Research on cloud storage technologies of typical crop growth environment monitoring data

Yuansheng Wang, Huarui Wu*, Lei Luo, Weihong Ma, Kun Liu

[1]*Beijing Research Center for Information Technology in Agriculture, Beijing 100097, China*

[2]*Key Laboratory of Agri-informatics, Ministry of Agriculture, Beijing 100097, China*

Received 2 August 2014, www.cmnt.lv

Abstract

The typical crop is an important part of regional economy, using of information technology tools to enhance the fine management of typical crop, improve the efficiency and level of agricultural production will play an important role to promote regional economic development. The Internet of things technologies can provide conditions for precise management of crop, but it will also lead to new technical application difficult problems, focusing on the storage and processing problem of the ”real time, high frequency, mass, rapid growth” monitoring data of the crop growth environment, this paper put forward a network management and scheduling method for massive data based cloud storage technologies to solve the large data storage and concurrently access bottlenecks of the traditional relational storage, and provided good technical conditions for the typical crop production management.

Keywords: typical crop, growth environment monitoring, cloud storage

1 Introduction

In the growth stage, monitoring of the growth environment of typical crop quantitatively can assist agricultural production managers grasp the crop growth information in time, control the growing conditions in a reasonable manner, so as to achieve the targets of intensive production, increasing yield and efficiency. Currently, through the sensor and the farmland wireless sensor network, the growth information of the crop such as light, temperature, humidity, CO_2 concentration can be monitored and transferred to the remote data centre, inputting them to the production management decision system as the parameters, the agricultural technical personnel can take targeted production measures timely [1].

There are many parameters in the typical crop growth environment monitoring. The environmental monitoring data can be real-time or every few seconds transmitted to the data center, while a data is send in 5 seconds, a sensor node can send 17,280 pieces of data to the server. If there are 1,000 transmit nodes the server will receive 17.28 million pieces data every day. In fact, as the number of typical agricultural product varieties, production field and network nodes increase. The data received will reach far more than this amount, and exceed the capacity of server, it will inevitably lead to many problems such as data volume growth too fast, insufficient database table storage, database system cannot be accessed normally, and etc. [2]. Against the storage and scheduling problems of "real time, high frequency, vast and fast-growing" monitoring data, the mainstream mass data cloud storage technologies were analysed in this paper, a storage and access method of typical crop growth monitoring data resource was designed and put forward based on massive database solving the bottleneck problem effectively.

2 Distributed cloud storage system

Distributed cloud storage system is one of the core technologies of cloud computing that can integrating a large number of different types of storage devices in the network and provides system functions of data storage and business access through cluster application, grid technology, distributed file systems, and etc. [3].

In this paper, the Internet of things technology was used to monitor the typical crop growth environment, which will inevitably bring large data storage and processing problems [4]. Because the big data has characteristics such as vast and varied, high-speed and mutation [5]. When facing with petabytes of mass data storage requirements, the traditional storage system will create bottlenecks in the capacity expansion and access performance, to resolve this difficult problem, the cloud storage, with its advantage of strong scalability, cost-effective and good fault tolerance can be used, which has been widely recognized in the industry [3].

Currently, the mainstream cloud storage technologies include distributed file system and massive database system. The former is mainly used to store unstructured data, the latter is mainly used to store structured massive data, and part of them can be set up on distributed file system though automatically extending storage space to help the database system automatically expansion and

****Corresponding author** e-mail: wuhr@nercita.org.cn

maintain stable performance. So as to make the growth of the data not be restricted by storage capacity and storage node.

