

대리 트랜잭션 모델에서의 공간 데이터 변경을 위한 완료 규약의 설계 및 구현

문선희^o 반재훈 홍봉희
부산대학교 컴퓨터공학과

Design and Implementation of Commit Protocol for Updating Spatial Data in the Surrogate Transaction Model

Sun-Hee Moon^o Chae-Hoon Ban Bong-Hee Hong
Dept. of Computer Engineering, Pusan National University

요 약

S-S-M(Server-Surrogate PC-Mobile Client)과 같은 3-계층 구조에서 이동 클라이언트를 이용한 공간 데이터 변경을 위해 대리트랜잭션 모델이 소개되었다. 이 모델에서는 이동 클라이언트의 공간 데이터 변경을 위한 트랜잭션간의 동시성 제어를 위하여 전통적인 낙관적 병행 기법인 확인 기법을 확장하였다.

분산 환경에서는 참여자가 완료를 요청하면 조정자는 데이터를 공유하는 모든 참여자에게 완료 또는 취소 여부를 물어 최종 완료를 수행하였으나 S-S-M 환경에서는 이동 클라이언트가 서버와의 접속을 해제하고 데이터를 변경한다. 따라서, 본 논문은 이동 클라이언트의 변경 완료 요청을 대리 PC를 통해 서버로 전달하고, 서버는 해당 트랜잭션의 충돌 여부를 검증하여 완료 또는 취소하는 완료 규약을 제시하고 이를 검증하기 위해 설계 및 구현한다.

1. 서론

S-S-M(Server-Surrogate PC-Mobile Client)과 같은 3-계층 구조에서 공간 데이터를 변경하기 위한 대리 트랜잭션 모델이 제시되었다[4]. S-S-M 구조에서 서버의 데이터는 대리 PC를 통해 이동 클라이언트로 전달되고, 이동 클라이언트는 서버와의 접속을 해제하고 공간 데이터를 변경한다. 또한, 이동 클라이언트가 변경한 데이터는 대리 PC를 통해 일괄처리 방식으로 서버에 전달된다.

대리 트랜잭션 모델에서는 이동 클라이언트들의 변경 트랜잭션간 동시성 제어를 위하여 전통적인 낙관적 병행 기법인 확인 기법을 사용하였으며, 변경 트랜잭션간 공간 데이터 사이의 공간 관련성 위배 여부를 검증하기 위하여 확인 기법을 확장하였다.

대리 트랜잭션 모델에서 이동 클라이언트가 변경한 내용에 대한 완료 규약이 필요하다. 분산 환경에서는 참여자가 조정자에게 완료를 요청하면 조정자는 트랜잭션을 공유하는 참여자들에게 반영 여부를 물어 최종 완료를 수행하였다. 대리 트랜잭션 모델에서는 이동 클라이언트가 서버와 연결을 해제하고 공간 데이터를 변경하며 이동 클라이언트의 완료 요청은 대리 PC를 통해 서버로 전달된다. 본 논문에서는 이동 클라이언트가 대리 PC에게 완료를 요청하면 대리 PC가 대신해서 서버에 변경 완료를 요청하며, 서버는 트랜잭션의 충돌 발생 여부에 따라 해당 트랜잭션을 반영하는 완료 규약을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 연구 대상인 S-S-M 구조와 대리 트랜잭션 모델을 소개하고 3장에서는 대리 트랜잭션 모델에서의 완료 규약을 설명한다. 4장에서는 완료 규약을 위한 시스템 구조 및 환경과 구현에 관하여 설명하며, 마지막으로 5장에서는 결론을 기술한다.

2. 대리 트랜잭션 모델

대리 트랜잭션 모델은 단일 GIS 서버와 다중 대리 PC와 다중 이동 클라이언트로 구성된 S-S-M(Server-Surrogate PC-Mobile Client) 구조를 대상으로 하며 서버, 대리 PC, 이동 클라이언트는 1:N:N의 관계이다. 서버의 공간 데이터는 대리 PC를 통해 이동 클라이언트로 전달되며, 이동 클라이언트가 변경한 데이터는 대리 PC를 통해 서버로 전달되어 최종 완료를 수행한다. 즉, 대리 PC는 서버에게는 클라이언트로 이동 클라이언트에게는 서버의 역할이다.

대리 트랜잭션 모델에서는 이동 클라이언트가 서버와의 접속을 해제하고 공간 데이터를 변경하므로 잠금과 같은 비관적 동시성 제어 기법은 잠금 획득을 위한 긴 대기 시간을 초래한다. 따라서, 일단 이동 클라이언트가 공간 데이터를 변경하고 변경한 내용이 다른 클라이언트의 트랜잭션과 충돌이 발생하면 해당 트랜잭션을 취소하는 낙관적 확인 기법으로 동시성을 유지한다. 공간 데이터는 disjoint, meets, insides, covers, equals, overlaps와 같은 공간 관련성이 존재한다. 따라서, 대리 트랜잭션 모델에서는 변경한 공간 데이터 사이의 공간 관

런성을 고려하여 확인 기법을 확장하였다. 그림 1 과 같은 경우 해당 트랜잭션은 충돌이 발생하지 않는다.

