Using MongoDB to Implement Textbook Management System instead of MySQL

Zhu Wei-ping
Computer and Engineering Department, CDAVTC
Chengdu China
email:zwp_xz@163.com

Li Ming-xin
Computer and Engineering Department, CDAVTC
Chengdu China
email:starlee2008@163.com

Chen Huan
YiBin Vocational and Technical College
YiBin China
email:ybzychen@163.com

Abstract: **With the development of internet Web2.0 technology, the traditional relational database is widely used in information management system. However, it is not effective, when we need to query a wide range of massive data, especially with multi-table join queries. Now, a kind of new technology emerged----NoSQL, which is non-relational database management system with format loose data storage, not support the join operation, the effective query capability etc advantages. This paper attempts to use NoSQL database to replace the relational database, applied to traditional information management systems, compare the two database technologies, give the key code of NoSQL implementation, and finally list the performance comparison of two schemes.**

Keywords: *MongoDB;MySQL; RDBMS; NoSQL*

I. INTRODUCTION

Traditional relational database use two-dimensional table to represent data, with strict consistency of database transactions, real-time to read and write, can implement complex SQL queries, especially multi-table related query. However, for a wide range of massive data queries, multi-table query is not effective. NoSQL is not a relational database management system. This system uses a non-relational model of data structure. Usually it does't support the join operation, while the query is efficiency. This paper attempts to use this new database technology in the textbooks management system, to implement high-performance data queries. In this paper, the relational database using MySQL. As the most popular open source database in the world, MySQL has the following features: small, fast, low cost [1]. NoSQL database system use MongoDB, which is document oriented and characterized by mass data storage, at the same time with good query performance [2].

Section II of the paper we introduce the concepts of non-relational database and the characteristics of MongoDB. The implementation of textbooks management system based on MongoDB will be described in section III. The fourth section we compare the performance of textbooks management system based on MySQL and MongoDB. In the fifth part, we point out some drawbacks of the NoSQL implementation and future work to do.

II. NON-RELATIONAL DATABASE AND MONGODB

NoSQL means non-relational database. The data structure of non-relational database is not fixed. There are key (key-value) storage type, document storage type and so on. MongoDB is a document oriented database. It's schema-free, which contains Database, Collection, and Document. One Database can have multiple Collections. Each Collection is a collection of Documents. The structure of MongoDB system is displayed in Figure 1. Collection can be created at any time, without predefined. It can also contains records with different schema documents, which means one record of a document has 3 attributes, and the next record in the document may has 10 attributes. The type of the property can be any basic data types, such as numbers, strings, dates, etc, or an array or hash, and even a sub-document. This can realize de-normalized data model and improve query speed [3]. The textbooks management system which is based on MongoDB will be used in this data model.
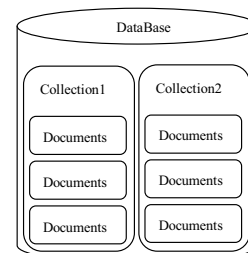

Figure 1 MongoDB document-oriented database

III. IMPLEMENTATION SCHEME

A. Textbooks Management System Data Model

a) Traditional relational data model

The system use traditional data model is based on MySQL that mainly contain students, teachers, textbooks,

subscriptions textbook, textbook use, textbook storage, entry textbook, delivery textbook, and other entities. Students associate with Classes and Departments by foreign key. The E-R data model of the system is shown in Figure 2, displaying that deptCode is one to many relationship with the teacherInfo, textBook and teacherInfo is many to many relationship. If you want to query a certain teacher's department and using his/her textbooks of information, you need to associate deptCode, teacherInfo, teaUSEbook and textBook four tables. Associated with multi-table query, you have to spend lots of time. To solve this problem, in this paper we attempt to use NoSQL to solve it.
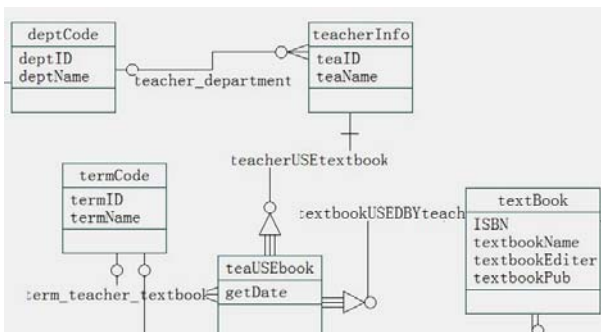

Figure 2 part of the relational data model of the system

b) NoSQL data model

Scheme free is one of the characteristics of NoSQL.According to system's requirements; two basic information, students' and teachers' information will be created in MongoDB. Subscription textbooks, used textbooks, textbooks storage, entry textbook, delivery textbook, can be set into the basic information of teacher with embedded documents in the Collection. But considering too many embedded documents are resulted in increasing size of one single document, we create a separate textbooks operating Collection, which is shown in Figure 3:
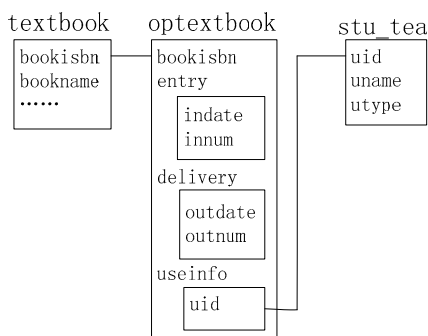

