

모바일환경에서의 트랜잭션 분류에 따른 확장 로킹 기법

권혁신*, 류봉수**, 김응모*

*성균관대학교 정보통신공학부 컴퓨터공학과

e-mail : mangcoda@ece.skku.ac.kr

An Extended Locking Protocol for Mobile Transaction Management Using Classifying Transactions to Two-Type

Hyeok-Shin Kwon*, Bong-Su Ryu**, Ung-Mo Kim*

**Dept of Computer Engineering, Sung-Kyun-Kwan University

요 약

본 논문에서는 모바일 데이터베이스 시스템에서 동시성제어의 정도를 높이고자 새로운 프로토콜을 제안한다. 트랜잭션의 동시성제어를 하는데 있어서 다중적인 접근으로부터 문제시 되는 데이터 일관성을 보장하고, 대역폭이 낮은 모바일 환경 속에서 일어날 수 있는 잦은 접속 단절과 잦은 이동에 대한 트랜잭션 처리의 효율을 높이고자 한다. 2PL(two-phase locking)로부터 확장된 프로토콜인 XAL(extended altruistic locking)모델의 연장으로서 양방향(Bi-directional) 기부연산에 따른 트랜잭션의 처리와 트랜잭션을 관독전용과 관독 및 갱신용 트랜잭션의 두 가지로 분류함으로써 이동 컴퓨터 환경에서 단기 트랜잭션의 처리성능의 정도를 높이고자 한다.

1. Introduction

모바일 환경은 낮은 대역폭이라는 특징으로 인해 잦은 접속단절과 높은 에러율을 갖는다[1]. 그러므로 모바일을 포함하는 분산 시스템에서 다중으로 발생하는 트랜잭션을 어떻게 데이터의 충돌없이 일관성을 지키며 신속하게 처리할건지가 관건이 되고 있다. 장기 트랜잭션으로 인해 유발 가능한 단기 트랜잭션의 장기적 대기현상, 즉, 모바일 데이터베이스에서 장기 트랜잭션이 발생하여 과도한 데이터 점유를 하게 되면, 잦은 단기 트랜잭션들의 성능저하를 크게 일으키게 되고 이는 곧 트랜잭션처리 성능의 저하를 초래하게 됨으로써 모바일 환경의 발전에 큰 성능저하의 원인이 된다. 그래서 본 논문에서는 2PL(two-phase locking) 확장 프로토콜인 XAL(extended altruistic locking)모델의 연장으로서

양방향(Bi-directional) 트랜잭션을 허용하고 트랜잭션들을 관독전용과 관독 및 갱신용으로 두 가지로 분류함으로써 이동 컴퓨터 환경에서의 단기 트랜잭션의 처리성능을 개선한다.

2. Basic Concepts

기존의 데이터베이스에서 사용되는 잠금(Locking) 기법과 타임스탬프(Time-Stamp ordering)기법은 모바일 데이터베이스 환경에서 트랜잭션을 효율적으로 처리하기에는 역부족이다.

데이터의 일관성은 표준 트랜잭션 스케줄링 기법인 2단계 잠금기법(2PL)에 의해 보장되어지고 있다. 2PL은 잠금(Lock)과 해제(Unlock)이라는 두 가지의 연산을 이용한다. 2PL의 확장개념인 이타적 잠금기법(Altruistic Locking:AL)은 기부(Donate)라는 연산

을 추가함으로써, 3가지 연산을 사용하여 일정 제약 아래 동시성제어의 정도를 높이고 있다. Extended Altruistic Locking(XAL)은 기부라는 연산이 가지는 영향력의 범위를 넓힌 개념이다. 2PL에서부터 확장된 이러한 동시성 제어 기법(protocol)들은 그 성능에 있어서 2PL보다 우수하다는 것은 이미 검증된 바 있다. 본 논문에서 제안하고자 하는 프로토콜인 Two-type Mobile Transactions over Extended Altruistic Locking(TMXAL)은 XAL과 모바일 트랜잭션 모델 중 하나인 Moflex Transaction Model[2]에 바탕을 두고 있다.

3. Related Works

3.1. Altruistic Locking Protocol (AL)

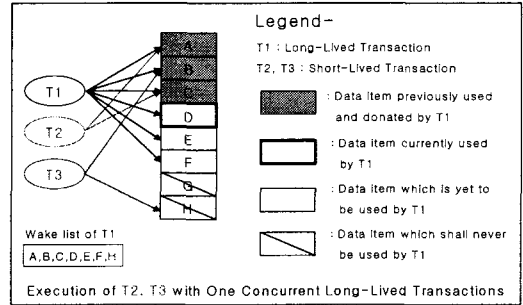
이타적 잠금기법(Altruistic Locking)[3]은 2단계 잠금기법(Two-Phase Locking)[4]을 보완한 것이다. 이타적 잠금 기법은 기부(donate)라고 하는 새로운 동시성 제어연산을 제공한다. 해제와 마찬가지로 기부는 객체를 잠금한 트랜잭션이 더 이상 그 객체를 필요로 하지 않을 때 데이터베이스 관리자에게 그 객체를 다른 트랜잭션들이 사용할 수 있도록 허락하도록 요청한다. 기부된 객체를 사용하려고 할 때 다른 트랜잭션이 사용한 후 기부한 객체들의 자취(wake)에 따라 그 트랜잭션이 기부하거나 사용했던 객체들에 대해서만 사용 요청을 할 수 있다. 각 트랜잭션들은 잠금하고 있던 객체들만을 기부할 수 있고 이미 기부된 객체에 대해서는 또 다시 처리를 할 수 없다. 즉, 기부된 객체를 처리하고자 하는 트랜잭션에 대해서 제약조건을 부여한다. 2PL에서와 마찬가지로 AL 또한 트랜잭션들이 동시에 동일한 객체에 대해서 잠금을 할 수 없다는 규칙을 지키는 프로토콜이다.

기부된 객체를 사용한 트랜잭션은 기부한 트랜잭션의 자취(wake)를 온전히 따라야 한다는 제약을 감화하여 그 자취에 속하지 않은 객체로의 요청을 가능케 하기 위해 고안된 잠금기법이 확장형 이타적 잠금기법(Extended Altruistic Locking: XAL)이다.

3.2. Extended Altruistic Locking Protocol (XAL)

이 기법은 장기트랜잭션에 의해 쓰이지 않을 객체가 장기트랜잭션의 자취리스트(wake-list)에 포함될

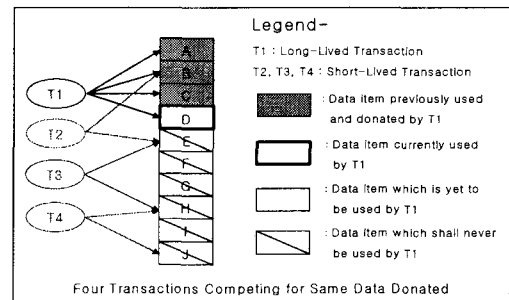
수 있다는 것이다. 그러한 처리를 위해서, XAL에서는 트랜잭션이 어떠한 객체를 처리할 것인가에 대한 정보를 가지고 있어야 한다[3].



< 그림 2. XAL의 기부에 의해 수행되는 예 >

장기트랜잭션 T1은 객체 A, B, C, D, E, F를 처리하고자 하고, 단기 트랜잭션 T3은 B와 F를 요청한다고 가정하자. T1은 A, B, C를 사용 후 기부하였고, T3은 T1이 기부한 B를 잠금하였다. 이어서 T1이 앞으로 절대로 사용하지 않을 H에 대하여 잠금을 요청하였다. H는 T1의 자취에 포함되지 않으므로 T1은 기본적인 AL기법에서는 실행이 중단된다. 그러나 T1이 H에 대해서 확장형 기부를 한다면, T3은 중단없이 트랜잭션을 성공적으로 수행할 수 있다. 그러나 이와 같이 스케줄러로 하여금 다른 트랜잭션들에 대한 확장형 기부를 수행하도록 허용하는 것은 언제나 좋은 구상이라고는 볼 수 없다.

T1이 데이터 항목들, A, B, C, 그리고 D를 순차적으로 접근한다고 가정해 보자.



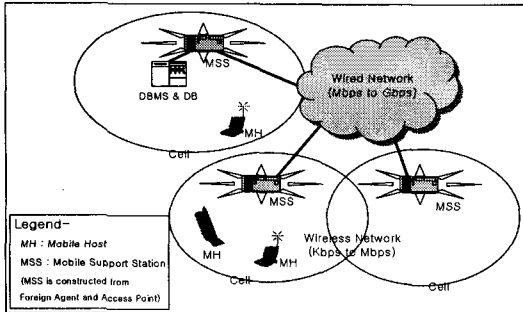
< 그림 3. Chain of Blockage >

만약 T2가 B보다 E를 먼저 요청하여 E를 얻은 후 B를 요청할 경우 E를 승낙할 수 없기 때문에 E를 얻지 못하고 T1이 모든 잠금을 해제하기를 기

다리게 된다. T3는 H를 얻은 다음, E를 얻기 위해 T2가 잠금해제하기를 기다리게 되고, T4또한 T3에 의해 기다리게 된다. 결국, chain of blockage가 만들어지게 되는 것이다. 본 논문에서는 others-first wake-later 방법론적 접근에서 기존 자취리스트에 다른 자취리스트를 포함시키는 방법으로 이 문제를 해결한다.

