韓國地形空間情報學會論文集 第5卷 第2號 1997年 12月 pp. 193-205

클라이언트-서버 환경의 매핑 시스템 개발을 위한 복제 일관성 모델에 관한 연구

A Study on the Replication Consistency Model for the Mapping System on the Client-Server Environment

이 병 욱* 박 홍 기**
Lee Byung Wook Park Hong Gi

要 旨

대용량의 매핑 자료를 다수의 사용자들이 효율적으로 공유하는데는 클라이언트-서버 환경에서 복제 일관성을 위한 분산 모델 개발이 요구된다. 기존의 분산 모델은 각 사이트들의 사본간의 일관성을 강조했으나 GUI를 이용한 화면과 사본간의 일관성이 고려되지 못하여 매핑 시스템과 같은 장기 트랜잭션에는 적합하지 않다. 매핑자료의 특성상 분산 환경에서 트랜잭션들의 일관성을 유지하는데는 시간 지연이 많으므로 병행효율이 중요하다. 본 연구에서는 디스플레이 록을 이용하여 GUI 화면과 사본들 사이의 일관성을 지원한다. 매핑자료의 특성을 이용하여 낙관적 병행제어 기법과 일관성 모델을 개선하여, 처리효율을 향상하는 일관성 모델을 제시한다.

ABSTRACT

It is required for multi-users to share massive mapping data effectively that distributed data model in the Client-Server environment is developed for the replication consistency. The existing model is not effective to the long transaction just like a mapping system, since it does not account for consistency between GUI screen and database replications even though it emphasizes on the replication consistency. The performance of concurrency control is very important for those long transactions, especially the mapping systems. This model is to support consistency between GUI screen and replicas using display locks. It suggests consistency model improving process performance by modifying memory consistency model and optimistic concurrency control for mapping data's characteristics.

*경원대학교 전자계산학과 부교수 ** 경원대학교 토목공학과 부교수

1. 서 론

컴퓨터 하드웨어와 소프트웨어 기술과 통신 기술의 비약적인 발전은 분산 환경의 보급을 촉진하였으며, 지리정보 시스템에 대한 수요도 폭발적으로 증가하였다. 도로, 통신, 상하수도, 가스, 국방 등의 부문에서 지리정보관리의 수요는 이제까지의 독립적인 요구로부터 제반 관련분야들의 통합적인 관리를 요구하게 되었다. 따라서 제반분야에서 독립적으로 자료를 관리하던 방식에서 벗어나 여러 관리분야에서 다양한 관련자들이 통합된 자료를 사용하므로 자료 공유와 함께 매핑자료의 효율적인 관리가 중요한 문제로 되었다.

매핑자료는 특성상 기본도의 기반 위에 여러 주제들에 대한 개별적인 계층을 부가하여 전체적인 관리를 하게된다. 여러 분야의 사용자들은 관련 매핑자료에 대한 접근을 동시에 요청하고, 자료를 갱신하며, 공유자료의 일관성을 요구한다. 따라서 분산환경 병행제어 기구의 복잡도를 증가시키고 자료처리 효율을 저하시킨다. 이에 대한 대안은 하드웨어와 소프트웨어 등에서 여러 가지가 제시되고 있지만 주로 근거리/원거리 고속 통신망을 이용하여 클라이언트-서버 모델을 매핑시스템에 알맞게 설계하고, 분산 병행제어 기법을 개선하는 분산시스템 소프트웨어에 대한 것이다.

본 연구는 현재 국가GIS 프로젝트의 일환으로 매핑 시스템을 WindowsNT에서 단독 시스템으로 개발하고 있으나 차기 개발의 목표에는 분산환경에서 클라이언트-서버 모델을 구축하는 것이 예상된다. 따라서 본 연구에서는 차기 목표를 위해 클라이언트-서버 모델의 서버에 수치지도 자료를 저장하고, 다수의 클라이언트들이 사본을 복제하여 지역에서 접근, 갱신, 저장할 수 있는 제반 기능들을 분석하여 가장 병목현상을 일으키는 병행제어에 관한 기술을 개선하는 것이다. 이것은 복제 일관성 알고리즘의 도입으로 가능하지만 매핑자료의 특성에 맞는 복제 일관성 알고리즘을 동기화 및 메모리 일관성 이론을 도입하여 개선하고자 한다.

본 연구의 목적은 대형 매핑자료를 서버에 집중관리하면서 다수의 클라이언트들이 사본을 복제하여 효율적으로 자료를 가공할 수 있도록 복제 일관성, GUI를 이용하는 화면과 데이터베이스 사이의 GUI 일관성 그리고 병행처리 효율과 통신효율을 향상하는 분산 매핑시스템을 설계하고 구현하는 것이다.

클라이언트-서버 환경에서 매핑 시스템의 복제 일관성 알고리즘의 성능을 개선할 수 있는 모델을 설계하고, 부분적으로 이 모델을 구현하며, 모의 자료를 이용하여 성능을 평가하고자 한다. 모델 설계에서 개발이 요구되는 주요 요소들은 다음과 같다.

- 모델 설계
  - 분산 모델에 의한 매핑시스템 설계
  - GUI 일관성과 복제 일관성을 지원하는 병행제어
- 알고리즘 설계
  - 성능 향상
  - GUI 일관성 향상
  - 복제 일관성 향상
  - 분산 병행처리 효율 향상
  - 통신 효율 향상

본 논문에서는 매핑 시스템의 주요 구성요소들의 문제점들과 개선 가능성을 분석하여 통합된 일관성 모델을 설계하고, 제시된 통합 일관성 알고리즘의 복잡도를 기존 알고리즘과 비교 분석하여 일관성 수준과 전반적인 처리효율을 평가하였다.

2. 매핑 시스템의 구성

2.1 매핑 시스템 개요

· 매핑자료의 특성: 매핑 자료는 공간자료와 속성자료로 구성되는데 공간자료는 일반적으로 대용량이며 비정형인 것이 일반적이므로 처리효율이 중요하다. 실제로 1:5000 지도의 한 도엽의 DXF 파일의 크기는 8M 바이트 정도이다. 또한 파일의 형태는 지도의 실제 지형에 따라서 그리고 관리하고자 하는 중점관리 분야에 따라서 내용과 형식이 매우 상이한 것이 보통이다.

