리눅스 기반의 고성능 병렬 미디어 스트림 서버 설계 및 구현

김 서 균[†]·김 경 훈[†]·류 재 상[†]·남 지 승[††]

요 약

멀티미디어 서버 시스템은 고용량이어야 하며 지속적으로 늘어나는 사용자수 뿐만 아니라 추가되는 새로운 저장 공간에 대한 우수한 확장성을 제공하여야 하는 것이 필수적이다. 일반적인 스트리밍 서비스의 경우, 사용자들은 서비스 초기의 지연시간에 어느 정도 관대한 편이지만 고품질의 서비스를 요구한다. 스트림 서버는 동영상을 사용자에게 전달할 때 데이터 저장공간으로부터 실시간으로 전송하여야 한다. 그러나 현재의 범용 서버 시스템은 이러한 요구사항을 충분히 반영하지 못할 뿐만 아니라 늘어나는 사용자 부하와 시스템 요구에 대한 고려, 그리고 미디어 데이터에 대한 반영이 이루어지지 못하여 점차 증가되는 사용자의 고화질 미디어 서비스 요구 사항을 충족시키지 못하고 있다. 본 논문에서는 리눅스를 기반으로 구현한 실시간 스트림 서버 시스템의 확장성 있는 구조가 고 대역폭 고화질 On-Demand 서버로서 효율적인 대안임을 보이며, 또한 QoS 요구 보장과 효율적인 시스템 관리 정책을 제시하여 범용 서버를 멀티미디어 저장 및 스트리밍에 적합한 환경의 클러스터로 구성하는 방법을 제시한다. 이 시스템의 특징은 서비스하고자 하는 파일들을 각 병렬 저장 시스템에 쪼개어 저장하는 시스템 RAID 기술을 사용하여 사용자 요구시 동시에 데이터를 전송하여 줌으로써 단일 미디어 서버보다 훨씬 우수한 성능을 발휘한다.

Design And Implementation of Linux Based Parallel Media Stream Server System

Seo-gyun Kim[†] · Kyung-hoon Kim[†] · Jae-sang Ryu[†] · Ji-seung Nam[††]

ABSTRACT

Multimedia service systems should have efficient capacity to serve the growing clients and new data. In the general streaming services, users can endure the small amount of time delay at the beginning of service. But they want to have good quality of service. A streaming server tries to transfer video files to clients from a repository of files in real time. The server must guarantee concurrent and uninterrupted delivery of each video stream requested from clients. To achieve its purpose, many stream servers adopt multi-processors, sufficient memory, and RAID or SAN in their systems. In this paper, we propose a Linux-based parallel media streaming server. It is superior to the other systems in the storing structure, fault-tolerance, and service capacity. Since this system supports the web interface, users can operate easily through the www. This system uses unique striping policy to distribute multimedia files into the parallel storage nodes. If a service request occurs, each storage node transmits striped files concurrently to the client. Its performance is better than the Single media streaming service because of the parallel architecture.

키워드 : 리눅스(Linux), 병렬 미디어 스트리밍 서버(Parallel media streaming server), 컨트롤 노드(control node), 저장노드(storage node), 스트라이핑(striping), 스트림(stream)

1. 서 론

멀티미디어 서버는 그 용량의 방대함뿐만 아니라 고속의 네트워크 자원, 또한 사용자 접근의 동시성 및 임의성, 자원의 분배 및 활용, 제공 컨텐츠 특성에 따른 제어 등 시스템 성능을 위한 많은 부분이 고려되어 설계되어야 한다[1]. 기존 솔루션은 상대적으로 취약한 네트워크 자원을 최대로 활용하기 위한 스트리밍 기술 위주로 연구 개발이 이루어졌다. 그러나 네트워크 인프라가 각 가정까지 확산되면서 이를 기반으로 한 새로운 고품질 서비스를 요구하게 되었다. 이러한 요구를 수용하기 위해서는 미디어 특성을 최대로 반영하는 고성능 미디어 서버가 필요하다. 다시 말하면 지역적인 멀티미디어 서비스 환경을 광역 네트워크로 확장하는 것으로 사용자에게 로컬 환경과 같은 수준의 서비스를 가능하게 해주는 강력한 서버가 필요하다.

