

분산 멀티미디어 병렬처리 시스템 연구

이민경*, 조동섭*

*이화여자대학교 컴퓨터공학과

e-mail : mg426@ewhain.net

A Study on distributed Multimedia Parallel Processing System

Min-Kyung Lee*, Dong-Sub Cho*

* Dept. of Computer Engineering, Ewha Womans University

요 약

통신환경의 발달은 그에 따른 응용시스템의 개발을 촉진하며 빠르게 변화하고 있다. 그래서 많은 사용자들이 이용하는 멀티미디어 데이터의 경우, 통신환경에서의 다양한 응용이 중요시되고 있다. 일반적으로 멀티미디어 데이터의 경우 크기가 크기 때문에, 데이터를 처리하는 데는 많은 노력이 필요하다. 각 시스템에서 처리할 수 있는 일의 양은 한정 되어 있기 때문에 대형 시스템 한대로 분산 환경에서의 멀티미디어 데이터를 처리하고 제어하는 데는 한계가 있다. 본 논문에서는 여러 대의 시스템을 병렬 연결해 하나의 시스템처럼 사용하게 한다. 병렬포트제어를 통해 연결된 시스템들은 상위레벨, 하위레벨과 연결상태 및 실행상태를 체크하며 통신한다. 본 논문에서 제안된 시스템은 관리자가 좀 더 빠르고 편리하게 여러 대의 시스템을 제어할 수 있도록 한다.

1. 서론

인터넷과 네트워크 환경의 발달은 그에 따른 응용 시스템의 개발을 촉진하고 있다. 2015 년까지 정보통신부 주관으로 구축되고 있는 광대역 정보통신망(B-ISDN) 사업계획에[4] 따라 통신환경은 빠르게 변화할 전망이다.

특히 많은 사람들이 이용하는 멀티미디어 데이터 데이터의 경우, 보다 더 다양한 방면에서의 응용이 필요하다. 기하급수적으로 늘어나는 인터넷 사용자들에게 더 좋은 서비스를 제공하기 위해서는 멀티미디어 데이터의 특성을 이해하는 연구가 계속되어야 한다.

멀티미디어 데이터는 컴퓨터를 통해서 종합 관리되는 영상과 사운드, 이미지가 결합된 복합매체로서 데이터의 크기가 크고 처리과정이 복잡하다[6].

시스템은 처리할 수 있는 양의 제한되어 있기 때문에 현대의 대형시스템을 사용해 용량이 큰 멀티미디어 데이터를 처리하는 데는 제한이 있다. 특히 분산 환경에서 많은 데이터를 처리해야 할 때에는 어려움이 따른다. 이때, 처리속도를 향상시키기 위해 고려해 할 수 있는 방법에는 여러 대의 시스템을 연결해 하나의 시스템으로 사용할 수 있게 하는 것이다. 시스템들은 관리시스템에 트리구조로 병렬 연결된다. 상위노드(node)에 전달된 명령은 그들의 하위노드로 전달된다. 본 논문의 2 장에서는 기존의 멀티미디어 처리 방법에 대해 알아보고 3 장에서는 분산 멀티미디어 병렬 처리 시스템을 제안한다. 4 장에서는 분산 멀티미디어 병렬처리시스템을 구현하고 5 장에서는 제안된 시스템의 장점에 대해 말한다. 마지막으로 6 장에서는 결론 및 향후 연구방향을 기술하고자 한다.

2. 관련연구

분산 환경에서 노드 컴퓨터의 적절한 동적 재구성 기능만 제공된다면 멀티 유저 환경을 제공하면서 컴퓨터의 사용효율도 극대화시킬 수 있다.

컴퓨터가 지식이나 멀티미디어를 처리하기 위해서는 엄청난 속도를 요구한다. 이때, 고속의 처리속도를 얻는 방법은 다음과 같은 2 가지가 있다[6].

첫 번째는 슈퍼컴퓨터에서 사용하는 방법으로 ECL(Emitter Coupled Logic)이나 GaAs(Gallium Arsenide) 같은 고속 소자를 이용하여 복잡한 연산을 수행하는 빠른 하드웨어를 직접 구현하는 방법이다.

두 번째 방법은 프로세서를 여러 개 사용하여 명령 처리 시스템을 구현하는 방법이다.

첫 번째 방법으로 분산 시스템을 사용하면 시스템 하나의 가격이 비싸질 뿐만 아니라, 시스템을 사용해서 얻을 수 있는 성능에도 한계가 있다. 그래서 주로 두 번째 방법이 이용된다.

