

# 병렬 VOD 시스템에서 서비스 모델의 성능분석

남정임\*, 남지승\*\*  
전남대학교 공과대학 컴퓨터정보통신공학과

## Performance Analysis of Service Model in Parallel VOD system

JeongYim Nam\*, JiSeung Nam\*\*  
Computer Information Communication Engineering Department  
Chonnam National University  
E-mail : \*jynam@jnu.ac.kr, \*\*jsnam@jnu.ac.kr

### Abstract

Media service model is divided into 3 models that are Client Pull, Server Push, and IPP(Interleaving Pull & Push) model. In most single VOD(Video On Demand) environment, Client Pull model was sufficient to play the movie Because most media contents has a low bitrate and resolution. But according to an increment of the demand of the high definition media, Client Pull model is not sufficient. Parallel VOD environment is made of several of VOD servers and provides the parallel media stream simultaneously for one client. We compared and analyzed the performance of service models with respect to network delay and data size in buffer in the single and parallel VOD environment and we found that IPP service model keeps the least network delay and stable client buffer state in the parallel VOD environment.

### I. 서론

현재의 인터넷 환경은 다양한 멀티미디어 데이터의 활용이 증가하면서 새로운 도약의 시기를 맞이하고 있다. 이러한 환경의 변화는 주문형 비디오, 원격 화상 회의 등의 상업용 서비스 보편화로 그 영향이 나타나고 있으며, 특히 인터넷상에서 스트리밍 기술을 기반으로 한 주문형 비디오 서비스가 증가하고 있다[1,2].

이러한 서비스 증가 추세는 보다 많은 클라이언트

지원을 보장하기 위한 서버성능의 향상과 효율적인 서비스 모델을 도출하게 되었으며, 또한 기존의 콘텐츠에 비해 월등히 높아진 해상도와 비트레이트는 다수의 서버를 이용한 병렬형 VOD 시스템과 이에 알맞은 서버/클라이언트간 미디어 서비스 모델의 개발로 이어졌다. [3,4,5].

주문형 비디오 서비스에 있어서 데이터 전송을 위한 서버/클라이언트간의 서비스 모델은 크게 3 가지로 나누어진다. 접속한 클라이언트에게 서버가 일방적으로 데이터를 전달해주는 방식 즉 주도권을 서버가 가지는 Server Push 서비스 모델과 클라이언트가 일정 크기의 데이터 블록을 요청하고 서버는 요청 받은 데이터 블록만을 전달하여 주는 방식 즉 데이터 흐름 제어의 주도권을 클라이언트가 가지는 Client Pull 서비스 모델, 그리고 이 둘을 병합한 IPP 방식으로 분할되어 발전하고 있다[6,7,8].

본 논문에서는 단일 VOD 환경에서의 서버/클라이언트 간 데이터 서비스 모델을 분석하고 이를 병렬 VOD 시스템에 적용하여, 보다 나은 서비스 모델의 제시와 사용자 QoS의 향상을 제공하고자 한다.

## II. 병렬 VOD 시스템

### 2.1 병렬 VOD 시스템 구조

병렬 VOD 시스템의 하드웨어는 다수의 미디어 저장 장치를 위한 충분한 용량과 큰 전송 속도를 보장하는 저장 장치, 미디어에 대한 네트워크로의 충분한 데이터 전송 능력을 보장하는 입출력 장치가 가장 중요한 요소를 차지하며, 소프트웨어는 미디어 데이터의 특성에 구애 받지 않는 효율적인 데이터 배치와 미디어 관리를 위한 사용자 인터페이스가 주문형 비디오 솔루션의 주요한 요소로 포함된다[1,2].

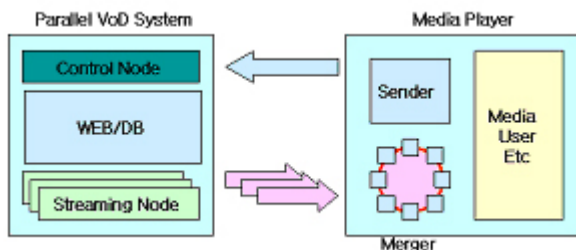


그림 1 병렬 VOD 시스템 구조

구현된 시스템은 기본적으로 Linux 운영체제를 기반으로 한 여러 대의 서버가 동시에 하나의 클라이언트를 위해 다중의 접속 경로를 갖는 형태로 구성된다. 서버 시스템은 하위 저장 서버의 제어흐름을 조정하고 관리하기 위한 컨트롤 노드와 미디어를 저장하고 사용자에게 서비스하는 스트리밍 노드군으로 구성된다.

