내장형 XML 저장 및 검색 시스템의 구현*

권준호° 권동섭 홍석진 곽민성 임우규 신효섭* 이석호
서울대학교 전기 컴퓨터공학부
{bluerain°, subby, jinny, cuty, wgihm}@db.snu.ac.kr hyoseop@samsung.com* shlee@cse.snu.ac.kr

Implementation of embedded XML Storage and Retrieval System

Joonho Kwon° Dongseop Kwon Seokjin Hong Minsung Kwak Woogyu Ihm Hyoseop Shin* Sukho Lee
School of Electrical Engineering and Computer Science, Seoul National University
Samsung Electronics Co., Ltd.*

요 약

XML (eXtensible Markup Language)은 확장성과 유연성을 통해 인터넷 상에서 데이타를 표현하고 교환하는 중요한 표준으로 자리잡고 있으며, XML 기반의 문서의 양도 증가하고 있다. 이러한 XML의 광범위한 사용에 따라 XML 저장 및 검색 시스템의 필요성이 증대되고 있다.

본 논문에서는 방대한 XML 데이타를 효율적으로 처리하고 검색하기 위해 XML 문서를 관계형 데이타베이스에 저장하고, 질의 언어로 XQuery를 사용하는 시스템을 설계하고 구현한다. 또한 다양한 XQuery의 실행을 통하여 제안한 시스템의 성능을 평가한다.

1. 서 론

XML(eXtensible Markup Language)[1]은 특유의 확장성과 유연성으로 인하여 인터넷 상에서 데이타를 표현하고 교환하는 표준안으로 이미 자리잡고 있다. XML은 이미 인터넷을 이용한 정보의 교환이나 응용 통합을 위한 핵심적인 기술로 논의되고 있다. 그리고, 이미 TV-Anytime[2], MPEG-7, ebXML, DocBook, XML-EDI, SyncML, UPnP 등의 다양한 응용 환경에서 사용되고 있다. 이러한 XML 문서의 광범위한 사용과 XML 기반 데이타의 증가에 따라 XML 데이타를 효율적으로 저장 및 검색하는 방법의 필요성이 증대되고 있다.

또한 XML은 그 구조적인 특징 때문에 기존의 RDBMS에 사용하는 SQL과는 달리 경로식에 기반한 구조 검색 기법이 반드시 필요하다.

본 논문에서는 하부에 RDBMS를 이용하여 XML 문서를 저장하고, 질의 언어로서 XQuery[3]를 사용하는 내장형 XML 저장 및 검색 시스템을 구현하였다. 또한 다양한 XQuery 질의를 실행하여 구현한 시스템의 성능을 평가하였다.

2. 관련 연구

XML을 저장 및 검색하기 위한 연구로서 전용의 시스템을 사용하는 eXcelon[4], Tamino[5]와 같은 연구가 있었으며, XML 문서를 RDBMS의 테이블에 저장하기 위한 방법으로 번호 부여 기법(numbering Scheme)[6] [7]에 관한 연구가 있었다.

3. 시스템의 구조

3.1 개요

XML 저장 및 검색 시스템은 하부의 RDBMS SQL 엔진 위에서 XML 문서에 대한 저장, 검색 기능을 수행한다. 시스템의 전체 구조는 그림 1과 같다.

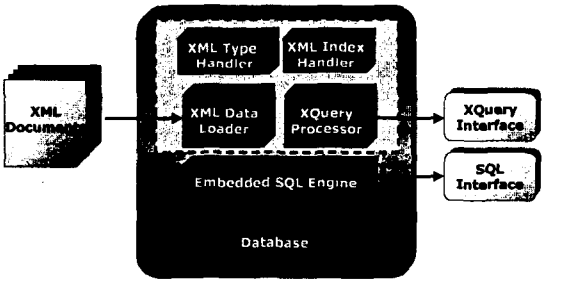

그림 1. 시스템의 구조

XML 데이타 로더는 XML 문서를 파싱하여 RDBMS의 테이블에 저장한다. XML 질의 처리기는 사용자로부터 XQuery 질의문을 입력 받아 이를 SQL문을 이용하여 처리하고 그 결과를 사용자에게 반환한다.

XML 데이타 로더와 XML 질의 처리기를 효율적으로 수행하기 위해서는 XML 문서의 각 노드의 데이타 타입을 찾아내야 할 필요가 있다. XML 타입 핸들러는 미리 정의된 데이타 타입 정보를 바탕으로 각 노드의 데이타 타입을 알려주는 역할을 수행한다.

XQuery 질의문에는 사용자가 자주 요구하는 엘리먼트가 있다. 이런 엘리먼트들에 대해 인덱스를 생성하면 질의 처리 시 빠른 속도를 기대할 수 있다. XML 인덱스 핸들러는 인덱스의 생성과 삭제, 검색에 대한 요청을 처리하는 역할을 수행한다.

