

모바일 컴퓨팅 환경에서 실시간 데이터베이스 시스템을 위한 확장된 이타적 로킹 기법

강은영*, 김세윤*, 김응모*
*성균관대학교 컴퓨터공학과
e-mail : eykang@skku.edu

An Extended Altruistic Locking for Real-Time Database Systems in Mobile Computing Systems

Eun-Young Kang*, Se-Yun Kim*, Ung-Mo Kim*
*Dept. of Computer Science, Sung-Kwun-Kwan University

요 약

모바일 컴퓨팅 환경에서 실시간 데이터베이스 시스템의 트랜잭션 처리시 2PL-HP 방법은 성능이 우수한 스케줄링 기법이다. 그러나 2PL-HP 방법은 높은 우선순위를 갖는 트랜잭션의 선행처리를 항상 보장하기 때문에 낮은 우선순위를 갖는 트랜잭션은 철회 및 블로킹이 불가피하여 기아(starvation) 현상을 초래한다.

본 논문은 이러한 기아 현상으로 인한 낮은 우선순위를 갖는 트랜잭션의 기다림과 철회를 최소화 하기 위해 이타적 잠금 기법에 우선 순위 기법과 2 버전 기법을 이용하였다. 데이터항목을 점유한 트랜잭션이 더 이상 그 데이터항목을 필요로 하지 않을 때 그 데이터항목을 기부함으로써 우선 순위가 낮은 트랜잭션들의 불필요한 재 시작 및 기다림을 방지한다. 이로써 마감시간 초과비율과 재 시작 비율을 감소 시킬 수 있어 시스템의 처리율을 향상 시킬 수 있다.

1. 서 론

최근 이동 통신 기술 분야의 급격한 발전으로 모바일 컴퓨팅 환경에서의 실시간에 대한 연구가 이루어지고 있으며, 모바일 분산 실시간 데이터베이스시스템(mobile distributed real-time database systems MDRTDBS)에서의 동시성 제어와 트랜잭션 스케줄링에 대한 연구가 활발하게 진행되고 있다.

분산 실시간 데이터베이스(DRTDBS)에 대한 동시성 제어와 트랜잭션 스케줄링은 많은 연구가 진행되어 왔으나, 모바일 분산 실시간 데이터베이스(MDRTDBS)에서의 연구는 미비한 편이다. 모바일 컴퓨팅 환경은 제한된 대역폭과 잦은 단절 등과 같은 제약사항 때문에 높은 어려움을 갖는다. 모바일 실시간 데이터베이스에 다중으로 발생하는 트랜잭션을 처리하는데 있어 데이터의 일관성(data consistency)을 지키며 시간적 제약성(timing constraints)을 보장하는 스케줄링 기법이 필요하게 되었다. 본 논문에서는 모바일 환경에서 soft 나 firm 실시간 트랜잭션을 위한 모바일 분산 실시간

데이터베이스(MDRTDBS)에 대한 동시성 제어 기법에 대하여 제안하고자 한다. 모바일 환경에서 우선순위가 높은 장기 트랜잭션이 장기간 데이터를 점유하고 있어 우선순위가 낮은 단기 트랜잭션이 소외된다면 동시성 정도가 감소 되는 현상은 자명한 일이다. 모바일 환경에서와 같이 실제로 동시성 정도가 증가될 필요가 있는 경우, 서비스 중단이 발생되기 전에 우선순위가 낮은 단기 트랜잭션의 대기시간을 최소화하여 처리해 줌으로써 대기현상을 줄일 수 있다.

본 논문에서 제안하는 동시성 제어 알고리즘은 트랜잭션이 점유한 데이터항목을 사용 후 기부함으로써 기존의 2PL-HP 에서의 불필요한 기다림과 철회를 줄여 동시성을 향상시킨다. 또한 실시간 시스템의 특성을 고려하여 관독전용 트랜잭션에 대하여 2-Version 기법을 적용하여 동시성 정도를 향상시킨다.

본 논문의 구성은 다음과 같다. 우선 2 장에서 기존의 동시성 제어를 위한 여러 가지 기법들을 살펴보고 3 장에서는 본 논문에서 제안한 기법들에 대하여 설명하고 이에 대한 타당성을 알아본다. 4 장에서는 결론

을 맺고 향후 연구과제를 제시한다.

