ORDBMS를 이용한 XML문서의 저장 및 질의

박성희[U] 박경현 김록원 남광우 류근호
충북대학교 데이터베이스 연구실
{shpark, khpark, rwkim, kwnam, khryu}@dblab.chungbukac.kr

Storing and Querying XML Data using ORDBMS

Sung Hee Park[U] Kyoung Hyun Park Kwang Woo Nam Rok Won Kim Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요 약

현재 XML 문서를 저장하고 이에 대한 질의를 처리하는 백엔드 저장소로써는 파일시스템, 기존의 RDBMS와 OODBMS를 이용하는 접근 방법이 있다. 또한 독자적으로 semistrucured 데이터에 대한 저장 및 질의를 처리 할 수 있는 데이터베이스 시스템이 존재한다. 따라서, 이 논문에서는 기존의 응용프로그램에서 이용하는 데이터와 통합을 잘 할 수 있는 RDBMS의 장점과 객체지향 DOM모델을 지원할 수 있는 OODBMS의 특징을 모두 수용할 수 있는 ORDBMS에서 XML 문서를 저장하고 저장된 데이터에 대한 질의를 할 수 있는 XML문서 처리시스템을 설계한다. 여기서, XML문서의 논리적 구조가 정해져 있지 않는 XML문서를 ORDBMS의 테이블 형태로 저장하는 여러 가지 방법을 제시하고, semistructured 데이터에 대한 질의의 특징인 패스표현을 효율적으로 지원하기 위해 패스 인덱스의 개념을 제시한다. 이렇게 함으로써 XML문서에 대한 질의를 ORDBMS에서 처리할 때 효율성을 높일 수 있다.

1.서론

XML을 포함하는 semistructured 데이터를 관리하기 위한 연구가 현재 계속 진행 중에 있다. 이러한 연구결과 XML문서를 처리할 수 있는 효율적인 시스템들이 등장하고 있다. 이러한 연구는 XML를 저장 및 질의할 수 있는 백엔드 데이터베이스 시스템으로 파일시스템, 기존의 RDBMS와 OODBMS 또는 semistructured 데이터를 지원하기 위한 독자적인 저장관리시스템이 이용된다[ABIT97,DEAU99a,MCHU97]. 그러나 이러한 시스템중 범용적으로 쓰여질 수 있도록 명확하게 검증된 것은 없다[FLOR99].

기존의 상업용 RDBMS를 사용할 경우, RDBMS의 우수한 성능의 이용과 기존의 응용프로그램에서 이용하는 데이터와 XML데이터에 대한 질의가 가능하다는 장점이 있다. 하지만 엘리먼트를 테이블로 표현시 많은 조인을 유발하고 set-valued 속성을 지원하지 않는 제약사항을 갖는다[SHAN99]. 또한 OODBMS를 이용 할 경우에는 엘리먼트에 대한 클러스터링을 기존의 시스템보다 더 잘 할 수 있다. 그러나 OODBMS자체가 대용량의 데이터에 대한 질의 처리가 성숙되어 있지 않다는 단점이 있다[FLOR99]. RDBMS와 OODBMS의 장점을 수용한 ORDBMS를 백엔드로 이용할 경우, 계층적 질의를 조인을 이용한 것보다 빠르게 처리할 수 있고 대용량의 데이터에 대한 질의를 안정성 있게 처리할 수 있는 장점을 갖는다. 그럼에도 불구하고 ORDBMS에서의 XML 문서 저장과 검색[SHIM99]에 관한 연구는 대부분 XQL을 지원하고 다른 XML문서와의 링크정보를 저장하지 않으므로 이러한 정보에 대한 검색이 불가능 하다는 단점을 갖는다.

이 논문에서는 XML을 저장하는 백엔드 데이터베이스로 ORDBMS를 이용하고 XML문서의 논리적 구조가 정해져 있지 않은 XML문서를 바탕으로 ORDBMS에 저장과 XML 데이터에 대한 질의가 수행될 수 있도록 XML-QL을 SQL로 변환할 수 있는 시스템을 설계한다. 또한, XML문서를 저장할 수 있는 효율적인 방법과 질의에서 이용되는 패스 표현의 처리시 효율을 높일 수 있는 패스 인덱스를 사용하는 방법을 제시한다.

