Application of NoSQL Database in Web Crawling

GU Yunhua, SHEN Shu, ZHENG Guansheng
College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
doi:10.4156/jdcta.vol5.issue6.31

Abstract

Web crawling is one of the most important applications of the Internet; the selection of database storage directly affects the performance of search engines. In the past few decades, the traditional relational databases almost monopolize all areas of the database applications. However, with the continuous development of web applications, they are facing the severe challenges. NoSQL database is the broad definition of non-relational data storage. This article gives the data storage structure and query of relational database and NoSQL database MongoDB to meteorological BBS information collection system, and the advantage and the disadvantage are listed in data structure, query and scalability. Compared to relational database, MongoDB supports schema-free, has great query performance with huge amount of data and provides easy horizontal scalability with low cost of hardware. It is more suitable for data storage in Web crawling.

Keywords: *NoSQL, MongoDB, Web Crawling*

1. Introduction

Web crawling is to filter and collect various information and resource from Internet, according to a certain theme, storage the information into the database, and construct the search engine for users. Because of the huge amount of information, the performance of search engine is mostly affected by storage form and the storage database. Web crawling requests great capacity of storage space and low hardware costs. Its requirements of consistency and integrity can be reduced, but need high availability, scalability and performance. The relational databases store data in the form of a two-dimensional table with strictly row and column format, and emphasize the consistency and integrity of data, thus the performance in the distributed data management has low efficient, resulting in poor horizontal scalability and high hardware costs when increasing the data storage [1-3]. With the increasing resources of information, the traditional relational database can no longer suffice requirement to query information [4,5]. Several famous search engines implement their own databases, such as Google's BigTable[6]. That has brought a new generation of database -- NoSQL database. This paper gives the solution of relational database and NoSQL to a certain meteorological BBS information collection system, and compares the advantage and disadvantage of them.

The rest of this paper is organized as follows: Section 2, we give a brief description of normal Web crawling system architecture and their convergence properties. The fundamental NoSQL database is discussed in Section 3 and we compare the characteristics between the NoSQL database and relational database in section 4. The further research on NoSQL database in information retrieval is proposed in Section 5.

2. Web Crawling System

2.1. Principles of Web Crawling

Web crawling is a process that automatically obtains information from the Web pages through the link relationships between them and expands to the entire Web. This process is mainly done by the Web Crawler which usually consists of spider, controller and original page library, as shown in Figure 1.
The spider crawls the pages from the Internet, extracts the URLs to URL database, and saves the pages to the original page library. As crawling a Web page usually takes seconds time to wait network communication, multiple spiders will be started to parallel process so as to improve the crawl rate. The

controller judges and distributes the URLs from the URL database to control the spider crawling the other pages until the URL database is empty.

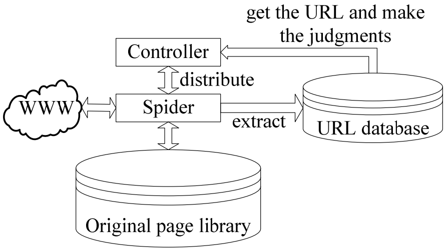

Figure 1. compositions of Web crawler

2.2. Meteorological BBS Information Collection System

2.2.1. System Introduction

Meteorological BBS information collection system filters and collects the posts of the representative meteorological BBS in Internet, including clud.weather.com.cn, www.cmabbs.com and so on. It provides a professional search engine database of meteorological information.

The system develops with Heritrix-1.14.4 and Eclipse using JDK 6.0. First, the certain extension classes of Frontier and Extractor in Heritrix are written for each BBS, run Heritrix to filter and collect the posts; second, use HTMLParse to gather the content we need from the posts and store to the text documents; then, build the word vocabulary and the lucene index, store the text documents to the database; at last, create a web site with search page, search by lucene index and get the ID number, and read all the post content from the database according to the ID and display on the page. This system will update the BBS information regularly to guarantee the real-time of search engine database.

2.2.2. Data Format

The posts in every BBS are saved to text documents in the uniform format and then stored to database. The format is shown in Figure 2.

```
[URL]
[title]
#floor
[postby]
[time]
[content]
#floor
[postby]
[time]
[content]
...
```

Figure 2: the data format

First is the URL and the title of the post, next is “#floor” which means the start of the first floor, then “postby” means the person who post this floor, then time and content of this floor. If there are more floors, “#floor postby time content” will be repeated.

