

미들웨어 기반의 분산 공간 데이터 변경 프로토콜의 설계 및 구현

정 민규⁰, 조 대수^{*}, 오 병우^{**}, 홍 봉희^{*}

^{*}부산대학교 컴퓨터공학과, ^{**}한국전자통신연구원

{mqchung, dsjo}@hyowon.pusan.ac.kr, bwoh@etri.re.kr, bhong@hyowon.pusan.ac.kr

The Design and Implementation of Update Protocol for Distributed Spatial Data based on Middleware

Min-Gyu Chung⁰, Dae-Soo Cho^{*}, Byung-Woo Oh^{**}, Bong-Hee Hong^{*}

^{*}Dept. of Computer Engineering, Pusan National University

^{**}Electronics and Telecommunication Research Institute

요 약

이 논문은 클라이언트에서 다수의 서버에 대한 공간 데이터의 변경 문제를 대상으로 한다. 분산 환경에서의 데이터 변경은 다음과 같은 요구 조건을 가진다. 첫째, 분산 환경에서는 서로 다른 구조의 클라이언트 및 서버에 대해 확장성을 고려해야 한다. 둘째, 공간 데이터 간에는 공간 관련성이 존재하기 때문에, 서로 다른 서버에 분산 저장된 공간 데이터간의 공간 관련성을 유지해야 한다. 셋째, 한 클라이언트에서의 변경은 여러 서버 및 다른 클라이언트와의 협동 작업을 통해 수행되어야 한다.

이 논문에서는 클라이언트에서의 공간 데이터 변경을 처리하기 위해 서비스 제공자를 설계, 구현한다. 서비스 제공자는 미들웨어 컴포넌트로서 동일한 인터페이스를 제공하기 때문에, 클라이언트 및 서버의 확장이 용이한 장점을 갖는다. 이 논문에서 제공하는 서비스 제공자는 2PC 기반의 변경 프로토콜에 따라 분산된 서버간에 존재하는 공간 관련성을 유지하면서 공간 데이터의 변경을 수행한다. 즉, 서비스 제공자는 클라이언트에서 변경을 요청한 객체에 대해 공간 관련성 여부를 검사하고, 공간 관련성을 가진 객체를 포함한 서버에 변경 내용을 전파한 후, 협동 작업을 통해 변경을 확정하거나 취소하게 된다.

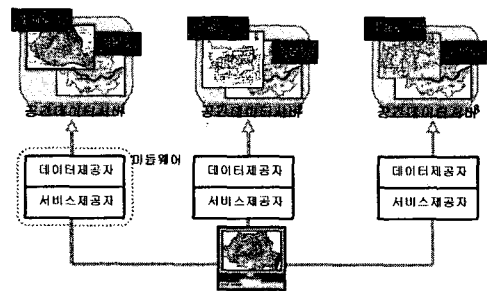
1. 서론

이 논문의 대상환경은 그림 1과 같다. 공간 데이터 서버는 분산된 주제도 데이터를 가진 오라클 공간 데이터베이스, 미들웨어는 오라클 데이터 제공자와 이 논문에서 제시하는 변경 프로토콜을 수행하는 서비스 제공자, 클라이언트는 미들웨어 인터페이스를 이용하는 변경 수행 모듈로 이루어져 있다. 이 환경에서 발생하는 문제들은 다수 서버에서 변경 영역 겹침(Overlap)이 가능하고, 제 2의 클라이언트가 다수 서버를 동시에 변경 가능하다는 가정으로 제한한다.

분산 공간 데이터에 대한 변경 프로토콜 수행의 주체인 미들웨어는 데이터 제공자와 서비스 제공자로 이루어진다. 데이터 제공자는 기존의 OLE DB 데이터 제공자를 이용하고, 서비스 제공자는 이 논문의 대상환경에 따라 새롭게 정의한다.

클라이언트에서 변경된 공간 데이터는 여러 분산된 서버 데이터와 공간 관련성을 가질 수 있다. 미들웨어 기반 환경에서는 서버 간의 통신은 고려하지 않으므로, 변경 데이터의 공간 관련성 검사를 위해 여러 분산된 서버에 대한 서비스 제공자에게 변경 전파할 필요성이 있다. 미들웨어 상의 공간 데이터 변경 전파 방식은 현재까지 제시된 바가 없다.

