

멀티미디어 서버 버퍼의 효율적인 사용에 의한 시스템의 성능개선

강신각*, 이대식, 이재영
한림대학교 컴퓨터공학부

An Improvement of System Performance by Usage of Efficient Buffer in Multimedia Server

S.G.Kang, D.S.Lee, J.Y.Lee
Hallym Univ.Department of Computer Engineering

요약

멀티미디어 데이터를 제공하는 서버 시스템에 동시에 여러 사용자가 서비스 요청을 해 왔을 때 데이터의 특성을 고려해서 적당한 지연시간을 줌으로써 시스템의 버퍼를 효율적으로 사용하게 하여 시스템의 성능을 향상시킨다. 다양한 데이터를 제공함에 있어서 시간에 따른 버퍼의 사용량의 변화를 보면 멀티미디어 데이터가 아닌 일반 데이터의 경우와 다르게 비 효율적인 면을 볼 수 있다. 이 비효율적인 면을 시간과 버퍼의 크기를 서로 교환 함으로써 버퍼의 사용량을 27%정도 줄이는 모델을 제시한다.

1. 서론

최근 들어 네트워크가 확장면서 클라이언트/서버의 구조를 통한 서비스가 활발하게 이루어지고 있다. 특히 시스템의 발달로 인해서 많은 양의 데이터를 가지는 멀티미디어 데이터 역시 클라이언트/서버를 통해서 사용자에게 서비스를 지원하고 있고, 앞으로 계속 활성화 될 것으로 예상된다.

멀티미디어 서버는 대용량의 고속 디스크 저장장치, 상당한 처리 성능, 많은 응용프로그램을 동시에 수행하는 능력을 제공할 수 있어야 한다. 이를 위해서 입출력 속도 면에서 가장 빠른 메모리의 효율이 중요시 된다[1]. 각 호스트의 메모리는 서버와 클라이언트 사이에서 데이터의 완충 역할을 담당하는데, 이 버퍼는 서버와 클라이언트 양쪽에 존재할 수 있으며, 어느 한쪽에만 존재 할 수 있다. 양쪽에 두는 경우, 서버쪽의 버퍼는 서버의 전송율과 클라이언트의 소비율 차를 보정하며, 클라이언트의 버퍼는 네트워크로 인한 전송률의 오차를 보정한다[2].

멀티미디어 데이터의 경우 시간에 매우 중요한 의미를 두기 때문에 적당한 시간에 데이터를 사용자에게 전송하는 것이 무엇보다 중요하다[3, 4]. 이를 위해서 기존에 있는 버퍼 대체 알고리즘을 연속적인 데이터에 적용하는 것은 적당하지 않다. 특히 여러 사용자가 동시에 서비스 요청을 해 왔을 때, 기존의 버퍼 운용 방법은 많은 양의 버퍼를 사용하게 되어 시스템의 성능 저하를 초래하게 되는데, 이 경우에 적용할 수 있는 버퍼 운용방법과 사용자 운용을 통해 버퍼의 사용량의 줄여서 멀티미디어 서버 시스템의 성능을 개선하는 방법을 제안 한다.

2. 멀티미디어 클라이언트/서버 시스템

2.1 시스템 모델

본 논문에서 사용하는 시스템 모델은 다음 그림1과 같이 서버와 클라이언트 양쪽에 버퍼를 사용하는 경우이다. 멀티미디어 서버는 디스크를 저장 기반으로 하고, 네트워크를 통해서 클라이언트의 서비스 요청이 들어오면 디스크에 있는 데이터를 클라이언트에게 제공한다.

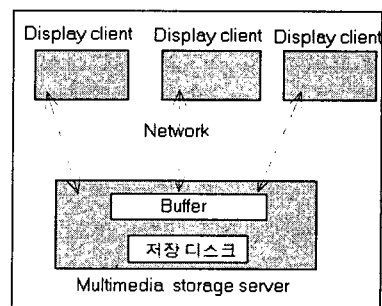


그림 1 시스템 모델

2.2 멀티미디어 데이터의 버퍼

멀티미디어 데이터가 일반 데이터와 다른점은 시간에 따른 연속적인 성격을 가진다. 이 특성은 사용자가 수신한 데이터를 재 수신하지 않는다면 수신한 데이터는 필요가 없다. 다음 그림 2에서 서버에서의 시간에 따른 버퍼 사용량을 보여 주고 있

다. DR은 데이터의 Display Rate이다[5].

