

XML 뷰 실체화 모델

황대현*, 강현철

중앙대학교 컴퓨터공학과

e-mail:dhhwang@dblabb.cse.cau.ac.kr, hckang@cau.ac.kr

Models of XML View Materialization

Dae Hyun Hwang, Hyunchul Kang
Dept of Computer Science & Engineering
Chung-Ang University

요 약

웹 상의 데이터 교환 표준으로 XML이 등장한 이래 XML 데이터를 관계 DBMS에 저장하고 질의 처리를 수행하는 기법에 관한 연구가 활발히 수행되고 있다. 전통적인 데이터베이스에서처럼 단일 또는 복수개의 XML문서에 대한 XML 질의를 통해 XML 뷰를 정의할 수 있고 이는 질의 처리 성능 향상을 위해 웹 환경에서 실체화될 수 있다. 본 논문에서는 XML 데이터를 관계 DBMS에 저장하였을 때 XML 질의의 결과를 실체뷰로 캐쉬하였다가 동일 질의가 반복될 때 캐쉬를 이용한 응답을 수행하기 위한 XML 뷰 실체화 모델을 세가지 제시하고 비교, 평가한다.

1. 서론

질의 결과를 실체뷰로 캐쉬하여 후속 질의 처리에 이용하는 기법은 중요한 질의 최적화 기법으로서 관계 데이터베이스 시스템에서 활발히 연구되었다 [13][17]. 실체뷰에 관한 주요 문제로는 (1) 뷰의 하부 데이터에 발생한 변경을 실체뷰에 점진적으로 반영하여 실체뷰를 갱신(refresh)하는 문제와 (2) 실체뷰를 이용하여 관련 질의를 재작성(rewrite)하여 처리하는 문제가 있다. 이들 문제는 90년대 후반부터 반구조적 데이터에 대해서 연구되기 시작하였고 [1][25][28][2][12][20] 최근들어 XML 데이터에 대해 연구되고 있다 [4][21][16][15][5][6][14][18][19].

본 논문은 XML 뷰의 실체화에 관한 것으로 즉, 위 두 문제 중 전자에 관한 것이다. 단, 기존 연구들이 XML 데이터의 하부 저장 구조로 반구조적 데이터 저장소(repository)를 대상으로 한 것에 반해 관계 DBMS를 대상으로 한다. XML이 웹 상에서 데이터 교환의 표준으로 등장한 이래 XML 데이터를 현재 가장 널리 사용되고 있는 관계 DBMS에 저장하는 기법이 그 실용적 중요성으로 인해 활발히 연구되고 있는 점을 고려할 때 [7][10][11][22][27], 본 논문에서 취한 관계 DBMS 기반의 XML 뷰 실체화는 중요하다.

XML 데이터를 관계 DBMS에 저장할 경우 기술적인 핵심 문제는 (1) 테이블 스키마 설계, (2) XML 질의의 SQL로의 변환, 그리고 (3) SQL 질의 결과의 XML로의 변환이다. 이들을 XML-관계 사상(XML-Relational mapping)이라 한다. 그림 1은 이들 중 두 번째와 세 번째의 변환을 나타낸 것이다. XML 질의가 주어지면 이는 먼저 질의의 소스 XML 데이터가 저장된 테이블(들)과 그 스키마에 따라 SQL 질의로 변환된다. 이 SQL문(들)이 수행되어 결과 셋 혹은 타플 스트림을 생성하게 된다. 이 결과는 XML 질의가 요구하는 결과 형태를 갖추기 위하여 XML 태깅이 이루어져 최종 XML 질의 결과인 XML 데이터로 변환된다.

본 논문에서는 그림 1과 같은 관계 DBMS를 기반으로 하는 XML 질의 처리 모델 하에서 XML 질의 결과를 캐쉬하는 즉, XML 뷰를 실체화하는 서로 다른 세가지 모델을 제시하고 이들을 비교, 평가한다. 본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 기술한다. 3절에서는 XML 뷰 실체화의 모델 세가지를 제시하고 평가한다. 4절에서는 결론을 맺는다.

2. 관련연구

XML 데이터의 저장소로 관계 DBMS를 이용할 때는 효율적인 XML-관계 사상이 필요하다. 이들에 관한 연구는 관계 DBMS에 XML 데이터를 저장하는 문제 [7][11][22] 그리고 관계 데이터의 XML 출판 문제 [3][8][9][23][24] 등에서 많이 다루어졌다. XML 데이터의 변경 문제는 [26]에서 연구되었다.

