각 파일들에 대한 디스크 액세스가 한번씩 증가할 때마다 하드디스크의 seek time은 5.2msec씩 증가하여 하드디스크의 전송속도를 저하시킨다. 그러므로 단위시간 안에 무작정 많은 클라이언트에게 서비스하는 것은 불가능하다. 하나의 시스템 하드디스크에서 여러 사용자를 지원하기 위해 단위 시간에 여러 개의 스트라이핑 데이터를 전송한다고 가정하자. 전송해야 할 데이터의 개수가 많아질수록 디스크 seek time이 증가하여 이로 인해 단위시간 동안 데이터를 전송할 수 있는 시간을 빼앗기게 되어 결과적으로 단위시간당 데이터 전송량을 감소시킨다. 이렇게 seek time때문에 시간적인 손실이 발생하여 전송하지 못한 양을 계산하여 이 값을 seek time 손실량(Lst)라 하며, 이 값은 다음과 같이 구해진다.

$$Lst = Rmax \times Ast \quad (1)$$

Ast(sec) : HDD average seek time  
 Rmax(Mbps) : HDD Formatted transfer rate(Max)  
 Lst(Mbit) : HDD seek time 손실량

Lst의 단위는 위 식에서 알 수 있듯이 Mbit이며, 개발에 사용된 Seagate HDD의 경우 Lst는

$$320(\text{Mbps}) \times 0.0052(\text{sec}) = 1.664\text{Mbit}$$

가 된다. 그러면 본 시스템에서 사용한 Seagate HDD의 기초 자료를 바탕으로 개발된 시스템의 성능을 평가해보고자 한다. 먼저 수식에 사용될 변수 및 상수들을 아래에 나열하였다.

N(대) : 저장 노드의 수

S(Mbit) : 파일에 대한 스트라이핑 크기

T(sec) : 단일저장노드가 임의의 클라이언트에게 S의 값으로 스트라이핑된 파일을 전송한 후 다시 그 클라이언트에게 전송하기 시작할 때까지 걸리는 라운드트립 시간

Tmax(N)(sec) : 단일저장노드가 임의의 클라이언트에게 Smax의 값으로 스트라이핑된 파일을 전송한 후 다시 그 클라이언트에게 전송하기 시작할 때까지 걸리는 라운드트립 시간

T(N)(sec) : 저장노드수가 N개인 전체시스템에서 접속사용자가 최대일 때 마지막 접속 클라이언트가 스트라이핑된 데이터를 처음 서비스 받기 위해 대기

해야 하는 최대시간

Ast(sec) : HDD average seek time

Lst(Mbit) : HDD seek time 손실량

Rmax(Mbps) : HDD Formatted transfer rate(Max)

Cx(대/sec) : 단일 저장노드가 임의의 스트라이핑 크기로 1초 동안 전송 서비스 가능한 클라이언트수

Rr(Mbps) : 단일 클라이언트가 영화를 감상할 때 원하는 화질을 유지하기 위해 요구하는 평균전송률

이러한 가정하에 단일 저장노드가 초당 Cx개의 클라이언트를 지원할 때 가능한 최대 스트라이핑 크기 Smax(Mbit)를 구하면

$$Smax = \frac{Rmax - (Lst \times Cx)}{Cx} = \frac{Rmax}{Cx} - Lst \quad (2)$$

또한, T(N)을 구하기 위해 먼저 단일노드에 대한 Tmax값을 Smax를 이용하여 구하면 다음과 같다.

$$Tmax = \frac{Smax}{Rr} \quad (3a)$$

만약 스트라이핑 크기를 Smax값보다 작은 값을 사용하면 T값을 구할 수가 있는데, 그 식은 다음과 같다.

$$T = \frac{S}{Rr} \quad (3b)$$

식 (3a)과 식 (3b)를 비교하면 Smax > S 이므로, Tmax > T가 된다. 식 (3b)에서 우리는 임의의 스트라이핑 크기를 줄때의 T값을 구할 수가 있다. 식 (3a)과 식 (3b)를 이용하여 T(N)값을 구할 수 있다.

