Transformation of XQL Queries Using XML Materialized Views

SooHee Kim 0+  ChanHo Moon 0+  YoungSung Kim+  Hyunchul Kang+  Sang-Koo Seo++
+Dept. of Computer Science and Engineering, Chung-Ang University
++Dept. of MIS, Kwangwoon University

요 약


1. 서론


본 논문에서는 XML 저장소에 저장된 여러 XML 문서들을 대상으로 하는 XQL 질의의 성능 향상을 위해 XQL 질의의 결과로 얻어진 XML 문서들을 저장하고 있는 XML 실제 블록(materialized view)의 효과를 연구하고, 이를 이용한 XQL 질의 변환에 대한 결과를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하며, 3장에서는 본 논문에서 제시하는 XML 저장 구조에 대해 설명한다. 4장에서는 XQL 질의 결과와 XML 실제 블간의 관계를 분석하고, 이에 따른 XQL 질의 변환 과정에 대해 설명한다. 5장에서는 결론을 덧붙인다.

* 본 논문은 한국 과학 재단의 특정 기초 연구 지원 사업에 의한 것이다.

2. 관련 연구


3. XML 문서의 저장 구조

본 절에서는 먼저 본 논문의 기본적인 가정을 기술한 후, XML 문서의 저장 구조에 대해 설명한다.

3.1 가정

본 논문에서는 XML 실제 블과 사용자로부터 주어지는 XQL 질의에 대해 다음과 같은 가정을 한다.

- XQL 질의는 XML 저장소에 저장된 여러 XML 문서들을 대상으로하며, XML 저장소에 저장된 모든 XML 문서는 공동된 하나의 DTD를 만족한다.
3.2 XML 저장소의 구성

본 논문에서 가정하는 XML 문서의 저장 구조는 크게 XML 저장소의 하부 XML 문서 영역과 XML 실제 루 영역으로 나누어 볼 수 있다. 각 영역의 XML 문서는 엘리먼트 단위로 분할되어 엘리먼트 레이블과 루 엘리먼트 레이블에 각각 저장된다. (그림 2)는 (그림 1)의 XML 문서를 저장한 예를 보여준다. 이들을 각 설명하면 다음과 같다.

DTD 태이블: (그림 2)의 (a)와 같이 XML 저장소의 모든 XML 문서가 만족하는 DTD의 구조를 트리 형태로 나타내었을 때, 각 엘리먼트에 대해 두드 엘리먼트로부터 위치를 표시하는 DTD(ID/DTD ID)와 해당 엘리먼트의 이름으로 구성된다.

XML 루 엘리먼트: XML 저장소의 모든 XML 문서를 (그림 2)의 (b)와 같이 XML 문서의 루 엘리먼트로 분할하여 저장한다. 엘리먼트 각각은 XML 문서를 구분하는 DID(Document ID)와 XML 문서의 트리 구조에서의 위치를 표시하는 EID(Element ID)[6]. 엘리먼트의 DTD 를 나타내는 DTDID와 엘리먼트 이름, 내용으로 구성된다.

DTD 메핑 태이블: (그림 2)의 (c)와 같이 XML 실제 루 엘리먼트와 XML 실제 루 엘리먼트에 대해 XML 실제 루 엘리먼트의 메핑정보를 저장한다. 실제 루 엘리먼트에 대한 XML 실제 루 엘리먼트의 메핑정보는 트리 구조의 XML 문서에서 해당 엘리먼트와 그 자손 엘리먼트를 모두 포함하므로, 트리의 각 엘리먼트 레이블이 나타날 때마다 메핑정보를 통해 알 수 있다. Path는 XML 저장소 내 XML 문서 트리 구조에서의 엘리먼트 경로를 나타낸다. 예를 들어, (그림 2)의 (c)와 같은 XML 루 엘리먼트 레이블에 대해, DTD 메핑태이블은 트리구조에 따라 DTD 메핑정보를 구성한다.

4.1 XML 실제 루의 포함 관계

REAL었습니다. XML 실제 루의 포함 관계는 (그림 3)과 같이 분류할 수 있다[7][8].

(a) XML 실제 루

(b) XML 실제 루의 포함 관계

(c) XML 실제 루의 포함 관계

(d) EID 메핑 태이블

(그림 2) XML 문서 저장 구조의 레이블 구성

(그림 3) XML 실제 루의 포함 관계 XML 실제 루의 포함관계[7]
트의 자식 또는 자식 엘리먼트가 된다.

