XML 변경 유효성 검증을 위한 경계락킹에 기초한 시퀀스 그룹 검증기법

최윤상[0] 박 석
서강대학교 컴퓨터학과
(heiyo@sogang.ac.kr, spark@dblab.sogang.ac.kr)

Sequence Group Validation based on Boundary Locking for validation of updating XML

Yunsang Choi[0] and Seog Park
Department of Computer Science, Sogang University

요 약

DTD에 의해서 문서의 형식이 정의된 valid XML을 XML 데이터베이스 시스템을 사용하여 관리하는 경우 XML의 변경은 그 변경 결과가 DTD에 대한 유효성(validity)을 만족시킬 때에만 수행되어야 한다. 이것은 다수의 사용자에 의해서 데이터가 공유되는 데이터베이스 시스템의 데이터 무결성과 관련되는 문제이기 때문에 XML 문서 변경에 대한 DTD 유효성은 XML 데이터베이스 시스템에서 중요한 속성이라고 할 수 있다.

변경 연산의 결과에 대한 XML의 유효성을 보장하기 위해서 변경의 유효성을 검증하는 방법을 사용할 수 있다. XML에서의 엘리먼트들은 순서 관계를 가질 수 있으며 DTD는 이러한 엘리먼트 순서 관계들을 정의하고 있기 때문에 이러한 유효성 검증 기법은 변경되는 데이터 아이템 외에도 주변의 데이터 아이템-엘리먼트-들에 대한 순서 정보를 필요로 한다. 그리고 데이터베이스와 같은 다중 사용자 환경에서 유효성 검증 기법이 정확하게 수행되기 위해서는 유효성 검증에 필요한 정보들이 다른 트랜잭션에 대해 변경되지 않도록 하는 병행수행 제어 기법을 필요로 한다. 이렇게 유효성 검증 기법과 병행수행 제어 기법이 관련을 가지고 있음에도 불구하고 기존의 유효성 검증 기법은 오직 검증의 효율성에만 초점을 맞추고 있다.

본 연구는 유효성 검증의 검증 범위를 최소화 시켜 pan-out 값이 큰 XML 문서에 대해서도 유효성 검증이 효율적으로 수행될 수 있고, 또한 유효성 검증을 위해 락킹되는 데이터 아이템의 수를 최소화 할 수 있는 시퀀스 그룹 검증 기법을 제안한다. 또한 이 검증 기법의 정확을 보장하면서도 높은 트랜잭션 병행수행 성능을 보장할 수 있는 경계 락킹 기법을 제안한다. 제안된 유효성 검증 기법과 경계 락킹 기법은 유효성 검증의 정확성을 위해 병행수행 성능이 저하될 수밖에 없는 기존의 기법들의 문제점들을 해결하여 XML 데이터베이스 시스템이 안정적인 성능을 제공할 수 있다는 것을 실험을 통해 확인할 수 있었다.

1. 서 론

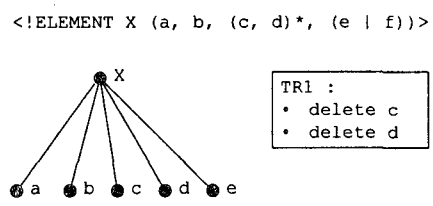
현재 XML은 웹 문서를 위한 표준 언어로써 뿐만 아니라 이기종 시스템 간의 데이터 교환을 위한 데이터 표현 언어로써 널리 사용되고 있다. XML 형식의 데이터가 증가함에 따라 다량의 XML 문서들을 공유하고 관리하기 위한 데이터베이스 시스템의 개발이 요구되었다. 하지만 XML이 가지고 있는 구조적인 특징들은 트랜잭션들이 병행 수행되는 데이터베이스 환경에서 복잡성을 야기할 수 있다.

XML에서 각 데이터 아이템들은 수평적, 수직적 관계로 서로 연결되어 있다. 부모-자식 관계(수직적 관계)는 XML의 데이터 아이템들이 계층적 구조를 가지고 있음을 의미하며, 이러한 계층적 구조 하에서의 경로 표현을 통해 데이터 접근이 이루어지기 때문에 동시 수행되는 트랜잭션들 간의 간섭의 문제는 다소 복잡해질 수 있다. 이러한 문제들은 트랜잭션 병행수행 제어에 대한 기존 연구들에서 지적된 바 있다.