시간	MBR(MBR(Ti))	disjoint	MBR(Ti) non-disjoint MBR(Tj)	
			Finish(Ti) < Start(Tj)	Start(Tj) < Finish(Ti) < Validation(Tj)
DATA-SET			Write-Set(Ti) ∩ Read-Set(Tj) = ∅	Write-Set(Ti) ∩ Read-Set(Tj) ≠ ∅
			SR(Write-Set(Ti), Write-Set(Tj)) = {meet, inside, cover, cover}	SR(Write-Set(Ti), Write-Set(Tj)) + {meet, inside, cover, cover}
Spatial-Operation	COMMIT		COMMIT	ABORT

그림 1. 대리 트랜잭션 모델에서의 확장된 확인 기법

3. 대리 트랜잭션 모델에서의 변경 완료 규약

3.1 변경 완료 규약

분산 환경에서의 2PC(two phase commit) 나 3PC 의 완료 규약에서는 참여자가 조정자에게 변경 트랜잭션 T 의 완료를 요청하면, 조정자는 <prepare T> 를 로그에 저장하고 트랜잭션 T와 데이터를 공유하는 참여자들에게 <ready T> 메시지를 보낸다. 모든 참여자들이 <ready T> 메시지에 대해 <ye s T> 메시지를 조정자에게 전달하면 조정자는 변경 트랜잭션 T를 반영하고 <commit T> 를 로그에 저장하며 이를 각 참여자들에게 전달함으로써 완료 규약은 끝난다(2). 서버-클라이언트 구조에서는 공간 데이터를 동시에 수정하는 클라이언트들이 서버에 변경 완료 요청하면 서버는 협동 작업자들에게 공간 데이터의 변경을 알리고 완료 또는 취소 여부를 물어 최종 완료를 수행하였다(3). 그러나, 대리 트랜잭션 모델은 서버-대리 PC -이동 클라이언트의 3-계층 구조이며 이동 클라이언트들은 서버와의 접속을 해제하고 공간 데이터를 변경한다. 본 논문에서는 이동 클라이언트가 변경 트랜잭션의 완료를 서버에 요청한 경우 서버에서는 클라이언트에게 완료 또는 취소 결정 메시지를 보내어 클라이언트의 완료 또는 취소 결정에 따라 최종 완료를 수행하는 방법을 대신하여 변경 트랜잭션이 다른 클라이언트의 트랜잭션과 충돌이 발생했는지 검사하여 그 결과에 따라 요청 트랜잭션의 완료를 수행한다.

공간 데이터 변경 트랜잭션 T 의 실행이 이동 클라이언트 MCI에서 실행되었다고 하고 MCI 의 트랜잭션이 대리 PC 인 SPCi를 통해 서버 S 에 전달되었다고 하면 SPCi 와 S 는 다음과 같은 완료 규약을 시작한다.

단계 1

이동 클라이언트 MCI가 대리 PC인 SPCi에 변경 트랜잭션 T와 완료를 요청하면 SPCi는 <prepare T> 를 로그에 기록하고 T와 완료를 서버에 요청한다.

단계 2

서버는 MCI에 대한 변경 정보를 확인하고 변경 정보가 존재하지 않으면 트랜잭션 T를 반영하며 <commit T>를 로그에 기록한다. 반대로 MCI에 대한 변경 정보가 존재하면 트랜잭션 T가 다른 클라이언트의 트랜잭션과 충돌되는지 검사한다. 만약 트랜잭션 T가 충돌이 발생하면 서버는 <abort T>를 로그에 기록하고 이를 SPCi에 전달한다. 트랜잭션 T가 충돌이 발생하지 않으면 <commit T>를 로그

에 기록하고 해당 트랜잭션 T를 반영하며 그 결과를 SPCi에 전송한다.

단계 3

SPCi는 서버에서 <commit T>가 도착하면 <commit T>를 로그에 저장하고 T를 반영하며 이를 MCI에게 전달한다. 만약 <abort T>가 SPCi에 전달되면 <abort T>를 로그에 저장하며 이를 MCI에 전달한다. MCI는 SPCi에서 <commit T>가 도착하면 지도 수정을 종료하며 <abort T>가 도착하면 변경 트랜잭션 T를 취소한다.