서비스 제공자를 통한 변경 전파에는 모든 변경 참여 서비스 제공자에게 전파하는 Broadcasting 방법과 일부 서비스 제공자에게 전파 하는



클라이언트 데이터 변경 시

그림 1 대상 환경

방법이 있다. 분산 공간 데이터 서버로의 접근에 따른 전역 트랜잭션은 독립된 중앙 트랜잭션 관리자가 조정하는 방법과 변경 주체 서비스 제공자가 조정하는 방법이 있다. 이 논문에서는 공간 데이터 변경을 공간 관련성에 기반하여 일부 서비스 제공자에게 전파하는 방법과 전역 트랜잭션에 대한 처리를 분리할 필요가 없는 변경 주체 서비스 제공자가 전역 트랜잭션을 조정하는 방법을 택한다. 이 논문에서는 변경 전파를 위해 서비스 제공자 인터페이스를 정의하고 대상 환경에 적합한 변경 전파 프로토콜을 설계하고 구현한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고, 3장에서는 미들웨어 기반의 공간 데이터 변경방법, 4장에서는 서비스

제공자 인터페이스 및 공간 데이터 변경 완료 프로토콜을 제시한다. 마지막으로 4장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

2.1 서비스 제공자

서비스 제공자는 OLE DB 구조 내의 중간자 형태로 존재한다. 즉, OLE DB 데이터 제공자에 대한 소비자로서, 다른 OLE DB 소비자에 대한 제공자이다. 서비스 제공자는 일반적으로 질의 처리 기능, 커서 기능, 분산 처리 기능 등을 수행한다.

이 논문에서 제안하는 서비스 제공자는 공간 데이터에 대한 변경 트랜잭션, 잠금 설정, 협동작업, 변경 수행을 위한 인터페이스 메소드를 제공한다.

2.2 분산 트랜잭션 관리

분산 데이터베이스 시스템에서의 트랜잭션은 지역 트랜잭션(Local Transaction)과 전역 트랜잭션(Global Transaction)으로 나뉜다[4]. 지역 트랜잭션은 트랜잭션이 초기화된 한 사이트(site) 내 데이터를 접근하고 각 데이터 서버에 의해 제어된다. 전역 트랜잭션은 트랜잭션이 초기화된 사이트와 다른 한 사이트 내 데이터 또는 여러 다른 사이트들 내 데이터를 접근하고 각 데이터 서버 자체적으로 제어될 수 없다.

[4]는 지역 트랜잭션들을 관리하는 전역 트랜잭션 관리자를 제시하고, 다중데이터베이스(Multidatabase) 시스템의 특징 및 변경 문제를 다루고 있다. 그러나, 서버로의 직접적인 통신이 없는 미들웨어 상의 공간 데이터 변경에는 적용하기 어렵다.

2.3 공간 관련성기반 2PC와 협동작업

[3]은 분산 공간 데이터 서버 간의 공간 관련성 기반의 변경 프로토콜과 협동 작업을 제시하였다. [3]의 변경 프로토콜에서는 영역 잠금과 분산 공간 관련성이 이용되었다. 영역 잠금은 사용자가 수정하고자 하는 특정 영역내의 객체들에 대하여 Shared 잠금을 미리 설정하고, 그 중 일부 객체들에 대한 Exclusive 잠금 요청을 허용하는 잠금이다. 분산 공간 관련성은 서로 다른 서버에 존재하는 두 공간 객체라 하더라도 위치 투명성 관점에서 존재하는 공간 관련성이다. 그러나, [3]의 환경은 분산 공간 데이터 서버 간의 중복 데이터만을 고려하고 있고, 클라이언트와 미들웨어를 고려하지 않았다.

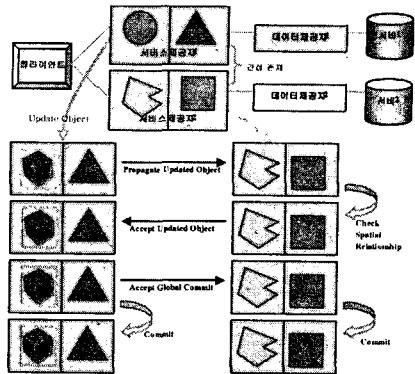


그림 2 미들웨어 기반의 공간 데이터 변경 방법

3.2 공간 관련성 검사 알고리즘

공간 관련성 검사 알고리즘은 다음과 같다.

```

METHOD ISpatialObjectUpdateService::CheckSpatialRelationship()
BEGIN
    PropagateObject: Object propagated at other Service Provider // 변경 전파된 공간 객체
    CurrentSP: Current Service Provider // 해당 서비스 제공자
    SRTable: SpatialRelationship Table // 공간 관련성 정보 테이블
    IF (Not Exist Update in CurrentSP)
        // 검사할 요청 받은 서비스 제공자에서 변경 존재
        IF (KeepSpatialRelationship(PropagatedObject, UpdateObject in CurrentSP))
            // 변경 전파된 객체와 변경되는 객체 간 공간 관련성 존재
            IF (KeepSpatialRelationship(PropagatedObject, Object in SRTable))
                // 공간관련성 정보 테이블 내 객체와 변경 전파된 객체 사이에 공간 관련성 존재
                Send Exist-Spatial-Relationship to CurrentSP
            // 공간관련성 정보 전달
        ELSE // 검사할 요청 받은 서비스 제공자에서 변경 존재
            IF (KeepSpatialRelationship(PropagatedObject, Object in SRTable))
                // 공간관련성 정보 테이블 내 객체와 변경 전파된 객체 사이에 공간 관련성 존재
                Send Exist-Spatial-Relationship to CurrentSP
            // 공간관련성 정보 알려줌
    END
    
```

