한국컴퓨터정보학회 하계학술대회 논문집 제24권 제2호 (2016. 7)

# 대용량 RDF 데이터의 효율적인 저장방법과 SPARQL 기반 검색방안 연구

오상윤*, 박지훈o
o*아주대학교 소프트웨어학과
e-mail: syoh@ajou.ac.kr* jihun1357@ajou.ac.kro

# Efficient Storing and SPARQL Search Scheme for Large Scale RDF Data

Sangyoon Oh*, Ji-Hoon Parko,
o*Dept. of Software, Ajou University

● 요 약 ●

시멘틱웹을 구축하는 표준언어인 RDF (Resource Description Framework)는 언어의 그래프 기반 특성으로 인해 일반적인 방식들로는 효과적인 저장과 추출이 어렵다. 더욱이 대용량 RDF 데이터의 저장과 추출에는 성능문제가 더욱 커지므로 많은 연구들이 이루어지고 있다. 본 논문에서는 SPARQL을 지원하면서 RDF 파일들을 효과적으로 저장하고 검색할 수 있는 저장방식에 대해 연구한 결과를 제시한다. RDF 데이터를 전처리를 통해 RDF의 트리플(주어:subject, 술어:property, 목적어:Object)에서 중복되는 주어(S)나 목적어(O)를 묶고, 사용자가 SPARQL 형식으로 검색했을 때 이용자가 주어부분을 변수로 두었는지 아니면 서술어 부분을 변수로 두어 찾는지에 따라 검색어와 유사한 단어 클러스터를 찾아준다. 동일 단어에 대해 여러 번 검색되던 부분을 한 번 검색으로 처리할 수 있기 때문에 효율이 높아진다.

**키워드:** RDF(Resource Description Framework), 시멘틱웹(semantic web), SPARQL(SPARQL), 클러스터링(Clustering)

## I. Introduction

인터넷의 급속한 발전으로 방대한 양의 데이터가 발생되고 있다. 따라서 웹 기반으로 발생하는 데이터를 효율적으로 저장하고 관리하기 위한 목적으로 컴퓨터가 쉽게 이해하고 처리할 수 있는 형태로 데이터를 구조화 시켜 저장하는 시멘틱웹이 제안되었다. 시멘틱웹의 표준언어로 W3C에서 제안한 RDF가 사용되고 있는데, RDF는 웹 상의 논리적인 관계를 표현할 수 있도록 주어(Subject), 술어(Predicate), 목적어(Object)와 같은 트리플(Triple) 구조를 사용한다. 여기서 주어 부분과 술어 부분은 URI 형태로 기술된다[1]. 방대한 양의 웹데이터 중에서 자신이 원하는 데이터만 골라내기 위해서는 기존에 RDBMS에서 SQL로 Query를 사용하듯이, RDF는 웹 상에 존재하는 RDF 데이터들로부터 원하는 데이터를 찾기 위해 SPARQL이라는 Query 형태를 이용한다. 본 논문에서는 대용량 RDF를 효과적으로 분산 저장하고 SPARQL을 지원하는 방식들을 조사하고, 저장 방식들이 Query에 성능에 어떠한 관계가 있는지 알아보며, 전처리 등을 통해 저장 및 검색처리 성능을 향상시키는 방안을 연구한다.

본 논문의 구성은 다음과 같다. 2장에서는 RDF의 구조 및 SPARQL에 관련연구를 설명한다. 그리고 3장에서는 검색의 속도를 높이기 위한 방법을 제안했다. 그리고 4장 결론에서는 개선하기 위한 결과값을 보여주고 이를 설명한다.

## II. Preliminaries

### 1. RDF 시스템 및 SPARQL

#### 1.1 시멘틱웹(Semanticweb)

Tim Berners-Lee 가 제안한 WWW(World Wide Web)은 URI 형태를 통해 정보를 찾아가는 것은 가능하였으나, 구조화를 통해 데이터베이스처럼 사용하는 것에는 한계가 있었다. 따라서 2001년에 Tim-Berbers Lee가 검색과 추론을 지원하는 시멘틱웹을 기존 자신이 제안한 WWW 위에서 동작하도록 제안하였다 [6]

이러한 배경에 따라 데이터 교환을 표현하기 좋은 언어인 XML(Extensible Markup Language)을 기반으로 링크의 관계를 더 표현하기 위한 언어로서 RDF가 새롭게 제안되었다.

#### 1.2 RDF( Resource Description Framework)

W3C에서 RDF 정의를 살펴보면 웹에 있는 Resource에 대한

정보를 표현하기 위안 언어이며 특히 웹 Resource의 Metadata를 표현하기 위한 언어이다[2]. 이러한 RDF 구조는 트리플 형태로 주어(Subject), 술어(Predicate), 목적어(Object)로 구성되어있다. 여기서 주어부분은 URI 또는 변수, 목적어 부분은 변수나 그대로의 리터럴 방식, 서술어 부분은 속성 값을 나타낸다[1]. 보통 일대일 대응처럼 되어있는데 여기서 목적어 부분이 다른 속성이거나 여러 가지 주어와 겹치는 부분이 있다면 1대1 관계 외에도 Fig.1과 같이 1대다 형태로 나타낼 수 있다.

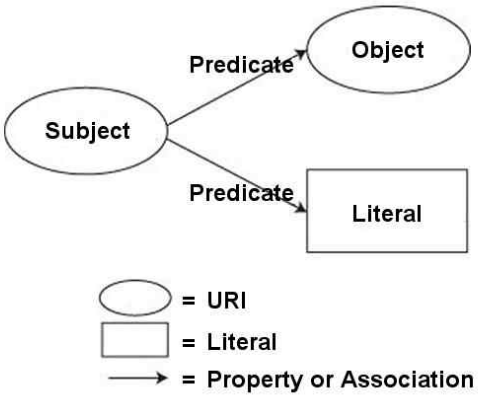


Fig 1. RDF Data Structure [5]

### 1.3 SPARQL

SPARQL은 관계형 Database에 저장된 데이터로부터 원하는 정보를 꺼내기 위해 SQL에 Query를 활용하듯이 웹에 공개된 RDF 데이터들로부터 원하는 데이터를 꺼내오기 위하서 사용되는 Query방식이다. SPARQL Query는 중요한 부분 3가지는 다음과 같다. 먼저 PREFIX에는 어디에서 찾을 것인지 URI를 병칭으로 지정 검색될 위치의 URI를 적는다. 그리고 SELECT에는 답을 원하는 타켓을 표현할 변수로 '?+변수로 표시하고 WHERE에서는 그 순서대로 주어, 술어, 목적어 순으로 Query의 조건을 넣는다. 간단하게 여러 개의 분산 환경에서 정보교환을 하는 형태이다[2].