두 번째 방법을 이용한 분산 시스템의 병렬처리를 위해서 구현이 비교적 쉬운 멀티프로세서시스템을 많이 이용한다. 멀티프로세서시스템은 대체로 공유버스와 공유메모리를 이용한다. 이 방법은 구현이 용이하지만 공유버스/공유메모리에서의 병목현상 때문에 프로세서의 수를 증가시킬 수 없다는 문제점이 있다. 또한 병렬처리 시스템은 인간중심적인 차세대 고급 정보 처리 컴퓨터나 지능형 컴퓨터에서 요구되는 추론 또는 처리속도를 얻을 수 없다는 단점도 가지고 있다.

일반적으로 병렬처리 시스템의 구현에서 가장 중요한 설계요소중의 하나가 주어진 태스크(Task)의 병렬성에 따른 그레놀래리티(Granularity)레벨이다. 그레놀래리티는 선명도를 묘사하기 위해 제공되는 정보의

량을 조절하는 것이다.

범용 컴퓨터에 대한 컴퓨터 사용자의 요구는 다양한 레벨의 그래픽래티티를 요구한다. 많은 수의 시스템이 필요한 태스크일 수도 있고 적은 수의 시스템이 필요한 태스크 일 수도 있다. 많은 수의 컴퓨터가 요구되는 태스크에 맞추어 구현된 시스템에서는 적은 수의 컴퓨터가 요구되는 태스크의 수행 시 사용 효율이 저하된다. 또한, 그 반대의 경우는 병렬처리를 통한 고성능이 얻어지는 효과가 떨어진다. 따라서 그래픽래티티 레벨의 조절은 중요하다.

멀티미디어 지식처리를 위한 병렬처리 컴퓨터 구조를 위해서는 여러 개의 유닛(Unit)이 필요하다.

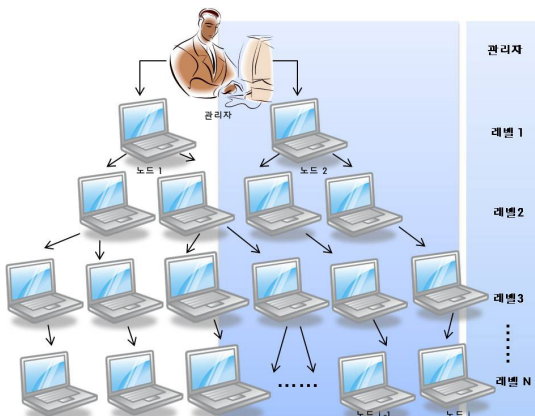
<표 1> 멀티미디어 데이터 처리를 위해 필요한 유닛

메시지스토리지유닛 (Message storage unit)	가공된 데이터를 파일로 저장하기 위한 유닛
네트워크인터페이스유닛 (Network interface unit)	외부네트워크와의 인터페이스를 제공하기 위한 유닛
연산유닛 (Processing unit)	현재 사용할 수 있는 유닛의 수를 판단하기 위한 연산을 하는 유닛
I/O 유닛 (Input/Output unit)	Input 과 Output 을 처리하기 위한 유닛
통신유닛 (Communication unit)	통신을 위한 채널을 제공하기 위한 유닛

<표 1>에 명시된 여러 가지 유닛들을 이용하여 사용자가 요청한 태스크에 대해 가변적인 할당을 하도록 해 사용자에게 최적의 상태로 시스템을 사용할 수 있도록 한다.

3. 분산 멀티미디어 병렬제어 시스템 설계 제안

본 논문에서는 관리자가 인터넷상에 분산되어 있는 멀티미디어 데이터를 쉽게 제어하기 위해 병렬포트제어를 이용한 병렬처리구조를 제안한다. 효율적인 관리를 위해 단말시스템의 추가 시 그래픽래티티를 고려한 레벨의 생성과 노드의 배치, 관리가 이루어질 수 있도록 한다.



(그림 1) 분산 멀티미디어 병렬제어 시스템 설계

(그림 1)에서와 같이 하나의 시스템은 두 개의 하위 노드 시스템을 가질 수 있고, 새로운 시스템이 추가 될 때마다 같은 레벨의 시스템에 차례로 하나씩 할당한다. 같은 레벨에 더 이상 할 당할 수 있는 영역이 없을 때에는 하위 레벨을 생성하고 위의 과정을 반복한다.

(그림 1)에서와 같이 레벨 1 은 2 개의 시스템을 할당 받았고, 레벨 2 는 2*2 개, 즉 4 개의 노드 시스템을 할당 받았다. 이러한 과정을 거쳐 N 번째 하위 레벨은 최대 2^N 개의 시스템을 할당 받을 수 있다.