· 트랜잭션의 특성: 속성자료는 짧은 시간에 처리할 수 있지만 공간자료를 화면에 디스플레이할 때 디스플레이가 끝날 때까지 긴 시간을 요하므로 장기 트랜잭션에 해당한다. 또한 엔티티를 갱신하려면 화면, 메모리, 데이터베이스를 모두 접근해야 하므로 더욱 장기 트랜잭션이 된다.

· 병행성 : 공간 자료는 다수의 사용자가 동시에 동일 엔터티에 대해 갱신을 포함하는 접근을 수행할 가능성이 적으므로 낙관적 병행기법을 적용할 수 있다.

· 일관성 : 처리 시간이 짧고 실행주기가 빠르고 순차적인 접근이 요구되는 응용분야에는 강한 일관성이 요구되지만, 매핑자료 처리 시에는 장기 트랜잭션으로서 순차적인 접근의 요구가 적으므로 약한 일관성을 적용하여 처리 효율을 향상하는 것이 바람직하다.

서버 : 대용량 매핑자료의 저장관리 기능
: 복제 일관성 유지를 위한 병행제어 기능
: CPU 소모가 많은 수치계산 기능

클라이언트 : 서버의 매핑자료를 읽고, 화면과 프린터에 출력하는 기능
: 매핑자료를 갱신하고 서버에 갱신을 반영하는 기능

입력 : DXF, SDTS

파일 출력 : TIFF, DXF, SDTS

내부 파일구조 : 내부처리 효율 향상을 위하여 매핑자료에 적합한 자료구조로 설계한다.

화면 출력 : 매핑 자료는 용량이 크고 시간이 길기 때문에 장기 트랜잭션이다. 전형적인 트랜잭션은 너무 엄격해서 클라이언트들의 데이터베이스들과 일관성 유지가 어렵다. 따라서 매핑 자료의 GUI 일관성을 위해 사용하기 어려우므로 새로운 모델을 도입한다.

◎ 문제점과 개선 가능성

· 자료 분산의 곤란 : 매핑 자료는 대용량이므로 모든 클라이언트의 디스크에 복제하는 것은 비효율적이다. 필요한 부분(한 도엽 단위)을 지역 메모리에 복제하는 클라이언트 캐쉬 기법을 도입하는 것이 효율적이다.

· 일관성 유지 : 다수의 사본이 각 클라이언트 캐쉬에 복제되어 있으나 동일한 자료를 동시에 충돌 접근하는 경우는 적다. 따라서 GUI를 이용한 일관성 향상을 위해 서버에서 록을 관리하여 약한 일관성을 유지함으로써 처리효율을 향상한다.

2.2 분산처리 일관성 모델

2.2.1 분산처리 모델

(1) 계층구조에 의한 분류

· 원격 데이터베이스 접근 모델 : 2계층으로 구성된 모델로서 응용 프로그램이 클라이언트에 있으므로 자료 요청시 서버에서 대량의 자료 이동 가능성이 있으므로 효율이 낮을 가능성이 많다. 응용 프로그램 변경 시 다수의 클라이언트들에서도 변경이 요구된다.

· 데이터베이스 서버 모델 : 2계층으로 구성된 모델로서 응용 프로그램이 서버에 위치하므로 클라이언트로 결과만을 이동하므로 대량의 자료 이동 가능성이 적다.

· 다계층 모델 : 2계층 모델의 문제점을 개선하기 위해 클라이언트와 서버의 기능을 완전히 분리하고 중간에 미들웨어를 설치하여 유연성과 확장성을 개선한 모델이다. 대규모 분산환경에서 별도의 인터페이스 프로그램을 작성하지 않아도 미들웨어를 통하여 서버를 일관성있게 접근할 수 있다. 특히 부하 균등화(Load Balancing) 기능과 트랜잭션의 효율적인 처리가 가능하다는 장점이 있다. 그러나 기술적으로 구현이 어렵고 개발시 비용이 많이 드는 단점이 있다.

(2) 자료접근 형식에 의한 분류

· 직접접근 방식 : 클라이언트는 지역에 자료를 저장하지 않으므로 서버의 자료를 직접 접근한다. 지역에서는 서버의 자료를 메모리로 이동하여 메모리에서 작업한 후에 결과는 서버에 저장한다. 읽기 접근 시에도 서버에서 자료를 이동해와야 하는 부담이 있다.

· 간접접근 방식: 클라이언트는 지역에도 자료를 저

장하므로 읽기 접근 비율이 많을수록 효과가 좋다. 그러나 쓰기나 자료 갱신 시에는 지역 자료와 서버자료의 일관성을 유지해야 하는 부담이 있다.

◎ 매핑 시스템의 적용

다계층 모델은 분산 트랜잭션 처리의 효율상 장점이 있으나 비정형 화된 자료의 공유처리에는 아직 문제점이 많다. 따라서 데이터베이스 서버 모델에서 비정형 화된 매핑자료를 분산 처리하는 것이 효과적이다.

직접접근 방식은 서버와의 동기화 및 일관성에 대한 문제가 적어서 효과적이지만 자료의 변동이 적거나 읽기 접근이 많은 경우에는 부적합하다. 매핑자료의 경우 서버에서만 자료를 갱신하고 클라이언트에서는 주로 읽기 접근을 많이 사용한다고 가정하면 간접접근 방식이 효과적이다. 따라서 매핑기술에 적합한 모델은 데이터서버 모델과 간접접근 방식을 조화 있게 통합하는 방법이다.

2.2.2 일관성 수준

분산 환경에서의 메모리 일관성 모델은 읽기 연산이 다른 사이트에서 최근에 갱신한 값을 반영하는지 여부에 따라서 일관성 수준을 분류한다. 즉 읽기 연산이 최근 값을 정확하게 반영하면 강한 일관성이고, 갱신한 값을 천천히 반영하면 약한 일관성이다. 그림 2.1에서 보는 바와 같이 사이트 1에서 갱신한 X값 1을 사이트 2에서 1로 읽으면 강한 일관성이고, 0으로 읽지만 나중에는 1로 읽을 수 있으면 약한 일관성이다. 따라서 강한 일관성과 약한 일관성 사이에도 시간적인 간격에 따라서 다양한 일관성으로 분류된다. 이렇게 일관성 수준이 다양하게 되는 이유는 다음과 같이 메모리 전송 시간보다 메모리 접근 시간이 크기 때문이다. 강한 일관성을 유지하기 위해서는 각 사이트의 (최소 읽기 시간 + 최소 쓰기 시간) < 메시지 전송 시간 사본들을 갱신한 후에 원본을 갱신하는 등의 분산처리기법을 적용해야 하므로 처리효율이 저하된다. 따라서 시급하게 강한 일관성을 유지하지 않아도 되는 응용에서는 약한 일관성의 개발이 요구된다.