2.1 THE DISTRIBUTED FILE SYSTEM

The Distributed file system is the core of the cloud storage system, the good scalability and fault tolerance features of the cloud storage system depend on the support of the distributed file system support. The current mainstream distributed file systems include GFS, HDFS, Lustre, fastDFS, PVFS, GPFS, PFS, ceph, TFS and etc. (Table 1), their design thinking is broadly similar, but each one has its own characteristics [3]. Among above systems, the fastDFS is developed by pure c language to achieve the similar lightweight architecture like Google distributed file system, the typical distributed file system was emphatically introduce as follow:

TABLE 1 Mainstream distributed file system

Name	Profile	Features
Google File System (GFS)	A scalable distributed file system, which was mainly used for the large-scale, distributed, intensive data storage.	Runs on a large number of low-cost hardware, provides fault tolerance and high performance services for a large number of users to handle huge amounts of data.
Hadoop Distributed File System (HDFS)	A distributed file system under distributed computing framework Hadoop launched by apache open source organization, providing the underlying support for distributed computing storage.	Taking the error detection and quick recovery as the core goal, having the advantages of high fault tolerance and high throughput, providing the data access model of “written once and repeatedly read” for HDFS applications, and the calculating can be migrated to the node near by the data.
Lustre File System	The distributed file system, launched by Clusterfilesystem Company, is a typical distributed file system based on object storage, the same data files can be divided into several objects stored in different object storage devices, large file I/o operations are assigned to different object storage devices on parallel implementation, so as to realize a large aggregate bandwidth.	Using the expensive and stable server as the file system node, combines the characteristics of the traditional distributed file system and the design concept of traditional shared storage file system. Having the advantages of more effective data management mechanism, the global data sharing, based-on-object-storage, intelligent storage and rapid deployment, so the random access performance is good, but it has none fault tolerance, the storage volume cannot exceed than 1 petabytes, and part of the data will not be accessed when a node failure.
General Parallel File System (GPFS)	A parallel Shared file system launched by IBM which was designed for Linux cluster system drawing on the virtual shared disk technology of IBM Linux cluster system. The compute nodes can parallel access the data of multiple disks in the system at the same time through using of exchange network, and achieve a high I/O bandwidth relying on this access method.	Storing large files on a different disk in a cycle way, reading and writing the small files by merging the operations at the same time, using dynamic election of metadata node to manage the metadata; In addition, the GPFS also has automatic recovery strategy based on log failure node and centralized data locking mechanism.
Parallel File System (PFS)	A distributed file system of Sun company, its main idea is spreading the files across multiple disks and server, and taking the multiple devices as a logical virtual disk to a unified management. PFS can span more than one storage system considering all storage devices at the whole PFS as one part of a virtual disk; When there are multiple nodes access the same file at the same time, PFS can provide access service for these nodes in parallel.	Built on Solaris operating systems, including host nodes, compute nodes, I/O master node and I/O slave nodes. Host node is the entrance of PFS for other systems, only the users who successfully log in to the host node can access the data file within PFS. Compute nodes are primarily used to manage communication system and memory resources of PFS. I/O master nodes are primarily responsible for the directory management and store management of file system, as well as providing services to read and write the stored data files. I/O slave nodes are only used to handle disk read/write operations and allocate of blank block.

These distributed file systems have the functions of spreading the files to the more network storage nodes, and providing the I/O of concurrent access. Considering the factors such as characteristic of agriculture industry, building cost, ease of use and mobility, this paper adopt the HDFS as the distributed file system of the distributed cloud storage system for typical crop production environmental monitoring.

2.2 THE MASSIVE DATABASE SYSTEM

The massive database systems mentioned in this article refers specifically to the NoSQL Database systems corresponding to cloud computing and distributed storage. The NoSQL Database is collectively called the database management system which is different from the traditional relational, compared to the traditional relational databases which domain the database world for 30 years. The biggest difference is not used SQL (Structured query language) query language; it can store data not in accordance with the fixed table mode, and usually has the characteristics of horizontal scalability. The term “NoSQL” first appeared in 1998, after ten years of development, it evolved into a real technology trend in 2009, pushed by the of big data requirements in the Web 2.0 era, it becomes extremely popular new area and is developing very quickly [7].