2. 관련연구

2.1 관계형 접근

STORED[DEUT99b]에서는 DTD가 없는 XML문서와 같은 semistructured 데이터를 관계형의 테이블 형태로 저장하는 방법을 연구하였으며, 이를 위해 데이터마이닝 알고리즘을 이용하였다. 현재 이용되는 semistructured 데이터에 대한 질의 언어들과 다른 질의언어인 STORED는 Lorel[ABIT97], XML-QL에 대한 패스 표현을 할 수 있는 기능이 제한적이다.

오라클8[ORAC98]에서는 관계형 엔진을 이용하여 XML문서에 질의할 수 있도록 기본적인 지원을 제공한다. 그러나 문서의 스키마로부터 질의 변환은 자동적으로 이루어지지 않고 수동적으로 이루어진다. 추가적으로 오라클 8에서는 XML문서에 대한 semistructured적 질의에 대한 지원을 제공하지 않고 질의결과를 XML문서로 변환을 위한 기본적인 기능을 지원한다.

관계형적 접근은 속성의 값으로 셋값을 지원하지 않기 때문에 엘리먼트의 저

장시 여러 개의 테이블로 나누어 저장된다. 따라서 하나의 단순 질의를 처리하기 위해 많은 조인이 발생하게 된다. 질의 결과를 복잡한 XML구조체로 표현이 어렵다는 단점이 있다[SHAN99,DEAU99a,FLOR99]. 그러나, RDBMS의 우수한 성능의 이용과 기존의 응용프로그램에서 이용하는 데이터와 XML데이터에 대한 질의가 가능하며 데이터의 통합이 쉽다는 장점이 있다[SHAN99,DEAU99a,FLOR99].

2.2 관계형에서의 XML문서의 저장

XML문서를 테이블로 매핑하여 저장하는 방법에 대한 기존의 연구는 다음과 같은 세 가지 방법으로 나누어진다[FLOR99,SHAN99].

• Bulk데이터 타입: 이 방법은 태그와 데이터를 포함한 XML문서를 직접 저장하는 방법이다. 이러한 접근 방법은 문서의 어셈블링이 효율적이지만 문서 검색을 위한 특별한 인덱스가 필요하고 문서 검색시 문서를 항상 파싱해야 하는 단점이 있다.

• Edge table: 하나의 테이블은 각각의 엘리먼트를 표현한 객체의 edge에 대한 정보로 표현한다. 하나의 튜플은 parentID, Label, Type, order와 ChildID로 구성된다. 각 속성에 대한 상세한 설명은 4.3에서 기술한다. 실제 값과 링크에 적합한 인덱스를 만들면 대용량의 데이터베이스에서 비효율적인 스캔을 피할 수 있다. 이러한 접근 방법은 복잡한 문서를 어셈블링하는데 오버헤드를 초래하나 복잡하지 않은 데이터에 접근하는 것은 bulk 데이터 타입보다 효율적이다.

• Binary table: 동일한 라벨을 갖는 edge를 하나의 분리된 테이블로 저장하고 다른 edge들은 위에 있는 edge테이블에 저장한다. 따라서 이렇게 분리된 테이블은 binary 테이블이며 이것은 다음과 같은 구조를 갖는다.

B_{label} (ParentID, ChildID, order,type)

이 방법은 Edge table방법과 같은 단점을 가진다. 그러나 주어진 라벨에 대한 엘리먼트를 검색할 때 유리하다.

3. 시스템 구성도

이 논문에서 제안한 XML문서 처리기 시스템은 XML문서 저장기, XML-QLTOSQL질의 변환기,XML문서 구성기로 구성된다. XML 문서 저장기는 파싱된 XML문서를 데이터베이스 내부에 테이블 스키마에 적합하게 저장하는 모듈이다. XML-QLTOSQL질의변환기는 XML-QL을 이용한 XML문서에 대한 질의를 DB에 존재하는 XML데이터에 질의하기 위한 SQL로 변환한다. 이렇게 변환된 SQL을 실행하여 얻은 결과는 테이블 형태의 결과이므로 이것을 다시 XML문서 형태로 재구성하는 XML 문서 구성기가 필요하다.

