XML 문서 저장 시스템의 성능 평가*

박민경°, 홍의경
서울시립대학교 전산통계학과 데이터베이스 연구실
{island, ekhong}@venus.uos.ac.kr

Performance Evaluation of XML Document Storage System

Min Kyung Park°, Eui Kyeoung Hong
Department of Computer Science and Statistics, University of Seoul

요 약

최근 정보교환을 위한 표준으로 XML의 활용이 늘어나면서 XML 문서의 저장 및 검색에 관한 연구가 활발히 진행되어 왔다. 본 연구에서는 저장하부구조로서 객체관계 데이터베이스시스템(Odysseus)뿐만 아니라 RDBMS를 이용하여 XML 문서 저장 시스템의 성능을 평가한다. XML 문서가 특정 DTD(Document Type Definition)를 따르게 되면 XML 문서를 파싱하고 그에 맞는 스키마를 설계해야 하는 단점이 있지만 DTD와 결합된 XML 문서는 XML-QL과 같은 질의 언어를 사용하여 훨씬 다양한 질의를 수행할 수 있게 한다. 따라서 DTD 의존적인 XML 문서 저장 시스템을 특정 자료를 통해 설계해 볼 필요가 있다. 여기서는 증권 정보를 이용해서 저장 시스템을 설계하고 대용량 데이터의 검색 시간을 측정함으로써 이 시스템의 성능을 평가한다. 또한 시스템간의 성능을 비교하고 성능 개선을 위한 방법을 제시한다.

1. 서론

XML(eXtensible Markup Language)이 인터넷 상의 정보를 효과적으로 표현하고 교환할 수 있는 표준이 되고 있다. 또한 반구조적인 데이터를 저장할 수 있는 표준 방법으로 인식되고 있다[1]. 인터넷 Web 문서뿐만 아니라 전자도서관, 전자 상거래, EC/EDI를 포함한 다양한 분야에서 XML을 활용하고자 폭넓은 연구를 하고 있으며, 이러한 XML 문서들을 효과적으로 저장하고 검색할 수 있는 XML 문서 저장 검색 시스템이 개발되어 왔다. 따라서 기존의 시스템들을 비교 평가하고 좀더 나은 XML 문서 저장 시스템을 연구할 필요가 있다.

본 논문에서는 저장하부구조로 객체관계 데이터베이스시스템뿐만 아니라 RDBMS를 이용하고, 기존의 대부분의 시스템들이 DTD 독립적이었기 때문에 특정 응용프로그램에서 사용될 수 있는 DTD와 그에 따른 스키마를 설계하는데, 본 논문에서는 증권정보를 이용하여 DTD를 설계한다.

시스템의 성공여부는 질의를 얼마나 잘 처리하는가에 달려있다. 따라서 하부 저장시스템의 성능이 질의 처리 효율성에 중요한 요소가 되므로 XML 문서를 저장하는 최상의 방법을 연구하는 것은 중요한 문제이다[8]. 이미 수많은 XML 문서 저장 모델 기법[2]과 XML 문서를 저장할 수 있는 여러 가지 방법들이 제안되어 왔다[4][6].

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

XML 저장 기법은 크게 세 가지로 분리할 수 있는데 화일시스템, 데이터베이스 시스템, 저장 관리자가 그것이다. 다음 절에서 이들 방법에 대해 간략하게 살펴볼 것이다.

2. 관련 연구

근래 10년 동안, 저장 모델과 저장 시스템에 관한 연구가 활발히 이루어져 왔다[2]. 하부저장시스템(예를 들어, 화일시스템 혹은 DBMS)을 무엇을 쓰는가부터 어떤 저장 방식(예를 들어, 분할 혹은 가상분할 방식)을 따르는가까지 그 방법은 다양하다.

XML 문서를 저장하는 가장 단순한 방법으로는, 별개의 운영체제 화일로 각각의 XML 문서를 저장하고, 질의에 의해 XML 문서가 액세스될 때마다 DOM 파서를 이용하여 문서를 파싱하는 것이다. 이 방법은 구현하기 쉬울 뿐만 아니라 XML 데이터의 용량이 적은 경우에 성능이 뛰어나지만, XML 문서가 액세스될 때마다 매번 파싱을 해서 메모리로 적재해야 하기 때문에 오버헤드가 발생한다. 또 다른 방법은 DBMS를 이용하는데, XML 데이터가 여러 개의 테이블에 나뉘어서 저장된다. 이 경우 XML 질의어(예를 들어 XML-QL)를 SQL로 변환해야만 한다. 마지막으로 SM(Storage Manager)기법인데, 이 방법은 XML 질의가 먼저 파싱되고, 적당한 질의 트리로 변환되고 최적화 된 이후에, 별도의 XML 질의 엔진에 의해 질의가 수행되는 방식이다[9]. 이 방법은 별도의 XML 질의 엔진을 구현해야만 한다.