4.2 XQL 절의 변환

4.2절에서 서술한 바와 같이 (그림 3)의 포함 관계는 해당 엘리먼트 간의 '부모/자식' 또는 '조상/손자'의 계층 관계로 생각할 수 있다. 따라서 (그림 2)와 같은 포함 관계를 분화하기 위해서, XQL 절의 언어는 엘리먼트와 XML XML 속의 부의 엘리먼트 간의 계층 관계를 이용하며, 이를 위해 XQL 절의 EID 매핑 테이블의 엘리먼트 경로로부터 다음과 같은 정보를 얻는다.

• XQL 절의 전체 경로: 노드의 가정에 따라 XQL 절의 경로 표현식으로 이루어진 경로 표현식으로 구성된 XQL 절의 종에서 `/' 또는 `*`를 포함한 XQL 절의 경로, DTD 데이터의 정보를 이용해 부의 엘리먼트로부터 시작하여 원하는 엘리먼트까지 `*/` 안전한 테이블을 이용한 XQL 절의 경로로 변환 가능하며, 이로 XQL 절의 전체 경로를 정의한다. 예를 들어, (그림 1)의 XML 문서에 대해, Catalog/Book/Pages와 같은 XQL 절의가 주어진다면, XML 절의 전체 경로는 Catalog/Book/ Pages와 같이 표시할 수 있다.

• 매핑 정보: XQL 절의 전체 경로 내 EID 매핑 테이블의 엘리먼트 경로에서, 부의 엘리먼트로부터 해당 엘리먼트까지 트리의 높이를 따라 매핑 경로로 정의한다. 예를 들어, Catalog/Book/Pages와 같은 전체 경로와 엘리먼트 경로에 대해, 매핑 경로는 된다.

• EID 매핑 테이블의 경로: XQL 절의 전체 경로 또는 EID 매핑 테이블의 엘리먼트 경로를 매핑 경로에 따라 두 개의 경로로 분할할 수 있다. 주어진 매핑을 중심으로 경로를 분할하였을 때, 분할된 경로의 일부분을 부를 경로, 나머지부분을 자식 경로로 정의한다. 예를 들어, Catalog/Book/Pages와 같은 전체 경로와 엘리먼트 경로를 매핑 경로를 중심으로 분할하였을 때, 부 경로는 Catalog/Book 이 되며, 자식 경로는 Pages 가 된다.

XQL 절의 EID 매핑 테이블의 엘리먼트로부터 얻어진 각각의 매핑 정보의 경로 정보를 비롯하여 (그림 3)과 같은 포함 관계를 분화할 수 있다. 또한 이러한 포함 관계에 따라 주어진 XQL 절의 자식 XML 문서에 대한 절이 XML 전체 경로에 대한 경로로 변환할 수 있다.

XQL 절의 결과와 XML 실제 부의 포함 관계에 따른 결과 변환 알고리즘이(그림 4)와 같이, 사용된 표기 (notation)는 (표 1)에 정리하였다.

<table>
<thead>
<tr>
<th>표 1</th>
<th>포지션</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>QXL 절의 전체 경로</td>
<td>q.level</td>
</tr>
<tr>
<td>target</td>
<td>변환된 경로의 대상 데이터</td>
<td>(0: XML 실제 부의 1: XML 문서의 2: 다른)</td>
</tr>
</tbody>
</table>

(그림 4) XQL 절의 결과로 변환 알고리즘

5. 결론

본 논문에서는 XML 저장소에 있는 XML 실제 부가 있다고 가정하고 이를 이용한 XQL 결과의 변환에 대해 연구하였다. XQL 절의 경로 정보를 XML 실제 부을 이용함으로써, XQL 절의 결과를 빠르게 반환할 수 있으며, 성능을 향상시킬 수 있다. 이를 위해 본 논문에서는 XML 문서 저장 구조로 엘리먼트 테이블과 EID 매핑 테이블을 구성하였고, 이를 이용하여 XML 결과와 XML 실제 부간의 포함 관계를 분화하였으며, 그에 따른 XQL 결과의 변환 알고리즘을 제시하였다.

6. 참고문헌