

클라이언트-서버 환경에서 공간 데이터 변경 트랜잭션의 회복 기법

박재관^o 김동현 홍봉희
부산대학교 컴퓨터공학과
{jckpack, pusover, bhhong}@hyowon.pusan.ac.kr

Recovery Scheme of Update Transactions of Spatial Data in Client-Server Environment

Jae-Kwan Park^o Dong-Hyun Kim Bong-Hee Hong
Dept. of Computer Engineering, Pusan National University

요 약

ISO TC211에서는 응용 스키마에서 공간 객체의 공간 관련성을 피쳐간의 위상 관련성으로 기술되어야 함을 규칙으로 정하고 있다. 따라서 공간 데이터가 변경되거나 변경 중 복구될 때도 객체간의 공간 관련성은 유지되어야 한다. 공간 관련성을 유지하기 위해서 기존의 회복 기법을 적용하면 수정의 결과가 다른 트랜잭션에 영향을 미치는 특징 때문에 많은 연쇄 복구가 발생한다. 따라서 공간 데이터 변경 트랜잭션의 회복에 적용하기 어렵다. 또한 협동 수정 트랜잭션의 회복 기법은 한 트랜잭션의 UNDO에 의해 모든 협동 트랜잭션이 연쇄 복구되는 문제가 있다.

본 논문에서는 클라이언트-서버 GIS에서 공간 관련성을 이용하여 관련된 트랜잭션을 선택적으로 복구하는 회복 기법을 제시한다. 그리고 서브-트랜잭션 단위의 UNDO를 위해 로그 레코드 구조를 제시하며 UNDO 결과를 효율적으로 전파하기 위한 기법을 제시한다.

1. 서론

지리 정보 운용에 대한 표준을 제시하고 있는 ISO TC211에서는 일 반 피쳐 모델(General Feature Model)을 정의하면서 피쳐간의 연관성을 4가지 그룹으로 분류하고 있다. 그 중 공간 관련성(Spatial Relationship)은 피쳐간의 위상 관련성을 명세 한다. 피쳐간의 공간 관련성이 있는 경우 응용 스키마에서는 공간 관련성이 위상 관련성으로 기술되어야 함을 규칙으로 정하고 있다. 따라서 공간 데이터가 변경될 때 공간 관련성은 유지되어야 하며 변경 중 복구에도 공간 관련성은 지켜져야 한다. 그러므로 공간 데이터 변경 트랜잭션의 회복 기법에 관한 연구가 필요하다.

기존의 관계형 데이터베이스에서의 회복 기법에서 사용하는 트랜잭션의 연쇄 복구는 데이터를 공유한 트랜잭션을 대상으로 한다[1,2,3]. 공간 데이터베이스에서 동시 수정하는 트랜잭션들은 대부분 공통의 데이터를 클라이언트 측에 가져와 수정을 하므로 이 기법을 적용하면 불필요한 연쇄 복구가 발생하는 심각한 문제가 있다. 협동 수정 트랜잭션의 회복 기법[4]은 지도의 오류를 방지하기 위해 동시 수정 영역에서 수정하고 있는 모든 트랜잭션을 연쇄 복구한다. 그러나 이 기법은 한 트랜잭션의 UNDO에 의해 모든 협동 트랜잭션이 연쇄 복구되며 점차 UNDO가 전파되는 문제가 있다.

본 논문에서는 클라이언트-서버 GIS에서 응용 스키마 설계자에 의해 정의된 객체의 공간 관련성을 이용하여 관련된 트랜잭션을 선택적으로 연쇄 복구하는 회복 기법을 제시한다. 그리고 서브-트랜잭션 단위의 UNDO를 위한 로그 레코드를 제시하며 트랜잭션의 UNDO 결과를

효율적으로 전파하는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구를 기술하고 3장에서는 공간 데이터 변경 트랜잭션의 회복 시 문제점과 회복 시 고려 사항을 설명한다. 4장에서는 공간 데이터 변경 트랜잭션의 회복 기법을 설명하며 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련연구