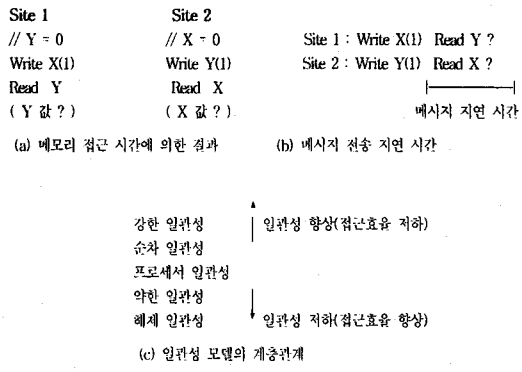

그림 2.1 메시지 지연 이론

그림 2.1(b)에서와 보는바와 같이 메시지 지연 시간 이전에 읽는 경우에는 최근 값을 반영하지 못하지만 이후에 읽으면 최근 값을 반영할 수 있다. 그림 2.1(c)는 일관성의 종류와 함께 일관성의 수준과 효율간의 관계를 보여준다. 따라서 응용이 허용하는 범위 내에서 가급적 일관성의 수준을 약하게 하는 것이 효율적이다.

2.2.3 복제 일관성 모델

분산환경에서 자료의 가용성을 높이기 위해 각 클라이언트에 사본들을 복제하면 통신효율을 향상할 수 있다. 그러나 자료 갱신시 한 사이트의 자료만 갱신하면 다른 사이트의 자료들과 일관성이 일치되지 않으므로 문제가 된다. 따라서 자료 갱신 시에는 모든 사본들을 동시에 갱신해야 하는 부담이 있다. 이러한 부담을 줄이면서 각 사본들이 항상 같은 값으로 읽혀지도록 하는 것을 복제 일관성이라고 한다.

다수의 사본들의 일관성을 유지하기 위해서 그리고 처리효율을 개선하기 위해 적합한 병행제어 기법을 적용하고 개선하는 것이 중요하다. 병행제어 기법의 종류에는 비관적 병행제어기법과 낙관적 병행제어기법이 있다. 전자는 충돌연산이 많을 때, 후자는 충돌연산이 적을 때 주로 적용된다.

일관성 유지에도 자료접근의 순서를 엄격하게 요구하는 강한 일관성과 접근순서가 엄격하게 요구되지 않는 약한 일관성이 있으며 그 사이에 일관성의 요구정도에 따라 다양한 일관성들이 있다. 금융기관의 여

수신 업무용 트랜잭션들은 엄격한 접근 순서가 지켜져야 하므로 강한 일관성이 요구되지만 매핑자료는 이미지 자료를 접근하고 갱신하는 것이므로 특히 읽기 전용의 접근이 많으므로 약한 일관성이 적합하다.

분산환경에서 복제 일관성을 해결하는 알고리즘은 다양하지만 가장 대표적인 낙관적 병행제어 기법들은 다음과 같다.

1) 통지 록(Notify) 알고리즘

통지록 알고리즘은 클라이언트가 서버에게 자료를 요청하고, 해당 객체와 메시지 순서 번호를 전송한다. 클라이언트는 트랜잭션을 낙관적으로 수행한 다음에 완료(commit) 요청과 갱신한 값 그리고 메시지 순서 번호를 서버에게 전송한다. 서버는 메시지 순서번호를 이용하여 록의 충돌 여부를 검사한 후 완료/포기 메시지를 클라이언트에게 보낸다. 서버가 갱신을 완료할 수 있으면 이에 영향을받는 모든 클라이언트들에게 통지 메시지를 보내서 모든 사본들을 갱신하는 것이 통지록 알고리즘이다.

서버는 공유 록과 배타 록을 관리하며, 메시지 순서 번호를 이용한 버전관리 개념을 도입하여 일관성을 유지한다. 즉 서버에서 록 충돌이 없어도 메시지 순서 번호가 다르면 클라이언트에 재검사를 요청함으로써 같은 버전인지를 확인한다. 낙관적 병행제어 기법을 사용하기 때문에 읽기/쓰기 비율이 낮으면 록 충돌로 인해 포기되는 트랜잭션이 많아지는 단점이 있다. 서버가 통지 메시지와 갱신된 객체의 새로운 값을 함께 전달할 때, 갱신에 영향을 받는 사본에 대하여 디스크 입출력의 부담이 있다. 그림 2.2의 통지 록 알고리즘

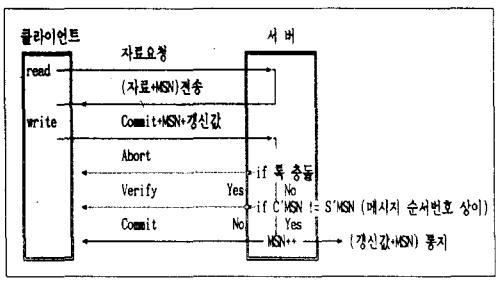

그림 2.2 통지 록 알고리즘

에서는 자료 요청과 Commit를 위한 4개의 메시지와 클라이언트 수만큼의 메시지가 필요하다.

2) 디스플레이 록(Display) 알고리즘

이 알고리즘은 객체를 디스플레이하는 중에도 갱신을 허가함으로써 GUI와 데이터베이스의 일관성을 효과적으로 유지하고 일관성 수준을 약화시켜 병행처리 효율을 개선한다. 그 이유는 그래픽 객체를 화면에 디스플레이하는데 시간이 많이 소요되기 때문에 긴 시간동안 록을 걸어둠으로써 야기되는 병행성 저하를 예방하는 것이다. 즉 공유 록(shared lock)을 디스플레이 록(display lock)으로 대치하여 객체가 디스플레이 되는 중에도 읽기와 갱신이 허용되도록 함으로써 일관성을 약하게 하였다. 그림 2.3에서와 같이 화면에 출력할 때는 디스플레이 록을 이용하지만 자료를 갱신할 때는 미리 배타 록을 얻어야 하므로 비관적 병행제어 기법을 도입하였다.

