

단일 공간 객체의 동시 변경을 위한 부분 잠금 기법

李东春^{0*} 서영덕^{**} 홍봉희^{**}
*부산대학교 GIS학과, **부산대학교 컴퓨터공학과
dongli@pusan.ac.kr, fydseo.bh@pusan.ac.kr

Partial locking schemes for concurrent updating of a single spatial object

Li Dong-Chun^{0*} Young-Dok Seo^{**} Bong-Hee Hong^{**}
⁰Dept. of GIS, Pusan National University ^{**}Dept. of Computer Engineering, Pusan National University

요약

지리정보시스템(Geographic Information System, GIS)에서 지도 수정 작업은 대화식으로 이루어지는 긴 트랜잭션이다. 도로 객체와 같은 큰 공간 도형을 수정하는 경우 한 작업자의 변경을 위한 잠금이 다른 작업자가 동일한 객체의 지리적으로 떨어져 있는 기타 부분에 대한 변경을 못하게 하는 문제가 존재한다. 예를 들면, 38번 국도를 수정하려고 할 때 하나의 큰 라인 객체 전체가 잠금 대상이 되어서 다른 지역의 38번 국도를 동시에 수정하지 못하는 문제가 있다. 본 논문에서는 크기가 큰 공간 객체를 동시에 수정하는 것을 허용하기 위하여 하나의 BLOB 데이터로 저장된 공간 데이터를 동시에 수정할 수 있는 부분 잠금 기법을 제안한다. 즉 하나의 공간 객체를 여러 덩어리 동시에 수정하기 위하여 공간 도형에 대한 메타 데이터를 유지하고 이를 기반으로 한 잠금 정보를 표현함으로써 긴 트랜잭션에서의 동시성 제어를 실현하는 이점을 제공한다.

1. 서론

지리정보시스템(Geographic Information System, GIS)에서 지도 수정 작업은 대화식으로 이루어지는 긴 트랜잭션이다. 이러한 긴 트랜잭션에서 지도 수정 작업을 진행 할 때 여러 클라이언트가 동시에 지도 데이터베이스에 접근하여 수정 작업을 진행하여 작업 동시성을 보장해주는 것은 공간 데이터베이스만 아니라 일반적인 DBMS에서도 중요한 부분이다 [1],[2],[3]. 여기서 잠금은 여러 클라이언트가 데이터베이스에 동시 접근가능 하도록 보장하는 중요한 기법으로 많은 연구가 수행되어 왔다. 하지만 이러한 기법은 기존의 관계형 DBMS를 대상으로 연구된 결과로서 지도 수정과 같은 긴 수행 시간을 요구하는 트랜잭션에서는 부적합하다는 연구 결과가 나왔다[4].

클라이언트-서버 환경에서 잠금 기법을 이용한 지도 수정은 데이터에 대한 쓰기 잠금 권한을 갖고 있는 트랜잭션만 수행할 수 있다. 한 데이터 객체는 여러 트랜잭션이 동시에 읽기 잠금을 요청하는 것을 수용할 수 있지만 두 트랜잭션이 같은 데이터에 동시에 쓰기 잠금을 요청하는 것은 허용하지 않는다. 그러나 공간 데이터는 기하적 속성을 갖고 있어서 여러 클라이언트가 도로 객체와 같은 큰 공간 도형을 수정할 때 만약 한 수정 트랜잭션이 변경을 위해 잠금을 설정하면 다른 트랜잭션은 지리적으로 떨어져 있는 동일한 공간 객체의 기타 부분을 동시에 변경하지 못하여 대기해야 하는 문제가 존재한다. 이러한 대기 문제는 지도 수정 작업이 수시간 심지어 수일, 수개월이라는 시간이 필요한 긴 트랜잭션이라는 특성을 고려하면 더욱 심각해진다.

이 문제를 해결하기 위하여 본 논문에서는 클라이언트-서버

환경에서 클라이언트 작업공간보다 큰 공간 객체의 기하 속성을 대상으로 수정 작업을 진행 할 때, 객체 전체가 아닌 수정하려는 일부 객체만 잠금 대상으로 고려 하여 변경 작업을 수행하는 객체 부분 잠금 기법을 제안하고 이에 관한 연구를 기술한다.