Tmax값에 저장노드수 N을 곱하면 Tmax(N)이 되며, T값에 N을 곱하면 임의의 스트라이핑 크기에 대한 T(N)값을 구할 수 있다.

$$Tmax(N) = Tmax \times N = \frac{Smax \times N}{Rr} \quad (4a)$$

$$T(N) = T \times N = \frac{S \times N}{Rr} \quad (4b)$$

식 (4b)에서 T(N)값을 결정하게 되면 역으로 해당 T(N)시간 안에서 최대의 클라이언트를 지원할 수 있는 스트라이핑 크기 S를 구할 수 있다.

$$S = \frac{T(N) \times Rr}{N} \quad (5)$$

그러면 스트라이핑 크기(S)에 따라 단위노드가 초당 수용 가능한 최대 클라이언트수 Cmax는 식 (2)에서 Smax를 S로 놓고 유도할 수 있으며 식 (6)과 같다.

$$C_{max} = \frac{R_{max}}{S + Lst} \quad (6)$$

스트라이핑 크기 S가 정해지면, 노드수(N)가 증가할 때마다 전체 시스템이 동시에 수용 가능한 최대 허용 클라이언트 수 Ymax는 간단히 다음과 같은 수식으로 전개할 수 있다.

$$Y_{max} = C_{max} \times T(N) = C_{max} \times \frac{S}{Rr} \times N \quad (7)$$

$$= \frac{R_{max}}{S + Lst} \times T(N)$$

<표 1>은 Rr값이 1Mbps일 경우, 저장노드 N의 변화에 따른 Ymax의 값과 T(N)의 값의 변화를 Cx값을 기준으로 나타낸 것이다. 이 표를 이용하여 VOD시스템을 구성할 때, 최대 동시사용자와 최종접속자의 최대대기시간을 고려하여, 최적의 시스템 구성을 찾아내고 전체시스템의 규모를 바로 결정할 수 있다.

예를 들어 최대 800명의 동시사용자를 지원하고, 접속시 최대 30초 이내에 영화를 볼 수 있는 시스템을 구성하려고 한다면, <표 1>에서 T(N)값이 30 이내인 값에 대응하는 스트라이핑 크기를 찾고, 여기에서 최대 사용자 Ymax값이 800이

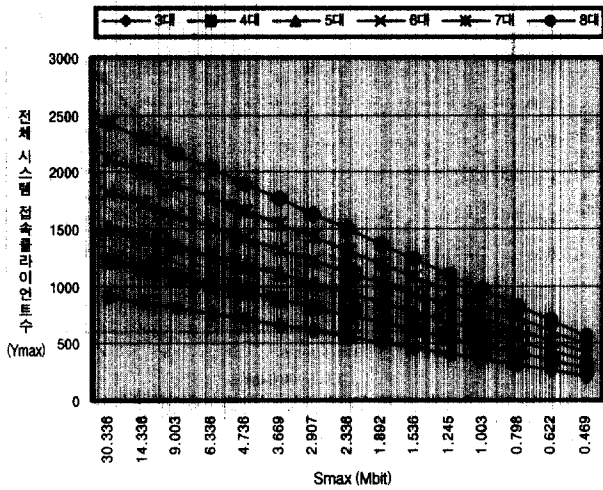
상인 범위를 찾으면 몇 대의 시스템으로 구성이 가능한지를 알 수 있다. 예에서는 저장노드 N이 3대이고, Smax가 9Mbit인 경우 810명을 수용할 수 있도록 하는 것이 가장 경제적이다. 만약 최대 동시사용자를 2,000명 이상으로 한다면, 표에서 알 수 있듯이 저장노드의 수는 7대 이상이어야만 한다.

(그림 4)와 (그림 5)는 <표 1>에서 Ymax값과 T(N)값의 변화를 그래프로 나타낸 것이다. (그림 4)에서 알 수 있듯이 스트라이핑 크기와 최대동시사용자와의 선형적인 관계를 알 수 있으며, 저장노드수에 따라 최대동시사용자의 한계를 알 수 있다. (그림 4)에서는 스트라이핑 크기를 작게 하면 할수록 수용 가능한 최대동시사용자수는 그에 비례하여 감소함을 알 수 있다. (그림 5)에서는 Smax에 따른 T(N)의 변화를 그래프로 표시한 것인데, 스트라이핑 크기가 클수록 최대접속대기시간은 급격히 늘어나는 것을 볼 수 있다. 다시 말하면, 스트라이핑 크기가 감소함에 따라 접속자의 초기대기 지연시간은 지수함수적으로 감소하여 스트라이핑 크기가 매우 작을 경우 빠른 응답시간을 제공하므로, 사용자는 버퍼링하는 동안 기다림 없이 바로 동영상 감상을 할 수 있다. 그러므로 RAID LEVEL0 구조를 가지는 시스템에서의 최적의 스트라이핑 크기는 최대동시접속자수와 최대지연시간의 적절한

