구조적 세미조인을 이용한 XML 경로 패턴 처리

손석현° 신효섭 서지위
건국대학교 신기술 융합학과
{myviki, hsshin, xuzhiwei}@konkuk.ac.kr

XML Path Pattern Evaluations Using Structural Semi-Joins

Seokhyun Son° Hyoseop Shin Zeiwei Xu
Department of Advanced Technology Fusion
Konkuk University

1. 서론

XML 질의 처리시, XML노드 들의 조상-후손(ancestor-descendant) 과 부모-자식(parent-child) 간의 관계는 중요한 이슈이다. 그 이유로는 XML 질의의 경로 표현식(path expression)은 각각의 구조적 관계의 조합이 중요한 역할을 차지 하고 있기 때문이다. 예를 들면, XPath 질의 중, "Paper[Title='XML']/Author" 의 경우 'Paper' 와 'Title', 'Paper' 와 'Author' 같은 두 개의 부모-자식 간의 구조적 관계가 포함되어 있다. 이전에 제시되었던 구조적 조인[2]은 수많은 XML 노드 사이에 구조적 관계를 통한 질의처리 방법을 최초로 제시한 반면, 여전히 XML의 경로 패턴 처리 과정에서 불필요한 중간결과를 야기 시킨다. 이와는 달리 구조적 세미조인[1]은 XML 경로패턴을 효율적으로 처리하기 위하여 조인 결과를 조상노드 혹은 자손노드로 한정하는 새로운 연산자이다. 일반적으로, XML 경로식 $n_1//n_2//.../n_k$ 를 처리할 때, 전 방향 방식과 역 방향 방식으로 나뉠 수 있다. 전 방향 방식은 i의 오름차순으로 (n_i, n_{i+1})을 차례대로 처리하고, 역방향 방식은 i의 내림차순으로 처리 한다. 어느 경우에나, 조상노드 또는 자손노드만 필요하다는 것을 알 수 있다. 본 논문에서는 구조적 세미조인을 이용한 효율적인 XML 경로패턴 처리 알고리즘을 제시한다.

2. 구조적 세미조인

스택기반의 기존 구조적 조인 알고리즘에 비하여 자손 노드만을 반환하도록 설계된 Strucutural-SemiJoin-Desc[1]는 계산 비용 및 메모리 효율성 면에서 뛰어남을 알 수 있다. 그 이유는 알고리즘 진행중 방문하는 조상노드의 가장 큰 시작위치값(start_pos)을 나타내는 조상 인디케이디를 통해서 기존 방법에서 사용한 스택을 대체하였기 때문이다. 마찬가지로 조상 노드 만을 반환하도록 설계된 Structural-SemiJoin-Anc[1]는 조상 노드를 저장하기 위해 스택을 사용하되, 매치되는 자손노드가 나오는 즉시, 조상노드가 반환되면서, 리셋이 된다. 그로 인해, 기존의 구조적 조인 알고리즘에서 사용되는 Self와 Inherit리스트를 필요로하지 않게 된다. Self와 Inherit 리스트의 크기는 이어지는 노드가 많을수록 더 욱 커지기 때문에 XML문서가 크거나 태그가 많을 경우, 더욱 뛰어난 효율성을 나타낼 것임을 알 수 있다. Strucutural-SemiJoin-Desc와 Structural-SemiJoin-Anc에 대한 자세한 설명은 논문[1]을 참조하면 된다.

3. 다중 경로식 처리 알고리즘(MultiPathJoin)

다중 경로식 처리 알고리즘(MultiPathJoin)은 주어진 경로 식을 전 방향 혹은 역방향으로 탐색하면서 구조적세미조인 알고리즘을 순차적으로 적용한다. 전 방향 처리의 경우 구조적 세미조인 알고리즘은 후손 노드를 반환하는 Strucutural-SemiJoin-Desc[1]를, 역방향의 경우에는 조상노드를 반환하는Structural-SemiJoin-Anc[1]를 사용한다. 그림 1은 전 방향 다중 경로 식 처리 알고리즘을 나타낸다.

```
1  Algorithm MultiPath_Foward (NumberofTags, Tagnames)
2  OutputList = NULL;
3  InputList1 <- NodeList(tagnames[1]);
4  for (i=1; i<numoftags-1; i++) {
5      InputList2 <- NodeList(tagnames[i+1]);
6      OutputList <- Strucutural-SemiJoin-Desc(InputList1, InputList2);
7      InputList1 <- OutputList;
8  }
9
10 return OutputList;
```

그림1. 전방향 다중 경로식 처리 알고리즘

4. 실험결과

실험에서는 구조적 세미조인 알고리즘[1]과 스택기반의 구조적 조인 알고리즘[2]을 다중경로 처리 알고리즘에 적용할 때의 성능을 비교한다. 실험에 사용된 XML 문서의 크기는 약 200M byte이다. 표1과 표2는 실험에 사용한 Tag의 개수와 다중 XPath질의문을 나타낸다. 그림 2에서는 다중 경로 식 처리 알고리즘을 통한 구조적 세미조인의 성능을 나타낸다. X축은 질의의 번호를 나타내며, Y축에서는 각 질의에 대한 반응 시간을 나타낸다. 모든 질의에 대해서, 구조적 세미조인 알고리즘이 스택기반의 구조적 조인 알고리즘보다 우수한 성능을 나타낸다. 특히, 역방향 다중 경로식 처리의 경우(질의 번호 2, 4, 6, 8)는, 최대 20배 이상의 속도 차이가 난다. 이는, 조상 노드에 대하여 자손 노드들 수가 증가할수록 더욱더 향상된 성능을 보여준다는 것을 나타낸다.

No.	Query	Return	Results
1	department//department //name//name	Descendant	94,836
2	Department[department [manager[name]]]	Ancestor	17
3	Department//department //manager	Descendant	398,768
4	Department[department [manager]]	Ancestor	17
5	Department//employee //email	Descendant	597,301
6	Department[employee [email]]	Ancestor	286,311
7	Department//department //department//email	Descendant	895,954
8	Department[department [department[email]]]	Ancestor	8

표2. 실험에 사용한 다중 XPath 질의문

Tag Name	Number of Tags
department	397,947
manager	597,302
employee	796,671
name	3,383,862
email	895,957

표1. 실험에 사용된 데이터 베이스 통계치

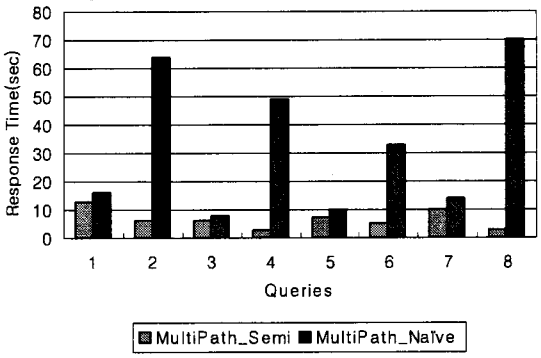

그림2. 다중 경로 처리식을 통한 알고리즘별 반응속도

5. 결론

본 논문에서는, 다중 경로 식 처리 알고리즘을 소개하고, 이전에 제시한 구조적 세미조인과 스택기반의 구조적 조인 연산자를 다중 경로 식 처리 관점에서 비교하였다. 다중 경로 식 처리 알고리즘은 전 방향 혹은 역방향으로 노드들을 순차적으로 처리하는 알고리즘이다. 실험을 통해, 구조적 세미조인을 이용한 다중 처리 알고리즘이 스택기반의 구조적 조인을 사용한 다중 처리 알고리즘 보다 우수한 성능을 나타냈으며, 역방향 처리의 경우 최대 20배 이상의 성능 차이를 보여주었다.

참고문헌

[1] Seokhyun Son, Hyoseop Shin and Zhiwei Xu .Structural Semi-Join : A light-weight structural join operator for efficient XML query pattern matching. In Proc. of the IDEAS 2007 conference, Baff, Canada, Sep. 2007.

[2] Divesh Srivastava, Shurug Al-Khalifa, H. V. Jagadish, Nick Koudas, Jinesh M. Patel, and Yuqing Wu. Structural joins: A primitive for efficient XML query pattern matching. In Proc. of the 2002 IEEE conference on Data Engineering, San Jose, USA, Feb. 2002.