기존의 범용 서버를 최적화된 미디어 서버로 이용하기 위해서는 하드웨어와 미디어 전송 및 관리 소프트웨어의 두 가지의 구성요소가 필요하며, 관리자들은 이들을 적절히

† 준 회 원 : 전남대학교 대학원 전자공학과
†† 종신회원 : 전남대학교 컴퓨터공학과 교수
논문접수 : 2001년 10월 8일, 심사완료 : 2001년 11월 12일

운용하기 위하여 많은 시간과 노력을 기울인다. 하드웨어는 많은 용량과 빠른 전송 속도를 갖는 저장 장치가 가장 중요한 요소이며 충분한 데이터 전송을 위한 I/O 장치 역시 중요하다. 저장장치의 물리적인 성능뿐만 아니라 효율적인 데이터 배치 및 관리 도구가 필수적인데 이것은 매우 임의적인 사용자 서비스 요구에도 불구하고 예상 가능한 다양한 변수가 존재하기 때문이다. 영화 서비스를 위한 서버의 경우 동시에 요구되는 미디어는 전체 저장 미디어의 극히 일부일 가능성이 있고 장시간 서비스를 받는 사용자의 특성에 따라 초기지연 시간을 조정하는 등 부족한 시스템 자원을 최대로 이용하게 할 수 있는 다양한 제어를 통하여 그 같은 상황에 맞추어 시스템 운영을 탄력적으로 할 수 있기 때문이다. 또한 새로운 형식의 미디어 파일 타입과 로컬 타입의 컨텐츠를 수정 없이 네트워크 상에서 이용하기 위한 제어부분 역시 미디어 솔루션의 중요 요소가 되었다.

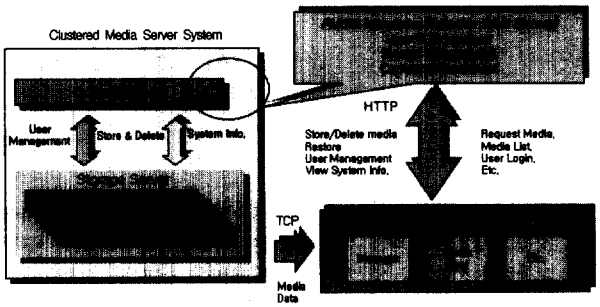
구현된 시스템은 리눅스를 채택하여 기본적으로 여러 대의 서버가 동시에 하나의 클라이언트를 위해 다중의 접속경로를 갖는 형태로 구성되었다. 미디어 종류와 관계없이 모든 파일 형식 지원을 위하여 기본적으로 TCP가 전송 프로토콜로 선택되었고 미디어 특성에 맞는 데이터 흐름을 자율적으로 조정하고 여러 서버의 제어흐름을 조정하고 관리하기 위한 제어 서버 노드가 별도로 존재한다. 이는 인터페이스용 웹 데이터, 데이터베이스 트랜잭션 데이터 등 다른 타입의 데이터 전송을 미디어 전송서버로부터 분리시키기 위한 목적으로 존재한다. 또한 사용자 인터페이스로서의 웹 서버와 미디어 관리용 데이터 베이스, 각 저장 서버에 데이터를 분산 저장하고 컨텐츠를 관리 또는 재배치하는 관리자용 부 프로그램들로 전체 시스템이 구성된다. 이와 같은 구조로 인하여 사용자 부하를 효율적으로 각 서버 노드에 분산시키고 새로운 노드와 저장 공간 추가시에 생기는 중복저장의 단점을 해결하는 새로운 형식의 시스템 구조를 가지게 되었다.

2. 본 론

2.1 시스템 구조 및 서비스 방법

구현된 VOD(Video-on-demand) 저장 서버는 하나의 컨트롤 서버와 이의 제어를 받는 최소 2대 이상의 저장 노드 전송 서버들로 구성된다. 컨트롤 서버는 사용자 인터페이스를 위한 웹서버와 DB가 설치되며 각 노드들의 부하와 시스템 자원상태를 제공받는 프로그램 그리고 컨트롤 노드의 데이터를 분산 저장하고 재배치, 관리하는 프로그램들로 구성된다. 저장 서버에는 클라이언트로부터 요구 받은 데이터를 전송하는 프로그램이 설치된다.

이와 같은 기능적인 분배는 저장노드가 다른 특성을 갖는 데이터 전송으로 인하여 디스크와 I/O 장치를 소모하는 오버헤드 감소를 위해 고려되었다. 저장 노드는 동일한 시스템 환경으로 구성하는 것이 가장 바람직하지만 다른 시스템들의 사용도 관리자의 올바른 배치와 시스템 관리로 성능을 최대화 할 수 있다[2]. 클라이언트에는 다중 접속경로를 갖는 데이터 수신 모듈이 탑재되고 이는 기존 플레이어 코덱에 데이터를 공급한다.