2. 관련 연구

2.1 2PL-HP(Two-Phase Locking High Priority)

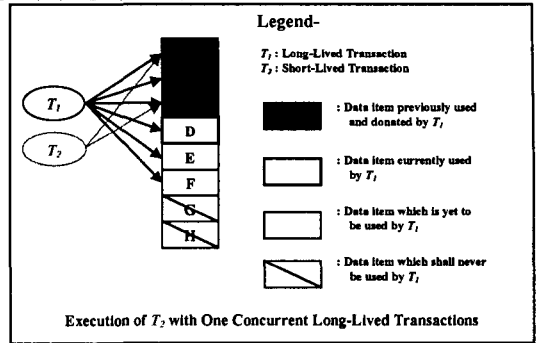
2PL-HP 는 기존 데이터베이스에서 가장 일반적인 방법인 2 단계 잠금 기법(Two-Phase Locking)에서 발전된 방법이다. 기존의 2 단계 잠금 기법의 경우 우선순위 역전(Priority Inversion) 및 교착상태(Deadlock)가 발생하는 문제점을 갖는다. 2 단계 잠금 기법은 우선순위를 전혀 고려하지 않기 때문에 실시간 데이터베이스에서는 적합하지 못하다[2]. 2PL-HP 방법은 충돌 발생 시 트랜잭션들의 우선순위를 비교하여 낮은 우선순위를 갖는 트랜잭션(Low Priority Transaction: LPT)을 철회시키고, 높은 우선순위를 갖는 트랜잭션(High Priority Transaction: HPT)이 수행되는 현재 가장 일반적인 방법이다.[1] 이 프로토콜은 낮은 우선순위를 갖는 트랜잭션을 철회시킴으로써 우선순위 역전 문제를 막는다.[3]

2.2 이타적 잠금 기법(Altruistic Locking Protocol(AL))

이타적 잠금 기법(Altruistic Locking)은 2 단계 잠금 기법(Two-Phase Locking)을 확장한 것이다. 이타적 잠금 기법은 잠금(lock), 해제(unlock)와 더불어 기부(donate)라고 하는 새로운 동시성 제어 연산을 제공한다. 해제와 마찬가지로 기부는 데이터항목을 잠금한 트랜잭션이 더 이상 그 데이터항목을 필요로 하지 않을 때 데이터베이스 관리자에게 그 데이터항목을 다른 트랜잭션들이 사용할 수 있도록 허락하도록 요청한다. 즉 기부연산과 잠금 연산은 2PL 로 연산될 필요가 없다. 이타적 잠금 기법은 장기 트랜잭션이 잠금한 데이터항목이 더 이상 사용되지 않을 것이라고 판단이 되면 그 데이터항목은 미리 잠금을 풀어 주어 다른 트랜잭션으로 하여금 그 데이터항목을 사용할 수 있도록 하는 것이다. 조기해제(즉, 기부연산)를 통한 대기 시간을 줄이는 효과를 보이고 있다. 또한 이타적 잠금 기법은 트랜잭션의 직렬성(serializability)을 보장하고, 트랜잭션들이 데이터베이스를 순차적으로 처리되도록 강요하는 등의 자료의 처리 방법에 대한 제약 조건이 없다. 한 트랜잭션이 다른 트랜잭션이 완료되기 전 사용하였던 데이터항목을 사용하려면 반드시 그 트랜잭션이 사용한 후 기부한 데이터항목들의 자취리스트(wake_list)를 따라 그 트랜잭션이 기부하거나 사용했던 데이터항목들에 대해서만 사용 요청을 할 수 있다. 기부된 데이터항목의 사용이 수락되면 해당 데이터항목을 사용할 수 있으며, 이 방법에 어긋나는 모든 트랜잭션들은 대기상태에 들어가게 된다. 만약 두 트랜잭션이 수행하려는 데이터항목들이 완전히 중복되지 않는다면 대기 시간을 현저히 줄일 수 있다. [8]

(그림 1)에서와 같은 경우 단기 트랜잭션인 T_2 는 두 단계 잠금 기법(2PL)에서는 데이터항목(A,C)을 사용하기 위해 장기 트랜잭션인 T_1 이 모든 잠금을 해제할 때 까지 기다려야 한다. 반면에 이타적 잠금 기법에서는 T_2 가 T_1 이 기부한 데이터항목들만을 요청하므로

T_2 는 바로 그 데이터항목들에 잠금을 할 수 있고 실행 되어 질 수 있다.