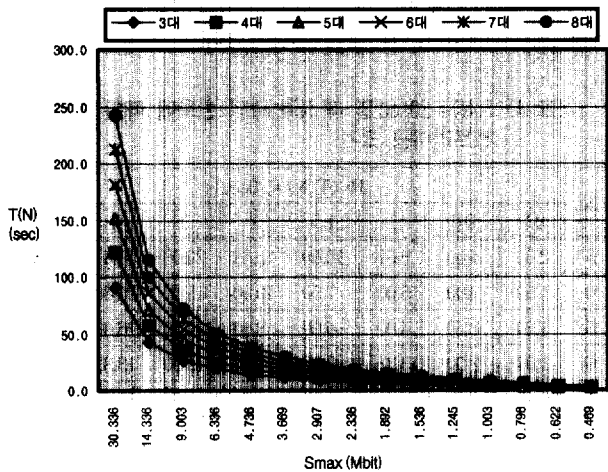
<표 1> 저장노드 N의 변화에 따른 Ymax의 값과 T(N)의 값의 변화 (Rr=1Mbps)

Cx	Lst * Cx	Smax	Rr = 1Mbps일때 저장노드수 N의 변화에 따른 Ymax						Rr = 1Mbps일때 저장노드수 N의 변화에 따른 T(N)					
			3대	4대	5대	6대	7대	8대	3대	4대	5대	6대	7대	8대
1	1.664	318.336	955	1273	1592	1910	2228	2547	955.0	1273.3	1591.7	1910.0	2228.4	2546.7
2	3.328	158.336	950	1267	1583	1900	2217	2533	475.0	633.3	791.7	950.0	1108.4	1266.7
3	4.992	105.003	945	1260	1575	1890	2205	2520	315.0	420.0	525.0	630.0	735.0	840.0
4	6.656	78.336	940	1253	1567	1880	2193	2507	235.0	313.3	391.7	470.0	548.4	626.7
5	8.32	62.336	935	1247	1558	1870	2182	2493	187.0	249.3	311.7	374.0	436.4	498.7
10	16.64	30.336	910	1213	1517	1820	2124	2427	91.0	121.3	151.7	182.0	212.4	242.7
20	33.28	14.336	860	1147	1434	1720	2007	2294	43.0	57.3	71.7	86.0	100.4	114.7
30	49.92	9.003		1080	1350	1620	1891	2161	27.0	36.0	45.0	54.0	63.0	72.0
40	66.56	6.336				1521	1774	2028	19.0	25.3	31.7	38.0	44.4	50.7
50	83.2	4.736						1894	14.2	18.9	23.7	28.4	33.2	37.9
60	99.84	3.669							11.0	14.7	18.3	22.0	25.7	29.4
70	116.48	2.907							8.7	11.6	14.5	17.4	20.4	23.3
80	133.12	2.336							7.0	9.3	11.7	14.0	16.4	18.7
90	149.76	1.892							5.7	7.6	9.5	11.3	13.2	15.1
100	166.4	1.536							4.6	6.1	7.7	9.2	10.8	12.3
110	183.04	1.245							3.7	5.0	6.2	7.5	8.7	10.0
120	199.68	1.003							3.0	4.0	5.0	6.0	7.0	8.0
130	216.32	0.798							2.4	3.2	4.0	4.8	5.6	6.4
140	232.96	0.622	261	348	435	522	609	696	1.9	2.5	3.1	3.7	4.4	5.0
150	249.6	0.469	211	282	352	422	493	563	1.4	1.9	2.3	2.8	3.3	3.8
160	266.24	0.336	161	215	269	323	376	430	1.0	1.3	1.7	2.0	2.4	2.7
170	282.88	0.218	111	148	186	223	260	297	0.7	0.9	1.1	1.3	1.5	1.7
180	299.52	0.114	61	82	102	123	143	164	0.3	0.5	0.6	0.7	0.8	0.9
190	316.16	0.020	12	15	19	23	27	31	0.1	0.1	0.1	0.1	0.1	0.2

trade-off로서 결정해야 함을 알 수 있다.