2.2.3. Data Characteristics

The structure of the post has the fixed URL and title, also has the unfixed floors. Each post has the different number of floors, so their structures are different. The system gets all the content of the post from the database according to ID, including all the floors. The structure design of the database needs to not only convenient to store the content from txt documents but also easy to query quickly.

2.3. Relational Database Solution

The amount of data is huge, and grows rapidly. Meteorological BBS information collection system involves many BBS, each BBS has more than ten thousand posts and more than hundred thousand floors, and more than hundred posts increased every day. Take the Cmabbs for example, in January 1,2011, Cmabbs has 26,153 posts and 237,746 floors, and increases 617 posts in one day. That requests the huge amount of storage space of database, and the space can be expanded with the increasing amount of data. In the case of such a large amount of data, the database must still ensure the good performance of query, and the expansion of the database will not affect the query performance.

The traditional relational databases store data in the form of a two-dimensional table with strictly row and column format, and emphasize the consistency and integrity of data. In the past few decades, the relational databases almost monopolize all areas of the database applications. For meteorological BBS information collection system, we use SQL Server 2005 as the instance of relational database to solve this application. Because the relational database requests strict two-dimensional table and no nested table, we create two absolute tables: tb_post and tb_postback, to store the posts and floors. The structure of tables is shown in Figure 3.

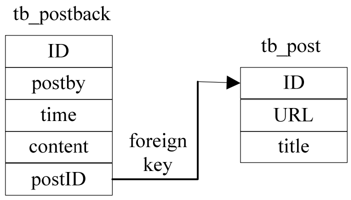

Figure 3. tb_post and tb_postback

Table tb_post stores the fixed information of post, including URL and title. Table tb_postback stores the unfixed of the post, the floors. Each floor stores in tb_postback as a record with a foreign key postID point its post. When the system query the post by the ID, SQL clause is as follows:

Select * from tb_post where ID = strID; Select * from tb_postback where postID = strID order by time desc.

3. NoSQL Database

NoSQL, as Not Only SQL, is the broad definition of non-relational data storage [7~9], whose development can be traced back to 1991, when the first edition of Berkeley DB was published. Berkeley DB is a key/value database for the embedded occasion which requires a high speed of inserting, reading and writing with a relatively simple data type. The design philosophy of NoSQL is by reducing the data consistency and integrity constraints, in exchange for high availability and partition tolerance(as horizontal scalability) to meet the requirement of Internet application with huge data amount storage, high performance and low-cost scalability. From 2007 to the present, a dozen of the popular products has emerged, such as Google's BigTable, Apache's HBase[10], FaceBook' Cassandra[11], 10gen' MongoDB, Yahoo!' PNUTS etc. These databases generally have the high performance of read and write and great horizontal scalability, which perform well in many practical Internet applications.

3.1. MongoDB solution

MongoDB is one of the most popular NoSQL database[12], whose main objective is to bridge the gap between key-value stores with high performance and scalability and traditional RDBMS with rich management, and take the advantages of both in one[13]. It is developed by the 10gen company, with the latest version 1.7.3 in November 16,2010, and now is used in many SNS (Social Networking Services) applications, such as shutterfly, foursquare , bit.ly etc. The Website "ChinaVisual", the largest community for creative people in China, moved from MySQL to mongoDB in early 2009, currently MongoDB powers its most major production and service, like file storage, session server, and user tracking [10].

MongoDB uses BSON (Binary JSON) with loose data structure as the data format, and document-oriented storage. It provides auto-sharding to achieve mass data storage, and supports full indexes. With the powerful query language syntax similar to the object-oriented language, MongoDB can achieve the most function of single-table query in relational database. It also supports atomic in-place update and two replication mechanisms of Master/Slave and replica set [9] [11].

MongoDB both has the high performance and scalability of key-value data store and rich data processing functions of traditional relational database[13], as shown in Figure 4. MongoDB‘s main features is as following:

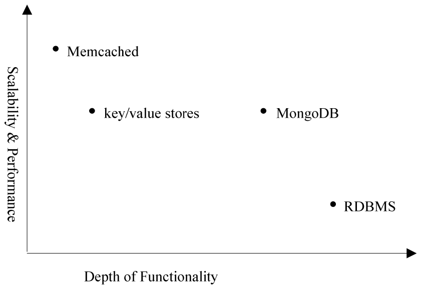

Figure 4. Feature Comparison

(1). Data model convenient to design. BSON format storage, document-oriented, documents map nicely to programming language data types, embedded documents and arrays reduces need for joins, schema-free for easy schema evolution.

