XML 문서에서 효율적인 질의를 위한 구조 조인 메커니즘

이명진° 김희경 채기준
이화여자대학교
{maya012°, whiteyo}@ewhain.net, {kjchae}@ewha.ac.kr

Structural join mechanism for efficient query in XML document

Myungjin Lee° Heekyung Kim Kijoon Chae
Ewha Womans University

요 약

인터넷 환경에서 데이터 저장과 전송을 위한 표준으로 XML이 부각되고 있다. 이에 따라 XML자체에 대한 관심뿐 아니라 XML 질의 처리를 위한 연구도 활발하게 이루어지고 있다. 이러한 연구는 XML문서 자체의 내용 뿐 아니라 구조에 대한 질의에 대해서도 진행되고 있다. 구조에 대한 질의를 위해 경로 질의가 사용되고 있으며, 최근 경로 질의를 효과적으로 처리하기 위하여 여러 가지 색인 기법들이 연구되고 있다. 본 연구에서는 기존에 제안된 대표적인 두 가지 색인기법인, 경로 색인 기법과 구조 조인 기법에 대한 특징에 대해 살펴보고, 구조 조인 기법에 대한 기존연구의 알고리즘의 성능을 개선시킬 수 있는 메커니즘을 제안하였다. 기존 연구에서는 질의에 해당하는 모든 엘리먼트들을 비교하여 결과를 뽑아내지만 제안한 메커니즘은 입력 값의 형태를 변형하여 비교하는 엘리먼트의 수를 줄여 더 적은 조인연산을 수행함으로서 성능을 향상시킬 수 있었다. 이러한 결과를 4절에서 실험을 통해 검증하였다. 실험 결과 기존 방법에 비해 실험한 질의에 따라 최대 34%, 최소 7%의 성능 향상을 이룰 수 있었다.

1. 서 론

최근 인터넷 환경에서 데이터 저장과 전송을 위한 표준으로서 XML이 부각되며 많이 사용됨에 따라, XML 저장소와 질의 처리에 관한 연구가 활발하게 이루어지고 있다. XML에서는 질의 처리를 위해 문서의 내용 뿐 아니라, 구조에 대한 질의가 가능한 경로 질의를 사용하고 있다.

이와 같은 경로 질의를 처리하기 위해서는 시스템에 저장하고 있는 XML 문서들을 하나씩 읽어서 경로 질의에 만족하는 지를 판단하기 위해 문서 구조를 탐색해야 한다. 즉, 각 문서를 디스크에서 메모리로 읽어 들여야 하기 때문에 경로 질의를 처리하기 위한 수행비용이 매우 커진다. 비용을 줄이기 위해 각 문서를 메모리에서 읽어들이는 대신 색인 구조를 두고 색인을 먼저 조사하여 경로 질의를 만족하는 문서의 목록을 얻음으로서 디스크 접근 빈도를 줄이고 질의 처리의 성능을 향상 시킬 수 있다.

위에서 설명한 경로 질의를 처리하는 데 따른 문제점을 해결하고 경로 질의를 효과적으로 처리하기 위하여 여러 가지 색인 기법들이 연구되었다.

본 연구에서는 제안된 색인 기법들에 대해 살펴보고 그 중 구조 조인 기법으로 제안된 기존 연구[1]를 분석하고 이 방법의 단점을 보완할 수 있는 기법을 제안하였다.

2. 관련 연구

2.1 경로 색인 기법

경로 질의 처리를 위해 초기에 경로 색인 기법이 많이 연구되었다. 이 기법은 DataGuide[2]에 이론적인 기반을 두고 있으며, XML 데이터 구조에서 발생 가능한 모든 경로에 대한 색인 그래프(index graph)를 별도로 구축하여 경로 질의 처리를 위해 원본 데이터 그래프보다 크기가 작은 경로 색인 그래프를 우선 탐색함으로써 질의 처리 비용을 줄이는 방법이다.

