

모바일 클라이언트-서버 환경에서 다중 큐를 이용한 낙관적 동시성 제어 기법의 설계

김흥기, 김동현, 조대수

동서대학교

Design of an Optimistic Concurrency Control Technique using Multi-queue on a Mobile Client-Server Environments

Hong-Ki Kim, Dong-Hyun Kim, Dae-Soo Cho

Dongseo University

E-mail : inthestream@nate.com, pusrover@dongseo.ac.kr, dscho@dongseo.ac.kr

요약

양방향 동기화 기법은 모바일 단말기를 활용하는 GIS 시스템에서 사용자들에게 최신의 시공간 데이터를 신속하게 전파하기 위한 변경전파 기법이다. 그러나 여러 클라이언트들로부터 시공간 데이터가 변경될 수 있기 때문에 시공간 데이터의 일관성 유지에 문제가 있다. 이 논문에서는 모바일 클라이언트에서 변경한 시공간 데이터의 변경충돌검사를 위한 타임스탬프를 정의하고, 동시에 동기화를 요청한 클라이언트들의 트랜잭션을 처리하기 위해 다중 큐를 이용한 동시성 제어 기법을 제안한다.

키워드

모바일, 시공간 데이터, 동기화, 동시성 제어

I 서론

양방향 동기화 기법[1]은 모바일 GIS 시스템에서 시공간 데이터를 신속하게 전파하기 위한 시스템이다. 기존의 모바일 GIS 시스템은 최신의 시공간 데이터를 CD나 유선 네트워크를 이용하여 배포하였으며, 연 2회 시공간 데이터의 갱신작업을 수행한다. 이러한 배포방식은 사용자가 실제 공간 정보와 차이가 있는 시공간 데이터를 이용하는 문제가 있었다. 양방향 동기화 기법은 모바일 단말기를 이용하여 현장에서 변경된 시공간 데이터를 직접 수집하고, 무선 네트워크를 이용하여 실시간으로 서버와 동기화를 수행하여 신속하게 최신의 시공간 데이터를 배포할 수 있다[2].

그러나 여러 모바일 클라이언트들로부터 동일한 시공간 데이터가 수집되어 서버로 동기화될 수 있다. 이러한 동기화 작업을 기존의 분산 데이터베이스 환경에서의 동시 트랜잭션 수행처럼 시공간 데이터의 일관성이 무너질 수 있다.

이 논문에서는 여러 클라이언트들로부터 요청되는 동기화 작업들로부터 시공간 데이터의 일관성 유지할 수 있는 낙관적 동시성 제어 기법을 제안한다. 제안하는 동시성 제어 기법은 다중 큐를 이용하여 대하여 동시에 동기화를 수행되는 클라이언트들의 변경충돌을 방지한다.

이 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고 3장에서는 다중 큐를 이용한 낙관적 동시성 제어 기법을 설명한다. 마지막으로 4장에서는 결론 및 향후 연구에 대해서 설명한다.

II 관련연구

분산 데이터베이스 환경에서 동시에 수행되는 트랜잭션으로 발생할 수 있는 변경충돌의 해결방법으로 낙관적 접근 방법과 비관적 접근 방법으로 나눌 수 있다.

비관적 접근 방법은 변경을 수행하는 데이터 영역에 대하여 잠금을 적용하여 여러 클라이언트가 동일한 데이터를 변경할 수 없게 한다. 이러한 접근 방법은 변경충돌이 자주 발생하고 트랜잭션의 수행시간이 짧은 환경에서 효과적이다. 그러나

* 본 연구는 국토해양부 첨단도시기술개발사업 - 지능형국토정보기술혁신 사업과제의 연구비지원(07국토정보C05)에 의해 수행 되었습니다.

변경 객체가 대용량이고, 서버와 단절된 상태에서 트랜잭션이 수행되는 모바일 GIS 환경에는 부적합한 접근방법이다. 큰 공간 객체를 처리하기 위해 부분 잠금 기법[3]을 적용한 연구도 있지만, 모바일 환경에 적용하기는 힘든 문제가 있다.

낙관적 접근 방법은 대한 동시에 수행되는 트랜잭션들의 변경충돌 빈도가 적은 경우에 효과적인 방법이다. 낙관적 접근 방법은 트랜잭션 수행을 두세 가지 서로 다른 단계로 나누어 수행한다. 보통 단계의 구분은 관독 단계, 검증 단계, 기록 단계로 나눈다. 각각 단계들은 타임스탬프를 할당 받아 트랜잭션의 직렬성을 비교한다. 그러나 모바일 환경에서는 서버와 모바일 클라이언트들이 가지는 시간 값이 동일하지 않을 수 있다. 이러한 경우 트랜잭션이 단계별로 가지는 타임스탬프를 신뢰할 수 없는 문제가 있다[4][5].

