역 색인을 이용한 경로 질의 기반 대용량 XML문서 검색

문경원, 황병연
가톨릭대학교 컴퓨터공학과
e-mail : {kwmoon, byhwang}@catholic.ac.kr

Retrieval of Large scaled XML Documents based on Path Query using Inverted indexes

Kyung-won Moon, Byung-yeon Hwang
Dept. of Computer Engineering, The Catholic University of Korea

요 약

1998년 XML 문서 표준이 제안된 이래, 다양한 응용 분야에서 XML은 데이터를 표현하는 표준으로 자리잡아 가고 있다. 특히, 인터넷상의 많은 데이터들이 XML 형태로 작성되고 변환됨에 따라 다량의 XML 데이터가 생성되고 있다. 따라서 현재 XML 문서의 저장 및 질의 처리 기법의 연구가 활발하게 진행되고 있다. 하지만 기존의 연구는 대용량 XML 문서를 다루기에는 미흡한 점이 있다. 본 논문에서는 인터넷상의 널리 퍼져있는 방대하고, 다양한 구조의 XML문서들을 대상으로 패스 기반 질의를 빠르게 처리할 수 있는 검색 기법을 제안한다. 제안된 기법은 인터넷상에 산재해 있는 여러 XML 문서를 관계형 데이터베이스에 효율적으로 저장하고 질의를 통해 인터넷상 XML 문서의 엘리먼트를 빠르게 검색하는데 주안점을 둔다. 먼저, XML 문서를 관계형 데이터베이스에 효율적으로 저장하는 계층형 XML 저장 기법을 제안하고, 정보 검색 시스템에서 많이 사용하는 역 인덱스를 사용하여 저장된 XML 문서에 대한 검색 성능을 향상시킨다.

1. 서론

최근 XML 문서의 다양한 검색기법에 관한 연구가 진행되고 있다. 기존의 관계형 데이터베이스의 고정된 스키마가 가지는 단점을 극복하기 위해 XML 문서 검색은 반구조화(semi-structure)[1]된 데이터 구조를 통해 정해진 스키마가 없이도 다양한 형태의 XML 문서를 다루는데 적합하게 설계되고 연구되어 왔다. 하지만, 대부분의 XML 질의 처리 연구는 정해진 문서 형식을 가지는 한정된 수의 문서들만을 고려하여 왔다. 이는 스키마가 정해져 있지 않고 다양한 형태의 XML 문서가 사용되는 인터넷 검색엔진과 같은 대용량 환경에서의 응용이 고려되지 않았음을 의미한다. 따라서 본 논문에서는 인터넷상에 널리 방대하게 사용되고 있는 다양한 형태의 XML 문서들을 다루는 방법에 관한 연구에 초점을 두었다.

본 논문에서 제안한 기법은 XRel[2]과 같이 관계형 데이터베이스를 기반으로 한다. 그러나 XRel과 같이 경로 질의에 대해 저장된 경로 전체의 비교를 통해 결과를 얻어내는 것이 아니고, 역 인덱스를 유지하여 경로 비교 연산을 현저히 줄임으로써 질의 수행 속도를 개선한다. 또한, XRel은 문서의 모든 경로를 저장하는 path테이블을 유지하는데, 문서의 개수가 증가하면 상당히 큰 저장 공간의 낭비를 초래하게 된다. 이러한 문제점을 본 논문에서 제안한 계층적 저장구조와 역 인덱스를 사용하여 저장 공간을 줄였다.

본 논문의 구성은 다음과 같다. 2장에서는 XML 검색과 관련된 연구에 대해 기술하고, 기존의 XML 검색 기법들이 대용량 XML 문서 검색에 적합하지 않음을 보인다. 3장에서는 본 논문에서 제안하는 XML의 계층적 저장 기법과 역 인덱스의 구조를 기술한다. 끝으로 4장에서는 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

XML 저장 및 검색 연구를 인덱싱 모델별로 분류를 해보면 그래프 기반 방법, 비트맵 기반 방법, 역 인덱스 기반 방법, 관계형 데이터베이스 기반 방법으로

크게 4가지 분류로 나누어 볼 수 있다.