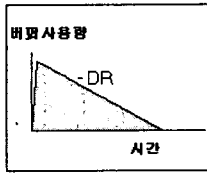


그림 2 멀티미디어 데이터의 버퍼 사용

버퍼 사용량이 시간이 흐름에 따라서 -DR의 기울기로 감소하는 것을 볼 수 있다. 그림은 사용가능한 버퍼의 크기가 충분한 경우로, 한번에 디스크에서 버퍼로 옮겨서 서비스 할 수 있지만 충분한 버퍼를 사용할 수 없는 경우에는 위의 그림과 같은 모습이 데이터 서비스가 끝날때까지 반복적으로 되풀이 된다 [6]. 서버의 입장에서 보면 송신한 데이터가 차지하고 있던 버퍼 공간을 다른 데이터가 사용할 수 있게 조정할 수 있다.

3. 지연시간 적용에 의한 버퍼 사용량

동시에 여러 가지의 데이터를 서비스 요청 했을 때 서버의 프로세서가 하나 인 경우 정확히 동시에 처리 할 수는 없지만, 각 데이터의 최대 버퍼 사용량이 거의 겹치게 된다. 이를 방지하기 위해서 적당한 지연시간을 각 사용자에 적용함으로써 전송된 데이터가 차지 하고 있던 공간을 다른 데이터가 사용할 수 있게 한다.

3.1 지연시간

지연시간은 그림2와 같은 모습이 데이터 전송이 완료될 때 까지 반복 되는 주기 함수로 생각하여 fourier transform을 적용하여 얻는다[7, 8].

버퍼의 크기를 결정하는 요소는 디스크에서 버퍼로 가져 오는 시간과 DR에 비례한다. 데이터 loading 시간은 디스크에서의 데이터 분산과 입도에 따라서 달라 지므로 이론적으로 계산하기는 어려우므로 실제 적용시에 걸리는 시간에 따라서 버퍼 크기를 할당하는 방법을 적용하고, 지연시간을 구하는데는 DR만을 생각한다.

주기가 2π 인 주기 함수 $p_i(t - \tau_i)$ 에 대해서 풀고, 지연 시간에 영향을 주지않는 상수를 모두 제거하면 다음식이 남는다.

$$|\sum_{i=1}^N DR_i \times e^{-j\tau_i}|^2$$

N 은 데이터 갯수이고, τ_i 은 i번째 데이터에 적용될 지연 시간이다. 지연시간을 찾는것은 위의 식을 최소로 만드는 τ_i 를 구하여 적용하는 것이다.

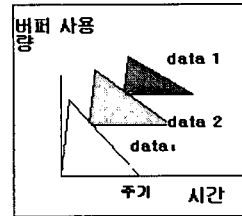


그림 3 지연 시간 적용에 의한 버퍼

3.2 지연 시간의 적용

지연 시간의 적용 범위는 버퍼에 올려진 데이터가 모두 전송될 때까지의 시간(1.round)이다. 지연 시간의 적용으로 각 데이터의 최대 버퍼 사용량 시간을 겹치지 않게 하기 위해서는 서버에서 지원하는 모든 데이터의 주기가 같아야 한다. 만약 조금이라도 다르면 서비스 시간이 긴 데이터들을 제공할 경우 주기가 오랜 시간 나타나기 때문에 겹치게 된다.

버퍼의 크기는 데이터의 DR과 디스크에서 입/출력 시간에 비례하므로 주기의 변화를 가져 올 수 있는 것은, 특히 디스크에 비연속으로 저장된 경우, 입/출력 시간의 변화에 의한 것이다. 따라서 데이터의 디스크에서 입/출력 시간을 계속 검사할 필요가 있고, 시간의 변화에 따라서 버퍼의 크기를 조정할 필요가 있다.

3.3 버퍼 모델

지연시간의 적용으로 전송된 데이터가 차지하고 있던 버퍼를 다른 데이터가 사용하기 위해서는 버퍼에서 계속적인 위치 이동을 해야 한다. 다음 그림은 시간 t1일 때 버퍼의 모습과 t2일 때의 모습을 보여 주고 있다. data1-2는 data1의 두 번째 페이지를 나타낸다.

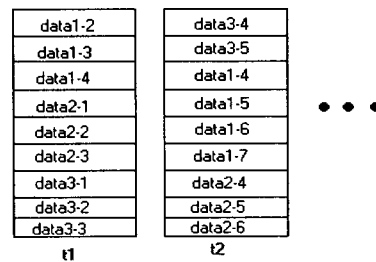


그림 4 버퍼 모델

3.4 서비스 순서 결정

동시에 들어온 서비스에 대해서 서비스 우선 순위를 결정해서 적용하면 디스크에서의 입/출력 시간을 단축함으로써 버퍼 사용량을 줄일 수 있다.

다음 그림에서는 데이터에 대한 상대적인 접근 시간을 보여 주고 있다. 시작점으로 다시 돌아오는 최단 경로를 찾는 문제로 생각 할 수 있다.