III 다중 큐를 이용한 낙관적 동시성 제어 기법

- 계층(Layer) : 전체 데이터를 주제별로 분류하여 생성한 계층
- 분할(Partition) : 각각의 계층을 지리적으로 분할한 영역
- DO(Delta Object) : 최종 동기화 시간(LST) 이후 서버에서 변경된 데이터
- DOS(Delta Object Set) : DO들의 집합
- TDO(Temporary Delta Object) : 최종 동기화 시간 이후 클라이언트에서 변경된 데이터
- TDOS(Temporary Delta Object Set) : TDO들의 집합
- CR(Copy Region) : 클라이언트가 복사해서 가지고 있는 데이터영역 중 동기화를 요청하는 계층, 분할, LST의 집합
- LST(Last Sync Time) : 클라이언트가 서버와 최종 동기화를 한 시간으로 계층, 분할 별로 관리
- LUT(Last Update Time) : 서버의 데이터가 최종적으로 갱신된 시간으로 계층, 분할 별로 관리

3.1 변경충돌검사

다중 큐를 이용한 낙관적 동시성 제어 기법은 변경충돌 검사를 위해 사용하는 타임스탬프가 시공간 데이터 계층 및 분할에 할당한다. 서버 시공간 데이터에서 사용하는 타임스탬프는 LUT이고, 모바일 클라이언트에 사용하는 타임스탬프는 LST이다. LUT는 자신이 할당된 계층, 분할에 시공간 데이터의 갱신이 발생하면 동기화를 시작한 시간 값으로 타임스탬프를 갱신한다. LST는 모바일 클라이언트가 동기화를 정상적으로 완료하면 동기화를 시작한 시간 값으로 타임스탬프를 갱신한다.

표 1. 변경충돌 검사

대상	파티션 단위	객체단위	변경충돌
서버 vs 클라이언트	$LUT(P_i) \leq LST(P_i)$		X
	$LUT(P_i) > LST(P_i)$	$DOS(P_i) \cap TDOS(P_i) = \emptyset$ $DOS(P_i) \cap TDOS(P_i) \neq \emptyset$	X O
클라이언트 vs 클라이언트		$TDOS(M_i) \cap TDOS(M_j) = \emptyset$ $TDOS(M_i) \cap TDOS(M_j) \neq \emptyset$	X O

모바일 클라이언트가 변경한 시공간 데이터에 대하여 동기화를 요청하면 표 1과 같이 변경충돌 검사를 수행한다.

- 서버 vs 클라이언트 : TDOS가 있는 모바일 클라이언트의 CR의 LST와 이에 대응하는 서버 CR의 LUT를 비교하여 $LST > LUT$ 이면 변경충돌이 없는 경우이고, $LST < LUT$ 이면 변경충돌 발생 가능하다.

$$\begin{cases} LST(P_i) < LUT(P_i) \dots \dots \dots \text{변경충돌 가능} \\ LST(P_i) > LUT(P_i) \dots \dots \dots \text{변경충돌 없음} \end{cases}$$

- 변경충돌 발생가능한 경우 서버의 DOS를 생성하여 TDOS와 같은 객체가 있는지 비교하여, 동일한 객체가 존재하면 변경충돌 발생이다.

$$\begin{cases} TDOS(P_i) \cap DOS(P_i) = \emptyset \dots \text{변경충돌 없음} \\ TDOS(P_i) \cap DOS(P_i) \neq \emptyset \dots \text{변경충돌} \end{cases}$$

- 클라이언트 vs 클라이언트들 : 동기화를 요청한 클라이언트가 서버와 변경충돌이 없고, 동기화 중인 클라이언트들이 있는 경우 실행한다. 동기화를 요청한 클라이언트의 TDOS와 동기화 중인 클라이언트들의 TDOS를 비교한다. 동일한 객체가 존재하면 변경충돌 발생이다.

$$\begin{cases} TDOS(M_i) \cap TDOS(M_j) = \emptyset \dots \text{변경충돌 없음} \\ TDOS(M_i) \cap TDOS(M_j) \neq \emptyset \dots \text{변경충돌} \end{cases}$$

3.2 다중 큐 삽입방법

변경충돌이 없는 모바일 클라이언트는 표 2의 규칙대로 다중 큐에 삽입한다. 표 2는 다중 큐의 큐 개수가 3개인 경우의 삽입규칙이다.