3.3 Mobile Computing Environment

모바일 환경이란, 기존의 분산 컴퓨팅 환경에서 떠나가 이동 호스트들의 일련의 작업들을 지원하기 위해서 확장 구성된 환경을 의미한다[5]. 모바일 환경을 간략히 도식화 하면 <그림 4>와 같다. 이동 지원국(Mobile Support Station: MSS)은 Home Agent(HA) 또는 Foreign Agent(FA) 그리고 Access Point(AP)로 이루어져 있다. 그리고 이동 호스트(Mobile Host: MH)는 모바일 유닛들을 의미한다.



< 그림 4. 모바일 환경 >

MH들은 각각의 셀을 담당하고 있는 MSS와 무선 컴퓨터 통신을 이용하며, MSS는 각 무선 통신 셀마다 하나씩 존재하여, 담당하고 있는 셀 내에 위치한 MH들의 관련 정보(소속정보, 보안 정보, 위치 정보 등)를 관리하고 있으며, MH가 다른 셀로 이동했을 때에는 관련 정보를 새로운 MSS로 옮겨서 이동 사용자에게는 아무런 부담 없이 정보 시스템을 이용할 수 있도록 Hand-off 작업을 수행한다. AP와 MH간의 통신, 정보관리등은 각 Agent들이 처리한다. 유선 네트워크에 비해 무선 네트워크는 전송속도와 대역폭이 현저히 낮기 때문에 잦은 접속 단절 등 많은 부분을 고려해야 한다. 무엇보다도 모바일 환경에서 동시성 제어는 트랜잭션의 대기시간이 길어짐에 따라, 불안정한 환경으로부터 발생 가능한 접속단절

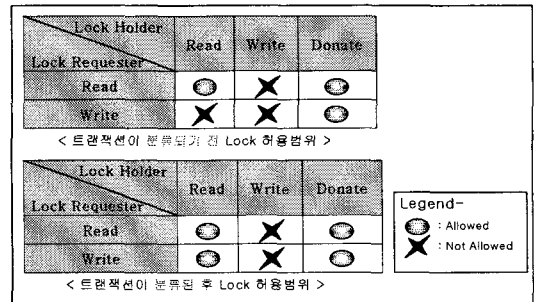
및 시간의 초과로 인한 철회와 같은 현상이 잦아질 수 있으므로, 그러한 제약을 가능한 줄여야 한다.

4. Proposed Protocol

4.1 Locking Protocol

4.1.1 2type Concurrency Control

2type 동시성 제어란, 분산화 환경에서 공유되는 데이터에 어떠한 기록 연산이 발생하면 기존의 데이터를 구버전으로 복사하고 새로운 데이터를 기록하는 다중버전의 한 예이다. 2-type을 사용해 판독 중 이던 트랜잭션에게 복사본을 판독하게 함으로써 데이터베이스의 일관성을 유지하도록 한다. 판독 트랜잭션은 주로 판독하기 위한 질의를 요청하는 트랜잭션으로서 기록연산은 이루어지지 않고 데이터베이스의 상태를 갱신하지도 않는다. 반면, 판독/갱신 트랜잭션은 판독과 적어도 한번 이상의 갱신연산을 하는 트랜잭션으로서 실행결과와 일관성(consistency)보장을 요하는 트랜잭션이다.



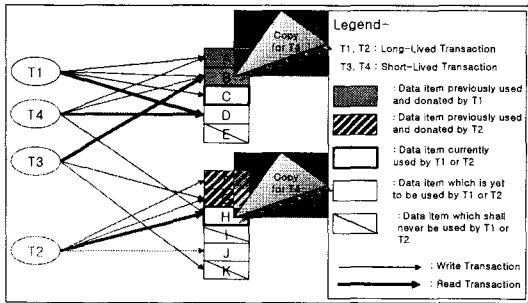
< 그림 5. 트랜잭션 분류에 따른 Lock 허용 범위 >

<그림 5>은 양방향 기부 가능한 XAL에서 트랜잭션을 분류하지 않았을 때의 잠금연산의 허용범위와 분류한 후의 허용범위를 보이고 있다.

