2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

멀티미디어 데이터 전송에 기반한 웹 서버 설계

남상준, 김태윤
고려대학교 컴퓨터학과
e-mail:sjnam@netlab.korea.ac.kr

The Design of WWW Server based on Multimedia Data Transfer

Sang-Jun Nam, Tai-Yun Kim
Dept. of Computer Science and Engineering, Korea University

요 약

정보 통신의 발달로 인터넷 사용자가 급속도로 증가하고 초고속 정보통신망을 비롯한 정보 인프라의 구축이 확대되면서 다양한 종류의 정보 서비스가 요구되고 있다. 이러한 서비스 중 가장 널리 사용되고 있는 것이 인터넷을 통한 웹(WWW) 서비스이다. 본 논문에서는 웹 서버에 기반한 멀티미디어 서비스에 대해 빠른 전송을 수행하기 위해 멀티미디어 데이터 전송을 커널 내부에서 이루어지개 웹 서버 시스템을 디자인하고 제안한다.

1. 서론

정보 통신의 발달로 인터넷 사용자가 급속도로 증가하고 초고속 정보통신망을 비롯한 정보 인프라의 구축이 확대되면서 다양한 종류의 정보 서비스가 요구되고 있다. 이러한 서비스 중 가장 널리 사용되고 있는 것이 인터넷을 통한 웹(WWW:World Wide Web) 서비스이다.

몇 년전부터 웹을 통한 다양한 정보의 공유는 이루어지고 있으며 최근 들어 웹을 통해 멀티미디어 데이터 요구 또한 급격히 증가하고 있다. 이에 따라 웹 서버에 접속을 시도하는 트래픽은 예전에 비해 늘어나고 있는 실정이다.

그러나 이러한 요구를 충족시키는 웹 서버 시스템의 발전은 미비한 상태이다. 앞으로의 웹 서버 시스템은 지금까지의 한정적이고 일반적인 범위에서 벗어나 멀티미디어 데이터의 실시간 처리 능력과 대용량의 데이터를 효율적으로 전송하는 새로운 웹 서버 시스템이 요구되어 진다.

멀티미디어 데이터와 같은 서비스에서 서버 측면에서 성능 향상을 높일 수 있는 요인은 문맥 교환과 데이터 복사를 줄이는 것이다. 이런 서버 시스템은 저장 장치와 네트워크 하부시스템 사이에서 주기적으로 데이터를 주고받는다. 기대되는 작업에 대해 웹 서버 시스템은 높은 성능으로 디자인되고 전송되어야 한다. 그러나, 일반적으로 존재하는 웹 서버 시스템은 이러한 성능개선을 보장하거나, 저장장치와 네트워크 하부시스템 사이의 효과적인 경로를 제어하지는 않는다. 대다수의 애플리케이션들은 사용자 모드와 커널 모드에서 과도한 데이터 복사가 발생한다. 멀티미디어 스트림 서비스와 같이 한번 접속된 서비스가 다른 변경 없이 연속적으로 전송된다면, 데이터 복사의 횟수를 줄이고 빠른 데이터 경로와 전송으로 서비스를 할 수 있을 것이다.

본 논문에서는 웹 서버에 기반한 멀티미디어 서비스에 대해 빠른 전송을 수행하기 위해 멀티미디어 데이터 전송을 커널 내부에서 이루어지게 서버 시스템을 디자인하고 제안한다.

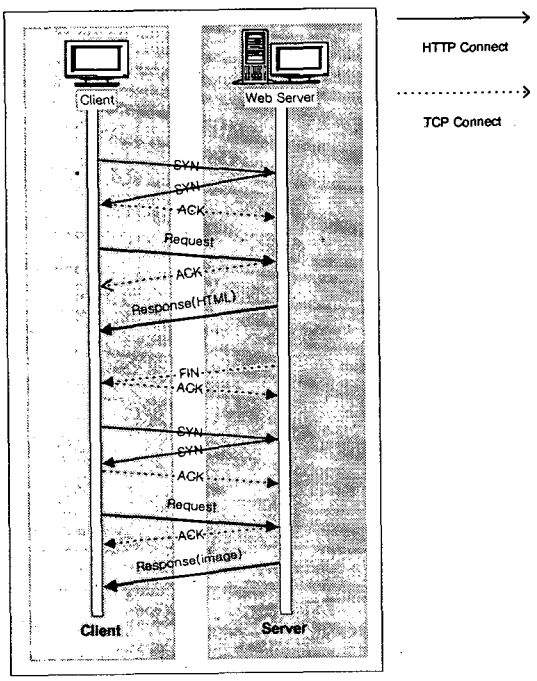
본 논문의 구성은 2장에서 관련 연구를 분석하고, 3장에서 본 논문에서 제안하는 웹 서버 시스템을 소개한다. 마지막 4장에서는 본 논문의 결과 및 향후 과제에 대해 기술한다.