XML 문서에서 형제 관계에 있는 엘리먼트들 사이에는 순서 관계(수평적 관계)가 존재한다. 엘리먼트 간의 순서관계는 유효순서가 존재하는 valid XML 문서를 사용하는 환경에서 특히 중요한데, 유효순서에 의한 수평적 관계의 문제는 유효성 검증 기법에 대한 연구들에서 다루어진 바 있다.

하지만 유효순서의 문제는 단순히 유효성 검증에만 영향을 미치는 것은 아니다. valid XML 문서가 데이터베이스에 저장된 경우 엘리먼트들 간의 유효순서는 트랜잭션의 병행수행 제어에도 영향을 미치게 된다. 어떤 트랜잭션에 의해 변경된 엘리먼트 순서의 유효성을 검증하는 과정은 필연적으로 변경 연산의 대상이 아니었던 다른 엘리먼트들에 대한 비명시적인 접근을 필요로 한다. 여러 트랜잭션들이 병행 수행될 때 유효성 검증의 정확성을 위해서는 이렇게 비명시적으로 접근되는 데이터가 다른 트랜잭션에 의해 변경되지 않도록 해야 한다. 이것은 트랜잭션의 병행수행 제어에서 엘리먼트의 유효순서 관계가 고려되어야 함을 의미한다.

또한 엘리먼트들의 유효순서는 유효한 변경을 수행하는 트랜잭션을 구성하는 변경 연산에 대한 예측을 가능하게 한다. 예를 들면 [그림 1]과 같은 예에서 TR1의 첫 번째 연산은 엘리먼트 c를 삭제하는데 TR1이 유효한 변경을 수행하는 트랜잭션이라면 TR1은 첫 번째 연산을 수행한 후에 새로운 c를 삽입하거나 혹은 d를 삭제하는 연산을 수행하여야 함을 예측할 수 있다. 이러한 예측을 고려하면 c가 삭제되는 순간에 미리 엘리먼트 d에 대한 락을 획득함으로써 다른 트랜잭션에 의한 간섭을 미리 배제할 수 있다. 그런 의미에서 d는 TR1 트랜잭션이 c를 삭제한 시점에서의 잠재적 접근 가능 데이터라고 할 수 있다.



[그림 1] valid XML 변경 예

엘리먼트들의 유효순서는 유효성 검증을 위한 비명시적 데이터 접근과 관련하여 유효성 검증의 효율성에 직접적인 영향을 미치기도 하지만, 트랜잭션의 병행수행 제어에도 영향을 미친다는 것을 설명하였다. 본 연구는 유효순서 관계에 있는 최소화된 엘리먼트 집합으로써 시퀀스 그룹을 소개하고, 시퀀스 그룹 단위로 유효성 검증을 수행하는 시퀀스 그룹 검증기법을 제안하고 시퀀스 그룹 검증기법의 정확성을 보장하며 시퀀스 그룹 단위의 변경을 수행하게 함으로써 보다 안정적인 트랜잭션 병행수행 성능을 제공할 수 있는 경계 락킹 기법을 제안한다.

2. 관련 연구

현재 대부분의 XML 데이터베이스 시스템들은 트랜잭션이 수

행하는 변경에 대한 유효성을 검증하기 위해서 문서 전체를 다시 파싱함으로써 트랜잭션의 유효성을 검증하는 방법을 사용하고 있다. 이러한 문서전체에 대한 검증 방법은 문서의 모든 데이터 아이템에 대해 비명시적으로 접근함으로써 변경 트랜잭션의 병행 수행을 불가능하게 하는 문제점을 가지고 있다. 또한 문서의 크기가 큰 경우 유효성 검증의 오버헤드가 커진다는 문제를 안고 있다.

보다 효율적인 트랜잭션의 유효성 검증을 위해서 [1]에서는 즉시부분 검증기법을 제안하였다. 여기에서 즉시검증의 아이디어는 트랜잭션 당 연산의 수를 하나라고 가정하였기 때문에 실제로 응용하기에는 문제를 가지고 있으므로, 부분검증의 아이디어만을 고려한다. 부분 검증기법은 유효순서 관계에 있는 엘리먼트 집합을 형제관계에 있는 모든 엘리먼트들로 정의하였다. 따라서 비명시적으로 접근되는 데이터의 범위는 형제 엘리먼트들로 제한되기 때문에, 문서전체에 대하여 검증하는 방법보다 더 나은 병행수행 성능 및 검증 효율성을 보장할 수 있다.