(그림 1) 전체 시스템 구조

사용자는 먼저 컨트롤 노드의 웹서버를 통하여 데이터 목록을 확인하고 사용자 인증 후 서비스 요청을 한다. 컨트롤 서버는 사용자가 선택한 데이터의 저장 구조와 위치 등을 표시한 메타 데이터를 사용자 클라이언트에 전송하고 이 메타 데이터를 이용하여 클라이언트는 저장 노드들과 다중 접속 경로를 설정하고 동시 수신 받은 데이터를 조합하여 미디어를 재생하게 된다. (그림 1)은 전체 시스템 구조를 나타낸 것이다. 새로운 저장 노드가 추가됨으로써 전체 시스템은 동시 사용자 로드를 전체 시스템에 분산한다. 또한 새로운 노드에 장착된 저장 디스크는 기존 저장 데이터의 재분산 저장을 통하여 전송 부하를 저장 노드간에 분산 할 수 있다. 극단적으로 전체 데이터의 재분산을 통하여 전체 시스템 부하를 분산할 수 있고 필요에 따라 앞에서 언급한 바와 같이 일부 인기 데이터에 대한 재분산만 실시하여 로드 균형을 맞출 수도 있다. 이것은 전체적으로 시스템 가용성에 중점을 둔 시스템의 설계 원칙을 반영하는 것으로써 관리자는 충분한 시스템 상황과 서비스 상황을 모니터하고 이와 같은 정책에 따라 시스템 환경을 구성한다.

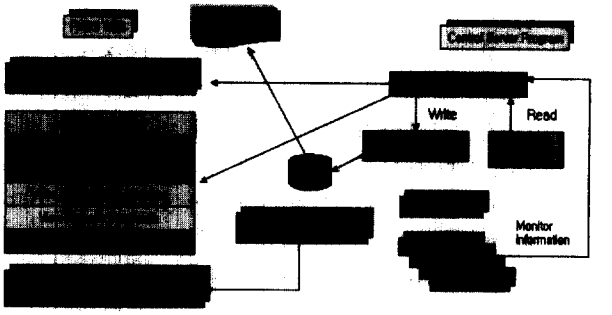
시스템은 데이터의 분산 저장정책에 의하여 전체 시스템 성능에 영향을 미치는데 효율적인 관리자의 시스템 환경 구성을 위한 올바른 시스템 성능 모니터링은 이러한 관점에서 매우 중요하다. 또한 시스템의 성능 관련 모니터링은 사용자 수용에 관한 적합한 정보를 수용 제어기에 제공하여야 하는데 잘못된 정책으로 인한 사용자 수용으로 인하여 전체 사용자에게 서비스 품질을 제공할 수 없는 상황이 발생하는데 이것은 최악의 상황으로 반드시 고려되어야 하는 부분이다. 따라서 성능을 최대화 할 수 있는 충분한 시스템 모니터링의 구현과 이를 토대로 시스템 가용성을 최대로 할 수 있도록 하는 시스템 구성 정책, 또한 이러한 정책을 통하여 산출된 값을 통하여 안정된 서비스 수준의 사용자 수에 따라 시스템을 운용하고 보다 안정된 고품질 미디어 서비스를 위해서는 적합한 예비율을 산출 시스템 폭주에 대비하여야 한다. 개발된 시스템은 관리자와 사용자 모두 웹 인터페이스를 통한 접근을 하는데 관리자는 미디어 데이터를 컨트롤 서버에 업로

드하여야 하고 이를 각 저장 서버에 스트라이핑 정책에 따라 분산 저장한다. 이와 같은 이중 데이터 저장은 컨트롤 노드에 중요한 원시 미디어 데이터를 저장하여 여러 노드에 걸쳐 데이터를 분산 배치하는 데서 생기는 데이터 손실과 에러에 대비하는 백업 및 복구 데이터로 사용할 수 있다. 이 같은 구조 때문에 컨트롤 서버에는 전체 저장 노드의 데이터 용량 만큼의 저장 공간이 필요하게 되나 사용자 데이터의 중요성에 따라 이러한 백업 대상 데이터의 보관 여부를 결정할 수 있다. 구현된 시스템은 시스템 RAID를 통하여 RAID 수준 이상의 성능을 기대할 수 있고, 시스템 확장은 추가 노드의 증설로 간단히 확장되며, 이에 대한 별도의 부하 분배기가 필요하지 않는 점 등이 주요 장점이다.