기존의 낙관적 병행제어 알고리즘들은 클라이언트에서 자료를 갱신하고 결과를 서버에 Commit 요청한다. 따라서 동기화 확인을 위한 병행제어 과정을 나중에 처리하기 때문에 캐쉬 록 또는 통지 록 알고리즘과 같이 읽기/쓰기 비율이 낮으면, 충돌 연산이 많아져서 포기되는 트랜잭션의 수가 많아지는 단점이 있다. 그러나 록의 충돌 가능성이 적은 시스템에서는 서버에서 병행제어를 수행하지 않고 클라이언트에서 자료들을 갱신하기 때문에 갱신 처리가 신속히 반영되

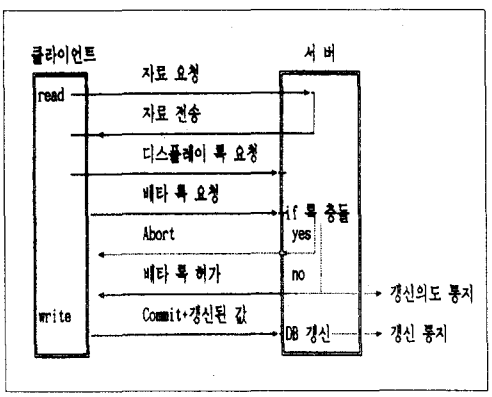

그림 2.3 디스플레이 록 알고리즘

는 장점이 있다.

콜백(Call Back) 알고리즘과 같은 비관적 병행제어 기법들은 록을 획득한 후에 자료를 갱신하므로 트랜잭션이 취소될 확률이 적다. 디스플레이 알고리즘은 GUI 일관성을 얻기 위해 디스플레이 록을 도입하였으며, 트랜잭션의 취소를 줄이기 위하여 비관적 병행제어 기법을 도입한 것이다.

· 매핑 시스템의 적용

매핑 자료는 수치지도관리의 특성상 지도를 도엽단위로 처리하며 여러 클라이언트들이 동시에 동일한 도엽을 읽고 화면에 출력하는 경우가 많다. 그러나 매핑 시스템에서는 여러 개의 자료 계층이 존재하고 각 계층은 상이한 응용분야에 적용되기 때문에 도엽단위로 읽고 접근하는 경우는 많이 발생할 수 있으나, 동일한 도엽의 동일한 계층에서 읽기/쓰기 및 쓰기/쓰기 간의 접근이 빈번하여 충돌 연산이 발생하는 경우는 많지 않을 것이다. 따라서 통지 록과 같은 낙관적 병행제어 기법을 적용하는 것이 바람직하다.

매핑 자료처리는 long 트랜잭션에 해당하기 때문에 디스플레이와 Database 간에 적절한 수준의 일관성 선정이 요구되며, GUI의 활용이 필수적인 추세이므로 GUI 일관성을 위해 디스플레이 록의 도입이 요구된다.

3. 복제 일관성 모델

3.1 시스템 구성과 지원 환경

3.1.1 시스템 구성과 기능

본 연구에서의 시스템 환경은 그림 3.1(a)와 같이 고속 근거리 통신망에 의한 클라이언트-서버 모델을 전제로 하며, 매핑 자료의 특성상 용량이 방대하므로 서버에서 데이터베이스를 집중 관리한다. 각 클라이언트는 그림 3.1(b)에서와 같이 네트워크 관리자를 이용하여 객체를 요구하고 화면에 디스플레이하고 갱신하며, 필요하면 자신의 데이터베이스에 저장할 수 있다. 클라이언트에서는 필요한 객체들의 사본을 부분적으로 전송 받아서 처리한다. 이 시스템에서의 각 구성원의 기능은 아래와 같이 정의한다.

· 서버 : 매핑자료의 저장하고 데이터베이스를 집중 관리한다. 각 클라이언트에 있는 사본들의 현황을 파악하고 병행제어 기능을 수행한다.

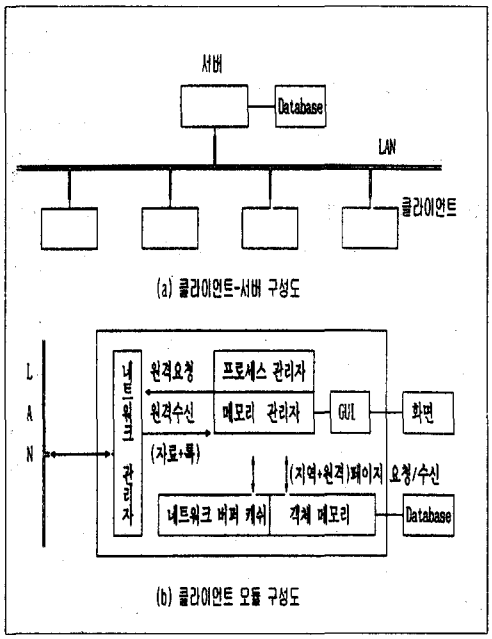

그림 3.1 제안 시스템 구성도

· 클라이언트 : 서버로부터 매핑자료 사본을 전송 받아서 한시적으로 저장장치에 저장하며, 사본을 읽거나 갱신하고 서버에 갱신 사항을 보고한다.

· 일관성 수준 : 매핑 시스템은 자료의 특성상 수치계산이 적고 이미지 자료의 접근이 주된 내용이므로 강한 일관성의 필요성이 적다. 따라서 효율 향상을 위해 약한 일관성을 적용한다. 즉 매핑자료 갱신 시 서버는 multicast로 클라이언트들에게 갱신을 예고하고 Commit되면 갱신값을 전송하여 복제 본들을 서버의 사본과 일치시킨다. 일관성의 내용은 다음과 같이 MRMW 대신에 MRSW를 적용한다. GUI 일관성을 유지하기 위해 디스플레이 록을 도입하여 객체를 화면에 출력하는 동안에도 객체를 갱신할 수 있도록 하

며, 화면과 데이터베이스의 일관성을 강화한다.

