

# 병렬 확장 VoD 서버에 대한 성능 분석

홍석래, 심재홍, 최경희, 김재훈  
아주대학교 컴퓨터공학과 시스템 소프트웨어 실험실

## Simulation of Parallel and Scalable VoD Server

Souck-Rae Hong, Jae-Hong Shim, Kyung-Hee Choi, Jai-Hoon Kim  
Division of Information & Computer Engineering  
Ajou University

### 요 약

최근 컴퓨터와 제반 기술들의 눈부신 발달로 인해 대용량의 멀티미디어 데이터를 다루는 서비스들이 많이 등장하고 있다. 주문형 비디오 서비스(Video on Demand)가 가장 대표적인 예라고 할 수 있으며, 이를 구현하기 위한 많은 방법들이 제안되고 있다. 본 논문에서는 현재까지 개발되어 있는 VoD 시스템에 관하여 살펴보고, 이를 기반으로 PASS(Parallel and Scalable Server) VoD 시스템을 제안한다. PASS VoD 시스템은 병렬 비디오 서버로서 SAN(System Area Network)로 연결된 independent proxy 구조와 비슷한 구조를 가지는 시스템이다. 본 논문에서는 PASS VoD 시스템에 대하여 설명하고 PASS VoD 시스템이 scalable 함을 시뮬레이션을 통하여 확인하였다.

#### 1. 서론

최근 컴퓨터의 눈부신 발달로 인해 단일 미디어를 취급하는 응용에서 멀티미디어를 다루는 응용으로의 전환이 이루어지고 있다. 뿐만 아니라 컴퓨터와 더불어 통신의 급격한 발달로 네트워크를 통하여 대용량, 연속성의 특징을 갖는 주문형 비디오 서비스(Video on Demand : VoD)등의 멀티미디어 서비스가 실시간으로 처리 가능하게 되었다. 이러한 서비스의 발달로 현재 여러 형태의 주문형 비디오 서비스 시스템이 제안되고 구현되고 있다.

주문형 비디오 시스템은 비디오 프락시(proxy)와 고객(client) 시스템으로 구성된다. 비디오 프락시는 비디오 데이터를 제공하며, 고객 시스템은 비디오 프락시에 접속하여 정보를 제공받는다.

주문형 비디오 시스템은 저장 장치의 관리, 시스템의 구조, 사용된 통신망 등의 하드웨어적인 사항들 뿐만 아니라 스케줄링, 스트라이핑, 버퍼링 등의 소프트웨어적인 사항에 의해서도 시스템의 확장성이 결정된다.

이에 본 논문에서는 이러한 확장성의 결정 요소에 따라서 PASS(PARallel and Scalable Server)[1]가 scalable 한지를 분석하기 위해 이를 모델링하고 심스크립트(Simscript II.5)[2]를 사용하여 시뮬레이션하고 성능을 분석하였다.

#### 2. VoD 시스템 관련연구

현재 VoD 시스템의 성능을 향상시키기 위한 연구가 활발히 진행 중이며, 대표적인 연구로 scalability와 고장허용 능력(fault tolerance) 그리고 부하분산(load balancing) 등을 향상시키기 위한 시스템 구조 모델에 대한 연구가 있다. 본 장에서는 가장 일반적인 시스템 구조인 단일 비디오 서버(single video server) 모델의 문제점을 살펴본 후에 이의 해

결 방안으로 제시된 병렬 비디오 서버(parallel video server) 모델의 구조에 대하여 알아본다.

#### 2.1 단일 비디오 서버 모델

단일 비디오 서버(single video model)란 각각의 서버가 독립적으로 사용자에게 영화를 제공하는 방식을 말한다. 단일 비디오 서버 모델은 현재 구성되어 있는 서버의 용량을 초과해서 요구가 들어올 경우 복제(replication)나 분할(partition)의 방법을 사용하여 새로운 서버를 추가할 수 있다. 복제는 서버의 용량을 초과하는 서비스의 요구가 있을 때 새로운 서버를 추가하고 이 서버에 데이터를 전부 복제하는 방식이다. 이것은 저장장치를 두 배로 요구한다. 따라서 데이터 중복으로 저장장치의 낭비가 많다. 또한 서버들이 독립적으로 동작하므로 어느 서버에 서비스가 몰렸을 때, 서버들 사이의 부하 분산에 어려움이 있다. 분할은 새로운 서버를 추가하되 기존의 영화 데이터들을 전부 복제하는 것이 아니라 각각의 서버에 분산시키는 것이다. 이 방법은 데이터의 중복에 의한 저장장치의 낭비는 없지만 어떤 비디오 서버들에만 인기 있는 비디오 타이틀이 많을 경우 부하 분산에 어려움이 있다.

