Analysis and study on the performance of query based on NOSQL database

Peng Wang, Yan Qi*, Hua-min Yang

School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China

Received 01 May 2014, www.cmnt.lv

Abstract

With the rise of the Internet web2.0 and the arrival of the era of big data, NOSQL database has already developed rapidly. The traditional relational database in the treatment of high concurrent Web2.0 pure dynamic website and large-scale data encountered bottleneck. This paper introduces the concept of NOSQL database and data storage model. Taking the HBase database as an example, it describes the system structure and data model of HBase, and it demonstrates data query efficiency of the HBase database which is superior to the relational database. In view of the HBase database storage mechanism, the HBase database and MySQL database for performance, which is proceeding comparison, the result shows that NOSQL database performance in processing large data has obvious advantage over the traditional relational database.

Keywords: data model, NOSQL, HBase, MySQL, performance analysis

1 Introduction

In the past 20 years, the traditional relational database has been used widely [1]. With the rapid development of Internet web2.0 and information technology, the size of the data is processed by a computer, which explodes growth. In the face of the emerging Web2.0 applications, traditional relational database cannot meet all kinds of applications, but for the database has higher data concurrent access ability and higher extensibility [2].

With the arrival of the era of big data, the scale of the data to PB or EB order of magnitude, at the same time data type is diversity data type, which mainly includes structured data, non-structured data and semi-structured data. In the face of massive data processing, 80% data is semi-structured and unstructured data. Compared with the structured data (i.e. data, stored in the database, can use the dimension table to the logical expression of the data). It is not convenient to use two-dimensional logical table of the database to represent the data that is called unstructured data, the unstructured data include office documents of all formats, text, images, the subset of XML and HTML of standard generalized markup language, all kinds of report forms, images and audio/video information and so on.

With the rapid development of network and information technology, especially the rapid development of Internet and Intranet technology, the number of non-structured data is increasing. At this time, the mainly limitation is more and more obvious exposed that relational database of management structured data. At this time, NOSQL database is the database oriented application design. At present, industrial and academic circles have the NOSQL database as the new research focus.

2 NOSQL

In today's era of big bank data, academia and industry research and application of large data is continuously deepening. Hence, advantages of NOSQL database is obvious over the traditional relational database.

NOSQL (Not only SQL) refers to the non-relational, distributed, and does not provide the design mode of ACID database. At the same time, NOSQL database does not fixed pattern of data [3]. It is a supplement for the system of relational SQL data.

NOSQL database management system is a different kind of relational database management system, Its data storage format can be loose, it usually does not support the JOIN operation and it is easy to scale horizontally. It can also be called relational database.

2.1 NOSQL DATA STORAGE MODEL

The traditional relational data model often uses the two-dimensional table structure to store and process data [4]. While the NOSQL data storage does not require fixed table structure which is based on the relational model, so there is usually no join operation, which is based on table. While NOSQL has a better performance advantage in data access than the relational database, from the source speaking, this is because the NOSQL data model is based on the requirement of high performance.

According to the data storage model, NOSQL database is divided into many types, including storage, document storage, Key-Value object storage, storage, storage, XML database and so on.

* ***Corresponding author*** e-mail: 1136161972@qq.com

2.2 RELATIONAL DATABASE AND NOSQL DATABASE

Compared with the relational database, the main advantage of NOSQL data storage management system lies in:

1) Do not make unnecessary complexity. Relational databases provide a wide variety of characteristics and strong consistency, but using the characteristics of relational data in some specific applications, some functionality of relational databases are seldom used. But NOSQL system uses little function to improve performance.

2) High throughput. In terms of throughput, the traditional relational data management system is much lower than some NOSQL database systems.

3) High level extension ability, not high-end hardware cluster. In the aspect of scale, NOSQL database system is able to extend, and do not need too much cost, the cost of scale has a big difference between NOSQL databases and methods of the relational database cluster [5].