MRMW(Multiple Read Multiple Write) : 모든 클라이언트에서 매핑자료를 요청하고 읽을 수 있으며 모든 클라이언트에서 사본의 갱신을 통하여 모든 사본들을 갱신할 수 있다. 갱신 자체는 서버부터 수행하지만 모든 클라이언트로 전파된다. 서버에서 록의 충돌이 발생하면 갱신 자체가 취소된다.

ROWA(Read One Write All) : 클라이언트들은 자신의 사본을 읽지만 갱신 시에는 서버에 갱신값을 전달하여 원본을 갱신하고, 모든 사본들을 갱신한다.

병행제어 기구 : 매핑 자료의 특성상 록 충돌이 많지 않을 것이므로 낙관적 병행제어 기법을 적용한다. 록 테이블과 록킹 프로토콜을 서버에서 집중관리하여, 자료전송과 동기화를 서버에서 통합 관리한다. 클라이언트에서 자료 갱신이 끝나면 서버에게 Commit를 요청하고, Commit되면 복제 본을 가진 클라이언트들에게 갱신값을 통지한다.

3.1.2 시스템 지원 환경

앞 절에서 시스템을 구성하는 각 모듈들이 정상적으로 실행되기 위해서는 처리단위, 버전관리, 내부자료구조 등을 설계해야 한다. 따라서 이들에 대한 기능을 다음과 같이 기술한다.

1) Granurarity

매핑 자료를 디스플레이하는 단위는 화면에 출력되는 도엽을 기본 단위로 하지만, 객체를 수정하는 경우에는 매핑 자료 객체를 구성하는 entity를 기본 단위로 한다.

수치지도 자료의 크기와 내용은 자료의 특성상 정형화되어 있지 않으므로 레코드 또는 페이지로 한정하기 곤란하다. 벡터 자료를 내부 자료구조로 변환하여 기억된 자료 중에서 엔티티 단위로 록을 관리한다. 따라서 록 테이블도 엔티티 단위로 관리하며, 자료전송과 무효화도 엔티티 단위로 관리한다.

2) 버전 관리

매핑자료는 자료의 오류를 수정하기 위한 갱신 이외에도 좌표변환, 투영변환 등의 자료의 변환이 있으며, 도면 접합을 수행하는 과정에서도 오차를 줄이기 위한 자료의 갱신이 요구된다. 그러나 좌표변환과 투영변환은 여러 가지 목적과 용도에 따르는 방법들이 다양하기 때문에 일반적인 데이터베이스처럼 직접 자료파일에 갱신을 할 수 없고 변환된 결과들을 분석 비교하는 과정이 요구된다. 따라서 갱신되는 자료마다 버전을 관리해야 하므로, 갱신 시마다 새로운 버전을 만들어 저장 관리한다.

3) 내부 자료구조

AutoCad 프로그램에서 내부 처리를 위해 DWG 파일을 사용하듯이 각 CAD, GIS 프로그램에는 내부 처리를 위해 내부 용도의 파일이 있다. AutoCad, Arc/Info, ER Mapper 등과 같은 프로그램에도 내부 자료구조를 사용하고 있으나 범용의 목적으로 개발된 것이므로 매핑 시스템만을 위한 자료구조로서는 매우 복잡하고 방대하여 처리효율에 문제가 있다. 따라서 매핑 전용으로 사용되는 신속한 처리가 가능한 간단한 자료구조의 개발이 요구된다.

매핑 시스템이 신속하게 처리되기 위해서는 검색과 수정에 효율적인 자료구조를 갖추어야 한다. 각 레이어를 구분하고 다양한 객체들을 신속하게 탐색하기 위해서는 포인터를 이용한 리스트 구조를 구성하고, 엔티티들을 신속하게 접근하기 위해서는 2분 탐색이 가능하도록 배열을 병용하여 설계하였다.

3.2 복제 일관성 알고리즘

클라이언트는 자료 요청시에 디스플레이 록을 함께 요청하고, 서버는 록 테이블에 이를 기억하고 자료와 록을 함께 전송한다. 클라이언트에서 자료를 갱신하면 서버에게 commit 요청과 갱신된 자료 값을 함께 전송한다. 서버는 록 테이블을 이용하여 록 충돌이 있으면 클라이언트에게 abort 메시지를 전송하고, 충돌이 없으면 commit 메시지를 전송하고 데이터베이스를 갱신하며 복제 본을 가진 다른 클라이언트들에게 갱신값을 통지한다. 그림 3.2에서 전반적인 처리절차를 기술하였다.

이 알고리즘은 통지 록 알고리즘의 통신 효율성과

디스플레이 알고리즘의 GUI 일관성 향상의 장점을 활용하여 효율적인 GUI 일관성 알고리즘을 설계하는 것이다. 이 알고리즘을 요소별로 구분하여 기능을 다음과 같이 설계하였다.

1) 통신 효율과 GUI 일관성

록의 충돌이 많은 응용분야에서 사용되는 콜백(Call Back) 알고리즘과 같은 비관적 병행제어 기법들

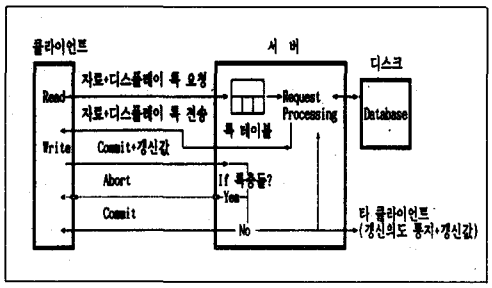

그림 3.2 복제 일관성 알고리즘의 처리절차

은 록을 요청하고 록의 해제를 통지하는 메시지 부담이 많은 단점이 있다. 매핑자료와 같이 록의 충돌이 적은 응용분야에서는 통지록과 같은 낙관적 병행제어 기법을 사용함으로써 메시지 부담을 감소할 수 있다. 디스플레이되는 시간이 긴 매핑 시스템에서는 화면과 데이터베이스의 일관성을 향상하기 위해 공유 록 대신에 갱신을 허용하는 디스플레이 록을 사용한다. 따라서 디스플레이되고 있는 객체를 갱신하려면 서버에서 록 충돌이 없으면 갱신의도를 미리 각 복제 본이 있는 클라이언트에 통지하고 시간적인 여유를 가지고 갱신된 자료를 전송한다.

