대용량 이종 XML 데이터 검색을 위한 RDBMS기반 인덱싱 기법

이성진○ 박명순
고려대학교 컴퓨터공학과
{jikidad○, myongsp}@ korea.ac.kr

Technology for Searching Massive XML Data with Different Schema

Sung-Jin Lee○, Myong-Soon Park
Dept. of Computer Science and Engineering, Korea University

요 약

최근 XML은 기업간 데이터 교환의 표준으로 자리잡았다. 기업간 데이터 교환은 필연적으로 대량의 XML형태의 데이터가 로그 형태로 보전되게 된다. W3C에서는 XML자료의 검색을 위하여 XQuery1.0을 발표하고 XML 검색문의 표준화를 제시하였다. 검색문과는 별도로 XML데이터의 저장 모델에 대한 연구 또한 활발히 진행되고 있다. 본 논문에서는 대용량 XML데이터를 RDBMS를 이용하여 저장하고 빠른 검색을 지원 할 수 있는 역 인덱싱 방안을 기초로 검색 성능을 향상 시킬 수 있는 방안에 대하여 연구하고 기존 방법과의 비교 실험을 통해 그 효과를 검증하였다.

1. 서 론

XML(eXtensible Markup Language)은 확장 가능한 마크 업 언어로서 전달하고자 하는 데이터뿐 아니라 자료에 대한 구조 정보까지 전달이 가능하여 최근 기업간 비즈니스 시스템을 중심으로 데이터교환 방식의 표준으로 자리 잡았다.[1] XML문서의 구조적인 편리함으로 인하여 많은 비즈니스 분야에서 다양한 형식의 XML문서가 등장하게 되었고 저장된 자료의 크기 역시 기하급수 적으로 늘어나고 있는 추세이다. 이 때문에 자료 검색의 복잡성과 속도 저하의 문제가 대두 되었고 이러한 문제 해결을 위해 W3C에서는 다양한 XML문서를 표준적으로 검색할 수 있도록 XPath[2] 나 XQuery[3]와 같은 XML검색 문을 발표 하였고 IBM[4], Oracle[5]등 세계적인 대형 IT 회사에서는 XQuery1.0 요구를 충족하는 Parser를 개발하여 경쟁적으로 발표하고 있다. 사용자들은 이를 통하여 마치 RDBMS의 저장된 자료를 SQL문장을 통하여 검색하듯이 복잡한 XML데이터를 원하는 조건에 맞추어 다양하게 검색할 수 있게 되었지만 파일 방식으로 저장된 XML데이터의 검색은 순차 검색을 적용하게 될 수 밖에 없어 대용량 XML 데이터의 경우 그 처리 속도의 문제는 계속적으로 남게 된다. 이를 위하여 검색방법과 별도로 다양한 형식의 XML 데이터를 RDBMS를 이용하여 효율적으로 저장하기 위한 인덱싱 방안이 연구되고 있다.

본 논문에서는 다양한 형식 구조를 갖는 대용량 이종 XML데이터를 RDBMS를 이용하여 빠른 검색을 보장 하도록 하는 역 인덱싱(Inversed Indexing) 저장 방안에 대하여 연구하고 이를 기초로 좀 더 검색 속도 측면에서 효율적인 저장 방안을 제안 하였으며 실험을 통해 본 제안이 효율적인 방안임을 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 연관된 관련연구에 대해 알아보고, 3장에서는 개선된 방안을 제안하며, 4장에서는 실험을 통해 기존 방안과의 효율을 검증하고, 5장은 결론 및 향후 연구방향에 대하여 기술한다.

2. 관련 연구

XML데이터를 RDBMS를 통하여 저장하여 빠르게 검색하려는 인덱싱 관련 연구는 꾸준하게 계속되어왔다.[6] [7] RDBMS를 이용한 인덱싱 방법으로 초기에는 고정된 XML구조를 XML 엘리먼트 Node정보를 이용하는 방식이 연구 되었지만 최근에는 다양한 XML구조 위주의 Path중심 인덱싱 방식[6] [7] 이 주로 연구되고 있다.