[4][21][16]에서는 XML 데이터에 대한 실체뷰의 점진적 갱신에 관해 연구하였다. [4][21]에서는 XML 데이터를 PDOM으로 저장하였을 때 XQL의 일부 기능으로 정의된 XML 뷰를 대상으로 연구하였다. 고려된 XML 변경은 XML 문서의 서브트리의 삽입과 삭제 그리고 리프 노드 값의 수정이다. APIX(aggregate path index)라는 이름의 보조 자료구조를 유지하여 XML 소스의 변경과 뷰와의 연관성을 효율적으로 검사한다. APIX는 뷰가 처음 계산될 때 구성되어 향후 XML 소스의 변경이 일어날 때마다 뷰와 함께 갱신된다.

[16]에서는 XQuery로 정의된 XML 뷰의 결과에 포함되는 XML 엘리먼트의 식별자들로 구성된 XML 뷰 인덱스(XVI)라는 구조를 사용하여 뷰를 실체화하는 기법을 제안하였다. [26]에서 제안한 모든 형태의 XML 변경을 고려하였고 변경이 발생하면 연관된 뷰의 XVI를 효율적으로 갱신하는 알고리즘을 제안하였다. XVI는 엘리먼트의 식별자만을 저장하고 있으므로 뷰가 요청되면 이들을 이용하여 뷰의 내용을 검색하는 과정을 필요로 한다. 그러나 최종 실체뷰를 유지하는 것에 비해 XVI의 용량 소모가 적어 많은 뷰의 실체화를 지원할 수 있다.

기타 관련 연구로 [1][25][28]에서는 반구조적 데이터에 대한 실체뷰의 갱신 문제를 다루었다. 또한 [5][6][14][18][19]에서는 웹 환경에서 XML 실체뷰를 이용하여 관련 질의를 재작성하여 효율적으로 수행하는 기법에 관해 연구하였다.

3. XML 뷰 실체화 모델

3.1. 개요

XML 뷰의 실체화를 모델링하기 위해서는 XML 질의 처리 모델과 XML 변경 모델이 전체되어야 한다. XML 뷰는 XQuery와 같은 XML 질의어 표현식으로 정의되며 따라서 질의의 최종 결과 또는 그것을 도출하기 위한 중간 결과가 XML 실체뷰가 된다. 따라서 XML 뷰 실체화는 XML 질의 처리가 어떤 과정을 통해 이루어지는가에 의존적이다. 또한 XML 뷰 실체화는 뷰의 하부 소스 데이터의 변경을 실체뷰에 반영하여 실체뷰를 점진적으로 갱신하는 것을 그 기본 요소로 하기 때문에 어떤 종류의 XML 변경을 고려하는지에도 영향을 받는다.

본 논문의 XML 질의 처리 모델은 1절의 그림 1로 설명한 것이다. 이는 성능 면에서 이미 검증이 된 관계 DBMS 엔진의 질의 처리 및 최적화 기능을 XML 질의 처리에 이용하자는 것으로, 관계 DBMS에 XML 데이터를 저장하는 문제를 다루는 연구에서 모두 채택하고 있는 모델이다. 한편, 본 논문의 XML 변경 모델로는 [26]의 연구에서 제시된 것을 채택하였다. [26]에서는 XML 질의어에 변경 구문을 추가하기 위하여 XQuery 문법의 확장을 제안하였는데 이 분야의 최초의 연구이자 아직까지는 가장 대표적인 연구이다. 이 모델은 거의 모든 형태의 XML 변경을 취급하고 있다. 또한 변경 대상인 XML 데이터가 관계 DBMS에 저장되어 있을 경우에 제안된 확장 XQuery의 변경 구문을 SQL의 변경 구문으로 변환하는 방법도 제시하였다 (그림 2).

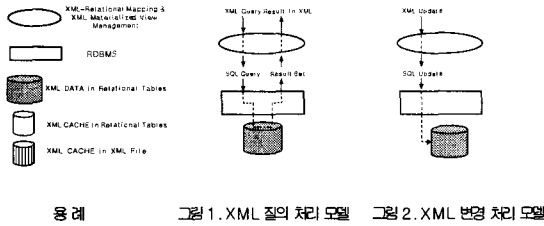
3.2. 세가지 모델

본 절에서는 XML 뷰의 실체화 모델 세가지를 제안한다. 이들은 각각 모델-R, 모델-Rx, 그리고 모델-Fx라 부른다. XML 뷰 실체화 모델의 구성요소로는 크게 네가지를 들 수 있다. (1) 캐쉬되는 데이터 객체, (2) 뷰를 정의하는 XML 질의가 처음 처리되고 그 중간 또는 최종 결과가 캐쉬되는 과정, (3) XML 변경이 처리되고 그로 인해 관련 뷰를 위한 캐쉬를 점진적으로 갱신하는 과정, 그리고 (4) 동일 질의가 반복 제기되었을 때 이를 캐쉬를 이용하여 처리하는 과정이 그들이다.