2) 낙관적 병행제어 알고리즘

클라이언트에서 객체를 갱신하고 서버에게 갱신값과 함께 comit를 요청하면, 서버는 갱신값을 버퍼에 반영한 다음에(read phase) 록의 충돌 여부를 검사한다(validation phase). 검사에서 통과되면 서버는 데이터베이스를 갱신하고(write phase) 클라이언트에게 commit를 통보하며 사본을 가진 클라이언트들에게 갱신값을 전송하여 일관성을 유지한다.

낙관적 병행제어기법에는 트랜잭션과 자료의 TS(Time Stamp)를 이용하는 방법과 트랜잭션의 TS만을 이용하는 기법이 있으나, 전자는 자료마다 TS를 유지해야하므로 데이터베이스에 부담이 가기 때문에 본 연구에서는 트랜잭션의 TS만을 이용하는 알고리즘을 채택하였다.

그림 3.3에서와 같이 각 트랜잭션은 지역 변수로서

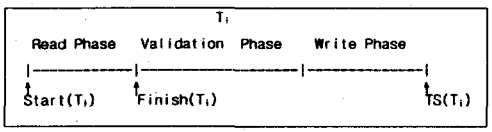

그림 3.3 충돌 검사를 위한 트랜잭션 T_i의 Time Stamp

Start, Finish, TS에 트랜잭션 T_i의 시작 시간, ReadPhase 완료 시간, 트랜잭션 T_i의 완료 시간을 기억하고 표 3.1과 같은 조건에 따라서 트랜잭션의 충돌 여부를 검사한다. 즉 조건 1은 트랜잭션 T_i가 완료된 후에 T_j가 실행되므로 충돌 연산이 아니다. 조건 2는 트랜잭션 T_j의 Read Phase 동안에 선행 트랜잭션 T_i가 종료되므로, T_i의 Write-set과 T_j의 Read-set이 중첩되지 않으면 충돌이 아니다. 조건 3은 트랜잭션 T_i와 후행 트랜잭션 T_j가 Read Phase에서 중첩되므로 T_i의 Write-set이 T_j의 Write-set이나 Read-set과 중첩되지 않아야 충돌 연산이 아님을 알 수 있다.

클라이언트-서버 환경에서 트랜잭션 T_{ij}(사이트 j에서 실행되는 트랜잭션 T_i)에 적용할 충돌 검사는 매핑 자료가 저장된 서버에서만 실행한다. 따라서 매핑 시스템에서는 중앙집중식 환경의 알고리즘을 그대로 사용할 수 있다.

표 3.2에서 보는바와 같이 클라이언트에서 디스플레이를 위해 서버에게 자료를 요청하면 서버는 상태에 관계없이 즉시 객체를 전송한다. 즉 록에 의해 트랜잭션들의 병행성이 저하되는 것을 막기 위해 서버는 디스플레이를 위해서는 항상 자료를 전송한다. 그러나 클라이언트에서 자료 갱신을 요청할 때는 서버의 상태가 디스플레이 록이면 객체를 즉시 전송하지만 배타 록일 때는 충돌 연산이므로 트랜잭션을 취소한다.

자료 갱신을 요청할 때 그리고 배타 록이 걸려있을 때 디스플레이 록을 요청하면 록 모드를 배타 록으로 한다. 따라서 록의 충돌은 배타 록이 중첩될 때만 발생한다.

그림 3.4는 표 3.1에서 정의한 충돌연산 검사 규칙에 의거하여 서버와 클라이언트의 병행제어 알고리즘을 기술한 것이다.

표3.1 Time Stamp에 의한 충돌 조건 연산표

조건 종류	조 건	내 용
조건 1	$TS(T_i) < Start(T_j)$	두 트랜잭션이 직렬로 처리
조건 2	$Start(T_j) < TS(T_i) < Finish(T_j)$ Write-set of $T_i \cap$ Read-set of $T_j = \emptyset$	전 트랜잭션이 끝나기 전에 후 트랜잭션이 시작될 때
조건 3	$Finish(T_i) < Finish(T_j)$ Write-set of $T_i \cap$ (Write-set of $T_j \cup$ Read-set of T_j) $= \emptyset$	후 트랜잭션의 Read Phase가 끝나기 전에 전 트랜잭션의 Read Phase가 끝났을 때

표3.2 록 호환 테이블

클라이언트 \ 서버	Display	Exclusive
Display	Y,Display	Y,Exclusive
Write	Y,Exclusive	N,Exclusive

Display : 자료를 디스플레이하기 위한 읽기 접근

Write : 자료를 새로 생성하거나 갱신하기 위한 쓰기

그림 3.4(b)에서 록의 검사는 표 3.2를 이용하여 배타 록이 중첩될 때만 충돌연산으로 간주하고 배타 록과 디스플레이 록이 중첩되면 록 상태만 배타 록으로 바꾸고 접근은 허용한다.

```
   Access()
// 서버에게 자료와 록을 요청하고 Database를 갱신함
     If memory fault
     then
// 원하는 자료가 클라이언트에 없을 때
       Send  Object  &  Display  Lock  Request
Message to server
// 매핑자료 객체와 디스플레이 록의 요청
     Receive  Object & Display Lock  from
       Server
     else;
// 원하는 자료가 클라이언트에 있을 때
     end-if
     if next Access = Write
// 충돌 가능성이 있는 연산인가?
     then
// 충돌 가능 연산
     Process Write Operation
// 쓰기 연산 수행
       Send  Commit & Updated  Value  to Server
// 갱신 Commit와 결과 값을 서버에 전송
       Receive Control Message from Server
// 서버로부터 제어 메시지 수신
       Send Acknowledgement  Message  to Server
       if Control Message = Abort
// 수신된 제어 메시지가 취소인가?
       then
// 서버에서 충돌 연산으로 확인
         Abort Transaction
       else
// 서버에서 정상 연산으로 확인
         Process Database Access Operation
     else;
// Read only Access
     end-if
     Finish
   End.
```