2.1 XRel 방식

XRel방식은 경로 문자열을 기반으로 경로 검색을 지원하도록 하는 방식이다. 예를 들면 " <경로1><경로2><경로3>참조값</경로3></경로2></경로1>" 형식의 XML 문서의 경우 발생되는 경로 문자열은 " #/경로1" ," #/경로1/경로2" ," #/경로1/경로2/경로3" 이러한 식으로 3개의 경로 정보가 Path테이블에 저장되며 Path테이블은 XML문서들에 대한 구조 정보를 나타내게 된다. 이와 별도로 해당 경로에 해당하는 발생 정보를 위해 Element테이블에는 경로가 발생된 문서 번호, 경로의 마지막 위치에 나타나는 엘리먼트의 시작 위치 번호와 종료 위치 번호, 형제 엘리먼트들과의 순서 등으로 구성되며, Attribute 테이블은 경로가 발생된 문서 번호, 경로의 마지막

위치에 나타나는 애트리뷰트의 시작 위치 번호와 종료 위치 번호, 그리고 애트리뷰트의 값으로 구성된다. 추가로 Element테이블의 발생 경로들이 포함하는 값들을 Text라는 테이블에 따로 관리하였다. 이 방식의 경우 Path테이블을 통하여 문서의 증가와 관계없이 해당하는 경로가 존재한다면 위의 Edge방식에 비하여 빠른 검색이 가능하다. 그러나 이 경우 상이한 문서가 증가함에 따라 Path테이블이 계속적으로 증가하게 되고 부모-자식 관계를 포함한 경로를 찾는 경우는 Path테이블 내의 모든 경로를 비교하여야만 하는 단점이 있다.[6]

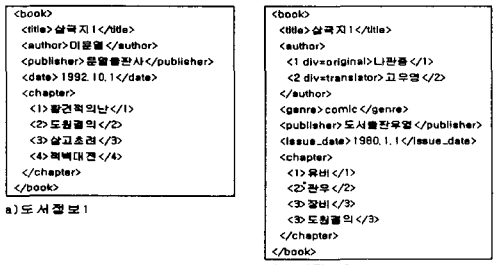
2.2 Inverted Indexing 방식

Inverted Indexing 방식은 XRel방식과 같이 경로 위주의 검색을 지원하지만 상이한 문서가 급격히 증가하여도 안정적인 검색 성능을 보장한다. 이 방식에서는 문서에서 사용된 모든 용어를 저장하기 위해 Term테이블을 정의하고 Term테이블은 발생된 Term값과 Term식별 번호로 구성된다. TermOccurrence테이블에서는 Term테이블의 용어들에 대한 발생 위치 정보를 저장하게 되는데 구성은 Term식별 번호와 문서 식별 번호, 경로 식별 번호, Term 발생 위치 번호로 구성된다. UniqPathElement테이블은 전체 XML문서들로부터 추출된 서로 다른 경로의 정보를 저장하게 된다. 이 테이블은 전체 XML문서들의 구조적인 요약 정보이며 저장되는 정보는 엘리먼트의 이름과 엘리먼트가 속해 있던 경로의 식별자, 엘리먼트가 경로에 나타난 깊이가 된다. PathOccurrence테이블은 모든 발생 경로에 대한 정보를 담은 테이블로서, UniqPathElement 테이블에 분할되어 저장된 경로들의 경로 식별자, 경로가 있는 문서 식별자, 경로의 시작과 UniqPathElement테이블을 통해 문서의 경로 존재 여부를 먼저 파악하게 되어 질의 비용을 절약하게 되고 또한 상이한 문서가 늘어나더라도 찾는 경로에 포함된 엘리먼트 정보들만 추출하여 비교 연산을 수행하므로, 경로의 유무를 빠르게 알아낼 수 있어 타 방식에 비하여 높은 검색 성능을 나타낸다. [7]

3. 제안 방식

Inverted Indexing방식은 상이한 문서인 경우도 다른 인덱싱 방안에 비하여 안정적이고 빠른 검색 결과를 나타낸다.[7] 그러나 Inverted Indexing 방식에서는 절대 경로를 가지고 조회하는 경우도 상대 경로로 조회하는 경우와 똑 같은 질의 비용을 사용하게 된다. 그 이유는 Inverted Indexing 방식에서는 모든 엘리먼트가 개별적으로 분리되어 UniqPathElement테이블에 저장되고 추가적으로 엘리먼트가 위치한 트리의 깊이 정보를 기초로 검색을 하기 때문이다. 본 논문에서는 Inverted Indexing방법과 XRel방식에서 사용하는 Path테이블을 추가하여 절대 경로를 조건으로 하는 검색인 경우 매우 효율성을 높일 수 있는 방안을 제안하였다.