네트워크상에 물리적으로 분리되어 있는 시스템들을 효율적으로 제어하기 위해서 관리자는 사용자들을 개별적, 동시적으로 제어할 수 있어야 한다.

관리자 시스템에서 실행하는 명령은 노드 1, 노드 2 시스템을 거쳐 전체 시스템에게 전달되어 동시적으로 실행될 수 있다. 한편, 노드 1 시스템에서 실행하는 명령은 노드 1 과 연결된 노드 3, 노드 4 시스템을 거쳐 노드 1 시스템을 상위 레벨로 가지는 모든 노드 시스템에게 전달될 수 있다. 멀티태스킹과 같이 필요한 시스템에만 부분적인 전달이 될 수 있다.

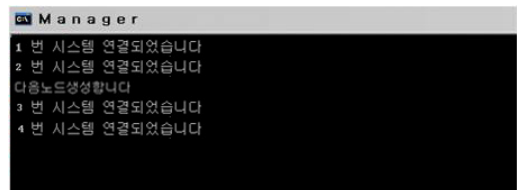
관리자에 의해 전달된 명령은 레벨 1 의 각 노드 시스템들은 전달되고 레벨 1 과 연결된 레벨 2,3,4 ... N 의 시스템에게 순차적으로 전달된다. 각 레벨의 노드 시스템들은 전달받은 명령을 실행한다.

예를 들어, 관리자가 12 대의 시스템에 데이터를 전송하려 할 때에는 12 대의 노드 시스템은 3 개의 레벨로 나누어 진다. 관리자에 의해 직접 전달되는 명령은 레벨 1 의 2 개의 노드 시스템뿐이고, 레벨 1 의 시스템들은 레벨 2 의 시스템들에 관리자 시스템으로부터 전송된 명령을 전달한다.

인터넷상에는 수 많은 시스템들이 연결되어 있기 때문에 N 개의 시스템을 제어하기 위해 N 번의 데이터를 전송해야 하는 기존의 시스템보다 병렬처리 시스템의 사용은 효율성이 높다.

4. 분산 멀티미디어 병렬제어 시스템 구현

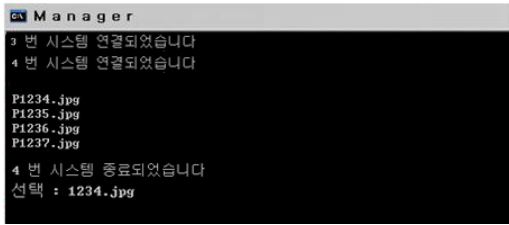
본 논문에서 제안하는 시스템은 Purebasic 을 사용하여 구현되었다. 시스템은 각각 실행화면과 제어화면을 가진다.



(그림 2) 관리자 시스템의 연결성 Test 화면

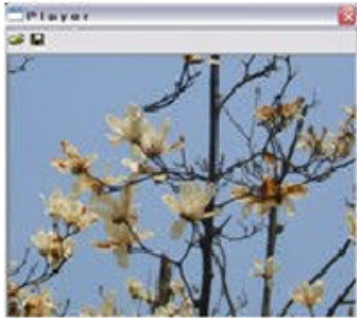
(그림 2)는 시스템 전체를 제어하기 위한 관리자 시스템에서 시스템에 연결되는 각 노드 시스템들을 테스트 하는 화면이다. 관리자 시스템은 전체 네트워크에 새롭게 추가되거나 제거되는 노드 시스템에 대한 전체적인 정보를 알고 있어야 한다. 또한, 그에 따

른 레벨의 생성과 제거 등 네트워크에 대한 종합적인 정보를 알고 있어야 한다.



(그림 3) 하위노드 시스템의 화면

(그림 3)은 하위노드 시스템의 화면이다. 노드 시스템이 관리자 시스템에 연결될 때마다 임의로 시스템의 id 를 생성한다. 시스템들은 모든 응답에 자신임을 표시하는 id 를 붙여서 보낸다. 따라서 상위 시스템은 자신에게 연결된 하위노드 시스템의 상태를 식별번호를 통해 쉽게 확인할 수 있다.



(그림 4) 하위노드 시스템의 실행화면

기본적으로 본 논문에서 제안하는 시스템은 미디어 플레이어 시스템으로 음악파일과 이미지파일을 실행할 수 있다. 선택된 파일에 따라 이미지뷰어나 뮤직플레이어가 실행된다. Jpg, gif 등 이미지 파일이 선택되면 (그림 4)와 같이 이미지뷰어가 실행되고 선택된 이미지가 보여진다. 이미지뷰어는 기본적으로 이미지를 보여주거나 저장하고 앞, 뒤의 이미지로 넘어가는 기능을 실행할 수 있다.