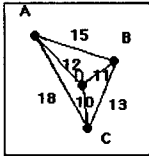


그림 5 상대적인 입출력 시간

그림에서 최단 거리를 구하면, 결과적으로 A-C경로만 거치지 않은 면 거치는 것에 비해서 4만분의 시간이 절약된다. 본 논문에서 제안하는 서비스 우선 순위 방법은 디스크에서의 입출력 시간의 순서가 서비스를 제공하는 중간에 바뀌지 않는다는 가정을 전제로 한다. 즉 디스크에 데이터가 연속적으로 저장된 경우를 말한다. 그 이유는 서비스 순서를 중간에 교체하는 경우 서비스에 차질이 생기기 때문이다.

4. 실험 및 분석

실험은 pentium 150Mhz, Linux5.2 red Hat의 운영체제를 사용하였고, 메모리 사용량은 배열을 할당하여 측정을 하였다. 실험에서 사용한 데이터는 50kbps, 40kbps, 20kbps를 사용했다.

다음 그림은 지연 시간을 적용하여 얻은 시간에 따른 버퍼 사용량을 보여 준다.

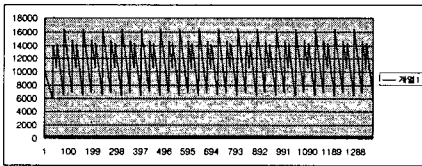


그림 6 지연 시간에 의한 버퍼 사용량

지연 시간을 적용하지 않았을 때와 비교했을 때 약 27%의 버퍼 사용량이 절약되었다. 그 만큼의 디스크 입출력 횟수가 감소한다.

지연 시간을 찾기 위해서 주기를 $(0, 2\pi)$ 로 계산을 했기 때문에 결과 값을 찾기 위해서, 즉 $\cos \tau_2 = -0.65$ 인 $\tau_2 = 157.66835499$ 이 되는데, 이론 값을 구할 때 소수점이 3자리 이하의 값을 버렸기 때문에 발생한 오차, 실험에서 적용 오차가 발생하였기 때문에 이론 값을 정확히 찾아내서 적용하면 더 많은 양이 절약 될 것으로 예상된다.

그러나 지연시간을 적용함으로써 동시에 서비스하는 것이 약간의 시간의 차이를 두게 되었고, 버퍼모델이 바뀜으로 인해서 모든 round에 대해서 버퍼에서의 데이터 위치가 바뀌어서 처리 시간이 증가하여 시스템의 성능 저하가 초래된다. 실험을 통해서 입증하지는 못했지만, 이 문제는 입출력 속도(기계적 속도)에 비해서 전자적인 처리 시간이 훨씬 빠르기 때문에 절약된 버퍼의 크기에 의한 입출력 감소를 증가하지 않을 것으로 예상된다.

그리고 서비스 우선 순위 결정은 본 논문에서 3개의 데이터에 대해서 실험을 했기 때문에 적용하지 않았다. 많은 수의 데이터를 동시에 서비스해야 할 경우가 발생했을 때 적용하면 더 많은 버퍼의 절약 효과를 얻을 것으로 예상된다.

5. 결론

서버 시스템에 동시에 여러 사용자가 서비스 요청을 해 왔을 때 프로세서가 하나인 경우, 정확히 동시에 수행 할 수는 없지만 입출력 시간이 특정시간에 모이게 되고, 버퍼의 사용량도 비효율성을 보이게 된다. 데이터의 특성을 고려해서 적당한 지연시간을 줌으로써 디스크에서의 입출력을 분산시켰고 결국 데이터 트랙픽을 감소 시켰다.

참고문헌

[1]S.V Raghavan/Satish K.Tripathi," Networked Multimedia Systems concepts,Architecture, and design, page 122-125
 [2].이시진, 양택훈,김영찬"데이터 전송률과 소비율에 기반한 제한 블록할당에서의 효율적 버퍼 크기 결정",정보학회논문지(A) 제 23권 제 10호(96.10)
 [3]Prashant J.Shenoy,pawan Goyal, and harrick M.Vin,"Issue In Multimedia server Design", (shenoy,pawang,vin)@cs.utexas.edu,1998
 [4] Banu özden, Rajeew Rastogi,Avi Silberschatz," Research Issues in Multimedia Storage Servers,1998",At&T Bell laboratories 600 Mountain Avenue murray Hill,NJ 07974
 [5]Edward chang and Yi-Yin chen,"Minimizing Memory Requirements in Media servers" ,http://www-db.stanford.edu/pub/papers/proofs.ps,1998
 [6]Babak hamidzadeh,Tsung-ping J.,"Dynamic scheduling techniques for interactive hypermedia servers",IEEE Transactions on consumer Electronics, Vol.45,No1,February 1999.
 [7].Edward chang and Yi-Yin chen,"Minimizing Memory Requirements in Media servers" ,http://www-db.stanford.edu/pub/papers/proofs.ps,1998
 [8].Ronald N.Bracewell," The Fourier Transform and Its Applications",p143-146,186