GIS 데이타베이스 구축을 위한 타일-기반 기하 데이타의 동시 합병 변경

이 상현*, 김 동현**, 홍 봉희**

*부산대학교 GIS학과, **부산대학교 컴퓨터공학과

Concurrent Merge Updates of Tile-based Geometry Data for Building a GIS DataBase

Sang-Hyun Lee*, Dong-Hyun Kim**, Bong-Hee Hong**

*Dept of GIS, Pusan National University, **Dept of Computer Engineering, Pusan National University

요 약

공간 데이터 수집과정을 통해 제작된 수치지도는 타일 기반의 교환 포맷으로 구성되어 있다. 이러한 타일 기반의 수치지도를 이용해서 공간 데이터베이스를 구축하기 위해 먼저 타일로 분리되어 있는 수치지도를 Seamless 수치지도로 재작성해야 하며 이를 위해 타일 합병 작업이 필요하다.

타일 합병 작업을 다수의 작업자가 동시에 수행할 경우, 작업의 선후 관계에 따라 작업을 잃어 버리거나(작업 손실(Lost-Work)) 또는 동일한 엔티티들에 대해 교차 잠금이 요청되어 교착상태(DeadLock)가 발생하는 등의 문제점들이 발생한다. 이러한 문제점들을 해결하기 위해 이 논문에서는 작업자 상호간의 메시지 교환을 통한 동시 합병 작업 알고리즘을 제시하고 그에 따른 수치지도 합병 처리기를 설계 및 구현한다.

1. 서론

수치지도는 측량 데이터, 항공사진, 인공위성 영상 등과 같은 원시 데이터를 해석 도화(Digital Mapping) 작업과 보완 측량 및 정위치 편집 작업을 통해 제작되며 하드웨어/소프트웨어적인 제약조건에 의해 수치지도는 많은 수의 타일로 구성된다.

공간 데이터베이스 구축을 위해 이러한 타일 기반의 수치지도를 이용할 경우에 먼저 Seamless 수치지도 제작을 위한 합병 작업이 선행되어야 한다[5]. 기존에는 각각의 수치지도에 대해 개별적으로 합병 작업을 수행한 후 합병된 수치지도를 반복적으로 합병하였다. 그러나 이러한 방법은 다수의 작업자에 의한 동시 작업이 전체 공정의 초기 일부만 가능하므로 작업의 생산성과 효율성이 점차 떨어지게 된다. 따라서 다량의 수치지도에 대한 반복적인 합병 작업 없이 다수의 작업자가 동시에 합병 작업을 할 수 있는 처리기에 대한 연구가 필요하다.

다수의 작업자가 수치지도에 대한 합병 작업을 동시에 진행할 경우에는 작업의 선후 관계에 따라 앞선 작업 내용을 잃어버리는 작업 손실(Lost-Work)이 발생할 수 있으며, 다수 작업자가 같은 엔티티들에 대해 서로 교차해서 잠금을 요청할 경우에 교착상태(Deadlock)가 발생할 수 있다는 문제점들이 나타난다[1] 이 문제점들을 해결하기 위해 [1]에서는 작업자 상호간의 메시지 교환을 통한 동시 합병 작업 프로토콜을 제안했다. 이 논문에서는 [1]에서 제안한 동시 합병 작업 프로토콜에 따른 세부 알고리즘을 제시하고 동시 합병 작업을 위한 수치지도 합병 처리기를 설계 및 구현한다. 수치지도 합병 처리기를 통하여 다수의 작업자들에 의한 작업의 동시성을 높일 수 있게 되기 때문에 수치지도에 대한 합병 작업의 생산성과 효율성을 높일 수 있게 된다.

이 논문의 구성은 다음과 같다. 2장에서는 관련연구를 기술하고, 3장에서는 동시 합병 작업 프로토콜에 따른 세부 알고리즘을 제시하며, 4장에서는 수치지도 합병 처리기의 설계 및 실제 구현화면을 보이며, 마지막으로 5장에서 결론 및 향후연구에 관하여 기술한다.