4) Object-relational mapping is not going to happen. In terms of large data storage, the data object is stored in a lot of NOSQL database systems, the cost of mutual transformation which is between relational model of database and object model in the program is reduced to zero.

3 HBASE

For large data storage, NOSQL has many mainstream databases. It includes HBase, Redis, CochDB and mongoDB and so on [6]. Below it will take the HBase of NOSQL database and MySQL of relational database as an example, and it explains the difference of performance between HBase and MySQL.

HBase is an open source, distributed, multiple version, the storage model for the column [7]. In addition, loose data is generally stored in HBase. Specifically, the data stored in HBase, which is between Ying Xie (key/value) and relational data [8].

HBase provides a capability similar to Bigtable on Hadoop. HBase is a sub project of the Apache Hadoop project. HBase is different from the relational database in general, it is a suitable for unstructured data storage database. Another difference is HBase, which is based on the column and not based on line model.

3.1 HBASE SYSTEM ARCHITECTURE

HBase server architecture is similar to the HDFS architecture, it follows a simple master/slave server architecture, it is mainly composed of HBase Master Server (HBase Master Server) and HRegionServer Server group [9]. In simple terms, HBase Master Server is responsible for the whole cluster. All of the servers in the HBase are coordinated by ZooKeeper, and ZooKeeper saves the data to HRegionServer mapping. HRegionServer is responsible for storing the large table in the HBase, and manages read/write data that users send. HBase logic table is defined to be a Region, which is stored in a HRegionServer. Relationship between the HRegionServer and the Region is a one to many relationships.

The internal of HRegionServer manages a series of HRegion objects, each HRegion corresponds to a Region of Table, HRegion is composed of a large number of HStores. Each HStore corresponds to a Column Family in the Table store, we can see that each Column Family which is actually a centralized storage unit, so the best way will put the column of common I/O characteristics into a Column Family, so it is the most efficient way. HStore storage is the core of HBase storage, which is composed of two parts: a part is MemStore, another part is StoreFiles. MemStore is Sorted Memory Buffer, first of all, the user written data will put into the MemStore, when the MemStore is full and it will flush into a StoreFile (the underlying implementation is HFile).When the number of StoreFile file grows to a certain threshold, it will trigger the Compact merge operation, and which multiples StoreFiles into a single StoreFile, the process of the merge will merge and delete data version. Thus it can be seen that : HBase is only increased data, all update and delete operations are performed at the compact during subsequent write operation, which makes the user enter the memory and return immediately, and this will ensure the high performance of HBase I/O. Figure 1 is a HBase architecture diagram.

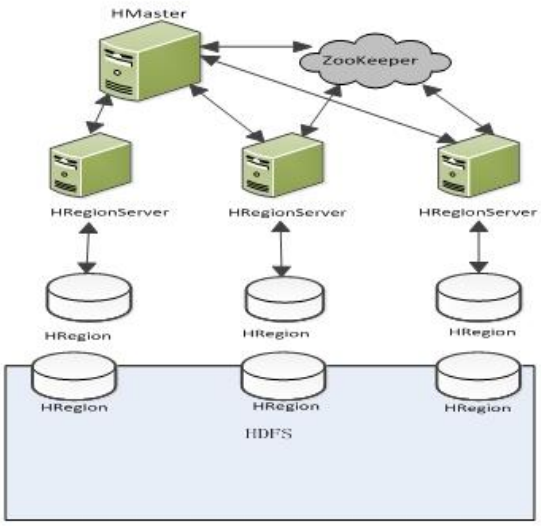


FIGURE 1 HBase architecture diagram

1) Client.

HBase client use RPC mechanism of HBase carries on communication with HMaster and HRegionServer, for the operation of the management, client and HMaster carry on RPC; for the operation of data read and write, the client and HRegionServer carry on RPC.

2) Zookeeper.

Zookeeper Quorum in addition to store-ROOT-address and HMaster address, HRegionServer will also take itself to register to Zookeeper by the Ephemeral, so

that HMaster can be perceived health status of each HRegionServer. In addition, Zookeeper is also to avoid the problems of single point of HMaster.