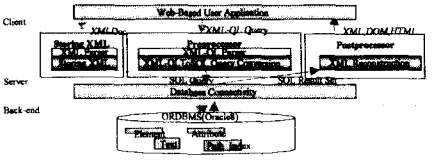

그림1. XML문서처리기 시스템 구성도

4. XML 문서의 모델링 및 저장

4.1 XML 데이터 모델

많은 연구에서 XML문서의 모델링을 위해 labeled directed graph을 이용한다. XML을 모델링할 때 순서가 있는 OEM 모델을 바탕으로 서로 다른 edge 타입을 확장한다. 하나의 XML문서는 객체들의 콜렉션이고 각각의 객체들은 단순 또는 복합객체이다. 단순 객체의 값은 모두 string으로 처리하며 복합객체는 라벨과 하위객체들의 집합(set(label, Oid))으로 구성된다[ABIT97]. XML문서에서의 링크를 지원하기 위해 두 가지 타입의 링크 타입을 지원한다. 하나의 문서내의 링크 타입을 나타내는 aggregation link와 다른 문서사이의 연결을 나타내는 association link 타입이다. 이러한 링크 타입은 XML문서를 어셈블링할 때 이용되며 하나의 XML문서를 어셈블링할 때는 aggregation link를 이용된다. 또한 association link를 이용하여 M대 N관계의 XML문서를 어셈블링이 가능하다.

4.2 XML 문서의 저장

XML 문서를 저장하기 위한 방법으로 set-valued 속성을 지원하는 ORDBMS의 특징을 이용하는 multiple-edge table방법과 기존 edge table 방법을 확장하는 두 가지 방법을 제시한다.

기존의 연구에 따르면 엘리먼트가 가지는 어트리뷰트와 XML문서의 링크에 대한 정보를 저장하는 부분이 미약하다. XML문서의 모델링에서 언급한 것처럼 링크를 저장할 수 있도록 Link Type 속성을 edge 테이블에 추가하고 XML문서의 어트리뷰트를 저장하기 위한 attribute 테이블을 추가한다. 이 논문에서는 edge table 방법을 확장하는 것을 선택한다. 이것에 대한 자세한 사항은4.3에서 설명된다.

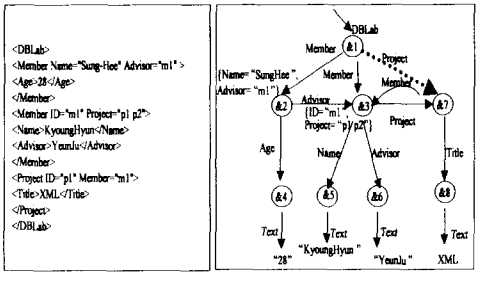

그림2. XML문서와 그래프

• multiple-edge table : 하나의 튜플은 주어진 하나의 노드에 대해 모든 외부 edge를 가지는 parent 테이블과 하나의 edge에 대한 정보를 가지는 중첩된 edge 테이블로 구성된다. parent 테이블은 parent-OID, edge의 배열로 구성된다. edge 테이블은 label, type ,child-OID로 구성된다. 이 방법은 배열을 이용하므로 edge의 순서가 별도의 속성이 없이도 알 수 있다. 그러나 대부분의 ORDBMS에서는 배열값의 속성에 대한 검색은 어려운 연산으로 남아있다.

4.3 Edge table 저장 방법

이 논문에서는 edge table방법을 지원한다. 그래프 상에 있는 노드는 엘리먼트를 나타내며 각각 eid에서 의해서 식별된다. 데이터베이스에 저장된 XML 데이터에 대한 접근을 위해서 XML문서별로 클러스터링을 한다. 따라서 클러스터 인덱스를 RootID를 이용하고 각 테이블에 문서의 RootID가 반복된다.

