

# 고성능 클러스터드 스트림 서버 시스템 설계 및 구현

진진한\*, 양현종\*, 남지승\*  
\*전남대학교 전자컴퓨터공학  
e-mail:jhjeon23@naver.com

## Design and Implementation of HA(High-Availability) Clustered Stream Server System

Jin-Han Jeon\*, Hyun-Jong Yang\*, Ji-Seung Nam\*  
\*Dept of Computer Engineering, Chon-Nam National University

### 요 약

스트리밍 서비스를 위한 서버 시스템은 실시간 데이터를 서비스하기 위한 충분한 용량을 가져야 하며 많은 클라이언트들에게 비디오 파일을 전송할 수 있어야 한다. 또한 지속적으로 늘어나는 사용자수와 추가되는 새로운 저장 공간에 대한 우수한 확장성을 제공하여야 한다. 그러나 현재의 범용 스트리밍 서버시스템은 이러한 요구사항을 충분히 반영하지 못할 뿐만 아니라 늘어나는 사용자의 부하와 시스템 요구에 대한 확장성 및 고화질·고용량의 미디어 데이터에 대한 반영이 이루어지지 못하여 점차 증가되는 사용자의 서비스 요구에 대한 사항을 충족시키지 못하고 있다. 본 논문에서는 이러한 사용자의 요구에 대하여 효과적인 QoS와 효율적인 Management 정책을 제시하고 이에 따라 개발된 우수한 병렬 미디어 스트리밍 서버(PMSS)를 소개한다.

### 1. 서론

클러스터 멀티미디어 서버의 구현에는 고속의 네트워크 인프라와 데이터의 분산, 그리고 서버 시스템의 자원의 효율적인 사용 및 사용자의 요구 등 많은 요소들을 고려해야만 한다. 특히 서버는 많은 클라이언트들을 서비스하기 위하여 충분한 자원과 저장장치들을 보유하여야만 한다. 즉, 로컬서비스 환경에서와 같이 사용자에게 좋은 품질의 서비스를 제공할 수 있어야만 한다[1].

실시간 미디어 서버에서는 특별한 하드웨어와 미디어 전송과 관리를 위한 소프트웨어가 필요하고, 이것들은 적절하게 관리하고 사용되어지기 위해서 많은 시간과 노력이 요구되어 진다[2,3].

하드웨어의 경우 많은 사용자들에게 데이터를 액세스하는 빠른 속도를 가지는 저장장치가 중요한 요소이며 충분한 데이터의 전송을 위한 I/O 장치 또한 역시 중요하다. 저장장치의 물리적인 성능뿐만 아니라 고용량의 미디어 데이터의 효율적인 배치 및 관리도구가 필수적이다. 그러나 이것은 매우 임의적인 사용자 서비스 요구임에도 불구하고 예상 가능한 변수가 다양하게 존재하기 때문에 탄력적으로 서버 시스템을 구성할 수가 없다.

이에 우리가 구현한 시스템인 PMSS(Parallel Media Server System)는 기본적으로 여러 대의 클러스터 된 서버가 동시에 하나의 클라이언트에게 미디어 스트리밍 서비스를 제공한다. Grid와 클러스터개념이 적용된 PMSS는 각 서버가 스트리밍 서비스를 위하여 자원을 공유한다[4].

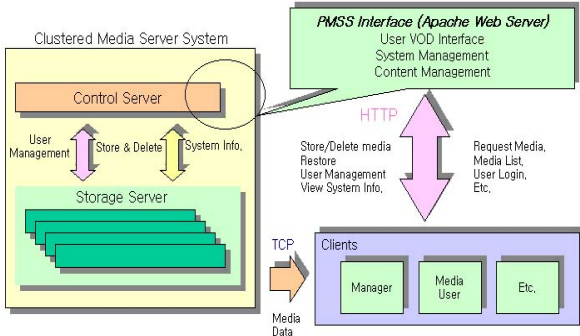
또한 미디어 타입과 관계없이 모든 파일 타입을 지원하고 데이터의 흐름을 각 미디어 특성에 맞게 전송하기 위하여 기본적으로 TCP를 전송 프로토콜로 사용하고 있다[5]. 그리고 일반 클라이언트와 미디어 관리를 위한 데이터베이스 및 하부 프로그램, 일반 클라이언트 관리 등을 위한 관리자에게 보다 사용이 용이하도록 사용자 인터페이스를 웹기반으로 제공하고 있다. 하부 프로그램은 서버의 데이터를 각 저장 서버에 분배하고 미디어의 관리 및 해당 미디어 데이터에 대한 재분배도 담당하며 서버의 모니터링 서비스도 실시한다. 이를 총괄적으로 해주는 시스템이 병렬미디어 스트리밍 서버(PMSS)이다.