#### 2.2 병렬 비디오 서버 모델

단일 비디오 서버 모델의 문제점들의 해결방안으로 여러 개의 서버에 비디오 데이터를 스트라이핑(striping) 시키는 병렬 비디오 서버(parallel video server)가 연구되고 있다[3]. 병렬 비디오 서버에서 서버 수준의 스트라이핑(striping)은 비디오 데이터의 중복이 필요하지 않고 scalable 하며 비디오의 인기에 상관없이 부하가 분배된다. 병렬 비디오 서버의 가장 큰 핵심은 비디오 데이터가 스트라이핑되어 있다는 것이다. 따라서 각 서버에 스트라이핑되어 있는 데이터

스트림들을 재정렬하고 합해 하나의 데이터 스트림을 만드는 작업이 필요하며, 이를 수행하는 것을 프락시(proxy)라고 부른다. 프락시를 구현하는 방법은 프락시가 시스템 내에서 존재하는 위치에 따라서 3 가지 방법으로 구현된다[3].

2.2.1 Proxy-at-Server

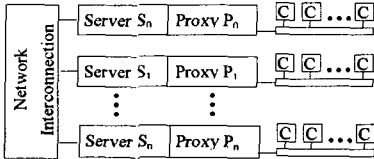


그림 1. Proxy-at-Server 구조[3]

서버들은 네트워크를 통해 지역적으로 연결되어 있다. 프락시는 클라이언트에게 하나의 비디오 데이터 스트림을 보내기 위해 지역 저장장치나 다른 서버들로부터 가져온 데이터를 합하는 역할을 한다. 그러나 이 구조는 프락시와 서버가 분리되지 않아 두 기능을 처리하는데 걸리는 부하와 서버간의 네트워크에 걸리는 부하가 많다는 것이 단점이다.

2.2.2 Proxy-at-Client

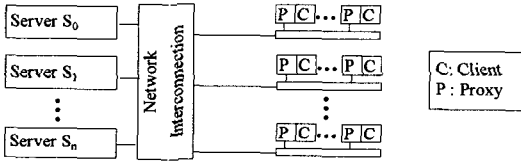


그림 2. Proxy-at-Client 구조[3]

이 구조에서 프락시는 클라이언트와 한 노드 내에 존재하며 서버에게 클라이언트 컴퓨터로 직접 데이터를 보내라고 요청한다. 프락시에서 처리가 끝난 후 비디오 데이터는 더 이상 다른 네트워크를 통하지 않고 클라이언트에게 간다. 그러므로 비디오 데이터를 단 한 번의 네트워크 전송으로 클라이언트에게 제공할 수 있다. Proxy-at-Server 구조, Independent Proxy 구조와 비교해 볼 때 네트워크에 걸리는 부하는 작으나 클라이언트에게 투명하지 않다. 또한 프락시가 서버 쪽에 위치하지 않으므로 프락시에 들어 있는 비디오 데이터를 재사용하여 다른 클라이언트에게 제공할 수 없다.

2.2.3 Independent Proxy

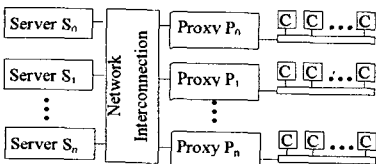


그림 3. Independent Proxy 구조[3]

이 구조는 프락시들을 별도의 컴퓨터에 두어 서버들과 네트워크를 통해 지역적으로 연결되어있고 다수의 클라이언트들과는 다른 외부 네트워크를 통해 연결된다. 이 구조에서 데이터는 먼저 서버의 저장장치에서 읽혀 프락시로 전달되고, 프락시는 여러 서버로부터 데이터들을 받아 처리하고 클라이언트에게 데이터를 전달한다. Independent Proxy 구조는 Proxy-at-Server 구조보다 네트워크 부하가 많이 걸린다는 단점이 있다. 그러나 투명성(많은 서버들 사이의 복잡한 관계를 감출 수 있다는 것)을 가지며 프락시의 데이터를 캐쉬(cache)하여 다른 클라이언트의 서비스를 위해 재사용할 수 있다는 장점을 가진다.