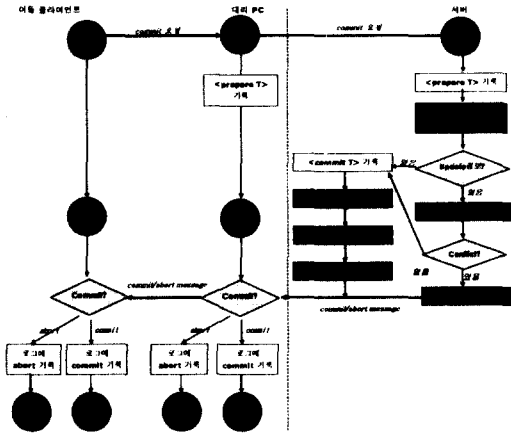


그림 2. 대리 트랜잭션 모델에서의 완료 규약

3.2 변경 완료 규약을 위한 로그

대리 트랜잭션 모델에서 공간 데이터 변경을 위한 로그는 서버와 대리 PC에 저장되며 그림 3 과 같이 정의된다. 대리 PC와 이동 클라이언트 사이는 사용자가 수동으로 sync 하므로 완료 규약을 수행하는 동안 이동 클라이언트에 로그를 남기지 않는다. LSN(Log Sequence Number) 는 로그 파일내의 로그 레코드의 논리적 위치를 나타내는 일종의 카운터이다. ST_ID와 MT_ID는 대리 트랜잭션과 이동 트랜잭션의 식별자를 의미한다.

(a) 트랜잭션의 시작

LSN	0	ST_ID	Start
-----	---	-------	-------

(b) 트랜잭션의 Commit 기법

LSN	0	ST_ID	prepare
-----	---	-------	---------

(c) 트랜잭션의 완료

LSN	MT_ID	ST_ID	commit
-----	-------	-------	--------

(d) 트랜잭션의 취소

LSN	MT_ID	ST_ID	abort
-----	-------	-------	-------

(e) 트랜잭션의 종료

LSN	0	ST_ID	Finish
-----	---	-------	--------

그림 3. 서버 및 대리 PC의 로그 레코드

(a)와 (e)는 대리 트랜잭션의 시작과 종료를 나타내며 (b), (c), (d)는 각각 이동 트랜잭션의 완료 요청과 완료 및 취소를 의미한다. 이동 클라이언트가 대리 PC에게 변경 완료를 요청하면 대리 PC는 그림 3-b를 로그에 기록하고 이를 서버에 전달한다. 서버는 대리 PC에서 완료 요청이 도착하면 그림 3-b를 로그에 기록하고 T의 충돌 여부를 검사한다. 만약 충돌이 발생하면 그림 3-c를 충돌이 발생하지 않으면 그림 3-d를 로그에 기록하고 이를 대리 PC에 전달한다. 대리 PC는 <commit T>가 도착하면 로그에 그림 3-c를 기록하고 <abort

T>가 도착하면 로그에 그림 3-d를 기록하며 이를 이동 클라이언트에게 전달한다.

4. 시스템 구조 및 구현

그림 4는 본 논문의 구현 환경 및 시스템 구조이다. 서버는 리눅스 환경의 공간 데이터베이스를 대상으로 구현하였으며, 대리 PC 와 이동 클라이언트는 각각 Windows 2000과 Windows CE 환경에서 Microsoft Visual C++ 6.0 과 Embedded Visual C++ 3.0으로 구현하였다.

4.1 시스템 구조

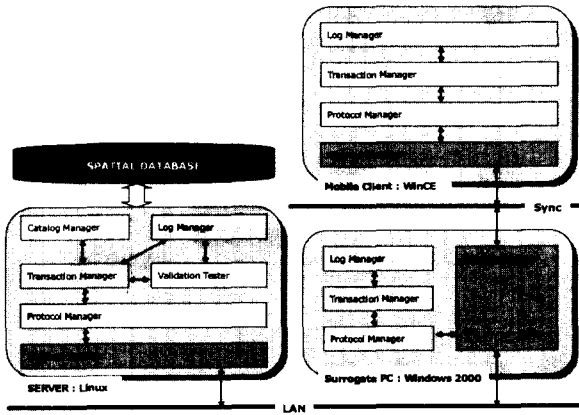


그림 4. 대리 트랜잭션 모델의 시스템 구조

서버의 수정 관리 모듈은 7 개의 모듈로 구성되며 각각의 기능은 다음과 같다.