(그림 4) Smax에 따른 총 수용가능 클라이언트수 (Ymax)



(그림 5) Smax에 따른 T(N)의 변화

이번에는 노드수와, T(N)값, 그리고 Rr값이 결정되었을 때, 최대동시사용자를 지원하기 위한 최적의 스트라이핑 크기(S)를 구하여 이를 각각 표와 그래프로 나타내 보자. 이들 표와 그래프를 통하여 각각의 주변여건에 따른 최적의 스트라이핑 크기를 찾아낼 수 있다. 이 결과는 여러 논문에서 지금까지 발표되었던 스트라이핑 정책과는 달리 절대적인 스트라이핑 크기는 존재하지 않으며, 시스템 구성 정책과 HDD의 종류에 따라 적합한 스트라이핑 크기가 달라진다는 것을 말해 준다. 이러한 결과를 바탕으로 하나의 시스템에 장착된 RAID의 최적 스트라이핑 크기를 결정하고자 하는 경우에도 바로 적용할 수 있다. 타 논문에서는 스트라이핑 크기를 결정할 때, 최대 동시사용자와 최종 사용자 접속시 초기대기시간을 전혀 고려하지 않은 상황에서 연구가 진행되었으며, 스트라이핑 크기에 따른 클라이언트 응답시간만을 측정하여 이를 최적의 스트라이핑 크기로 결정하곤 하였다. 그러나, 이번 연구결과

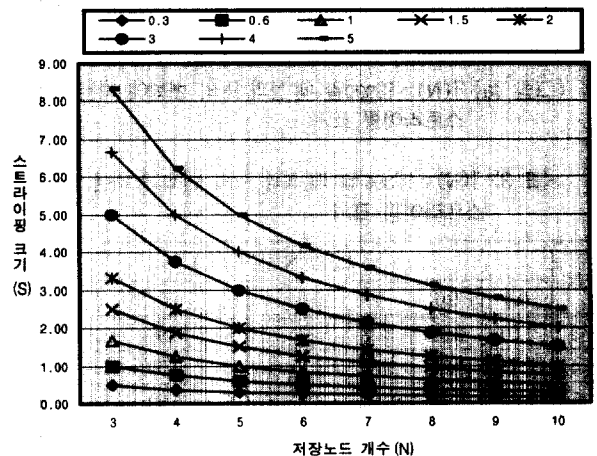
에서 분명히 알 수 있듯이 스트라이핑 크기는 동시사용자의 수에 따라서 응답시간이 달라지게 되며, 따라서 최대사용자 수와 최대지연응답시간을 고려하지 않은 연구는 무의미하다.

이를 확인하기 위해 <표 2>~<표 5>에 T(N)이 결정되었을 때, 저장노드수 N과 Rr값에 따른 최적 스트라이핑 크기를 나타내었다. 이 표들을 통하여 실제 구현시 적절한 스트라이핑 크기를 구현하는데 도움이 될 것이다. 스트라이핑 크기가 최적으로 될 때 Ymax값 또한 최대값을 가지게 될 것이다.

<표 2> T(N) = 5sec 일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

N	Rr(T(N) = 5sec)							
	0.3	0.6	1	1.5	2	3	4	5
3	0.50	1.00	1.67	2.50	3.33	5.00	6.67	8.33
4	0.38	0.75	1.25	1.88	2.50	3.75	5.00	6.25
5	0.30	0.60	1.00	1.50	2.00	3.00	4.00	5.00
6	0.25	0.50	0.83	1.25	1.67	2.50	3.33	4.17
7	0.21	0.43	0.71	1.07	1.43	2.14	2.86	3.57
8	0.19	0.38	0.63	0.94	1.25	1.88	2.50	3.13
9	0.17	0.33	0.56	0.83	1.11	1.67	2.22	2.78
10	0.15	0.30	0.50	0.75	1.00	1.50	2.00	2.50

