

# 실시간 데이터베이스에서 시간일관성 만족을 위한 낙관적 병행수행 제어

박연희 이정환 김명준  
충북대학교 전자계산학과  
(foreverg cdevil mjkim)@rtlab.chungbuk.ac.kr

## Optimistic Concurrency Control for Satisfying Temporal Consistency in Realtime Database

Yeon-Hee Park Jung-Hwan Lee Myung-Jun Kim  
Dept. of Computer Science, Chungbuk National University

### 요 약

실시간 데이터베이스 시스템에서 낙관적 병행수행 제어는 충돌 해결 시에만 우선순위를 고려하기 때문에 교착상태가 발생하지 않는 시간적 측면에서의 장점으로 잠금기법보다 우수하다고 알려져 있다. 하지만 시간 일관성을 가지지 않는 데이터의 사용은 시스템에 부정적 영향을 준다. 따라서 본 논문에서는 기존의 동적 조정을 사용하는 낙관적 병행수행기법에 데이터의 시간 일관성을 고려한 낙관적 병행 수행 제어기법을 제안한다.

### 1. 서론

실시간 데이터 베이스 시스템은 트랜잭션이 시간제약을 가지고 있고 주된 성능평가의 척도가 평균반응시간이 아닌 종료시한을 만족시키는 트랜잭션의 비율이라는 점에서 전통적인 데이터 시스템과는 다르다. 실시간 데이터베이스에서의 시간제약이란 그 종료시한 뿐만이 아니라 트랜잭션이 접근하는 데이터의 시간 일관성도 포함이 된다.

실시간 데이터베이스에서 일관성은 논리적 일관성과 시간 일관성으로 나뉘어진다. 논리적 일관성은 데이터의 읽기 쓰기 작업 시에 공유되는 데이터에 관하여 데이터베이스에 일관된 상태로 유지하는 것을 말하며 시간 일관성은 트랜잭션이 로딩하는 데이터가 그 순간의 현실성을 반영해야 하는 것을 말한다.

본 논문은 다음과 같이 구성된다. 2절에서는 관련된 연구들을 살펴보고 3절에서는 데이터의 논리적, 시간 일관성에 대해 알아본다. 4절에서는 제안하는 동적 조정을 사용하는 낙관적 병행수행 제어 기법에 대해 알아보고 5장에서 결론을 맺는다.

### 2. 관련연구

기존의 실시간 데이터베이스 시스템에서 시간제약을 만족시키기 위한 병행수행 기법으로는 크게 두 가지로 나뉘어진다. 첫 번째 방법은 2단계 잠금기법으로 트랜잭션의 우선 순위에 기초하여 트랜잭션의 수행을 잠그거나 푸는 방법으로 시간제약을 해결한 방법이다. 일반적인 데이터베이스 시스템에서는 2단계 잠금기법이 가장 우수한 성능을 보이고 있다. 또 2단계 잠금기법의 교착 문제를 해결하기 위하여 예측 가능한 블러킹 시간을 갖는 우선순위 상한 프로토콜이 제안되었다.

두번째 방법은 낙관적 병행수행 제어 기법으로 트랜잭션의 충돌이 발생했을 때만 우선 순위를 고려하는 방법이다. 이 기법은 높은 비율의 철회와 순환적 재시작으로 기존 데이터베이스 시스템에서 잠금기법보다 성능이 떨어진다고 연구되어왔다. 하지만 충돌 해결 시에만 우선순위를 고려하기 때문에 교착상태가 발생하지 않는 시간적 측면에서의 장점으로 실시간 데이터베이스에서는 잠금기법보다 성능이 우수함이 연구결과를 통해서 보여주고 있다. 이는 실시간 데이터베이스의 성능평가가 종료시한을 만족하는 트랜잭션의 비율이라는 점이 중요하다 볼 수 있다.

이러한 방법들은 좀더 향상된 병행수행을 제공하고 종료시한을 만족하지 못하는 트랜잭션의 수를 줄이는 것을 목표로 한다.

하지만 이들 방법에는 데이터의 시간 일관성을 고려하지 않는다는 문제점이 있다. 데이터의 시간 일관성은 트랜잭션의 수행시에 현실값을 반영한다는데 큰 의미를 부여한다. 따라서 데이터가 시간일관성을 만족해야 올바른 트랜잭션의 수행과 결과값을 기대할 수가 있다.

시간 일관성을 고려하지 않는 기존의 연구들의 문제점을 해결하기 위하여 병행수행 제어시에 절대적, 상대적 시간일관성을 만족하는지를 검사하도록 하여 시스템의 성능을 향상시키는 방법을 살펴보도록 한다.

### 3. 데이터의 일관성

실시간 데이터베이스에서 일관성의 기준은 논리적 일관성과 시간 일관성으로 나뉘어진다. 실시간 데이터는 논리적 일관성을 만족함과 동시에 시간 일관성을 만족해야만 데이터의 일관성을 만족했다고 본다.