2.1 그래프 기반 방법

Lore(for Lightweight Object Repository)[1]와 APEX[3](An Adaptive Path Index for XML Data)가 이 분류의 대표적 연구이다.

스텐포드 대학에서 제안한 Lore는 그래프 기반 OEM 데이터 모델을 사용하여 XML 데이터를 저장하고, XML 문서 스키마는 Dataguide[3]를 통해 유지하게 된다. Lore는 XML 문서들이 빈번하게 추가되거나 삭제되면 OEM 데이터 구조의 갱신과 Dataguide갱신 비용의 부하가 상당히 커서 빈번한 삽입이 일어나는 대용량 XML 문서 저장 및 검색 기법으로는 부적합하다. APEX는 그래프를 통해서 XML 데이터의 구조정보를 표현하고, 검색에 자주 사용되는 경로를 해쉬 트리에 저장해 놓고 검색 속도를 향상 시키는 방법이다. 하지만, 새로운 경로가 등장할 때 마다 그래프의 수정이 일어나게 되고, 이와 같은 상황은 인터넷 검색 환경에서 빈번하게 발생 할 것이다.

2.2 비트맵 기반 방법

비트맵 기반 연구는 BitCube[7,8]와 X-Planeb[9]가 대표적인 연구이다.

BitCube는 XML 검색에서 빠른 검색 속도를 통해 뛰어난 성능을 입증한 3차원 비트맵 인덱스 기법이다. 하지만 BitCube는 문서의 증가에 따라 급격하게 인덱스의 크기가 증가하게 되고, 이로 인해 연산 수행 속도가 저하되는 문제점을 보인다. X-Planeb는 이러한 BitCube의 단점을 보완하기 위해 기존의 3차원 배열 구조를 연결 리스트로 변형하여 저장 공간을 축소하였고, 경로 축약 기법을 추가하여 검색 성능을 향상하였다.

2.3 역 인덱스 기반 방법

인터넷 정보 검색에서 가장 많이 사용하는 인덱스 기법으로 역 인덱스 방식을 꼽을 수 있다. 역 인덱스 기반 방법의 연구는 관계형 데이터베이스와 혼용하여 사용되는 사례가 많은데, XML질의를 SQL로 변환하여 질의 처리를 하는 시스템[10,11]에서 많이 사용한다. 본 연구에서도 대량의 XML 문서를 관계형 데이터베이스에 저장하고, 경로에 대한 역 인덱스를 유지하여 검색 속도를 향상하였다.

2.4 관계형 데이터베이스 기반 방법

XRel은 관계형 데이터베이스 기반 방법의 대표적 연구이다. XRel은 XML 데이터를 그림 1과 같은 네 개의 테이블에 저장한다.

Path(label_path_id, label_path)
Element(document_id,label_path_id, start_postion, end_position, sibling_order)
Text(document_id,label_path_id,start_position, end_position, value)
Attribute(document_id,label_path_id, start_position, end_position, value)

그림 1. XRel의 저장 테이블

경로에 대한 질의를 SQL구문으로 변환하여 경로 전체에 대한 비교를 통해서 결과를 추출하기 때문에 문서의 수가 증가할수록 그 검색 성능은 현저히 떨어지게 된다. 또한, 모든 XML 문서에 포함된 경로를 Path 테이블에 저장하기 때문에 문서가 증가할수록 상당히 큰 저장 공간을 낭비하게 된다. 결과도 4개의 테이블의 조인을 통해 얻기 때문에 문서의 증가에 따른 조인 오버헤드가 존재한다.

3. 계층적 저장구조와 역 인덱스

3.1 계층적 저장 구조

제안하는 기법에서 XML 데이터를 저장하는 기본적인 방법으로 계층적 저장 구조를 사용한다. 계층적 저장 구조는 XML의 계층적 구조를 잘 반영하는 구조로 관계형 데이터베이스에 저장된다. XRel에서는 Element, Text, Attribute에 대한 각각의 테이블을 두어 경로 질의 시 Equi-Join을 통해 위의 테이블들을 조인해서 결과를 도출하지만, 본 논문에서는 계층적 저장구조를 사용함으로써 다수의 테이블을 하나로 통합하였다. 또한, XRel에서와 같이 XML 문서에서 추출한 경로들을 따로 보관하지 않아 저장 공간의 효율을 높일 수 있었다.

```
<School>
  <Student>
    <Name>
      <First>문</First>
      <Last>경원</Last>
    </Name>
    <SNum>200442073</Snum>
    <Address>인천광역시</Address>
    <Age>25</Age>
  </Student>
  <Student>
    <Name>
      <First>안</First>
      <Last>성아</Last>
    </Name>
    <SNum>200021229</Snum>
    <Address>서울특별시</Address>
    <Age>23</Age>
  </Student>
</School>
```

