중복을 고려한 효율적인 RDF 프로버넌스 압축 기법

Efficient RDF Provenance Compression Scheme Considering Duplication

한 지 은, 육 미 선, 노 연 우, 김 대 윤, 임 종 태, 복 경 수, 유 재 수
충북대학교 정보통신공학과

Han ji-eun, Yook mi-sun, Noh yeon-woo, Kim dae-yun, Lim jong-tae, Bok kyoung-soo, Yoo jae-soo
Chungbuk National University

요약

본 논문에서는 대용량의 프로버넌스를 압축 저장하기 위한 OPM 기반의 RDF 프로버넌스 압축 기법을 제안한다. 제안하는 기법은 이미 존재하는 데이터 프로버넌스 및 새로운 데이터 프로버넌스를 사전을 기반으로 숫자 데이터로 인코딩한다. 또한 데이터 처리의 중복되는 부분은 서브그래프를 통해 압축한다.

I. 서론

컴퓨팅 기술 및 네트워크의 발달로 사용자들은 유용한 정보들을 공유하는 서비스가 활발하게 이용되고 있다. 이러한 서비스의 하나로 최근 위키피디아나 e-science같은 협업 저장소 환경이 등장했다. 데이터 프로버넌스는 데이터의 근원 정보나 히스토리를 나타내는 메타데이터이며 이력정보를 관리하는데 유용한 데이터이다. 이러한 데이터 프로버넌스를 관리하기 위해 IPAW'06에서 처음 Open Provenance Model(OPM)이 등장했다[3]. 데이터가 지속적으로 활용됨에 따라 데이터 프로버넌스는 원본 데이터에 비해 수십 배의 대용량 데이터가 될 수 있으며 많은 중복이 발생한다. 데이터 프로버넌스가 많이 발생하는 서비스인 위키피디아에서는 하나의 문서를 여러 명의 사용자가 변경할 수 있다. 또한 한명의 사용자가 여러 문서를 생성, 변경 하는 등 다양한 활동을 할 수 있다. 이러한 데이터 프로버넌스는 원본데이터에 비해 수십 배에 달하게 된다.

기존의 프로버넌스 압축 기법으로 중복되는 데이터 프로버넌스 레코드와 노드를 삭제하고 동일한 서브트리를 찾아서 중복을 제거를 수행하며 상속과 분해를 통해 데이터 프로버넌스를 관리하는 기법이 있다[1]. 하지만 RDF 프로버넌스 데이터를 관리하지 못한다. 또한 웹 그래프 기반의 압축방법과 사전기반의 인코딩 기법을 결합시킨 프로버넌스 압축 기법에서는 표준 OPM에 적용하기 어려우며, 조상 노드의 중복이 발생하지 않을 경우 압축 효율이 저하된다[2].

본 논문은 OPM을 이용하여 대용량의 데이터 프로버넌스를 관리하기 위한 압축 기법을 제안한다. 제안하는 기법은 데이터 프로버넌스를 사전을 기반으로 인코딩하여 저장을 하고 데이터 처리의 중복되는 부분은 서브그래프로 만들어 압축 저장을 한다.

* 교신저자 : yjs@chungbuk.ac.kr

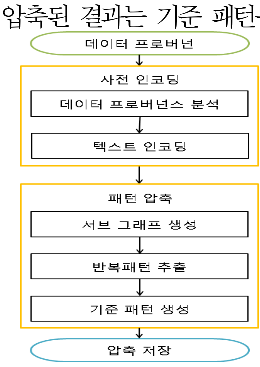
이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(No.2013R1A2A2A01015710)과 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음(IITP-2015-H8501-15-1013

Ⅱ. 제안하는 RDF 프로버넌스 압축 기법

1. 특징

데이터 프로버넌스는 원본데이터에 비해 수십 배에 달할 수 있다. 이러한 문제점을 해결하기 위해 기존의 기법과는 달리 OPM 기반의 데이터 프로버넌스 저장 기법을 제안한다.

[그림 1]은 제안하는 기법의 압축 과정을 나타낸다. 데이터 프로버넌스가 입력되면 사전 인코딩 모듈에서 데이터 프로버넌스를 분석한다. 분석된 데이터 프로버넌스는 텍스트 인코딩을 통해 숫자 데이터로 변환한다. 패턴 압축 모듈에서는 데이터 프로버넌스에서 서브 그래프를 생성한 뒤 서브 그래프의 반복 패턴을 분석하여 추출한다. 반복 패턴 추출 시 일정 수치 값 이상 나온 패턴을 추출한다. 마지막으로 추출된 패턴을 기반으로 기준 패턴을 생성한다. 최종 압축된 결과는 기준 패턴을 통해 저장된다.

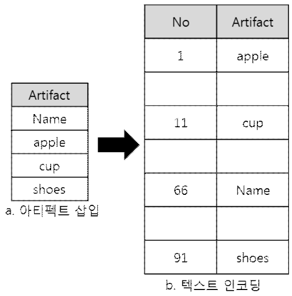


▶▶ 그림 1. 제안하는 기법의 압축 과정

2. 텍스트 인코딩

데이터 프로버넌스는 원본 데이터에 비해 수십 배에 달하는 대용량 데이터이고 이 또한 문자열 데이터로 이루어져 있다. 실제 데이터를 문자열 데이터로 저장할 경우 많은 공간을 차지한다. 그래서 이를 해결하기 위해 텍스트 인코딩을 통해 문자열 데이터를 숫자 데이터로 변경한다. 텍스트 인코딩은 아티펙트와 에이전트, 프로세스 모두 적용한다.