논리적 일관성은 여러 트랜잭션들이 수행하여 마치 하나의 트랜잭션의 끝나고 다른 트랜잭션이 수행되는 것처럼 동시에 수행되지 않으면 직렬화 되었다고 하고 이를 논리적 일관성을 만족했다고 한다.

실시간 시스템에서의 시간 일관성은 절대적 시간 일관성과 상대적 시간 일관성으로 나뉜다.

절대적 시간 일관성이란 트랜잭션이 데이터(X)를 로딩했을 때 그 데이터가 저장된 시간(t(X))으로부터 절대적 시간허용구간(avi<sub>x</sub>)안에 존재하여야 한다는 것이다.

$$t(X) \leq \text{current\_time} \leq t(X) + \text{avi}_x$$

상대적 시간 일관성이란 한 트랜잭션이 읽어오는 데이터의 집합을 R={X<sub>1</sub>, ..., X<sub>j</sub>}라 할 때 임의의 데이터 사이에 시간 관계가 상대적 시간 허용구간(rvi(R))안에 있어야 한다.

$$|\text{avi}(X_i) - \text{avi}(X_j)| \leq \text{rvi}(R) \quad , \quad \forall X_i, X_j \in R$$

### 4. 제안하는 동적 조정을 사용하는 낙관적 병행수행 제어 기법

제안하는 기법에서는 트랜잭션에 대한 종료시한은 준경성 종료시한으로 가정한다.

전방향 검증(forward validation)은 검증단계에서 트랜잭션의 직렬 순서가 트랜잭션의 도착순서에 의해 결정된다는 것을 가정하고 있다. 따라서 검증단계의 트랜잭션은 재시작이 되지 않는 한 항상 실행되고 있는 트랜잭션보다 선행된다. 우리는 이 가정이 불필요할 뿐만 아니라 이 가정에 기초한 검증수행은 불필요한 재시작을 초래한다. 이런 재시작은 피해야 한다. 따라서 Lee와 Son은 낙관적 병행수행 제어기법의 문제점으로 지적되어 온 철회와 재시작을 최소화하기 위하여 직렬화순서의 동적 조정 기법을 사용하고 있다.

각각의 트랜잭션 T에 대하여 수행동안 임시적으로 다

음과 같은 자료구조를 가진다.

TS : 타임스탬프  
 TI : 타임스탬프 인터벌  
 Ta : 액티브 트랜잭션  
 Di : 데이터  
 RS : T의 판독집합  
 WS : T의 기록집합  
 RTS : 판독 타임스탬프  
 WTS : 기록 타임스탬프

#### 4.1 검증 단계

판독단계에서의 모든 트랜잭션은 타임스탬프 인터벌을 할당받는데 이 인터벌은 트랜잭션의 실행동안 참조된 임시 직렬화 순서를 기록한다. 실행 시작에서 타임스탬프 인터벌은 무한대 값을 가진다. 그리고 데이터 연산이나 다른 트랜잭션의 검증에 의해서 트랜잭션의 직렬화 순서가 조정될때마다 타임스탬프 인터벌이 그 의존성을 표현하기 위해서 적용된다.

[검증단계 알고리즘]

```
select TS(Tv) from TI(Tv);
foreach Ta adjust(Ta);
foreach Dj in RS(Tv) update RTS(Dj);
foreach Dj in WS(Tv) update WTS(Dj);
commit WS(Tv) to database;
end
```

액티브 트랜잭션의 타임스탬프 인터벌의 적용은 충돌 타입과 그에 대응하는 순서에 따라 직렬순서를 기록하는 과정이다. 이는 다음과 같이 기술할 수 있다.

[adjust() 알고리즘]

```
foreach Dj in RS(Tv)
    if Dj in WS(Ta) then Ti(Ta):=TI(Ta) ∩ {Ts(Tv),);
    if TI(Ta)=[ ] then restart(Ta);
foreach Dj in WS(Tv){
    if Dj in RS(Ta) then
        TI(Ta):=TI(Ta) ∩ [0, TS(Tv)-1];
    if Dj in WS(Ta) then
        TI(Ta):=TI(Ta) ∩ [TS(Tv),);
    if TI(Ta)=[ ] then restart(Ta);
end
```

#### 4.2 판독단계

트랜잭션 검증에서 액티브 트랜잭션의 타임스탬프 인터벌의 조정은 commit 트랜잭션과 동시에 실행되는 트랜잭션에서 수행되는 데이터 연산사이에 직렬 순서를 기록하는 과정이다. 액티브 트랜잭션은 남은 데이터 연산을 실행해야하기 때문에 남은 연산에 의해 야기되는 실행 순서는 다른 비직렬 수행을 야기하는지 결정하기 위해 체크 되어야 한다. 만약 그렇다면

액티브 트랜잭션은 재시작 되어야 한다.