3) HMaster.

HMaster has no single point problem, HBase can start a number of HMasters, through the Master Election mechanism of Zookeeper to ensure that there is always a Master which is operating, the function of HMaster is mainly responsible for the management of Table and Region:

- manage users on the Table to add, delete, change, check operation;
- manage Load balancing management of HRegionServer, and adjust the Region distribution;
- in Region Split, assign to the new Region;
- when the HRegionServer is shut down, and HMaster is responsible for Regions migration of the failed HRegionServer.

4) HRegionServer.

HRegionServer is mainly responsible for responding to user I/O request, read and write data to HDFS file system, HRegionServer is one of the core modules in HBase.

3.2 DATA MODEL

Big data is special body big, particularly large data sets, the data category, and such data sets cannot use the traditional database tool to grab, management, and its content to its processing [10]. Big data generally refers to the 10 TB (1 TB = 1024 GB) amount of data of scale. But in practical applications, many enterprise users put multiple data sets together, the amount of data has already formed the PB level. In order to solve the limitation of relational database in processing large data theory and implementation, we use the HBase of NOSQL database for data storage.

In the HBase database, data logical structure is also table, but the table structure of the HBase database is different from the relational database. HBase is a sparse long-term storage, multi-dimensional, sort of mapping table. The first row in the table of the HBase is the structure information of table, in the definition of the table. Firstly, to define the column family of the table, the column family is composed of many columns, when the users store the data in the table and each line has a sortable primary key and any number of columns [11].

TABLE 1 Conceptual view of HBase table data

Row key	Timestamp	Column family: C1		Column family: C2	
R1	T3	C1:1	Value1-(1/1)		
R1	T2	C1:2	Value1-(1/2)		
R1	T1			C2:1	Value1-(2/1)

In the HBase table, some columns are blank; the blank columns will not be stored.

1) Conceptual view.

HBase table is conceivable to be a great relationship mapping, it will be able to locate the specific data by row key, row key + a timestamp or row key+ column, Table 1 is the conceptual view of HBase table data.

There is a data in this table: R1, and there are two column families: C1 and C2. Among them, C1 has two data, C2 has a data.

2) Physical view.

From conceptual view of HBase at each table is made up of many lines, but in the physical storage, table is stored in the column, the following Table 2 is the physical view of HBase table data.

TABLE 2 Physical view of HBase table data

Row key	Timestamp	Column family:C1	
R1	T3	C1:1	Value1-(1/1)
R1	T2	C1:2	Value1-(1/2)
Row key	**Timestamp**	**Column family:C2**	
R1	T1	C2:1	Value1-(2/1)

Some columns are blank in the conceptual view, this column will not be stored, actually when request this blank cell, it returns null.

3.3 JAVA API

HBase in the data storage process, the main classes covered include: HBaseAdmin, HTable, Put, Get, Scanner and ResultScanner etc [12]. The following will describe the HBaseAdmin class, HTable class, Scanner class, ResultScanner class, and talk about methods of the above mentioned class in detail.

1) HBaseAdmin.

Relationship:org.apache.hadoop.hbase.client.HBaseAdmin.

Role: provide an interface to manage the information of HBase database table. There is include the methods that it provides: create table, delete the table, list item, make table valid or table invalid, and add or remove column family members of a table and so on.

Table 3 shows the main function of HBaseAdmin class.

TABLE 3 HBaseAdmin class

Function	Describe
addColumn(String tableName, HColumnDescriptor column)	To add a column to an existing table
createTable(HTableDescriptor desc)	Create a table, synchronous operation
deleteTable(byte[] tableName)	Delete an existing table
enableTable(byte[] tableName)	Make the table valid state
disableTable(byte[] tableName)	Make the table invalid state

2) HTable.

Relationship:org.apache.hadoop.hbase.client.HTable.

Role: It can be used to keep communication with HBase directly. This method is not thread safe for update operations.

Table 4 shows the main function of HTable class.