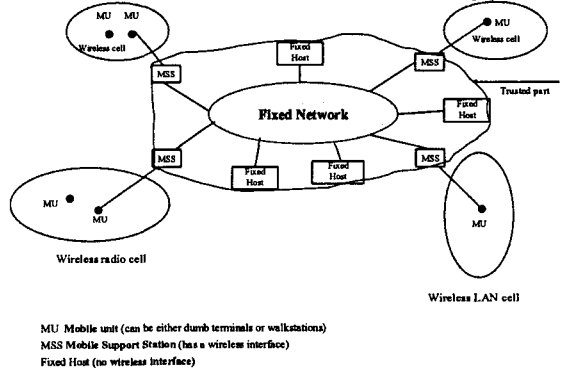


(그림 1) 이타적 잠금 기법의 예

트랜잭션들은 데이터항목들에 대해서 반드시 기부를 해야 할 필요는 없다. 즉, 기부할 것인지 여부는 선택할 수 있는 것이다. 기부연산은 어떠한 트랜잭션이 해제를 하기 전에 데이터항목을 기부한다. 그러므로 다른 트랜잭션으로 하여금 기부된 데이터항목을 잠금 할 수 있도록 해주기 때문에 매우 유용하게 사용된다.

2.3 모바일 환경

모바일 환경을 도식화 하면 (그림 2)과 같다.[9]



(그림 2) 모바일 환경

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 유닛을 의미한다. 이동 호스트 들은 각각의 셀을 담당하고 있는 이동 지원국과 무선 컴퓨터 통신을 이용하며, 이동지원국은 각 무선 통신 셀마다 하나씩 존재하여, 담당하고 있는 셀 내에 위치한 이동호스트들의 관련 정보(소속정보, 보안 정보, 위치정보 등)를 관리하고 있으며, 이동호스트가 다른 셀로 이동했을 때에는 관련 정보를 새로운 이동지원국으로 옮겨서 이동 사용자에게는 아무런 부담 없이 정보 시스템을 이용할 수 있도록 Hand-off 작업을 수행한다. 유선 네트워크에 비해 무선 네트워크는 전송속도와 대역폭이 현저히 낮기 때문에 잦은 접속 단절 등 많은 부분을 고려해야 한다. 무엇보다도

모바일 환경에서 동시성 제어는 트랜잭션의 대기 시간이 길어짐에 따라 불안정한 환경으로부터 발생 가능한 접속 단절 및 시간의 초과로 인한 철회와 같은 현상이 잦아질 수 있으므로, 그러한 제약을 가능한 줄여야 한다.

3. 제안하는 동시성 제어

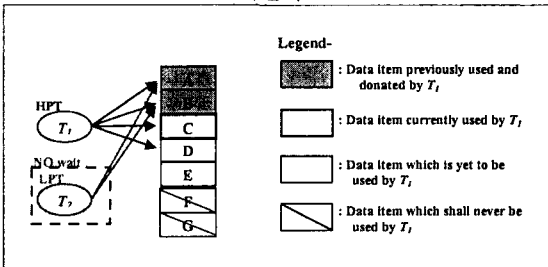
실시간 데이터베이스 시스템에서 2PL-HP 를 사용할 경우 발생할 수 있는 성능 저하 요인을 해소하고자 한다. 첫 번째, 우선순위가 높은 장기 트랜잭션에 의해 우선순위가 낮은 단기 트랜잭션이 오랜 시간 대기해야 하는 문제가 발생하게 된다. 두 번째, 우선순위가 낮은 장기 트랜잭션은 우선순위가 높은 단기 트랜잭션에 의해 계속적으로 철회 되는 기아 현상(starvation)이 발생할 수 있다.

3.1 동시성 제어 Protocol (AL-HP)

본 논문에서는 위와 같은 문제를 좀 더 해소하고자 이타적 잠금 기법에 우선순위기법을 이용하였다.

