韓國精密工學會 '95年度 秋季學術大會論文集 pp.801~805

서버,클라이언트를 이용한 분산형 멀티미디어 데이터베이스 구축

(Construction of Distributed Multimedia Database using by server and client)

아주대학교 산업공학과

하태용, 신용백 , 왕지남

Abstract

멀티미디어(화상, 음성, 하이퍼텍스트)의 데이터는 다른 데이터 와 상이점이 많아 운용에 어려움이 있다. 대용량의 저장용량(storage)의 필요, 데이터 통신의 어려움 등이 빠르게 발전하는 멀티미디어 기술에 제약조건이 된다. 본 연구에서는 멀티미디어 데이터가 일반적인 텍스트(Text) 데이터 와 같이 높은 수행도(Performance) 및 안정적인 데이터베이스(DataBase)로 구축되어 효율적으로 운용 되는데 중점을 둔다.
아울러 대량의 정보를 처리하기 위하여 서버(server)와 클라이언트(Client)기법을 이용한 분산처리로 실시간 처리 및 데이터 저장의 한계를 극복하조자 한다.

1. 서론

본 연구에서는 멀티미디어 데이터 (화상,음성,하이퍼 텍스트)를 대상으로 데이터 들의 특성을 조사하고 이의 효율적인 운용을 위하여 관련 데이터베이스, 네트워크(Network), GUI(Graphic User Interface)에 따른 운용 등을 다룬다. 멀티미디어는 텍스트 데이터(Text data) 를 포함한 음성, 정지영상,동 영상 등의 미디어 데이터 들의 집합체로써 Static 데이터 와 real-time 데이터 로 구분할 수 있다. Static 데이터는 텍스트와 정지 화상이 있고 ,real-time 데이터는 오디오(Audio), 비디오(Video)등이 이에 포함될 수 있다.

멀티미디어 데이터의 특성을 살펴보면 아래와 같다.
1)아날로그로 된 데이터는 디지털로 전환이 필요하다.
비디오, 오디오등 물리적특성을 갖는 아날로그 데이터는 0과1로 표현할 수 있도록 디지탈(Digital)화 해야 한다. 이는 컴퓨터의 운용을 위한 기본적인 정규화(Normalizing)에 해당된다.
2)데이터 양이 방대하여 대용량의 저장용량를 요구한다.
음성이나 화상 데이터 는 보통의 텍스트 데이터에 비해 엄청난 데이터 를 갖는다.
3)시간에 종속적이다.(Real Time)
텍스트 데이터와 마찬가지로 on-line을 위한 처리 요구에 합당해야 한다.
4)미디어간 통신의 경우 동기화가 요구된다.(화상과 음성 동기)
영화와 같이 화상과 음성이 공존하는 경우의 데이터 에서는 그 동기화가 매 우 중요하다.
5)화면출력과 편집을 위한 GUI 툴(Tool)이 필요하다.
데이터의 조회 및 편집을 위하여 텍스트환경이 아닌 윈도우즈환경에서 툴이 필요하다.

멀티미디어 데이터처리를 위해 가장 간편한 방법은 화일스트림포인터(File Stream Pointer)를 이용하여 처리하는 것이다. 데이터베이스내용에는 화상이나 음성 등의 데이터를 직접 취급하는 것 이 아니라 현재 화일형태로 저장되어있는 데이터들의 화일 포인터를 관리함으로써 쉽게 구현할 수 는 있다 그러나 데이터 자체의 분산처리가 불가능하며 데이터 Handling을 위해서는 많은 제약 점이 있는 바 가장 원시적인 처리 방법이다. 본 논문에서는 멀티미디어 데이터 자체로써 이진(Binary) 데이터 만을 대상으로 한다. 지금까지 멀티미디어는 하드웨어및 소프트웨어 측면에서 매우 많은 발전이 진행되었으며 데이터 의 저장 및 통신을 위한 알고리즘 측면에서도 경이로운 성과가 있었다.
그러나 아직도 대용량의 이미지 및 음성 데이터 를 광역에서 실시간으로 운용하는데는 그 한계가 있