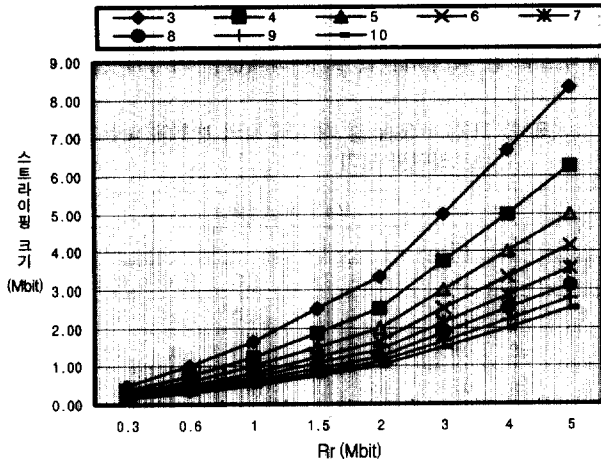
(그림 6)~(그림 7)은 이들 표를 알기 쉽도록 그래프로 표현하였다. (그림 6a)와 (그림 7a)에서는 T(N)값이 각각 5sec, 10sec로 고정되었을 때, 저장노드의 수에 따른 최적 스트라이핑 크기의 변화 그래프를 보여 주는데, 저장노드의 수가 증가함에 따라 완만한 하향곡선을 따라 감소한다. 또한 Rr 값이 커질수록 최적 스트라이핑 크기가 증가함을 알 수 있다. (그림 6b)와 (그림 7b)는 Rr 값과 최적스트라이핑 크기의 관계를 중심으로 나타낸 그래프인데, Rr이 증가함에 따라 최적 S 값이 선형적으로 비례하여 증가한다.



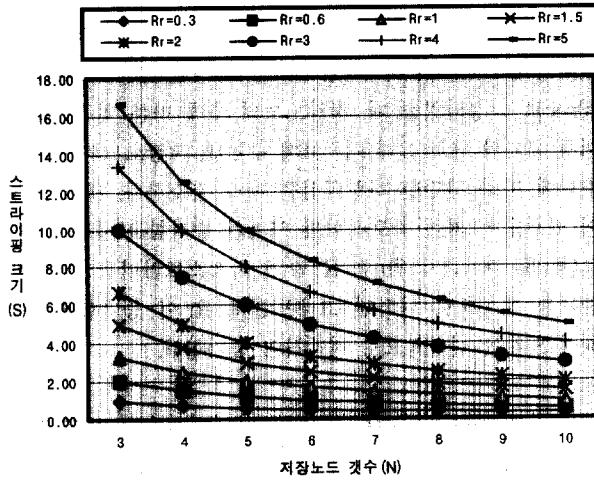
(그림 6a) T(N) = 5sec일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

이들 표와 그림에서 알 수 있듯이 동일한 초기지연대기시

간을 가지도록 할 경우, Rr값의 증가에 따라 최적 스트라이핑 크기는 증가하고, 저장노드의 수가 증가할수록 최적의 스트라이핑 크기는 감소한다. 결론적으로 서비스하고자 하는 데이터의 종류와 저장노드의 수에 따른 스트라이핑 크기를 달리하여 시스템에서 최대의 성능을 발휘할 수 있음을 보여준다.



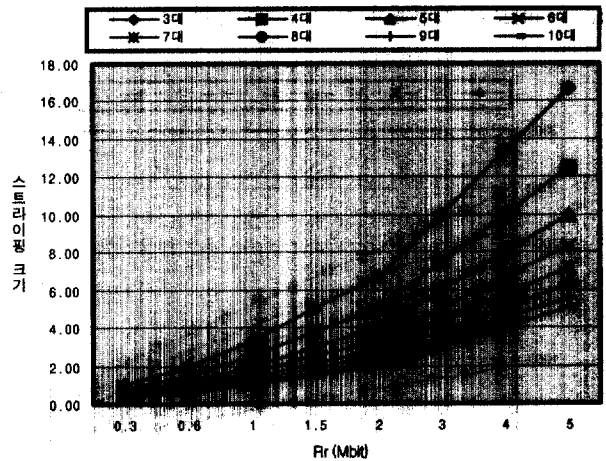
(그림 6b) T(N) = 5sec일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기