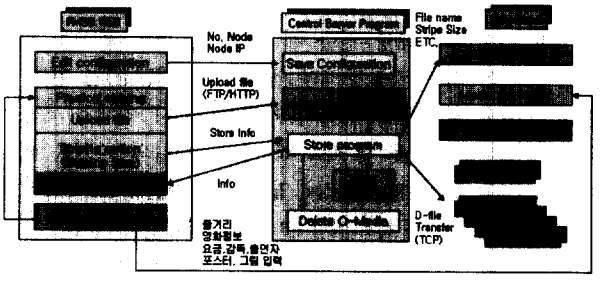
사용자가 미디어 서비스를 이용하는 과정은 다음과 같다. 사용자는 컨트롤 서버에 접속하여 관리자가 데이터를 분산 저장하면서 자동으로 생성된 미디어 정보를 통해 서비스 요구 데이터를 결정하고 컨트롤 서버로부터 실제 데이터가 저장된 분산 서버들과 저장 스트라이핑 사이즈, 서버 오류로 인하여 서비스 중단시 재전송을 맡게 될 백업 시스템 서버 정보 등 스트리밍 서비스를 위한 정보를 가진 메타 데이터를 전송 받는다. 전송된 메타 데이터를 통한 정보로 사용자 클라이언트는 각각의 저장서버와 연결을 설정하고 저장 정보와 분산 데이터 크기에 따라 적합한 버퍼링과 전송 데이터 블록을 전송 받고 이를 조합하여 미디어를 재생하게 된다. 각 저장 서버는 사용자 스트림 정보를 컨트롤 서버에 전송함으로써 컨트롤 서버는 현재 수용되어 있는 사용자 정보 서비스 중인 미디어 정보 및 타입 등의 관리 데이터를 보유하게 된다. 이것은 기존의 스트리밍 서버가 메타 데이터를 획득한 불법 사용자로 인한 문제를 해결 할 뿐만 아니라 새로운 사용자를 수용하는 정책을 위한 입력 데이터의 역할은 한다.

(그림 2)는 시스템의 서비스 인터페이스 작성과 관리를 위한 구조를 나타낸 것이고, (그림 3)은 데이터 저장 구조를 도식화 한 것이다. 미디어 서버를 위해서는 기존 서버에 단지 미디어 데이터를 네트워크 상으로 전달하고 이 데이터를 수신하고 재생하는 프로그램들로 기본적으로 그 역할을 수행한다. 하지만 범용 시스템들로 많은 다수의 동시 접속자에게 서비스를 제공하여야 하는 서버를 구성하기에는 좀 더 미디어 서비스 측면의 시스템 구성이 필요하다. 개발 시스템은 이미 기본적인 성능이 검증된 상태로 포스트림에서 제품으로 출시되었으며 다양한 형태로 테스트되고 문제점이 지속적으로 개선되고 있다. 본 시스템은 분산 저장과 병렬 전송 등의 특징적인 방법을 사용하여 시스템의 성능과 부하 분산을 수행하였으며 기본적으로 미디어 서비스 시 요구되는 기능에 대한 부프로그램들로 효율적이고 고성능의 미디어 서버 시스템을 설계 구현하였다.

구현한 병렬 미디어 스트림 서버는 RAID(Redundant Arrays of Inexpensive Disks) 레벨 0의 개념을 네트워크 시스템에 확장 적용한 구조를 가진다. RAID level0는 하나의



(그림 2) 시스템 인터페이스 및 관리 구조

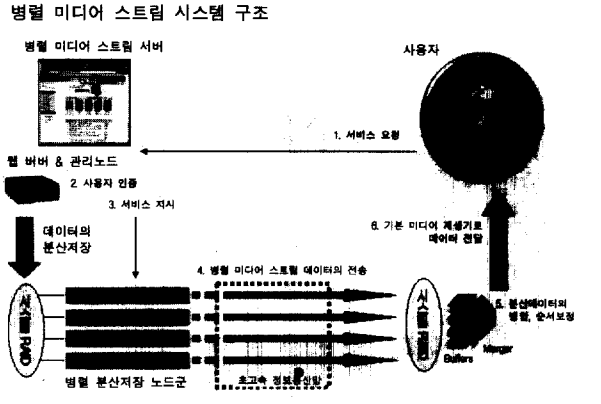


(그림 3) 데이터 저장 구조

큰 파일을 디스크어레이 내의 모든 디스크에 스트라이핑이라는 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 가용성과 시스템 확장성을 확대하였다. LINUX를 채용하여 시스템의 안정성을 확보하였고, 또한 웹 기반의 편리하고 다양한 관리도구를 개발하여 서버관리자의 편의성을 증가시켰다.