3. 미들웨어 기반의 분산 공간 데이터 변경

3.1 미들웨어 기반의 공간 데이터 변경 방법

- 참여 서버: 클라이언트에서 변경한 객체들이 다수 서버에 분산되어 있을 때 공간관련성 제약 조건에 따라 발생하는 서브 트랜잭션의 대상이 되는 공간 데이터 서버
- 변경 주체 서비스 제공자: 참여 서버로 발생하는 서브 트랜잭션들을 관리하는 기능을 하는 서비스 제공자. 변경의 주체가 됨.
- 협동 작업: 변경 주체 서비스 제공자가 분산 서버의 공간관련성 제약 조건을 만족하기 위하여 변경 내용을 다른 서버의 미들웨어로 2PC형태로 전파하여 사용자의 승인을 받아내는 작업.

그림 2는 설계한 서비스 제공자가 수행하는 변경 프로토콜의 예를 보이고 있다. 이 때 분산된 공간 데이터에 대한 공간 관련성 검사의 필요성으로 서비스 제공자 간의 간섭이 발생한다. 간섭 해결을 위해 공간 데이터 변경을 전파하여 공간 관련성을 검사하고 참여 서버에서 변경을 승인하면 변경 트랜잭션 완료룰 수행하는 프로토콜을 제시한다.

4. 서비스 제공자 구현

4.1 시스템 구조

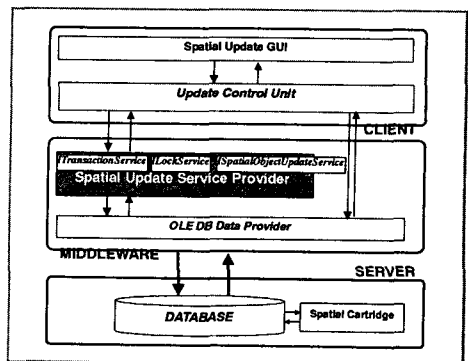


그림 3. 시스템 구조

그림 3은 이 논문에서 제시하는 시스템 구조를 나타낸다. 시스템 구조는 크게 클라이언트, 미들웨어, 서버의 세 부분으로 나뉜다. 클라이언트에서 사용자 인터페이스를 통해 변경을 수행하고 미들웨어의 서비스 제공자에게 변경 트랜잭션 수행을 요구한다. OLE DB 제공자의 인터페이스를 이용하여 서비스 제공자는 분산 공간 데이터 변경 프로토콜을 수행한다. 미들웨어에서 수행된 변경 프로토콜을 통해 분산 서버의 데이터는 공간관련성의 제약조건에 부합하는 변경을 수행한다.

4.2 서비스 제공자 인터페이스 메소드

그림 4는 이 논문에서 설계한 주요 인터페이스이다. 서비스 제공자의 ITransactionService, ILockService, ISpatialUpdateService 의 인터페이스 메소드들은 변경 트랜잭션 시작과 종료, 전역 트랜잭션 관리, 변경 전파, 공간관련성 검사, 잠금 처리 등의 서비스를 수행한다. 4.1에서 제시한 시스템 구조에 기반하여 분산 환경에서 각 서비스를 설계하였다.

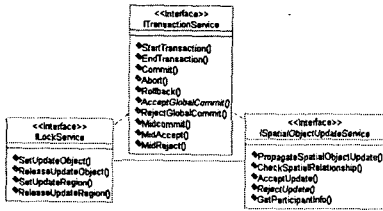


그림 4 서비스 제공자 인터페이스

4.3 공간 관련성 기반의 변경 프로토콜

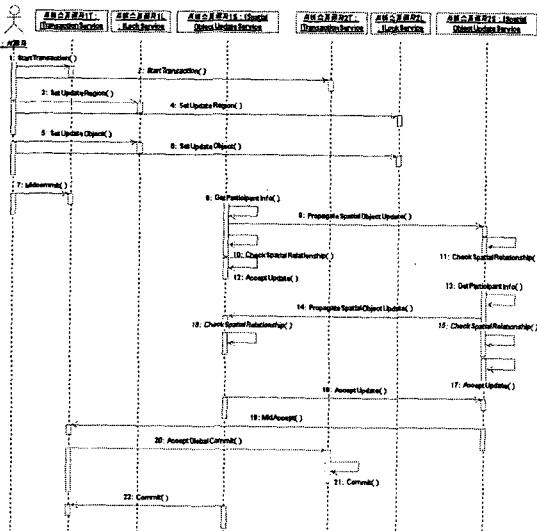


그림 5 변경 프로토콜

그림 5은 이 논문에서 제시하는 변경 프로토콜을 나타낸다. 프로토콜 처리 과정에 대한 설명은 다음과 같다.