(그림 7a) T(N) = 10sec일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

<표 3> T(N) = 10sec 일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

N	Rr (T(N) = 10sec)							
	0.3	0.6	1	1.5	2	3	4	5
3	1.00	2.00	3.33	5.00	6.67	10.00	13.33	16.67
4	0.75	1.50	2.50	3.75	5.00	7.50	10.00	12.50
5	0.60	1.20	2.00	3.00	4.00	6.00	8.00	10.00
6	0.50	1.00	1.67	2.50	3.33	5.00	6.67	8.33
7	0.43	0.86	1.43	2.14	2.86	4.29	5.71	7.14
8	0.38	0.75	1.25	1.88	2.50	3.75	5.00	6.25
9	0.33	0.67	1.11	1.67	2.22	3.33	4.44	5.56
10	0.30	0.60	1.00	1.50	2.00	3.00	4.00	5.00



(그림 7b) T(N) = 10sec일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

실제 시스템 구현시에 가장 먼저 고려할 사항은 최대동시 사용자가 우선이므로 이번에는 최대동시사용자수 Ymax를 먼저 고정하고, 이에 따라 T(N)값을 지정할 때 필요한 스트라이핑 크기를 <표 6>과 같이 계산하였다.

<표 4> T(N) = 20sec 일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

N	Rr (T(N) = 20sec)							
	0.3	0.6	1	1.5	2	3	4	5
3	2.00	4.00	6.67	10.00	13.33	20.00	26.67	33.33
4	1.50	3.00	5.00	7.50	10.00	15.00	20.00	25.00
5	1.20	2.40	4.00	6.00	8.00	12.00	16.00	20.00
6	1.00	2.00	3.33	5.00	6.67	10.00	13.33	16.67
7	0.86	1.71	2.86	4.29	5.71	8.57	11.43	14.29
8	0.75	1.50	2.50	3.75	5.00	7.50	10.00	12.50
9	0.67	1.33	2.22	3.33	4.44	6.67	8.89	11.11
10	0.60	1.20	2.00	3.00	4.00	6.00	8.00	10.00

<표 5> T(N) = 30sec일 때 N과 Rr의 변화에 따른 최적 스트라이핑 크기

N	Rr (T(N) = 30sec)							
	0.3	0.6	1	1.5	2	3	4	5
3	3.00	6.00	10.00	15.00	20.00	30.00	40.00	50.00
4	2.25	4.50	7.50	11.25	15.00	22.50	30.00	37.50
5	1.80	3.60	6.00	9.00	12.00	18.00	24.00	30.00
6	1.50	3.00	5.00	7.50	10.00	15.00	20.00	25.00
7	1.29	2.57	4.29	6.43	8.57	12.86	17.14	21.43
8	1.13	2.25	3.75	5.63	7.50	11.25	15.00	18.75
9	1.00	2.00	3.33	5.00	6.67	10.00	13.33	16.67
10	0.90	1.80	3.00	4.50	6.00	9.00	12.00	15.00

(그림 8)은 <표 6>을 그래프로 나타낸 것으로 최대동시사용자가 정해졌을 때의 최적 스트라이핑 크기 변화를 알 수 있다. 이 그림에서 Ymax가 증가할수록 스트라이핑 크기는

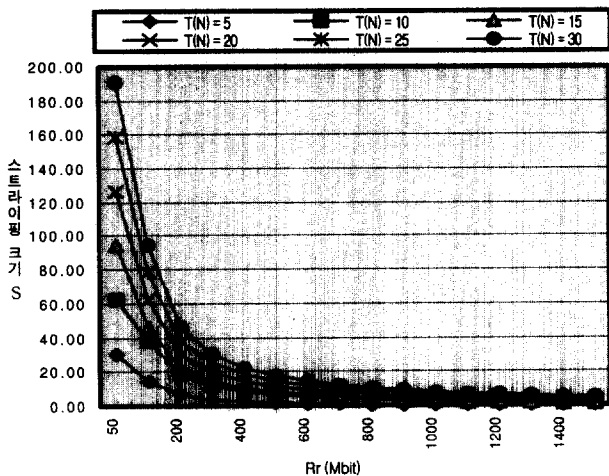


지수함수로 감소함을 알 수 있다.