TABLE 4 HTable class

Function	Describe
close()	Release all resources or hang in the update of internal buffer
exists(Get get)	Check whether the Get of the value specified instance exists in HTable column
get(Get get)	Get the Corresponding to some cells access to the values of the specified row
getScanner(byte[] family)	Get scanner instance of the current given column family
getTableDescriptor()	Get HTableDescriptor instance of the current table
getTableName()	Get the table name
put(Put put)	Add the values to table

3) Scanner.

Relationship:org.apache.hadoop.hbase.client.Scan.

Role: According to the conditions to obtain for a batch records.

Table 5 shows the main function of Scanner class.

TABLE 5 Scanner class

Function	Describe
SetCaching()	Set the number of cache, reasonable setting the cache amount, and it can improve the query efficiency
SetBatch()	Specify the maximum number of returned Cell
SetStartRow()	Specify the start line
SetEndRow()	Specify the end of the line (not including the trip)
SetFilter()	Specify a Filter to Filter out unwanted messages

4) ResultScanner.

Relationship: Interface.

Role: The client obtains the value of the interface. Table 6 shows the main function of ResultScanner class.

TABLE 6 ResultScanner class

Function	Describe
Next()	Get the next row values
Close()	Close the scanner and release its allocated resources

Users mainly use the Scanner class and ResultScanner when querying data, in order to illustrate the above Scanner class and ResultScanner class that provide methods, take an example, a logging system has a demand, it is based on an operation to find out all the operations of the whole session. Suppose that we find all the operations of SID = 100000, the code is as the following:

```
Scan scan = new Scan ();
String startkey=DATETIME + "100000"
String endkey= DATETIME +"100000"+"99999"
scan.setStartRow(Bytes.toBytes(startKey));
scan.setStopRow(Bytes.toBytes(endKey));
ResultScanner rs = currentTable.getScanner(scan).
```

By using the Scanner class provides the method of setStartRow and setStopRow, the method of setStartRow and setStopRow limit the search scope, and they effectively improve the query speed of data.

4 Case analysis

The physical store of HBase is based on the column store [13], but the relational database is based on the table structure and mode of storage. The data operation of HBase is different from the data operation of the relational database. For example, HBase only has query, insert, delete, etc, and between the table and table are separated, there is no connection between the table and the table, but relational database has many functions and join operations.

However, structure of HBase table does not need the connection operation, but it can solve the problem that the connection operation solves. The solution is adding a specific keyword in a row [14], you can put all the data of the connection together.

Below we will take a Storage table of the global logistics distribution system as an example to simulate the HBase table design. As is known to all, in the relational database, the storage table structure of the global logistics system is as shown in Table 7 to Table 9. In the three tables, it assumes that the data of a year is about 100 terabytes to 1000 terabytes, in the query of data, the speed will be very slow, so the design of the three tables according to HBase table model, which are designed into two tables, this will effectively improve the query speed of data.

TABLE 7 Storage

S_ID	S_Name	S_*Area*	S_Call	S_Address
Storage ID	Storage Name	Storage Area	Call	Address

TABLE 8 Parts

P_ID	P_Name	P_Rule	P_Unit price	P_Describe
Parts ID	Parts Name	Parts Rule	Unit price	Describe

TABLE 9 Stock

St_uid	St _bid	St _Quantity
Stock uid	Stock bid	Quantity

In the HBase database, data storage mode will be as shown in Table 10 to Table 11.