Now mainstream NOSQL Database includes Cassandra, Hive, HBase, MongoDB, CouchDB, DynamoDB, Oracle NoSQL, IBM InfoSphere BigInsights, and so on. Among them, Hive, HBase, MongoDB, Oracle NoSQL were used commonly at present, their detailed information was emphatically introduced as follow:

TABLE 2 Mainstream mass-database system (NoSQL databases)

Name	Profile	Features
Hive [8]	Hive is a data warehouse architecture based on Hadoop file system, it uses the MapReduce programming technology, achieving part of the SQL statement, provides the programming interface like SQL.	Hive storage based on Hadoop file system, it does not have a dedicated data storage format itself, also can't establish the index of data, users can very free to organize the tables in Hive, they only need to tell hive the column separators and line separators in the data in order to parse the data when creating a table.
HBase	The Hadoop Database, is a highly reliable, high-performance, column-oriented, scalable distributed storage systems.	Hbase can be built with large-scale structured storage cluster on cheap PC server. It is an open source implementation of Google BigTable using Jave language [7]. HBase is loosely-data stored, its storage mode between the mapping (key/value) and the relational data [8].
MongoDB [7]	An open source product between relational database and NoSQL database is one the most abundant NoSql database in function, most like a relational database product.	As a database based on distributed file storage, MongoDB is developed by C++ language, its query languages are very powerful, and it also supports data index.
Oracle NoSQL DB	Oracle NoSQL Database is designed specifically for huge amounts of data management, and it can access the unstructured data, and can be horizontal scalability to hundreds of high availability nodes.	Can be built with Hadoop file system, by means of its own extension mechanism, it can be setup on multiple high-availability node, and has no single point of failure.

These NoSQL database systems can scale horizontally, support high concurrent read and write. They was used not only the traditional relational database model, but also the storage models such as key-value, document type, column, pattern database, XML storage and so on. Most of them are key-value storage [9], the data storage is not bound by ACID theory of traditional relational database, and can expand by freeways, stagger the time, separate partition, store in more than one network node. Taking into account the technological superiority of the horizontal scalability of Oracle NoSQL system itself, bidirectional data exchange with the Hadoop distributed file system and the developing technologies for mass data applications, this article selects the Oracle NoSQL DB as the data storage system of typical crops growth environment monitoring.

3 System design

In order to make cloud storage service technology be applied to the production practice of typical crop growth environment monitoring effectively, this article adopts the sensor acquisition equipment researched and developed independently to get the data, transmitting the data collected from the field to the cloud storage devices in data centre. The cloud storage centre provides uniform data access service of storing the data in the distributed storage unit of the network node automatically, in a logical view, the storage facilities for application are used as a whole storage unit. The horizontal scalability and the parallel processing of mass data are completed by the massive database system and the distributed file system automatically, the overall technical framework of the system is as follows (Figure 1):

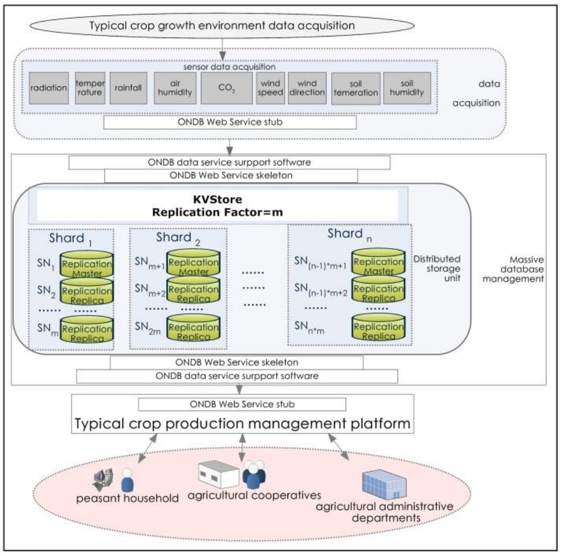


FIGURE 1 Overall cloud storage technology framework for typical crops growth environmental monitoring

3.1 THE DISTRIBUTED FILE SYSTEM

For convenience of system design and implementation, the micro-series of weather station equipment "green cloud grid" which was researched and developed independently was adopted in terms of the field data acquisition and the background data communications can be customized independently.

3.1.1 Acquisition device models

For comprehensive monitoring of typical crop growth environment, the 9 parameters sensor acquisition equipment was chosen to gather the radiation, temperature, humidity, precipitation, carbon dioxide (CO_2) concentration, wind speed, wind direction, soil moisture and soil temperature data through a set of equipment at the same time.