- QS : 갱신 작업이 있는 큐의 개수를 나타낸다.
- 중복영역 : 다중 큐에 삽입될 모바일 클라이언트의 CR과 동기화 중인 모바일 클라이언트의 CR이 동일한 영역이 존재하는지 나타낸다.
- 중복 큐 : 다중 큐에 삽입하려는 모바일 클라이언트의 CR과 동일한 영역이 존재하는 모바일 클라이언트가 삽입된 큐의 수를 나타낸다.
- 중복영역검사 후 큐의 상태와 중복 큐의 수에 따라 삽입될 데이터가 어디에 삽입되어야 하는

지 나타낸다.

표 2. 다중 큐 삽입규칙

큐의 상태	중복영역	중복 큐의 수	삽입방법
QS = 0		0	비어있는 큐에 삽입
QS = 1	$CR(Q) \cap CR(M) \neq \emptyset$	1	비어있는 큐에 삽입
	$CR(Q) \cap CR(M) = \emptyset$	0	중복이 발생한 큐에 삽
QS = 2	$CR(Q) \cap CR(M) \neq \emptyset$	0	비어있는 큐에 삽입
	$CR(Q) \cap CR(M) = \emptyset$	1	중복이 발생한 큐에 삽입
QS = 3	$CR(Q) \cap CR(M) \neq \emptyset$	0	데이터가 가장 적은 큐에 삽입
	$CR(Q) \cap CR(M) \neq \emptyset$	1	중복이 발생한 큐에 삽입
	$CR(Q) \cap CR(M) \neq \emptyset$	2	데이터가 많은 큐에 삽입
	$CR(Q) \cap CR(M) = \emptyset$	3	데이터가 가장 많은 큐에 삽입

다중 큐에 삽입방법이 큐의 상태와 중복 큐의 수에 따라 다른 이유는 서로 다른 동기화 작업간의 직렬성을 보장하기 위해서이다.

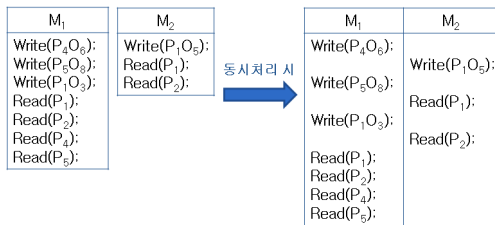


그림 1. M1과 M2의 동시처리 스케줄

예를 들어 다중 큐에서 처리중인 동기화 작업이 있을 때 새로운 동기화 작업을 삽입하려고 한다. 이때 중복영역검사 없이 비어있는 큐가 있어 비어있는 큐에 새로운 동기화 작업을 삽입하였다 가정하자. 기존에 처리중인 작업과 새로운 동기화 작업이 서로 중복된 영역이 있다면 다음 그림과 같은 직렬성에 문제가 발생할 것이다.

그림 1은 중복영역검사 없이 M1이 M2보다 먼저 동기화를 요청하여 동시에 동기화를 처리한 경우이다. 오른쪽 스케줄 표를 보면 M2는 M1이 Write(P1O3)한 작업을 읽지 못하고 동기화를 완료하게 된다.

IV 결론

모바일 GIS시스템은 여러 클라이언트들이 개별적으로 변경된 시공간 데이터를 수집하여 동기화를 수행함으로 시공간 데이터의 일관성 유지에 문제가 발생한다. 이 논문에서는 모바일 환경에서 여러 클라이언트로부터 요청되는 동기화 작업에 대하여 시공간 데이터의 일관성을 유지할 수 있는 동시성 제어 기법을 제안한다. 제안한 동시성 제어 기법은 다중 큐를 이용한 낙관적 동시성 제어 기법이다. 다중 큐는 동시에 수행되는 동기화 작업의 직렬성을 보장한다.

참고문헌

- [1] 김홍기, 김동현, 조대수, “모바일 환경에서 필드 업데이트를 지원하는 시공간 데이터의 양방향 동기화 시스템의 설계 및 구현”, 한국해양정보통신학회논문지 제14권 4호 2010
- [2] 이해진, 김진석, “모바일 환경에서 공간데이터 동기화 시스템 설계,” 한국정보과학회, 2004년 가을 학술발표논문집 제31권 제2호, 2004. 10, pp. 184~186
- [3] 서영덕, 김동현, 홍봉희, “큰 공간 객체의 변경을 위한 동시성 제어,” 정보과학회논문지, 데이터베이스 제 32권 제1호, 2005. 2, pp. 100~110
- [4] 김대호, 정병수, 이승룡, “실시간 보안 데이터베이스 시스템을 위한 낙관적 동시성 제어 기법,” 정보과학회논문지, 데이터베이스 제 27권 제1호, 2000. 3, pp. 42~52
- [5] 김대호, 정병수, 이영구 “이동 데이터베이스 시스템에서 타임스탬프를 이용한 낙관적 동시성 제어 기법” 한국 인터넷 정보학회 ,7권 4호, 2006. 8, PP.79~90