

분산 멀티미디어 플레이어 기반 웹 미디어 동기화에 관한 연구

이민경, 조동섭
이화여자대학교 컴퓨터공학과

A Study about Web Media Synchronization based on Distributed Media Player

Min-kyung Lee, Dong-sub Cho
Dept. of Computer Science and Engineering, Ewha Womans University

Abstract - 인터넷의 급속한 발달은 새로운 형태의 응용시스템 뿐만 아니라, 다양한 산업분야에 영향을 미치고 있다. 멀티미디어 환경의 사용자들은 새로운 정보에 대한 욕구와 차별화된 서비스를 요구함에 따라 새로운 인터넷 기반으로 변화를 추진시키고 있다. 특히, 멀티미디어 데이터는 사운드와 이미지, 영상이 결합된 매체이므로 데이터의 크기가 크며, 전체적인 네트워크의 오버헤드를 발생시키는 요인이 될 수 있다. 따라서 본 논문에서는 기존의 네트워크에서 발생할 수 있는 멀티미디어 데이터에 대한 오버헤드를 최소화하기 위해 여러 대의 시스템을 병렬포트로 제어하여 이를 분산 환경에 적용시키고자 한다. 이에 제안방식에서는 병렬시스템에서의 동기화 문제뿐만 아니라, 기존 네트워크 시스템의 오버헤드, 효율적인 네트워크 및 데이터 관리 방식을 제공한다.

1. 서 론

컴퓨터와 통신 기술의 발달에 따라 다양한 매체들을 컴퓨터를 이용하여 종합적으로 관리하고자 하는 시도가 이루어지고 있다. 과거에 컴퓨터를 이용하지 않고는 생각할 수 없었던 작업들이 컴퓨터를 이용해 이루어짐에 따라 TV(Television)나 VCR(Video cassette recorder), 영사기를 통해 재생되던 영화를 컴퓨터가 처리할 수 있는 데이터로 변환해 컴퓨터 모니터를 통해 이용하는 일이 가능하게 되었다.

멀티미디어 데이터는 음성과 문자, 영상 등이 혼합된 복합매체로 정보의 양이 크고 처리 과정이 복잡하다[1]. 생활의 데이터를 가상공간에서 이용할 수 있는 데이터로 변환하기 위해서는 여러 가지 과정을 거쳐야 한다. 영상이나 음성의 아날로그(Analogue) 신호를 디지털(Digital)신호를 바꾸는 인코딩(Encoding)과 인코딩을 통해서 전환된 디지털 신호를 사용자가 사용할 수 있는 영상이나 음성으로 바꾸는 디코딩(Decoding)기술, 처리된 데이터의 전송하기 위해 고속의 전송 선로와 다양한 대역폭을 제공하기 위한 교환기술, 그리고 영상과 음성 신호의 처리 기술이 필요하다. 멀티미디어 매체들은 영상과 소리를 주로 취급하기 때문에 정보의 양이 방대하고 이들 자원의 처리가 매우 복잡하고 까다롭다. 우리가 이용하는 동영상이나 화상전화 등의 매체가 멀티미디어의 대표적인 예라고 할 수 있다.

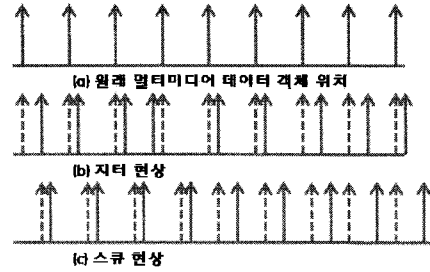
또한, 정보 통신 기술의 발달로 지역적으로 떨어져 있는 컴퓨터 시스템들을 데이터 통신망으로 연결하고, 이를 통해 다양한 정보 서비스를 제공하는 분산 시스템이 실현되고 있다[1]. 웹을 이용해 분산된 공간에 있는 시스템을 한 공간에 있는 것처럼 제어하기 위해서는 네트워크상에서의 데이터의 처리가 특히 중요하다. 특히, 멀티미디어 응용서비스 중에서도 동영상과 화상전화와 같은 서비스는 데이터의 도착 시간과 순서 등이 중요한 때가 있다. 인터넷과 같은 기존의 네트워크 환경은 전송되는 데이터에 대한 확실한 보장을 해주지 않기 때문에 멀티미디어 데이터를 처리하는 네트워크상에서의 연구가 필요하다.