3.1.2 Equipment parameters

The work parameters of sensors acquisition device are as follows (Table 3):

TABLE 3 Working parameters of crop growth monitoring sensor

Sensor	Quantum	Precision	Resolution
Air temperature sensor	[-40,123.8] ℃	±0.5 ℃(25℃)	0.01 ℃
Soil temperature sensor	[-55, 125]℃	±0.5 ℃(-10℃- 85℃)	0.04 ℃
Air humidity sensor	0-100%	±4.5%RH	0.03%RH
Light intensity sensor	0-6 lux or 0-20 lux	5%	0.1 lux
Wind sensor	0-30 m/s	0.1 m /s	0.1 m /s
Wind direction sensor	16 directions	/	1/16 directions
Carbon dioxide(CO_2) concentration	0-2000 ppm, 0-5000 ppm, 0-10,000 ppm Optional;	Measurement accuracy: ± 5%; Repeatability: ± 1%.	/
Soil moisture sensor	0-100%	Within 0 ~ 50% (m^3/m^3): ±3% (m^3/m^3)	/
Rainfall sensor	Daily rainfall (0.0 mm - 999.8 mm) Total rainfall (0.0 mm - 9999 mm)	±4 %, ±1 Rainfall counts per hour 0.01 ″- 2.00 ″ (per hour: 0.2 mm and 50.0 mm); ±5 %, ±1 Rainfall counts per hour 2.00 ″- 4.00″ (per hour: 50.0 mm 100.0 mm)	/

3.1.3 Data format

At the scene of the data collection in the crop field, the CAN (control area network) bus was arranged to transmit the data from the sensors to the remote data centre in the form of packet message transmission.

3.2 MASS DATA MANAGEMENT

According to the layer structure of the mass database, it including interface layer, logical model layer, distribution layer and persistence layer [10], in the data centre more than one PC server were chosen to deploy Oracle NoSQL DB. The first step is to set the parameters such as the number of storage Nodes (Store Nodes), the total number of partitions, replicator (Replication Factor), divided area (Shard), and so on. Secondly, the network topology of the cloud storage system must be deployed to make the data storage can scale automatically, can be exchanged with the Hadoop distributed file system, and the data can be stored without being limited by the capacity of the single storage node. Thirdly, research and develop the corresponding Web Service cloud storage middleware of mass data management independently. Fourthly, deploy the middleware skeleton program on the server side to receive the read/write request from clients, and active the middleware itself to complete the physically actions. Finally, deploy the stub program of the middleware both on the PC in CAN bus, and on the PC of typical crop production management platform, so as to realize the integrated management of mass data logically.

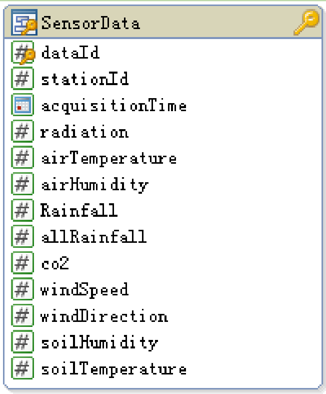
4 Key technologies of system implementation

Currently, not like the relation database, the massive databases have neither the data models such as row and line pattern, one-to-one/many-to-many pattern, nor the metadata management mechanism like the relational database management system, without metadata archives, it is difficult to accurately know the exact content stored of the massive database. In this paper, using for reference of the experience of the relational data model building, the Web Service components for modelling, storage and dispatch of the massive data were researched and developed for typical crop growth environment monitoring.

4.1 MASSIVE DATA MODELLING

Most Nosql Databases adopt the key-value pattern or the similar way to store data, so it is difficult to use an intuitive model to describe it. In this paper, according to the business application requirements of the typical crop growth environment monitoring, the data model was still designed as a relational data model to express. Only when the actual program process run, the business logic middleware will execute the different data management mechanism from DBMS system in a much higher efficiency through distributed data process. So this article uses Adobe Fiber Data model to define the model of the typical crop growth monitoring data, the detail model was designed as follows (Table 4):