기존의 관계 데이터베이스에서는 긴 트랜잭션을 서브-트랜잭션으로 나누고 부분 복구로 회복 비용을 줄이는 방안이 제시되었다[1,2,3]. [1]은 기존의 단일 트랜잭션에서의 회복 기법을 서브-트랜잭션에 적용하기 위해서 로그-기반 복구(Log-Based Rollback) 알고리즘을 제시하였다. 그리고 [1]에서는 서브-트랜잭션의 특징을 정의하고 제시된 트랜잭션의 회복 기법에 대한 무결성을 확인하였다. 그러나 이러한 기법은 같은 데이터를 공유하면서 공간 데이터를 동시 수정하는 환경에 적용하면 많은 연쇄 복구가 발생하는 문제가 있다. 그리고 공간 관련성에 대한 고려가 없으므로 지도에 오류가 발생할 수 있다.

공간 데이터의 수정을 위해 공간 관련성을 고려한 방법이 제시되었다[4]. [4]는 지도 수정에서 협동 작업을 하는 모든 클라이언트들의 작업을 부분 복구함으로써 수정 오류의 여지를 제거하는 방법을 제시하였다. 그러나 이러한 방법은 UNDO하지 않아도 수정 오류가 전혀 발생하지 않는 서브-트랜잭션들까지 UNDO를 강요하는 문제가 있다. 그리고 비용의 지도 수정 작업은 수정의 취소를 최소화 해야 한다. 따라서 UNDO 발생시 반드시 연쇄 복구되어야 할 트랜잭션을 찾아내는 새

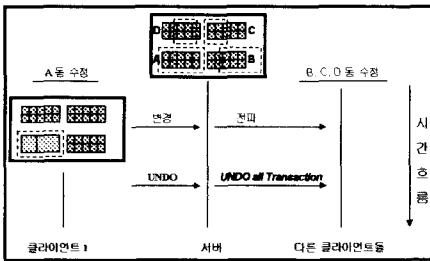
로운 방법이 필요하다.

3. 문제 정의

이 장에서는 공간 데이터 변경 트랜잭션의 회복 시 문제점에 대해 기술하고 회복 시 고려 사항에 대해 설명한다.

3.1 공간 데이터 변경 트랜잭션의 회복시 문제점

공간 데이터의 변경 트랜잭션은 효율성을 높이기 위해서 다른 트랜잭션에 의해 수정 중인 데이터의 읽기를 허용한다. 그러나 공유 데이터에 따라 연쇄 복구하는 것은 불필요한 다수 트랜잭션의 UNDO를 초래하므로 허용될 수 없다.



[그림 1] UNDO시 발생하는 문제점 : 불필요한 연쇄 복구 발생

[그림 1]에서 이러한 문제를 예를 들어 설명한다. [그림 1]은 A, B, C, D 4개의 아파트 등이 존재하고 있고 공사로 인해 지도를 변경하게 되는 시나리오다. 클라이언트1은 아파트 A동 지역을 위와 같이 2개의 상가로 지도를 변경하고 전파한다. 다른 트랜잭션들도 각각의 수정(아파트 B, C, D동)을 한다. 문제는 클라이언트1이 전파 후, 지도에 오류가 있어 UNDO를 하게 될 때 A동을 읽은 다른 독립적으로 수정 중인 트랜잭션도 UNDO하게 된다는 것이다.

미 완료된 데이터 읽기를 허용하는 지도 수정 환경에서 UNDO를 줄이는 새로운 방법이 제시되어야 한다. 이 논문에서는 UNDO를 최소한으로 줄이면서도 지도에 오류가 발생하지 않도록 하기 위해 공간 관련성에 의한 종속성에 따라 UNDO하는 방법을 제시한다.