(2). High performance. No joins and no transactions; full indexes with indexing to embedded documents and arrays; in-place update; optional asynchronous writing.

(3). High availability. Replica set with automatic master failover, strong robustness.

(4). Easy scalability. Automatic sharding with order-preserving partitioning, which makes the performance of distributed reading and writing fast and efficiently, easy horizontal scalability with low cost of hardware, no joins and no transactions makes distributed query simple and fast.

(5). Rich query language. MongoDB supports conditional operators, regular expressions, query on embedded document and array, and can replace most of SQL queries, complicated aggregate operations can be implemented by utility functions and map/reduce.

According to the official documents, when the amount of data is more than 50GB, the access speed of MongoDB is 10 times faster than MySql. The efficiency of concurrent read and write of MongoDB is pretty good, 5 thousand to 15 thousand read and write requests can be processed per second according to the official performance test. To support huge amount of data storage, MongoDB has a great distributed file system GridFS which provides a mechanism for transparently splitting a large file into multiple small chunks.

3.2. Design and Implement

The system uses mongodb-win32-i386-1.6.3 with development environment Eclipse + JDK 6.0, OS Windows XP(32bit).

We first create a database InfoDB and one collection to each BBS, create codes as following:

```
Mongo mo = new Mongo("localhost",27017);// specify the database server
DB InfoDB = mo.getDB("InfoDB");// get database InfoDB, if not exist create it
DBCollection baiducoll = InfoDB.getCollection("baiducoll");// create collection for Baidu Post Bar.
```

We create one document for each post in Post Bar, specify fixed fields including title, URL and floor 1, as the type of floor 1 is document which has the following fields: postedby, time and content. The not-fixed fields depend on the amount of post replies. If a reply is posted, field floor i is added, as the same type of document including postedby, time and content. Figure 5 displays the design of database.

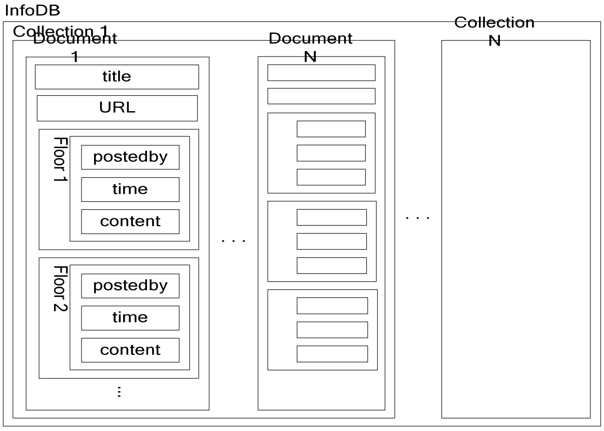

Figure 5. Design of Database

The codes to create a document are as following:

1. Create a document by class BasicDBObject, add fields with simple data type:

```
BasicDBObject doc = new BasicDBObject();//create document
doc.put("URL","http://tieba.baidu.com/f?kz=78242.html" );// add key/value pair to field URL
doc.put("title", "Dalian’s weather forecast");// add key/value pair to field title
```

2.To the document type of field, first create a document by class BasicDBObject as embedded document. Read the post floor 1, fill the information to embedded document, then add the embedded document to field floor 1 in document.

```
BasicDBObject floor = new BasicDBObject();//floor create the embedded document floor
floor.put("postedby", author);// add the field postedby
floor.put("time", time);// add the field time
floor.put("content", content);// add the field content
doc.put("floor1" , floor);// add the embedded document floor as the field floor1 to document doc
```

3. Read each floor of the post replies circularly, repeat the code in step 2, add field floor i to document.

4.add document to collection:

```
baiducoll.insert(doc);
```

When the system query the post by the ID, Mongo query is InfoDB.baiducoll.find({ id : strID}).

3. Comparison of Solutions

Compare the solution between relational database and NoSQL database, there are some conclusions as following.