### 2. PMSS의 구조 및 서비스 방법

그림1에서 보여지는바와 같이 PMSS는 크게 Control 서버와 Storage 서버로 구성되어 있다. 그리고 이러한 내용은 그림 2에서 보는바와 같이 웹기반으로 각 사용자 및 관리자에게 인터페이스를 제공한다. Control 서버는 Storage 서버로 관리자가 업로드하는 데이터를 분배, 저장, 재활당 등의 프로세스를 실행하고 해당 미디어 데이터에 대한 관리를 수행한다. 또한 여러 대의 Storage 서버가 있을 경우 비디오 파일을 나누어 각각의 Storage 서버에 분배하고 저장하는 역할을 수행한다. Storage 서버는 이렇게 Control 서버에 의해 분배된 미디어 파일을 사용자가 요구할 때에 전송을 담당한다.

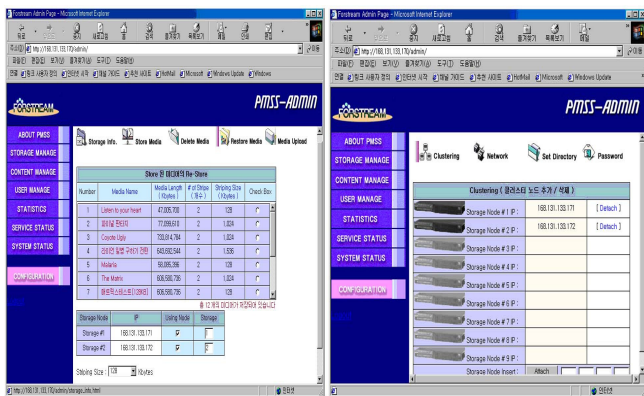
이런 Control 서버와 Storage 서버의 기능적인 분할은

사용자 및 모든 데이터를 관리하는 Control 서버로의 과도한 부하를 피하기 위함이다. 또한 클라이언트의 경우 데이터의 요구를 Control 서버에게 요청을 하지만 실제적인 데이터의 전송은 Storage 서버로부터 다중의 접속경로로부터 전송받게 된다.



(그림 1) PMSS(Parallel Media Streaming Server) 전체 시스템 구조

일반 클라이언트는 먼저 Control 서버의 웹서버를 통하여 미디어 데이터의 리스트를 확인하고 사용자 인증 후 서비스 요청을 한다. Control 서버는 클라이언트가 선택한 데이터의 저장구조, 위치 등을 표시한 메타데이터를 클라이언트에게 전송하고 이 메타데이터를 이용하여 클라이언트는 Storage 서버와 다중 접속 경로를 설정하게 되고 동시에 수신 받은 데이터를 조합하여 미디어 파일을 재생하게 된다. 새로운 Storage 노드가 추가되면 Control 서버는 동시에 사용자 노드를 전체 시스템에 적용하여 분산하게 된다. 또한 새로운 Storage 서버에 장착된 저장 디스크는 기존 저장 데이터의 재분산(re-striping) 저장을 통하여 전송 부하를 Storage 서버 간에 분산 할 수 있으며 극단적으로 전체 데이터의 재분산이나 필요한 미디어 데이터만의 재분산을 통하여 전체시스템의 부하를 감소시킬 수가 있다. 이러한 미디어의 재분산은 웹기반의 모니터링 시스템을 통하여 시스템 상황과 서비스 상황을 모니터 하고 필요할 때 위의 정책에 따라 시스템 환경을 재구성 할 수 있다.