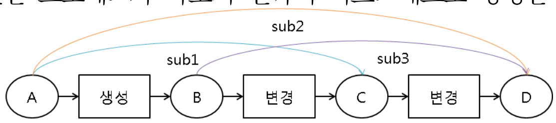
처음 텍스트 인코딩은 알파벳 순서로 하며 인덱싱은 일정한 간격으로 한다. 만약 추가적인 데이터가 삽입되어 기존의 인덱싱 번호가 가득 찰 경우 새로운 인덱싱 번호를 검색한다. [그림 2]는 텍스트 인코딩을 적용한 예이다.



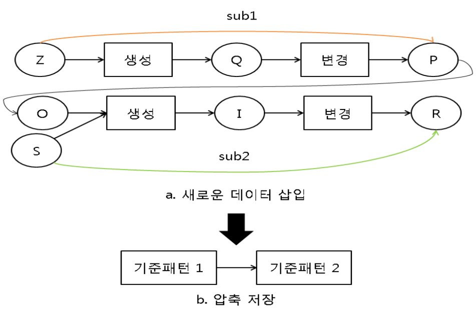
▶▶ 그림 2. 텍스트 인코딩

3. 패턴 압축

데이터 프로버넌스를 처리하는 패턴은 동일하게 반복되는 경우가 많다. 예를 들어, 문서 사용의 패턴을 보면 그 문서를 생성한 후 사용자들이 사용하다가 필요한 부분을 변경 하는 등 여러 가지의 문서에 대해 유사한 사용패턴을 보인다. 본 논문에서는 이를 이용하여 동일한 프로세스가 반복될 경우 서브그래프를 생성한다. [그림 3]는 데이터 프로버넌스 그래프에서 서브그래프를 추출하는 과정을 나타낸다. 이때 동일한 패턴의 프로세스 처리를 서브그래프로 추출하는 과정은 프로세스 단위로 동일한 프로세스가 나오기 전까지 서브그래프로 생성한다.



▶▶ 그림 3. 서브 그래프 생성



▶▶ 그림 4. 패턴 압축

[그림 4]는 제안하는 기법에 따라 패턴 압축된 과정을 나타낸다. 제안하는 기법에서는 추출된 서브그래프 중 동일한 서브그래프가 일정 수치 값 이상이 나오면 반복되는 서브그래프를 기준 패턴으로 하여 압축한다. [그림 4]에서와 같은 sub1과 sub2가 동일하므로 기준 패턴으로 생성 되며 [표 1]과 같이 스트링 데이터로 변환되어 저장된다. 최종 결과는 기준패턴으로 변환된 노드로 저장하여 데이터 프로버넌스의그래프를 압축한다.

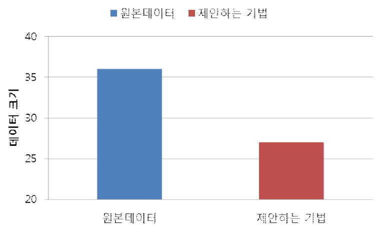
표 1. 기준 패턴 변환

기준패턴 1	기준 패턴 2
Z/Q/P	O,S/I/R

Ⅲ. 성능평가

본 논문에서는 제안하는 기법의 우수성을 보이기 위해 성능평가를 수행하였다. 성능평가에서는 원본데이터의 압축율를 확인한다. 성능평가 환경은 Intel core i5-4440 CPU는 3.10GHZ, 메모리는 4GB를 가지는 시스템에서 JAVA 8.0으로 구현하였다.

[그림 5]는 원본데이터의 크기와 제안하는 기법을 적용하여 원본 데이터를 압축한 데이터의 크기를 보여준다. 성능평가 결과는 제안하는 기법을 적용한 데이터의 크기가 실제 원본 데이터에 비해 35%이상 감소한다.



▶▶ 그림 5. 제안하는 기법의 압축률

V. 결론

본 논문에서는 데이터 프로버넌스를 효율적으로 관리하기 위한 저장 모델을 제안했다. 제안하는 모델은 Open Provenance Model을 이용하여 사전에 기반한 인코딩을 수행하고 반복되는 프로세스는 기준 패턴으로 만들어 중복되는 부분을 감소시켰다. 성능평가 결과 원본 데이터의 크기를 35%이상 줄였다. 향후 연구로 질의의 정확성과 검색성능을 비교하여 제안하는 기법의 우수성을 입증할 것이다.

▌참 고 문 헌▐

[1] A. Chapman, H. V. Jagadish, and P. Ramanan, “Efficient provenance storage”, Proc. SIGMOD Conference, pp.993-1006, 2008

[2] Y. Xie, K. Muniswamy-Reddy, D. Feng, Y. Li, and D. D. E. Long, “Evaluation of a hybrid approach for efficient provenance storage,” TOS, 9(4), pp.14, 2013

[3] P. Groth, S. Miles, S. Munroe, S. Jiang, V. Tan, J. Ibbotson, and L. Moreau. “The open provenance specification”, Technical report, 2006.