이 장에서 우리는 최대동시사용자수, 저장 노드수, 최대대기시간, 사용자 요구대역폭에 따라 저장노드의 최적 스트라이핑 크기는 결정되며, 여러 조건에 따른 Trade-off가 필요하다라는 것을 알 수 있었다. 그러므로 최적 스트라이핑 크기는 고정된 값이 아니라 위에서 언급한 모든 조건을 바탕으로 결정하여야 한다. 이 결과는 본 시스템 뿐만 아니라 RAID 시스템에도 적용할 수 있으며, 향후 저장장치 서버 설계 시에도 중요한 요소로 영향을 미치게 될 것이다.

〈표 6〉 Ymax를 고정하였을 때 T(N)의 지정에 따른 최적 스트라이핑 크기(S)

Seagate ST16840 HDD에서 Ymax가 정해질 때 T(N)값의 지정에 따른 최적 스트라이핑 크기 S						
Ymax	T(N) = 5sec	T(N) = 10sec	T(N) = 15sec	T(N) = 20sec	T(N) = 25sec	T(N) = 30sec
50	30.34	62.34	94.34	126.34	158.34	190.34
100	14.34	38.34	46.34	62.34	78.34	94.34
200	6.34	18.34	22.34	30.34	38.34	46.34
300	3.67	11.67	14.34	19.67	25.00	30.34
400	2.34	8.34	10.34	14.34	18.34	22.34
500	1.54	6.34	7.94	11.14	14.34	17.54
600	1.00	5.00	6.34	9.00	11.67	14.34
700	0.62	4.05	5.19	7.48	9.76	12.05
800	0.34	3.34	4.34	6.34	8.34	10.34
900	0.11	2.78	3.67	5.45	7.22	9.00
1000	0.00	2.34	3.14	4.74	6.34	7.94
1100	0.00	1.97	2.70	4.15	5.61	7.06
1200	0.00	1.67	2.34	3.67	5.00	6.34
1300	0.00	1.41	2.03	3.26	4.49	5.72
1400	0.00	1.19	1.76	2.91	4.05	5.19
1500	0.00	1.00	1.54	2.60	3.67	4.74



(그림 8) Ymax를 고정하였을 때 T(N)의 지정에 따른 최적 스트라이핑 크기(S)

### 5. 결 론

본 논문에서는 구현한 리눅스 기반의 병렬 미디어 스트림 서버를 기술하고, 최대 성능을 실현하기 위한 저장 파일들의 스트라이핑 정책을 제시하였다. 일반적으로 VOD서버의 경우, 다중 프로세서와 큰 용량의 메모리를 탑재하고, 저장장치로는 RAID나 SAN을 채택한 고가의 장비에 NT 기반의 VOD 솔루션을 탑재한 경우가 대부분이었다. 또한 이들 시스템에서 파일 저장시 스트라이핑 정책을 심각하게 고려하지 않음으로 인해서 시스템의 성능을 최적으로 사용할 수가 없었다. 이들 시스템 대부분은 최대 사용자를 고려하지 않은 일반적 환경에서의 응답속도만 고려할 수 밖에 없었다. 그러나 본 논문에서 구현된 병렬 미디어 스트림 서버는 한계사용자 서비스 상황을 예측하고 이를 기반으로 시스템 RAID라는 저장장치 클러스터링 기법을 도입하여 최적의 스트라이핑 정책과 이를 위한 시스템 구성 요건을 제시함으로써 타 시스템에 비하여 성능을 크게 향상시킬 수 있었다. 이 시스템은 하나의 컨트롤 노드와 여러 대의 저장노드들로 구성되는데, 저장노드의 수는 성능 요구에 따라 유연한 확장이 가능하다. 그러나 네트워크 대역폭과 스트라이핑 크기를 고려한 한 시스템의 저장노드의 수는 10개 이내로 제한하는 것이 바람직하며, 저장 노드들을 네트워크에 분산할 경우에는 설계에 따라 그 개수를 얼마든지 늘여날 수 있다. 저장 노드들은 각 클라이언트에 대하여 균일한 부하분산을 통하여 서비스를 제공하는데 접속 클라이언트 수가 적거나 많음에 관계없이 서비스 부하를 동일하게 나누어 제공한다. 이 시스템은 서비스하고자 하는 파일의 특성에 따라 쪼개고자 하는 최적의 스트라이핑 크기를 제시하여, 각 서비스 종류별, 저장 노드들의 수에 따른 시스템 최적화를 이룬다. (그림 6)과 (그림 7)에서 알 수 있듯이 동일한 초기지연대기시간을 가지도록 할 경우, Rr값의 증가에 따라 최적 스트라이핑 크기는 증가하고, 저장노드의 수가 증가할수록 최적의 스트라이핑 크기는 감소한다. 즉, 서비스하고자 하는 데이터의 종류와 저장노드의 수에 따른 스트라이핑 크기를 달리하여야 시스템에서 최대의 성능을 발휘할 수 있음을 보여준다. (그림 8)에서는 Ymax가 증가할수록 스트라이핑 크기는 지수함수적으로 감소함을 알 수 있다.