주5일제 근무의 시행 등, 여러 요소로 인한 사람들의 여가시간의 증가는 멀티미디어매체를 이용하려는 사용자들의 증가 또한 포함하고 있다. 기하급수적으로 늘어나는 사용자들의 요구에 대해 단순히 더 많은 시스템을 연결하는 것만으로는 시스템의 성능 향상에 큰 영향을 미치지 못한다. 대규모 병렬처리 시스템에서 처리속도를 극대화하기 위해서는 하드웨어뿐만 아니라 소프트웨어 적인 요소들도 고려되어야 한다.

본 논문의 2장에서는 분산 멀티미디어 환경에서 발생할 수 있는 문제점을 기술하고 3장에서는 문제점을 해결하기 위한 동기화 기능을 추가한 분산 멀티미디어 제어 시스템을 제안한다. 4장에서는 본 논문에서 제안방식을 분석하고 마지막으로 5장에서는 결론 및 향후 연구방향에 대해 기술하고자 한다.

2. 관련 연구

본 장에서는 분산 멀티미디어 환경에서 발생할 수 있는 문제점인, 지연현상과 스큐현상 그리고 동기화에 대해 알아본다.



<그림 1> 지터 및 스큐 현상

2.1 지연 기술 (delay description)

네트워크에서 지터(Jitter)현상과 스큐(Skew)현상은 시스템간의 통신에 장애 요소가 되고 있다[6].

2.2.1 지터현상

<그림1>의 (b)에서 객체들의 실제 위치는 의도된 위치에서 임의의 값만큼 벗어나 있다. 지터는 수신지에 도착하는 패킷들의 지연되는 정도가 임의적으로 변화되는 현상을 의미한다. 지터의 발생으로 동영상 화면이 순간적으로 빠르게 혹은 느리게 진행되기도 한다. 늦게 도착한 패킷은 재생을 하지 못하게 됨으로써, 멀티미디어데이터의 응용에서 품질 저하의 원인이 되기도 한다.

2.2.2 스큐현상

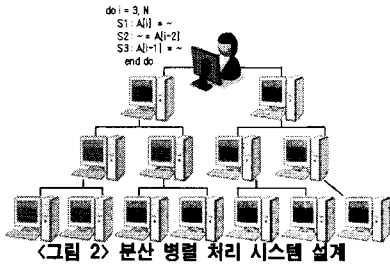
<그림1>의 (c)에서 의도된 위치와 멀티미디어 객체들의 실제 표현 위치의 차이가 일정하게 증가하는 것이다. 지연스큐로 인해 동영상 화면이 점점 느리게 재생되기도 한다. 스큐의 발생은 동영상 화면이 점차적으로 느리게 진행되게 한다. 늦게 도착한 패킷은 재생 시에 음성과 영상이 분리된 화면으로 나타날 수 있으며 품질 저하의 원인이 되기도 한다.

2.2 동기화 (Synchronization)

일반적인 동기화의 의미는 자신과 상대 간에 서로 정보를 일치시킨다는 의미를 갖는다[4]. 동기식 디지털 전송의 경우, 통상적으로 수신되는 비트들을 패킷의 단위로 나누어 동기를 유지한 채, 통신하도록 하여 효율을 개선하고 속도를 높이기 위해 설계된 방식이다. 동기화는 송신측 및 수신측 간의 데이터를 주고받는 시점을 정확하게 유지시키도록 하는 것이다. 데이터가 요구되는 멀티미디어 데이터의 통신시 정교한 시간 동기화가 필요하다. 영상과 사운드 등의 멀티미디어 데이터 처리가 한꺼번에 이루어지지 않는다면 사용자 입장에서는 소리와 영상이 따로 움직이는 화면을 접하게 되는 경우도 있다.

3. 분산 멀티미디어 플레이어 기반 웹 미디어 동기화 설계

정보를 효율적으로 교환하고, 처리하기 위한 통신시스템이 네트워크이다. 크기가 큰 멀티미디어 데이터를 효율적으로 처리하기 위해 한 대가 아닌 다수의 시스템을 연결해 사용한다. 노드 시스템들은 관리자 시스템에 레이블을 형성하며 연결되고 연결 장치들은 전송 매체로 서로 연결되어, 네트워크를 형성한다.