예제에 관한 설명

- $T_1 \rightarrow T_2 \rightarrow T_3$ 의 순서로 트랜잭션이 실행된다.
- T_1 은 장기 트랜잭션(Long-lived transaction)이고 T_2 와 T_3 은 단기 트랜잭션(Short-lived transaction)이다.
- 각 트랜잭션은 우선순위가 있으며 간단하게 $HPT > MPT > LPT$ 순으로 표기한다.



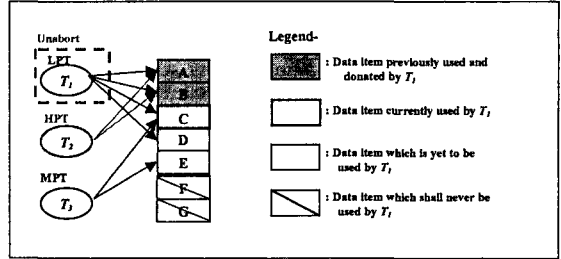
(그림 3) 예제 1

첫 번째는 기존의 2PL-HP 의 경우 (그림 3)과 같은 상황에서 우선순위가 높은 장기 트랜잭션인 T_1 에 의해 우선순위가 낮은 단기 트랜잭션인 T_2 는 T_1 이 종료할때 까지 대기해야 한다. 하지만 본 논문에서 제안한 프로토콜의 경우 T_2 가 T_1 이 기부한 데이터항목만을 사용할 경우 T_2 는 T_1 이 종료하기를 기다리지 않고 데이터항목에 접근 트랜잭션을 수행 먼저 종료함으로써 성능향상의 효과를 볼 수 있다.

두 번째는 기존의 2PL-HP 의 경우 (그림 4)와 같은 상황에서 장기 트랜잭션인 T_1 은 그보다 우선순위가 높은 단기 트랜잭션인 T_2 에 의해서 철회 되어야만 하는 상황이다. 하지만 본 논문에서 제안한 프로토콜은 T_1 이 기부한 데이터항목(A,B)만을 사용하는 T_2 는 T_1 을 철회하지 않고 데이터항목을 사용하고 트랜잭션을 먼저 종료할 수 있다.

하지만 T_2 가 종료되고 T_1 이 종료되기 전에 T_1 보다 우선순위가 높은 T_3 가 T_1 의 데이터항목(C)를 요구하

게 되면 T_1 이 철회되며 T_2 까지 철회해야 하는 문제가 발생한다.



(그림 4) 예제 2

이는 상대적으로 우선순위가 낮은 T_3 에 의해서 우선순위가 높은 T_2 가 철회되는 심각한 문제이다. 이에 트랜잭션을 판독전용과 갱신트랜잭션으로 분류하여 판독전용 트랜잭션에 판독과 갱신을 동시에 처리할 수 있도록 2-Version 방식을 적용하였다.

3.2 판독전용 트랜잭션을 위한 2-Version 동시성 제어(2VAL-HP)

판독전용 트랜잭션(Read-only Transaction : ROT)은 어떠한 데이터 항목에 대해서도 오직 판독 연산만을 수행하고, 결코 갱신 연산을 수행하지 않는 트랜잭션이다. 판독전용 트랜잭션은 다량의 데이터 항목에 대해 장기간 판독 연산만을 수행하는 성질을 가지기 때문에 갱신 트랜잭션의 수행을 상당히 오랜 기간 동안 지연시키거나, 갱신 트랜잭션에 의해 쉽게 취소될 수 있다. 따라서, 판독전용 트랜잭션을 갱신 트랜잭션과 구분하여 처리하게 되면, 데이터 충돌을 감소시켜 시스템의 병행 수행 정도를 높여준다.

증권시장에서 증권의 시세를 원하는 사용자의 요구와 레이더 추적 시스템에서 비행기의 움직임을 추적하는 등과 같이 여러 실시간 응용에서 판독전용 트랜잭션이 높은 비중을 차지하고 있다. 그러나, 실제계의 판독전용 트랜잭션은 우선순위가 낮고 장기간 실행되며 갱신 트랜잭션은 우선 순위가 높은 것이 특징이다. 이때 데이터 충돌이 발생하면 우선순위가 낮은 판독전용 트랜잭션은 취소 될 수 있다. 이것은 실시간 데이터베이스의 성능을 저하시키는 중요한 요인이다.