3.2 회복 시 고려 사항

일반적인 회복 기법은 ACID의 기본 개념을 바탕으로 한다. 이런 기법에서는 회복 시 트랜잭션 전제가 복구되어야 한다. 그러나 지도를 수정하는 트랜잭션은 수 분에서 수 시간이 걸리는 작업이므로 전체 트랜잭션 작업을 취소하는 것은 비용이 크다. 이러한 전통적인 고 비용의 회복 방법 보다는 긴 트랜잭션을 서브-트랜잭션으로 나누어 단계적으로 회복하는 방법을 이용하여 비용을 줄일 수 있다. 따라서 서브-트랜잭션 단위로 UNDO하기 위한 로그 레코드가 제시되어야 한다. 그리고 수정 작업 중인 모든 클라이언트에게 UNDO의 결과를 전파해야 한다. 그러나 즉시 전파는 서버의 메시지 과부하 문제를 발생시키므로 이를 조절하는 방법이 필요하다. 이때는 즉시 전파와 지연 전파의 두 가지 방법을 적용해서 메시지 순간 부하를 줄이고 작업의 정확성도 유지할 수 있다.

4. 공간 데이터 변경 트랜잭션의 회복 기법

이 장에서는 지도의 무결성을 위해 UNDO되어야 할 서브-트랜잭션을 선택하는 방법을 기술하고 로그 레코드의 구조와 연쇄 복구의 예를 살펴본 후 시스템의 구성과 UNDO의 전파 방법에 대해 설명한다.

4.1 연쇄 복구 트랜잭션의 선택 방법

어떤 트랜잭션의 수정에 종속적으로 수정하여 그 트랜잭션의 UNDO 발생 시 잇달아 UNDO되는 트랜잭션을 연쇄 복구(Cascading Rollback) 트랜잭션이라고 정의한다. 수정의 취소를 결정한 서브-트랜잭션은 기존의 로그-기반 회복 기법을 이용해서 복구 된다. 그러나 연쇄 복구 트랜잭션들은 기존의 방법으로 처리할 수 없다. 이것은 공간 데이터 회복 시 공간 관련성에 의한 종속성을 고려해서 해결될 수 있다. 즉 공간 관련성에 의한 트랜잭션의 종속성에 따라 연쇄 복구 트랜잭션을 선택한다. 트랜잭션의 종속성은 다음에서 설명한다.

협동 수정 시 각 클라이언트들의 수정 영역은 8가지 관계가 가능한데, 이 중에서 'Disjoint'를 제외한 경우에 상호 중첩 영역이 존재하고 이를 NDJ(Non Disjoint area)로 정의한다[5].

NDJ내에서 서로 다른 두 트랜잭션이 공간 관련성이 존재하는 객체를 수정했을 때 두 트랜잭션의 관계를 공간 관련성에 의해 트랜잭션의 종속성(이하 '종속성')이라고 정의한다.

종속성은 수정 전과 수정 후의 관계(이하 '변경'), 복구 전과 복구 후의 관계(이하 '복구')에 따라 결정되는데 그 조건은 [표 1, 2]와 같다.

[표 1] '변경'에 관련하여 연쇄 복구 여부

변경 전	T _A 변경 후	T _B 변경 후	연쇄 복구
독립	독립	독립	X
독립	독립	종속	0
독립	종속	독립	X
독립	종속	종속	0
종속	독립	독립	X
종속	독립	종속	0
종속	종속	독립	X
종속	종속	종속	0

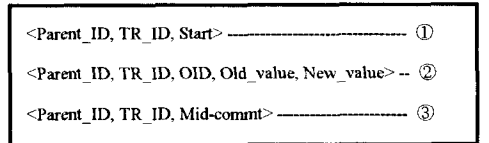
[표 2] '복구'에 관련하여 연쇄 복구 여부

복구 전	T _A 또는 T _B 복구 후	연쇄 복구
독립	독립	X
독립	종속	0
종속	독립	X
종속	종속	X

위의 표는 T_A와 T_B가 협동 작업할 때 둘 중 하나가 UNDO시 나머지 하나도 UNDO해야 하는지를 나타내었다. [표 1]은 '변경'에 의해 검사되는 조건이며, [표 2]는 '복구'시 검사되는 조건이다. T_A 또는 T_B가 이 두 가지 중 하나에 해당하면 두 트랜잭션은 종속성이 존재한다.

