XML and NoSQL DBMS: Migration and Benchmarking

Author:
Prakash Thapa

Prof. Dr. Marc H. Scholl
Prof. Dr. Daniel Keim

Christian Grüen

Konstanz University

November 11, 2014

Abstract

XML and NoSQL database are two growing field in second generation database system, They share some similarities as well as they have some significant difference. This thesis focus on the comparative analysis of these two database system based on the Use cases and existing solution, we will discuss the data processing, query pattern and Information Retrieval(IR)

....

Zusammenfassung(German Abstract)

XML und NoSQL

Acknowledgments

The completion of this master thesis would not have been possible without the support of many people.

Contents

1 Introduction

1.1 Motivation

Few years of time XML [1] was *de facto* data exchange format which enabled people to do previously not that easy thing that time like exchange of content of Microsoft's office documents exchange through HTTP connections. But in recent years a bold transformation has been a foot in the world of Data exchange. The more light weight, less bandwidth consumer JSON(JavaScript Object Notation)[1] has been emerge not just as an alternative to the XML but as rather as potential full Blown successor[2]. Even though these two format has their own pros and cons, the rise of JSON as key in data exchange format, new database technologies so called NoSQL are also emerges and getting success in their own way. The rate of new research papers in these system are increasing in recent years.

1.2 Contribution

The main contribution of this thesis is that it provide the necessary techniques and algorithms migration of data from XML database to NoSQL databases. More specifically, It will focus on Document store databases MongoDB, Couchbase and Rethinkdb. To complete this task it is necessary to understand general architecture and data model of each of these database as well and the Information Retrieval(IR). At the second part, conversion of Queries in XML data to individual NoSQL database is also .....

.....

1.3 Overview

This thesis is divided into three main sections. The first section define the Techniques and necessary algorithms to convert XML to JSON Data format. The second section will Systems and scope of work. On third section, we see the performance and comparative analysis of each of these systems. The work is structured as follows: Chapter 2: Chapter 3: Chapter 4: Chapter 5:

2 Preliminaries

2.1 Semi-structured data

2.1.1 XML and JSON

2.2 Mapping

2.2.1 Friendly

2.2.2 Unfriendly

2.2.3 Array and Object

2.2.4 Mapping approaches

2.2.5 Summary

2.3 XML Database

2.3.1 XML Query Language

2.4 NoSQL database

2.4.1 Key/Value storage

2.5 document oriented database

2.5.1 Querying NoSQL database

3 Related work

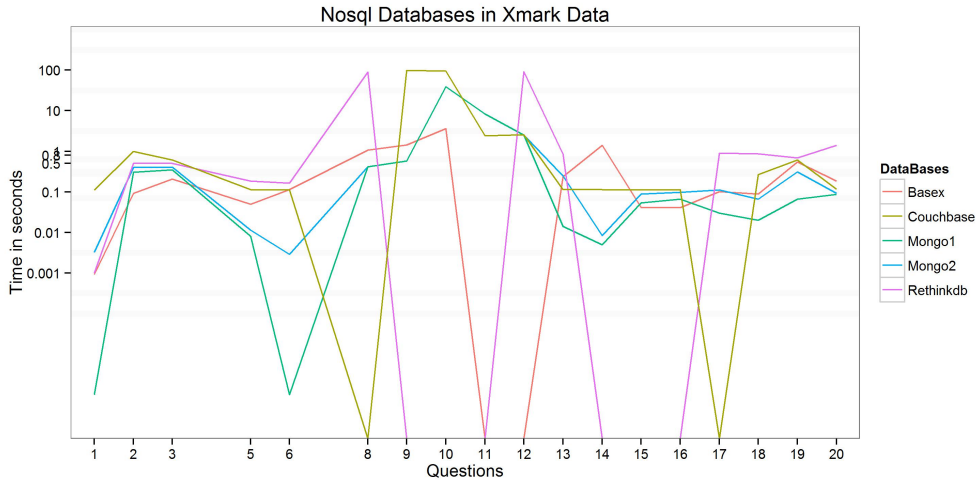


Figure 1: An overview of some important indexing structures developed over years

4 System/Environment

4.1 BaseX

Code 1: A simple KML example representing a Point

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<document>
<placemark>
  <name>New York City</name>
  <description>New York City</description>
  <point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </point>
</placemark>
</document>
</kml>
```

numbers

Code 2: JSON Data

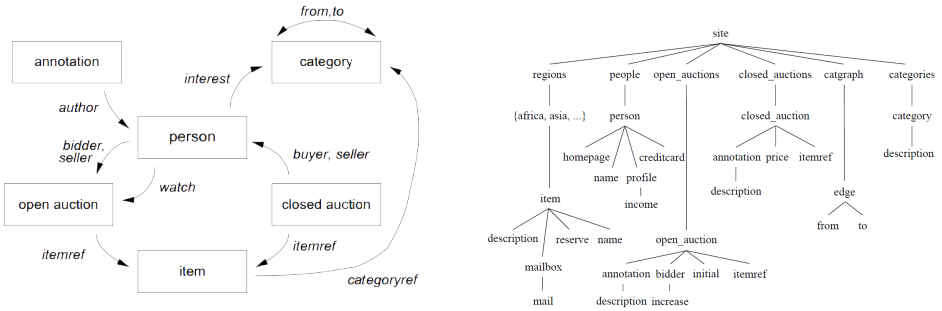
5 Performance/Experiments

5.1 XMark

The XML benchmarking project XMARK [2] dataset is single record with large and complex tree structure. It is one of the most popular and most commonly used XML Benchmark [3]. It uses a small executable tool called *xmlgen* that enables to create a synthetic XML Dataset according to fixed DTD of an Internet auction database. The xmlgen produces the dataset that is platform independent and accurately scalable ranging from a minimal document to any arbitrary size limited by the capacity of the system.

5.1.1 Dataset

XMARK dataset is single record with huge and complicated tree structure [4].



(a) Reference in *XMark* dataset tree Fig [2]

(b) Reference in *XMark* dataset tree Fig [2]

Figure 2: XMARK data tree and reference

figure need to create

5.1.2 Queries

coming...

5.2 Evaluation of test devices

5.3 XMark data into NoSQL Database

A synthetic XMARK dataset consist of one(huge) record in tree structure [4]. However, As mentioned in 5.1, each subtree in schema, items, person, open_auction , closed_auction etc contains a large number of instances in the database which are indexed in it's own. In most of NoSQL system, this scenario is different, each instance has it's own index structure, the dataset cannot be in just a huge block.. As the data modeling of NoSQL do not match this single structure-encoded sequence, we breakdown it's tree structure into set of sub structure without losing the overall data and create index for each of them. As the data modeling from one NoSQL system is different to another unlike most of the XML databases, which have more similarcomparison structure than NoSQL. So we need to define modeling for each of those database separately.

This should be elaborated, *language should be imporve*

???

5.3.1 MongoDB

Mongodb uses the concept of *collections* and *documents* to model data. collections are the grouping of documents which generally have similar schemas. Data in Mongodb has flexible schema where collections do not enforce document structure rather requirements of our application. . Documents are modeled as a data structure following the JSON format which is composed of key and value pair. There are two principle that allow application to represent documents and their relationship: *reference* and *embedded documents*.

A collection is analogous to collection in XML database????

Reference Reference store the relationships between data by including links and references from one document to another as in Figure 3(a). The application can resolve these reference to access the related data

need to change following image/json according to xmark data

Mongodb data model pg4

(a) Reference document (b) Embedded document

Figure 3: Mongodb document structure

Embedded Embedded documents captures relationships between the data by storing related data in a single document structure. The documents in this method are structured as sub-documentsin the in the form of Array or/and Object [5].

...??

Indexing Each document in Mongodb is uniquely identified by a field *_id* which is a primary index. Hence the collection is sorted by _id by default. [5]

but note that this primary key index is not a clustered index in Database terms. I.e. the index entries only contains pointers to actual documents in the MongoDB data files. Documents are not physically stored in the order of _id on disks.?? ::expand...page:9

5.3.1.1 XMARK in Mongo

numbers

Code 3: Mongodb data representation of XMARk data

```
{
        "_id": "person0",
        "doctype": "people",
        "name": "Kasidit Treweek",
        "emailaddress": "mailto:Treweek@cohera.com",
        "phone": "+0 (645) 43954155",
        "homepage": "http://www.cohera.com/~Treweek",
        "creditcard": "9941 9701 2489 4716",
        "profile": {
                "income": 20186.59,
                "interest": [{
                        "category": "category251"
                }],
                "education": "Graduate School",
                "business": "No"
        }
        ....
}
```

Code 4: XMARK data with of *person0*

```
<person id="person0">
<name>Kasidit Treweek</name>
<emailaddress>mailto:Treweek@cohera.com</emailaddress>
<phone>+0 (645) 43954155</phone>
<homepage>http://www.cohera.com/~Treweek</homepage>
<creditcard>9941 9701 2489 4716</creditcard>
<profile income="20186.59">
```

```
        <interest category="category251" />
        <education>Graduate School</education>
        <business>No</business>
        </profile>
        ...
        </person>
```

5.3.1.2 Queries

5.3.2 Couchbase

5.3.3 Rethinkdb

5.4 Benchmarking

5.5 Summary

6 Discussion

7 Conclusion

8 Future Work

storing in the memory

References

[1] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, et al. Extensible Markup Language (XML) 1.0 (Fifth Edition). http://www.w3.org/TR/xml, November 2008.

[2] Albrecht Schmidt, Florian Waas, Martin Kersten, Michael J Carey, Ioana Manolescu, and Ralph Busse. Xmark: A benchmark for xml data management. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 974–985. VLDB Endowment, 2002.

[3] Irena Mlỳnková. Xml benchmarking: Limitations and opportunities. Technical report.

[4] Haixun Wang, Sanghyun Park, Wei Fan, and Philip S. Yu. Vist: A dynamic index method for querying xml data by tree structures. In *In SIGMOD*, pages 110–121, 2003.

[5] Xiaoming Gao. Investigation and comparison of distributed nosql database systems.

List of Figures

List of Tables

Listings