[관독단계알고리즘]

```
foreach Di in RS(Ta) read(Di);
TI(Ta):=TI(Ta)∩ [WTS(Di),);
if TI(Ta)=[ ] then restart(Ta);
foreach Di in WS(Ta) pre-write(Di);
TI(Ta):=TI(Ta)∩ [WTS(Di),)∩ [RTS(Di),)
if TI(Ta)=[ ] then restart(Ta);
end
```

#### 4.3 시간적 일관성 검증

시간 일관성 만족을 위하여 로딩한 시간 데이터의 가장 작은 타임스탬프와 가장 큰 타임스탬프에 대한 기록을 유지한다. 이 두 값의 차이의 절대값이 상대적 시간 일관성보다 작으면 상대적 시간 일관성을 위반한 것이므로 즉시 트랜잭션을 취소하고 재시작한다.

[시간일관성 검증 알고리즘]

```
foreach Di in RS(Ta)
if tnow > t(Di) + avi(Di) then restart(Ta)
foreach Di, Dj in RS(Ta)
if |lavi(Di) - avi(Dj)| > rvi(RS(Ta))
then restart(Ta)
end
```

#### 4.4 기록단계

기록단계에서는 검증 단계를 거친 트랜잭션이 수행한 결과집합을 데이터베이스에 영구 반영한다. 만약 트랜잭션이 종료시한을 넘긴 것이 확인된다면 시스템에서 바로 제거시켜버린다.

### 5. 결론

실시간 데이터베이스 시스템에서 기존의 병행수행 제어 기법은 데이터의 논리적 일관성을 만족하면서 시간 제한을 만족시키는 트랜잭션의 비율을 높이도록 많은 연구를 해 왔다. 하지만 논리적 일관성 뿐만 아니라 시간 일관성도 실시간 데이터베이스에서는 시스템의 정확성의 면에서 중요한 요소이다. 시간 일관성을 만족하지 못하는 데이터는 트랜잭션의 잘못된 결과값을 야기하게 되고 이는 시스템에 큰 오류로 작용할 수 있다. 본 논문에서는 동적 조정을 사용하는 낙관적 병행수행기법에 접목시켜 실시간 데이터베이스 시스템에서 요구되는 논리적 일관성과 시간적 일관성을 만족하기 위한 알고리즘을 보였다.

본 논문에서 사용된 알고리즘은 지금 수행평가 단계이며 그 수행결과는 시스템의 정확성과 기능면에서 많은 향상을 보일 것으로 예상된다.

추후로 연구가 되어야 할 부분을 살펴보면 본 논문에서 사용하는 데이터의 시간 일관성은 고려가 되었지만 데이터 충돌 해결에서 트랜잭션의 종료시한 정보는 사용하지 않았다. 이 정보를 기본 짜임에 넣는다면 더 좋은 병행수행 제어 알고리즘이 될것이라 생각한다.

### 6. 참고 문헌

- [1] Abbot R. and H. Garcia - Molina, "Scheduling RealTime Transaction: A Performance Evaluation," ACM Transactin on Database Systems, Vol. 17, No.3, pp.513-560, September 1992.
- [2] J.Huang, J.A.Stankovic, K.Ramamritham and D. Towsely, "Experimental Evaluation of Real-Time Optimistic Concurrency Control Systems," Proceedings of the 17th Internaitonal Conference on VLDB, pp. 35-46, Sep. 1991
- [3] J. Lee and S. H. Son, "Using Dynamic Adjustment of Serialization Order for Real-Time Database Systems," 14th IEEE Real-Time Systems Symposium pp.66-75, Raleigh-Durham, North-Carolina, December 1993.
- [4] J.R.Haritsa, M.Livny and M.J.Carey, "Dynamic Real-Time Optimistic Concurrency Control," Proceedings 11th IEEE Real-Time Systems Symposium, pp.94-103, Dec. 1990.
- [5] Lee.J and S.H.Son, "Optimistic Concurrency Control Protocol for Real-Time Database Systems," 3rd International Symposium on Database Systems for Advanced Applications, Daejon, Korea, April 1993
- [5] M.Xiong, J.Stankovic, K.Ramamritham, D.Towsley, and R.sivasankaran, "Maintaining Temporal Consistency : Issues and Algorithms," Proc. of the First International Workshop on Real-Time Databases:Issues and Applications, 1996.
- [6] M. Xiong, R. Sivasan karan, K. Ramamritham, J.A. Stankovic and D. Towsely, "Scheduling Transactions with Temporal constraints : Exploiting Data Semantics," 17th IEEE Real-Time system Symposium, Washington, D.C., 1996.
- [7] Y.Lin and S.H.Son, "Concurrency Control in Real-Time Database by Dynamic Adjustment of Serialization Order," 11th IEEE Real-Time Systems Symposium, pp. 104-112, Dec.1990