다.데이터 의 저장 및 그 운용은 컴퓨터의 역사와 함께 했으며 데이타베이스는 매우 중요한 분야로서 연구되고 있고 ,프로젝트의 성패를 좌우하는 결정적인 문제로 대두될 수 있다. 현재 가장 많이 사용되는 데이터베이스는 관계형 데이터베이스로서 그 예를 제품명으로 살펴보면 인포믹스,오라클, 사이베이스,인그레스,쿱타,기타 SQL(Structured Query Language)서버 등이 있는데 모두가 관계형 데이타베이스임을 알 수 있다. 이러한 관계형 데이타베이스로써 멀티미디어의 운용을 구현하고자 한 많은 연구가 있었지만 , 본 연구에서는 계층적 구조의 데이터베이스를 이용하여 데이터 의 분산을 꾀하며 Access Time을 최소화 하여 응답시간를 빠르게 하고 데이터 저장용량의 최적화를 실현하여 저장용량효율을 높이고자 한다. 또한 사용자와 친숙한 GUI구현으로 그 동안 멀티미디어 처리에 한계적인 요소를 극복하고 데이터 특성을 감안한 최적의 기법을 제시하고자 한다.
본 논문은 다음과 같이 구성된다.
제 2절에서는 멀티미디어 데이터 가 갖는 특성 대비 구현에 따른 문제점을 정의하고 제3절에서는 2절에서의 문제점을 상세히 분석하며 대안을 정립한다.제4절에서는 분석된 내용을 토대로 적용하고 제5절에서는 결론 부분으로 이루어진다.

2. 문제의 정의

멀티미디어 데이터 는 대용량을 요구한다.실시간 처리를 위해서는 데이터 엑세스 타임(Access Time)을 단축하는 것이 무엇보다도 중요한 관건이다. 따라서 데이터 를 한곳에 모아서 처리하는 중앙 집중식 혹은 호스트(Host)중심의 데이터베이스 처리는 바람직하지 못하다. 아울러 데이터베이스에 있어서 고정 크기 레코드(Record) 및 필드(Field)역시 이 요구에 적합하지 않다. 이는 멀티미디어 데이터는 그 크기에 있어 상당히 불규칙적이기 때문이다. 따라서 기존 관계형 데이타베이스가 가지고 있는 테이블(Table)단위 및 데이터 타입(Type)에 따른 구속력 있는 데이터 저장 방식은 그 운용과 저장용량의 효율성 측면에서 비효율적이고 많은Fragmentation산출의 원인이 될 수 있다.

다음은 데이터 입출력의 단위가 수행도에 미치는 영향을 고려하여 텍스트 데이터 와는 다른 각도에서 데이터 의 I/O(Input/Output) 블럭(Block) 크기를 결정해야 될 것이다. 멀티미디어 데이터 의 크기가 텍스트 데이터에 비하여 큰 것을 감안하여 I/O의 블럭 크기 역시 커야 되겠지만 자칫하면 많은 Fragmentation을 발생시켜 저장 용량의 효율을 떨어트릴 수 있다. 이는 또한 네트워크 구성 시 통신을 위한 기본적인 요소로서도 중요한 변수가 될 것이다. 이러한 난점을 해결하기 위해 본 논문에서는 분산처리를 바탕으로 하는 계층적 구조의 데이터베이스 구조를 적용하고, 동적 변수(Variable)를 이용한 데이터 의 형태 및 크기에 맞도록 데이터베이스를 이용하여 GUI로 이를 실현하고자 한다.

3. 분석 및 대안 제시

1) 계층적 구조의 데이터베이스 구조

계층적 구조의 데이터베이스는 관계형 데이터베이스 및 네트워크 데이터베이스와 같은 데이터베이스 모델의 한 종류로서 어느 한 노드(NODE)가 다른 노드를 지칭함으로써 하위 노드를 종속하는 트리(TREE)형태를 취하는 구조이다. 아래는 전형적인 트리 구조의 대표적인 예인데 각 노드는 하위 노드를 참조하여 궁극적인 종단 노드까지 가는데 빠른 시간 및 균등한 엑세스 타임이 소요된다. 아울러 각 데이터는 보조기억장치(HDD)에 저장하며 이를 지칭할 참조키(Reference Key)를 메모리(Memory)에 상주시킴으로써 가장 빠른 패스(Path)로 데이터를 엑세스할 것이다. 이는 관계형 데이타베이스를 이용하여 테이블의 특성치와 관계성을 가지고 특정 데이터를 엑세스하는 것 보다 빠른 수행도를 거둘 수 있다.

Root

메모리에 상주

Data

보조기억장치

또한 중복 요소를 피할 수 있으며 짧은 깊이(DEPTH)를 가지고도 많은 데이터를 Reference 할 수 있으며 이는 또한 분산처리를 위하여 다른 네트워크를 참조하여 원거리 데이터(remote data)를 빠른 시간으로 엑세스 할 수 있다.