4.2 로그-기반의 트랜잭션 복구

트랜잭션의 복구를 위해서 사용되는 가장 일반적인 방법은 로그(Log)를 이용한 방법이다. 본 논문에서도 로그를 이용해서 트랜잭션을 복구하기 위해 [그림 2]와 같은 구조의 로그 레코드를 정의한다.



[그림 2] 로그 레코드

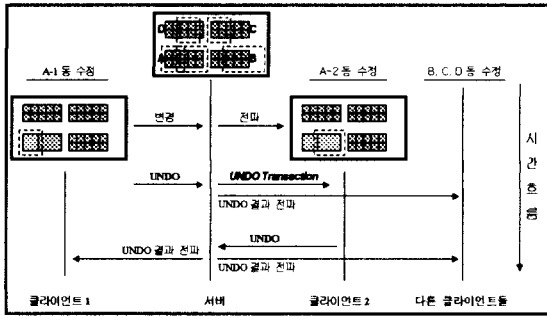
상위 트랜잭션은 일반적으로 정의된 로그 레코드를 그대로 사용하고 서브-트랜잭션의 경우는 [그림 2]와 같이 3가지의 로그 레코드를 이용한다. Parent_ID는 서브-트랜잭션의 상위 트랜잭션의 식별자이고,

TR_ID는 서브-트랜잭션의 식별자이며, OID는 변경된 객체의 식별자이다. 그리고 Old_value와 New_value는 각각 변경 이전과 이후의 객체를 의미한다. 3가지 로그 레코드는 다음과 같은 의미를 나타내기 위해 사용된다. ①은 서브-트랜잭션이 시작되었음을 나타내고 ②는 OID에 해당하는 객체를 변경 했음을 나타내며 ③은 서브-트랜잭션이 완료 했음을 나타낸다.

트랜잭션의 복구는 역 방향으로 복구될 트랜잭션의 변경 로그 레코드(Update Log Record)를 찾아 해당 객체들을 이전 객체(Old_value)들로 대체함으로써 이루어진다. 복구될 서브-트랜잭션의 시작 로그 레코드까지 검색되면 복구가 완료된 것이므로 시작 레코드와 변경 레코드 그리고 mid-commit 레코드를 로그 파일에서 제거한다.

4.3 연쇄 복구의 예

NDJ에서 협동 수정한 트랜잭션들 중 UNDO가 발생하면 종속성을 고려하여 UNDO 한다. 이러한 방법의 UNDO 시나리오를 [그림 3]에서 설명한다.

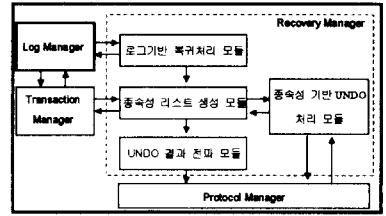


[그림 3] UNDO 시나리오

[그림 3]는 본 논문의 회복 기법이 [그림 1]에서 발생했던 문제점을 해결하면서 객체간의 공간 관련성을 유지할 수 있음을 보이고 있다. 클라이언트1은 A동의 한 부분(A-1)을 수정하고 클라이언트2는 A동의 나머지 한 부분(A-2)을 수정하고 있어서 이 두 클라이언트는 협동 수정하고 있는 셈이다. 나머지 클라이언트들은 [그림 1]과 마찬가지로 B, C, D동을 수정하고 있다. 즉 클라이언트1과 클라이언트2와는 독립적인 클라이언트들이다. 클라이언트1이 A-1의 부분을 수정하고 이 결과에 따라 클라이언트2는 A-2를 수정한다. 이 때 클라이언트1의 UNDO결정에 A동을 읽은 모든 트랜잭션을 UNDO하던 기존의 방법과 달리 협동 작업 중이면서 종속성이 존재하는 클라이언트2에만 UNDO하도록 메시지를 보낸다. 나머지 다른 클라이언트에는 UNDO 결과만을 보내 A-1동을 UNDO시킨다. 그리고 UNDO 요청을 받은 클라이언트2는 서버에 UNDO 결정 메시지를 보내고 UNDO 후 서버는 UNDO 결과를 전파한다. 결과적으로 제시된 기법을 적용하면 지도의 객체간 공간 관련성이 유지되며 연쇄 복구도 최소한으로 발생하게 됨을 알 수 있다.

