분산 환경에서 객체 관계 데이터베이스 시스템을 이용한 XML 문서 저장 시스템 구현 및 성능 비교*

정성룡[O] 홍의경
서울시립대학교 컴퓨터과학부
{kuanggaeto[O], ekhong}@venus.uos.ac.kr

Implementation and Performance Evaluation of XML Storage System using OR-DBMS in Distributed Environment

Seongryong Jeong[O] Eui Kyeong Hong
Dept. of Computer Science, University of Seoul

요 약

유비쿼터스 연구가 진행되면서 서로 데이터 모델이 맞지 않는 이기종 데이터베이스 간의 데이터 교환이 중요한 문제로 대두되었다. XML 문서는 데이터 호환성이 뛰어나 유비쿼터스 환경에 맞는 데이터 교환 수단으로 적합하다. 데이터베이스 시스템의 장점때문에 XML 문서를 객체 관계 데이터베이스에 저장하는 연구가 진행되었다. 객체 관계 데이터베이스 시스템에 XML 문서를 저장하고 질의하는 방법에는 여러 가지가 있다. 이 논문에서는 분산 환경에서 객체 관계 데이터베이스 시스템에 XML 문서를 저장하고 질의하는 방법을 논의하고, 성능 평가를 수행한다.

1. 서 론

XML[1]은 정보 공유 환경의 표준 매체로 자리잡아가고 있으며, 이기종 시스템간의 정보 교환을 용이하게 하는 특성을 가지고 있어 EC/EDI, 전자도서관, 전자 상거래 등 다양한 응용 분야에서 사용되고 있다. 기존의 데이터베이스들은 서로 데이터 모델과 시스템이 다를 경우에 데이터 교환이 상당히 어려웠다. 유비쿼터스 연구가 진행되면서 다른 데이터 모델이나 시스템간의 정보 교환이 중요시되고 있다. 각 개인의 의료 기록을 관리하는 집합적인 시스템이 있다면 정부의 주민등록 데이터베이스와 각 병원의 의료 데이터베이스 등 서로 다른 환경의 데이터베이스들이 상호 정보 공유를 하여야 한다. XML 문서는 이러한 환경에서 문서 교환의 표준으로 사용하기에 적합하다.

데이터베이스는 데이터를 관리하는 데에 있어 파일 시스템에 비해 뛰어나다. 이러한 장점들을 사용하여 XML 문서를 데이터베이스 시스템에 저장하는 기법이 연구되어 왔다. 그중 객체 관계 데이터베이스 시스템(ORDBMS)에 XML 문서를 저장하는 연구가 활발히 진행되고 있다[2].

이 논문에서는 분산 환경에서 객체 관계 데이터베이스 시스템에 XML 문서를 관리하는 방법에 대해 연구한다. 2절에서는 XML 문서를 저장 및 질의하는 관련 연구를 알아보고, 3절에서는 이 논문에서 제시한 시스템 설계에 대해 설명하며, 4절에서는 실험의 결과에 대해 분석하고 마지막으로 결론을 맺는다.

*본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

2. 관련 연구

2.1 XML 문서 저장

2.1.1 LOB

객체 관계 데이터베이스 시스템에서는 매우 큰 데이터를 저장하기 위해 LOB(Large Object)라는 데이터 형을 사용한다. LOB에는 문자열을 저장하는 CLOB(Character LOB)과 바이너리 데이터를 저장하기 위한 BLOB(Binary LOB)이 있다. XML 문서를 하나의 문자열로 생각한다면 객체 관계 데이터베이스 시스템의 CLOB을 이용하여 저장할 수 있다.

2.1.2 분할 저장 기법

분할 저장 기법[3]은 XML 문서의 내용을 엘리먼트 단위로 나누어서 저장하고 검색시 구조 정보를 참조하여 해당 요소 노드나 하위 엘리먼트들을 재구성하여 검색 결과를 반환하는 기법을 말한다. 분할 저장 기법은 XML 문서를 여러 개의 릴레이션을 이용하여 저장한다.

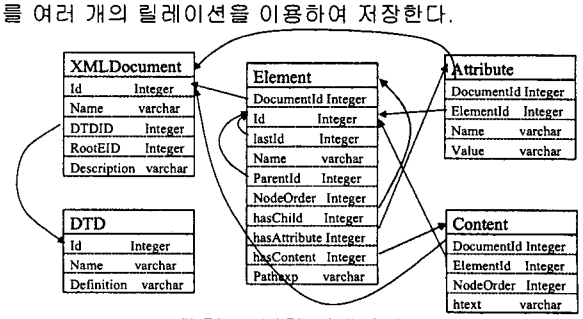

그림 1. 분할 저장방식

2.1.3 XML Type

오라클, MS SQL Server 등의 상용 데이터베이스에서는 XML을 지원하기 위해 XML Type[4]이라는 데이터 형을 만들어 기존 데이터 타입만으로는 지원하지 못하는 효율적인 기능을 지원하고 있다.

2.1.4 Native XML

Native XML 방식은 데이터베이스가 아닌 디렉토리에 XML 문서를 저장하며 운영체제의 디렉토리 관리 기법에 영향을 받는다.