XML 데이터를 나타내는 데이터 그래프는 많은 식별 경로들(identical paths)로 구성된다. 색인 기법들은 이러한 많은 식별 경로들을 데이터 그래프 내 한 개의 경로로 통합시킨다. 이러한 통합을 이용하여 색인 그래프(index graph)라 불리는 작은 크기의 그래프 형태를 지니도록 할 수 있다. 그래서 원본 데이터 그래프의 탐색이 아닌 색인 그래프를 탐색함으로써 질의를 처리할 수 있게 된다.

그러나 DataGuide를 비롯한 경로 색인 기법은 다음과 같은 문제점을 갖는다. 원본 데이터 그래프에 비해 충분히 작은 크기의 색인 그래프를 가진다는 것을 보장할 수 없다는 것과 색인 그래프의 구조가 고정적이지 않다는 특징을 갖는다. 마지막으로 분기(branching)를 가지는 경로 질의를 지원할 수 없다는 단점을 지닌다.

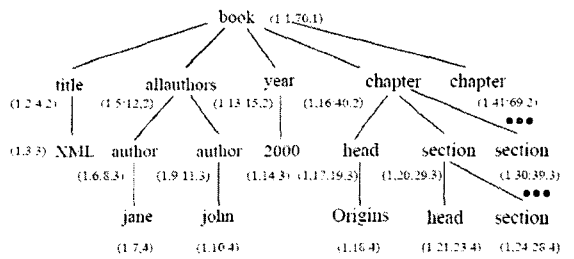
본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음.

2.2 구조 조인 기법

경로 색인 기법이 가지는 여러 가지 문제점을 해결하고자 구조 조인 기법이 제안되었고 이에 대한 연구가 활발하다. 구조 조인 기법은 인접한 두 엘리먼트들 사이에 존재하는 구조적인 포함 관계를 살펴 질의를 처리한다. 엘리먼트들 사이의 포함 관계 규명을 위한 XML 데이터 트리 내의 모든 노드들을 깊이 우선 탐색 기법으로 위치 정보를 부여하고 질의내의 각 엘리먼트에 해당되는 데이터 리스트를 역색인 내에서 한 번씩 스캔하면서 위치 정보를 서로 비교하여 판별한다. 구조 조인을 사용하면 문서 전체에 대한 탐색을 하지 않고 해당하는 엘리먼트들의 역색인 리스트만을 비교하여 질의 처리를 수행하는 장점을 갖는다.

3. 제안한 효율적인 구조 조인 메커니즘

XML문서는 트리 모델로 전환 후 질의 처리 등이 이루어진다. 경로 질의를 처리하기 위한 구조 조인 기법을 수행하기 위해 Numbering Scheme의 구축이 선행되어야 한다. Numbering Scheme이란 XML 문서에 대한 파싱 단계에서 각 노드들에 대한 시작과 끝, 레벨에 대한 위치 코딩을 하는 것을 말한다. (DocId, StartPos: EndPos, LevelNum) 와 같은 형식으로 표현한다. 다음 [그림 1]은 XML 문서를 트리형식으로 표현한 것에 Numbering Scheme을 이용하여 노드의 번호를 붙인 것이다.



[그림 3] XML예제의 트리표현과 Numbering Scheme

경로 질의를 처리하기 위하여 구조 조인 기법은 전체 경로 질의를 이항관계구조(binary structural relationship)로 분해한다. [그림 1]에 나타난 예제의 경우 조상-자손 관계로는 (book, author)가 있으며, 부모-자식관계로는 (book, title), (title, XML) 그리고 (author, jane)이 존재한다. 이러한 관계는 Numbering Scheme으로 매겨진 각 엘리먼트의 위치 정보(StartPos:EndPos)를 이용하여 규명할 수 있다. 조상-후손 관계의 판별을 위해서는 조상에 해당하는 엘리먼트의 (시작:끝) 위치가 후손의 (시작:끝)을 포함하는지를 평가한다. 예를 들어 author의 시작인 5가 book의 시작인 1보다 크고 book의 끝인 70이 author의 끝인 4보다 작기 때문에 author가 book의 후손임을 알 수 있는 것이다. 만약 엘리먼트들 사이가 부모-자식 관계라면 부모에 해당하는 엘리먼트의 (시작:끝) 위치가 자식의 (시작:끝)을 포함하면서 동시에 레벨의 차이가 1인지를 추가로 확인한다. title이 book의 후손임을 위의 예와 비슷한 과정을 통해 알아낸 다음 title의 레벨(2) 과 book의 레벨(1)의 차이가 1이므로 title은 book의 자식이 되는 것을 알 수 있다.

조상-자손관계 혹은 부모-자식관계(e1, e2)에서 e1, e2에 대응되는 트리 노드의 리스트를 각각 AList = [a1, a2,...] 와 DList = [d1, d2,...]이라고 하자. 각 리스트는 (DocId, StartPos) 값으로 정렬되어 있다.

기존의 제안된 구조 조인 기법[1]은 AList와 DList의 엘리먼트들의 위치정보와 레벨정보를 이용하여 결과 값을 생성한다. 이 방법을 사용할 경우 각 엘리먼트에 대응하는 모든 데이터를 추출하여 비교 연산을 수행한다는 단점을 지닌다. 이를 보완하기 위하여 우리는 입력 데이터를 변형하여 비교 대상을 가치지기 할 수 있도록 하였다.

다음 [표 1]은 우리가 기존연구를 보완하여 제안한 알고리즘이다.

[표 1] Modified-Tree-Merge-Anc

```
Algorithm Modified-Tree-Merge-Anc (AList, DList)
/* Assume that all nodes in AList and DList have
the same DocId */
/* AList is the list of potential ancestors, in sorted
order of StartPos */
/* DList is the list of potential descendants in sorted
inverse order of StartPos */

begin-desc = DList->firstNode; OutputList = NULL;
for (a = AList->firstNode; a ! = NULL; a =
a->nextNode) {
  for (d = begin-desc; (d ! = NULL && d.StartPos
  < a.StartPos); d = d->nextNode){
    /* skipping over unmatchable d's */ }
  begin-desc = d;
  for (d = begin-desc; (d ! = NULL && d.EndPos <
  a.EndPos); d = d->nextNode) {
    if (!(a.startPos < d.startPos))  break;
    else if ((a.StartPos < d.StartPos) && (d.EndPos
    < a.EndPos){&& (d.LevelNum = a.LevelNum +
    1)]) {
      /* the optional condition is for parent-child
      relationships */
      append (a,d) to OutputList; }
  }
}
```

4. 실험 및 검증

실험을 하기 위해 MS-Windows XP 환경의 512MBye 메인 메모리를 가진 Pentium 4-1.9GHz 와 MS Visual Studio 6.0 을 이용하여 시뮬레이션 하였다.

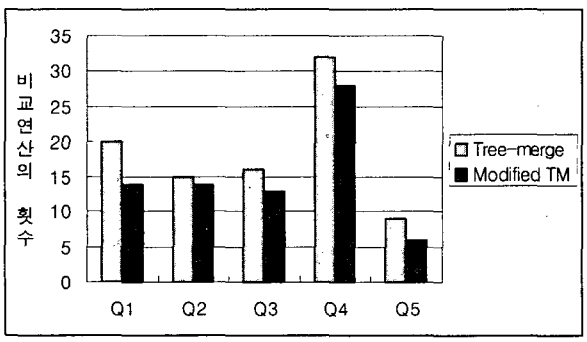
경로 질의에 대한 구조 조인 기법의 효율성을 측정하기 위하여 [표 1]과 같이 여러 가지 형태의 경로 질의를 사

용하였다. 또한 실험을 위한 데이터로는 기존 연구[1]의 XML 예제를 사용하였다.

[표 2] Query path Expression

Query	XQuery Path Expressing
Q1	book[title='XML']//author[.='jane']
Q2	book/chapter[head='Origins']
Q3	book[year='2000']//author
Q4	book/chapter[section]/head[.='Origins']
Q5	Book/allauthors/author[.='john']

제안한 방법을 검증하기 위해 기존 연구 방법 중 Tree-Merge ancestor와 제안한 방법이 각각의 경로 질의에 대해 OutList를 생성하기 위하여 얼마나 많이 비교연산을 수행하는 지를 측정하였다.
다음 [그림 2]은 실험 결과이다. 위의 [표 1]의 각각 질의에 대해 OutList를 만들기 위한 Tree-merge알고리즘과 이 연구에서 제안한 알고리즘의 비교연산의 횟수에 대한 그래프이다.



[그림 4] 기존 Tree-merge 알고리즘과 제안한 기법의 비교 연산 횟수의 결과

질의 Q1에 대해 Tree-merge는 20번을 비교연산을 하여야 하지만 제안한 기법에서는 필요 없는 비교연산을 줄임으로써 14번만 비교연산을 해도 같은 결과가 나오게 된다.
제안한 방법이 얼마나 성능을 향상시키는지에 대하여 수치적으로 알아보기 위해 다음과 같은 식을 사용하였다

$$P \approx \frac{\overline{|Q_i|} - |Q_i|}{\overline{|Q_i|}} \times 100 \qquad \text{[수식 1]}$$

P는 성능향상정도를 나타내며 $|Q_i|$는 i번째 질의를 본 연구에서 제안한 방법으로 수행한 경우의 비교 횟수이다. $\overline{|Q_i|}$는 기존연구의 방법으로 수행한 경우의 비교 횟수이다.
[수식 1]을 사용하여 계산 한 결과 각각의 질의에 대해 약 30%, 7%, 19%, 13%, 34%의 성능 향상이 이루어졌음을 알 수 있었다.
본 연구에서는 크기가 작고 간단한 XML문서와 질의들을 실험에 사용하였지만 좀 더 복잡하고 방대한 XML문서와 질의들을 사용할수록 보다 큰 성능향상을 가져 올 것을 기대한다.

5. 결론

본 연구에서는 색인 기법 중 하나인 구조 조인 기법의 단점을 보완하여 경로 질의를 좀 더 효율적으로 처리하는 새로운 구조 조인 기법을 제안하였다.
구조 조인 기법은 질의 내의 각 엘리먼트들의 정보와 엘리먼트들 사이의 관계 정보를 이용하여 조상(부모)리스트와 자손(자식)리스트를 생성하고 각 엘리먼트의 데이터를 비교하여 포함 관계를 규명하는 기법이다.
기존의 구조 조인 기법은 경로질의에 해당하는 조상(부모)리스트와 자손(자식)리스트 내의 모든 엘리먼트의 데이터를 다 비교하여 포함 관계를 규명하였다. 그에 반해 본 연구에서 알고리즘의 입력으로 받는 자손(자식)리스트와 조상(부모)리스트를 변형하여 비교가 이루어지기 전 가치지기를 수행 할 수 있었다. 이러한 방법으로 더 적은 조인 연산을 수행함으로써 불필요한 비교를 줄일 수 있었고, 기존 방법에 비해 최대 34%, 최소 7%의 성능 향상을 이룰 수 있었다.

6. 참고문헌

[1] S. Al-Khalifa et al. "Structural Joins: A Primitive for Efficient XML Query Pattern Matching" In IEEE International Conference on Data Engineering, 2002.
[2] Goldman R., and Widom J. "Dataguides: enabling query formulation and optimization in semistructured databases" In proc. of the conference on Very Large Data Bases, 1997.