2. 관련연구

수치지도를 이용해서 공간 데이타베이스를 구축하는 기존의 상용 프로그램들이 이미 개발되어 실제 산업현장에서 사용되고 있다 그러나 기존의 프로그램들은 수치지도 한 매에 대한 작업을 한 사람의 작업자가 작업할 수 밖에 없도록 되어 있기 때문에 다수의 작업자가 동시에 작업을 하고자 할 경우에는 각각의 작업자가 따로 작업을 한 뒤 또 다른 작업자가 이들을 모아서 합병하는 작업을 해야만 한다. 이러한 작업방식은 인적, 물적, 시간적인 비용의 증대를 초래하게 되는 비효율적인 방식이다

[2]에서는 클라이언트-서버 환경에서 클라이언트 캐쉬의 일관성 유지와 변경 전파에 대한 알고리즘을 제안하고 있다. 특히, 다수의 클라이언트가 중복된 영역에 대한 수정 작업을 할 경우의 트랜잭션 모델을 위해 [3]에서 정의한 영역 잠금에 기반한 CS(Cached Shared)-잠금, CX(Cached eXclusive)-잠금 등의 확장된 잠금

기법을 제안하고 있다. 그러나, 작업공간 간의 경계선에서 만나는 엔티티들에 대해서는 영역 잠금 방식으로는 불필요한 영역까지 잠금을 설정해야 하는 불합리성이 발생하게 된다. 그러므로 서로 관련성이 없는 클라이언트들끼리 불필요한 메시지 통신을 하게 되며 그로 인한 오버헤드가 발생하게 된다.

[4]에서는 이 논문과 같이 클라이언트마다 독자적인 작업공간을 가지고 객체 변경을 하며 변경 전의 객체를 중복 저장하고 있는 다른 클라이언트에게 변경을 전파를 하는 알고리즘으로 "두 단계 델타 합병(two-phase delta-merge)"을 제안하고 있다. 그러나 이 알고리즘은 클라이언트가 객체를 수정할 때마다 변경 전파를 하기 때문에 잦은 객체 수정이 발생할 경우에는 메시지 과부하 현상이 나타나며 자신의 작업공간의 경계선과는 만나지만 자신의 작업공간 외부에 존재하는 객체에 대해서는 수정 권한을 가질 수 없다는 한계가 있다.

3. 동시 합병 작업 알고리즘

3.1 동시 합병 작업 프로토콜

이 논문에서는 작업자의 작업 영역을 설정하기 위해 작업공간(Work Space)을 정의한다. 작업공간은 수치지도 1매 이상의 집합으로 구성되며 하나의 작업공간은 전체 대상 지도를 구성하는 하나의 타일이 된다 각 작업자는 자신의 작업공간에 대한 엔티티만 수정할 수 있다고 가정하며 작업공간에 대한 작업자의 권한을 설정하기 위해 [3]에서 정의한 영역 잠금(region lock)을 사용한다.

클라이언트는 작업공간의 외곽 경계선에서 끝점이 서로 만나거나 근접해 있는 엔티티들에 대한 수정작업을 하고자 하는 경우에는 먼저 서버에게 동시 합병 작업 요청 메시지를 보내어 관련 클라이언트와의 합의에 의해 작업을 공유한다 그림1에서는 동시 합병 작업 프로토콜을 보이고 있다.

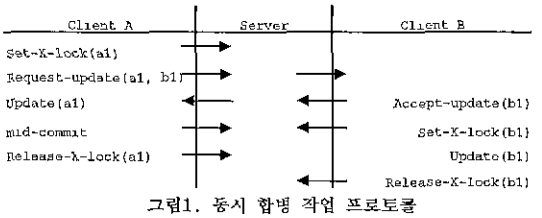


그림1. 동시 합병 작업 프로토콜

클라이언트 A가 자신의 작업공간에 포함된 엔티티인 a1을 클라이언트 B의 작업공간에 포함된 엔티티인 b1과 합병 시키고자 할 경우에 클라이언트 A는 클라이언트 B의 동의를 얻어야만 작업을 수행할 수 있다.

먼저 클라이언트 A가 서버로부터 경계선 작업 대상 엔티티에 대한 WRITE 잠금을 얻은 뒤 수정작업을 수행한 후 서버에게 동시 합병 작업을 요구하는 메시지(MSG_REQ_EDGEMATCH, MSG_REQ_MERGE)를 보낸다. 서버는 작업공간 정보를 참조하여 관련 클라이언트(클라이언트 B)에게 동시 합병 작업을 요청하는 메시지(MSG_REQ_EDGEMATCH, MSG_REQ_MERGE)를 보낸다. 클라이언트 B가 서버의 요청을 accept한 경우에는 요청 메시지에 포함된 수정 엔티티에 대한 정보를 이용하여 자신의 캐시를 수정하고 동시에 서버에게 작업 허가 메시지 (MSG_REP_OK)를 보낸다. 서버는 MSG_REP_OK 메시지를 클라이언트 A에게 전달하고 데이타에 대한 수정작업은 클라이언트 A의 MSG_REQ_UPDATE 메시지에 의해 수행된다. 클라이언트가 서버에게 동시 합병 작업 요청 메시지를 보낼 때와 서버가 관련 클라이언트에게 요청 메시지를 보낼 때에는 기하정보를 포함한 동시 합병 작업에 필요한 기타의 모든 정보를 보내지만 그 외의 경우에는 네트워크 통신 오버헤드를 줄이기 위해 메시지의 OID를 이용한다.