TABLE 10 Storage table of HBase

Row key	Column Family		Column Family	
<S_ID>	Info	value	parts	value
<S_ID>	Info: S_Name	The name	parts:<P_ID>	<St_Quantity>
<S_ID>	Info: S_Area	The area	parts:<P_ID>	<St_Quantity>
<S_ID>	Info: S_Call	The call	parts:<P_ID>	<St_Quantity>
<S_ID>	Info: S_Address	The address	parts:<P_ID>	<St_Quantity>

TABLE 11 Parts table of HBase

Row key	Column Family		Column Family	
<P_ID>	Info	value	storage	value
<P_ID>	Info: P_Name	The Name	storage:<S_ID>	<St_Quantity>
<P_ID>	Info: P_Rule	The Rule	storage:<S_ID>	<St_Quantity>
<P_ID>	Info: P_Unit price	The Unit price	storage:<S_ID>	<St_Quantity>
<P_ID>	Info: P_Describe	The Describe	storage:<S_ID>	<St_Quantity>

From the above 5 can be seen in the Table, it can be completed operations in a relational database, the operations not only can be finished in HBase, also can have a better performance, the Row key is an index in HBase, Therefore, data query in HBase, query speed is faster than a relational database.

5 Performance test and result analysis

Hardware environment: OS is Ubuntu Server 13.04, HBase version is 0.20.6, Hadoop version is 1.0.1, 10 sets of ordinary PC cluster, on the cluster uses HDFS as a file system [15]. We will compare the performance of NOSQL HBase database with the relational database MySQL, test two for performance difference between 1000000 data query.

Figure 2 to Figure 7 is in the same case, throughput and transactions per second of HBase database and throughput and transactions per second of MySQL database.