Figure 3 part of the system's data model based on MongoDB

B. System Implementation

In this system, using Java language to develop system, access and operate the database needs the driver that provided by MongoDB. The key step to fulfill as the following, first, using Java program to connect the database, and then insert the data, finally, get the results sets by query condition.

a) Establish a connection

In order to make Java programs connect with MongoDB database, you need to install the drivers and set the environment variables. Then import the development kit of MongoDB in the application program. After that you can easily connect to MongoDB database, the following are the key codes:

    Mongo m = new Mongo ("DBserver", 27017); // connect database with the database server name and port number
    DB db= m.getDB("dbname"); //open database

b) Insert the data

In the traditional JDBC development, Java program establish the connection with database firstly, then pass the SQL statements to program, at last insert the data into the database, which is process-oriented thinking. it is not benefit to program design and system maintenance. MongoDB database use object-oriented thinking to implement system, first, according "textbook" to create a DBCollection object coll, and then create a BasicDBObject object doc, add textbooks information to the object doc , and finally add the object doc to the object coll , at last insert data into MonoDB database.

    DBCollection coll = db.getCollection("textbook");
    BasicDBObject doc = new BasicDBObject();
    doc.put("bookisbn", "9787561829318");
    doc.put("bookname", " Oracle database Tutorial ");
    doc.put("bookpress", " Tianjin University Press");
    coll.insert(doc);

c) Query the data

1) Query a document. Use DBObject to indicate query conditions, or use nested multilayer to represent complex conditions, then it will return results for object of DBObject.

Query data in MongoDB database follow object-oriented thinking too. To create DBCollection object coll for determining the scope of the query by getting Collection "textbooks" in the database. Later create BasicDBObject object cond, and write the query condition into cond, finally put the object cond into the method findOne of the object coll, the query results return to DBObject object ret.

    DBCollection coll = db.getCollection("textbook");
    BasicDBObject cond = new BasicDBObject(); // Query matching conditions
    cond.put("bookisbn", "9787561829318");
    DBObject ret = coll.findOne(cond);

2) Query document collection, use DBCursor return a result set.

The following steps are the same as above mentioned. Put the object cond   into the method find of the object coll, the query results back to a cursor (DBCursor) object ret, which contains a set of objects DBObject. You can use the loop statement to process the data one by one.

    cond.put("bookprice", new BasicDBObject("$gt", 50));
// Price is greater than 50
    DBCursor ret = coll.find(cond);
    while (ret.hasNext()) {

//data processing ret.Next();
}

Above steps are the key to the implementation for the system. Comparing to using JDBC, MySQL solution, the system adopts the object-oriented method to implement, developers can easily develop and maintain. The following section will display the efficiency of two different implement methods.

IV. PERFORMANCE COMPARISON

In the performance testing, enter 100,000 textbooks information data into database. The cost time of MongoDB and MySQL were recorded as shown in figure 4. The left columns show the average time cost for inserting 100,000 textbooks information. The right columns show the average time cost for querying 2,000 records from 100,000 textbooks records .In order to facilitate the graphical display, set original data multiply Coefficient for 100. In the performance of inserting data, MongoDB spends less times than MySQL, which testified it is improving query efficiency.
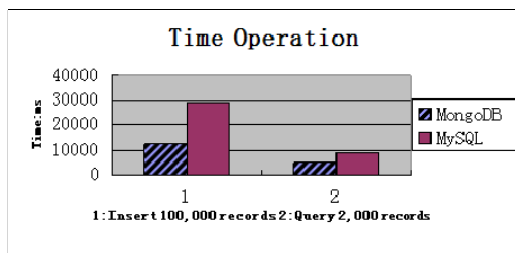

Figure 4 MySQL and MongoDB insert, query time

V. CONCLUSION

The system is developed based on MongoDB 1.6, which is compare to MySQL implement scheme, inserting and querying data by MongoDB. It has obvious advantages. However, in the development process also encountered the references of MongoDB which is less than MySQL. We have to spend much more time on problem-solving.and post-maintenance issues is not easier than MySQL. But the development of MongoDB in the application system will continue to research.

REFERENCES

[1] MySQL database development history http://www.lovoinfo.com/shujukujishu/2665.html

[2] A discussion of NoSQL database - why should non-relational database [OL]http://robbin.javaeye.com/blog/524977

[3] Pan Fan, From MySQL to MongoDB – Visual China's R oad of NoSQL China [J], Programmer, 201006, 79-81

[4] Huang Xian-li, NoSQL non-relational database development and application[J], Fujian PC, 201007,30,45

[5] Zhong Wang-wei, Huang Xiao-ou, Based on C / S and B / S mixed-mode library management system [J], Modern computer, 200708,124-126

[6] Bruce Eckel,Thinking in Java(Second Edition)[M], Machinery Industry Press, 2002

[7] Charles A.Bell, In-depth understanding of MySQL [M], Posts & Telecom Press, 2010

[8] Nicholas C.Zakas, JavaScript Advanced Programming （Second Edition）[M], Posts & Telecom Press, 2010

[9] Brett McLaughlin, Java 与 XML（Photocopy Edition）[M], Southeast University Press, 2007

[10] Lauriat.S.M., Ajax Architecture and Best Practices in depth [M],Posts & Telecom Press,2009