결론으로 최대동시사용자수, 저장 노드수, 최대대기시간, 사용자 요구대역폭에 따라 저장노드의 최적 스트라이핑 크기는 결정되며, 여러 조건에 따른 Trade-off가 필요하다. 그러므로 최적 스트라이핑 크기는 고정된 값이 아니라 본론에서 언급된 모든 조건을 바탕으로 결정하여야 한다. 이 결과는 본 시스템뿐만 아니라 RAID 시스템에도 적용할 수 있으며, 향후 저장장치 서버 설계 시에도 중요한 요소로 영향을 미치게 될 것이다.

참 고 문 헌

[1] Dan A., Sitaram D., Shahabudden P., "Scheduling Policies for an On-Demand Video Server with Batching," Proceedings of the 2<sup>nd</sup> ACM Multimedia Conference, Sna Fransco, CA, pp.25-32, 1994.

[2] Debasish Ghose, Hyung-joong Kim, "Delivery Data in a VoD System," Journal of Telecommunications and Information, Vol.2, 1998.

[3] 고정국, 김길용, "소규모 VOD시스템의 저장 서버로서 디스크 배열 구조의 분석", 한국정보처리학회논문지 제4권 제3호, 1997.

[4] Peter M. Chen and David A. Patterson, "Maximizing Performance in a Striped Disk Array," Proc. Of the 1990 International Symposium on Computer Architecture, pp.322-331, May, 1990.

[5] <http://www.Microsoft.com/korea/ntserver/mediaserv/exec/overview/MedOverview.asp>.

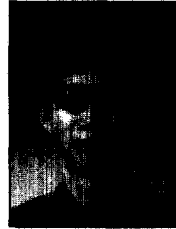
[6] <http://www.Microsoft.com/korea/ntserver/mediaserv/exec/comparison/ServVTheater.asp>.

[7] Seth McEvoy, "Microsoft Windows Media Player 7 Handbook," Microsoft, 2000.

[8] 차호정, 홍지훈, "효율적 자원 관리를 제공하는 SCI 기반의 VOD 서버", 정보과학회논문지(A) 제24권 제5호, 1997.

[9] Dowd, Kevin, and Charles Severance, "High Performance Computing," O'Reilly&Associates, 1998.

[10] Foster, Ian T., "Designing and Building Parallel Programs : Concepts and Tools for Parallel Software.," Addison-Wesley Publishing Co., 1995.



김 서 균

e-mail : skkim466@dreamwiz.com

1992년 전남대학교 전자공학과(공학사)

1996년 전남대학교 전자공학과(공학석사)

1997년~현재 전남대학교 전자공학과 박사과정

1993년~1994년 ㈜펜앤브레인즈시스템즈

1998년~2000년 DK정보통신 사장

2000년~현재 ㈜포스트립 기술이사

관심분야 : 멀티미디어 네트워크, 병렬VOD서버, 실시간 통신 시스템, 병원정보시스템



남 지 승

e-mail : jsnam@chonam.ac.kr

1981년 인하대학교 전자공학과(공학사)

1985년 University of Alabama, Electrical Engineering(공학석사)

1992년 University of Arizona, Electrical & computer Engineering(공학박사)

1992년~1995년 한국전자통신연구소 선임연구원

1995년~현재 전남대학교 컴퓨터공학과 부교수

관심분야 : 컴퓨터 네트워크, 실시간 시스템, 병원정보시스템