(그림 4)에 병렬 미디어 스트림 서버의 개요를 묘사하고 서비스 순서를 간략히 나타내었다. 그림을 보면 먼저 사용자가 웹을 통하여 병렬미디어 스트림 서버에 접속하여 동

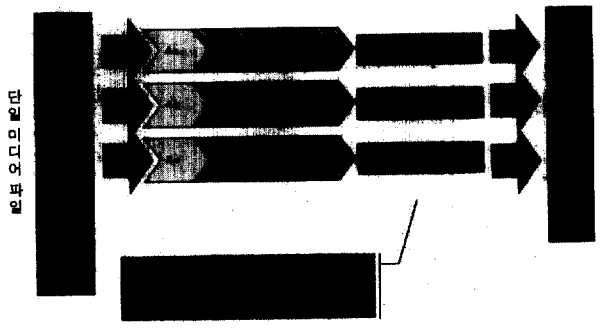


(그림 4) 병렬미디어 스트림 서버의 개요도

영상 서비스 요청을 한다. 관리노드는 로그인 과정을 통한 인증으로 허가된 사용자 여부를 확인하고, 사용자에게 저장 노드와 요청 데이터의 정보가 담긴 메타파일을 전송함과 동시에 저장 노드들에게 해당 사용자에 대한 서비스 지시를 내린다. 이때부터 저장 서버들은 동시에 각 사용자에게 접속되어 미리 스트라이핑되어 저장되어 있는 스트림 데이터를 전송한다. 사용자측에서는 각 저장 서버들로부터 도착된 데이터를 필터를 통하여 데이터를 재조합하고, 미디어 재생기로 보낸다.

2.2 시스템 RAID

서버 데이터의 대량 전송 시 가장 큰 병목 현상을 나타내는 부분은 바로 저장장치의 액세스이다. RAID는 원래 데이터 손실을 방지하기 위하여 개발되었으나 액세스 속도가 늦은 하드디스크들을 병렬로 배열하여 하나의 파일을 배열된 디스크들에 쪼개어 저장하였다가 파일전송 요구 시 동시에 동작함으로써 시스템 I/O버스를 최대한 이용할 수 있도록 개선되었다. 그러나 그 구축비용이 만만치 않을 뿐 아니라 가격 대 성능비가 충분하지가 않다. 그러므로 RAID 시스템 구축비용으로 시스템을 여러 대 구입하여 이들 시스템별 저장 장치들을 네트워크로 연결하여 그 구조가 마치 RAID 레벨 0의 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 5)는 한 파일이 각 시스템에 스트라이핑 되어 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 저장서버에서는 스트라이핑된 조각들이 하나의 파일로 재구성되어 같은 이름으로 저장된다. 바로 이점이 RAID LEVEL 0와 다른 점이다. 이 것의 최고 장점은 하나의 파일이 각기 다른 시스템에 스트라이핑되어 분산 저장되어 데이터 전송 요구 시 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 없이도 균일한 시스템 부하분산을 이룰 수 있다.



(그림 5) 시스템 RAID 개념도

2.3 성능 평가

이 절에서는 단일 서버 시스템과 저장노드수가 3대인 병렬미디어 스트림 서버를 기준으로 비교분석을 하여 제안한 병렬 스트리밍 서버의 성능을 평가하였다. 비교 분석을 위하여 오퍼레이팅 시스템(NT&Linux)의 차이에 따른 부하처리속도는 동일하다고 가정하였다.

사용자들이 요구하는 데이터량에 따라 서버가 이들 요구를 수용하고 데이터를 전송해 줄 때 사용자가 늘어남에 따라 전송지연이 발생하게 된다. 이 때 서버의 한계 용량 이상으로 사용자가 접속하게 되면 전송지연이 증가하게 되어 서버에 접속한 사용자들은 원활한 동영상 서비스를 받지 못하게 된다. 따라서 스트림 서버의 전송지연시간을 계산하여 허용 가능한 최대지연시간 안에 해당되는 서비스 용량을 서버의 최대수용량으로 하여 이를 비교하고자 한다.

먼저 NT 기반의 단일 서버의 전송 지연시간을 계산하기 위하여 전송지연시간을 유도하였다. 우선 수식에 사용되는 변수를 정의하면 다음과 같다.