그러나 최근의 연구들을 살펴보면, 저장하부구조로 DBMS를 이용하고 있다[4][6]. [6]은 XML 데이터를 관계형 데이터베이스에 저장하는 방법을 실험하였다. 이 연구는 성능 평가를 위해, 실제 XML 데이터에 대해 질의를 수행하는 동안의 응답 시간이 아니라 성능 기준으로서 조인(join)연산 시간을 이용했다. [4]는 관계형 데이터베이스에 XML 문서를 저장할 수 있는 여러 가지 매핑방법을 개발하고 성능 평가를 하였다. [4]의 방법을 좀더 확장한 연구가 [8]이다. 이 논문에서는 객체 기법을 개발하고 성능을 평가하는데 단지 Shore 객체 관리자의 범위 안에서만 그 성능을 평가하지만 관계형 데이터베이스 시스템으로 바로 확장이 가능하도록 하였다. 또한 B-트리를 이용한 인덱싱 기법까지 그 범위를 확장했는데 DBMS를 이용하면 자체에서 제공하는 B-트리나 R-트리 인덱스를 사용할 수 있다.

3. XML 문서 저장

본 절에서는 객체관계 데이터베이스의 특징을 사용하여 XML 문서를 저장한 [9]의 모델을 이용한다. 이 모델은 DTD에 관계없이 XML 문서를 저장할 수 있도록 DTD 독립적이며, 문서의 빈번한 수정이 있을 것으로 가정하여 문서를 엘리먼트 단위로 쪼개어 저장하는 '분할저장' 방식을 사용하였다. 그리고 XML 문서의 요소들 중 가장 기본적인 XML 문서, DTD, 엘리먼트, Content, 애트리뷰트 정보들에 관해서만 저장하는 것으로 가정한다. DTD 독립적인 문서와 그렇지 않은 경우의 성능 평가를 위해 이 모델의 기본적인 특징을 이용하여 DTD 의존적인 모델을 설계하는데 이 DTD는 증권정보를 따르도록 했다. <그림 1>이 증권정보를 표현하는 DTD이며, <그림 2>가 이 DTD를 따르는 예제 XML 문서이다.

```
<!ELEMENT Stockdata (#PCDATA)>
<!ELEMENT STOCK (기업명, 코드, 일자, 종가, 시가, 고가, 저가,
 전일대비, 거래량, 거래대금)>
<!ELEMENT 기업명 (Stockdata)>
<!ELEMENT 코드 (Stockdata)>
<!ELEMENT 일자 (Stockdata)>
<!ELEMENT 종가 (Stockdata)>
<!ELEMENT 시가 (Stockdata)>
<!ELEMENT 고가 (Stockdata)>
<!ELEMENT 저가 (Stockdata)>
<!ELEMENT 전일대비 (Stockdata)>
<!ELEMENT 거래량 (Stockdata)>
<!ELEMENT 거래대금 (Stockdata)>
```

<그림 1> 증권 DTD

XML문서의 저장은 다음과 같은 과정을 거치게 된다. 일단 XML 문서는 파서에 의하여 파싱되며, 이는 DOM(Document Object Model) 트리의 형태로 결과가 얻어진다. 파싱의 결과로 얻어진 DOM트리는 저장관리기에 의하여 테이블에 매핑되어 저장된다[10].

저장관리기는 문서의 루트노드부터 DFS방식으로 노드를 순회하며 각 노드의 정보는 해당 테이블에 저장되는데, 노드의 타입이 엘리먼트이면 Element 테이블에 엘리먼트명과 부모엘리먼트의 OID, 경로 등의 정보가 저장되며 애트리뷰트가 있는지를 체크하여 애트리뷰트는 Attribute 테이블에 애트리뷰트가 나열된 순서와 함께 애트리뷰트의 이름과 값을 저장한다. 이러한 과정을 자식노드를 재귀적으로 순회하면서 모든 엘리먼트를 데이터베이스에 저장한다.

