論 文

자료 기지 관리를 위한 낙관적 동시성 제어 기법의 재시작 부담

RESTART OVERHEAD IN OPTIMISTIC CONCURRENCY CONTROL SCHEMES FOR DATEBASE MANAGEMENT

이 혜 경
(경인여자전문대학 사무자동화과)

◀ 차 례 ▶

요 약
Ⅰ. 개 요
Ⅱ. 관련연구
1. 원형기법에 의한 검증
2. 순간검증방법에 의한 검증
Ⅲ. 비교 및 분석
1. 재시작 부담
2. 검증에 소요되는 시간
Ⅳ. 결 론

요 약

자료 기지 관리 체계(Database Management System; DBMS)에서 실시간 처리를 만족시킬 수 있는 동시성 제어 기법 중의 하나가 낙관적 기법이다. 낙관적 기법은 카네기 멜론 대학교의 Kung에 의해 제안된 이래 여러 가지 변형이 시도되었다. 그러나 아직도 변형된 기법들에 대한 성능 차원의 비교가 이루어지지 않아서 실시간 체계 등에 적용시 기법의 선정에 어려움을 겪고 있다. 본 논문에서는 Kung이 제안한 원형 기법과 이의 대표적 변형 기법들의 근간이 되는 순간 검증을 이용한 기법을 비교하였다. 원형 기법은 거래의 직렬화 가능성을 파괴하지 않는 거래간의 충돌까지 철회되어야 하는 단점이 있기 때문에 충돌된 거래를 재시작하는 측면에서는 순간 검증 방법보다 불필요한 재실행을 해야 하는 만큼의 시간적 부담이 크다. 그러나 원형 검증 방법은 버퍼 보유 효과를 제공하므로 이를 제공하지 않는 순간 검증

기법보다 충돌된 거래의 재시작으로 인한 부담이 적다. 왜냐하면 충돌된 거래에 대한 실행시 필요한 모든 객체들이 이미 준비되어 있다면 거래의 재실행시 드는 시간적인 부담이 적기 때문이다. 처리 기간이 긴 장기거래 위주로 실행하는 환경에서는 순간 검증을 이용한 기법이 불리하였다. 왜냐하면 한 거래가 처리 단계에 있을 때 실행을 완료한 거래들의 수많은 순간 검증을 해야 하기 때문이다.

Ⅰ. 개요

분산 자료 기지 체계(distribute database system)는 지역적으로 분산된 자료 기지들의 집합을 컴퓨터 통신망으로 연결하여 개념적으로는 하나의 단일 자료 기지 체계와 같은 기능을 제공해 주는 시스템이다. 자료 기지 체계에서 사용자는 거래(transaction)를 이용하여 실제 자료 기지를 액세스한다. 여기서 거래한 도저히 더 이상 분리될 수 없이 한 단위로 수행되어야 하는 원자적인 연산, 즉 판독 연산 및 기록 연산들의 집합으로서 자료 기지 상태의 일관성(consistency)을 유지하는 단위 코드를 가르킨다. 다수의 사용자가 한 자료 기지를 공유하는 시스템에서는 동시성을 높이기 위해 거래를 끼어들기(interleaving)형태로 처리한다. 따라서 여러 거래의 연산들의 실행 순서가 섞여 버리는 경우가 존재하게 된다. 이렇게 여러 연산들이 혼합되어 실행되는 경우에 연산의 실행을 제어하는 기능이 DBMS에 존재하지 않으면 자료 기지는 비일관적 상태에 돌입하는 잘못된 결과를 초래할 수도 있다. 이러한 문제점을 해결하는 제어 기능을 동시성 제어라 한다. 다르게 표현한다면, 동시성 제어란 동시에 자료 기지를 액세스하는 여러 사용자들에게 전체 자료 기지를 마치 혼자 사용하는 것과 같은 효과를 제공해 주는 것이다.