TABLE 4 Data model of crop growth environmental monitoring data



	The data model diagram	Data items (Field) Chinese name	Data items (Field) English name	Type
The data model		Serial number	dataId	integer
		Node ID	stationId	integer
		Acquisition time	acquisitionTime	datatime
		Illumination (Lux)	radiation	float
		Temperature (°c)	airTemperature	float
		Humidity (%)	airHumidity	float
		Daily rainfall (mm)	Rainfall	float
		Total rainfall (mm)	allRainfall	float
		CO2 Concentration (PPM)	CO2	float
		Wind speed (m/s)	windSpeed	float
		Wind direction	windDirection	string
		Soil water content (%)	soilHumidity	float
		Soil temperature (°c)	soilTemperature	float

4.2 MASSIVE DATA PERSISTENCE

In order to facilitate the migration from DBMS-orient application to mass-database-orient applications, decrease the difficulty of the application development, and can use the JPQL grammar similar to relational database to access massive database data in general case. This paper adopted the data persistence framework to define a Java object model corresponding to the data model in section 4.1 at first, and wrote a web service entity rwSensorData as follow (Table 5):

TABLE 5 Crop growth environment monitoring data Java Object model

	Java Object Model(DTO)	Key programs (pseudocode)
Java Object Model		see code below


```
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import javax.persistence.Query
import javax.persistence.*;
import org.eclipse.persistence.nosql.annotations.NoSql;
import org.eclipse.persistence.nosql.annotations.DataFormatType;
@Entity
@NoSql(dataFormat=DataFormatType.MAPPED)
public class SensorData {
      @Id
      @GeneratedValue
      private double dataId;
            ……..
      @Basic
            private Double windSpeed;
         public double getDataId() {
            return dataId;
         }
         public void setDataId(double dataId) {
            this.dataId = dataId;
         }
         ……
         public Double getWindSpeed() {
            return windSpeed;
         }
         public void setWindSpeed(Double windSpeed) {
            this.windSpeed = windSpeed;
         }
}
```

From the formal point of view, the model above was similar to the Java object model of relational database. The differences mainly consists of the note symbols started with "@" which inform the data persistence framework to perform the corresponding operation in Oracle NoSQL database. So it is convenient to integrate it with traditional application and then based on the data persistence management object (EntityManager). The middleware rwSensorData can use Sensordata entity object as data transfer unit to achieve the basic interfaces of massive data such as read, write, delete, and modify operation.

4.2.1 Writing massive data

The monitoring data is stored into massive database in decentralized order and by parameters. As soon as data transited from sensors to server-side, the framework program (skeleton) of entity class rwSensorData will call the set method of SensorData, pass the data value from sensors to SensorData object at first, then call the persist method of data persistence framework to store the data into the massive databases taking the SensorData entity object as unit. Each of the "field" is individually deposited, the entire process of writing data is completed in a concurrent and distributed environment, and therefore there is no data congestion.

4.2.2 Reading massive data

In application layer, using JPQL persistence language, which is consistent with the relational database can read the massive data from Oracle NoSQL. The underlying operation is completed by data persistence framework through calling Oracle ONDB API for data reading.

4.2.3 Other massive data operation

Similarly, based on the data persistence management object (EntityManager) of massive data persistence framework, the other operations such as modifying and deleting of massive data can also be finished, through calling the find method of the EntityManager. The object data can be located for modification by assign new value, or deleting the data by call the remove method.

5 Application effectiveness analysis

Finally, in this paper, the "tomcat 6.0.14+BlazeDS" technology architecture was used under the JavaEE environment, the multi-node storage, multiple copies of fault-tolerant and free extension of mass data cloud storage system was deployed based on Oracle NoSQL DB, and the typical crop production management decision platform prototype was researched and developed to gather the monitoring data from crop planting field and provide production guidance services for the users at the other network nodes.