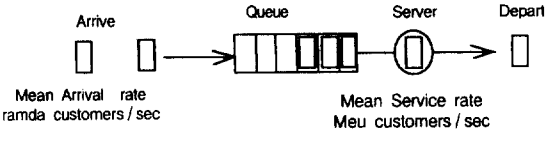
- 비트율에 의한 서버 용량 : μ_1(bits/sec)
- 요구 도착률 : λ_1(frames/sec)
- 패킷 길이 : C(bits/frame)

서버가 수용 가능한 최대 비트율은 μ_1이므로, 각 사용자에 대해 초당 전송해야 할 프레임의 크기를 C라 하면 서버가 단위 시간 안에 서비스 가능한 최대 프레임 수는 다음과 같다.

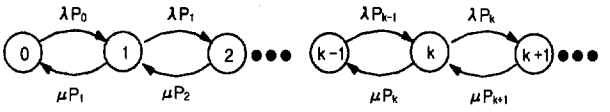
$$\text{서버 서비스율} : \mu = \frac{\mu_1}{C}\ (frames/\sec)$$

여기에서 평균지연시간 T를 구하기 위하여 큐잉 모델을 도입하여 유도하였다. 이를 위해 M/M/1 큐잉 모델을 사용하였다. 이 모델은 Birth-Death process라 불리며 다음과 같이 정의된다.

- Birth-Death process의 범위
  - 시간독립성 : 사용자 요구의 도착 및 서비스 시간은 이전의 서비스요구 도착 및 서비스 시간과 독립적이다.
  - 분리된 상태 공간 : 시스템 내에서의 사용자에 대한 상태 공간은 분리되었다.



(그림 6) M/M/1 큐잉 모델



mean number of transitions/sec from state 0 to 1 = λP_0

mean number of transitions/sec from state 1 to 0 = μP_1

(그림 7) M/M/1 큐잉 모델의 상태 천이도

(그림 6)에 M/M/1 큐잉 시스템 모델을 나타내었다. Pk를 시스템(큐와 서버)내에 k개의 사용자가 있는 안정 상태 확률이라 하면 이 모델의 상태 천이도는 (그림 7)과 같이 표현된다. 여기서 시스템 내의 평균 사용자수 N은

$$N=\frac{\lambda}{\mu-\lambda} \quad (1)$$

이다[8]. 여기서, Little's Result에 의하여 시스템 내의 평균 사용자 숫자는 평균 도착률과 시스템 내에서의 사용자가 소요한 평균대기시간의 곱으로 식 (2)와 같이 놓을 수 있다.

$$N=\lambda T \quad (2)$$

그러므로 서비스 시간을 포함한 총 지연시간 큐잉 지연은 다음과 같다.

$$T=\frac{N}{\lambda}=\frac{\rho}{(1-\rho)\lambda}=\frac{1/\mu}{(1-\rho)}=\frac{1}{\mu-\lambda} \quad (3)$$

위의 결과를 바탕으로 $\lambda_1=\lambda$로 놓으면 단일 서버에 대한 T는 다음 식이 된다.

$$\text{평균시간지연}: T_1=\frac{1}{\frac{\mu_1}{C}-\lambda} \quad (4)$$

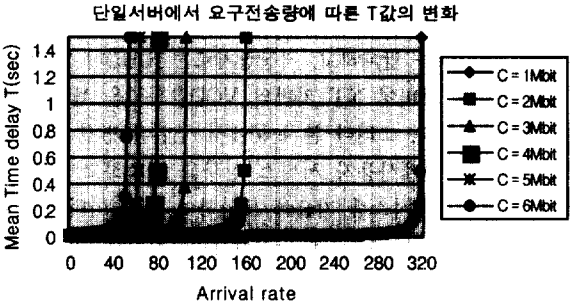
이제 단일 서버 시스템을 기반으로 병렬 미디어 스트림 서버의 T를 유도하여 보자. 병렬 스트림 서버인 경우 사용자가 요청한 스트림 서비스를 각 서버들이 균일하게 분할하여 데이터전송을 담당하므로, μ 값을 다음과 같이 다시 정의하고 수식을 유도하면 다음과 같다.