1. StartTransaction : 변경 트랜잭션을 시작한다.
3. SetUpdateRegion : 변경할 영역을 설정한다.
5. SetUpdateObjects : 변경할 영역 내에서 변경할 객체들을 지정한다. 즉 Exclusive lock의 대상이 될 객체들을 선택한다.
7. MidCommit : 변경한 내용을 mid commit한다.
9. PropagateSpatialObjectUpdate : 변경한 내용을 다른 서버의 서비스 제공자에게 전파한다.

10. CheckSpatialRelationship : 공간 관련성 정보에 따라 변경 가능 유무를 검사한다.
12. AcceptUpdate : 변경에 대해 승인을 한다.
19. MidAccept : 전파된 변경 내용 또는 지역 트랜잭션을 승인한다.
20. AcceptGlobalCommit : 전역 트랜잭션을 승인한다.
21. Commit : 해당 서비스 제공자 트랜잭션을 commit한다.
22. EndTransaction : 변경 트랜잭션을 끝낸다.

4.4 구현

이 논문에서 구현한 서비스 제공자의 구현 환경은 Window NT환경에서 C++언어를 이용했으며, 라이브러리는 MFC를 사용하였다. 사용한 공간 데이터 서버는 공간 카트리지를 지원하는 DEC 유닉스 오라클 서버와 NT 오라클 서버이고, OLE DB 제공자는 오라클 OLE DB 제공자를 이용하였다.

그림 6은 공간 데이터 변경 구현을 보이고 있다. 오라클 데이터 제공자와 서비스 제공자를 이용하여 분산된 두 오라클 서버의 데이터 대해 변경 작업을 수행하였다.

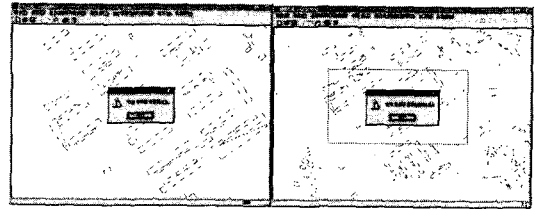


그림 6 변경 프로토콜 구현

5. 결론 및 향후 연구 방향

이 논문은 클라이언트에서 분산 공간 데이터 서버의 변경을 위한 서비스 제공자를 구현하였다. 서비스 제공자는 다음과 같은 기능을 제시하였다. 첫째, 변경 전파 기법을 도입하여, 공간관련성을 유지하였다. 둘째, 공간관련성에 기반한 변경 전파 완료 프로토콜을 제시하여 변경에 대한 분산 트랜잭션 처리를 하였다. 셋째, 서버 의존적인 재작성 문제를 해결하기 위해 서버의 모듈을 최소로 하여 미들웨어 상에서 분산 공간 데이터 변경 인터페이스를 구현하였다.

향후 연구로는 이 논문에서 제시한 변경 트랜잭션을 확장하여, 미들웨어 기반의 다수 클라이언트 트랜잭션과 서버 트랜잭션을 동시에 관리 가능한 병렬 수행에 대한 연구가 필요하다.

6. 참고 문헌

- [1] Open GIS Consortium, OpenGIS Simple Features Specification for OLE/COM Revision 1.1, 1999
- [2] Jose A.Blakeley, "Data Access for the Masses through OLE DB", Proc. Of the 1996 ACM SIGMOD, pp. 161-172, 1996
- [3] Yuri Breitbart, and Avi Silberschatz, "Multidatabase Update Issue", Proc. Of the 1988 ACM SIGMOD, pp. 135-141, 1988
- [4] Jin-oh Choi, Young-sang Shin, and Bong-hee Hong, "Update Propagation of Replicated Data in Distributed Spatial Databases", Proc. Of 10th International Conference on DEXA'99, pp. 952-963, 1999
- [5] Yuri Breitbart, Raghavan Komondoor, Rajeev Rastogi, S.Seshadri, Avi Silberschatz, "Update Propagation Protocols For Replicated Databases", Proc. Of the 1999 ACM SIGMOD, 1999