본 논문에서 사용할 XML문서모델은 N개의 상이한 구조를 갖는 대량의 XML문서이다. 논의를 간단히 하기 위하여 저장되는 문서는 Non-DTD기반으로 가정을 한다. 그림 1은 2개의 XML문서의 예이다. 그림1의 a), b)두 문서는 모두 같은 도서에 대한 정보를 나타내는 XML문서지만 각각의 문서 구조는 상이하다.



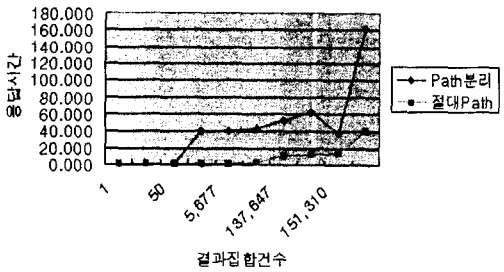
[그림1] XML문서의 예

XML문서로부터 저장 해야 할 정보를 추출하기 위하여 XML문서를 DOM트리 구조로 변경하여 각 노드의 발생 순번과 값 그리고 트리의 깊이 등을 기초로 정보를 만들게 된다. 이렇게 생성된 XML문서 정보를 Inversed Indexing모델링에 의해 4개의 테이블에 저장하게 된다.[7] 애트리뷰트의 표현은 애트리뷰트를 포함하는 엘리먼트의 자식 엘리먼트와 동일하게 표현하게 된다. Inverted Indexing방식에서는 그림1의 " book/chapter/1" 의 값 " 황건적의난" 을 검색하기 위해서는UniqPathElement테이블을 3번 Self-Join해야 한다. 이유는 이미 설명한 바와 같이 UniqPathElement테이블을 이용하여 각각의 엘리먼트 값과 트리 깊이 정보를 같이 찾아 보아야 하기 때문이다. 본 논문에서는 이러한 Inverted Indexing방식을 기초로 하여 XRel방식에서 사용하는 Path테이블을 절대 경로 검색을 위해 추가하도록 하였다. 추가된 AbstractPath테이블에는 XML문서가 가지고 있는 절대 경로 정보를 저장하게 되고 이를 통하여 일반적인 검색 조건의 대다수를 차지하는 절대 경로를 통한 검색에 있어 대폭적인 성능 향상을 기대할 수 있다. 예를 들면 " /경로1/경로2/경로3" 로 표현되는 경로에 대하여 UniqPathElement 테이블에 " /경로1" ," /경로2" ," /경로3" 의 3개의 엘리먼트 정보 가 저장되는 것과 별개로 " /경로1" ," /경로1/경로2" ," /경로1/경로2/경로3" 와 같이 발생할 수 있는 모든 절대 경로 3개를 추가적으로 AbstractPath테이블에 저장한다. 이렇게 함으로써 절대 경로를 조건으로 한 검색인 경우는 엘리먼트 개수만큼 조인을 해야 하는 UniqPathElement 테이블을 사용하지 않고 AbstractPath테이블을 이용하여 단 한번의 검색을

통해 자료 유무를 찾을 수 있게 된다. 본 논문에서 제시한 방식은 기존의 Inverted Indexing 방식에 비하여 AbstractPath테이블 공간만큼의 추가적인 저장 공간을 요구한다. 그러나 추가되는 저장 공간은 추가로 저장 대상이 되는 XML문서가 기존에 등록된 XML형식과 다른 형식을 가질 때만 추가 된다는 점에서 그 증가 폭은 극히 적다. 물론 발생하는 모든 XML문서 마다 모두 각기 다른 형식을 갖는다고 전제 한다면 저장 용량이 문제가 되겠지만 이런 경우는 현실적으로 발생 가능성이 없기 때문에 본 논문에서는 저장 공간에 대한 문제는 고려하지 않았다.

