

무선 이동 네트워크 환경을 위한 트랜잭션 길이-감안 락킹 기법

김문정 김응모 엄영익
성균관대학교 전기전자및컴퓨터공학부
{tops, umkim, yieom}@ece.skku.ac.kr

Transaction Length-Sensitive Locking Scheme for Wireless Mobile Network Environments

Moon-jeong Kim Ung Mo Kim Young Ik Eom
School of Electrical and Computer Engineering, Sungkyunkwan Univ.

요 약

LT란 비교적 오랜 시간을 필요로하는 트랜잭션으로, 현재 LT들이 갖는 문제점들을 고려한 많은 기법들이 제안되어 있다. 무선 이동 네트워크 환경에서는 이동 호스트들이 대부분 ST를 발생시킬 확률이 높으며, 따라서 ST의 성능을 향상시키기 위한 AI, XAL등의 기존 알고리즘들이 유용하게 사용될 것이다. 그러나 이러한 환경 내에 존재하는 호스트들은 단절상태를 자주 경험하게 되며, 락을 소유한 이동 호스트들이 단절상태로 인해 필요이상으로 오래 락을 소유하게 될 수 있으므로 다른 ST들의 데이터베이스 객체에 대한 접근을 방해하는 결과를 초래할 수 있다.

본 논문에서는 기존의 데이터베이스 관리기법을 가능한 한 그대로 이용하면서, 무선 이동 네트워크 환경의 특징인 단절상태로 인한 성능저하를 최소화하기 위한 기법을 제안하고자 한다. 특히 향후 데이터베이스 응용 서비스 요구 확대가 기대되므로 보다 유용한 기법이 될 것이다.

1. 서론

무선 이동 네트워크 환경(wireless mobile network environments)이란, 무선망을 이용하여 노트북, 팜탑과 같은 휴대용 컴퓨터를 통해 사용자의 이동 중에도 네트워크 서비스를 제공받을 수 있는 환경을 말하며, 현재 이러한 환경을 위한 다방면의 연구가 활발히 진행중이다[1]. 특히, 최근 무선통신 기술 및 이동통신 기술과 하드웨어 기술의 급속한 발전으로 무선 이동 네트워크 환경에서의 데이터베이스와의 연동 및 멀티미디어 데이터베이스의 응용 서비스 분야가 큰 관심 분야가 되고 있는 실정이다.

LT(long transaction)란 많은 데이터베이스 객체들을 접근하거나, 긴 계산을 수행하거나 사용자로부터 입력을 받기 위해 기다리는 등의 이유로 비교적 오랜 시간을 필요로하는 트랜잭션을 말한다. 일반적으로 2PL을 사용할 경우, 이러한 LT들은 보다 일반적인 ST(short transaction)에 의한 데이터베이스 객체들의 접근을 방해하고 심지어 교착상태(deadlock)을 초래하기도 하므로 심각한 문제를 야기시킬 수 있으며, 현재 이러한 LT들을 고려한 여러 가지 기법들이 제안되어 있다[2,3,4,5,6].

본 논문에서는 기존의 ST 성능 향상을 위한 데이터베이스 관리기법들을 가능한 한 그대로 이용하면서 무선 이동 네트워크 환경의 특징인 단절상태로 인한 성능저하를 최소화하기 위한 기법을 제안하고

자 한다.

본 논문의 2절에서는 기존의 락킹 기법들에 대해 간단히 소개하며, 3절에서는 제안하는 락킹 기법을 소개한다. 4절에서는 본 제안 기법을 간단히 평가하며, 5절에서 결론을 맺는다.

2. 관련연구

데이터베이스 시스템에서 가장 중요한 문제 중의 하나는 동시성 제어(concurrency control) 문제이다. 특히 현재의 네트워크 환경이 분산 환경으로, 또한, 무선 이동 환경으로 확대될수록 여러 사용자들이 동시에 같은 데이터베이스 객체들을 접근하는 문제에 대한 보다 효율적인 해결 기법들이 요구된다.