3.2.1. 모델-R

이 모델은 관계 DBMS 엔진이 관계 뷰의 실체화 기능을 지원할 경우에 적용할 수 있는 것으로 XML 뷰를 정의한 XML 질의가 변환되어 생성된 SQL문(들)의 결과로 생성된 타플 스트림을 캐쉬한다. 관계 DBMS에서 뷰의 계산은 테이블(들)을 대상으로 SQL문을 수행하여 그 결과로 역시 테이블을 얻는 것이다. 즉, 타플 셋(들)을 대상으로 연산을 하여 타플 셋을 결과로 얻는 과정이다. 따라서 관계 뷰의 실체화란 결과 타플 셋을 캐쉬하는 것이며 이것의 점진적 갱신은 소스 타플 셋에 대해 발생한 변경을 반영하는 것이다.

그림 3은 모델-R의 상기 모델 구성 요소 (2), (3), 그리고 (4)의 과정에서 수행되는 처리와 데이터 흐름을 나타낸 것이다: (a) XML 질의가 처음 처리될 때 SQL문에 의해 검색된 타플 셋이 태깅되어 결과를 반환하는 한편 그 타플 셋을 캐쉬하게 된다. (b) 이후 XML 변경이 발생하면 이는 SQL 변경문으로 변환되는데 이들에 의한 소스 테이블 타플의 변경을 캐쉬된 타플 셋에 반영하여 갱신한다. 이 과정은 DBMS 엔진이 담당한다. (c) 동일 질의가 다시 제기되면 이는 캐쉬된 타플 셋에 대한 SQL문으로 변환되어 캐쉬를 검색하고 그 결과를 태깅하여 최종 결과를 반환하게 된다.



이 테이블의 스키마는 실제뷰 갱신에 필요한 컬럼을 갖는다. (b) 이후 XML 변경이 발생하면 이는 SQL 변경문으로 변환되어 이들에 의한 소스 테이블 타플의 변경이 일어나는 것과 함께 이와는 별도의 과정으로 XML 변경의 의미를 반영하여 캐쉬된 타플 셋을 점진적으로 갱신한다. (c) 동일 질의가 다시 제기되면 이는 캐쉬된 타플 셋에 대한 SQL문으로 변환되어 캐쉬를 검색하고 그 결과를 태깅하여 최종 결과를 반환하게 된다. 이들 과정은 관계 DBMS 상에서 작동하는 XML-관계 사상과 XML 실체뷰 관리를 담당하는 모듈에 의해 수행된다.

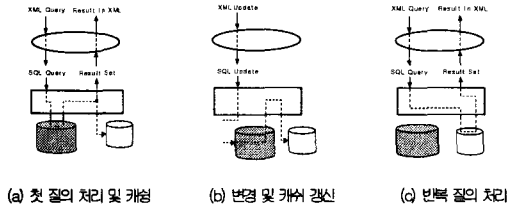


그림 3. XML 뷰 실제화: 모델-R

3.2.3. 모델-Fx

이 모델에서는 캐쉬의 대상이 다른 두 모델과는 달리 XML 질의의 최종 결과인 XML 문서 파일이다. 그림 5는 모델-Fx의 상기 모델 구성 요소 (2), (3), 그리고 (4)의 과정에서 수행되는 처리와 데이터 흐름을 나타낸 것이다. (a) XML 질의가 처음 처리될 때 SQL문에 의해 검색된 타플 셋이 태깅되어 결과를 반환하는 한편, 이 최종 질의 결과로 생성된 XML 데이터 (또는 문서) 파일을 캐쉬한다. 이는 관계 DBMS 테이블 내에 CLOB 컬럼으로 저장되거나 파일 시스템에 저장될 수도 있다. (b) 이후 XML 변경이 발생하면 이는 SQL 변경문으로 변환되어 이들에 의한 소스 테이블 타플의 변경이 일어나는 것과 함께 이와는 별도의 과정으로 XML 변경의 의미를 반영하여 캐쉬된 XML 데이터를 점진적으로 갱신한다. (c) 동일 질의가 다시 제기되면 캐쉬된 XML 데이터를 바로 반환한다. 이들 과정은 관계 DBMS 상에서 작동하는 XML-관계 사상과 XML 실체뷰 관리를 담당하는 모듈에 의해 수행된다.

