세그먼트 기반 XML 문서 필터링[1)]

권준호[O1] Praveen Rao[2] 문봉기[2] 이석호[1]
[1]서울대학교 전기컴퓨터공학부
joonho@db.snu.ac.kr shlee@snu.ac.kr
[2]아리조나 주립대학교 전산학과
{rpraveen, bkmoon}@cs.arizona.edu

An XML Document Filtering based on Segments

Joonho Kwon[O1] Praveen[2] Rao Bongki[2] Moon Sukho Lee[1]
[1]School of Electrical Engineering and Computer Sciences, Seoul National University
[2]Department of Computer Science, University of Arizona

최근 XML 문서 필터링에 기반한 출판-구독(publish-subscribe) 시스템이 많은 관심을 받고 있다. 전형적인 출판-구독 시스템에서, 구독자들은 XPath 언어로 명세된 프로파일로 자신들의 관심을 표현하고, 새로운 내용들은 사용자 프로파일에 대하여 매칭 여부를 판단하여 관심을 가지고 있는 사용자들에게만 배달된다. 구독자의 수와 그들의 프로파일이 증가할수록, 시스템의 확장성이 출판-구독 시스템의 중요한 성공 요소가 된다.

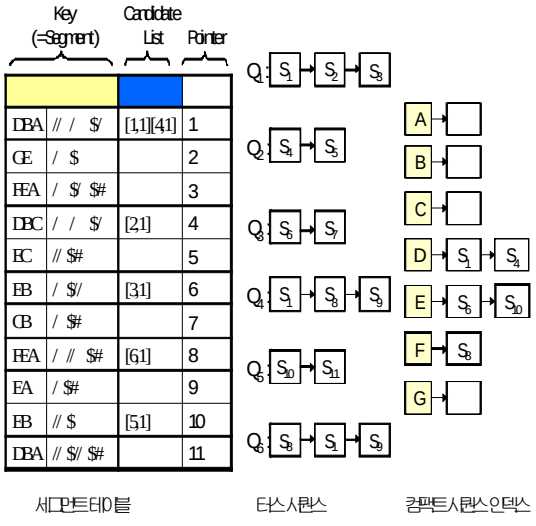
FiST 시스템[1]은 가지형 패턴과 입력 문서를 Prufer 시퀀스로 변환하여 전체 가지형 패턴 매칭(holistic twig pattern matching)을 수행한다. FiST의 매칭 알고리즘은 점진적인 서브시퀀스 매칭 단계와 브랜치 노드 검증 단계의 두 단계의 과정으로 결과들을 찾는다. 그렇지만 FiST 시스템은 사용자들 간의 유사하거나 동일한 프로파일을 공유하여 효율적으로 처리하지는 못하였다. 이 논문에서는 YFilter[2] 시스템과 같이 사용자 프로파일 의 공유 처리를 통한 성능 향상을 위하여 FiST를 확장한 세그먼트 기반의 XML 문서 필터링 시스템인 SFiST 시스템을 제안한다.

SFiST 시스템은 XML 문서 필터링에서 중복된 처리를 없애기 위해서 가지형 패턴의 사용자 프로파일에서 세그먼트를 추출하여 해시 기반의 세그먼트 테이블에 저장하고 유지한다. 세그먼트는 아래의 정의 1에 의해 구할 수 있다.

정의 1. 가지형 패턴을 변환한 프로파일 시퀀스의 첫 시퀀스 노드에서 시작하여 각각의 노드를 검사하면서 공유 세그먼트의 끝인지 여부를 검사한다. 다음 공유 가능한 세그먼트는 끝으로 표시한 시퀀스 노드의 다음 시퀀스 노드부터 시작한다. 세그먼트의 끝 여부는 아래의 두 가지 조건으로 결정할 수 있다.
(1) 시퀀스 노드가 내부 브랜치 노드이면, 이 노드는 세그먼트의 끝을 의미한다.
(2) 시퀀스 노드가 프로파일 시퀀스의 마지막 노드이면, 이 노드는 세그먼트의 끝을 의미한다.

그림 1(a)와 같이 사용자 프로파일을 세그먼트에 기반한 해시 테이블 형태의 터스 시퀀스로 표현하고, 효율적인 필터링을 위한 인덱스 구조인 컴팩트 시퀀스 인덱스에도 세그먼트를 이용한다. 그림 1(b)는 세그먼트에 기반한 인덱스 구조를 사용한 점진적인 서브 시퀀스 매칭 알고리즘을 보여준다.

1) 본 연구는 2007년도 두뇌한국21사업과, 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원 사업(IITA-2006-C1090-0603-0031)의 연구결과로 수행되었음



(a) 인덱스 구조

```
Input: L - Prufer 시퀀스 레이블
procedure FindSubsequence(L)
1  CurrentSegments ← CompactSequenceIndex[L];
2  foreach Segment s in CurrentSegmenst do
3     if extendedStackTest(s) is true then
4        foreach pair [seqId, segPos] in Candidate List of s do
5           Segment seg ← (segPos+1)th segment of Q_seqId;
6           segL ← the first prufer sequence label of seg;
7           Add the pair [seqId, segPos+1] into the
                Candidate List of seg;
8           Copy seg into the CompactSequenceIndex[segL];
        endfch
     endif
  endfch
end
```

(b) 점진적인 서브시퀀스 매칭

그림 1 세그먼트 기반 인덱스 구조와 알고리즘

DBLP와 Treebank DTD를 이용하여 생성한 사용자 프로파일과 XML 문서를 이용한 실험을 통하여 다음과 같은 결과를 얻을 수 있었다.

(1) 서로 다른 10,000개의 데이타를 생성한 후 중복도를 5에서 25까지 변화시켰을 때, 중복도가 증가함에 따라 필터링 시간도 증가함을 알 수 있다. 또한 중복도가 증가함에 따라 FiST 시스템과 SFiST 시스템의 시간 차이가 점점 더 벌어지는 경향을 알 수 있다. SFiST 시스템이 기존의 FiST 시스템보다 DBLP 데이타셋의 경우에는 최대 52%, Treebank 데이타의 경우에는 최대 83%의 속도 향상을 보였다.

(2) 사용자 프로파일의 수를 50,000개부터 150,000개 까지 25,000개씩 증가시킨 실험에서, 사용자 프로파일이 증가함에 따라 필터링 시간도 증가지하지만, SFiST 시스템의 필터링 시간이 FiST 시스템의 필터링 시간보다 덜 증가하는 결과를 얻었다.

(3) XML 문서 필터링 시스템의 중요한 척도인 메모리의 사용량을 비교한 결과, 세그먼트와 세그먼트 테이블을 사용하는 SFiST 시스템이 FiST보다 최대 40% 적은 양의 메모리를 사용하는 것을 알 수 있다.

(4) SFiST 시스템은 DBLP 보다 구조가 복잡하고 엘리먼트 개수가 많은 Treebank DTD를 사용한 데이타셋을 이용한 실험에서 더 효율적이었음을 알 수 있었다.

참고문헌

[1] Joonho Kwon, Praveen Rao, Bongki Moon, Sukho Lee, “FiST: Scalable XML Document Filtering by Sequencing Twig Patterns", In Proceeding of the 31st VLDB Conference, pp. 217-228, 2005.

[2] Yanlei Diao, Mehmet Altinel, Michael J. Franklin, Hao Zhang and Peter Fischer, “Path sharing and predicate evaluation for high-performance XML filtering,” ACM Trans. Database Syst, 28(4) : 467-516, 2003.