- 클러스터 노드 용량 : μ_1(bits/sec)
- 요구도착률 : $\frac{\lambda_1}{N}(frames/\sec)$
- 패킷 길이 : C(bits/frame)
- 서버 서비스율 : $\frac{\mu_1}{C}(frames/\sec)$

그러므로,

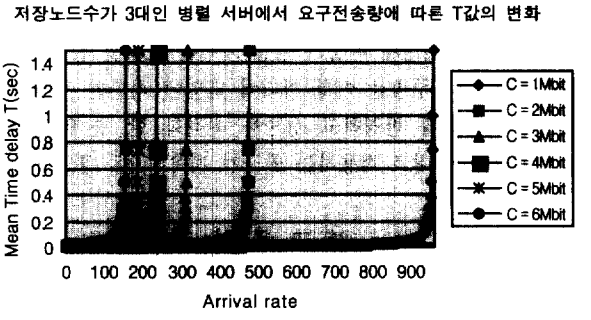
$$\text{평균시간지연}: T_N=\frac{1}{\frac{\mu_1}{C}-\frac{\lambda}{N}}=\frac{N}{\frac{\mu_1}{C}N-\lambda} \quad (5)$$

성능 비교를 위하여 단일서버의 T값에 대한 그래프와 병렬서버에 대한 T값의 그래프를 그려서 비교해보자. μ_1을 구현에 사용된 서버 저장장치의 액세스 속도와 동일한 320Mbps라 하고, 초당 전송할 패킷의 길이를 1Mbits/frame라 하면, 서버 서비스율은 320frames/sec가 된다. 이를 이용하여 평균시간지연 T를 계산하여 이를 그래프로 표현하였는데, 단일 서버에 대한 그래프는 (그림 8)에 나타내었고, 병렬서버에 대한 그래프는 (그림 9)에 나타내었다.



(그림 8) 단일서버에서 요구전송량에 따른 T값의 변화

(그림 8)(그림 9)를 살펴보면, 사용자가 늘어남에 따라 T값은 아주 조금씩 증가하다가, 서버의 최대수용량에 가까워질 때 T값이 급격히 증가함을 알 수 있다. 실시간 동영상 스트림 서비스를 하기 위해서는 T가 1의 범위 내에 있어야 하므로, 1의 범위 내에 있는 도착률의 수치가 최대수용가능 자수가 된다. 두 그래프의 결과를 비교해 보면 단일 서버에 비해 병렬 서버의 도착률이 매우 크다는 것을 알 수 있다. 이로써 병렬 미디어 스트림 서버는 단일 미디어 서버보다 최소 3배 이상의 성능 차이를 보임을 증명하였다.



(그림 9) 저장노드수가 3대인 병렬 서버에서 요구전송량에 따른 T값의 변화

3. 결 론

미디어 서버뿐만 아니라 서버시스템은 그 안정적인 운영이 무척 중요시된다. 본 시스템 역시 이와 같은 안정성 보장을 위해 백업 서버 운영방안 또한 다수의 서버의 공동 작업인 서비스 형태의 보완을 위한 연구를 지속적으로 수행하고 있다. 향후 인터넷 서비스의 주종을 이룰 데이터는 고용량 고품질의 멀티미디어 데이터가 될 것이다. 네트워크의 인프라는 마치 프로세서의 발전 속도와 같이 비약적으로 발전하고 있으나 이와 같은 추세에 맞는 서비스 시스템은 상대적으로 부