점차적으로 CPU와 I/O장치의 처리 속도가 빨라짐에 따라 DBMS는 시스템에 대한 동시성의 정도를 증대시킬 수 있어야 한다. DBMS에 대한 이러한 요구가 자료간의 분쟁을 심화시킬 가능성이 있다. 오늘날의 DBMS에서는 아주 보편적으로 발생하는 실행 시간이 긴 장기거래, 또는 컴퓨터 지원 설계 환경에서 발생하는 복합 거래(complex transaction)로 인하여 자료간의 분쟁이 심하게 발생될 수 있다. 이러한 점을 해결하기 위해 "엑세스 불변화"(access invariance)방식[FRAN90]을 이용하려는 새로운 경향이 대두되고 있다. 엑세스 불변화란 재시작되는 거래는 그 거래가 처음 실행될 때처럼 같은 객체(object)들 안에서 같은 연산을 수행할 가능성이 높다는 것을 의미한다. 충분한 버퍼가 확보된다면 철회(abort)된 거래에 의해 참조된 객체는 기억 장치에 계속 머물 수가 있다. 결과적으로, 거래의 재실행시 다시 철회될 가능성을 획기적으로 줄일 수 있다. 이러한 것을 버퍼 보유 효과(buffer retention effect) [YU92]라고 칭한다.

이러한 개념의 도입으로 인하여 낙관적 동시성 제어 (optimistic concurrency control: OCC) [KUNG81]에 대한 새로운 관심이 모아지고 있다. 일반적으로 DBMS에서 동시성 제어 기법으로는 잠금(locking) 기법이 널리 알려져 있다. 잠금 기법은 다른 동시성 제어 기법보다 충돌이 자주 일어나고 거래의 길이가 길어서 거래의 철회와 복귀(rollback), 그리고 재시작

(restart) 에 대한 비용이 커지는 경우에 성능이 우수하다. 그러나 잠금 기법의 문제점은 동시성 정도의 경감과 교착 상태 (deallock) 를 해결하기 위해 부가적인 노력을 필요로 하고 있다. 따라서 실시간 상황에서 과연 잠금 기법을 사용해도 좋은지에 대한 현실적 가치에 대한 의문이 제기되고 있다. 낙관적 기법에 관심을 두었다.

낙관적 동시성 제어 기법은 거래간에 충돌 빈도 수가 작고 거래의 철회, 복귀, 그리고 재실행시 드는 비용이 적다는 가정하에서 거래의 실행 과정에서는 동시성 제어에 관한 행위는 취하지 않고 거래의 실행이 끝난 후, 작업 공간에서 이루어진 갱신 자료 집합(write-set)과 판독 자료 집합(read-set)을 가지고 거래의 정확성을 조사하는 기법이다. 거래가 처음 실행될 동안에 거래에 필요로 하는 객체들을 선채취(prefetch)함으로써 거래의 실행이 중단될 필요가 없다. kung이 제안한 occ 원형 기법을 변형한 기법[ROBI82,FRAN85,PRAD86]은 대부분 거래의 실행을 재시작함으로써 드는 부담을 줄이는데 주로 중점을 두고 있다. 이러한 기존의 변형 기법들은 충돌되는 거래들에 대해서 가능하면 빨리 간단히 철회시킴으로써 부담을 줄이고 있다.

그러나, 버퍼 보유 효과를 받아들인다면 현존하는 OCC 기법들에서 채택된 검증 방법(validation method)들은 다른 측면에서 검토되어야 한다. 즉, 거래의 실행 도중에 거래간의 충돌로 인하여 충돌된 거래들을 즉각적으로 소멸시키는 대신 기존의 OCC 기법과 같이 주기억 장치로 거래의 실행시 필요한 객체들을 가져와서 정상적으로 거래를 종료하는 방법이 더 효과적일 것이다. 왜냐하면 버퍼 보유 효과를 이용함으로써 충돌된 거래들에 대한 모든 필요한 객체들이 이미 주기억 장치에 머물 수가 있기 때문이다. 이런 관점에서 볼 때 거래의 실행이 종료된 후 직렬화된 기록들을 더욱 잘 인식할 수 있는 검증 기법을 고안해야만 한다.