4.1.2 Transaction Processing of TMXAL

TMXAL은 XAL의 확장으로서, 기부연산의 방향성을 증대하고 트랜잭션의 분류를 이용하여 트랜잭션처리 효과를 증대하고 있다. 판독전용 연산의 트랜잭션이 많은 환경에서 이 프로토콜은 그 성능의 효과를 극대화 할 수 있다. <그림 6>에서 트랜잭

션 T1, T2, T3, T4는 순차적으로 발생한다. T1은 A와 B를 성공리에 사용 후 기부하였다. T1은 객체 C에 갱신 연산을 하고 있는 중이다. T2는 F와 G의 실행을 성공리에 마치고 기부하였고 H에 관독 연산을 하고 있다. 기본 XAL에서는 T3가 G를 요청했을 때, 그 요청은 허락되어지지 않는다. T4의 경우, D객체는 T1이 앞으로 사용할 객체이므로 허락되어지지 않으며, H를 요청했을 때 이미 T2에 의해 사용 중인 객체이므로 요청이 허락되어지지 않는다.



< 그림 6. 제안 프로토콜의 트랜잭션 처리 >

제안 프로토콜(TMxAL)을 적용할 경우, 각 트랜잭션이 객체에 접근할 때 어떤 연산을 위한 접근인지, 그 정보를 가지고 있으므로, T3가 T1에 의해 기부된 B를 문제없이 얻은 다음, 객체 G를 요청할 때, 장기 트랜잭션 T2의 흔적리스트를 T1의 흔적리스트에 포함시킴으로써 T2가 기부한 객체 G의 요청은 허락되어진다. 객체 K 또한 XAL의 룰에 따라 사용 가능하므로, T3는 바로 실행가능하게 된다. T4가 요청하고자 하는 객체 B는 이미 T3가 관독전용 연산을 하고 있는 중이다. T4는 객체 B에 갱신연산을 하고자 하고 있다. 이 경우 T4의 처리를 위해 객체 B에 대한 복사본을 생성하게 되고, T3는 그 복사본을 사용하게 됨으로써 T4가 문제없이 객체 B를 사용하게 된다. 다음으로 객체 D에 관독전용 연산을 요청하게 되는데, 그 객체는 장기 트랜잭션 T1이 관독전용 연산을 하고 있는 중이므로 문제없이 D를 사용할 수 있다. 객체 H에 대한 T4의 요청 또한 객체 B로의 접근 연산과 마찬가지로, 요청과 동시에 복사본 생성에 따른 트랜잭션 처리를 할 수 있게 되므로, 위의 모든 트랜잭션은 동시에 신속하게 처리되어 질 수 있는 것이다.

모바일 환경에서는 이와 같이 장기 트랜잭션들과 동시에 발생하는 수많은 단기 트랜잭션을 신속하게

처리해야만 한다. 트랜잭션들은 모바일 환경에서 대기 시간이 길어질수록 접속단절 및 철회의 가능성이 높아지므로, 신속한 처리는 당연 중요한 필요조건이라 하겠다. 이와 덧붙여, 위의 <그림 6>의 예에서, 생성되게 되는 복사본들은 요청하는 주체(Subject)에게 주어짐으로써, 접속단절 환경에서 복사본을 이용하여 할 수 있는 로컬작업을 마치고 다음 객체로의 접근으로 재시작(Restart)할 수 있는 것이다.

5. Conclusions and Future Work

TMxAL은 기본적으로 모바일 환경에서 접속단절이라는 부분을 처리하였다. 기존의 XAL보다 트랜잭션의 처리 성능을 높이기 위해 고안되었다. 분산화된 모바일 환경에서 Hand-Off와 각 트랜잭션의 길이 및 우선순위, MH의 셀(Cell)에서의 상주시간, 그리고 접속단절의 단절률등에 따른 TMxAL의 처리에 관한 연구가 현재 전개되고 있으며, 보다 나은 결과를 얻으리라 본다.

참고문헌

- [1] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing" IEEE Computer, Apr. 1994
- [2] Kyong-I Ku and Yoo-Sung Kim, "Moflex Transaction Model for Mobile Heterogeneous Multi-database Systems," Proceedings of IEEE 10th International Workshop on Research Issues in Data Engineering, 2000
- [3] K. Salem, H. Garcia-Molina and J. Shands, "Altruistic Locking," ACM Transaction on Database System, Vol. 19, No. 1, pp. 117-169, March 1994.
- [4] P. A. Bernstein, V. Hadzilacos and N'. Goodman, "Concurrency Control and Recovery in Database Systems," Addison-Wesley, Massachusetts, U.S.A., 1987
- [5] Siwoo Byun, Songchun Moon, "Resilient data management for replicated mobile database systems," Data & Knowledge Engineering 29, 1999, pp. 43-55
- [6] H. Garcia-Molina, "Using Semantic Knowledge for Transaction Processing in a Distributed Database," ACM Trans. on Database System, Vol. 8, No. 2, pp. 186-213, June 1983.