논문의 전체 구성은 아래와 같다. 2장에서는 지금까지 클라이언트-서버 환경에서 트랜잭션에 기반한 공간 데이터 변경에 관한 연구를 기술하고 3장에서는 기존에 있는 연구 중에서 해결 안된 문제점을 제시한다. 4장에서는 3장에서 제시한 문제점에 관하여 부분 잠금 개념을 제안하고 부분 잠금 기법에 의한 해결책을 제시한다. 마지막으로 5장에서는 결론 및 향후 연구과제를 제시한다.

2. 관련연구

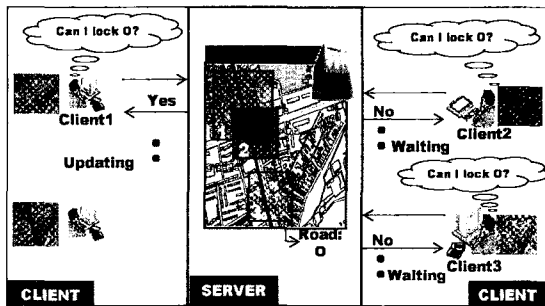
클라이언트-서버 지리정보시스템 환경에서의 지도 수정 작업은 서버에 있는 공간 객체들을 네트워크를 통해 클라이언트로 전송하여 화면에 출력 한 후 대화식으로 이루어지는 긴 트랜잭션이다. 긴 트랜잭션은 특수 트랜잭션으로써 그 일관성 및 동시성 지원 문제가 계속 논의되어 왔었다[5,6,7]. [5]는 SAGA모델을 제시하고, LLT(long lived transaction)의 동시성 지원을 위한 잠금의 문제점을 지적하고, LLT를 일련의 서브-트랜잭션으로 분할한 후 동시 실행시키는 기법을 제안하였다. 이때 발생하는 종속성 문제는 트리거 메커니즘으로 해결하고 있다. [6]에서는 분산된 공간 데이터베이스에서 중복 데이터의 일관성 제어를 위하여 확장된 잠금을 사용한 영역잠금법 및 공간 관련성 바운드 WRITE 잠금법을 제안했다. 이중에 영역 잠금은 공유 잠금의 확장으로서 중복된 공간 객체들의 그룹에 대한 공유 잠금이다. [7]에서는 클라이언트-서버 환경에

서 잠금 기법에 기반 한 변경 전파에 대한 알고리즘을 제안하고 있다. 특히, 다수의 클라이언트가 중복된 영역에 대한 수정 작업을 진행 할 경우의 트랜잭션 모델을 위하여 [6]에서 정의한 영역 잠금을 기반으로 한 CS(Cached Shared)-잠금, CX(Cached eXclusive)-잠금 등의 확장된 잠금 기법을 제안하고 있다.

위에서 언급한 연구들은 클라이언트-서버 환경에서 지도 데이터 수정 시, 클라이언트와 클라이언트 사이에서 수행되는 트랜잭션 제어와 데이터 사이에 존재하는 공간 관련성에 의한 변경 중속성을 해결하기 위하여 제안되었다. 그러나 공간 객체의 기하적 속성을 표시하는 기하 데이터가 [8]에서 제안한 바와 같이 BLOB 형식으로 한 레코드의 필드에 저장 되어 있을 경우, 잠금의 최소 단위가 레코드라는 점을 고려하면 여러 클라이언트의 공간 객체 사이에 공간 관련성 충돌이 발생하지 않는 경우에도 같은 객체를 수정한다는 이유로 동시에 쓰기 잠금 요청을 허용하지 않는다. 본 논문에서는 기존에 연구된 공간 데이터 제어의 잠금 기법을 바탕으로 BLOB 형식으로 저장된 공간 데이터의 기하적 속성을 여러 클라이언트가 동시에 수정 가능하게 하는 BLOB 데이터 부분 잠금 기법을 제안하여 지도 수정의 동시성을 향상시킨다.