Ⅱ. 관련 연구

원형 기법을 포함한 OCC 기법들은 기존의 전통적인 잠금 기법에서 지불해야 할 잠금 테이블 유지 부담과 교착 상태의 처리에 대한 부담을 제거하기 위하여 연구되었다. 이러한 기법들은 판독(read), 검증(validation), 기록(write)의 세 단계로 구성된다. 거래들은 판독 단계 동안에 아무 제약 없이 수행이 되지만 거래가 완료되기 전에는 데이터베이스의 일관성을 확실히 보존할 수 있는지 반드시 검증되어야 한다. 만일 검증 작업에 실패하면 충돌을 해결하기 위한 전략이 적용된다.

1. 원형 기법에 의한 검증

OCC 원형 기법에서의 거래는 판독, 즉 거래의 코드가 실행 과정에서는 동시성 제어를 위한 아무런 검증 조치도 취하지 않고 다른 거래들의 방해를 받지 않고 실행되며, 처리 내용은 각 거래의 작업 공간에 저장하고, 실행을 완료하기 직전에 이 실행 결과를 검증한다. 직렬화 가능한 실행(serializable execution)을 해온 거래의 실행 결과는 실제 자료 기지에 기록하고, 직렬화 불가능한 실행(nonseriallizable execution)을 야기한 거래는 재시작하게 된다.

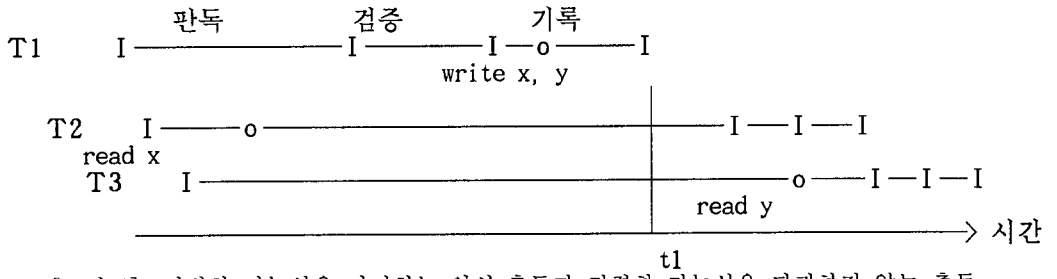
kung의 원형 검증 방법에서 세 단계, 즉 판독/검증/기록의 세 단계를 거쳐 실행을 완료(commit)하게 된다. 판독 단계에서는 필요한 데이터 항목을 읽어서 판독 자료 집합을 만들면서 거래를 실행하고, 실행 결과로 갱신 자료 집합을 작업 공간에 만든다. 검증 단계에서는 판독 단계에서 실행된 연산들에 대해 다른 거래들과의 직렬화 가능성을 조사하게 된다. 따라서 검증 단계에서 직렬화 가능성이 오직 보장된 거래만이 기록 단계에서 거래의 실행 결과를 실제 자료 기지로 이동하여 기록하게 된다. 검증 단계에서 직렬화 가능성이 보장되지 못한 거래는 작업 공간의 실행 결과를 무시해 버리며, 추후 재시작하게 된다.

```
알고리듬 1(Kung의 원형 검증 방법)
    valid := TRUE;
    FOR Tj FROM T(i-finish) DO
        IF read-set (Ti) ∩ write-set(Tj)=o
            THEN
                valid := FALSE;
    IF valid
      THEN write-phase 돌입 /* 기록 단계 실행, 거래 번호부여 */
      ELSE abort /* 철회 */
```