3. PASS(Parallel and Scalable Server)

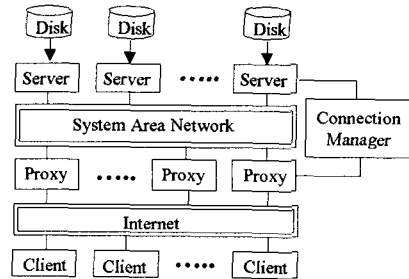


그림 4. PASS 시스템의 구조

PASS는 병렬 비디오 서버로서 SAN(System Area Network)로 연결된 independent proxy 구조와 비슷한 구조이다. PASS 시스템의 구성요소로는 사용자에게 영화를 서비스해주는 프락시(proxy), 프락시에게 SAN(System Area Network)을 통해 영화를 공급해주는 서버(server), 사용자와 영화를 서비스해 줄 프락시를 연결하는 CM(Central Manager)으로 구성된다.

프락시는 요청한 비디오 데이터를 각 서버로부터 서비스 받는다. 프락시에서는 영화 데이터에 대한 순서제어 및 서비스된 영화 데이터에 대한 캐쉬 역할을 하기 위하여 버퍼링을 하게 된다. 프락시에서의 버퍼링은 전체 통신의 부하를 줄이는 중요한 역할을 하는 부분이라고 할 수 있다.

서버는 프락시로부터 데이터 요구가 들어올 때마다 디스크에서 영화 데이터를 읽어서 프락시에게 전달한다. 영화 데이터는 각각의 디스크에 스트라이핑 된다.

CM은 프락시의 사용자에게 대한 서비스 가능 여부를 결정하고, 선택된 비디오를 서비스하기 위하여 가장 적절한 프락시와 연결해준다.

PASS는 서버와 프락시를 초고속 통신망을 이용한 SAN(System Area Network)으로 연결하여 데이터 전송부하를 현격히 줄일 수 있다. 시스템 전반적인 관점에서 보았을 때, SAN은 시스템 버스와 같은 역할을 한다고 볼수 있으며, 병렬 처리 시스템에서 분산된 자원을 효과적으로 활용하기 위한 필수요건이라고 할 수 있다.

4. PASS 시뮬레이션

4.1 전체구조

프락시는 데몬(Daemon), 샌더(Sender), 리퀘스터

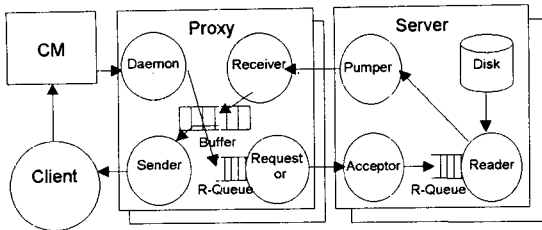


그림 5. PASS 시뮬레이션 모델

(Requestor), 리시버(Receiver) 프로세스와 프락시 버퍼, 서버스 요청을 처리하기 위한 큐 등으로 구성된다. 데몬 프로세스는 각 프락시당 하나 씩만 존재하며 클라이언트로부터 온 영화 서비스 요청을 취합하여 리퀘스터에게 보내는 역할을 한다. 센더 프로세스는 각 클라이언트마다 하나 씩 생성되며, 해당 프락시의 버퍼로부터 클라이언트에게 보낼 영화데이터를 찾아서 클라이언트에게 전송하는 역할을 한다. 영화 데이터에 대한 순서 제어도 센더 프로세서에서 이루어진다. 리퀘스터 프로세스는 데몬프로세스로부터 영화 서비스 요청을 받아서 영화 데이터가 스트라이핑 되어 있는 서버에게 영화 데이터를 요청하는 역할을 한다. 리시버는 서버로부터 영화 데이터를 받아서 프락시 버퍼에 버퍼링시키는 역할을 하며, 이와 동시에 버퍼를 관리하는 역할도 한다.

서버는 액셉터(Acceptor), 리더(Reader), 펌퍼(Pumper)의 프로세스와 영화 데이터를 스트라이핑하고 있는 디스크로 구성된다. 액셉터 프로세스는 프락시로부터 요청받은 영화데이터를 취합하는 역할을 하며, 필요한 영화 데이터를 리더에게 요청하는 역할을 한다. 리더는 액셉터로부터 요청받은 영화 데이터를 디스크로부터 읽어서 펌퍼에게 공급하는 역할을 한다. 펌퍼는 리더로부터 받은 영화 데이터를 서비스해야 할 프락시에게 보내주는 역할을 한다. CM(Connection Manager) 프로세스는 전체 시스템에 단 하나만 존재하며, 모든 클라이언트로부터 영화에 대한 요청을 받은 후, 가장 적절한 프락시에게 영화를 요청하는 역할을 한다. 클라이언트(Client) 프로세스는 일정한 분포에 의거하여 서비스 받을 영화를 선택하며, 이를 CM에게 요청하는 역할을 한다.