2.1 2-단계 락킹(2PL:2-Phase Locking) 기법

이 기법은 가장 대중적으로 적용되는 락킹 기법이다. 이 기법에서, 데이터베이스의 각 객체들을 접근하려는 트랜잭션들은 먼저 해당 객체들에 대해 락을 얻은 후에 접근이 가능하며, 트랜잭션이 종료되기 전에는 반드시 락을 해제시키고 나서 종료되어야 한다[5]. 이 기법은 구현이 단순하다는 장점을 가지는 반면, 교착상태의 위험성을 갖는다. 현재 여러 가지 변형 기법들이 많이 연구되고 있으며, 특히 엄격한 2-단계 락킹(Strict 2PL)기법이 가장 일반적으로 사

용된다.

2.2 이타적인 락킹(AL:Altruistic Locking) 기법
이 기법은 2PL의 단순한 확장 기법으로, LT에 의해 락킹된 객체들 중에 해당 LT에 의해 더 이상 접근되지 않을 객체에 대해서는 해당 LT가 완료(commit)되기 전에, 다른 ST에게 락킹을 허용할 수 있도록 하는 기법이다. 즉, 특정 조건하에서 하나의 객체에 대해 하나 이상의 트랜잭션들이 락을 소유할 수 있도록 함으로써 동시성을 향상시키려는 기법이다[6].

2.3 확장된 이타적인 락킹

(XAL:Extended Altruistic Locking) 기법

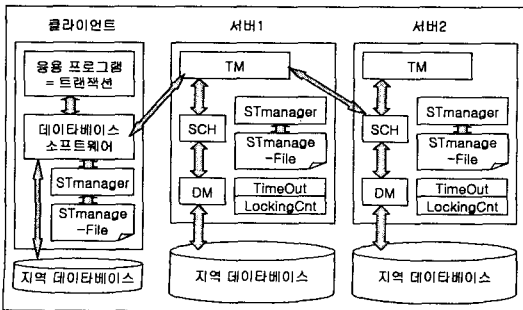
이 기법은 AL의 확장기법이며, 동시성을 보다 향상시키기 위한 기법이다[6]. 이 기법에서는 Lock, Unlock, Donate와 함께 Mark라는 새로운 연산자를 추가로 제공한다.

3. 제안 기법

3.1 개요

무선 이동 네트워크 환경에서의 이동 호스트들은 기존의 유선 네트워크 환경에 비해 잦은 단절상태를 경험하게 된다. 또한 무선 이동 네트워크 환경의 특성상, 이동 호스트들의 데이터베이스에 대한 접근시 대부분이 ST를 발생시킬 것이다. 이동 호스트 내에서 수행되는 ST가 락(lock)을 소유한 상태에서 단절상태에 놓이게 되면, 다른 호스트들에 의한 트랜잭션의 접근을 방해하는 결과를 갖게 될 것이며, 이는 ST의 성능을 향상시키기 위한 락킹 프로토콜(AL, XAL)들의 효과를 급격히 저하시키는 결과를 초래하게 될 것이다.

본 논문에서 제안하는 기법은, 이동 호스트들은 비교적 ST들을 발생시킬 확률이 높다는 가정하에서, 모든 서버들은 락을 요구하는 ST들에 대해 일정시간(Timeout)을 적용시켜, 락을 허용한 후 일정시간을 초과한 경우에 해당 객체에 대해 서버 임의로 락을 해제(unlock)할 수 있도록 하는 기법으로, 전체적인 시스템 모델은 그림 1에서 보인다.



(그림 1) 제안 기법의 시스템 모델

물론 단절상태를 겪은 후의 이동 호스트는 스스로 해당 트랜잭션이 락을 소유한 객체에 대해 일정시간이 초과했음을 알게 되므로, 해당 트랜잭션을 취소(abort)시키고 다시 시작(restart)하여야 한다.