Mp3, wma 등의 음악 파일이 선택되면 뮤직플레이어가 실행되고 재생, 일시 정지, 정지, 볼륨조절 등의 기본적인 뮤직플레이어 기능을 가진다. 선택된 파일의 이름이 실행 목록에 추가된다. 한 시스템에서 이미지뷰어나 뮤직플레이어는 각각 개별적으로 실행할 수도 있고 동시에 실행할 수도 있다.

또한, 상위 노드의 명령에 따라 연결된 각 시스템에서 독립적으로 실행되기도 하고 동시에 실행되기도 한다. 기본적으로 상위노드, 관리자의 제어에 의해 작동하지만 각 시스템의 윈도우 이벤트에 의해서도 작동할 수 있다.

5. 장점

본 논문에서 제안하는 시스템은 분산 멀티미디어병렬처리 시스템으로 기존의 인터넷상에 유니캐스트(Unicast) 방식을 사용해 스타형으로 연결된 시스템보

다 처리효율을 높이고자 하였다.

소켓은 사용자에게 의한 최적화가 가능하므로, 통신에 필요한 기능만을 최소화하여 빠르고 쉬운 제어를 할 수 있게 하였다. 기능을 수행할 때마다 실행 메시지를 작성, 상위노드에 전달 함으로써 상위시스템이 자신에게 연결된 하위 시스템의 상태를 체크하며 통신할 수 있다. 기존의 링형시스템에서는 관리자시스템이 모든 노드 시스템들을 관리해야 했지만 본 논문에서 제안된 병렬처리 시스템에서는 관리자 시스템은 연결된 모든 시스템들의 배치와 레벨 생성만 관리하기 때문에 그 부하가 적게 이루어진다. 각 하위 노드에 대한 제어가 상위 노드에서 이루어짐으로써 기존의 관리 시스템 한 대가 가지던 부하를 다수의 시스템으로 나눌 수 있고 윈도우 이벤트에 따른 개별적 제어도 가능하기 때문에 편리하게 사용할 수 있다.

6. 결론

인터넷과 네트워크 환경의 급속한 발달은 응용프로그램의 개발 또한 촉진하고 있다. 분산환경에서 멀티미디어 데이터의 사용을 위해 관리자는 시스템에 연결된 다른 시스템들을 효율적으로 제어해야 한다. 기하급수적으로 늘어나는 사용자들을 제어하기 위해서 관리 시스템의 처리속도 또한 향상되어야 한다.

본 논문에서는 관리시스템의 처리속도를 향상하기 위해 병렬제어시스템을 제안하였다. 관리자시스템에 연결된 시스템들은 트리 구조를 가지고 같은 레벨에서의 노드 시스템의 연결이 끝나면 하위레벨을 생성하고 노드 시스템들을 할당한다. 이에 따라 관리자의 명령은 상위 레벨의 노드 시스템들을 통해 하위 레벨의 노드 시스템들로 쉽게 전달 될 수 있으므로 관리 시스템에서 처리하는 일의 양도 줄일 수 있다.

본 논문에서 제안된 시스템은 병렬처리 과정에서 중요한 데이터의 동기화에 관한 부분을 처리하지 못했지만 향후 연구를 통해서 보완하겠다.

참고문헌

- [1] A Satoshi Itaya, Naohiro Hayashibara, Tomoya Enokido.Makoto Takiwaqa, "Distributed Coordination for Scalable Multimedia Streaming Model," IEEE2006
- [2] Richard M.Adler, "Distributed Coordination Models for Cliend/Server Computing," IEEE Computer, pp14-22. April, 1995
- [3] Robert Orfali, Dan Harkey, "Client/Server Programming with Java and CORVA"-2nd, 1998
- [4] 정운호, 김두현, 임영환, 김병기, "인지적 특성을 기반으로 한 화상회의용 멀티미디어 동기화 기법", 한국정보과학회 가을학술발표논문집 제 23 권 제 2 호,1996
- [5] 최덕호, 박용진, 김원태, "명시적 멀티캐스트 프로토콜 및 응용의 구현", 한국정보과학회 가을 학술발표논문집 vol.29, No2, 2002
- [6] 김명호, 이운준, 정연돈, "멀티미디어 시스템 개론", 홍릉과학출판사, 2006
- [7] <http://www.purebasic.com>