2.2 분산 데이터베이스

현재 일부 상용 데이터베이스는 데이터베이스 링크, 분산 자원 관리 등의 분산 데이터베이스를 위한 기술들을 지원한다.

```
SELECT d.deptname as 부서명칭, e.empname as 사원명,
        e.salary as 연봉
FROM dept@eagle01 as d , emp@eagle02 as e
WHERE d.deopno=e.dept
        AND e.salary between 20000 and 30000;
```

2.3 XML 질의 처리

SQL 2003[5]에서는 XML 문서들에 대한 질의를 수행하기 위해 SQL/XML 문법을 지원하여 SQL 표현식 안에서 XPath, XQuery 등의 질의가 가능하다.

```
SELECT XMLQuery(
' for $p in
   doc(" /public/doc1.xml" )/doc1/body/project
Let $c := doc(" /public/costcencters.xml" )
              //costcenter[@costcenterno =
                    $p/@costcenterno]/@name
Where $p/@budget > 4000
Order by $p/@projectno
Return <project Name = " {$p/summary}"
        CC=" {$c}" />'
RETURNING CONTENT) FROM DUAL;
```

3. 미들웨어 시스템 설계

분산 데이터베이스에서 XML 문서를 저장 및 질의하기 위해 CLOB, XML Type, 분할 저장방식, Native XML 등 4가지 저장 방식에 따라 저장하고 질의할 수 있는 미들웨어를 설계하였다.

각 서버에 Native XML 저장 방식을 지원하는 객체 관계 데이터베이스를 이용하여 분산 데이터베이스를 구성한다. 논리적으로 하나로 통합된 저장 공간 위에 XML 문서를 저장하는 로더와, SQL/XML을 이용하여 XML 문서들에 질의를 수행하는 질의 처리기를 구성한다. 이에 대한 모델은 그림 2와 같다.

이를 위해 그림 3처럼 물리적으로 각 서버위에 미들웨어 시스템를 배치하여 저장 및 질의를 수행할 수 있도록 하였다.

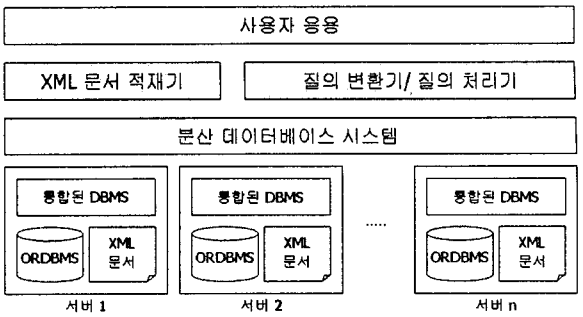

그림 2. 통합 아키텍처(논리적 모델)

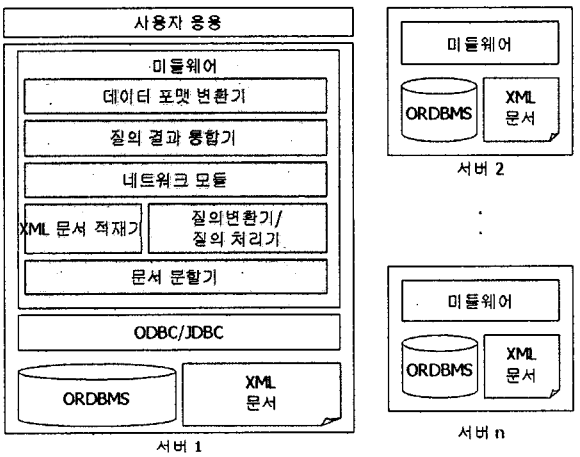

그림 3. 통합 아키텍처(물리적 모델)

미들웨어는 문서를 저장하는 요청이 들어왔을 때 문서를 각 서버에 저장될 조각으로 나누어 각 서버에 전달한다. 각 서버는 요청이 들어온 문서를 해당하는 방법으로 저장한다. 또한 문서에 대한 질의가 들어오면 미들웨어에서 질의를 해석하여 해당하는 서버에 나누어 전달하고 각 서버는 이 질의를 수행한다. 각 서버에서 얻은 결과를 합쳐서 사용자에게 질의 결과 전체를 반환하게 된다.

미들웨어는 두개의 통신 채널을 사용하는데 하나는 제어 채널로 사용하고 다른 하나는 데이터를 전송하는 채널로 사용한다. 이 채널을 통하여 각 노드의 미들웨어는 다른 노드의 미들웨어와 통신을 하며 질의를 수행하게 된다. 각 저장 방식마다 최적화를 위해 질의에 필요한 데이터들의 형태가 다르기 때문에 통신시 데이터 전송량도 다르게 된다.

미들웨어 시스템은 J2SDK 1.4버전을 이용하여 구성하였으며 데이터베이스는 오라클 10g를 사용하였다. 네트워크 환경은 최대 전송 속도가 10Mbps인 근거리 통신망(LAN) 환경에서 실험을 하였다. 분산 환경을 구성할 때 각 서버마다 저장 방식과 사용자 뷰가 서로 다르더라도 질의 수행이 가능하고 질의 결과를 처리하여 각 사용자에게 맞는 뷰를 통해 결과를 확인할 수 있도록 설계하였다.