- 트랜잭션 관리 모듈(transaction manager module)
클라이언트에서 요청하는 트랜잭션의 처리와 정보를 관리하는 모듈
 - 카탈로그 관리 모듈(catalog manager module)
클라이언트의 정보, 변경 정보들을 관리하는 모듈
 - 프로토콜 관리 모듈(protocol manager module)
클라이언트 요청 메시지에 대한 전체적인 프로토콜을 관리하는 모듈
 - 로그 관리 모듈(log manager module)
클라이언트의 트랜잭션과 관련된 로그를 관리하는 모듈
 - 충돌 감지 모듈(validation tester module)
공간 데이터 변경 트랜잭션이 다른 트랜잭션과 충돌하는지 여부 체크
 - 메시지 관리 모듈(message manager module)
클라이언트와의 통신을 위해 패킷을 분석하고 조립하는 모듈
- 대리 PC 와 이동 클라이언트는 트랜잭션 관리 모듈, 프로토콜 관리 모듈, 메시지 관리 모듈, 로그 관리 모듈의 4 개의 모듈로 구성되며 각 모듈의 역할은 다음과 같다.
- 트랜잭션 관리 모듈(transaction manager module)
클라이언트에서의 트랜잭션을 처리하거나 정보를 관리하는 모듈
 - 프로토콜 관리 모듈(protocol manager module)
클라이언트의 요청/응답 메시지에 대한 프로토콜을 관리하는 모듈
 - 로그 관리 모듈(log manager module)
트랜잭션과 관련된 로그를 관리하는 모듈
 - 메시지 관리 모듈(message manager module)
서버/클라이언트와의 통신을 위해 패킷을 분석하고 조립하는 모듈

4.2 완료 규약의 구현

그림 5는 두 이동 클라이언트에서 변경한 데이터에 대한 완료 요청이 수행되는 과정을 보인다. 그림 5-a에서는 서버로부

터 공간 데이터가 대리 PC 를 통해 이동 클라이언트에 전달되었다. 그림 5-b에서 두 이동 클라이언트는 각각 공간 데이터를 수정하고 입력하는 작업을 수행하였다.

그림 5-c에서와 같이 이동 클라이언트 C_A가 먼저 공간 데이터 삽입 트랜잭션에 대한 완료 요청을 하면 서버는 C_A의 트랜잭션이 다른 트랜잭션과 충돌하는지를 검증하고 충돌이 발생하지 않았으므로 C_A에게 <commit T> 를 전송하였다. 이후 클라이언트 C_B가 완료를 요청하면 C_B의 트랜잭션은 C_A의 트랜잭션과 충돌하므로 서버는 대리 PC 에게 <abort T> 를 전달한다.

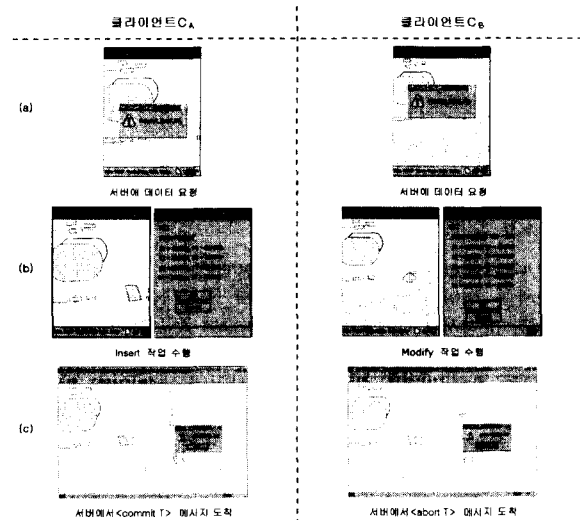


그림 5. 변경 완료 구현의 예

5. 결론 및 향후 연구

본 논문에서는 이동 환경에서 공간 데이터 변경을 위한 3-계층의 S-S-M 구조를 연구 대상으로 하였다. S-S-M 환경에서의 대리 트랜잭션 모델은 트랜잭션의 동시성을 제어하기 위하여 전통적인 낙관 기법의 확인 기법을 확장하였다. 또한, 이동 클라이언트의 완료 요청은 대리 PC 를 통해 서버로 전달되며, 연결이 해제된 이동 클라이언트를 고려하여 서버에서 트랜잭션간 충돌 검증을 통하여 최종 완료를 수행하는 완료 규약을 제시하였으며 이를 설계하고 구현하였다.

참고 문헌

- [1]. G. H. Forman, J. Zahorjan, " The Challenges of Mobile Computing" , IEEE Computer, Apr, 1994
- [2]. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, " Database System Concepts" , third edition
- [3]. 신영성, 최진오, 조대수, 홍봉희, " 클라이언트 변경 트랜잭션에서 동시성 및 일관성 제어" , '99 한국정보과학회 추계 학술발표논문집, vol 26, no 2, pp. 323-325, 1999
- [4]. 문선희, 반재훈, 홍봉희, " 이동 클라이언트의 공간 데이터 변경을 위한 대리 트랜잭션 모델" , 2001 춘계 학술발표논문집, 한국정보과학회, pp. 241-243, 2001