3.2 요구사항

본 논문에서 제안하는 기법은 기존의 데이터베이스 시스템 구조를 가능한 한 그대로 따르며, 추가적으로 STmanager를 데몬으로 사용하여 본 논문에서 제안하는 주요 관리작업을 담당하도록 한다. 각 서버는 락을 허용한 ST에 관한 정보를 STmanageFile에 기록하여 관리하도록 하며, STmanageFile의 형식은 그림 2에서 보인다.

TransactionID	ObjectID	LockingTime	RemainedTime
---------------	----------	-------------	--------------

(그림 2) STmanageFile의 형식

필드 TransactionID는 해당 ST를 구분하기 위한 필드이며, 필드 ObjectID는 해당 ST에 의해 락킹된 객체를 구분하기 위한 필드로 사용된다. 필드 LockingTime은 락킹된 현재 시간을 기록하기 위한 필드이며, 필드 RemainedTime은 락킹이 허용된 일정시간을 기록하였다가 매초마다 감소시켜 0이 되는 시점을 감시하기 위해 사용된다. 각 서버의 STmanager가 사용할 전역변수들은 그림 3에서와 같다.

Timeout = 일정상수	// ST에 대해 허용될 락의 최대 시간
LockingCnt = 0	// 해당 서버에서 ST에 의해 락이 허용된 객체들의 수

(그림 3) STmanager가 사용할 전역변수들

3.3 동작과정

클라이언트가 자신의 응용프로그램을 실행하는 도중에 데이터베이스 연산을 발견하게 되면, 먼저 자신의 지역 데이터베이스 소프트웨어로 해당 연산을 요구하게 된다. 지역 데이터베이스 소프트웨어는 지역 데이터베이스를 이용하여 해당 연산을 수행하며, 만일 자신의 지역 데이터베이스에 없는 객체라면 가까운 원격 서버의 TM(Transaction Management)에게 요청하게 된다. TM은 자신의 지역 데이터베이스 내에 존재하는 객체인지를 확인하고, 존재한다면 자신의 지역 SCH(Scheduler)에게 요청하고, 그렇지 않으면 해당 데이터를 가진 다른 서버의 SCH로 요청하게 된다. SCH는 AL 또는 XAL 알고리즘을 수행한다고 가정한다. 해당 알고리즘에 적용된 연산은 DM(Database Management)으로 전달되거나 거절(reject) 또는 지연(delay)될 수 있다. DM은 지역 데이터베이스를 접근하여, 해당 데이터를 SCH에 전달하고, SCH는 그 결과를 TM에게 전달하게 된다. TM은 연산의 결과와 연산이 ST에 의한 락인 경우에 락킹이 허용되는 일정시간을 클라이언트의 데이터베이스 소프트웨어로 함께 전달하고, 본 논문에서의 새로운 기법을 적용시키기 위해 알고리즘 1에서와 같은 작업절차를 수행하는 STmanageProcedure를 실행한다.

(알고리즘 1) STmanageProcedure의 작업절차

```

// call from TM
{
  increase LockingCnt by one;
  if (LockingCnt equals one)
    run STmanager;
  insert (TransactionID, ObjectID)
    to STmanageFile;
  record current time to LockingTime;
  record TimeOut to RemainedTime;
}
// call from STmanager
if (LockingCnt equals zero)
  Kill STmanager;
    
```

TM으로부터 호출된 STmanageProcedure는 전역변수 LockingCnt를 1만큼 증가시킨다. LockingCnt가 1이라면 ST에 의해 락킹이 허용된 객체가 처음이므로 STmanager를 실행시키고, 또한 STmanageFile에 하나의 엔트리를 삽입한다. 만일 STmanager로부터 호출되는 경우라면, STmanageFile의 엔트리 하나가 삭제된 경우이다. 따라서 LockingCnt를 확인하여 0이면 STmanager를 제거하게 된다.

서버의 STmanager는 매초마다 STmanageFile 내의 각 엔트리가 갖는 필드 RemainedTime의 값을 1씩 감소시키다가 임의의 RemainedTime 값이 0이 되면 해당 엔트리를 STmanageFile에서 제거하고, 해당 객체에 대해 서버 임의로 락을 해제한 후, STmanageProcedure를 호출하게 된다.