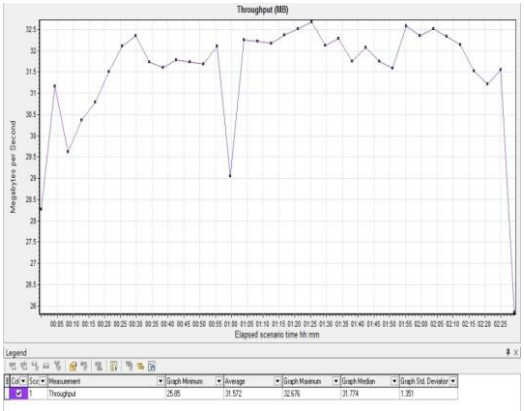


FIGURE 2 Throughput of HBase database

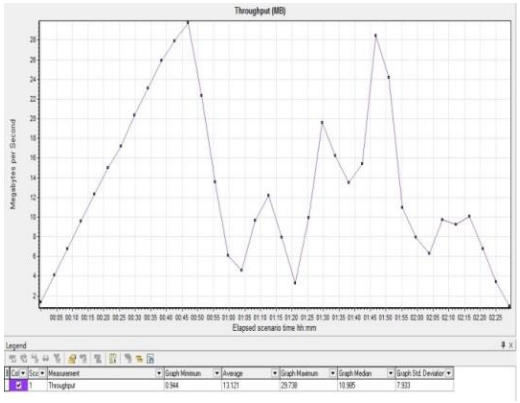


FIGURE 3 Throughput of MySQL database

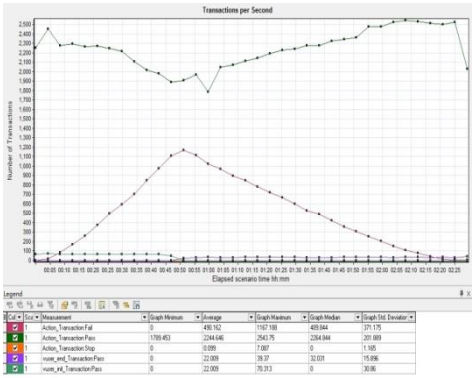


FIGURE 4 Transactions per Second of HBase database

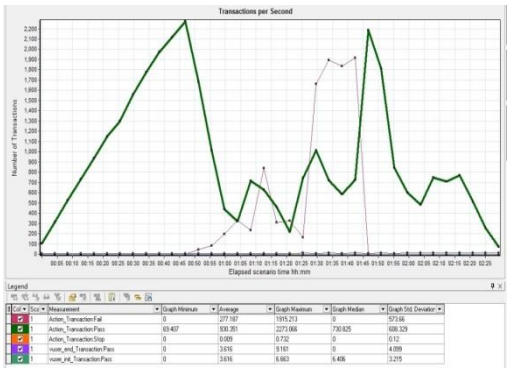


FIGURE 5 Transactions per Second of MySQL database

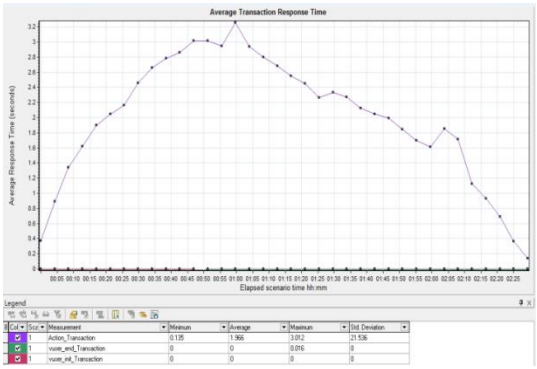


FIGURE 6 Average Transaction Response Time of HBase database

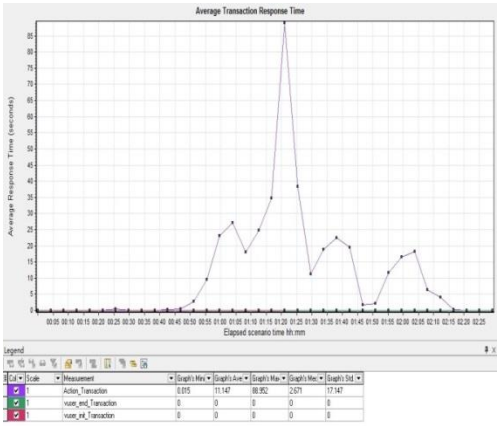


FIGURE 7 Average Transaction Response Time of MySQL database

Seen from the Figure 2 and Figure 3, when operation is the same and the number of concurrent is the maximum, the throughput of HBase is close to the average value, the throughput of MySQL reaches the minimum value in

addition to the starting point and ending point. When the time of test is the same, the average of HBase throughtput is 31.572MB, the average of MySQL throughtput is 13.121MB, the average of HBase throughtput is the average of MySQL throughtput 2.046 times. HBase curve is more stable than MySQL curve under the same condition.

Seen from the Figure 4 and Figure 5, when operation is the same and the number of concurrent is the maximum, HBase and MySQL through the number of transactions reach a low value. HBase in the corresponding numerical and maximum number of the through transactions per second is about 300, while the MySQL in the corresponding numerical and maximum number of the through transactions per second is about 2000.When the time of test is the same, the average of HBase transactions per second is 2244.646, the average of MySQL transactions per second is 930.351, HBase transactions per second is the average of MySQL transactions per second 2.41 times.

Seen from the Figure 6 and Figure7, when operation is the same and the number of concurrent is the maximum, the average transaction response time of the HBase is close to average, average transaction response time of the MySQL reaches the maximum value. When the test time is same, the average value of the HBase average transaction response time is 1.966s, the average value of the MySQL average transaction response time is 11.147s, the average value of HBase average transaction response time average is less than the MySQL average transaction response time.

Table 12 shows the experimental comparison result of HBase database and MySQL database.

From the above data show that: the maximum of the throughput per second in the HBase database is 32.676 MB, the minimum is 25.85MB, the average is 31.572MB. The maximum of the transactions per second in the HBase database is 2543.75, the minimum is 1789.453, the average is 2244.646.

TABLE 12 Experimental comparison result

	HBase			**MySQL**		
	max	*average*	*min*	*max*	*average*	*min*
Through-put(MB)	32.676	31.572	25.85	29.738	13.121	0.944
Transactions per second	2543.75	2244.646	1789.453	2273.066	930.351	69.407
Average transacttion response time(S)	3.012	1.966	0.135	88.952	11.147	0.015

The maximum of the throughput per second in the MySQL database is 29.738 MB, the minimum is 0.944MB, the average is 13.121MB. The maximum of the transactions per second in the MySQL database is 2273.066, the minimum is 69.407, the average is 930.351.