First, Data Structure. In MongoDB, we put a post with all the floors in one document. Because MongoDB supports schema-free, we don’t have to design the structure beforehand, and it can be modified at run time. The fields in each document do not need to be same, which can be set depending on the actual situation when programming. How many floors there are, and how many fields we can add. In relational database, we create two tables, that one stores the post, and the other stores the floors with a foreign key to join the post, which ensures the data consistency. Those two data structures can both store the data appropriately, but the relational database stores in two tables with a foreign key which is a little complicated.

Second, Query. MongoDB supports embedded document to implement nested, so we can store a post with all the floors in one document that we can efficiently get the whole post by query the id. Relational database stores the post and the floors in two tables, with the post id we query these two tables and get the post and all the floors.

The amount of the posts and the floors are enormous. In relational database, the size of tb_post and tb_postback is huge. The records of tb_post are over ten thousands and the records of tb_postback are over hundred thousand. Relational database has poor query performance in this large size of table. MongoDB is pretty good at the query with huge amount of data, according to the official documents, when the amount of data is more than 50GB, the access speed of MongoDB is 10 times faster than

MySQL.

Last, Scalability. Auto-sharding of MongoDB can provide easy horizontal scalability with low cost of hardware, replica set of data replication can provide automatic failover, and order-preserving of sharding makes query fast. Because of strong consistency and integrity of data, the performance of relational database in the distributed data management is not good, resulting in poor horizontal scalability. To store large amount of data, the relational database has to buy advanced server with more storage space which is high-cost.

4. Conclusions

Web crawling is one of the most important applications of the Internet, whose select of database storage directly affects the performance of search engines. This article gives the solution of relational database and NoSQL database MongoDB to meteorological BBS information collection system. The data storage structure and query are designed, and the advantage and the disadvantage are listed in data structure, query and scalability. Relational database has multiple tables' storage with foreign key, sharp decline in query performance with huge amount of data, and vertical scalability with high cost. Compared to relational database, MongoDB supports schema-free, has great query performance with huge amount of data and provides easy horizontal scalability with low cost of hardware. It is more suitable for data storage in Web crawling.

5. References

[1] Debajyoti Mukhopadhyay, Sukanta Sinha , "An Algorithm for Construction of High Efficient Web Page Tree ", JCIT, Vol. 5, No. 5, pp. 44 ~ 57, 2010.
[2] Debajyoti Mukhopadhyay, Sukanta Sinha , "A Novel Approach for Domain Specific Lucky Web Search", JCIT, Vol. 5, No. 5, pp. 72 ~ 80, 2010.
[3] YaJun Du, HaiMing Li, "An Intelligent Model and Its Implementation of Search Engine", JCIT, Vol. 3, No. 2, pp. pp.57 ~ pp.66, 2008.
[4] Daniel Peng, Frank Dabek. " Large-scale Incremental Processing Using Distributed Transactions and Notifications". Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, 2010.
[5] Xie Yi,Gao hong wei and Fan chao dong. "A Survey on NoSqlDatabase[J] "."Communication of modern technology",08,46-50 , 2010
[6] R. Agrawal et. al. "The Claremont Report on Database Research". Berkeley, California, 2008.
[7] NoSql. http://nosql-databases.org/
[8] "NoSQL Relational Database Management System: Home Page". Strozzi.it. 2007-10-02. http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page. Retrieved 2010-10-29.
[9] "NOSQL 2009". Blog.sym-link.com. 2009-05-12. http://blog.sym-link.com/2009/05/12/nosql_2009.html. Retrieved 2010-10-20.
[10] HBase: structured storage of sparse data for Hadoop. http://www.rapleaf.com/pdfs/hbase_part_2.pdf.
[11] Lakshman, Avinash; Malik, Prashant. Cassandra — A Decentralized Structured Storage System. Cornell University. http://www.cs.cornell.edu/projects/ladis2009/papers/lakshman-ladis2009.pdf. Retrieved 2010-10-22.
[12] MongoDB. http://www.mongodb.org/， Retrieved 2010-10-22.
[13] http://www.mongodb.org/display/DOCS/Production+Deployments， Retrieved 2010-10-27.
[14] Chen Zhang, Hans De Sterck. "Supporting Multi-row Distributed Transactions with Global Snapshot Isolation Using Bare-bones HBase". The 11th ACM/IEEE International Conference on Grid Computing (Grid 2010), Oct 25-29, 2010, Brussels, Belgium,2010