(a) 갱신을 요청한 클라이언트의 자료처리 루틴

```
   Validation ()
// 클라이언트에서 요청한 갱신 트랜잭션의 충돌연산 검사 루틴
   begin
// 충돌연산 검사 대상 : 트랜잭션 Tj
```

```
    if  TS(Ti) < Start(Tj)
// 조건 1의 경우 두 트랜잭션이 직렬로 처리될 때
    then  Pass() 처리
// 정상 확인
    else if  Start(Tj) < TS(Ti) < Finish(Tj)
// 조건 2의 경우
    then if Ti의Write-Set과 Tj의 Read-Set이 동일하고 록이 충돌되는가?
        then Conflict () 처리
// 충돌 확인
        else Pass() 처리
// 정상 확인
    else if Finish(Ti) < Finish(Tj)
//조건 3의 경우
        then  if  Ti의  Write-Set이  Tj의  Read-Set  또는 Write-Set과 동일하고 록이 충돌되는가?
            then Conflict () 처리
// 충돌 확인
            else Pass() 처리
// 정상 확인
    end-if
    Finish
  End
  Conflict ()
// 충돌연산시 요청 클라이언트에게 취소 지시.
 begin
   Send Abort Message  to Requesting  Client.
// 클라이언트에게 취소 메시지 전송
    Receive  Acknowledgement  Message from Requesting  Client.
    Finish
  End.
  Pass()
// 충돌연산이 아닌 경우에 데이터베이스를 갱신하고
  begin
// 사본을 가진 각 클라이언트들에게 갱신을 지시함.
    if Ti's action is write
// 쓰기 연산인가?
    then
// 쓰기 연산일 때
      Lock data
// 배타록 설정
      Update database
// Database 갱신
      Release lock
// 배타록 해제
      Send  Update  Message  to  All Clients(contains replicaion).
// 사본들의 갱신
      Receive  Acknowledgement  Message  from  All Clients.
    else;
// 읽기 전용이면 통과
    end-if
    Finish
  End.
```

(b) 서버의 충돌연산 검사 루틴

```
  Update()
// 클라이언트에서 사본 갱신하는 루틴
  begin
    Do until Update Message Received from Server
// 갱신 요청 메시지 대기
      Wait
    end-do
    Update Replica
// 사본을 갱신
  Send Acknowledgement  Message to Server
// 인지 메시지를 서버로 전송
    Finish
  End.
```

(c) 사본을 가진 클라이언트의 자료 갱신 루틴

그림 3.4 낙관적 병행제어 알고리즘

3.3 성능 평가

3.3.1 시스템 및 실험 환경

매핑 시스템의 환경은 WindowsNT가 가능한 Personal Computer 또는 Work Station 을 서버로 하였으며 클라이언트는 IBM PC로 하였다. 하드웨어는 서버와 클라이언트모두 아래와 같이 PC로 구성하였으며, 소프트웨어의 호환성 유지를 위해 마이크로소프트사의 Visual C++5.0을 사용하였다.

- 서버 : IBM PC/Pentium , 128M byte, 8giga HDD
  Windows-NT 4.
- 클라이언트 : IBM PC/Pentium , 64M byte, 2giga HDD
  Windows 95
- 개발 언어 : Visual C++ 5.0
- 네트워크 환경 : EtherNet 10Base5, 10M bps
- 입력 자료 : DXF Version 12, TIFF 파일

표 3.3 알고리즘별 메시지 수의 비교

연산 종류 / 알고리즘	최대 메시지 수		비 고
	읽 기	쓰 기	
디스플레이 록	3	6 + C	
제안 모델	2	4 + C	

표3.3에서 C는 사본을 가진 클라이언트의 수로서 디스플레이 록은 갱신 의도 통지와 갱신 통지 메시지 2개가 있으나 갱신 의도 통지는 전체 통신 시간을 연장시키지 않으므로 통신 효율 평가 시에는 하나의 메시지로 계산한다.

3.3.2 평가 분석

평가의 기준은 클라이언트-서버 시스템간의 통신 효율을 개선하는 것과 분산된 객체들 사이에 복제 일관성 그리고 GUI 화면과 자료 객체들 사이의 일관성 향상에 있다.

(1) 통신 효율

분산 시스템에서의 통신 효율은 일반적으로 읽기/쓰기 비율과 사본을 가진 클라이언트의 수에 의하여 영향을 받으며 효율 측정은 각 사이트간에 전송되는 메시지의 수를 기준으로 한다. 읽기/쓰기의 비율은 응용분야에 따라서 수배에서 수백배 범위에 있으나 본 연구에서는 3배에서 20배까지의 범위로 축소하여 효율을 측정하며 메시지의 수 Y는 정리(3.1)과 같이 행렬식으로 계산한다.

Y = [x (1-x)][Read # Write #] ----- 정리(3.1)

Y : 메시지 수
x : 읽기의 백분율
1- x : 쓰기의 백분율
x의 범위 : 읽기/쓰기 비율이 3배에서 20배의 경우
(0.75 < x < 0.9524)
Read # : 읽기 연산시의 메시지 수
Write # : 쓰기 연산시의 메시지 수 + C
C : 사본 갱신을 위한 메시지 수

그러므로 디스플레이 록 알고리즘의 메시지 함수 $Y_{display}$는 식(3.1)과 같다.

$$Y_d = [X \quad (1-X)][3 \quad 6+C]$$
$$= 6 - 3X + (1-X)*C \qquad (3.1)$$

제안 모델의 메시지 함수 $Y_{proposed}$는 식(3.2)와 같다.

$$Y_p = [X \quad (1-X)][2 \quad 4+C]$$
$$= 4 - 2X + (1-X)*C \qquad (3.2)$$

클라이언트의 수 O(C)는 국내에서 매핑 시스템을 사용하는 지방자치 단체의 경우를 고려할 때 1대에서 최대 8대로 간주하고 그 중간값 4를 취하면 평균적으로 동시에 접근하는 비율을 50%로 할 때 2대로 간주할 수 있다. 따라서 식(3.1)과 식(3.2)는 다음과 같이 변환할 수 있다.

$$Y_d = 8 - 5X \qquad (3.3)$$
$$Y_p = 6 - 4X \qquad (3.4)$$

이상의 두 식에서 X의 범위가 0.75에서 0.9524로 증가함에 따라(즉 읽기/쓰기 비율이 3배에서 20배로 증가함에 따라) 제안 모델의 효율은 29.4%에서 32.3%로 향상된다. 즉 읽기/쓰기 비율이 증가하면 제어용 메시지의 수가 상대적으로 감소하여 통신 효율이 향상된다.