4. 제안기법의 평가

본 논문에서 제안하는 기법은 ST에 대해서만 서버로부터의 강제적인 락킹 해제를 허용하는 기법으로, 단절상태로 인해 필요이상으로 락을 소유하게 되는 경우를 방지하기 위한 기법이다. 또한 본 논문에서 제안하는 기법은 기존의 ST 성능향상을 위한 데이터베이스 락킹 알고리즘들을 그대로 적용시킬 수 있으므로 구현이 간단하다는 장점을 갖는다.

이는 무선 이동 네트워크 환경의 특성상 이동 호스트들이 발생시키는 트랜잭션들의 대부분이 ST일 확률이 높으며, 데이터베이스 응용 서비스가 보다 확대되는 경우에 더욱더 많은 ST를 발생시키게 된다는 가정을 둔 기법이라 할 수 있다.

ST들은 각 객체에 대한 락을 소유하면서 락킹이 허용되는 최대 일정시간을 함께 전달받게 되므로, 고정 호스트 또는 단절상태가 아닌 이동 호스트의 경우에는 일정시간 이전에 스스로 락킹을 해제하거나 서버로 락킹에 대한 연장을 요구할 수 있다. 그러므로 강제로 락킹 해제를 겪게 되는 트랜잭션을 수행하는 호스트는 단절상태를 겪은 이동 호스트의 경우가 된다. 따라서 이동 호스트의 경우, 자신이 락을 소유한 객체가 있는 경우에 STmanager에 의해 서버로부터 락을 허용 받으면서 함께 받은 일정시간을 관리하다가 단절로 인해 일정시간이 초과된 경우에는 스스로 해당 트랜잭션을 취소한 후에 다시 실행시키도록 한다.

5. 결론

무선 이동 네트워크 환경에서는 다양한 연결상태가 존재하게 되며, 이러한 다양한 연결상태에 대해 적응성 있는 데이터베이스 응용서비스를 제공하기 위한 노력이 요구된다. 본 논문에서는 특히, 잦은 단절상태를 경험하게 되는 이동 호스트들에 의해 필요이상으로 오랫동안 락이 소유됨으로써 발생할 수 있는 동시성 저하를 방지하기 위한 기법을 제안한다.

본 논문에서 제안하는 기법은 기존의 ST 성능 향상을 위한 데이터베이스 관리기법들을 가능한 한 그대로 이용하면서 무선 이동 네트워크 환경의 특징인 단절상태로 인한 성능저하를 최소화하기 위한 기법이며 구현이 간단하다.

최근, 무선 이동 네트워크 환경에서의 데이터베이스와의 연동 및 멀티미디어 데이터베이스 응용 서비스 요구가 급속히 증가하는 실정으므로 보다 효율적인 락킹 알고리즘에 대한 많은 연구가 필요할 것이다.

참고문헌

- [1] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, Apr. 1994.
- [2] A. A. Farrag, M. T. Ozsu, "Using semantic knowledge of transactions to increase concurrency," ACM Transactions on Database Systems, Vol. 14, No. 4, pp. 503-525. Dec. 1983.
- [3] M. J. Carey, W. A. Muhanna. "The performance of multiversion cocurrency control algorithms," ACM Transaction Computing Systems. Vol. 4, No. 4. pp. 338-378, Nov. 1986.
- [4] M. L. Ahuja, J. C. Browne. "Concurrency control by pre-ordering entities in databases with multi-versioned entities," In Proceedings of the International Conference on Data Engineering. Los Angeles. pp. 312-321. Feb. 1987.
- [5] P. A. Bernstein, V. Hadzilacos, N. Goodman, "Concurrency Control and Recovery in Database Systems," Addison-Wesley, Massachusetts, U.S.A., 1987.
- [6] K. Salem, H. Garcia-Molina, J. Shands, "Altruistic Locking," ACM Transactions on Database Systems, Vol. 19, No. 1, pp. 117-169, March 1994.