족하다. 많은 수요의 멀티미디어 서버가 필요하게 될 것이며 따라서 다수 사용자에게 안정적으로 서비스를 제공할 서비스 시스템 역시 지속적으로 연구되어야 한다. 구현한 시스템은 병렬 전송과 분산저장으로 기본적으로 시스템 확장을 부하 분배기나 백업 시스템 등 의 도움 없이도 구성 할 수 있도록 개발되었다. 또한 미디어 서비스를 위한 각종 정책과 관리도구들을 활용해 미디어 서버의 성능을 최대화 할 수 있도록 고려했다. 일반적으로 VOD서버의 경우, 다중 프로세서와 큰 용량의 메모리를 탑재하고, 저장장치로는 RAID나 SAN을 채택한 고가의 장비에 NT 기반의 VOD솔루션을 탑재한 경우가 대부분이었다. 그러나 본 논문에서 구현된 병렬 미디어 스트림 서버는 한계사용자 서비스 상황을 예측하고 이를 기반으로 병렬 저장 시스템 기술을 도입하여 단일 미디어 스트림 서버에 비하여 성능을 크게 향상시킬 수 있었다. 이 시스템은 하나의 컨트롤 노드와 여러 대의 저장노드들로 구성되는데, 저장노드의 수는 성능 요구에 따라 유연한 확장이 가능하다. 저장 노드들은 각 클라이언트에 대하여 균일한 부하분산을 통하여 서비스를 제공하는데 접속 클라이언트 수가 적거나 많음에 관계없이 서비스 부하를 동일하게 나누어 제공한다. 단일 미디어 스트림 서버와의 성능을 비교한 (그림 8)(그림 9)를 살펴보면, 사용자가 늘어남에 따라 서비스 시간 지연값은 아주 조금씩 증가하다가, 서버의 최대 수용량에 가까워질 때 그 값이 급격히 증가함을 알 수 있다. 실시간 동영상 스트림 서비스를 하기 위해서는 서비스 시간 지연값 T가 1의 범위 내에 있어야 하므로, 1의 범위 내에 있는 요구도착율이 최대수용가능자수가 된다. 두 그래프의 결과를 비교해 보면 단일 서버에 비해 병렬 서버의 요구 도착율이 매우 크다는 것을 알 수 있다. 이로써 병렬 미디어 스트림 서버는 단일 미디어서버보다 최소 3배 이상의 성능 차이를 보임을 증명하였다. 향후 서비스의 주종을 이룰 미디어 타입에 적합한 정책을 연구하여 시스템에 반영하고 성능을 최적화 할 수 있는 관리 구성 값을 찾아내고 적용하는 연구가 지속적으로 필요하다.

참 고 문 헌

[1] Huang-Jen Chen T. D. C Little, "Storage Allocation Policies for Time-Dependent Multimedia Data," IEEE Trans. On Knowledge and Data Engineering, Vol.8, No.5, pp.855-864, 1996.

[2] S. Moyer and V.Sunderam, "Parallel I/O as a Parallel Application," Journal of Supercomputer Application, Vol.9, No. 2, 1995.

[3] MicroSoft, "MicroSoft Media Service SDK," 1998.

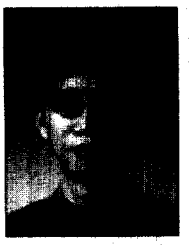
[4] Dan A., Sitaram D., Shahabudden P., "Scheduling Polices for an On-Demand Video Server with Batching," Proceedings of the 2nd ACM Multimedia Conference, Sna Fransco, CA, pp.25-32 1994.

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

[6] 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.

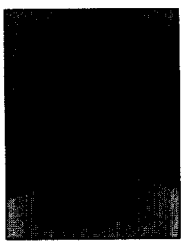
[7] Foster, Ian T., "Designing and Building Parallel Programs : Concepts and Tools for Parallel Software," Addison-Wesley Publishing Co., 1995.

[8] Michael K. Molloy, "Fundamentals of Performance Modeling," Macmillan Publishing Company, pp.161-180, 1989.



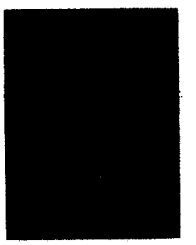
김 서 균

e-mail : skkim@chonnam.ac.kr
1992년 전남대학교 전자공학과(공학사)
1996년 전남대학교 전자공학과(공학석사)
1997년~현재 전남대학교 전자공학과 박사과정
1993년~1994년 ㈜펜앤브레인즈시스템즈
1998년~2000년 DK정보통신 사장
2000년~현재 ㈜포스트림 기술이사
관심분야 : 멀티미디어 네트워크, 병렬VOD서버, 실시간 통신 시스템, 병원정보시스템



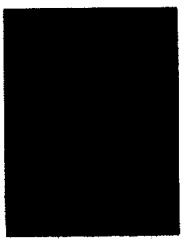
김 경 훈

e-mail : pluit@mdclab.chonnam.ac.kr
1998년 대불대학교 컴퓨터과학과(이학사)
2000년 전남대학교 컴퓨터공학과(공학석사)
2000년~현재 전남대학교 컴퓨터공학과 박사과정
2000년~현재 ㈜포스트림 기술팀장
관심분야 : 병렬VOD서버, 실시간 통신 시스템



류 재 상

e-mail : jsisang@mdclab.chonnam.ac.kr
1997년 전남대학교 컴퓨터공학과(공학사)
1999년 전남대학교 컴퓨터공학과(공학석사)
2000년~현재 전남대학교 전자공학과 박사과정
2000년~현재 ㈜포스트림 기술팀장
관심분야 : 병렬VOD서버, 실시간 통신 시스템, 병원정보시스템



남 지 승

e-mail : jsnam@chonnam.chonnam.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