XML 문서를 검색할 때, 사용자의 질의요구는 질의 생성기에 의하여 SQL로 변환되어 질의가 수행되며, 질의의 결과는 결과

```
<?xml version="1.0" encoding="euc-kr" ?>
<STOCK>
  <기업명>
    <Stockdata>samyang</Stockdata>
  </기업명>
  <코드>
    <Stockdata>080</Stockdata>
  </코드>
  <일자>
    <Stockdata>2001.03.30</Stockdata>
  </일자>
  <종가>
    <Stockdata>10800</Stockdata>
  </종가>
  <시가>
    <Stockdata>10800</Stockdata>
  </시가>
  <고가>
    <Stockdata>11000</Stockdata>
  </고가>
  <저가>
    <Stockdata>10700</Stockdata>
  </저가>
  <전일대비>
    <Stockdata>-100</Stockdata>
  </전일대비>
  <거래량>
    <Stockdata>10040</Stockdata>
  </거래량>
  <거래대금>
    <Stockdata>10862</Stockdata>
  </거래대금>
</STOCK>
```

<그림 2> 증권 DTD를 따르는 XML 문서

집합의 형태로 질의결과 생성기에 넘겨지고 여기서 XML형태로 결과를 재구성하여 사용자에게 보여지게 된다.

4. 성능 평가

앞선 연구들의 성능 평가를 살펴보면, 화일시스템이나 저장관리자 방식에 비해 DBMS를 이용하는 경우 성능이 가장 나쁜 것으로 나타난다[8]. 그럼에도 본 논문에서 굳이 저장하부시스템으로 DBMS를 이용하는 이유는 웹 상의 수많은 데이터가 DBMS(특히 RDBMS)에 저장되어 있고, 따라서 XML 문서를 데이터베이스에 저장하는 경우 데이터베이스의 우수한 성능을 이용할 수 있고, 기존의 응용시스템의 데이터를 함께 사용할 수 있는 장점이 있기 때문이다.

본 절에서는 앞서 제시한 XML 저장시스템 하에서 DTD 독립적인 XML 문서와 의존적인 문서간의 성능 차이, 저장하부시스템을 RDBMS와 ORDBMS로 했을 때의 차이, 그리고 3절에서 제시한 저장 시스템과 기존의 저장 시스템과의 성능을 비교 평가해 보고자 한다.

그러나 실제 저장 시스템의 성능을 평가한 연구가 많지 않아 절대적 평가는 불가능하다. 본 논문에서는 [8]의 Edge기법, [3]의 저장 시스템과 우리 시스템의 성능을 비교 평가한다.

실험에 사용된 XML 문서는 셰익스피어 희곡 37편(7.65M)과 증권정보(10M), DBLP database(60M)를 이용하였다. 저장 수행 시간은 클라이언트에서 데이터베이스 서버에 접속하고, 해당 화일을 데이터베이스 서버에 전송하는 시간과 파싱되고 테이블에 저장되기까지의 시간을 모두 포함한다. 검색 수행 시간은 서버에 접속하고, 질의 요구를 SQL로 변환하는 시간과 실제 질의를 수행하여 얻어진 결과를 화면에 보여주는데 걸리는 시간을 모두 포함한 것이다. 비교평가를 위한 RDBMS는 오라클, ORDBMS는 오디세우스를 이용하였다.

<표 1>의 결과가 보여주듯이, DTD 독립적인 XML 문서보다 DTD 의존적인 XML 문서의 저장 시간이 좀더 길다. DTD

를 따르는 문서의 경우, 해당 XML 문서가 타당한지를 체크해야 하므로 파싱을 하는데 더 많은 시간이 소요된다.

<표 2>검색의 경우, DTD 의존인 경우에 응답 시간이 더 짧음을 알 수 있다. DTD에 따라 스키마가 고정되므로, 훨씬 다양한 질의를 수행할 수 있기 때문이다.

	DTD 독립	DTD 의존
10K	1.23s	1.87s
100K	8.64s	10.56s
200K	21.32s	26.12s
300K	31.91s	39.02s

<표 1> DTD 유무에 따른 저장 시간

	DTD 독립	DTD 의존
SQ_1	0.42s	0.21s
SQ_2	0.13s	0.11s
SQ_3	1.2s	0.97s
SQ_4	1.79s	1.08s

<표 2> DTD 유무에 따른 검색 시간

<표 3>의 결과를 살펴보면, 하부저장시스템이 RDBMS인 경우와 ORDBMS인 경우 질의 처리 시간의 거의 비슷하다는 것을 알 수 있다.