5.1 CLOUD STORAGE SYSTEM DEPLOYMENT AND OPERATING RESULTS

In this paper, the cloud storage system was designed according to the persistence layer, logic model layer, distribution layer and interface layer, the specific deployment way and function are introduced as follows:

TABLE 6 Cloud storage system deployment

Component	Deployment	Function
Interface layer	Deployed the rwSensorData in Spring + BlazeDS framework	Provides resource pooling, connection pools
Distribution layer	Deploying 5 network nodes of Oracle NoSQL Nodes and Hadoop nodes, (1 master node and 4 slave nodes, the initial configuration data storage capacity of each node is 20T)	Automatically scale data in multiple node, maintaining the node data autonomy
Logical model	EclipseLink JPA Java Object -Map	Keeping the same modeling method with the relational database, the difference only exits in physical storage in the underlying interface.
Persistence layer	EclipseLink 2.4.1 JPA for Oracle NoSQL	Shielding the differences in the underlying interface of data provider, the application layer provides the consistent style of data persistence interfaces with the relational database.

According to the above design, when the monitoring data is received via PC or SMS module from sensor node to the cloud storage system, an idle instance from the interface pool (instance) at the server-side will be assigned to complete data storage operation. The distribution layer will store the receive decentralized data into different network nodes in real time, the data that has the same key value will be automatically stored in a partition (Figure 2), and the database system will store the copies of data according to pre-configured number of replication factor of the cloud storage system at the same time. The distributed storage was autocompleted by NoSQL and Hadoop due to the data access operation is responded by interface pool. The data operation will be completed in instantaneous, and there will not occur server goes down, slow response and stopped responding phenomenon.

Use data query interface of Oracle NoSQL, the data stored in a massive database can be seen as the following Figure 2:

```
1 [CUSTOMER, D5BA62AE-3356-4C0A-AE04-48DC6D07B893, ID]D5BA62AE-3356-4C0A-AE04-48DC6D07B893
2 [CUSTOMER, D5BA62AE-3356-4C0A-AE04-48DC6D07B893, NAME]AMCE
3 [KJCDIGITALMARKET, 19545, Boneless sheep hind legs]19545|Boneless sheep hind legs|18|18.5|19|General||2010-11-06 00:00:00|http://www.xinfadi.co
4 [KJCDIGITALMARKET, 19557, Beef heart]19557|Beef heart|5|5|5|General||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.php?mid=14amp;
5 [CUSTOMER, 481A8827-3DF2-41D8-8CD4-D850CD4C34D2, ID]481A8827-3DF2-41D8-8CD4-D850CD4C34D2
6 [CUSTOMER, 481A8827-3DF2-41D8-8CD4-D850CD4C34D2, NAME]Smith
7 [KJCDIGITALMARKET, 19554, Steak]19554|Steak|12|12.25|12.5|General||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.php?mid=14amp;
8 [KJCDIGITALMARKET, 19539, Sheep ribs]19539Sheep ribs|13|13.5|14|||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.php?mid=14amp;p
9 [KJCDIGITALMARKET, 19542, Mutton kidney]19542|Mutton kidney|20|20|20|||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.php?mid=14
10 [KJCDIGITALMARKET, 19543, Mutton kidney]19543|Mutton kidney|17|17.5|18|General||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.p
11 [KJCDIGITALMARKET, 19546,Boneless sheep forelegs]19546|Boneless sheep forelegs17|17.5|18|General||2010-11-06 00:00:00|http://www.xinfadi.com
12 [KJCDIGITALMARKET, 19550, Shelter]19550|Shelter|12|12|12|General||2010-11-06 00:00:00|http://www.xinfadi.com.cn/do/listform.php?mid=14amp;p
13 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, DATAID]1
14 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, STATIONID]1
15 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, ACQUISITIONTIME]2012-05-03 07:30:00
16 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, RADIATION]1700
17 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, AIRTEMPERATURE]15
18 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, AIRHUMIDITY]10
19 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, RAINFALL]0
20 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, CO2]26.1
21 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, WINDSPEED]21.5
22 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, WINDDIRECTION]SSW
23 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, SOILHUMIDITY]30
24 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, SOILTEMPERATURE]20
25 [SENSORDATA, FA18AE16-A43F-4437-861B-DF1990976782, VERSION]1
```