컨트롤 노드는 미디어를 스트리밍 노드군으로 스트라이핑하고 미디어 콘텐츠를 관리 재배치하는 미디어 관리 모듈, 사용자 접속요청과 서비스 상태 정보를 관리하는 클라이언트 관리 모듈, 컨트롤 서버와 저장 서버의 부하측정을 위한 시스템 정보 모듈로 구성되고 이를 위해 웹 서버와 데이터베이스 서버 기능을 갖춘다.

스트리밍 노드에는 컨트롤 노드의 제어 하에 클라이언트가 요구하는 데이터를 전송하는 프로그램이 설치되며 실제적인 데이터 전송을 담당한다. 이러한 기능적 분배는 서로 다른 타입의 데이터 전송 예를 들어 인터페이스용 웹 데이터, 데이터 베이스 트랜잭션 데이터 등을 전송 서버로부터 분리시키기 위한 목적으로 존재한다.

클라이언트에는 다중 접속 경로 즉, 다수의 서버로부터 전송되어지는 미디어 스트림을 병합하여 플레이어

에 공급하는 역할을 수행한다[6,9].

### 2.2 서비스 모델의 종류

병렬 주문형 비디오 시스템에서 서버와 클라이언트 간에 데이터를 전달하는 서비스 모델로는 Server Push 모델과 Client Pull 모델 그리고 두 모델의 장점만을 취한 IPP 모델로 크게 나뉜다[3,4,5].

Server Push 서비스 방식은 단일서버 주문형 비디오 시스템에서 일반적으로 연구되었으며 데이터 흐름의 주도권을 서버가 가지게 되며 클라이언트로부터의 특정 요청이 필요 없이 주기적으로 적절한 데이터를 클라이언트로 전송하는 방식으로, 브로드캐스트 서비스에 적합하나 클라이언트로부터의 역 채널이 없는 관계로 데이터를 공급하는 서버에서 데이터 전달 속도를 적절하게 조절하여야만 클라이언트에서의 버퍼 오버플로우 및 언더플로우를 방지할 수 있다는 단점이 있다[6,7,8].

Client Push 서비스 방식은 클라이언트가 데이터 흐름의 주도권을 가지고 서버에 원하는 데이터를 요청하면 서버는 요청된 데이터를 클라이언트로 전달하는 요청/응답 형식을 가지는 일종의 Polling 방식으로서, 일반적인 운영체제의 파일 시스템 I/O 가 여기에 해당하며 클라이언트에서의 버퍼 언더플로우 및 오버플로우를 클라이언트가 제어 할 수 있다. 이처럼 Server Push 방식에서의 클라이언트에 비해 보다 능동적인 역할을 수행한다는 장점을 가지는 반면 유니캐스트 서비스에 보다 적합하고, 클라이언트는 서버로 요청 메시지를 전송할 추가 채널을 가지고 있어야 하며, 서버는 요청을 처리하기 위해 연속적으로 인터럽트 되어야하고 다수의 클라이언트로 인한 확장성 병목 현상이 쉽게 만들어진다는 단점을 가진다. WMT(Window Media Technology) 분야에서 주로 사용되는 방식이다[6,9].

IPP 서비스 방식은 Server Push 방식과 Client Pull 방식이 결합된 서비스 방식으로 Client Pull 방식과 동일하게 데이터 흐름의 주도권을 클라이언트가 가지며, 단일 요청에 대한 다중 응답의 서비스 형태를 가진다. 클라이언트는 역 채널을 통하여 원하는 데이터 블록을 요청하게 되고 이 요청 메시지에는 요청하는 블록의 번호와 블록의 수가 포함되며 이를 통하여 서버는 요청 받은 번호의 블록부터 지정된 블록의 수만큼 해당 블록 데이터를 클라이언트에게 전송하는 서비스 방식이다. 이를 통하여 데이터 전달 속도조절을 통한 클라이언트 버

퍼의 능동적 제어가 가능하고, 과도한 메시지 전송을 줄임으로써 미디어 데이터의 네트워크 전달 지연 문제를 해결할 수 있다[6].

### III. 병렬 VOD 시스템에서의 서비스 모델 성능분석

단일 서버 환경에서의 IPP 모델은 클라이언트가 역채널을 통하여 원하는 데이터 블록을 요청하고 이 요청 메시지에는 요청하는 블록의 번호와 블록의 수가 포함되며 이를 통하여 서버는 요청 받은 번호의 블록부터 지정된 블록의 수만큼 해당 블록 데이터들을 클라이언트에게 전송하는 서비스한다.

병렬 서버 환경에서의 IPP 모델은 네트워크 지연과 서버 처리시간만을 고려한 작업시간의 총량에 있어서의 각 서비스 모델간의 비교 분석한다. 단일 서버의 경우 세 가지 모델 모두를 비교하였으나 Server Push 과 IPP 방식이 거의 비슷한 지연시간을 소모하고 Server Push 모델이 대부분 브로드캐스트 서비스에만 국한적이기에 병렬 환경에서는 Client Pull 모델과 IPP 모델을 비교 분석하였다.

클라이언트가 다수의 서버에게 요청 메시지를 전달하는 시간차는 실제적으로 네트워크 전달 지연 변이에 대부분 흡수된다. 또한 서버의 수를 S 라 할 때, 각각의 서버는 동일한 작업 처리 능력을 가진다. 즉, 클라이언트가 요청하는 S 개의 메시지는 동시에 출발하며, 각각의 서버는 동일한 작업 처리 시간값을 가진다.

병렬 서버 환경에서의 Client Pull 서비스 모델과 IPP 방식을 적용할 경우의 데이터 전달 지연시간은 단일 서버 환경의 소요시간을 서비스에 참여하는 스트리밍 서버의 수로 나눈 값이 된다. 즉 병렬 서버 환경에서의 Client Pull 서비스 모델과 IPP 방식을 적용할 경우의 데이터 전달 지연시간은 서버의 수만큼 지연시간이 감소하게 되는 것이다.

단, 본 실험에서는 클라이언트 측에서 데이터를 소비하는 데 걸리는 시간은 포함되지 않았다. 즉 각각의 서버로부터 전송되어온 데이터 블록을 클라이언트 버퍼로 수신될 때까지의 시간만을 고려하였으며 클라이언트가 데이터를 정렬하고 다시 디코딩 모듈로 넘겨 비디오 데이터가 play 되도록 하는데 걸리는 시간은 제외하고 계산되었다.

### IV. 성능평가

인터넷의 경우 여러 시스템들이 연결되어 있고, 서비스의 다양화에 따라 각 네트워크 세그먼트들의 전송 데이터 유형이나 전송용량 및 패킷 손실 등의 트래픽 특성이 매우 가변적으로 나타난다. 여기에서 사용된 네트워크 전송간 평균지연은 다음과 같은 측정 시나리오를 통해 계산되었다.

본 시험의 측정 시나리오는 가장 네트워크 부하가 심한 것으로 조사된 오후 2시부터 4시 사이의 시간에 측정되었다. 실험에 사용될 데이터는 MPEG-2 HD 데이터 파일로서 20Mbitrate 즉, 약 2.5Mbyte/sec 의 전송률을 가지는 미디어 파일이다. 위 미디어 데이터는 16Kbyte 크기를 스트라이핑 단위로 나누어지며 병렬 서버 환경의 경우 서버의 수(S)만큼 분산되어 저장된다.

다음은 병렬 서버 환경에서 Client Pull 모델을 이용하였을 경우 시간에 따른 클라이언트 버퍼의 데이터 적재량 변화도이다. 서버의 처리시간을 100msec, 서버의 스트라이핑 블록 크기(B)는 16kbyte 일 때 서버의 수를 3, 4, 5, 10 으로 증가시키면서 측정된 그래프로 버퍼 언더플로우나 오버플로우가 발생할 경우 버퍼 적재량의 측정을 중지하게 설정되었다.

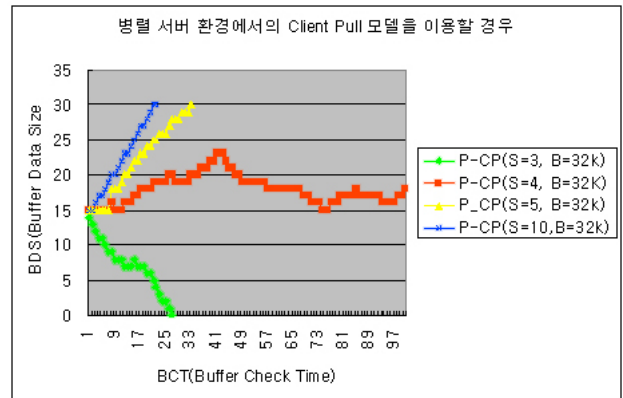


그림 2 병렬 VOD 환경에서의 Client Pull 모델

Client Pull 모델을 이용할 경우 서버의 수에 의해 버퍼 데이터 적재량의 변화가 매우 심한 것을 확인 할 수 있었다. 이러한 현상은 앞에서 언급된 것처럼 데이터 전달 지연을 통해 발생하는 현상으로 서버의 수가 3 일 경우 버퍼 데이터 적재량의 급격한 감소로 인해 버퍼 언더플로우가 발생하게 되고, 서버의 수가 4 인 경우 버퍼 내 데이터 적재량이 큰 범위의 오실레이션을

일으킴으로써 버퍼의 안정성을 침해하게 되며 결과적으로 서비스의 질 하락을 유발할 수 있다. 서버의 수가 5 이상일 경우에만 버퍼 데이터 적재량 증가로 인해 버퍼의 안정화 상태가 유지된다. 이와 같이 서버의 수에 민감하게 반응하는 클라이언트 측 버퍼 상황은 추후 클라이언트 버퍼 설계와 서버 시스템의 확장성에 문제를 발생시키는 원인이 될 수 있다.

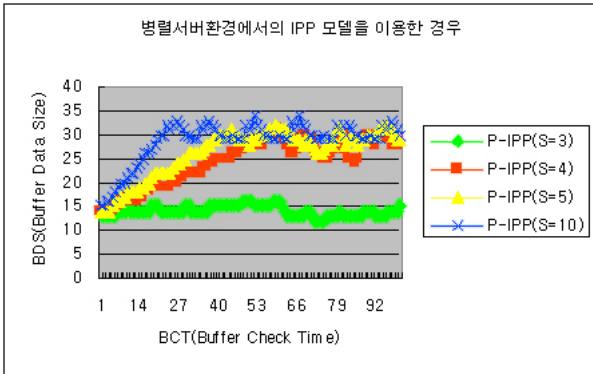


그림 3 병렬 VOD 환경에서의 IPP 모델

그림 3 은 병렬 서버 환경에서 IPP 모델을 이용하는 경우의 그래프로 스트라이핑 블록의 크기가 16KByte 이며 실험을 위해 실제 버퍼의 크기를 기존 버퍼보다 20% 크게 설정하였다. 서버의 수가 4 이상인 경우 충분한 버퍼 데이터 적재량을 유지하게 되며, 또한 서버의 수가 3 인 경우에도 Client Pull 모델에 비해 보다 작은 오실레이션을 일으키며 이를 통하여 클라이언트 버퍼 안정화와 사용자 QoS 를 보장할 수 있음을 확인 할 수 있다.

## V. 결론

본 논문은 VOD 환경에서의 서비스 모델을 분석하고 적용함으로써 보다 나은 사용자 QoS 를 제공하고자 한다. 이를 위해 병렬 VOD 환경에서는 Client Pull 모델과 IPP 모델을 전달지연과 클라이언트 버퍼 내 데이터 잔여량 측면에서 비교 및 분석하였다.

실험을 통해 병렬 VOD 환경에서 IPP 서비스 모델이 가장 적은 전달지연과 보다 안정적인 클라이언트 버퍼를 유지함을 알 수 있으며 이를 통해 사용자에게 보다 나은 서비스를 제공할 수 있음을 검증하였다. 추후 클라이언트 버퍼로 들어오는 데이터의 병합 시간에 대한 연구가 진행되어야 한다.

## 참고문헌

- [1] D. J. Gemmell, H. M. Vin, D. D. Kandlur, P. V. Rangan, and L. A. Rowe, "Multimedia storage servers: A Tutorial," IEEE Multimedia Mag. Vol.27, pp.40-49, May, 1995.
- [2] D.P. Wu, Y.W.T. Hou, and W.W. Zhu, "Streaming Video over the Internet : Approaches and Directions," IEEE Transactions on Circuits & Systems for Video Technology, Vol.11, No.3, 2001.
- [3] Y. B. Lee, "Parallel video servers--A Tutorial," IEEE Multimedia Mag. Vol.5, No. 2, pp.20-28, 1998.
- [4] 김서균, 김경훈, 류재상, 남지승, "리눅스 기반의 고성능 병렬 미디어 스트림 서버 설계 및 구현," 정보처리학회 논문지 Vol.8-A, No.4, pp.0287-0292, 2001.
- [5] 이민홍, 김경훈, 남지승, "PMSS 시스템에서 서버/클라이언트 간 서비스 모델의 성능분석," 정보처리학회 논문지 Vol.11-A, No.03, pp.0329~0334, 2004.
- [6] J.P.Martin-Flatin, "Push vs. Pull in Web-based Network Management," In Proc. of the Integrated Network Management VI, pp.3-18, May, 1999.
- [7] T.Kiao, "Global Information Broadcast: an architecture for Internet push channels," IEEE Internet Computing, VOL.04, NO.04, PP.0016-0025, July, 2000
- [8] J.Y.B. Lee, "Staggered push - a linearly scalable architecture for push-based parallel video servers," IEEE Trans on Multimedia, VOL.04, NO.04, pp.0423-0433, December, 2002.
- [9] Microsoft, "Microsoft Media Service SDK," 2005.