3. 문제정의

<그림 1>에서 보는 바와 같이 클라이언트1,2,3이 서버에 저장되어 있는 도로 객체 O의 서로 다른 부분을 수정하기 위해 쓰기 잠금을 서버에 요청하고 있다. 만약 클라이언트1이 쓰기 잠금을 설정하고 수정 작업을 진행하고 있으면 기존 방법에서는 클라이언트2,3은 클라이언트1의 작업이 끝날 때까지 대기해야 한다. 클라이언트2는 클라이언트1과 겹친 영역에 속하는 부분을 수정하려는 경우, 공간 관련성에 의해 충돌 문제가 존재하여 대기할 수 있다. 그러나 클라이언트3은 지리적으로 수정할 영역이 분리되어 공간 관련성이 없지만, 단지 같은 공간 객체를 수정한다는 이유로 수정 작업을 대기해야 한다. 공간 객체의 기하적 속성이 경우에 따라 전체 데이터베이스 영역에 분포될 수 있다는 점을 고려하면 이러한 대기 문제가 발생하는 가능성은 매우 높다.



<그림 1> 여러 클라이언트가 같은 객체에 대해 동시 수정 시 문제
본 논문에서는 <그림 1>의 도로 객체의 예와 같이 동일 객체의 서로 다른 부분을 동시 수정하려 할 때, 수정 트랜잭션 동시성을 향상할 수 있는 동일 객체 부분 잠금 기법을 제안한다.

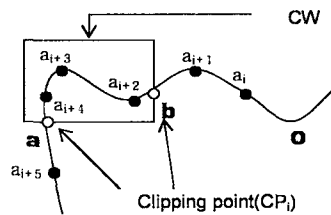
4. 해결방안

본 논문에서 다루는 공간 객체의 기하적 속성 데이터는 [8]에서 공간 데이터를 저장하는 두 가지 포맷 중의 한 가지로 제안한 WKBGOMETRY(Well Known Binary Representation for Geometry)와 같이 BLOB(Binary Large Object)형식으로 공간 데이터베이스에 저장되어 있는 데이터이다. 본 논문에서는 BLOB 형식으로 표현한 공간 객체 레코드의 필드에 저장되는 공간 객체의 기하적 속성을 부분적으로 수정하는 것을 허용하

기 위하여 부분 객체라는 개념을 정의하고 부분 객체 잠금 기법을 제안한다.

4.1 부분 객체

수정 대상인 공간 객체와 수정 작업을 진행하는 클라이언트의 작업 영역이 겹칠 때 공간 객체와 클라이언트 작업 영역이 겹치는 부분을 그 공간 객체의 부분 객체라고 정의한다. 클라이언트 작업공간과 수정 객체가 교차하는 점을 클리핑 포인트라 하며, 부분 객체는 그 사이에 존재한다.



<그림 2> 부분 객체의 정의

클라이언트의 작업공간을 CW, 수정 대상인 포인트 셋으로 구성된 공간 객체를 $O = \{a_1, a_2, a_i, \dots, a_n\}$ 라고 정의 할 때 부분 객체 O_i 는 <그림 2>에서 표기한 바와 같이 $O_i = \{a_{i+2}, a_{i+3}, a_{i+4}\}$ 로 정의하고 a, b는 해당 되는 클리핑 포인트이다.

4.2 부분 잠금 프로토콜

4.2.1 부분 잠금

기존 연구에서는 쓰기 잠금과 읽기 잠금은 RDBMS에서 한 레코드를 최소 잠금 단위로 한다. BLOB 형식으로 한 레코드로 저장 되는 공간 데이터의 공간 속성을 더 세분 하여 잠금을 설정 할 수 없다. 이러한 문제를 해결 하기 위하여 부분 잠금 개념을 도입한다. 부분 잠금은 PX(Partial-eXclusive)와 PS(Partial-Shared)잠금 모드로 분류하여 아래와 같이 정의 한다.

정의 1: 한 공간 객체 O를 구성하는 부분 객체 O_i 에 대하여 수정 하고자 할 때, O_i 또는 O_i 와 겹치(intersect)는 기타 부분 객체들이 수정 중이라면 대기해야 하는 쓰기 잠금 모드를 PX(Partial-eXclusive)잠금이라 정의한다.
정의 2: 한 공간 객체 O를 구성하는 부분 객체 O_i 가 한 클라이언트에 의해 캐싱 될 때 설정되고 캐쉬에서 교체 될 때 해제되는 잠금 모드를 PS(Partial-Shared)잠금이라 정의한다.