The maximum of the Average Transaction Response Time in the HBase database is 3.012S, the minimum is 0.135S, the average is 1.966S. The maximum of the Average Transaction Response Time in the MySQL database is 88.952S, the minimum is 0.015S, the average is 11.147S.

The experimental results show that: HBase curve is more stable than MySQL curve under the same condition. Therefore, performance advantage of the HBase database is superior to the MySQL database.

6 Conclusions

This paper introduces the main stream HBase database of NOSQL, and elaborates the data model and related features of the HBase database. It compares the performance of the HBase database with the relational database. The experimental result shows that: the performance of HBase in processing large data is relational database that cannot match advantage. At the same time, NOSQL database in emerging applications also shows a good momentum [16], and NOSQL database has injected new energy into the database of the industry.

References

[1] Hecht R, Jablonski S 2011 NoSQL Evaluation *International Conference on Cloud and Service Computing* 336-41

[2] Jing H, Haihong E, Guan L 2011 Survey on NoSQL database Pervasive Computing and Applications (ICPCA) *6th International Conference* 363-6

[3] Lu J 2013 Big data challenge and NoSQL database technology *Beijing Electronic industry press* (*in Chinese*)

[4] Wang S, Sa S 2006 Introduction to database system *Beijing Higher education press (in Chinese)*

[5] Franke C, Morin S 2011 Distributed Semantic Web Data Mangement in HBase and MySQL Cluster *IEEE 4th International Conference on Cloud Computing* 105-12

[6] Zhu J, Wang J, Zhang J 2014 Based on NoSQL database of research and application technology of data query C*hina railway science (in Chinese)*

[7] Lu J 2012 Hadoop in actual combat *Beijing Mechanical industry press* (*in Chinese*)

[8] HBase:bigtable-like structured storage for hadoop hdfs[EB/OL].http://hadoop.apache.org/hbase 2010.

[9] Borthakur D 2011 The hadoop distributed file system: Architecture and design [EB/OL] http://hadoop.apache.org/hdfs

[10] Zhao G, Yi H 2013 The historical opportunity of the era of large data: industrial transformation and scientific data *Beijing Tsinghua university press (in Chinese)*

[11] Li M, Jia H 2012 Cloud data management of the new data model *Computer engineering and design*

[12] Vora M N 2011 Hadoop-HBase for Large-Scale Data *International Conference on Computer Science and Network Technology* 601-5

[13] Huang J, Ouyang X, Jose J, Wasi-ur-Rahman M, Wang H, Luo M, Subramoni H, Murthy C, Panda D K 2012 High-Performance Design of HBase with RDMA over InfiniBand *Parallel & Distributed Processing Symposium (IPDPS) 2012 IEEE 26th International* 774-85

[14] Zhang Y, Ma Y, Meng X 2011 A kind of efficient space keyword query strategy based on HBase *The wind of science and technology*

[15] Wang Hongyu 2011 Hadoop platform application in cloud computing *Software* **32**(4) 36-9

[16] Quan Zhi, Song Jing-jing 2011 Based on the analysis of NoSQL database technology *With the network information system (in Chinese)*

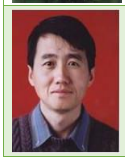
Authors

Peng Wang, born in June, 1973, Zhuolu County, Hebei Province, China

Current position, grades: associate professor of Changchun University of Science and Technology.
University studies: Master of engineering, doctor of engineering in reading of Changchun University of Science and Technology.
Scientific interest: software engineering and information systems, database and data mining.
Publications: More than 10 papers, 6 Ei retrieval among 10 papers.

Yan Qi, born in November, 1987, Heishan County, Liaoning Province, China

Current position, grades: postgraduate student in reading of Changchun University of Science and Technology.
University studies: database theory and software.



Hua-min Yang, born in November, 1963, Wangqing County, Jilin Province, China

Current position, grades: Dr., Doctoral tutor, professor of Changchun University of Science and Technology.
University studies: virtual reality and digital media, image processing.