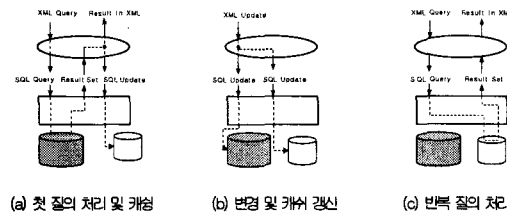


그림 4. XML 뷰 실제화: 모델-Rx

4. 비교 및 평가

전절에서 제안된 세가지 모델을 비교하면 서로 간에 장단점이 있다. 모델-R은 관계 DBMS 엔진의 실체뷰 기능을 이용하므로 구현이 가장 간단하다. 그리고 XML-관계 사상 및 실체뷰 관리 모듈로부터 요청되는 SQL문이 다른 모델에 비해 적다는 장점이 있다. 그러나 실체뷰의 관리에 있어 뷰를 정의하는 XML 질의나 XML 변경의 의미를 이용할 여지가 없는 것이 최대 단점이다. 또한 뷰의 내용을 구성하기 위해 검색된 타플 셋을 캐쉬하므로 동일 질의의 제기 시 태깅 과정은 매번 수행되어야 하는 단점도 있다.

3.2.2. 모델-Rx

이 모델은 관계 DBMS 엔진이 제공하는 관계 뷰의 실체화 기능에 의존하지 않는다. 이 모델에서 캐쉬의 대상은 모델-R의 그것을 XML 질의의 의미(semantic)를 반영하여 한번 더 처리 과정을 거친 타플 셋이다. 또한 XML 변경 발생 시 캐쉬된 타플 셋의 점진적 갱신은 XML 변경의 의미를 반영하여 수행된다. 그림 4는 모델-Rx의 상기 모델 구성 요소 (2), (3), 그리고 (4)의 과정에서 수행되는 처리와 데이터 흐름을 나타낸 것이다: (a) XML 질의가 처음 처리될 때 SQL문에 의해 검색된 타플 셋이 태깅되어 결과를 반환하는 한편 이 타플 셋으로부터 질의 결과에 불필요한 내용은 제거한 후 별도의 테이블에 저장한다.

한편, 모델-Rx와 모델-Fx는 실체뷰의 관리에 있어 XML 질의나 XML 변경의 의미를 이용하므로 모델-R에 비해 캐쉬 갱신 과정이 더 효율적이다. 모델-Rx는 뷰의 최종 결과 구성을 위한 태깅 작업이 뷰 요청 때마다 매번 이루어지는 점이 모델-R과 같지만 모델-Fx는 태깅 과정이 필요없는 것이 장점이다. 그러나 모델-Fx에서 캐쉬된

데이터는 XML 포맷이므로 점진적 갱신을 위해서는 이를 DOM 트리 등으로 파싱하는 과정이 필요하다. (파싱 부담을 덜기 위해 PDOM 포맷을 사용할 수도 있다.)

5. 결론

본 논문에서는 XML 데이터를 관계 DBMS에 저장하였을 때 XML 질의의 결과를 실체뷰로 캐쉬하였다가 동일 질의가 반복될 때 캐쉬를 이용한 응답을 수행하기 위한 XML 뷰 실체화 모델을 세가지 (모델-R, 모델-Rx, 그리고 모델-Fx) 제시하였다. 웹 환경에서 원격 XML 소스에 대한 XML 질의 결과를 실체뷰로 유지하는 기법의 필요성과 가장 널리 활용되고 있는 관계 DBMS에 XML 데이터를 저장하고 검색하는 기법의 중요성을 감안할 때 본 논문에서 제시한 모델들은 실용적 연구 가치를 갖는다 하겠다. 제시된 모델 중 모델-Rx에 기반을 둔 실체뷰를 지원하는 XML 저장 시스템 프로토타입은 오라클 8i를 이용하여 Java로 Windows Server 2000 상에서 이미 개발되었다 [15]. 개발된 시스템으로 수행된 성능평가 결과 XML 뷰 실체화를 통한 XML 질의 처리의 효율성을 확인할 수 있었는데 이는 모델-Rx의 실용성을 의미한다.