3.2 동시 합병 작업 메시지

동시 합병 작업을 위해 클라이언트와 서버간의 메시지는 아래와 같이 구분한다.

- MSG_REQ_EDGEMATCH – 서버에게 경계부합 작업 요청
- MSG_REQ_MERGE – 서버에게 엔티티 합병 작업 요청
- MSG_REQ_UPDATE – 서버가 관련 클라이언트에게 Update 통지
- MSG_REP_OK – 작업 수락/정상 작업
- MSG_REP_CANCEL – 작업 불허/비정상 작업
- MSG_REP_CANCEL_LOCK – Locking상태

또한, 이러한 메시지들과 함께 하나의 송수신 패킷에 포함되는 정보는 다음과 같다.

- 메시지 ID
- 수정 엔티티의 속성정보(레이어명)
- 수정 전의 엔티티 기하정보
- 수정 후의 엔티티 기하정보

3.3 세부 알고리즘

동시 합병 작업을 위해 클라이언트와 서버는 각각 작업을 수행한다. 그림2의 (가)와 (나)는 각각 클라이언트와 서버의 경우에 대해 동시 합병 작업을 위한 세부 알고리즘을 나타낸다. 클라이언트는 동시 합병 작업 요청 메시지를 서버에게 보내는 부분, 서버로부터 받는 두 부분으로 나누어지며, 서버로부터 전달되어오는 메시지의 경우 동시 합병 작업 요청 메시지와 캐시 일관성 유지를 위한 변경 요청 메시지로 나뉜다. 서버는 클라이언트가 요청하는 작업에 대해 응답하거나 작업 결과에 따라 다른 클라이언트들에게 처리 결과 메시지를 전파한다.

```
Client CoWork for SendMessage {
  Select Entities by user
  Send Request-Message to server
}

Client CoWork for ReceiveMessage {
  Get Reply from server
  Switch (Reply)
  {
    Case MSG_REP_OK :            Proceed Request Work
    Case MSG_REP_CALCEL :
    Case MSG_REP_CANCEL_LOCK : Cancel Work
    Case MSG_REQ_UPDATE :       Update Cache Map
  }
}
```

(가) 클라이언트 세부 알고리즘

```
Server CoWork for ReceiveMessage {
  Got Message from Client (Request or Related)
  Switch(Message){
    Case MSG_REQ_EDGEMATCH :
    Case MSG_REQ_MERGE :
        If (CheckLock is TRUE)
          Reply MSG_REP_CANCEL_LOCK to Request-Client
        Else {
          Search Related-Client
          Send Message to Related-Client
        }
```

```
Case MSG_REP_OK :
      Write Data to Log
      Search Request-Client
      Reply Message to Request-Client
  Case MSG_REP_CALCEL :
      Search Request-Client
      Reply Message to Request-Client
  Case MSG_REQ_UPDATE :
      Update Digital Map
      Search Prop-Clients
      Propagate Update Entities
      Write Data to Log
  }
}
```

(나) 서버 세부 알고리즘

그림2. 동시 합병 작업 프로토콜 세부 알고리즘

4. 수치지도 합병 처리기 설계

1.1 환경

구현을 위한 기본 환경은 그림 3에서 보듯이 Windows98을 클라이언트의 기본 OS로 하고 Windows NT를 서버의 기본 OS로 하여 Visual C++ 5.0을 이용하여 구현하였다. 수치지도는 국립지리원에서 제작한 서울특별시 1/5000 도엽 수치지도(DXF)를 사용하였다.

1.2 설계

협동 작업 알고리즘을 기반으로 한 시스템 구성도는 그림3과 같다.