3.2 저장 구조

XML 문서를 RDBMS에 저장하기 위하여 [6], [7] 등의 논문에서 제시한 방법과 유사한 번호 부여 기법을 사용한다. 이러한 방법을 이용하면 XML 문서 내부의 구조

* 이 논문은 2003년도 두뇌한국21사업에 의하여 지원되었음

를 나타내는 DTD나 XML Schema[8] 등의 정보가 없는 XML 문서도 저장할 수 있을 뿐 아니라, 서로 다른 구조를 가지는 다양한 XML 문서를 하나의 시스템에 저장할 수 있는 장점이 있다.

번호 부여 기법은 XML 문서를 각 노드 별로 분리하여 저장하고 이 때 각 노드간의 계층 구조는 시작 위치(start position)과 종료 위치(end position)의 두 번호를 이용하여 저장한다. XML 문서의 하부 노드는 항상 상부 노드에 완전히 포함되는 특징이 있으므로, 이 두 번호를 이용하면 특정한 노드가 어떤 다른 노드와 조상~후손 관계를 가지는지 알 수 있다.

본 논문에서는 표 1과 같은 테이블들을 사용한다.

표 1. 테이블의 종류

테이블 종류	설명
NODE	XML 문서의 각 노드와 애트리뷰트에 관련된 정보 저장
DOCUMENT	XML 문서의 원문을 불필요한 공백을 제외하여 저장
VALUE (int, float, string,text, datetime)	각 노드가 가지는 실제 값을 값의 타입에 맞게 나누어서 저장

3.3 질의 언어

질의 언어는 XQuery는 2002년 4월에 수정된 XQuery 1.0의 문법을 따르고 있으며, XQuery 문법에서 핵심적인 부분을 선별하였으므로 XQuery의 핵심 부분 집합(core subset)이라 할 수 있다. XQuery의 전체 문법은 매우 방대한데, 본 시스템에서 채택한 FOR, WHERE, RETURN, SORTBY의 구문을 이용하면 사용자가 원하는 대부분의 질의를 구성할 수가 있다.

```
FOR   $pi in //ProgramInformation,
      $be in //BroadcastEvent
WHERE $be//PublishedTime >= '2002-08-21T13:00:00'
  AND $pi/@programId = $be/@crid
RETURN $pi//Title/text(), $be//PublishedTime/text()
```

위의 XQuery 질의문은 2002년 8월21일 13시 이후에 방송을 시작하는 프로그램의 제목과 방송시간에 대한 정보를 보기 위한 질의이다.

3.4 질의 처리 방법

본 논문의 시스템에서 질의 처리는 두 단계로 구분된다. SQL을 이용하여 XQuery의 FOR절과 WHERE절을 처리하는 전처리 단계와 SQL의 결과로 돌아온 내용을 분석하여 XQuery의 RETURN절을 처리하는 후처리 단계로 나눌 수 있다.

사용자가 입력한 XQuery를 SQL의 변환하는 전처리 단계는 그림 2와 같다.

```
입력 : XQuery  출력 : SQL
1. FOR 절에 나타난 정규 경로식을 SQL로 변환
2. WHERE 절에 나타난 식을 타입 정보와 인덱스 정보를
   참고하여 SQL로 변환
3. RETURN절에 나타난 변수 부분만 SQL로 변환
```

그림 2. XQuery의 SQL 변환 과정

후처리 단계는 그림 3과 같은 동작을 수행한다.

```
입력 : XQuery의 RETURN절, XQuery의 SORTBY절,
       SQL 문
출력 : XQuery의 최종 결과

1. 입력으로 들어온 SQL을 수행
2. SQL의 결과 중에서 RETURN절의 내용만 선택
3. SORTBY절의 컬럼으로 SQL 결과를 정렬
4. RETURN절에 나타난 태그를 만들어 최종 결과 생성
```

그림 3. 후처리 단계의 수행 과정

4. 성능 평가

4.1 실험 환경

실험은 768MB의 메모리를 가지고 있고, 리눅스 커널 2.4.2 버전이 설치된 Intel Pentium III 1GHz CPU 기계에서 수행하였다. 시스템의 개발 언어는 C와 C++을 사용하였으며, 하부의 RDBMS로 MySQL[9]을 사용하여 시스템의 성능을 측정하였다.

실험에 사용한 데이타는 TV 방송 프로그램과 관련된 정보를 표현하는 TV-Anytime[2]의 메타데이타의 규격을 따라서 생성하였다. 작은 크기의 문서는 3일 분량의 소규모 채널의 방송 데이타만 가지고 있는 525KB이고, 큰 크기의 문서는 1주일 분량의 모든 채널의 방송 데이타를 가지고 있는 3.4MB 정도의 문서이다.

4.2 실험 결과

큰 크기의 XML 문서에 대해서는 표2와 같은 질의를 수행하고 처리 시간을 측정하였다.