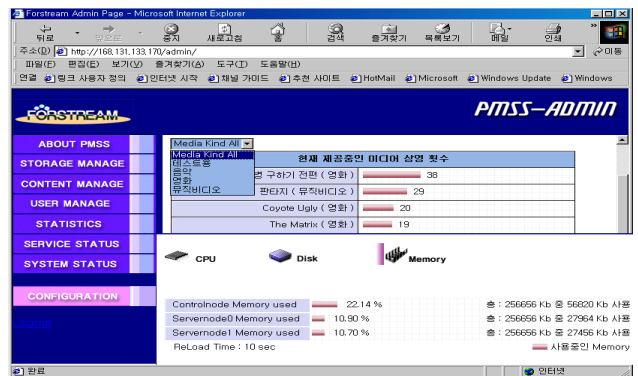


(그림 2) PMSS 관리자의 사용자 인터페이스

그림2는 시스템 확장과 저장 공간 및 미디어 관리를 위한 웹기반의 시스템 인터페이스를 나타낸 것으로 관리자가 데이터의 분산 저장 정책에 의하여 전체시스템을 효율적으로 관리하고 상황변화가 극심한 네트워크 환경을 모니터링 할 수 있다. 이러한 시스템 성능 모니터링은 현재 서버의 사용자 및 저장장치와 네트워크의 부하에 따른 올바른 미디어 서버 시스템의 구성을 유도 할 수 있다. 또한 관리자가 모니터링 한 데이터를 통하여 안정된 서비스 수준의 사용자 수에 따른 시스템을 운용하고 보다 안정된 고품질의 미디어 서비스를 위해서는 적합한 예비율을 산정하여 시스템 폭주에 대비하여야 한다.

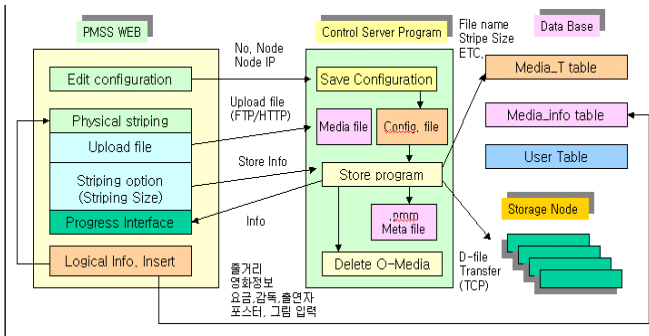
이와 같은 고려로 인하여 시스템은 다소 불편한 관리 매커니즘을 갖는데 이것은 앞에서 설명한 부분에 대한 세세한 배려 때문이라고 할 수 있다. 개발 시스템은 관리자와 클라이언트 모두 웹 인터페이스를 통한 접근을 하는데 관리자는 미디어 데이터를 Control 서버에 업로드 하여야 하고 이를 각 Storage 서버에 Striping 정책에 따라 분산 저장하여야 한다. 이와 같은 이중 데이터 저장은 컨트롤 노드에 중요한 원시 미디어 데이터를 저장 하여 여러 노드에 걸쳐 데이터를 분산 배치하는 데서 생기는 데이터 손실과 에러에 대비하는 백업 및 복구 데이터로 사용할 수 있다. 이 같은 구조 때문에 Control 서버에는 전체 저장 노드의 데이터 용량만큼의 저장 공간이 필요하게 되나 사용자 데이터의 중요성에 따라 이러한 백업 대상 데이터의 보관 여부를 결정할 수 있다. 시스템의 오류에 대비한 백업 공간의 낭비는 어느 시스템에서 발생 하는 문제로서 단일 시스템으로 구성되었다 하더라도 이 같은 디스크 사용은 해결 할 수 없는 문제이다.

RAID 디스크의 물리적 복구 가능한 레벨을 적용하거나 SAN 같은 지능형 저장장치를 채용하더라도 짧은 시간 시스템을 복구하거나 서비스 보장을 위해서는 별도의 백업 데이터가 필요하다. 구현 시스템에도 이러한 저장 장치를 채용하면 같은 효과를 기대 할 수 있으나 많은 비용을 소모하므로 효율적이지 않다. 우리가 개발한 PMSS는 시스템 클러스터를 통하여 RAID의 수준의 성능을 기대 할 수 있고 거기에 시스템 확장은 추가 노드의 증설로 간단히 확장되고 이에 대한 별도의 부하 분배가 필요하지 않다.



(그림 3) PMSS 시스템 자원 상황 모니터링

그림 3는 현재 Storage 서버의 프로세서, 메모리, 디스크 등의 자원 상황을 모니터링 하는 관리자 인터페이스를 나타낸 것이다. 이와 같이 단순한 모니터링뿐만 아니라 여기에 표시되는 데이터는 새로운 사용자를 수용하고 시스템을 관리하는 데 좀 더 유용하게 사용된다. 새로운 사용자 서비스 요청을 Control 서버가 받게 되면 수용제어 프로그램은 현재 각 서버에서 전송되는 시스템 모니터링 정보와 서비스 요구가 발생한 데이터의 타입과 특성을 반영하여 서비스 수용 가능 여부는 물론 한계범위 안에서 사용자 서비스 시작 시점을 결정하게 된다.