4. 실험 및 결과

사전 정보를 가지고 있는 dictionary.xml 파일의 hw 엘리먼트의 첫 글자에 따라 각 문서들을 분할하여 각 서버에 분할된 XML 문서를 저장한다. dictionary.xml에 적용되는 DTD는 다음과 같다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a (#PCDATA)>
<!ELEMENT bib (#PCDATA)>
<!ELEMENT cr (#PCDATA)>
<!ELEMENT def (#PCDATA | cr)*>
<!ELEMENT dictionary (e+)>
<!ELEMENT e (hwg, vfl?, et?, ss)>
<!ATTLIST e
        id ID #IMPLIED
>
<!ELEMENT et (cr)+>
<!ELEMENT hw (#PCDATA)>
<!ELEMENT hwg (hw | pr? | pos?)+>
<!ELEMENT loc (#PCDATA)>
<!ELEMENT pos (#PCDATA)>
<!ELEMENT pr (#PCDATA)>
<!ELEMENT q (qd, a?, w, bib?, loc, qt)>
<!ELEMENT qd (#PCDATA)>
<!ELEMENT qp (q+)>
<!ELEMENT qt (#PCDATA | cr | i | b)*>
<!ELEMENT s (def, qp)>
<!ELEMENT ss (s+)>
<!ELEMENT vd (#PCDATA)>
<!ELEMENT vf (#PCDATA)>
<!ELEMENT vfl (vd* | vf+)+>
<!ELEMENT w (#PCDATA)>
<!ELEMENT i (#PCDATA)>
<!ELEMENT b (#PCDATA)>
```

문서를 저장하는 총 응답 시간은 각 서버에 저장하는 엘리먼트들을 설정하는 연산 시간, 각 서버에 엘리먼트의 집합을 전달하는 네트워크 전송 시간, 각 서버에서 할당된 엘리먼트들을 저장하는 시간의 합으로 표현할 수 있다.

각 엘리먼트를 저장할 위치를 결정하는 평균 시간 : T_e

전체 엘리먼트의 수 : N

서버 n에 저장되는 엘리먼트의 수 : N_n

서버에 엘리먼트 하나를 저장하는 평균 시간 : T_s

엘리먼트 하나를 전송하는 시간 : T_t

라고 한다면,

$$\text{전체 걸리는 시간} = T_e * N + \text{MAX}[T_s * N_1, T_t * N_2 + T_s * N_2, \cdots T_t * N_n + T_s * N_n]$$

CLOB, XML Type, 분할 저장 방식, Native XML 저장방식 등 4가지 저장 방식에 따라 단일 서버 저장, 완전 중복 저장(n=2, n=3), 분산 저장(n=2, n=3) 실험을 하여 각각의 성능을 비교 측정한다. 표 1은 10MB의 데이터를 저장할 때 저장 방식에 따른 저장 시간을 측정한 결과이다.

표 1. 저장 방식에 따른 저장 시간 (단위:sec)

	CLOB	XML Type	분할 저장 방식
단일 서버 저장	35	91	1203
완전 중복 저장 (n=2)	462	367	3131
분산 저장 (n=2)	293	241	1590
완전 중복 저장 (n=3)	468	350	3215
분산 저장 (n=3)	223	204	1219

단일 서버에 대한 저장이 아닌 경우 각 서버에서 저장되는 시간에 비해 네트워크 비용이 상대적으로 크므로 저장되는 시간의 영향은 줄어든다. n=3인 경우는 n=2인 경우에 비해 각 서버에 전송되는 데이터 양이 적으므로 동시에 데이터를 전송하는 경우 빠른 응답시간을 보이게 된다.

5. 결론 및 향후 연구 목표

유비쿼터스 환경에서 필요한 자유로운 데이터 교환을 위해서 데이터 모델로 XML을 사용할 수 있다. 이 실험에서 알수 있듯이 응용에 따라 저장 방식마다 저장, 검색 질의에 대한 성능에 차이가 있으므로 응용에 맞추어 XML문서를 어떤 데이터 타입으로 저장할지를 판단해야 한다.

향후에는 적은 CPU 연산으로 XML 문서를 저장할 수 있는 기술과 XML 전송시의 네트워크에 대한 부하를 줄일 수 있는 기술 그리고 전송 과정에서의 보안 문제에 대한 연구가 필요하다.

6. 참고문헌

[1] W3C, " Extensible Markup Language(XML) 1.0," http://www.w3c.org/XML, Feb.1998.

[2] D. Florescu and D. Kossman, "Storing and Querying XML Data using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.24-27, 1999.

[3] 고영기, 홍의경, " 분할 저장 시스템에 적합한 XPath 질의 처리기 설계," 한국정보과학회 가을학술대회 논문집 29(2), pp.52-54, 2002.

[4] " Oracle," http://www.oracle.com/technology/tech/xml/xmldb/index.html, May 2005.

[5] A. Eisenberg and J. Melton, " Advancements in SQL/XML," ACM SIGMOD Record 33(3), pp.79-86, Sept. 2004.