원형 검증 방법에 의하면 각 거래들은 기록 단계를 실행한 후에 고유 거래 번호(transaction number)를 부여받는다. 따라서 한 거래 Ti의 검증 단계에서는 Ti가 처리 단계의 실행을 완료한 거래 중에서 가장 작은 거래 번호를 갖는 거래(Ti_start라고 하자.)까지의 거래에 대해서 검증하면 된다. 본 원형 검증 방법에 있어서 거래간의 직렬화 가능성을 파괴하지 않는 충돌까지도 철회되어야 하는 면에서 비효율성이 있을 수 있다. 원형 검증 방법 알고리즘에서 살펴보면 거래간의 충돌을 검증하기 위해서 거래의 판독 자료 집합 전체를 가지고 검증하기 때문에 거래가 불필요하게 철회되는 경우가 존재한다. 다음 예 1은 이를 보여준다.

예 1(원형 검증 방법의 비효율성):

거래 T1이 자료 x와 y를 기록하며, T2가 x를 판독하고, T3가 y를 판독한다고 가정하자(그림 1). T1은 T2와 T3의 판독 단계에서 검색한 x와 y를 기록 단계에서 실제 자료 기지에 기록하였으므로 원형 검증 알고리즘을 사용하여 검증한다면 T2와 T3는 시간 t1에 모두 철회되어야 한다. 여기서 T2와 T1사이의 충돌은 T2가 T1에서 x의 값을 기록하기 전에 미리 판독을 했기 때문에 직력화 가능성을 파괴하는 심각한 악성 충돌이나, T3와 T1 사이의 충돌은 T3가 y의 값을 T1이 이미 자료 기지에 기록한 후에 판독을 했기 때문에 직렬화 가능성을 파괴하지 않는 충돌임을 관찰하게 된다. 따라서 T3는 철회되지 않아도 시스템의 전체적인 직렬화 가능성은 보장되나 알고리즘 1에 의하면 반드시 철회되어야 한다.



[그림 1] 직렬화 가능성을 파괴하는 약성 충돌과 직렬화 가능성을 파괴하지 않는 충돌

2. 순간 검증 방법에 의한 검증

원형 검증 방법이 가지고 있는 단점을 보완하기 위해 순간 검증(snapshot validation)을 이용한 OCC[SCHL86]가 제안되었다. 원형 검증 방법에서는 전술한 바와 같이 불필요하게 거래가 철회되는 경우가 있는데, 이러한 문제를 줄이기 위해 새로운 검증 방법을 고안하였다. 실행을 완료한 거래(Tj라 하자.)의 갱신 자료 집합과 판독 단계를 수행 중인 거래(Ti라 하자.)가 Tj의 수행이 됐음을 알리는 메시지가 받아 들여 진 순간까지 판독한 판독 자료 집합(순간 포착 장면(snapshot)이라고 하자.)만을 가지고 검증을 수행하게 된다. 따라서 한 거래가 판독 단계에 있을 동안 실행을 완료한 거래의 수만큼 순간 검증을 하게 되며, 최후의 검증 단계에서는 실행을 완료한 거래 중 순간 검증되지 않은 거래들에 대해서만 검증을 수행하게 된다.

위의 검증 방법을 사용함으로써 반드시 철회하여야 할 거래만을 철회하게 하며, 철회하여야 할 거래에 대해서는 판독 단계를 완전히 실행할 때까지 기다리지 않고 철회함으로써 시간과 자원의 낭비를 줄일 수 있다. 원형 검증 방법의 결점을 보완하기 위한 방법은 순간 검증 방법을 논하고 같은 동일한 상황하에서 각 검증 방법에 의해 실행되는 결과를 비교해 본다.

그림 1에서 순간 검증을 이용하는 OCC 기법에서는 거래 T1이 실행을 완료한 시각(T1이라고 하자.)에서 거래 T2와 T3가 각각 순간 검증을 실시하므로 T2와 T1사이의 충돌은 직렬화 가능성을 파괴하고, T3와 T1사이의 충돌은 직렬화 가능성을 파괴하지 않음을 알 수 있다. 따라서 순간 검증을 이용하는 OCC기법에서는 실행을 완료한 T1에 의해 T2는 철회되지만 T3는 철회되지 않는다.

직렬화 가능성을 파괴하는 악성 충돌과 직렬화 가능성을 파괴하지 않는 충돌을 구별하기 위해서 각 거래의 판독 자료 집합은 사용된 자료의 순서에 따라 저장되어 있어야 하며, 거래는 세 단계를 거쳐 실행을 완료하게 된다.

(1) 판독 단계: 필요한 자료를 읽어서 작업 공간에 판독 자료 집합을 만들고, 판독 자료 집합을 이용하여 거래를 실행하게 된다. 판독 단계 수행 도중에 다른 거래의 실행 완료 메시지가 들어오면 실행 중인 거래의 순간 포착 장면과 실행을 완료한 거래의 갱신 자료 집합을 가지고 순간 검증을 수행하게 된다.

(2) 검증 단계: 각 거래들은 거래 번호들 부여한다. 부여받는 거래 번호보다 작은 번호를 가지고 아직 실행 중에 있는 거래에 대해 검증을 실시한다.

(3) 기록 단계: 해당 거래의 작업 공간에 있는 거래의 실행 결과를 실제 데이터베이스에 기록하고, 거래의 실행이 완료했음을 실행 중인 다른 거래에게 통보해 줌으로써 다른 거래가 순간 검증을 실시하게 한다.

순간 검증을 이용하는 낙관적인 방법에서는 검증을 검증 단계에서는 한 번 하는 것이 아니라 처리 단계의 중간 어디에서나 실행을 완료한 다른 거래에 대해 실행 중인 거래의 순간 포착 장면을 가지고 순간 검증을 실시한다.

Ⅲ. 비교 및 분석

1. 재시작 부담

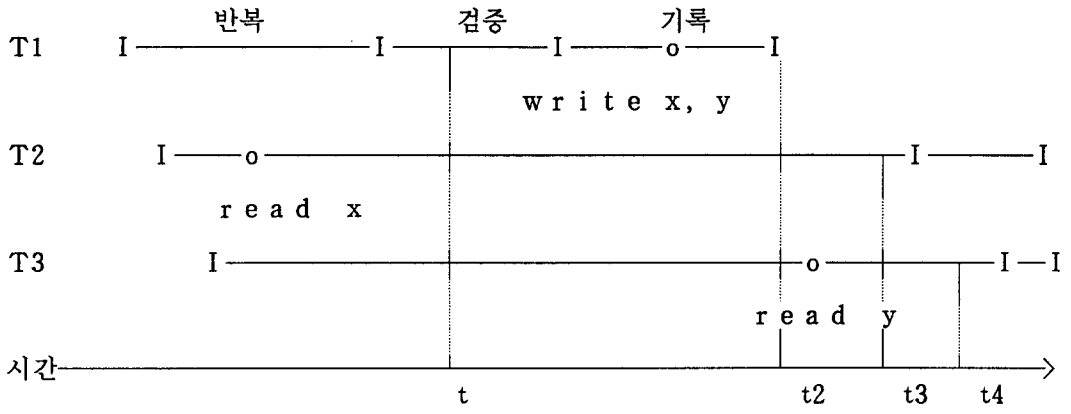
OCC 기법들은 각각의 검증 방법을 바탕으로 2가지의 부류 즉, "타인 피해"(kill-base) 검증 방법과 "자신 포기"(die-base) 검증 방법으로 분류된다. 타인 피해 검증 방법에서는 오나료되는 거래들은 추돌 되는 거래들이 즉시 재시작을 하도록 유도한다. 방송형 검증 방법 [ROBI82]은 이 부류에 속한다. 또한 한겨레와 충돌 상태에 있는 어$떤 거래가 이미 완료되었을 때 거래는 재시작될 수 있다. 원형 검증 방법과 순간 검증 방법들은 이 방식을 토대로 하고 있다. 그러나, 순간 검증 방법은 원형 검증 방법과 다른 점이 있다. 즉, 순간 검증 방법은 어떠한 거래 Ti가 종료될 때마다 모든 활동 중에 있는 거래들은 거래 Ti의 갱신 자료 집합에 대하여 각각 충돌되는 것이 없나 검증한다. 만일 어떤 거래가 검증 단계에서 실패를 했다면 그 거래는 스스로 소멸되고 만다. 그러므로 순간 검증 방법은 충돌된 거래들을 하나의 거래가 완료될 때마다 순간 포착 장면을 가지고 검색해서 거래의 실행 도중에 즉시 철회될 수 있다.

먼저 거래가 철회되어 재시작되는 빈도 수를 토대로 비교할 수 있다. 원형 검증 방법은 거래간의 충돌을 검증하기 위해 거래의 판독 자료 집합 전체와 갱신 자료 집합 전체를 가지고 검증하기 위해서는 순간 검증 방법보다 거래가 불필요하게 철회되어야 하는 경우가 존재한다. 다음의 예 2는 이를 보여준다.

예 2(원형 검증 방법에 의해 불필요하게 철회되는 거래의 예):

그림 2와 같은 거래들의 계획에 있어서 원형 검증 방법은 거래 T2와 거래 T3는 거래 T1′와 충돌한다. 왜냐하면 T1의 갱신 자료 집합은 T2와 T3의 판독 자료 집합과 중복이 되기 때문이다. 반면에 순간 검증 방법을 이용하면, T2는 T1의 검증 단계에서 시간 t1에 철회되고 T1의 기록 단계 후에는 시간 t2에서 철회된다. 왜냐하면 T1이 기록을 하기 전에 x를 읽었기 때문이다. 반면에, T3는 T1이 y를 이미 기록한 후에 y를 판독하기 때문에 정상적으로 완료된다. 그러므로 원형 검증 방법은 T2와 T3가 철회되어 실행이 재시작되어야 한다. 그러

나 순간 검증 방법 T2만이 최종적으로 시간 t2에 철회되기 때문에 T2만 재시작하면 된다. 따라서 원형 검증 방법은 순간 검증 방법보다 실행의 재시작으로 인한 부담을 더 크게 얻게 된다. 즉, 원형 검증 방법은 순간 검증 방법보다 거래가 불필요하게 철회되어야 하는 심각한 단점이 있다. 거래가 철회된다는 것은 곧 철회된 거래가 언젠가는 재시작되어야 하는 부가적인 부담을 안게 된다. 특히 다수의 사용자가 하나의 데이터베이스를 공유하는 시스템인 경우에 동시성을 높이게 되는데 이런 환경하에서 거래가 불필요하게 철회됨으로써 얻게 되는 시간적인 낭비는 무시할 수 없다. 특히 OCC는 충돌이 적게 일어나는 환경을 가정으로 하여 원형 검증 방법,순간 포착 검증 방법들이 존재하기 때문에 불필요한 철회로 인한 거래의 재시작 부담은 심각할 수 있다.



[그림 2] 직렬화된 실행 스케줄

다음으로 철회된 거래의 조속한 재시작 측면에서 비교해 보면, 철회된 거래들이 빠른 시간내에 재시작을 하기 위해서는 거래의 수행시 요구되는 모든 데이터 항목들이 주기억 장치에 미리 채취되어 있어야 한다. 그러나 순간 검증 방법은 그림 2의 예에서의 철회된 T2의 조속한 재시작을 위하여 데이터 항목들이 미리 주기억 장치에 미리 채취되지 않는다는 단점이 있다. 그러나 원형 검증 방법은 거래 수행에 필요한 모든 데이터 항목이 거래가 처음 실행될 때 미리 채취되므로 철회된 거래를 조속히 재시작할 수는 있으나 반대로 데이터 항목의 선채취로 인한 부가적인 노력을 해야만 한다.