〈그림 2〉 분산 병렬 처리 시스템 설계

〈그림2〉에서와 같이 하나의 시스템은 두 개의 하위 노드 시스템을 가질 수 있고, 새로운 시스템이 추가될 때마다 같은 레벨에 노드 시스템을 차례로 하나씩 할당한다. 같은 레벨에 더 이상 할당할 수 있는 영역이 없을 때에는 하위 레벨을 생성하고 위의 과정을 반복하며 노드 시스템을 추가시킬 수 있다. 상위 레벨 시스템에 하위 레벨 시스템이 연결될 때, 하위 레벨의 노드 시스템들은 임의의 번호를 할당 받고 임의의 번호는 곧 자신의 id로써 시스템의 식별에 필요하다.

하위 노드 시스템들은 자신에게 전달된 명령을 실행 할 때마다 실행 메시지를 상위레벨의 시스템에게 전달한다. 상위레벨 시스템은 자신에게 연결된 하위레벨의 2개의 시스템 중에 어떤 시스템으로부터 전달된 명령인지 각각의 id를 통해 식별할 수 있다.

분산 병렬처리 시스템에서 명령은 상위시스템에서 하위시스템으로 전달된다. 이때, 관리자시스템에서 전달되는 명령은 시스템에 연결된 노드 시스템이 아닌 레벨의 수만큼 전달된다. 상위레벨에서 전달되는 명령이 연결된 하위레벨의 두 노드 시스템에 동시에 전송이 되지 않는다면 하위레벨의 시스템은 같은 레벨에 있는 시스템임에도 불구하고 동작을 달리하게 된다. 특히 상위레벨에서 전달되는 명령이 종속성[4]을 가지는 경우에는 먼저 수행된 시스템에 의해 변경된 데이터가 다른 시스템에 영향을 미치게 될 수도 있다.

이러한 문제를 해결하기 위해 순차적인 프로그램을 자동적인 병렬처리가 가능한 형태로 변환해 주기 위한 재구조화 기법이 다양하게 존재한다. 그렇지만 분산 멀티미디어의 응용에서 복잡한 데이터의 종속을 갖는 반복(Recurrence) 루프나 복수 지정문 등을 포함하는 재구조화 기법들을 적용할 수 없는 루프가 상당히 존재한다[4].

```
do i = 3, N
  S1 : A[i] = ~
  S2 : ~ = A[i-2]
  S3 : A[i-1] = ~
end do
```

〈그림 3〉 예제 루프

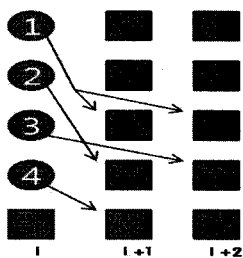
〈그림3〉에 제시된 예제문에서 쓰기, 읽기, 쓰기의 연산을 반복한다. 이 경우, 상위시스템에서 자신에게 연결된 두 하위 시스템에 같은 명령을 전달했을 때, 먼저 실행된 시스템에 의해 i의 값이 변경되어 원하는 동작을 하지 못하는 경우가 발생할 수 있다.

만약에 재구조화 방법을 적용하여 병렬처리가 가능해졌다고 해도 대개의 경우 본래의 순차 프로그램보다 프로그램 코드수가 늘어나거나 연산시 소요되는 오버헤드가 커진다.

본 논문에서는 순차적으로 전달되는 명령의 올바른 수행순서를 유지하기 위해서, 각 명령에 동기화 변수, SC(Statement Counter)를 할당 하도록 한다. 이 동기화 변수는 관리자 시스템에 연결된 모든 노드 시스템과 공유되도록 한다.

예를 들어, 〈그림3〉의 예제루프에서와 같은 반복문에서 i가 문장 S1의 수행을 완료 한 후, SC의 값을 i로 증가하기 전에 SC = i-1이 될 때까지 대기하게 한다. 따라서 SC의 값이 증가하기 전에 루프문에서 i가 증가하기 전에 시스템상의 모든 루프를 실행하게 한다.

초기에 첫 번째 반복이 k라면 SC는 k-1의 값을 가진다. 반복 i가 sink 문장 Sq를 실행하기 위해서는 이전의 해당 소스 문장의 수행을 점검한다.



〈그림 4〉 동기화 수행 방법

〈그림4〉와 같이 종속 관계가 있는 루프문장을 수행할 때, 순차적으로 동기화 변수의 상태를 수정한다. 한 대의 상위레벨의 시스템으로부터 전달 받은 제어 명령이 두 대의 하위레벨 시스템으로 전달될 때, 다른 한쪽의 시스템으로부터 변경된 변수에 의해 받는 영향을 최소화 할 수 있도록 하고 있다.

4. 제안방식 분석

본 논문에서는 분산 멀티미디어 플레이어를 기반으로 한 웹 미디어 동기화에 대해 연구하였다. 인터넷 상에 흩어져 있는 방대한 양의 멀티미디어 데이터를 제어하기 위해서는 병렬처리 시스템을 사용하였다. 병렬처리 시스템은 유연한 확장과 불필요한 시스템의 제거가 용이하다.

시스템은 명령에 의해 실행이 될 때마다 실행 메시지를 작성, 상위노드에 전달함으로써 상위시스템이 자신에게 연결된 하위시스템의 상태를 체크하며 통신할 수 있다. 실행메세지에는 임의적으로 부여된 고유 id가 붙어서 전송되기 때문에 상위 레벨의 시스템은 자신에게 연결된 하위 레벨의 2대의 시스템 중 어느 쪽의 메시지인지를 식별할 수 있다. 따라서 메시지에 의한 개별적인 동작 제어를 할 수 있게 해야 한다. 기존에 링형시스템에서는 관리자시스템이 모든 노드시스템들을 관리해야 했지만 본 논문에서 제안된 병렬처리 시스템에서는 관리자 시스템은 연결된 모든 시스템들의 배치와 레벨생성만 관리하기 때문에 그 부하가 적게 이루어진다. 또한 각 하위 노드에 대한 제어가 상위 노드에서 이루어짐으로써 기존의 관리 시스템 한 대가 가지던 부하를 다수의 시스템으로 나누어 관리 할 수 있고 윈도우 이벤트에 따른 개별적 제어도 가능하기 때문에 편리하게 사용할 수 있다.

또한, 네트워크 환경에서 발생하는 지터나 스큐에 의한 지연을 줄이기 위해 동기화를 적용, 지연에 의해 발생하는 에러를 최소화 하였다. 동기화 변수를 공유하기 때문에 문장의 종속 관계에 따라 순차적으로 동기화 변수의 상태를 수정할 수 있다.

5. 결론

통신환경의 발달에 따라 다양한 멀티미디어 응용이 가능해지고 있다. 기하급수적으로 늘어가는 사용자들의 다양한 욕구를 충족시키기 위해서는 대형 시스템이 필요하다. 그렇지만 대형 시스템은 구성에도 많은 비용이 들 뿐만 아니라, 아무리 대형 시스템이라도 처리할 수 있는 데이터의 양이 한정되어 있어 어려움이 있다. 그래서 이용할 수 있는 방법이 여러 대의 시스템을 연결하여 하나의 시스템으로 사용하는 것이다.

병렬포트로 연결된 시스템에서 멀티미디어 데이터를 이용할 때는, 지터나 스큐의 발생이 따른다. 지터나 스큐의 발생은 멀티미디어 데이터를 이용하는 사용자에게 불편을 끼치기 때문에 싱크(sync)를 맞추는 개념의 동기화가 필요하다. 본 논문에서는 순차적으로 전달되는 명령의 올바른 수행순서를 유지하기 위해서, 각 명령은 동기화 변수, SC를 할당 하도록 한다. 이 동기화 변수는 관리자 시스템에 연결된 모든 노드 시스템과 공유되고 시스템에서의 모든 문장을 수행한 후에, SC의 사용이 끝나면 다시 초기값을 주는 방식을 사용하도록 한다.

〈참 고 문 헌〉

[1] Satochi Itaya, Naohiro Hayashibara, Tomoya Enoki do. Makoto Taki waqa, "Distributed Coordination for Scalable Multimedia Streaming Mode", P 134-146, 2000
 [2] Richard M.Adler, "Distributed Coordination Models for Client/Server Computing", IEEE Computer, P14-22, 1995
 [3] Robert Orfali, Dan Harkey, "Client/Server Programming with Java and CORVA-2nd", 1998
 [4] 박두순, 이광형, 황종선, 김병수, "병렬처리를 위한 동기화 기법", 정보과학회지 제13권 제 7호 P130-142, 1995
 [5] 박수환, 이창건, 하은용, "분산 컴포넌트 시스템에서 동기화된 시간의 정밀도를 높일 수 있는 기법" 한국컴퓨터종합학술대회 논문집 vol.35, No1(B), P500-503, 2008
 [6] 김명호, 박용진, 김원태, "명시적 멀티캐스트 프로토콜 및 응용의 구현" 한국정보과학회 가을 학술발표논문집 vol.29, No2, 2002
 [7] 김명호, 이윤준, 정연돈, "멀티미디어 시스템 개론", 홍릉과학출판사, 2006
 [7]http://www.purebasic.com