2) 데이터 의 분산

SERVER

process#1

process#2

process#3

process#4

process#5

client#1

client#2

client#3

client#4

client#5

대량의 데이터를 저장하고 그 효율적인 운용을 위해서는 데이터 의 분산이 선행되어야 할 것이다. 자체 처리를 위한 데이터 는 가급적 로컬(Local)에 저장하고 필요한 경우의 데이터 만 네트워크를 이용하여 주고받는 방식을 채택한다. 본 논문에서는 이러한 분산처리를 위하여 데이터 들의 집합체인 DATASET을 어느 기계(Machine)에 셋팅(Setting)할 것인지를 매핑(Mapping)할 것인 지 서버와 클라이언트 기법을 이용하여 이를 분산처리 하였다.그러나 수행도 향상을 위하여 가급적 데이터 의 지역화(Localization)을 고려해야 함으로 업무 설계를 각 특성에 맞도록 분산시켜 설계하고 적정한 자원을 위한 합리적인 의사결정이 이루어 져야 된다. 아울러 데이터 의 증가로 인하여 하드웨어의 증설이 요구될 때 데이타베이스 구조의 변경이나 데이터베이스 운용에 많은 시간이 걸려 현업무에 지장을 초래하는 경우가 배제되어야 한다. 따라서 이를 위해서는 하드웨어의 이식성(Portability)이 뛰어나 소프트웨어의 변경 없이 하드웨어의 추가가 간편해야 한다. 또한 하드웨어 추가로 인한 비용 측면에서 경제성 평가가 이루어 져야 된다.

3). 동적 저장 방식(Dynamic Storage)

앞서 언급한 바가 있지만 고정 크기의 레코드 및 필드의 채택은 대량의 정보를 저장하기 위한 방안으로는 적합하지 않다. 이는 데이터 의 크기와 관계없이 불필요한 메모리를 점유함으로써 대량의 Fragmentation를 발생시킬 수 있기 때문이다. 따라서 본 논문에서는 데이터의 형태을 정하지 않고 하나의 데이터 형태만으로 처리한다. 즉 데이타의 형태를 문자열(String)만으로 처리한다. 또한 필요한 노드가 발생할 때 그 노드를 생성시키며, 데이터 역시 노드 생성시 데이터 크기에 따라 저장하는 동적 저장 형태를 취하는 것 이다. 여기서 멀티미디어 데이터의 특성을 감안하여 효율적인 저장 방식을 볼 때 먼저 데이터의 크기를 파악,정해진 I/O 블럭과 비교하여 최적의 한 방안이 적용되어야 할 것이다. 따라서 본 논문에서는 다음과 같은 대안을 제시 한다.

이 대안을 위하여 다음과 같은 기호를 이용하였다.
B : Input/Output을 위한 블럭의 크기 (Byte)
N : 멀티미디어 데이터 의 크기(Byte)
Q : N을 B로 나눈 몫
R : N을 B로 나눈 나머지

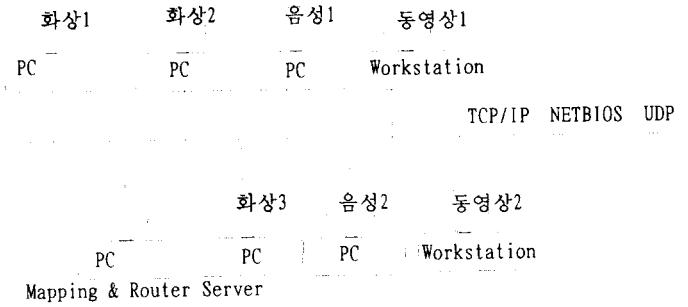
먼저 N을 B로 나누어 Q와 R을 구한 뒤 데이터 저장 및 조회 시 B 크기 만큼으로 Q회를 수행하고 나머지 R를 처리하는 알고리즘(Algorithm)이다. 이는 데이터 의 Fragmentation을 최소화 하고 I/O의 수행도를 극대화 하기 위한 방안이다. I/O의 블럭 크기가 2048 Byte이며 멀티미디어 데이터 가 화상으로, 그 크기가 300,123 Byte로 가정한다면 B:2048 , N:300,123 , Q=N/B=146, R=1115이다 . 즉 I/O횟수는 총 147회로 2048 Byte씩 146번 1115 Byte가 1번으로 이루어져 최소의 Fragmentation과 적은 I/O 엑세스로 수행도를 향상시킴을 알 수 있다. 따라서 데이터 의 성격에 맞게 I/O 의 블럭 크기를 설정한다면 더욱 효과적인 결과를 기대할 수 있을 것이다.