2. 검증에 소요되는 시간

각 거래들이 거래의 철회 여부를 검증하는 검증 단계에서 소요되는 시간적인 측면을 고려해 보자면 검증 당시의 각 검증하고자 하는 거래의 판독 자료 집합과 갱신 자료 집합의 크기를 먼저 고려해야 한다. 즉 판독 자료 집합과 갱신 자료 집합의 크기가 큰 경우 원형 검증 기법은 순간 검증 방법에 비하여 보다 많은 시간이 소요된다. 왜냐하면 원형 검증 기법은 각 거래의 해당되는 거래들의 판독 자료 집합과 갱신 자료 집합과 갱신 자료 집합을 모두 다

비교하여 철회 여부를 결정하는 반면 순간 검증 방법은 거래가 완료될 당시의 순간 포착 장면에 해당되는 판독 자료 집합과 갱신 자료 집합만 철회 여부를 검증하는 작업에 참여하기 때문이다.

순간 검증 방법은 하드웨어 자원에 대한 처리 시간을 줄일 수는 있지만 반면에 버퍼의 사용(buffering)으로 인한 단점들이 나타난다. 즉, 순간 검증 방법은 거래가 처음 실행될 동안에 그 거래에서 필요로 하는 모든 데이터 항목들을 주기억 장치로 옮겨 올 수 없다. 그러므로 이것은 버퍼 보유 효과를 사용할 수 있는 동시성 제어 기법에는 매우 적합하지 않다.

반면에 동시성의 정도가 강한 사용자 공유 시스템에서 처리 기간이 긴 장기 거래를 실행해야 할 경우에는 순간 검증 방법은 원형 검증 방법에 비하여 검증에 소요되는 시간이 길어진다. 왜냐하면 장기 거래가 실행을 완료하기 전에 다른 여러 개의 단기 거래들이 미리 완료를 할 경우 순간 검증 방법인 경우 단기 거래들이 완료할 때마다 그 당시의 순간 포착 장면을 가지고 검증을 실시하기 때문에 실행 중인 장기 거래는 실행을 완료하는데 시간적인 부담을 안게 된다.

표 1은 앞에서 소개한 OCC기법들에 대한 비교를 요약한 것이다.

Ⅳ. 결론

낙관적인 동시성 제어 기법은 다수의 사용자가 하나의 자료 기지를 공동으로 사용하는 시스템에서 동시성을 높이기 위해 사용되는 거래 스케쥴링 기법 중 하나이다. 낙관적인 방법이란 거래간의 충돌이 적게 일어나고 거래의 철회와 복귀 그리고 재실행에 드는 비용이 적은 가정하에서 거래가 실행되는 과정에서는 동시성 제어를 위한 연산별 점검은 취하지 않고 실행이 끝난 후에, 주기억 장치 작업 공간에서 이루어진 실행 결과를 가지고 거래의 정확성 여부를 조사하는 방법이다. 이 기법은 충돌 빈도가 작고 거래의 길이가 짧아서 거래를 재실행하기 위한 비용이 적은 경우에 다른 동시성 제어 기법보다 우수한 성능을 갖는다.

낙관적인 검증 기법은 철회된 거래가 즉시 재시작하는 기법과 오직 완료 요청을 받았을 때, 즉 그 거래와 충돌된 거래가 이미 완료되었을 때에 재시작을 할 수 있는 기법으로 분류할 수가 있다.