4.4 시스템의 구성

회복을 위한 시스템의 구성은 [그림 4]과 같다. UNDO 결정된 서브-트랜잭션은 로그 기반 복구 처리 모듈에서 UNDO를 처리하고 [표 1]과 [표 2]에 따라 연쇄 복구될 서브-트랜잭션의 리스트를 종속성 리스트 생성 모듈에서 생성한다. 그리고 해당하는 서브-트랜잭션에 UNDO를 전파하는 것은 종속성 기반 UNDO 처리 모듈에서 담당한다.



[그림 4] 시스템의 구성

마지막으로 UNDO 결과 전파 모듈에서 클라이언트에게 UNDO 결과 전파하는 일을 담당한다.

4.5 UNDO 결과의 전파

클라이언트-서버 환경에서 서버는 트랜잭션을 복구한 후 그 결과를 클라이언트들에게 전파되어야 한다. 이때 메시지 부하를 줄이기 위해 협동 작업 중인 클라이언트에는 즉시 전파(immediate propagation)를 하고 나머지에는 지연 전파(deferred propagation)를 하는 방법을 각각 적용한다. 즉시 전파는 복구에 영향을 받을 수 있는 협동 트랜잭션들에게 적용해서 복구의 결과가 그 클라이언트에 즉시 반영되어 수정에 오류가 발생하지 않도록 한다. 그러나 복구에 의해 직접적인 영향을 받지 않는 독립적으로 수정 중인 트랜잭션들은 즉시 전파하지 않고 클라이언트의 요청에 의해 복구를 전파하는 기법을 사용한다. 이것은 UNDO시 최소한의 무결성을 보장하면서 순간 부하를 줄이는 방법이다.

5. 결론 및 향후 연구

본 논문에서는 클라이언트-서버 GIS 환경에서 동시 수행되는 공간 데이터 변경 트랜잭션들의 회복 기법을 제시하였다. 트랜잭션들의 연쇄 복구를 최소한으로 줄이면서 공간 관련성도 유지하기 위해 공간 관련성에 의한 트랜잭션의 종속성에 따라 회복하는 기법을 설명하였다. 그리고 서브-트랜잭션 단위로 UNDO 하여 회복의 비용을 줄이며 UNDO 결과를 효율적으로 전파하는 방법을 제시하였다.

향후 연구 과제는 본 논문에서 제시한 회복 기법을 클라이언트-서버 GIS 환경에서 구현하는 것이다. 더 나아가 미들 웨어를 사용하여 분산 공간 데이터베이스에서 지도 수정 트랜잭션의 회복 기법에 대한 연구가 필요하다.

6. 참고 문헌

- [1]. J. Eliot, B. Moss, "Log-Based Recovery for Nested Transactions", Proceedings of the 13th VLDB Conference, Brighton 1987, pp.427-423, 1987.
- [2]. Theo Haerder, "Concepts for Transaction Recovery in Nested Transactions", ACM, pp.239-248, 1987.
- [3]. C. Mohan, "ARIES/CSA: A Method for Database Recovery in Client-Server Architectures", ACM SIGMOD, vol23, no2, pp.55-66, 1994.
- [4]. 최진오, 홍봉희, "분산된 지리정보시스템에서 새로운 잠금 기법을 이용한 중복된 공간 데이터의 변경 전파", 한국정보과학회 논문지, vol26, no9, pp.1061-1072, 1999.
- [5]. 신영상, 최진오, 조대수, 홍봉희, "클라이언트 변경 트랜잭션에서 동시성 및 일관성 제어", '99 한국정보과학회 가을 학술발표논문집 vol26, no2, pp.323-325, 1999.