그림 2. XML 문서 예

표 1은 위의 그림 2를 계층적 저장구조로 저장한 예를 보인 것이다. XMLDocuments 테이블은 계층적 저장구조를 표현한 다양한 필드로 구성된다.

표 1. XMLDocuments 테이블

did	pid	name	ref	level	step	value
1	1	School	R	0	0	null
1	2	Student	1	1	1	null
1	3	Name	2	2	2	null
1	4	First	3	3	3	문
1	5	Last	3	3	4	경원
1	6	SNum	2	2	5	200442073
1	7	Address	2	2	6	인천광역시
1	8	Age	2	2	7	25
1	9	Student	1	1	8	null
1	10	Name	9	2	9	null
1	11	First	9	3	10	안
1	12	Last	10	3	11	성아
1	13	SNum	9	2	12	200021229
1	14	Address	9	2	13	서울특별시
1	15	Age	9	2	14	23

- XMLDocuments

(did, pid, name, ref, level, step, value)

XMLDocuments 테이블에서 did는 문서에 대한 고유 식별자이고, pid는 경로에 대한 고유 식별자이다. name은 element나 attribute의 명칭이다. ref는 해당 노드의 부모 노드 참조자로 부모 노드의 pid값을 저장한다. level은 해당 노드의 level값을 저장하고, step은 노드의 전위순회 순서를 나타낸다.

3.2 역 인덱스 구조

경로 테이블의 역 인덱스는 element나 attribute의 name을 keyword로 사용한다. 전통적인 역 인덱스와 마찬가지로 본 논문에서 사용하는 역 인덱스는 그림 3과 같이 keyword와 포스트 리스트의 쌍으로 구성된다.

keyword	posting_list
School	<1,(1)>,<2,(1)>,<3,(1)>.....
Student	<2,(2)>,<3,(2)>,<4,(2)>.....
Name	<3,(3)>,<4,(3)>,<5,(3)>
First	<4,(4)>,<11,(4)>
Last	<5,(4)>,<12,(4)>
	.
	.
	.

그림 3. Inverted-index table

포스팅 리스트는 <pid, sequences> 의 쌍으로 구성되는데, pid는 해당 키워드가 존재하는 경로의 경로 참조자 이고, sequences은 경로가 시작하는 위치부터 해당 키워드가 나타나는 위치까지의 거리들의 집합이다.

3.3 알고리즘

다음 그림 4는 본 논문에서 제안한 XML 문서 검색의 Pid 검색 알고리즘을 보여주고 있다. 입력으로 들어오는 패스 표현은 'a/b//c' 와 같은 XPath 기반 형식을 따른다.

입력으로 들어오는 패스와 매치되는 키워드를 역 인덱스 리스트 L에서 찾아서 List[n]에 저장하게 되는데 키워드와 포스팅 리스트 쌍으로 저장된다.