2. 관련 연구

2.1 웹 서버의 연결 절차

웹 서버는 HTTP 프로토콜로써 클라이언트의 요청을 관리한다.

<그림 1>은 현재 HTTP의 연결 요청과 수락에 대해 나타낸다[1].



<그림 1> 서버와 클라이언트의 연결 과정

<그림 1>의 간단한 절차를 살펴보면 클라이언트가 먼저 웹 서버에게 서비스를 요청한다. 이 메시지를 받은 웹 서버는 서버의 상태를 보고 비어있는 포트가 남아있으면 성공 메시지를 보낸다. 이로써 TCP 연결 확립이 되고, 다음부터 서버의 하드디스크에서 문서와 데이터를 전해 주게 되는 것이다.

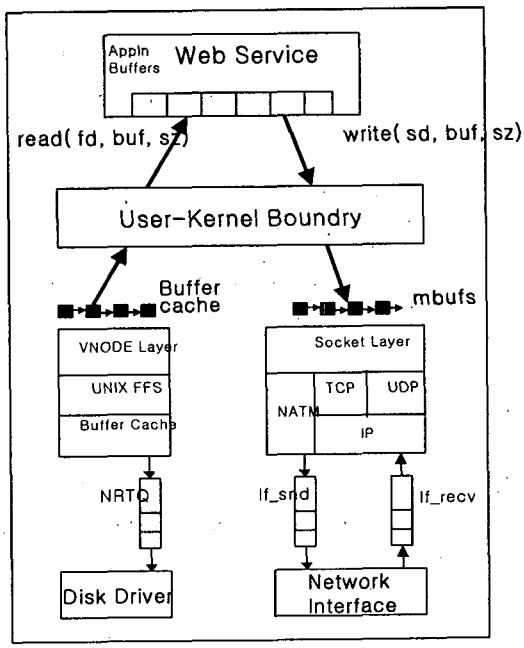
문서와 같이 일반적인 작은 크기의 입출력을 행하는 전통적인 텍스트와 바이너리 파일은 전송이 잘 이루어진다. 그러나 멀티미디어 데이터(오디오, 비디오, 애니메이션 등)와 같은 캐싱 속성을 가진 것들은 기대하는 것만큼의 효과를 보지 못한다.

서버의 전송 경로의 메커니즘은 지금처럼 많은 양의 데이터의 이동을 고려하지 않고 설계가 되었기 때문에 멀티미디어 데이터 전송시 낭비적인 작업이 발생한다.

2.2 일반적인 서버 전송 메커니즘

<그림 2>는 현재 UNIX 운영체제 시스템에서의 네트워크 I/O 시스템의 계층구조를 보여주고 있는, 애플리케이션에서 데이터를 요구하고 그 데이터가 네트워크 인터페이스로 전송되는 일반적인 서버의 메커니즘을 나타낸 그림이다.

위의 <그림 1>에서 클라이언트의 접속을 받고 서버가 요청을 수락한 다음 문서나 다른 멀티미디어 데이터를 요청할 경우의 서버가 데이터를 가져오는 경로는 보여주는 그림이 아래의 <그림 2>와 같다.



<그림 2> 일반적인 서버의 전송 메커니즘

<그림 2>의 서버는 사용자 공간에서 분명히 네트워크 장치까지는 두 단계의 데이터 복사가 일어난다.