#### 4.2 시뮬레이션 파라메타

이 실험에서 사용된 파라메타들을 다음과 같다. 영화 한 편은 1000 개의 페이지로 구성되며, 각 페이지의 크기는 4KByte 이다. 프락시 내에 있는 버퍼의 크기는 4MByte 이며 영화 데이터 1000 개를 버퍼링할 수가 있다. 각 클라이언트와 CM, 센더와 클라이언트 사이에는 INET-10Mbps 의 성능을 지닌 일반적인 이더넷을 사용한다고 가정-가 존재하며, 영화 데이터 한 페이지를 전송하기 위한 네트워크 지연시간은 400 $\mu$  sec 가 걸린다. 프락시와 서버, CM 과 프락시 사이에는 SAN(System Area Network)-645Mbps 의 성능을 지닌 Myrinet[4] 카드를 사용한다고 가정-가 존재하며, 영화 데이터 한 페이지를 전송하기 위한 네트워크 지연시간은 62 $\mu$  sec 가 걸린다. 서버에서 클러스터링 되어 있는 영화 데이터 한 페이지를 디스크로부터 읽어 오는데 걸리는 시간은 하드디스크의 속도(20Mbyte/sec)와 SCSI 카드의 속도(3.3Mbyte/sec)를 고려했을 때, 1212 $\mu$  sec 가 걸린다.

#### 4.3 실험 결과

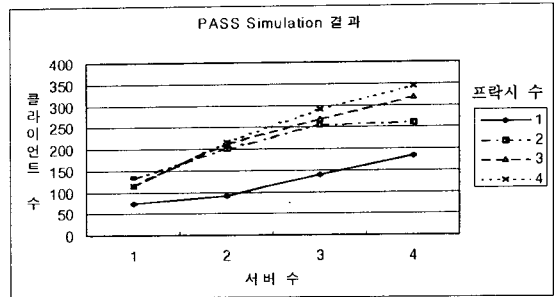


그림 6. PASS 시뮬레이션 결과

그림 6 은 PASS VoD system 모델에 대한 시뮬레이션 결과이다. 이 결과는 모든 클라이언트들이 동시에 서비스를 요청했다고 가정했을 때, 몇 개의 클라이언트가 정상적으로 서비스를 받을 수 있는지를 실험한 결과이다. PASS VoD system 이 calable 함을 알아보기 위해서 프락시와 서버의 개수를 늘려가며 최대 허용가능한 클라이언트의 수를 측정하였다.

시뮬레이션 결과를 통해서 다음과 같은 사실들을 알 수 있다. 첫째, 프락시의 수가 고정되어 있고 서버의 수만 증가될 때에는 동시에 서비스할 수 있는 클라이언트의 개수가 명확하게 scalable 하게 늘어난다. 둘째, 서버의 수가 고정되어 있고 프락시의 개수만 증가될 때에는 동시에 서비스할 수 있는 클라이언트의 개수가 늘어나기는 하지만 첫 번째의 경우와 비교하면 그 증가폭이 크지 않음을 알 수 있다.

#### 5. 결론 및 향후 연구 과제

본 연구 결과 프락시나 서버의 개수에 대하여 scalable 한 결과를 보이는 VoD 시스템을 구현할 수 있었다. 시스템이 각각의 자원이 추가됨에 따라 그 성능이 scalable 하게 늘어나기 때문에 시스템의 확장이 용이하다.

그러나 좀 더 객관적인 평가를 위해서는 이번 실험을 통하여 나온 결과와 동일조건 하에서 다른 모델들을 실험한 결과를 비교,평가할 필요가 있다. 또한 PASS 모델의 성능 향상을 위하여 좀 더 효율적인 데이터 스트라이핑 기법, 연속 저장 매체에 대한 버퍼 관리 기법[5]의 연구가 필요하다.

#### 참고문헌

- [1] 최경희 외 8 인, "멀티미디어 데이터의 효율적인 스트라이핑 저장 및 검색에 관한 연구," 중간연구 보고서, 전자부품종합기술 연구소, 1998.
- [2] Edwark C. Russell, "Building Simulation Models with SIMSCRIPT II.5," CACI Products Company, 1989.
- [3] Jack Y.B. Lee, "Parallel Video Servers: A Tutorial," IEEE Multimedia Vol. 5, No. 2, April/June 1998.
- [4] Myricom, "Myrinet User's Guide", <http://www.myri.com/scs/documentation/mug>.
- [5] B. Ozden, R. Rastogi, and A. Silberschatz, "Fellini - a file system for continuous media," Technical Report 113880-941028-30, AT&T Bell Laboratories, Murry Hill, 1994.