(그림 4) PMSS 데이터 저장 구조

그림4는 PMSS에서 클러스터 된 Storage 서버의 데이터 저장 구조를 도식화 한 것으로 미디어 서버를 위해서는 기존 서버에 단지 미디어 데이터를 네트워크 상으로 전달하고 이 데이터를 수신하고 재생하는 프로그램들로 기본적으로 그 역할을 수행한다. 하지만 범용 시스템으로 다수의 동시 클라이언트에게 서비스를 제공하여야 하는 서버를 구성하기에는 좀 더 미디어 서비스 측면의 시스템 구성이 필요하다. 개발 시스템은 이미 기본적인 성능이 검증된 상태로 1차적인 버전이 출시되었으며 급변하는 인터넷 및 미디어환경에 맞도록 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 하부프로그램들로 효율적이고 고성능의 클러스터된 미디어 서버 시스템을 설계 구현 하였다.

### 3. 결론 및 향후 연구과제

안정적인 운영체제는 미디어 서버뿐만 아니라 일반적인 범용 서버에서도 중요한 요소이다. 이러한 안정성을 보장하기 위하여 우리는 PMSS(Parallel Media Server System)을 개발하였다. PMSS의 큰 장점은 Parallel 전송 방식과 분산 저장 정책이다. 이러한 방식의 채용은 시스템 부하를 크게 줄이고 또한 어떤 다른 부하분배기의 필요 없이 쉽게 확장을 할 수 있다는 점이다. 또한 PMSS는 다양한 관리 및 모니터링 툴을 제공하여 미디어 서버의 성

능관리를 효율적으로 수행할 수 있다는 것이다. 그리고 데이터의 분산저장을 통하여 실제적인 데이터의 저장 공간은 늘어나지만 데이터의 복구 및 백업을 위한 별도의 장치를 요구하지 않아 비용면에서도 효과적이라고 할 수 있다.

PMSS는 기존의 싱글 서버와는 다르게 다수의 Storage 서버를 통한 다수의 접속채널로 데이터를 전송하므로 네트워크의 인프라가 크지 않더라도 적절한 QoS를 기본적으로 보장한다. 이처럼 우리의 시스템은 병렬전송, 분산저장을 기본으로 시스템의 확장성에서도 뛰어난 성능을 보였다. 향후 미래의 인터넷은 네트워크 인프라가 커지고 고용량, 고화질의 미디어 데이터서비스가 요구되어지는 시대가 도래 할 것이다. 따라서 이러한 미래 인터넷환경에 맞도록 미디어 서버의 개발은 필요할 것이며 지속적인 연구가 진행되어져 할 것이다.

### 참고문헌

- [1]. Wang, J.a , Liu, J.b , Peng, Y.b , Jin, S.b “ BECP: The scheduling policy based on best-effort in clustered VOD servers” Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS 2(2005), pp. 534-538
- [2] Hefeeda, M., Bhargava, B. "On-demand media streaming over the Internet" Proc. FTDCS'03(2003). San Juan, Puerto Rico, May
- [3] Chen-Lung Chan; Shih-Yu Huang; Hung-Hsuan Chen; Wei-Hao Tung; Jia-Shung Wang, "An application-level multicast framework for large scale VOD services" Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on Volume 1, 20-22 July 2005 Page(s):98 - 104 Vol. 1
- [4] Guo,L.,Chen,S.,Zhang,X. "Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery" IEEE Transactions on Knowledge and Data Engineering 18 (5)(2006), pp. 669-682
- [5] Microsoft, Microsoft Media Service SDK", 1998
- [6] Steven W. Carter, Darrell D. E. Long, and Jehan-François Pâris "An efficient implementation of interactive video-on-demand", Proceedings of Modeling, Analysis and Simulation of Computer and Telecommunication Systems (2000) 172-179
- [7] Ailan Hu. "Video-on-Demand broadcasting protocols: A comprehensive study", In Proceedings of the IEEE Infocom, volume 1, Anchorage, Alaska, USA (2001) 508-517[1] Roger S. Pressman "Software Engineering A Practliners' Approach" 3rd Ed. McGraw Hill