XMLDocument테이블은 XML문서의 파일이름 및 문서의 RootID를 나타낸다. XML-QL에서의 in Clause를 처리하기 위해 요구된다. Element 테이블은 각각의 edge에 대한 정보를 튜플로 가지며 그림3과 같다. RootID는 edge를 후손으로 가지는 Root객체의 OID이다. ParentID와 ChildID는 edge의 시작 노드와 다겟노드를 나타내며 label은 두 개의 노드를 연결하는 링크의 라벨을 나타낸다. 링크타입은 링크가 aggregation또는 association인지를 나타낸다. C.LType은 자손노드의 타입을 나타낸다. 만일 자손노드가 값을 가지는 단순 객체이면 " string" 값을 갖고 하위에 다른 노드를 가지면 " NODE" 값을 갖는다. attribute 테이블은 엘리먼트가 가지는 어트리뷰트를 나열된 순서와 함께 attribute의 이름과 그 값을 저장한다. PathIndex 테이블은 패스표현을 처리하기 위해 패스의 깊이가 2인 패스 인덱스를 테이블로 만든다. 이러한 pathIndex 테이블은 패스 표현을 처리하기 위한 element 테이블에 대한 조인 인덱스이다.

■XMLDocument

DocID	RootID	DocFileName
1	1	dbmem.xml
2	9	bib.xml

■Text

RootID	ElementID	Value
1	5	KyoungHyun
:	:	:
9	8	XML

■Element

RootID	ParentID	ChildID	Label	L.type	C.L.Type	Order
1	0	1	DBLab	AGGR.	NODE	1
1	1	2	Member	AGG.R.	NODE	1
:	:	:	:	:	:	:
9	1	7	Project	ASSO.	NODE	3
9	4	8	Title	AGG.R	String	1

■Attribute

RootID	ElementID	Attribute_Name	Value[list]	Ordinal
1	2	Name	SungHee	1
1	2	Advisor	m1	2
:	:	:	:	:
9	7	ID	p1	1
9	7	Member	m1	2

■PathIndex

RootID	SourceID	MidID	TargetID	PathExpression
1	1	1	2	DBLab.Member
1	1	1	4	DBLab.Project
:	:	:	:	:
9	1	7	8	Project.Title

그림 3. edge table

5. 패스 표현의 처리

5.1 단순 패스 표현

ORDBMS에서 패스 traversal은 인덱스된 어트리뷰트를 이용해서 조인의 연속으로 처리할 수 있다. 또한 ORDBMS에서 순환적인 질의를 지원함으로써 패스 traversal을 처리시 RDBMS보다 더 높은 성능을 제공한다. 그러나 패스 traversal의 비용은 패스의 길이가 증가에 따라서 증가된다. 따라서 이 논문에서는 긴 패스 traversal을 최적하게 처리하기 위해 패스인덱스를 제안한다. 패스인덱스는 element테이블 사이에 조인 조건 (ChildID=ParentID)을 처리하기 위한 element 테이블의 조인 인덱스이다. 패스 인덱스를 사용할 경우 패스의 길이가 n개인 패스 표현의 처리가 가능하다. 즉, 패스가 3인 패스 표현을 처리할 경우에는 패스길이가 2인 패스인덱스 테이블과 element 테이블을 조인하여 처리한다. 패스의 길이가 길어지면 패스 인덱스 테이블 또한 증가되므로 이 논문에서는 패스의 길이를 2를 갖는 패스 인덱스 테이블을 이용한다. 다음은 이러한 패스인덱스를 만들기 위한 SQL명령이다.

```
select distinct e1.rootid,e1.childid,e2.parentid,e2.childid,e1.label,e2.label
from element e1, element e2 ,where e1.childid=e2.parentid;
```