하지만 부분 검증기법은 pan-out 값이 큰 XML 문서에 대해서는 효율성 문제를 가지고 있다. valid XML 문서의 형식을 정의하는 DTD가 순환적인 엘리먼트 정의를 포함하고 있지 않은 경우, 그 문서는 depth는 고정되고, 카디널리티에 의한 수평적인 팽창에 의해서만 문서의 크기가 증가하게 된다. 이러한 문서의 경우, 문서의 크기가 커질 때 비명시적 데이터 접근의 양이 많아지게 되어 유효성 검증의 오버헤드는 커지게 된다. 또한 잠재적 접근가능 데이터의 양 역시 커지기 때문에 병행수행 성능 역시 저하될 수 있다는 문제를 가지고 있다.

XML 데이터베이스에서 트랜잭션 병행수행 제어기법에 대한 연구들로는 [2],[3],[4]와 같은 연구들이 수행되었다. [2]의 연구는 XML 문서의 데이터 아이템들 간의 수평적 관계를 고려하여 well-formed XML의 관점에서 노드 삽입/삭제의 순서 충돌을 해결하는 락킹 프로토콜을 제안하고 있다. 하지만 [2]의 연구는 valid XML에서의 엘리먼트들 간의 유효순서에 의해 발생할 수 있는 트랜잭션의 유효성 검증과 병행수행의 문제는 고려하고 있지 않다. [3],[4]의 연구는 DGLOCK과 Path locking 기법이라는 락킹 기법을 제안하고 있다. 하지만 이 기법들은 모두 XML 데이터의 수직적 관계, 즉 계층적 구조 하에서 발생하는 트랜잭션 충돌 문제에 초점을 맞추고 있다.

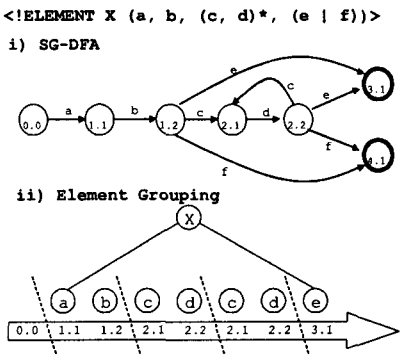
3. 시퀀스 그룹 검증

유효순서 관계에 있는 엘리먼트 집합에 대한 정의가 유효성 검증의 효율성과 병행수행 제어에서의 락킹 단위와 관련이 있다는 것을 앞에서 설명하였다. 부분 검증 기법에서의 유효순서 관계에 있는 엘리먼트 집합은 형제관계에 있는 엘리먼트들이다. 따라서 유효성 검증의 단위 및 변경에 대한 락킹의 단위 역시 형제관계 엘리먼트들이 된다. 이것은 유효성 검증의 정확성을 위해 변경되는 엘리먼트의 모든 형제 엘리먼트들에 대해 공유모드 락이 필요하는 것을 의미하고, 또한 형제 엘리먼트들이 잠재적으로 변경될 가능성을 가지고 있기 때문에 형제 엘리먼트들에 대해 배타모드 락을 얻을 필요가 있다는 것을 의미한다.

이러한 모든 형제 엘리먼트들에 대한 배타모드 락킹은 계층적 구조를 갖는 XML의 특성상 그것들의 하위 노드에 대한 다른 트랜잭션의 접근을 허용하지 않기 때문에 시스템의 병행수행 성능을 크게 저하시킬 수 있다는 문제를 가지고 있다. 따라서 유효순서 관계에 있는 엘리먼트 집합을 더 작은 범위로 줄임으로써 이러한 문제를 해결할 필요가 있다.

(정의1) 시퀀스 그룹 *DTD의 엘리먼트 선언에서 오직 순차내용모델(Sequence Content Model)로만 선언된 엘리먼트들의 집합.*

[그림 2]는 형제 관계에 있는 엘리먼트들을 시퀀스 그룹으로 그룹화하는 예를 보이고 있다. 이러한 엘리먼트 그룹화는 시퀀스 그룹으로 그룹화된 상태 라벨을 갖는 SG-DFA(State Group - Deterministic Finite Automata)를 사용함으로써 가능하다.



[그림 2] 유효순서 관계에 의한 엘리먼트 그룹화

SG-DFA는 일반적인 DFA(Deterministic Finite Automata)와 유사하지만 시퀀스 그룹에 의한 Group ID와 그룹 내에서의 순서 정보를 갖는 Sequence ID에 의해 상태 라벨이 정의된 상태들로 정의된다는 차이점을 가지고 있다.

DTD의 엘리먼트 선언은 결정적(deterministic)으로 선언되어야 하기 때문에 모든 엘리먼트 선언은 그와 동등한 DFA를 갖는다. 그리고 엘리먼트 선언에서 순차내용모델(,)과 선택내용모델(|), 카디널리티(*,+,?)는 DFA에서 순차전이, 분기, 루프의 형태로 나타나기 때문에 순차전이를 하는 상태들을 그룹화함으로써 DFA를 쉽게 SG-DFA로 변환할 수 있다.

시퀀스 그룹은 엘리먼트들 사이의 최소한의 유효순서 정보를 가지는 단위가 될 수 있다. 그것은 선택내용모델과 카디널리티는 엘리먼트들이 나타날 수 있는 조건과 횟수를 제한하는 내용모델이라는 점과는 달리 순차내용모델은 순수하게 순서정보를 제공하는 내용모델이기 때문이다.

시퀀스 그룹을 유효순서 관계에 있는 엘리먼트 집합으로 정의함으로써 유효성 검증에 의해 발생하는 비명시적 데이터 접근의 양을 기존의 방법에 비해 줄일 수 있게 된다. 즉, 모든 형제 엘리먼트들을 읽는 것이 아니라 같은 시퀀스 그룹의 엘리먼트들만을 읽으므로써 엘리먼트들의 유효순서에 대한 검증이 가능하다.

■ 시퀀스 그룹 검증기법

어떤 엘리먼트 Ep의 자식인 엘리먼트 시퀀스 그룹 SGx가 두 엘리먼트 시퀀스 그룹 SGy, SGz와 SGy-SGx-SGz의 순서로 인접해 있고 트랜잭션 Tx에 의해 SGx'로 변경되었다고 할 때 다음과 같이 Tx의 SGx'에 대한 변경 유효성을 검증하는 기법을 시퀀스 그룹 검증 기법이라 한다. (Ep에 대한 DFA는 M이라 하자)

i) SGx가 Tx에 의해 변경되기 전에 SGy의 마지막 엘리먼트 ey에 의한 M의 전이 상태 Qy와 SGz의 첫 엘리먼트 ez에 의한 M의 전이 상태 Qz를 저장한다.

ii) Tx가 모든 연산을 다 수행한 후 승인(commit)될 때 SGx'의 모든 엘리먼트와 ez에 의해 M이 상태 Qy에서 Qz로 전이될 수 있는 검사한다. 전이할 수 있다면 Tx가 수행한 변경 SGx↦ SGx'은 유효한 변경이다.

시퀀스 그룹 검증기법은 SG-DFA의 상태 전이가 결정적(deterministic)이라는 것에 기초하여 시퀀스 그룹 단위의 유효순서 검증을 수행하는 기법이다. 그리고 이 기법이 변경된 엘리먼트 순서의 유효성을 검증할 수 있음은 (정리 1)을 통해 쉽게 증명될 수 있다.

(정리 1) *스트링 $I=(e_1e_2...e_x\ \sigma\ e_ye_{y+1}...e_n)$가 어떤 DFA M에 의해 승인될 때 상황 $[Q_x, \sigma\ e_y...e_n]$과 $[Q_y, e_{y+1}...e_n]$이 발생한 후 승인상태에 도달한다고 하자. 그리고 스트링 $I'=(e_1e_2...e_x\ \sigma'\ e_y...e_n)$에 대해 I'의 서브스트링 $(\sigma'\ e_y)$가 M에 대해 상태 Q_x를 Q_y로 전이시킬 수 있다면 스트링 I'도 M에 의해 승인되는 스트*

링이다.