여기서 PS 잠금은 일반 Shared 잠금과 다른 점은 다른 클라이언트들의 수정을 위한 잠금 요청을 허용한다는 것이다. 이는 동시성 지원을 위해 하나의 클라이언트가 부분 객체를 캐쉬에 저장 하고 있다고 해서 다른 클라이언트들이 이 데이터를 수정하지 못하게 하는 것을 막기 위한 것이다.

4.2.2 잠금 호환성

잠금 호환성은 표1과 같다. 한 객체에 대한 write 잠금이 설정되어 있지 않으면 그 객체 일부에 대한 PX 잠금 요청을 허용한다. 그리고 PX 잠금 후 수정 중인 부분 객체 대하여 PS 잠금을 걸고 읽는 것을 동시성 지원을 위해 허용한다. 즉 지도 수정이 긴 트랜잭션이므로 공간 데이터가 변경되고 있는 동안 dirty read를 허용한다. 그러나 데이터 수정은 PX 잠금 권한을 얻어야 하며 PX 잠금은 서로 호환되지 않으므로 lost update 문제는 발생하지 않는다. 여기서 발생하는 데이터의 부정확성 문제는 수정 작업 끝난 후 수정 결과 변경 전파를 통해 해결한다. write 잠금은 서버에서 주기적으로 조사하여 객체가 수정 중이 아니면 데이터베이스에 업데이트 할 때 설정하는 잠금이다.

	ps	px	write
Ps	yes	yes	no
Px	yes	no	no
Write	no	no	no

<표 1> 잠금 호환성

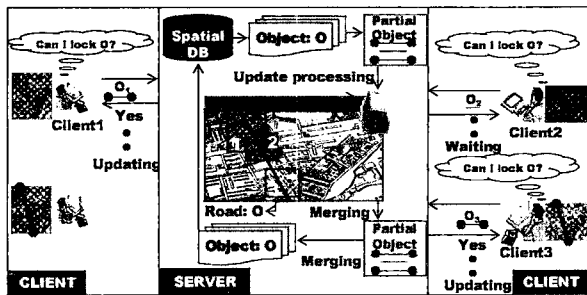
여기서 부분 잠금 설정 및 PX 잠금 충돌 여부를 확인하기 위하여 부분 객체를 표기하는 클리핑 포인트 정보도 함께 갖고 있어야 한다<표 2>. 클라이언트가 수정 작업을 미치고 서버에 합병 요청할 때 서버에서는 클리핑 포인트를 이용하여 객체 어느 부분이 수정되었는지 조사하고 객체를 업데이트한다. 이때 기타 클라이언트에 의해 설정된 PS 잠금 영역을 함께 조사하여 영역이 중복되면 변경된 부분을 전파한다.

필드	필드 정의
tr_id	클라이언트의 트랜잭션 아이디
oid	부분 잠금을 시도하는 공간 객체의 아이디
cp ₁	부분 객체를 표기하는 클리핑 포인트 좌표(x,y)
cp ₂	부분 객체를 표기하는 클리핑 포인트 좌표(x,y)
status	잠금 필드

<표 2> 부분 잠금 테이블의 정의

4.3 부분 잠금 과정

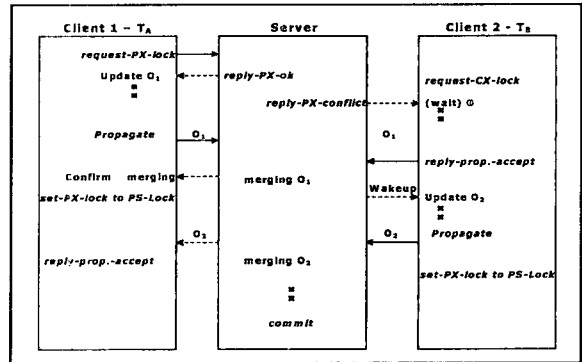
<그림 3>에서 도시한 바와 같이 클라이언트1,2,3이 동시에 도로 객체 O에 대한 수정 작업을 진행할 때 도로 객체 O는 각 클라이언트 작업 공간에 의해 O₁, O₂, O₃로 표시하는 3개의 부분 객체로 동적 생성된다. 생성된 데이터 정보는 서버에서 관리되고 클라이언트1,3이 설정한 부분 객체 O₁, O₃는 지리적으로 분리되어 있고 서로 겹치지 않는다. 즉 각 부분 객체에 대하여 동시에 PX 잠금을 설정하고 수정 작업을 진행할 수 있다. 하지만 클라이언트1,2와 같이 수정 대상으로 하는 부분 객체가 서로 겹치는 경우에는 클라이언트2는 클라이언트1의 수정 작업이 끝날 때 까지 대기해야 한다<그림 4>.