첫 번째 복사는 read 함수이다. 이것은 커널의 버퍼 캐쉬에서 사용자 버퍼 공간까지 데이터를 옮기는 경우이다. 두 번째 복사는 write 함수이다. 이것은 소켓 계층에서 발생하는 복사로 사용자 공간 버퍼에서 커널에 있는 mbuf chain까지의 전송을 말한다. 이 접근은 일반적인 작은 크기의 입출력을 행하는 전통적인 텍스트와 바이너리 파일은 전송이 잘 이루어진다. 그러나 멀티미디어 데이터(오디오, 비디오, 애니메이션 등)와 같은 캐싱 속성을 가진 것들은 기대하는 것만큼의 효과를 보지 못한다.

이러한 서비스는 메모리 공간을 많이 사용한다. 또한, 데이터는 종종 이것이 재 사용될 수 있음에도 불구하고 대체될 수 있다. 성능상의 불이익이 잉여 복사로 인해 이루어지게 되는 것이다.

데이터 복사를 최소화하는 연구는 이전부터 시작되었다. 보호되는 영역을 만들어 물리적 데이터 전송을 최소화하는 fbufs[2], 수신측의 커널 영역에 식별자를 두어 네트워크 장치와 애플리케이션의 버퍼 사이의 경로를 줄이는 Splice[3], 성능 향상을 위해 프로그래밍 언어 수준의 제어와 데이터 구조를 명확히 하는 SPIN[4], mmbuf(multimedia mbuf)를 사용하여 새로운 데이터 구조를 커널 버퍼로 생성해 멀티미디어 데이터 전송을 지원하는 MARS[5] 메커니즘 등이 있다.

그러나 이러한 메커니즘들은 공통적으로 다음과 같은 단점을 가진다.

- 커널내 새로운 구조체 생성으로 과도한 메모리 할당이 이루어진다.
- 디스크 장치와 네트워크 장치 사이에 새로운 구조체의 생성으로 인한 다른 장치와의 호환성 결여 문제가 발생한다.

본 논문에서는 이러한 현상들을 줄이기 위해 새로운 웹 서버 서비스에 대한 전송 메커니즘을 제안한다. 제한된 메커니즘으로 웹 서비스를 할 경우 멀티미디어 데이터를 요청하는 사용자들의 요구를 효과적으로 충족시켜 줄 수 있는 모델이다.

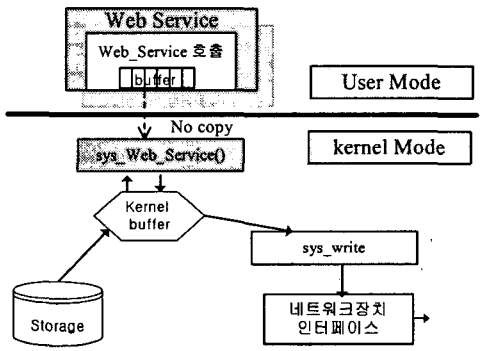
3. 제안된 웹 서버 전송 메커니즘 모델

기존의 웹 서버의 경우 클라이언트의 접속 요청을 받고 멀티미디어 데이터를 전송 할 경우 과도한 문맥 교환과 데이터 복사 현상이 발생한다. 이에 대해 <그림 3>과 같은 새로운 웹 서버에 대한 전송 구조를 제안한다.

다음에 전송할 데이터가 무엇인지 아는 멀티미디어 데이터인 경우 일반적인 웹 서버 시스템에서는 반복적으로 사용자 모드와 커널 모드를 넘나들면서 데이터를 이동시키며 불필요한 데이터 복사와 문맥 교환으로 인한 오버헤드가 발생하게 된다.

제안된 웹 서버 전송 메커니즘의 흐름은 다음과 같다. 일단 HTTP 프로토콜에 의해 클라이언트와의 접속이 연결되었다. 클라이언트가 멀티미디어 데이터를 요구할 경우 일반적인 웹 서버에서는 <그림 2>와 같은 반복적인 작업으로 멀티미디어 데이터를 서비스한다.

그러나 제안된 웹 서버 모델에서는 클라이언트의 멀티미디어 데이터의 요청을 받을 경우 새로운 웹 서비스 시스템 콜로 제어가 넘어간다. 이 시스템 콜에서 패킷들을 커널 영역에서 처리를 하게 된다. 이렇게 처리된 패킷을 커널 내부에서 네트워크 장치 인터페이스로 넘겨준다.