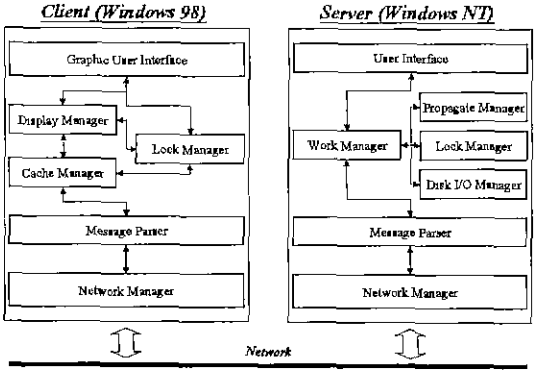
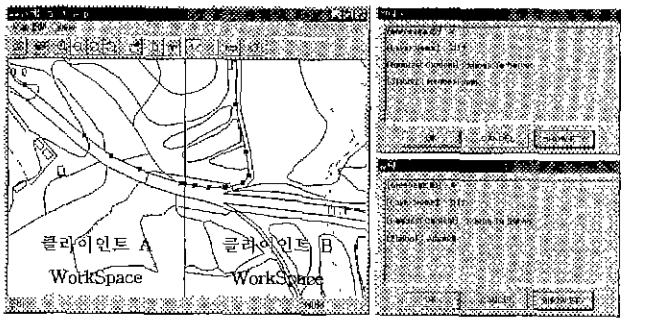


그림3. 시스템 구성도

그림3의 각 Manager는 다음과 같다.

- Graphic User Interface : 작업자와의 통신을 담당하며, 작업자가 요구하는 작업을 내부의 각 Manager들에게 전달한다.
- Display Manager : Cache Manager가 관리하고 있는 공간 데이타를 화면에 출력하고, GUI에 의해 변경되는 정보들과 서버로부터 전달되어 오는 정보들을 작업자에게 출력한다
- Cache Manager : 현재 클라이언트에 캐시되어 있는 수치지도의 공간 데이타를 관리한다.
- Message Parser : 클라이언트와 서버사이에 송수신되는 메시지를 Parsing한다.
- Lock Manager : 클라이언트의 영역 잠금 설정/해제 및 엔티티에 대한 WRITE 잠금 설정/해제를 담당한다.
- Work Manager : Lock요청, 동시 합병 작업 요청, 변경 전파, 데이타 요청 등 클라이언트에서 요구는 작업을 처리한다.
- Propagate Manager : 작업의 처리 결과에 따라 필린 클라이언트에게 처리 결과를 전파한다.

4.3 구현화면



(a) 클라이언트 A 측 화면 (b) 메시지 박스

그림 4. 구현화면

그림 4(a)는 동시 합병 작업의 대상 엔티티를 표시하며, (b)는 클라이언트 A가 클라이언트 B에게 그림 4(a)의 대상 엔티티들에 대한 합병 작업을 요청할 때의 메시지 박스와 클라이언트 B의 응답 메시지 박스를 각각 보여주고 있다.

5. 결론 및 향후 연구

이 논문에서는 타일 기반의 수치지도를 이용한 공간 데이터베이스 구축 시에 발생하는 작업손실, 교착상태 등의 문제점들을 해결하기 위해 작업자 상호간의 메시지 교환을 통한 동시 합병 작업 알고리즘을 제시하였다. 그리고 이 알고리즘을 적용한 수치지도 합병 처리기를 실제 구현함으로써 위 문제점들을 해결함과 동시에 동시 합병 알고리즘의 유효성을 보였다.

향후에는 구현 과정에서 고려하지 않았던 메시지 패킷의 최적화 및 메시지 트래픽의 최적화 전략 수립이 요구된다. 또한 이 논문에서는 클라이언트의 로컬 디스크에 존재하는 수치지도에 대한 데이터 수정이 아니라 단지 메모리 캐시 상의 데이터에 대한 교체만을 고려하였다. 따라서 디스크의 수치지도에 대한 캐시 교체 전략에 대한 연구가 필요하다.

참고문헌

[1] 이상현, 김동현, 홍봉희, "공간 데이타베이스 구축을 위한 수치지도 타일 합병 트랜잭션 모델", '99 개방형 지리 정보 시스템 학술회의 논문집 제2권 2호, pp149-156

[2]신영상, 최진오, 홍봉희, "클라이언트-서버 환경에서 캐쉬 공간 데이타의 변경 전파", 한국정보과학회 '99 봄학술발표논문집(B), 제26권 1호, pp 86-88.

[3]JinOhChoi, YoungSang Shin, BongHee Hong, "Update Propagation of Replicated Data in Distributed Spatial Databases", International Conference on DEXA 99, 1999

[4]AmSuk Oh, JinOh Choi, BongHee Hong, "An Incremental Update Propagation Scheme for a Cooperative Transaction Model", Preceedings of the International Conference on DEXA, 1996.

[5]부산시, "GIS를 이용한 도시정보관리 전산화 기본계획 연구", 1997.2