(증명) *DFA에서는 모든 전이함수는 오직 하나의 상태만을 결과로 갖기 때문에 M은 입력 스트링 l'에 대해서도 상황 $[Q_x, \sigma' e_y...e_n]$이 발생할 것이 분명하다. 마찬가지로 l'에 의해 M의 상황 $[Q_y, e_{y+1}...e_n]$이 발생한다면 l'는 M에 의한 승인상태에 도달할 수 있다. 따라서 l'가 M을 $[Q_x, \sigma' e_y...e_n]$에서 $[Q_y, e_{y+1}e_n]$으로 이동시킬 수 있다면 l'는 M에 의해 승인될 수 있다.*□

그리고 여러 트랜잭션들이 병행 수행되는 환경에서 시퀀스 그룹 검증이 정확하게(correct) 수행되기 위해서는 변경되는 시퀀스 그룹의 엘리먼트들에 대한 다른 트랜잭션의 변경 및 읽기 연산이 수행되지 않아야 하며, 그 시퀀스 그룹과 인접한, 전후의 두 엘리먼트들이 다른 트랜잭션에 의해 변경되지 않아야 한다. 이러한 조건들은 병행수행 제어 메커니즘에 의해 보장될 수 있어야 하고, 본 연구에서는 그것을 위한 경계 락킹 기법을 제안한다.

4. 경계 락킹

서론에서 설명한 것처럼 XML 변경의 단위로써의 시퀀스 그룹은 잠재적 데이터 접근 가능 범위라 할 수 있는데, 시퀀스 그룹 단위의 데이터 변경이 가능하기 위해서는 적절한 병행수행 제어기법이 필요하다. 또한 앞 절에서 설명한 시퀀스 그룹 검증 기법은 시퀀스 그룹 단위의 유효 순서 검증을 수행하기 위해서 병행수행에 대한 제약조건들을 가지고 있다.

■ 경계 락킹 기법

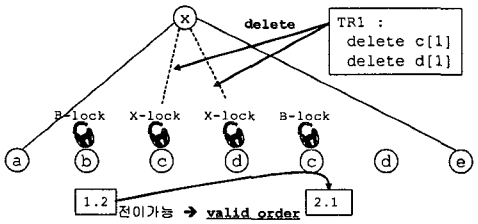
경계 락은 변경 연산 insert와 delete에 의해 생성되며 연산이 수행되기 전에 변경될 엘리먼트에 대해 다음과 같은 절차를 통해 경계 락과 배타모드 락을 설정한다. 다음의 절차는 하나의 락을 설정하는 것처럼 원자적(atomic)으로 수행되도록 보장되어야 한다.

i) SG-DFA를 이용하여 변경되는 엘리먼트 시퀀스 그룹의 경계를 알아낸다.

ii) 변경되는 엘리먼트 시퀀스 그룹에 인접하는 두 엘리먼트에 시작 경계 락과 끝 경계 락을 설정한다. 시작/끝 경계 락은 락킹의 대상 엘리먼트에 의한 DFA의 전이 상태 정보를 가진다.

iii) 두 경계 락 사이에 있는 모든 엘리먼트들-즉 변경되는 엘리먼트 시퀀스 그룹의 엘리먼트들-에 배타모드 락을 건다.

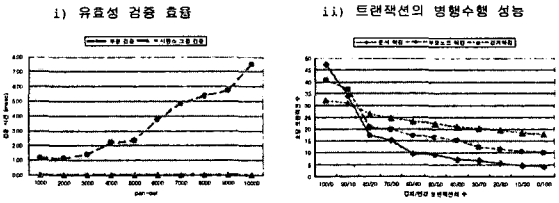
경계 락킹 기법은 시퀀스 그룹 단위의 변경 뿐만 아니라 시퀀스 그룹 단위의 유효 순서 검증을 가능하게 하는 중요한 정보를 락킹 정보에 포함시켜 관리하는 기법이다. 기본적으로 경계 락킹은 변경되는 시퀀스 그룹에 대해서는 배타모드 락킹을 사용하고, 그와 인접한 두 엘리먼트들에 대해서는 SG-DFA의 상태 정보를 가지고 있으며 일종의 공유모드 락(shared lock)인 **경계 락(boundary lock)**을 사용한다. 경계 락은 시작 경계락과 끝 경계락의 두 타입으로 나뉘며, 저장된 SG-DFA 상태 정보를 이용하여 시퀀스 그룹 검증을 가능하도록 한다.