DBLab이라는 루트로부터 시작되는 멤버의 이름을 검색하는 XML-QL 질의를 패스인덱스를 이용하여 패스 표현을 처리하는 질의의 예가 아래에 있다. 이 질의에서는 DBLab.Member.Name인 패스 길이가 3인 패스표현을 처리해야 한다. 질의를 처리하기 위해서 패스 인덱스 테이블에서 DBLab.Member 패스를 갖는 ChildID를 셀렉션하고 이 셀렉션된 튜플의 집합과 엘리먼트의 값을 저장한 Text테이블과 조인하여 패스 표현에 해당하는 값을 얻는다 . 패스인덱스를 사용하지 않을 경우에는 각각의 패스에 결합되는 ChildID를 찾기 위해 element 테이블을 3번의 조인과 3개의 중첩된 질의를 실행해야 한다. 따라서 패스인덱스를 사용하면 조인의 수가 감소된다.

```
select T1.value
from element e1, Text T1
where e1.label='Name' and e2.childID = T1.EID and
      e1.ParentID In (
            select p1.targetid
            from pathindex p1
            where p1.pathexpression='DBLab.Member');
```

6. 결론 및 향후 연구방향

이 논문에서는 ORDBMS에서 XML 문서를 저장하고 질의할 수 있는 XML 문서 처리시스템을 설계하였다. 즉, XML문서를 순서가 있는 labeled directed graph로 모델링하고 그래프 상에 존재하는 객체를 테이블로 저장하는 2가지 방법들을 제시하였다. 그 중에서 edge table 접근법을 이용하여 XML문서를 테이블로 매핑하는 상세한 방법을 제시했으며, 하나의 문서 내에서뿐만 아니라 다른 문서와의 링크관계를 저장하도록 하였다. 이러한 저장방법은 여러 가지 XML 문서에 대한 질의 언어 중 XML-QL을 지원할 수 있다.

semistructured 데이터에 대한 질의의 특징인 패스표현을 지원하기 위해서 패스인덱스 개념을 제시하였다. 패스인덱스를 이용할 경우 조인의 수 와 패스 표현 내에 포함된 계층적인 traversal을 감소할 수 있다. XML 문서의 루트로부터 시작되는 패스표현뿐만 아니라 임의의 객체로부터 시작되는 패스표현도 지원할 수 있다는 장점이 있다. 뿐만 아니라 이러한 시스템을 ORDBMS에서 구현함으로써 패스표현을 실행하기 필요한 계층적 질의를 시스템자체에서 지원한다는 장점이 있다.

향후 연구로는 XML-QL질의를 SQL로 변환을 위한 질의변환처리기의 구현과 이러한 질의에 대한 결과를 XML문서로 재구성하기 위한 XML문서 구성기를 개발하는 것이다.

참고문헌

[ABIT97] Serge Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, Janet L. Wiener: The Lorel Query Language for Semistructured Data. Int. J. on Digital Libraries 1(1): 68-88 (1997)

[FLOR99] Daniela Florescu, Donald Kossmann, "Storing and querying XMLData using an RDBMS", Bulletin of the Technical Committee on Data Engineering, Sep 1999 Vol.22 No.3.

[DEUT99a] Alin Deutsch, Mary F. Fernandez, Daniela Florescu, Alon Y. Levy, Dan Suciu: A Query Language for XML. WWW8 / Computer Networks 31(11-16): 1155-1169 (1999)

[DEUT99b] Alin Deutsch, M.Fernandex, D.Suciu, "Storing Semi-structured Data with STORED", Proceedings of ACM SIGMOD Conference, Philadelphia, Pennslyvania, May 1999.

[MCHU97] Jason McHugh, Serge Abiteboul, Roy Goldman, Dallan Quass, Jennifer Widom: Lore: A Database Management System for Semistructured Data. SIGMOD Record 26(3): 54-66 (1997)

[ORAC98] Oracle Corporation, XML Support in Oracle 8 and byeond, chinal white paper, http://www.oracle.com/xml/documents

[SHAN99] Jayavel Shanmugasundaram, Kristin Tufte, Chun Zhang, Gang He, David J. DeWitt, Jeffrey F. Naughton: Relational Databases for Querying XML Documents: Limitations and Opportunities. VLDB 1999: 302-314

[SHIM99] Takeyuki Shimura, Masatoshi Yoshikawa, Shunsuke Uemura: Storage and Retrieval of XML Documents Using Object-Relational Databases. DEXA 1999: