A Review and Design of Framework for Storing and Querying RDF Data using NoSQL Database

Chanuwas Aswamenakul[1], Marut Buranarach[2],
and Kanda Runapongsa Saikaew[1*]

[1] Department of Computer Engineering, Faculty of Engineering,
Khon Kaen University, Khon Kaen, Thailand
chanuwas.a@kkumail.com, krunapon@kku.ac.th
[2] Language and Semantic Technology Laboratory
National Electronics and Computer Technology Center (NECTEC), Pathumthani, Thailand
marut.bur@nectec.or.th

Abstract. This paper reviews existing systems and describes a design of RDF database system that uses NoSQL database to store the data which aims to enhance performance of the Semantic Web applications. RDF data is a standard of data in the form of Subject-Predicate-Object called Triples and stored in database called Triple Store. Typically RDF database system uses SPARQL query language to query the RDF data from Triple Store database, e.g. Jena TDB. Our design of RDF database system uses NoSQL database, i.e.,MongoDB, to store the data in JSON-LD format and query by using query API of NoSQL database. We will use the Berlin SPARQL Benchmark to compare the performance of Triple Store and NoSQL systems.

Keywords: Semantic Web application framework, RDF database, NoSQL

1 Introduction

Currently the amount of data has increased excessively with a variety of formats. The Semantic Web technology aims to provide standards and facilitate analyzing such big data. The Semantic Web uses RDF data to describe the data on the web in form of Subject-Predicate-Object called "triples" [1] that makes the data to have the standard data model.

In the present, there are many approaches to store and query RDF data. One approach to store RDF data is Triple Store designed for storing the triples format of RDF data [2] and queried by using SPARQL query language. However, from the Berlin Benchmark results [3], Triple Stores show poor performance when compared to the relational database systems. NoSQL database removes some features of relational databases and uses other data models to improve the performance of database. This has motivated many works to store RDF data by using NoSQL database.

This paper reviews existing systems and designs a framework to store RDF data in NoSQL database. One of the main goals is to design a Semantic Web application framework that uses RDF data with NoSQL database, i.e., MongoDB. The ultimate

* Corresponding author

objective is to provide a better support for researchers in developing the Semantic Web applications.

2 Review of NoSQL-based RDF Database

This section reviews some of RDF database systems that use NoSQL to store the RDF data including Neo4j [4] , AllegroGraph [5] , H2RDF [6] , Oracle NoSQL [7] , MonetDB [8] and CumulusRDF [9]. The comparison is based on some criteria of database software such as Implementation language, Database Model, SPARQL1.0, SPARQL1.1, Trigger, Transaction Concept, Secondary Index, Consistency Concept, Partitioning Method, Replication Method, Concurrency, Map Reduce, Durability and Security. Table 1 provides a review summary of RDF database systems that use NoSQL database.

Table 1. Review summary of RDF database systems that use NoSQL database

Name	Neo4j	AllegroGraph	H2RDF	Oracle NoSQL	MonetDB	CumulusRDF
Implementation language	Java	Common Lisp	Java	Java	C	Java
Database Model	Graph Database	Graph Database, Document store Database	Column Store Database	Key-Value Database	Column Store Database	Column Store Database
SPARQL 1.0	Yes	Yes	Yes	Yes	Yes	Yes
SPARQL 1.1	Yes	Yes	Yes	Yes	No	Yes
Trigger	Yes	No	Yes	No	Yes	Yes
Transaction Concept	ACID	ACID	Configure ACID + Visibility	ACID	ACID	Configure ACID(Lightweight Transaction)
Secondary Index	Yes	Yes	Yes	No	Yes	Yes
Consistency Concept	Eventual consistency	Strong consistency	Strong consistency	Several consistency policies	Strong consistentcy	Tunable consistency
Partitioning method	Cache Sharding	Sharding	Sharding	Sharding	None	Sharding
Replication method	Master-slave	Master-slave	Master-slave	Master-slave	None	Selectable replication factor
Concurrency	Yes	Yes	Yes	Yes	Yes	Yes
MapReduce	No	No	Yes	Yes	Yes	Yes
Durability	Yes	Yes	Yes	Yes	Yes	Yes
Security	Security Rule	Filter per User and/or Role	Access Control List (ACL)	User and Role Permission	fixed user and password by admin	Object Permission

3 Framework Design

This section describes our design for an application framework representing system architecture that compares the Triple Store-based implementation with the NoSQL-based implementation. We also provide query translation that represents some example translation of basic SPARQL queries adapted from the Berlin Benchmark [3] to MongoDB queries.

In a system architecture based on the OAM framework [10], we compare between Triple Store based implementation and NoSQL based implementation. The Triple store based implementation uses Jena TDB to store the RDF data and OAM API that uses SPARQL to query the data from Jena TDB. In NoSQL based implementation, we use RDF to JSON-LD Converter to convert RDF data format to JSON-LD format, which is JSON-based format designed for Linked data [11], and use JSON-LD Parser to parse and import JSON-LD data to MongoDB. The OAM API then uses MongoDB query API to query the data from MongoDB.

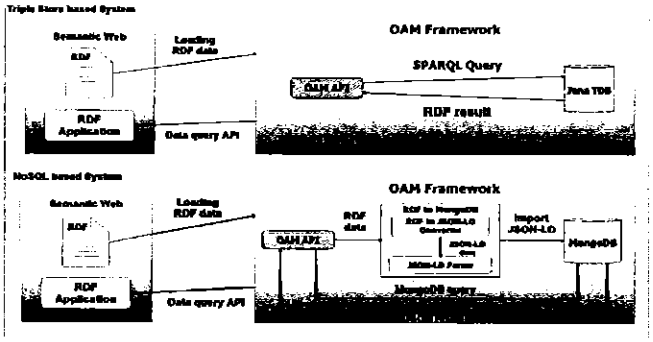

Fig. 1. Architecture of the OAM framework using Triple Store vs. NoSQL RDF database

Table 2 illustrates some query translation based on the Berlin SPARQL benchmark. In Table 2, query 1 shows an example of query using FILTER, ORDER and LIMIT. Query 2 shows an example of query using OPTIONAL. Query 3 shows an example of query using regular expression.

Table 2. Sample query translation based on the Berlin SPARQL Benchmark

Query Description	SPARQL	MongoDB query
1. Find products for given product type and value of property numeric1 must be greater than 318 then results ordered by value of label and limit number of results by 10.	SELECT ?product ?label WHERE {?product label ?label ?product a ProductType56 ?product PropertyNumeric1 ?value FILTER (?value > 318) } ORDER BY ?label LIMIT 10	db.collection.find({label : {$exists : true}, types : 'ProductType56', PropertyNumeric : {$gt : 318}} ,{label : 1}).sort({label : 1}).limit(10)
2. Retrieve the basic information of products and products may not have property numeric2 (OPTIONAL in SPARQL).	SELECT ?label ?comment ?propertyTextual1 ?propertyNumeric2 WHERE {Product127 label ?label Product17 comment ?comment Product1277 PropertyTextual1 ?propertyTextual1 OPTIONAL { Product1277 PropertyNumeric2 ?propertyNumeric2 } }	db.collection.find({_id : 'Product1277', label : {$exists : true}, comment : {$exists : true}, PropertyTextual : {$exists : true}} , {_id : 0, label : 1, comment : 1 , PropertyTextual1 : 1 , PropertyNumeric2 : 1})
3. Find products having a label that contain given string by using regular expression.	Select ?product ?label where { ?product label ?label ?product type Product FILTER regex(?label, "dung")}	db.collection.find({label : {$regex : 'dungs'} , '@type' : 'Product'} , {label : 1})

4 Conclusions and Future Work

This paper has proposed the design of RDF database system by using MongoDB to store the data in JSON-LD format and its query API. In the future, we will conduct the performance comparison of Triple Store, MongoDB RDF Database, and relational database using the Berlin SPARQL Benchmark. Several techniques will be investigated to improve the performance of the MongoDB RDF Database.

Acknowledgement

The financial support from Young Scientist and Technologist Programme, NSTDA (YSTP: SP-56-NT03) is gratefully acknowledged.

References

1. RDF [Online]. Available: http://www.w3.org/RDF/

2. Triple Store [Online]. Available: http://www.w3.org/wiki/RdfStoreBenchmarking

3. Bizer, C., Schultz, A.: The berlin sparql benchmark. International Journal on Semantic Web and Information Systems (IJSWIS) 5(2), 1–24 (2009).

4. Neo4j [Online]. Available: http://docs.neo4j.org/chunked/2.0.4/

5. AllegroGraph [Online]. Available: http://franz.com/agraph/allegrograph/
6. Papailiou, N., Konstantinou, I., Tsoumakos, D., Koziris, N.: H2RDF: Adaptive Query Processing on RDF Data in the Cloud. In WWW, 2012.

7. Oracle NoSQL database [Online]. Available: http://docs.oracle.com/cd/E26161_02/html/RDFGraph/

8. MonetDB [Online]. Available: https://www.monetdb.org/Home

9. Cudré-Mauroux, P., Enchev, I., Fundatureanu, S., Groth, P. T., Haque, A., Harth, A., Keppmann, F. L., Miranker, D. P., Sequeda, J. & Wylot, M. (2013), NoSQL Databases for RDF: An Empirical Evaluation. International Semantic Web Conference (2) , Springer, pp. 310-325 .

10. Buranarach, M., Thein, Y., Supnithi, T.: A Community-Driven Approach to Development of an Ontology-Based Application Management Framework. In: Takeda, H., Qu, Y., Mizoguchi, R., and Kitamura, Y. (eds.) Semantic Technology. pp. 306–312. Springer Berlin Heidelberg (2013).

11. JSON-LD [Online]. Available: http://json-ld.org/