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject
WHERE {
?subject rdfs:label "Hadoop"
```

Fig 2. An Example of SPARQ

예를 들어 Fig.2 의 SPARQL 정보는http://dbpedia.org/resource/ 내에서 http://www.w3.org/2000/01/rdf-schema#에 있는 rdfs: 술어 정보를 바탕으로 "Hadoop" 목적어를 가진 label을 subject 변수에 나타내겠다는 의미이다[3,4].

## III. The Proposed Scheme

RDF 데이터를 검색하기 위해서는 저장되어 있는 데이터를 일대일 방식으로 모두 대조해 봐야 하기 때문에 많은 시간이 걸린다. 데이터 검색에 많은 조건을 두거나 URI의 적절한 범위를 제한해주지 않는다면 검색이 되지 않는 경우도 많이 발생한다. 그리고 대용량 RDF 데이터의 경우 대상 데이터의 크기로 인해 발생하는 검색처리 속도가 매우 느리기 때문에 효율적인 검색하는 방법을 연구한다면 이 결과는 유의미한 결과가 될 것이다.

제안하는 저장 및 검색 방법은 다음과 같다. 전처리 단계에서 원본 RDF 그래프의 묶음과정을 수행한다. 검색 대상이 되는 RDF 데이터의 크기를 줄이기 위해 원본 RDF 그래프에서 이름이 비슷하고 관련 있는 node를 묶어 clustering 된 그래프를 생성한다.

전처리 과정에서 그래프를 묶기 위해서는 중복되는 주어(Subject)끼리 혹은 서술어(Object)끼리 같은 용어나 URI일 경우의 Node를 Clustering한다. 실제로 RDF 데이터들을 살펴보면 같은 주어나 목적어가 따로 분리되어 중복되어 저장되는 경우도 많다. 중복되어 저장하는 되면 그만큼 검색에 불필요한 시간이 낭비된다. 그렇기 때문에 RDF의 Subject 부분이나 Object가 비슷한 부분의 node를 Clustering 하는 것이다.

비슷한 이름끼리 전 처리된 RDF 그래프를 바탕으로 사용자에게 SPARQL문을 입력 받아 찾기 위한 변수 주어 혹은 목적어 부분 중에 비슷한 부분을 사용자에게 보여주는 시스템이다. 묶어있는 자료에서 찾기 때문에 2차 검색이 있지만, 검색 시간이 줄어들기 때문에 유용할 것이다.

## IV. Experiment Results and Conclusions

본 실험은 제안 전처리과정을 적용한 방식과 기존 방식을 비교하여, 제안 방식의 우수성을 보이고자 하였다. 실험에서는 같은 데이터에 대해, 질의응답 시간이(ms 단위로) 작을수록 좋은 성능이라고 가정하였다. 클러스터링을 하지 않은 대조군으로 RDF 자료를 검색할 때 주로 쓰이고 있는 Apache Jena를 사용했다. 실험을 할 때 RDF 전처리 클러스터링 과정을 직접 하기 위해 Fig. 3과 같은 정보를 담은 작은 용량(8 triples)의 RDF 파일을 수정했다. 실험결과는 Table 1에 정리하였다.

```
Problems  @ Javadoc  Declaration  Console
<terminated> main [Java Application] C:₩Program Files₩Java₩
The database contain for:
  Num3 Smith
  Num2 Smith
  Sarah Jones
  Num5 Smith
  Num1 Fn1
  John Smith
  Sarah Smith
  Becky Smith
  Sarah Fn1
  Matt Jones

소요시간 : 499ms
```

Fig 3. RDF Data Sets for Experiments

Table 1. RDF Query Experiment Results

| | Set 1 | Set 2 |
|---|---|---|
| Naive (Jena) | 490ms | 470ms |
| Proposed (전처리) | 499ms | 570ms |

위 Table 1에서 확인할 수 있듯이 실험 RDF의 크기가 작아서, 클러스터링(전처리)한 Proposed 부분이 조금이나마 성능이 높았다. 데이터 사이즈가 작아 추가 연구 및 실험을 통해 대용량 RDF로 처리하는 실험을 한다면 성능차이가 명확할 것으로 생각한다.

## ACKNOWLEDGEMENT

"본 연구는 미래창조과학부 및 정보통신기술진흥센터의 SW중심대학 지원사업의 연구결과로 수행되었음"

"This research was supported by the MISP(Ministry of Science, ICT & Future Planning), Korea, under National program for Excellence in Software supervised by the IITP(Institute for Information & communications Technology Promotion)"

## References

[1] An Hyeokju, Lee Seonghui, Kim Haksu, Design of a Question-answering System Based on SPARQL, p.154 , 2014.

[2] Batselem Jagvaral, Lee Wangon, Kim Gangpil, Park Yeongtaek SPARQL Query Processing in Distributed In-Memory System, p1112, 2015.

[3] Mun Hyeon Jeong, Sung Jung Hwan, Kim Young Ji, A Desin and Implementation of Efficient Storge Structure for a Large RDF Data Processing, p.255~262 2007

[4] Park Dongjin, A Study on the Metadata Modeling for Result Informatin Using RDF, p.293, 2007

[5] Andreas Gerlach, "Web of Data," http://www.appelgriebsch.org/Web-Of-Data/