클라이언트의 수를 매개변수로 효율을 측정하기 위해 읽기/쓰기 비율을 10배로 고정하면(X=0.909) 두 모델의 메
시지 함수는 식(3.1)과 식(3.2)로부터 다음과 같이 유도된다.

$$Y_d = 3.267 + 0.09 * C \qquad (3.5)$$
$$Y_p = 2.178 + 0.09 * C \qquad (3.6)$$

두 식에서 클라이언트의 수 C를 1대에서 8대로 증가시키면 통신 효율의 향상은 32.4%에서 27.3%로 변화한다. 즉 클라이언트의 수가 적을수록 갱신 메시지의 수가 감소하여 효율이 향상된다.

따라서 제안 모델의 전반적인 통신효율 향상은 평균적으로 30.35% 정도에 달한다

(2) 일관성과 병행성

제안 모델은 분산된 객체들 사이에 일관성을 약화하여 자료처리 효율을 향상한다. 즉 디스플레이 록으로 인하여 객체가 디스플레이되는 중에도 갱신이 가

능하게 하여 병행처리 효율이 증가한다. 특히 갱신되는 자료도 디스플레이하기 때문에 화면과 사본간의 일관성이 향상된다. 또한 낙관적 병행제어 기법을 이용하기 때문에 클라이언트에서 처리되는 데이터가 항상 최신의 상태를 유지하는 확률이 높아진다. 특히 매핑 자료의 성격상 쓰기는 서버에서만 실행하고 클라이언트에서는 읽기만 실행하므로 충돌 연산이 적어서 취소되는 트랜잭션의 비율이 낮을 것이다. 이것은 일관성 향상과 함께 병행처리 효율을 향상하게 한다.

4. 결론

컴퓨터와 통신기술의 비약적인 발전으로 클라이언트-서버 환경의 매핑 시스템 개발이 데스크탑 컴퓨터 환경에서 가능하며, 현실적으로도 현장의 수요가 급증하고 있다. 수요 확장 에 따라서 사용자 편의성 향상을 위한 GUI 활용이 필수적인 요소로 부각되고 있으므로 GUI와 데이터베이스간의 일관성과 사본들간의 복제 일관성이 중요한 해결 과제가 되고 있다. 이와 함께 대용량의 매핑 자료의 특성상 병행제어 효율을 향상해야 현실적 효용성을 제고할 수 있다고 본다.

기존의 통지록 알고리즘과 같은 낙관적 병행제어 기법들은 통신 효율은 좋으나 GUI 일관성이 결여되어 있으며, 디스플레이 록과 같은 알고리즘은 GUI 일관성은 좋으나 비관적 병행제어 기법을 이용하기 때문에 병행처리 효율이 낮아서 문제가 되고 있다. 본 연구의 목적은 통신 효율과 GUI 일관성을 함께 향상할 수 있는 일관성 모델을 설계하는 것이다. 따라서 본 연구의 핵심은 클라이언트들간의 사본에 대한 복제 일관성을 지원하기 위해 메모리 일관성 기법을 약화하는 것과 트랜잭션들의 동시 처리효율을 향상하기 위해 낙관적 병행제어 기법을 도입하고, GUI 화면과 데이터베이스 객체 사이의 일관성을 향상하는 것이다.

제안 모델은 디스플레이 록 알고리즘과 같이 GUI 일관성 기법을 지원하면서도 효율 면에서 통지록 알고리즘과 같은 처리 효율을 유지한다. 즉 디스플레이 록보다 30.35%의 처리효율을 향상한다. 이것은 통지록 알고리즘과 같은 낙관적 병행제어 기법을 사용하면서도 메모리 일관성을 약화하여 GUI 일관성을 지원함으로써 일관성과 병행처리 효율을 향상한 것이다. 이것이 가능한 이유는 매핑자료의 특성상 충돌 연산 가능성이 적기 때문이다.

장래 연구 방향은 자료 접근과 갱신의 가용성을 향상하는 것이다. 현재는 MRSW 방식으로 자료를 읽고 갱신하지만 앞으로는 클라이언트마다 데이터베이스를 설치하고 모든 클라이언트가 자신의 사본을 읽고 갱신하는 MRMW 기법을 개발하는 것이다.

감사의 글

본 연구는 1996년도 과학기술처 국가GIS기술개발에 관한 연구비 지원에 의해 이루어졌으며 이에 감사드립니다.

참 고 문 헌

1. 과학기술처, "지리정보시스템 활용방안", Dec. 1993.
2. R. Laurini, D. Thompson, "Fundamentals of Spatial Information Systems," Academic Press, 1992.
3. 상공부, "소규모 지도정보 시스템 기술개발에 관한 연구," *1차년도 중간 보고서*, Aug. 1993.
4. (주) 캐드랜드, 우정에스아이, (주)유신정보통신, "GIS DB용 데이터 포맷 변환도구개발," 한국통신 선로기술연구소, Dec. 1995.
5. S. Ceri, G. Pelagatti,"Distributed Databases Principles & Systems," pp227-237, McGraw-Hill, 1984.
6. S. Kelly, N. Roussopoulos, S. Baras, "Consistency and Performance of Concurrent Interactive Database Applications," *12th International Conf on Data Engineering, IEEE*, 1996.
7. Kevin Wilkinson, Marie-Anne Neimat, " Maintaining Consistency of Client-Cashed Data," *16th International Conf. on VLDB*, IEEE, 1990.
8. M. Carey, M. Frankline, M. Linvy and E.

Shekita"Data Caching Tradeoffs in Client Server DBMS Architectures," *ACM SIGMOD Conf.*, Denver June, pp 357-366, 1991.

9. M. Carey, M. Frankline, M. Linvy,"Local Disk Caching for Client-Server Database Systems," *19th International Conf. on VLDB,* IEEE, pp. 641-654, 1993.
10. D. Mosberger, "Memory Consistency Models," *Operating System Review,* Vol. 27, ACM Press, pp.18-26, 1993.
11. M. Stumn and S. Zhou, "Algorithms Implementing Distributed Shared Memory," *IEEE Computer,* Vol. 23, No. 5, pp. 54-63, May 1990.
12. K. Stathatos , S. Kelly, "Consistency and Performance of Concurrent Interactive Database Applications," *12th International Conf. on Data Engineering,* New Orleans, pp 602-609, Mar. 1996.