<그림 3> 부분 잠금을 이용한 BLOB 데이터의 수정 과정
클라이언트1의 수정 작업이 끝나고 서버에 수정 결과에 대한 합병(Merging)작업을 요청하면 서버에서 작업 결과를 클라이언트3에게도 변경 결과를 전파한다. 변경 전파가 성공적으로 끝나면 수정 결과를 객체 O에 합병한다. 클라이언트3은 변경 전파된 데이터를 확인하여 수정 작업을 계속 진행할 필요가 있는지 판단하고 필요하면 수정 작업을 시작한다.

기존에는 n개 클라이언트가 한 공간 객체를 수정 할 때 n-1개 클라이언트가 대기하는 문제가 존재한다. 그러나 공간 데이터를 부분 객체로 분할하여 부분 잠금 기법을 적용하면 동시에 수정 작업을 진행할 수 있다. 즉, 동일 객체에서 수정하려는 부분이 겹치는 작업이 끝날 때까지만 대기하면 되어서 수정 작업의 동시성을 높일 수 있다. 예를 들면 <그림 3>에서 클라이언트1,3과 같이 같은 객체지만 서로 지리적으로 분리되어 있는 부분을 수정할 때 동시에 작업을 진행할 수 있도록 보장해 준다. 서버에서는 주기적으로 잠금 테이블을 조사하여

수정 작업이 끝난 객체를 공간 데이터베이스에 최종 COMMIT을 시도한다. 여기서 PX 잠금 충돌은 부분 객체 정의에서 부분 객체를 구성 하는 Segments중에 최소한 한 개가 중복되었다는 것을 알 수 있다. 충돌여부 조사는[9]에서 제시한 segment intersection 알고리즘을 이용하여 해결 할 수 있다.



<그림 4> BLOB 데이터 수정을 위한 부분 잠금 프로토콜 예

5. 결론 및 향후 연구

본 논문에서는 부분 객체 개념을 제안하여 클라이언트-서버 지리정보시스템 환경에서 공간 객체의 기하적 속성에 대한 수정 작업 수행 시 동시성을 높인다. 즉 기존에 연구해 왔던 객체 단위의 수정 작업 방식과 달리 BLOB 형식으로 되어 있는 기하적 속성 데이터를 클라이언트의 작업 공간에 의해 세분화된 부분 객체 단위로 분할하여, 동시 수정 작업을 진행할 수 있는 부분 객체 잠금 기법을 제안하고 있다.

향후 연구에서는 부분 잠금을 이용하는 실제 시스템에 대한 설계 및 구현 작업을 진행할 것이다.

참고 문헌

- [1] Eswaran, K., Gray, J., Lorie, R. and Traiger, I. "The Notions of Consistency and Predicate Locks in a Database System" Comm. ACM, 19,11,pp. 624-633, 1976
- [2] Papadimitriou, C. "The theory of data base concurrency control" Computer Science Press, 1986
- [3] Bernstein, P., Hadzilacos, V. and Goodman, N. "Concurrency Control and Recovery in Database Systems", Addison-Wesley, 1987
- [4] Thomasian, A., "Two-phase Locking Performance and its Thrashing Behaviour", ACM TODS, 18, 4, pp. 579-625, 1993
- [5] H. G. Molina, K. Salem, "Sagas", in Proc. Of ACM SIGMOD Int. Conf. on Management of Data, pp. 249-259, 1987
- [6] 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
- [7] 신영상, 최진오, 홍봉희, "클라이언트-서버 환경에서 캐쉬 공간 데이터의 변경 전파," 99 한국정보과학회 봄 학술발표논문집, vol 26, no 1, pp. 86-88, 1999
- [8] Open GIS Consortium, Inc. "Open GIS Simple Features Specification For SQL", Aug, 1997
- [9] Joseph O'roupe, "Computational Geometry", pp. 222, 2001