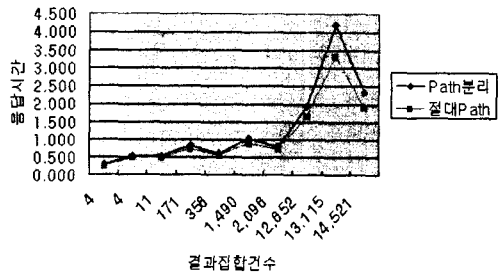
4. 성능실험

본 제안의 실험은 테스트 시스템으로 CPU Pentium 4 2.0G 1개, M/M 256Gbyte, RDBMS는 MySQL 4.0.9으로 하였고, 사용된 XML 데이터는 국내 보험회사에서 사용된 방카슈랑스 업무용 대외 전송용 자료 92,000건 310 Mbyte이며, 프로그램은 JDK1.3을 이용하여 Java로 코딩하였다. 검색 조건은 절대 경로를 조건으로 값을 찾는 경우와 절대 경로와 값을 조건으로 해당 XML문서 자체를 찾는 2가지 형태로 실험을 실시하였으며 종류별로 10개의 서로 다른 조건을 사용하였다. 다른 검색 유형은 기존의 Inverted Indexing방식과 동일함으로 배제하였다.



[그림1]절대경로로 값을 검색하는 경우 결과건수별 성능비교

절대 경로를 이용하여 값을 검색하는 경우 그림 2에서 확인 할 수 있듯이 결과 집합의 건수가 많을수록 AbstractPath테이블을 이용하는 경우 성능이 매우 향상되어 평균적으로는 80.3%의 성능이 향상되었다.



[그림2]절대경로와 값을 조건으로 검색하는 경우 결과건수별 비교표

또한 그림3에서와 같이 절대 경로와 값을 조건으로 XML 문서를 찾는 경우는 평균 15.6%의 성능 향상을 보였는데 두 가지 실험 결과가 큰 차이를 보이는 이유는 첫번째 실험의 검색 조건이 절대 경로 한 가지인 반면, 두 번째 실험의 경우는 절대 경로와 값을 사용하여 두 가지 조건으로 검색했기 때문에 결과 집합의 건수가 상대적으로 적게 발생한 것이 이유로 판단된다. 실질적인 가정을 위하여 두 가지 실험에서 결과 집합 건수가 100건 이하인 경우만을 비교했을 때에는 첫 번째 실험의 경우는 17.4%, 두 번째 실험의 경우는 5.8%의 성능 향상을 보였다.

5. 결론 및 향후 연구과제

현실 비즈니스에서 활용되는 XML문서는 점차 대용량화 되어가고 있으며 문서의 형식도 매우 복잡해지고 있다. 이로 인하여 XML데이터에 대한 검색과 저장에 대한 연구는 중요한 연구 과제로 대두되고 있다. 본 논문에서는 대용량 데이터를 이용한 실험을 통하여 실제 검색에서 가장 많이 사용되고 있는 절대 경로를 조건으로 하는 검색의 경우에 기존 방식에 비하여 매우 효과적인 방안을 제안하였다. 향후 또 다른 검색 조건에 대한 추가적인 개선 방안과 데이터의 저장 공간을 절약할 수 있는 방안을 계속적으로 연구 할 것이다.

6. 참고문헌

[1] Mark Birbeck외, Professional XML 2nd Edition, Wrox Press, 2002.

[2] XML Path Language(XPath) Version1.0 W3C Recommendation,http://www.w3.org/TR/xpath,Nov.1999.

[3] XQuery 1.0 : An XML Query Language W3C Working Draft, http://www.w3.org/TR/xquery, 2002.

[4] JavaCC : parse trees, and the XQuery grammar, http://www106.ibm.com/developerworks/xml/library/x-javacc1, IBM, 2003.

[5] OJXQI: Oracle XQuery Prototype , http://otn.oracle.com/sample_code/tech/xml/xmldb/xmldb_xquery download.html, Oracle, February 2003.

[6] Masatoshi Yoshikawa and Toshiyuki Amagasa, XRel : A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases, ACM TOIT, Volume 1, Number 1, pp 110-141, August 2001.

[7] Kyung-Sub Min and Hyoung-Joo Kim, An RDBMS-based Inverted Index Technique for Path Queries Processing on XML Documents with Different Structures , KISS Database, Volume 30, Issue 4, August 2003.