4). 효과적인 네트워크 구성

분산처리는 지역에 없고 원거리에 있는 데이터라 할 지라도 지역에 있는 데이터 처럼 사용하는 데 그 의미가 있다.이는 효과적인 네트워크이 구성되었을 때 바로 가능할 수 있다. 네트워크는 데이터 분산뿐만 아니라 명령 수행에 대하여 대한 분배를 의미하기 때문에 개개의 자원을 효과적으로 사용하여 전반적인 수행도 향상을 꾀하는데 그 의미가 있다 할 수 있다. 네트워크은 범위에 있어 근거리 통신망(LocalArea Network) , 도시지역통신망 (Metropolitan Area Network), 원거리 통신망(Wire Area Network) 등이 있다. 최근 광케이블에 의한 초고속 통신망은 멀티미디어 데이터 등의 대량 데이터 전송에 매우 효율적인 네트워크로 특히, 동화상의 실시간 처리를 위해서는 더욱 그렇다. 앞에서 언급한 데이터 의 입출력 수행도 향상을 위한 I/O 블럭 크기가 중요한 것 처럼 네트워크에 있어서도 한번에 이동되는 데이터 의 크기도 매우 중요한 요소이므로 신중이 고려해야 한다.

4. 적용 사례

일반 PC(DOS 운영체제)6대와 워크스테이션(Workstation :UNIX 운영체제)2대를 네트워크로 구성하여 분산처리를 한다.



이때 한 PC를 분산을 위한 매핑 및 네트워크 경로배정(Router)를 위한 서버(Server)로 이용하며 각 클라이언트(Client)는 PC에서는 화상 및 음성 데이터를 입력하고 조회하는 시스템으로 구성한다. 데이터의 I/O 블럭 크기는 2048 Byte로 설정하였고, 한번에 통신할 수 있는 데이터 패킷(Packet)중에 데이터 크기 역시 2048 Byte로 고정하여 입출력 및 네트워크 통신의 효율을 높이고자 하였다. 아울러 동 영상 및 음성 등 데이터 의 크기에 따라 그 크기를 가변할 수 있다. GUI는 윈도우즈환경에서 Visual Basic과 Visual C++를 사용하여 Balance Tree 형태의 계층적 구조의 데이터베이스를 이용하여 이를 구현하였다.

5.결 론

네트워크를 기반으로 대용량의 멀티미디어 데이터 의 분산처리 시스템을 구현하고자 할 때 무엇보다 중요한 것은 대용량의 데이터를 여러 곳으로 분산이 필수 적이며 아울러 실시간 처리를 위하여 데이터 엑세스 타임이 빨라야 될 것 이다. 이러한 요건을 만족시키기 위하여 데이터베이스는 계층적 구조를 이용하였으며, 데이터 특성에 맞는 I/O 블럭 크기를 설정하고 최소의 입출력 횟수로 많은 양의 데이터 처리를 시도하고자 하였다. 아울러 Fragmentation의 최소화에도 역점을 두었으며, 분산처리는 안정적이고 빠른 속도의 네트워크를 요구하므로 데이터에 적합한 통신 크기를 결정하는 것도 중요하다. 고정 크기의 레코드나 필드의 사용을 배제하고 가변 크기를 채택하여 전반적인 저장용량 크기를 최적화하였고, 데이터 증가 시에 하드웨어 증설을 쉽게 하여 시스템 확장을 용이하도록 하였다. 아울러 멀티미디어 특성에 맞도록 GUI(Graphic User Interface)부분도 강화하여 데이터의 재생을 효과적으로 하였다. 이와 같은 대안으로 초대형의 멀티미디어 데이터의 경우라도 효율적인 System을 구축할 수 있다.

참 고 문 헌

[1] Korth, H.F. and Silberschatz, A., *Database System Concepts*
[2] Korth, H. F., (1983) *Database System Processing*
[3] Chu, W.W., *Optimal File Allocation in a Multiple Computer System*
[4] Casey, R.G., *Allocation of Copies of File in an Information Network*
[5] Koegel J., *Multimedia Systems*, ACM Press, Addison-Weseley Publishing Company, 1994
[6] Koegel, J., C., and Rutledge, J., 'Toolkits for Multimedia Interface Design', Xhibition '92. Vol. 2., pp275-285, 1992
[7] Hodges, M., and Sasnett, R. Multimedia Computing, Addison_Wesley, 1993
[8] InterSystem사 'DTM System Management'
[9] 조유근 *시스템 프로그래밍*
[10] SCOTT JAROL *Visual Basic Multimedia*