[그림 3] 경계 락킹과 시퀀스 그룹 검증

[그림3]은 경계 락킹과 시퀀스 그룹 검증이 어떻게 상호작용하는지를 보이고 있다. 경계 락킹에 의해 첫 번째 c, d 노드가 배타모드로 락킹되기 때문에 TR1의 두 연산들은 다른 트랜잭션의 간섭없이 수행될 수 있다. 또한 부분검증 기법을 위한 병행수행 제어에서는 모든 형제노드들이 락킹됨으로써 그 모든 하위 노드에 대한 다른 트랜잭션의 접근이 불가능했던 것과는 달리 c, d에 대해서만 배타모드 락킹을 하기 때문에 다른 트랜잭션들의 병행수행이 가능하게 되었다. 그리고 변경되는 시퀀스 그룹의 두 인접 엘리먼트에 대한 경계 락이 DFA의 전이 상태 정보를 가지고 있기 때문에 유효 순서 검증이 효율적으로 수행될 수 있다.

5. 성능 평가



[그림 4] 성능 실험 결과

[그림 4]의 i)는 문서의 pan-out 값 증가에 따른 유효성 검증의 소요시간을 측정한 결과이다. 실험에서 부분검증 기법이 pan-out값 증가에 따라 검증 오버헤드가 커지는 것과는 달리 시퀀스 그룹 검증은 pan-out 값 증가에 따른 영향을 거의 받지 않음을 알 수 있다. ii)는 변경 연산을 수행하는 트랜잭션 비율의 증가에 따른 throughput를 측정한 결과이다. 변경 트랜잭션의 비율이 작은 경우에는 락킹 오버헤드가 큰 경계락킹이 다소 낮은 성능을 보이지만, 변경 트랜잭션의 비율이 20%가 넘는 순간부터 경계 락킹이 다른 유효성 검증 기법을 위한 락킹 기법들보다 나은 성능을 가지게 됨을 볼 수 있다. 즉, 경계 락킹기법은 변경 트랜잭션 비율의 증가에 따른 성능 하락 폭이 가장 작아 보다 안정적인 성능을 제공한다고 할 수 있다.

6. 결론

시퀀스 그룹 검증 기법과 경계 락킹은 valid XML 문서에 대한 변경 단위로써 노드가 아닌 시퀀스 그룹이라는 추상화된 데이터 접근 단위를 사용함으로써 유효성 검증에 의한 비명시적 데이터 접근과 valid XML에 대한 변경 특성에 의한 잠재적 접근 가능 데이터의 범위를 줄이므로써, 유효성 검증의 효율성과 트랜잭션 병행수행 성능을 향상시킬 수 있었다.

다른 XML 트랜잭션 병행수행 기법들과는 달리 경계 락킹은 데이터들의 수평적 관계를 고려한 락킹 기법이기 때문에 XML의 계층적 구조하에서 발생하는 다른 충돌 문제들에 대해서는 고려하지 않았다. 이것에 대한 추후 연구가 필요하다.

참고 문헌

[1] Sang-Kyun Kim, Myungcheol Lee, Kyu-Chul Lee, "Immediate and Partial Validation Mechanism for the Conflict Resolution of Update Operations in XML Databases", WAIM 2002, Lecture Notes in Computer Science, Vol.2419, pp.387-396, Aug 2002

[2] Sven Helmer, Carl-Christian Kanne, Guido Moerkotte, "Evaluating Lock-based Protocols for Cooperation on XML Documents", SIGMOD Record, Vol33, No.1, March 2004

[3] Torsten Grabs, Klemmens Böhm, Hans-Jörg Schek, "XMLTM: Efficient Transaction Management for XML Documents", CIKM'02, McLean, Virginia, USA, November 2002

[4] Stijn Dekeyser, Jan Hidders, Jan Paredaens, "A Transaction Model for XML Databases", World Wide Web Journal, Kluwer, 2003