표 2. 큰 크기의 문서에 사용한 질의의 특성

q1	의미	인어 아가씨와 대망의 출연진 목록
	특성	괄호, AND, OR 연산자의 사용
	질의	FOR $g in //GroupInformation//BasicDescription, $cre in $gi//CreditsItem WHERE ($g//Title = '인어 아가씨' OR $gi//Title = '대망') AND $g contains $cre AND $cre/@role = '출연진' RETURN $g//Title/text(), $cre//GivenName/text()
q2	의미	신동엽이 출연하는 프로그램 목록
	특성	OR 연산자의 사용
	질의	FOR $pi in //ProgramInformation, $cre in //ProgramInformation//CreditsItem WHERE $pi contains $cre AND $cre//GivenName = '신동엽' RETURN $pi//Title/text()

작은 크기의 XML 문서에 대해서는 표3과 같은 질의를 사용하여 처리 시간을 측정하였다.

표 3. 작은 크기 문서에 사용한 질의의 특성

q3	의미	'대박가족'의 출연진 목록
	특성	조인 형태의 질의, CONTAINS 연산자 사용
	질의	FOR $pi in //ProgramInformation, $cre in $pi//CreditsItem WHERE $pi//Title = '대박가족(107회)' AND $pi contains $cre AND $cre//Role = '출연진' RETURN $pi//Title/text(), $cre//GivenName/text() SORTBY (GivenName)
q4	의미	2002년 8월21일 19시 이후 혹은 2002년 8월22일 19시 이후에 방송하는 드라마의 제목과 방송시간을 시간대로 정렬
	특성	조인 형태, 괄호의 사용, OR와 AND 연산자 사용, 태그를 사용한 RETURN 절
	질의	FOR $pi in //ProgramInformation, $be in //BroadcastEvent WHERE (($be//PublishedTime > '2002-08-21T19:00:00') OR ($be//PublishedTime > '2002-08-22T19:00:00')) AND $pi//Genre/Name = '드라마' AND $pi/@programId = $be//@crid RETURN <Drama> <Title>{$pi//Title/text()}</Title> <PublishedTime>{$be//PublishedTime/text()} </PublishedTime></Drama> SORTBY (PublishedTime)

표 2와 3의 질의를 수행한 결과는 그림 2와 3과 같다.

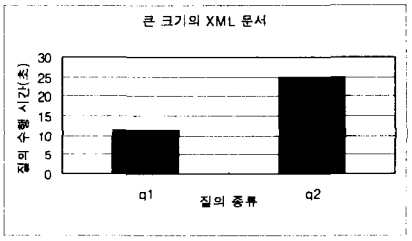

그림 2. 큰 크기 문서에서 질의 처리 결과

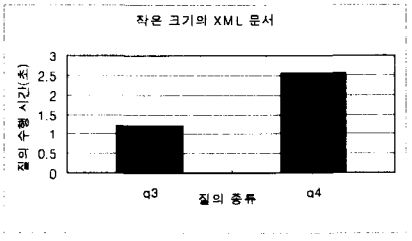

그림 3. 작은 크기 문서에서 질의 처리 결과

5. 결 론

본 논문에서는 XML 문서를 저장하고, 검색할 수 있는 시스템을 제안하였다. 이 시스템에서는 XML 문서를 위한 질의 언어로는 최근 XML 표준 질의 언어로 제안된 XQuery를 이용하므로써 데이타의 내용뿐 아니라 XML 문서의 구조도 경로 질의를 이용하여 쉽게 처리할 수 있는 표준 질의 인터페이스를 제공한다. 또한 대량의 XML 문서 환경에서도 효율적으로 데이타를 처리할 수 있도록 하위 저장구조로는 관계 데이타베이스 엔진을 기반으로 한다.

향후 과제로서 XML 스키마도 지원하도록 타입 핸들러를 개선하고, 완전한 XQuery의 형식을 지원할 수 있도록 질의 언어를 확장할 수 있을 것이다.

참고문헌

[1] Tim Bray, Jean Paoli, C.M. Sperberg-McQueen and Eve Maler, "Extensible Markup Language(XML) 1.0 second edition W3C recommendation," Technical Report REC-xml-20001006, World Wide Web Consortium, October 2000.

[2] TV Anytime Forum, http://www.tv-anytime.org.

[3] XQuery 1.0 : An XML Query Language, http://www.w3.org/TR/xquery/

[4] eXcelon, http://www.exln.com

[5] H. Schoning, "Tamino - A DBMS designed for XML," Proc. of the 17th ICDE conference, April 2001

[6] C. Zhang, J. Naughton, D. Dewitt, Q. Luo, G. Lohman, "On Supporting Containment Queries in Relational Database Management Systems," Proc. of the 2001 ACM SIGMOD Conference, May 2001

[7] Q. Li and Bongki Moon. "Indexing and Querying XML Data for Regular Path Expressions," Proc. of the 27th VLDB Conference, 361-370, September 2001

[8] XML Schema, http://www.w3.org/XML/Schema

[9] MySQL, http://www.mysql.com/