<표 4>의 System1은 3절에서 제안한 우리 모델이고 System2는 [8]에서 제안된 Edge 기법이다. 우리의 시스템은 ORDBMS를 이용하여 테스트되었고, [8]을 제외한 다른 시스템의 경우 검색 시간만을 보여주고 있어 저장 시간을 비교할 수가 없었다.

	RDBMS	ORDBMS
Query_1	0.17s	0.21s
Query_2	0.13s	0.11s
Query_3	1.35s	1.28s
Query_4	2.58s	2.49s

<표 3> 하부시스템에 따른 성능 결과

	System1	System2
420KB	51.23s	82s

<표 4> 저장 시간 비교

[7]에서 셰익스피어 희곡을 이용한 10개의 질의를 제안했다. 이 질의를 우리의 시스템에도 적용하여 질의 응답 시간을 측정했다. 이 시스템을 SYU라 하고 DataChannel에 의해 구현된 XQL 모듈[3]은 DM99라 한다. Odysseus는 우리의 시스템을 가르킨다. <표 5>에서 볼 수 있듯이 3절에서 제안한 우리의 시스템이 조금 더 좋은 성능을 보이고 있음을 알 수 있다.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Odysseus	0.12	0.15	0.14	0.14	0.16	0.26	0.28	0.34	1.01	0.95
SYU/Postgres	0.15	0.15	0.16	0.18	0.19	0.34	0.35	0.37	1.30	1.04
DM99	0.01	0.01	0.01	0.01	0.01	6.52	3.15	6.38	7.04	8.96

<표 5> 질의 집합에 대한 응답 시간

5. 결론 및 향후 연구 방향

본 연구에서는 XML 문서를 효과적으로 저장하고 검색할 수 있는 XML 문서 저장 시스템을 설계하고 증권정보를 이용하여 다양한 성능 평가를 해 보았다. DTD 독립적인 모델과 DTD 의존적인 모델을 각각 설계하고 하부저장시스템을 달리하여 질의 처리 시간을 측정했으며 다른 XML 문서 저장 시스템과의 성능 비교 평가를 수행했다.

향후에는 데이터베이스에 저장된 다른 데이터와 XML 데이터를 통합할 수 있는 방법에 관해 알아보며, 웹 기반의 정보 시스템에서 찾아볼 수 있는 다중 질의를 효율적으로 처리할 수 있는 방법과 병렬처리에 대해 연구할 예정이다.

6. 참고 문헌

[1] T.B. Ray, J. Paoil, C.M.S Perberg-McQueen, Extensible Markup Language (XML) 1.0, http://www.w3.org/TR/REC-xml, 1998.

[2] G. Copeland and S. Khoshafian, "A Decomposition Storage Model," Proc. of the ACM SIGMOD Conf., pages 268-279, Austin, TX, May 1985.

[3] DataChannel and Microsoft. DataChannel-Microsoft Java XML Parser(Beta2) 1.0., http://www.datachannel.com/xml_resources/developers/, February 1999.

[4] D. Florescu and D. Kossmann, "A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database," Rapport de Recherche No. 3680 INRIA, Rocquencourt, France, May 1999.

[5] A. R. Schmidt, M. L. Kersten, M. A. Windhouwer, F. Waas, "Efficient Relational Storage and Retrieval of XML Documents," Workshop on the Web and Databases (WebDB), Dallas, TX, May, 2000.

[6] J. Shanmugasundaram, K. Tufte, C. Zhang, G. He, D. J. DeWitt, and J. F. Naughton, "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. of 25th Int'l Conf. on VLDB, Edinburgh, Scotland, UK, pp.302-314, 1999.

[7] T. Shimura, M. Yoshikawa, and S. Uemura, "Storage and Retrieval of XML Documents Using Object-Relational Databases," DEXA99, pp.206-217, 1999.

[8] F. Tian, D. J. DeWitt, J. Chen, and C. Zhang, "The Design and Performance Evaluation of Alternative XML Storage Strategies," (submitted for publication), 2000.

[9] 김훈, "객체관계 데이터베이스를 이용한 XML 문서 저장 모델 설계," 석사학위논문, 서울시립대학교 전산통계학과, 2001년 2월.

[10] 한상웅, "ORDBMS를 이용한 XML 문서 저장 시스템의 설계 및 구현," 석사학위논문, 서울시립대학교 전산통계학과, 2001년 2월.