제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

병렬 VOD 환경에서의 서비스 모델의 성능분석

이민홍*, 김경훈*, 남지승*
*전남대학교 컴퓨터공학과
e-mail : leemh@mdclab.chonnam.ac.kr

Performance Analysis of Service Model on Parallel VOD Environment

Min-Hong Lee*, Kyung-Hun Kim*, Ji-Seung Nam*
*Dept. of Computer Engineering, Chonnam National University

요 약

본 논문은 단일 VOD 환경에서의 서버/클라이언트 간 데이터 서비스 모델을 분석하고 이를 병렬 VOD 환경에 적용함으로써 보다 나은 사용자 QoS(Quality of Service)를 제공하고자 한다.

이를 위해 단일 VOD 환경에서는 클라이언트가 데이터를 요청하고 서버가 전달해주는 Client Pull 모델과 서버 측에서 일방적으로 데이터를 전달해주는 Server Push 모델을 네트워크상 전달지연 측면에서 분석하고, 위 두 모델을 통합한 IPP(Interleaving Pull & Push) 서비스 모델과 Client Pull 모델을 전달지연과 클라이언트 버퍼 내 데이터 잔여량 측면에서 병렬 VOD 환경에서 적용 및 비교하였다.

시뮬레이션을 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 있다.

1. 서론

현재의 인터넷 환경은 다양한 멀티미디어 데이터의 활용이 증가하면서 새로운 도약의 시기를 맞이하고 있다. 이러한 환경의 변화는 서비스 매체의 변화를 이끄는 계기가 되었으며, 텍스트 기반의 단일 매체 서비스에서부터 인터넷을 통한 비디오, 오디오, 텍스트 등과 같은 다양한 매체들을 하나로 종합한 멀티미디어 응용 서비스 형태로 발전하였고 이를 위한 연구가 활발히 진행되고 있다.

이러한 연구를 기반으로 주문형 비디오 (VOD : Video On Demand), 원격 화상 회의, 홈쇼핑 등의 상업용 서비스들이 현재 보편화 되어지고 있으며, 특히 인터넷상에서 스트리밍 기술을 기반으로 한 주문형 비디오 서비스가 증가하고 있다[1].

현재의 주문형 비디오 서비스에 있어서 데이터 전송을 위한 서버/클라이언트간의 서비스 모델로는 접속한 클라이언트에게 서버가 일방적으로 데이터를 전달해주는 방식 즉 주도권을 서버가 가지는 Server Push[2] 서비스 모델과 클라이언트가 일정 크기의 데이터를 요청하고 서버가 요청 받은 데이터를 전달하여 주는 방식 즉 데이터 흐름 제어의 주도권을 클라이언트가 가지는 Client Pull[3][4] 서비스 모델로 양분되어 발전되고 있다.

본 논문에서는 단일 서버 VOD 환경에서의 서버/클라이언트 간 데이터 서비스 모델을 분석하고 이를 통해 보다 나은 서비스 모델의 제시와 다수의 서버를 이용하는 병렬 VOD 환경에 적용함으로써 사용자 QoS (Quality of Service)의 향상을 제공하고자 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 VOD 시스템 구조를 소개하고, 3 장에서는 서비스 모델의 종류를 소개하고 각 모델에 대한 성능을 분석한다. 4 장에서 시뮬레이션을 통한 산출 결과를 그래프로 도시한다. 5 장에서 결론을 내리고 향후의 연구과제에 대해 기술한다.

2. VOD 시스템 구조

기존의 범용 서버를 VOD 솔루션으로 이용하기 위해서는 하드웨어와 미디어 전송 및 관리 SW 의 두 가지의 구성 요소가 필요하다. H/W 는 많은 용량과 빠른 전송 속도를 갖는 저장 장치와 충분한 데이터 전송을 위한 I/O 장치가 가장 중요한 요소를 차지하며, 데이터의 특성에 구애 받지 않는 효율적인 데이터 배치 및 SW 를 통한 능률적인 미디어 관리가 VOD 솔루션의 주요 한 요소로 포함된다.

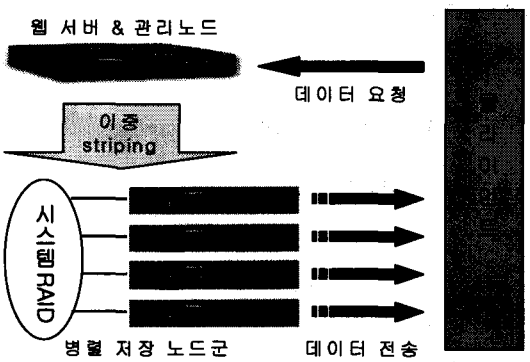

그림 2-1 VOD 시스템 구조

그림 2-1 은 전체적인 시스템 구조를 나타낸 것이다. 구현된 시스템은 기본적으로 여러 대의 서버가 동시에 하나의 클라이언트를 위해 다중의 접속 경로를 갖는 형태로 구성되었다. 미디어 특성에 맞는 데이터 흐름을 자율적으로 조정하고, 여러 서버의 제어흐름을 조정하고 관리하기 위한 제어 서버 노드가 별도로 존재한다. 이는 다른 타입의 데이터 전송 예를 들어 인터페이스용 웹 데이터, 데이터 베이스 트랜잭션 데이터 등을 미디어 전송 서버로부터 분리시키기 위한 목적으로 존재한다. 또한 사용자 인터페이스로서의 웹 서버와 미디어 관리용 데이터 베이스, 각 저장 서버에 데이터를 분산 저장하고 컨텐츠를 관리 재배치하는 관리자용 부 프로그램들로 전체 시스템이 구성된다.

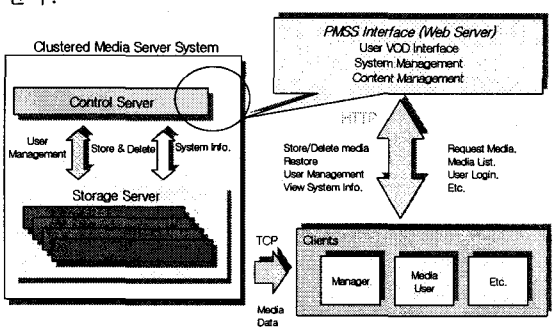

그림 2-2. 시스템 인터페이스 및 서비스 방식

저장 서버에는 클라이언트가 요구하는 데이터를 전송하는 프로그램이 설치된다. 이러한 기능적 분배는 다른 특성을 갖는 즉, 미디어 데이터와 다른 제어 메시지 데이터 전송으로 인하여 디스크와 I/O 장치를 소모하는 overhead 감소를 위해 고려되었다. 클라이언트에는 다중 접속 경로를 갖는 데이터 수신 모듈이 탑재되고 이는 기존 플레이어에 수신된 데이터를 공급하는 역할을 수행한다. 전체적인 서비스 방식은 그림 2-2 에 나타내었다.

3. 서비스 모델의 종류 및 성능분석

Server Push 서비스 방식은 단일서버 VOD 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 버퍼 언더플로우를 방지할 수 있다는 단점이 있다.

Client Pull 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 오버플로우 및 버퍼 언더플로우를 클라이언트가 제어 할 수 있다. 이처럼 Server Push 방식에서의 클라이언트에 비해 보다 active 한 역할을 수행한다 라는 장점을 가지는 반면 유니캐스트 서비스에 보다 적합하고, 클라이언트는 서버로 요청 메시지를 전송할 추가 채널을 가지고 있어야 하며, 서버는 요청을 처리하기 위해 연속적으로 인터럽트 되어야 하고 다수의 클라이언트로 인한 확장성 병목 현상이 쉽게 만들어진다 라는 단점을 가진다. 현재 윈도우 WMT(Window Media Technology) 분야에서 주로 사용 되어지는 방식이다.

Interleaving Push & Pull 은 Server Push 방식과 Client Pull 방식이 결합된 서비스 방식으로 클라이언트는 역 채널을 통하여 원하는 데이터 블록을 요청하는데 이 요청 메시지에는 요청하는 블록의 번호와 블록의 수가 포함되며 이를 통하여 서버는 요청 받은 번호의 블록부터 지정된 블록의 수만큼 해당 블록 데이터들을 클라이언트에게 전송하는 서비스 방식이다.

위의 세 가지 서비스 모델의 전체적인 서비스 흐름은 다음과 같다. 먼저 버퍼 상태를 파악하고 비디오 상영에 필요한 데이터 요청 메시지를 네트워크를 통해 서버에 전달하는 단계인 데이터 요청, 서비스 스케줄러를 통해 요청 받은 데이터를 디스크로부터 전달받아 해당 클라이언트에게 전송하는 단계인 서버 스케줄링, 서버로부터 전송되기 시작한 데이터가 네트워크를 통해 클라이언트로 전달되는 단계인 데이터 응답, 그리고 클라이언트 버퍼로 들어와 쌓인 데이터를 소비하는 단계인 데이터 소비 순으로 전개된다.

3.1 단일 서버 환경에서의 서비스 모델의 성능 분석

단일 서버 환경에서 세 가지의 서비스 모델 즉 Server Push 방식, Client Pull 방식, IPP 방식을 비교하여 보고 각 방식의 장단점을 분석한다. 단, 여기에서의 비교 분석 대상은 네트워크 지연과 서버 처리 시간만을 고려한 작업시간의 총량으로 한정한다.

단일서버 환경에서 클라이언트에서 메시지가 출발하는 시간을 0 으로 정하고, 클라이언트가 수신할 블록의 수를 B, 한 블록의 크기는 Q(byte)이다. 그리고 클라이언트가 데이터를 수신한 이후 새로운 메시지를 생성하고 출발시키는 데까지 걸리는 시간은 무시한다.

3.1.1 Client Pull 방식

이 방식은 클라이언트에서 서버에게 데이터 요청 시 하나의 블록씩만을 요구하며, 데이터 요청, 서버 처리, 데이터 수신의 과정을 통해 서비스가 이루어진다. i 번째 블록을 요청한 후 해당 블록을 수신하는 데 소요되는 시간을 Ti, 클라이언트에서 서버까지 블록 요청 메시지를 전달하는 데이터 요청 단계에서 소비되는 시간을 Treq(i), 서버에서의 메시지를 처리하는 서버 스케줄링 단계의 소비 시간을 Tsch(i), 서버에서 클라이언트까지의 블록 데이터가 전달되는 데이터 응답 단계의 소비 시간을 Tres(i) 이라고 한다. 여기에서 Ti 는 데이터 소비단계의 처리 시간은 제외된다.

i 개의 블록 데이터를 전달되는 데 소비되는 총 시간 Tclientpull 은 다음과 같다.

$$Tclientpull=B*(Treq+Tsch+Tres) \quad .....(식4.1)$$

3.1.2 Server Push 방식

이 방식은 클라이언트에서 서버로의 역 채널은 존재하지 않으며, 처음 서비스를 위한 한 번의 서비스 요청, 서버 처리, 데이터 수신의 과정을 거치며 서비스가 이루어진다.

i 개의 블록 데이터를 전달되는 데 소비되는 총시간 Tserverpush 은 다음과 같다.

$$Tserverpush=Treq+B*Tsch+Tres \quad ..... (식4.2)$$

3.1.3 IPP(Interleaving Pull & Push)

클라이언트는 역 채널을 통하여 원하는 데이터 블록을 요청하고 이 요청 메시지에는 요청하는 블록의 번호와 블록의 수가 포함되며 이를 통하여 서버는 요청받은 번호의 블록부터 지정된 블록의 수만큼 해당 블록 데이터들을 클라이언트에게 전송하는 서비스 방식이다.

$$TIPP = N*Treq +(B+N)*SR+N*Tres \quad .....(식 4.3)$$

3.2 병렬 서버 환경에서의 서비스 모델의 성능 분석

다음은 병렬 VOD 서버 환경에서 네트워크 지연과 서버 처리시간만을 고려한 작업시간의 총량에 있어서의 각 서비스 모델간의 비교분석이다. 단일 서버의 경우 세 가지 모델 모두를 비교하였으나 Server Push 과 IPP 방식이 거의 비슷한 지연시간을 소모하고 Server Push 방식이 대부분 브로드캐스트 방송에만 국한적이라는 사실을 통해 병렬 환경에서는 Client Pull 방식과 P-IPP 방식만을 비교 분석한다.

클라이언트가 다수의 서버에게 요청 메시지를 전달하는 시간차는 실제적으로 네트워크 전달 지연에 흡수된다. 또한 서버 각각의 성능을 동일하게 가정함으로써 서버의 작업 처리 시간은 동일하고 서버에서의 데이터 출발시간 또한 동일하다. 비교 분석을 위해 다음과 같이 가정한다.

1. 클라이언트가 요청하는 S 개의 메시지는 동시에 출발한다.
2. 서버의 작업 처리 시간은 동일하다.

위와 같은 가정을 통해 병렬 서버 환경에서의 Client Pull 서비스 모델과 IPP 방식을 적용할 경우의 데이터 전달 지연시간 TP-CP, TP-IPP 은 다음과 같다.

$$TP-CP = Tclientpull/S \quad ..... (식 4.4)$$
$$TP-IPP = Tinterleaving / S \quad ..... (식 4.5)$$

결국,

$$TP-CP=(B*(Treq+SR+Tres))/S \quad ..... (식 4.6)$$
$$TP-IPP = (N*Treq+(B+N)*SR+N*Tres)/S \quad ..(식 4.7)$$

이 된다.

즉 병렬 서버 환경에서의 Client Pull 서비스 모델과 IPP 방식을 적용할 경우의 데이터 전달 지연시간 TP-CP, TP-IPP 은 서버의 수(S)만큼 지연시간이 감소하게 되는 것이다.

단 위의 식에서는 병렬 서버 시스템 구조에서 고려되어야 하는 클라이언트 데이터 소비 단계의 시간이 포함되지 않았다. 즉 각각의 서버로부터 전송되어온 데이터 블록을 클라이언트가 정렬하고 다시 디코더 모듈로 넘겨 비디오 데이터가 PLAY 되도록 하는데 걸리는 시간은 제외하고 계산된 결과이다.

4. 실험 결과

실험에 사용될 데이터는 MPEG-2 데이터 파일로서 2.5Mbitrate 즉 약 300Kbyte 의 전송률을 가지는 미디어 파일로써 전체 길이 820Mbyte, 상영길이 약 40 여분의 특성을 갖는 데이터이다. 2.5 Mbit/sec 의 bitrate 를 가지는 위 미디어 데이터는 300kbyte 의 스트라이핑 사이즈 단위로 하여 약 2700 여 개의 블록으로 나누어지며 병렬 서버 환경의 경우 서버의 수

(S) 만큼 분산되어 저장된다. 즉 S=5 인 경우 하나의 서버 당 540 개의 스트라이핑 블록으로 나누어져 저장된다.

그림 5-1 은 병렬 서버 환경에서 각 서비스 모델간 데이터 전달 소요시간들을 비교한 그래프이다.