'/' 인 패스 구분자인 경우 포스트 리스트에서 pid가 같은 것 중 시퀀스의 차이가 1인 것을 임시 포스팅 리스트에 저장하고, '//' 인 경우는 시퀀스가 큰 것을 저장한다. 이를 List의 길이만큼 루프를 돌려 최종적으로 포스팅 리스트를 반환하게 되면 Pid를 얻을 수 있다.

```
SearchPids(P,L)
Input : PathExpr P, Inverted_Index_List L
Output : Matched Set of Posting_List
Algorithms :
begin
initialize List[n]; // L에서 P의 노드 n개와 매치되는 Keyword와 Posting_List 쌍을 저장
 for(i=0;i<n;i++) do
 begin
  if(List[i] ⊄ L_keywords_list) then
  return null;
 end
 for(i=n;i>0;i--) do
 begin
  for(j=0;j<List[i].Posting_List.length;j++) do
  begin
   for(k=0;k<List[i-1].Posting_List.length;k++) do
   begin
    if(List[i].Posting_List[j].pid ==
       List[i-1].Posting_List[k].pid) then
     if(List[i].separator == "/") then
      if(List[i].Posting_List[j].sequence-1 ==
         List[i-1].Posting_List[k].sequence) then
       tmp_P_list.append(List[i-1].Posting_List[k]);
       break;
     else if(List[i].separator == "//") then
      if(List[i].Posting_List[j].sequence >
         List[i-1].Posting_List[k].sequence) then
       tmp_P_list.append(List[i-1].Posting_List[k]);
       break;
     end
    end
    if(tmp_P_List == null)
    then
       return null;
    else
       List[i-1].Posting_List = tmp_Posting_List;
 end
 return List[0].Posting_List;
end
```

그림 4. Pid 검색 알고리즘

4. 결론

본 연구는 인터넷 환경에서 전형적으로 존재하는, 다양하고 방대한 양의 XML 문서들의 검색에 초점을 두었다. 제안한 기법의 주요 기술은 관계형 데이터베이스 기반 방법과 역 인덱스 기법을 혼용한 경로 검색 기법이다. 앞으로 인터넷 XML 검색엔진 등의 대용량 환경에서 다양한 활용이 기대된다.

향후 과제로는 대량의 XML 문서를 대상으로 하는 경로 축약 기법에 관한 연구가 필요하다.

참고문헌

[1] J.McHugh, S.Abiteboul, R.Goldman, D.Quass, and J.Widom, "Lore: A Database Management System for Semistructured Data," ACM SIGMOD Record, Vol. 26, No. 3, pp.54-66, 1997.

[2] M.Yoshikawa and T.Amagasa, "XRel: A Path-Based Approach to Storage and Retrieval of XML Documents using Relational Databases," ACM Transactions on Internet Technology, Vol. 1, No. 1,pp.110-141, 2001.

[3] C.W.Chung, J.K.Min, and K.Shim, "APEX: An Adaptive Path Index for XML Data," Proc. of the ACM SIGMOD, Vol. 31, No. 2, pp.121-132, 2002.

[4] R,Goldman and J. Widom, "Dataguides: Enabling Query Formulation and Optimization in Semistructured Databases," Proc. of the 23rd Int. Conf. on Very Large Data Bases, pp.436-445, 1997.

[5] C.Zhang, J.Naughton, D.Dewitt, Q.Luo, and G. Lohman, "On Supporting Containment Queries in Relational Database Management Systems," Proc. of the ACM SIGMOD, Vol. 30, No. 2, pp.425-436, 2001.

[6] B.Cooperm, N.Sample, M.J.Franlin, G. R.Hjaltason, and M.Shadmon, "A Fast Index for Semistructured Data," Proc. of the 27^{th} Int. Conf. on Very Large Data Bases, pp.341-350, 2001.

[7] J.Yoon, V.Raghavan, V.Chakilam, and L.Kerschberg, "BitCube: A Three-Dimensional Bitmap Indexing for XML Documents," J. of Intelligent Information System, Vol. 17, pp.241-254, 2001.

[8] J.Yoon, V.Raghavan, and V.Chakilam, "BitCube: Clustering and Statistical Analysis for XML Documents," 13th Int. Conf. on Scientific and Statistical Database Management, Virginia, 2001.

[9] 이재민, 황병연, "xPlaneb: XML 검색을 위한 3차원 비트맵 인덱스," 정보과학회논문지, 31권, 3호, 2004년.

[10] 민경섭, 김형주, "상이한 구조의 XML 문서들에서 경로 질의 처리를 위한 RDBMS 기반 역인덱스 기법," 정보과학회논문지, 30권, 4호, 2003년.

[11] 서치영, 이상원, 김형주, "XML 문서에 대한 RDBMS에 기반을 둔 효율적인 역색인 기법," 정보과학회논문지, 30권 1호, 2003년.