원형 검증 방법과 순간 검증 방법은 후자에 속한다. 순간 검증 방법과 원형 검증 방법은 검증 단계에서 실패를 하게 되면 검증에 참여한 거래는 스스로 소멸되고 만다. 소멸된 거래가 재시작하는데 있어서 Ⅲ장에서 비교한 바에 의하면 원형 검증 방법은 버퍼 보유 효과가 제공된다면 순간 검증 방법보다 철회된 거래를 조속히 재시작시키는 면에서는 우수하였다. 그러나 이로 인한 하드웨어 자원의 낭비와 부가적인 노력이 필요하다.

원형 검증 방법은 순간 검증 방법에 비하여 거래가 불필요하게 철회될 가능성이 높기 때문에 거래가 처음 실행될 동안에 철회될 위기가 아주 높다. 그리고 철회된 거래가 재시작하기 위해 소요되는 시간적인 부담을 얻게 된다. 예 2에서도 살펴본 바에 의하면 원형 검증 방법은 충돌에 대하여 순간 검증 방법보다 융통성이 없기 때문에 거래가 처음 실행할 동안에

철회로 유도될 가능성이 매우 높다. 그러므로 앞으로는 모든 거래들의 수행을 진행한 후 가능한 한 많은 직렬화된 거래의 기록들을 인식할 수 있는 검증 방법을 고안할 필요가 있다.

[표 1.] OCC 기법에서 채택하는 검증 기법들의 비교

관점 \ 검증 방법		순간 검증 방법	원형 검증 방법
검증 방법 부류		"자신 포기" 검증 기법	"자신 포기" 검증 방법
거래의 충돌여부 검증방법		처리 검증 기록단계 별로처리함 거래의 완료시 다른 수행중인 거래들에 대한 순간검증 실시함	처리 검증 기록 단계별로 처리함 검증단계시 다른 거래들에 대한 충돌여부 검사 기록단계에서 철회여부를 결정함
버퍼보유효과		제공하지 않음	제 공
겨래의 재시작 부담측면	재시작 빈도수	작다	크다
	조속한 재시작	불가능함	가능함

참고문헌

1. [FRAN85] P. A. Franaszek, J. T. Robinson, "Limitations of Concurrency in Transaction Processing," ACM Trans. Database Syst,. Vol. 10, No. 1, 00. 1-28, Mar. 1985.
2. [FRAN90] P. A. Franaszek, J. T. Robinson, and A. Thomasian, "Access Invariance and Its Use in High Contention Environments," In Proc. 6th Int. conf. on Data Eng.,pp. 47-55, Feb. 1990.
3. [KIM88] Y. s. Kim, "AHybrid Concurrency control Schemes Using Snapshot Validation in Distributeed Database System ," M.S. Thesis, Dept. of Computer Science, KAIST, 1988.
4. [KUNG81] H. T. Kung and K. T. Robinson, "On Optimistic Methods for Concurrency Control," ACM Trans. Database Syst., Vol. 6, No.2, pp. 213-226, June 1981.
5. [ROBI82] J. T. Robinsin, "Experiments with Transaction Processing on Multiprocessor," IBM, Yurktown Heights, NY, Res. Rep. RC9725, Dec. 1982.
6. [SCHL86] G. Schlageter, U. Pradel and R. Unland, "Redesign of Optimistec Methods: Inproving Performance and Applicability," Proceedings 2nd International Conference on Data Engineering, May 1986, pp. 466-473.
7. [YU92] P. S. Yu and D. M. Dias, "Analysis of Hybrid Concurrency Control Schemes for a High Data Contention Environment." IEEE Trans. Software Eng., Vol. 18, NO 2, Feb. 1992.

이 혜 경 정회원

- 1979년 숭실대학교 전자계산학과 졸업
- 1985년 미국 일리노이대학교(Urbana-Champaign)전산학과 졸업(공학석사)
- 1988년 국립 천안공업전문대학 전자계산과 전임강사
- 1989년 영국 Edinburgh대학교 방문연구
- 1990년 ~ 1993년 동덕여자대학교 전자계산과 강사
- 1993년 ~ 1995 현 경인여자전문대학 사무자동화과 전임강사