본 논문에서는 앞의 프로토콜(AL-HP)을 이용하였을 경우 판독 전용 트랜잭션과 판독되어 발생할 수 있는 문제점을 크게 다음 두 가지로 지적한다. 첫째, 우선순위가 낮은 장기 판독전용 트랜잭션은 우선순위가 높은 갱신 트랜잭션에 의해 자주 취소된다. 둘째, 우선순위가 높은 장기 판독전용 트랜잭션은 우선순위가 낮은 갱신 트랜잭션을 취소시킨다.

이에 동시성 정도를 높여 시스템의 재시작 비율의 성능을 향상시키기 위해 트랜잭션을 분류하여 판독전용 트랜잭션에 2-Version의 성질을 적용하였다. 다중버전에 기초를 둔 프로토콜들 중에서 2V2PL(2-Version 2 Phase Locking)은 공인버전(consistent version)과 작업버전(working version)이라는 2 개의 버전을 가진다.

[성질 1] 모든 트랜잭션 T_i 에 대해서, T_i 의 모든 판

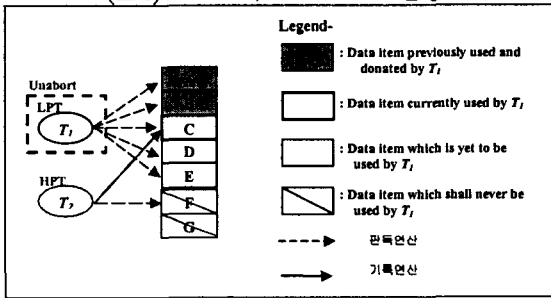
독역산 혹은 기록역산들은 공인역산보다 먼저 수행되며 T_1 의 공인역산은 완료되기 전에 수행된다. 2V2PL 프로토콜은 판독(Read), 기록(Write), 공인(Certify) 로크를 가진다. 판독역산은 데이터 항목을 반드시 공인버전으로부터 읽는다. 기록역산은 새로운 데이터를 작업버전에 기록한다. 트랜잭션은 완료하기 전에 공인로크를 취하여 작업버전에 있는 데이터 항목을 공인버전으로 변경하여야만 한다. 따라서 각 데이터 항목은 작업버전을 갖게 되지만, 직렬화를 만족시키기 위해서 트랜잭션들은 항상 공인된 공인버전의 데이터 항목들만 판독한다.

(표 1)은 2V2PL 와 2VAL-HP 프로토콜에서의 호환성 표이다.

$T_R \backslash T_H$	R	W	C
R	Y	Y	N
W	Y	N	N
C	N	N	N

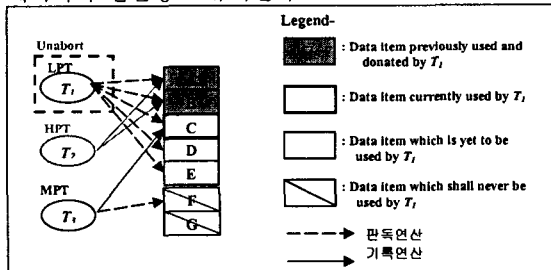
T_H : 로크소유트랜잭션
 T_R : 로크요구트랜잭션
 R: 판독역산
 W: 기록역산
 C: 공인역산
 Y: 공유가능
 N: 공유불가

(표 1) 2V2PL 와 2VAL-HP 호환성 표



(그림 5) 예제 3

다음과 같은 예제 3의 경우 우선순위가 낮은 장기 판독전용 트랜잭션 T_1 은 우선순위가 높은 단기 갱신 트랜잭션 T_2 에 의하여 취소되거나 철회되어야 한다. 그러나 2VAL-HP의 호환성 표에 의하여 기존의 데이터를 판독하고 있는 T_1 은 T_2 에 의하여 철회되지 않으며 갱신 트랜잭션 T_2 는 새로운 작업버전에 데이터를 기록한다. 이때 판독전용 트랜잭션 T_1 은 공인버전에 있는 데이터를 판독함으로써 계속적으로 수행을 완료하여 데이터의 일관성도 유지한다.