<그림 3> 제안된 웹 서버 전송 모델

<그림 3>과 같이 제안된 웹 서버 모델이 주는 장점은 다음과 같다.

- 입출력의 개선

웹 서버로 들어온 클라이언트의 멀티미디어 데이터의 요청을 새로운 시스템 콜 한번의 호출로 인해 멀티미디어 데이터 전송을 커널 내부로 옮겨 이에 대한 불필요한 데이터 복사와 문맥 교환을 줄였다.

- 기존의 커널 메커니즘 수용

기존의 웹 서버 메커니즘의 전송 경로를 제어함으로써 다른 장치와의 호환성도 유지하게 되었다.

본 논문은 웹상에서 다양한 사용자의 요구에 의해 웹 서버에 대한 멀티미디어 데이터 요구가 늘어남에 따라 기존의 웹 서버의 전송 메커니즘을 수정하여 이러한 요구에 충족하는 새로운 전송 메커니즘을 제안하였다.

제안된 메커니즘은 웹 서비스시 멀티미디어 데이터 요청을 새로운 시스템 콜을 커널 내부에 둠으로써 적어도 2번 이상 발생하는 데이터 복사가 1번으로 줄어들었고, 문맥 교환도 커널 내부에서 발생하므로 그 수행속도가 웹 서비스 애플리케이션보다 빠

르게 일어난다. 이로써 늘어나는 인터넷 확산과 광대한 멀티미디어 데이터를 요구하는 사용자들에게 빠른 전송을 할 수 있다.

4. 결론 및 향후 연구 과제

본 논문에서는 웹 서버 환경하에서 멀티미디어 데이터 전송시 기존의 메커니즘을 커널 내부에 내장함으로써 기존의 서버 환경보다 전송 속도가 향상 될 수 있는 모델을 제안하였다. 기존의 운영체제는 멀티미디어 데이터 전송시 발생하는 문맥 교환과 데이터 복사로 인해 발생하는 오버헤드를 새로운 시스템 콜을 만들어 커널 영역으로 전송 구조를 변경함으로써 성능상의 이익을 보고자 하였다[6].

웹 서버 시스템에서의 성능 향상을 위해 실시간 전송을 보장해야 하는 멀티미디어 전용 웹 서버의 연구가 활발히 진행중이다.

향후 연구 과제로는 제안된 웹 서버 모델을 운영체제에 맞게 변형하여 기존의 웹 서버 시스템과의 정량적인 성능 평가를 수행하는 것이다.

<참고 문헌>

[1] Holtman, K. and Andrew Mutz, "Transparent Content Negotiation in HTTP", Internet Draft, IETF HTTP WG, 1996.

[2] P. Druschel and L. L. Peterson. "Fbufs: A highbandwidth cross-domain transfer facility", In Proceedings of the Fourteenth ACM Symposium on Operating System Principles, pp. 189-202, Dec. 1993.

[3] Kevin Fall and Joseph Pasquale, "Improving Continuous-Media Playback Performance with In-kernel Data Paths", Proceedings of the International Conference on Multimedia Computing and Systems, May 14-19, Boston, Massachusetts. IEEE-CS, pp. 100-109, 1994.

[4] B. N. Bershad, S. Savage, P. Pardyak, E. G. Sirer, M. E. Fiuczynski, D. Becker, C. Chambers and S. Eggers, "Extensibility safety and performance in the SPIN operating system", Proceedings of the fifteenth ACM symposium on Operating systems principles, pp. 267-283, December 1995.

[5] Buddhikot, M. and Chen, X. "Project MARS: WWW based Scalable, Interactive Multimedia Recording and Playback Services", Winner of the first prize at the Research Poster Competition, ACM SIGCSE/SAC'97, San Jose, CA, Feb 27 - Mar 1, 1997.

[6] Buddhikot, M. and Chen, X. "Project MARS: WWW based Scalable, Interactive Multimedia Recording and Playback Services", Winner of the first prize at the Research Poster Competition, ACM SIGCSE/SAC'97, San Jose, CA, Feb 27 - Mar 1, 1997.