참고문헌

- [1] S. Abiteboul et al., "Incremental Maintenance for Materialized Views over Semistructured Data," Proc. Int'l Conf. on VLDB, 1998, pp. 38-49.
- [2] D. Calvanese et al., "Answering Regular Path Queries Using Views," Proc. Int'l Conf. on Data Eng., pp. 389-398, 2000.
- [3] M. Carey et al., "XPERANTO: Publishing Object-relational Data as XML," Proc. Workshop on the Web and Databases, May 2000.
- [4] L. Chen and E. Rundensteiner, "Aggregate Path Index for Incremental Web View Maintenance," Proc. 2nd Int'l Workshop on Advanced Issues of E-Commerce and Web-based Information Systems, 2000.
- [5] L. Chen and E. Rundensteiner, "ACE-XQ: A Cache-aware XQuery Answering System," Proc. Workshop on the Web and Databases, 2002.
- [6] L. Chen et al., "XCache - A Semantic Caching System for XML Queries," Proc. ACM SIGMOD Int'l Conf. on Management of Data, 2002.
- [7] A. Deutsch et al., "Storing Semistructured Data with STORED," Proc. ACM SIGMOD Int'l Conf. on Management of Data, 1999, pp. 431-442.
- [8] M. Fernandez, et al., "SilkRoute: Trading Between Relations and XML," Proc. the 9th WWW Conf., 2000, pp. 723-746. 2000.
- [9] M. Fernandez et al., "Efficient Evaluation of XML Middle-ware Queries," Proc. ACM SIGMOD Int'l Conf. on Management of Data, 2001, pp. 103-114.
- [10] D. Florescu and D. Kossmann, "Storing and Querying XML Data Using an RDBMS," IEEE Data Eng. Bulletin, Vol. 22, No. 3, Sep. 1999, pp. 27-34.
- [11] D. Florescu and D. Kossmann, "A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database," Tech. Rep., INRIA, France, 1999.
- [12] D. Florescu et al., "Query Containment for Conjunctive Queries with Regular Expressions," Proc. Int'l Symp. on PODS, 1998, pp. 139-148.
- [13] A. Gupta and I. Mumick, "Materialized Views: Techniques, Implementations, and Applications," 1999, MIT Press.
- [14] V. Ilristidis and M. Petropoulos, "Semantic Caching of XML Databases," Proc. Workshop on the Web and Databases, 2002.
- [15] H. Kang, H. Sung, and C. Moon, "Deferred Incremental Refresh of XML Materialized Views: Algorithms and Performance Evaluation," Proc. the 14-th Australasian Database Conf., Feb. 2003, pp. 217-226.
- [16] Y. Kim, C. Moon, and H. Kang, "XML View Indexing: Issues and Solutions," Proc. Int'l Conf. on Information and Knowledge Engineering, Jun. 2002, pp. 327-333.
- [17] A. Levy et al., "Answering Queries Using Views," Proc. of ACM Int'l Symp. on PODS, 1995.
- [18] P. Marron and G. Lausen, "Efficient Cache Answerability for XPath Queries," Proc. the 2nd Int'l Workshop on Data Integration over the Web, 2002, pp. 35-45.
- [19] C. Moon, S. Kim, and H. Kang, "Processing XML Path Expressions Using XML Materialized Views," to appear in Proc. the 20-th British Nat'l Conf. on Databases, Jul. 2003.
- [20] Y. Papakonstantinou and V. Vassalos, "Query Rewriting for Semistructured Data," SIGMOD Proc. Int'l Conf. on Management of Data, pp. 455-466, 1999.
- [21] L. Quan et al., "Argos: Efficient Refresh in an XQL-Based Web Caching System," Proc. Workshop on the Web and Databases, 2000, pp. 23-28.
- [22] J. Shanmugasundaram et al., "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. Int'l Conf. on VLDB, 1999, pp. 302-314.
- [23] J. Shanmugasundaram et al., "Efficiently Publishing Relational Data as XML Documents," Proc. Int'l Conf. on VLDB, 2000, pp. 65-76.
- [24] J. Shanmugasundaram et al., "Querying XML Views of Relational Data," Proc. Int'l Conf. on VLDB, 2001, pp. 261-270.
- [25] D. Suciu, "Query Decomposition and View Maintenance for Query Languages for Unstructured Data," Proc. Int'l Conf. on VLDB, 1996, pp. 227-238.
- [26] I. Tatarinov et al., "Updating XML," Proc. ACM SIGMOD Int'l Conf. on Management of Data, 2001, pp. 413-424.
- [27] F. Tian et al., "The Design and Performance Evaluation of Alternative XML Storage Strategies," ACM SIGMOD Record, Vol. 31, No. 1, Mar. 2002, pp. 5-10.
- [28] Y. Zhuge and H. Garcia-Molina, "Graph Structured Views and Their Incremental Maintenance," Proc. Int'l Conf. on Data Engineering, 1998, pp. 116-125.