(그림 6) 예제 4

예제 4의 경우 낮은 우선순위의 장기 판독전용 트랜

잭션 T_1 은 높은 우선순위인 단기 갱신 트랜잭션 T_2 에 의해 취소되어야 하나, T_1 에 의해 기부된 데이터항목(A,B)을 사용함으로써 T_1 을 철회하지 않고 먼저 수행을 종료하였다. 하지만 T_2 가 종료되고 T_1 이 종료되기 전에 T_1 보다 우선순위가 높은 갱신 트랜잭션 T_3 가 T_1 의 데이터항목(C)을 요구하게 되면 T_1 이 철회되며 T_2 까지 철회해야 하는 문제가 발생한다.

이는 상대적으로 우선순위가 낮은 T_3 에 의하여 우선순위가 높은 T_2 가 철회되는 심각한 문제이다. 그러나 2VAL-HP의 호환성 표에 의하여 T_3 는 새로운 작업버전에 데이터항목(C)을 기록하며, T_1 은 취소되지 않고 계속적으로 공인버전의 데이터항목(C)을 계속적으로 판독 함으로써 데이터의 일관성도 유지한다.

지금까지 살펴본 바와 같이 본 논문에서 제안하는 동시성 제어 알고리즘은 우선순위 역전현상을 발생시키지 않으며 낮은 우선순위의 트랜잭션을 항상 철회하지 않음으로써 기아 현상을 완화한다. 또한 데이터베이스의 직렬성을 항상 보장한다.

4. 결론 및 향후 과제

본 논문에서는 모바일 환경에서 분산 soft 나 firm 실시간 데이터베이스에 우선순위에 기반한 이타적 로킹 기법을 적용하는 동시성 제어 방법을 제안하였다. 장기 트랜잭션이 해당 데이터항목을 오랜 기간 로킹하여 성능이 저하되는 것을 기부역산을 통해 단기 트랜잭션을 처리할 수 있도록 하였다. 우선순위가 낮은 단기 트랜잭션들의 불필요한 재 시작 및 기다림을 방지함으로써 시스템의 처리율을 향상시킬 수 있으며, 마감시간 초과와 재 시작 비율을 감소하여 동시성 정도를 높이는 효과를 가져온다. 향후 과제는 제안된 본 기법에 대한 성능 평가와 이를 유비쿼터스 환경에 적용하는 것을 연구 중에 있다.

참고 문헌

- [1] Abott R., Garcia-Molina H., "Scheduling Real-Time Transactions: A Performance Evaluation", *International Conference on Very Large Data Bases*, 1-1, 1988
- [2] Bernstein P.A., Hadzilacos V., Goodman N., *Concurrency Control and Recovery in Database Systems*, Addison-Wesley, 1987
- [3] Sha, L., Rajkumar, R., Lehoczky, J., "Concurrency Control for Distributed Real-Time Databases" *ACM SIGMOD Record*, vol.17,82-98, 1988
- [4] Sha, L. Rajkumar, R., Lehoczky, J., "Priority Inheritance Protocols: An Approach to Real-Time Synchronization" *IEEE Transaction on Computers*, vol.39, 1175-1185, 1990.
- [5] L. Sha, R. Rajkumer, S.H. Son, and C.H. Chang, "A Real-Time Locking Protocol," *IEEE Transaction on Computers*, Vol. 40, No. 7, pp. 793-800, July 1991.
- [6] Ulusoy, O., Belford, G. G., "Real-Time Lock Based Concurrency Control in a Distributed Database System", *International Conference on Distributed Computing Systems*, 136-143, 1992
- [7] Ulusoy, O., "Lock-Based Concurrency Control in Distributed Real-Time Database Systems", *Journal of Database Management* vol. 4, 3-16, 1993.
- [8] K. Salem, H. Garcia-Molina and J. Shands, "Altruistic Locking," *ACM Transaction on Database System*, Vol. 19, No. 1, pp. 117-169, March 1994.
- [9] Barbara, D., *Mobile Computing and Databases-A Survey*, Knowledge and Data Engineering, IEEE Transactions on, Vol. 11, pp. 108-117, Jan.-Feb. 1999