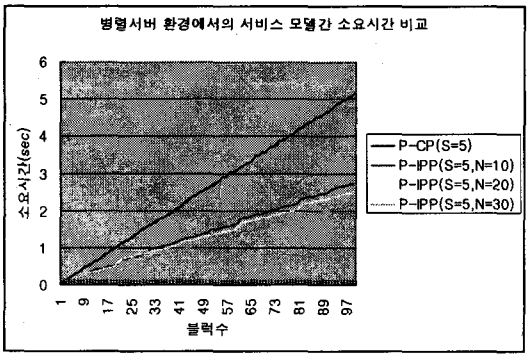

그림 5-1 병렬서버 환경에서의 서비스 모델간 소요시간 비교

전송 시간의 측면에서 Server Push 방식은 네트워크 지연에 큰 영향을 받지 않으므로 수식과 거의 흡사하게 송 수신되는 블록의 수가 증가하며 본 실험에서는 비교대상에 포함시키지 않는다.

Interleaving Pull & Push 방식은 한번에 전송되는 블록의 수를 N=10 으로 고정하여 측정한 것으로 처음 10 개의 블록에 대한 소요시간은 Server Push 방식과 동일하지만 블록 요청 회수가 증가함에 따라 소요시간도 증가함을 확인할 수 있다. 또한 N 의 수가 증가할수록 전체적인 소요시간은 줄어들어 점차적으로 Server Push 방식의 소요시간과 비슷해지게 되며 N=10, 20, 30 으로 조절한 위의 경우 소요 시간 감소를 확인할 수 있다.

반면 Client Pull 방식은 Interleaving Pull & Push 방식과 Server Push 방식에 비하여 훨씬 많은 시간을 소모하게 되며 이로 인해 클라이언트 측에 많은 양의 메모리 요구와 빈번한 서비스 끊김 현상을 초래한다.

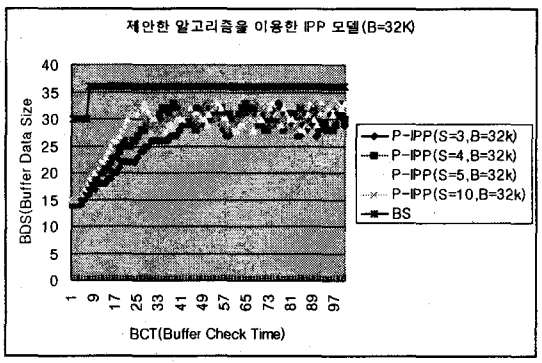

그림 5-4 병렬 서버 환경에서의 IPP 모델(B=32K)

그림 5-2 는 병렬 서버 환경에서 IPP 모델을 이용하는 경우의 그래프이다. 일정시간 이상 버퍼 안정화 상태가 유지될 경우 여분의 버퍼가 추가되었으며 서버의 수에 관계없이 추가된 버퍼가 오버플로우와 네트워크 지연을 위해 활용되는 것을 확인 할 수 있다.

5. 결론 및 향후 연구 과제

본 논문은 단일 VOD 환경에서의 서버/클라이언트 간 데이터 서비스 모델을 분석하고 이를 병렬 VOD 환경에 적용함으로써 보다 나은 사용자 QoS(Quality of Service)를 제공하고자 한다.

이를 위해 단일 VOD 환경에서는 클라이언트가 데이터를 요청하고 서버가 전달해주는 Client Pull 모델과 서버 측에서 일방적으로 데이터를 전달해주는 Server Push 모델을 네트워크상 전달지연 측면에서 분석하고, 위 두 모델을 통합한 IPP(Interleaving Pull & Push) 서비스 모델과 Client Pull 모델을 전달지연과 클라이언트 버퍼 내 데이터 잔여량 측면에서 병렬 VOD 환경에서 적용 및 비교하였다.
시뮬레이션을 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.

참고문헌

[1] Y. B. Lee, "Parallel video servers--A Tutorial," IEEE Multimedia Mag. vol. 5, no. 2, pp.20-28, 1998

[2] 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.

[3] S. Rao, H.Vin, and A. Tarafdar, "Comparative Evaluation of Server-push and Client-Push Architectures for Multimedia Servers," In Proc. of the 6th International Workshop on Network and Operation System Support for Digital Audio and Video, April, 1996

[4] 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.