FIGURE 2 Data stored in the massive database system

In Figure 2, the data whose primary key is “SENSORDATA” can be think as its database table name is “sensordata” like in relational database, the minor keys “FA18AE16-A43F-4437-861B-DF1990976782” can be think as the row recorder number in the relational database, the same batch of data of radiation, temperature, humidity, rainfall, carbon dioxide (CO2), soil moisture, soil density, wind speed, wind direction and soil temperature have the same minor keys (for example, “FA18AE16-A43F-4437-861B-DF1990976782”) and the same version number. The name of the last minor key can be interpreted as the field names in the relational database, and key value followed by “field” is its specific data values. Comparison with the relational database, the data in massive database is organized as "one by one grain”, it hasn’t fixed format, and hasn’t constraints of ACID theory in the relational database, the style is very similar to the Hash table in the Java language, so as to the performance efficiency is very high, and it is easier to extend in storage.

5.2 PERFORMANCE EFFICIENCY ANALYSIS

In order to understand the cloud storage system efficiency intuitively, in this paper, a simple test environment of “64 Bits PC, 8G Memory, 3 × 3 Storage nodes, 10 concurrent visits” was deployed under the experimental conditions. The response time was observed respectively on the amount of data request of 20-160 records by the benchmarks test of cloud service, the test results are as follows (Figure 3):

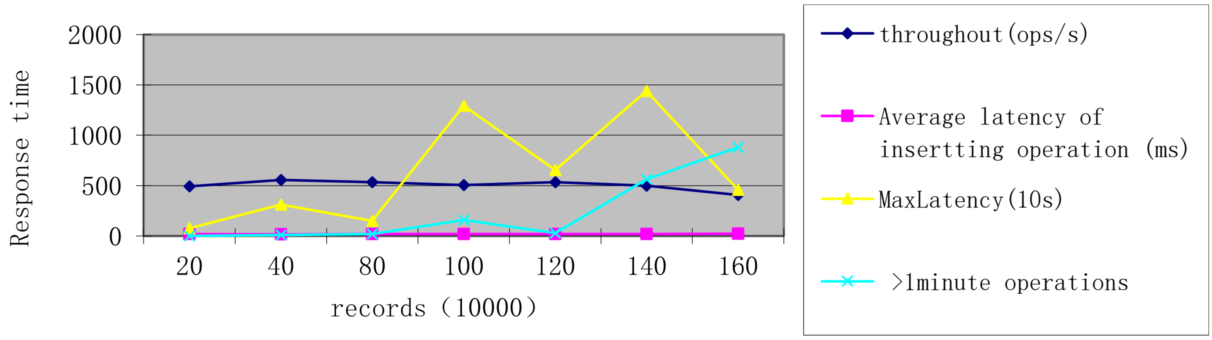


FIGURE 3 Massive database and relational database data access response time comparison chart

From above figure test case, to keep the high concurrent access throughout performance of the mass data as a whole, the mass database system stored the data “field by field” according to the data column. If the topology was designed reasonably, it always keep comparison smooth of data operation response performance. The average delay of insetting data insert shows slight fluctuations along a horizontal line, as the data scale of increased, the data throughput performance will decline, and the maximum delay time and the operations over 1 minute will increase, but all the data will be stored at some point later than the request time eventually.

6 Conclusion and discussion

Against the technical problems of traditional database on massive data processing, the cloud storage technology for typical crop growth monitoring data was researched in this paper. The data storage method was put forward based on Oracle NoSQL massive database, which can transmit and stored the huge data of radiation, temperature, humidity, rainfall, carbon dioxide (CO2) concentration, soil water, soil temperature, wind speed, wind direction from sensor network nodes to the cloud storage centres of typical crop production management platform respectively. Moreover, solved the efficient access issues of the "real-time, high-frequency transmissions, the fast-growing" massive monitoring data, the main findings was made as following:

1) The common clouds distributed storage systems was invested and analysed. The cloud storage method suitable for typical crop growth monitoring mass data was put forward using the storage framework of combination the Oracle NoSQL and Hadoop FS, and solved the bottlenecks of concurrent access of massive data by means of the distributed storage mechanism for the mass data among multiple network node.

2) In order to reduce the difficulty of migrating legacy applications to the massive database system application target, the cloud service framework for mass data access was raised to be consistent with the physically distributed storage system. The cloud service interface for mass data access was researched and developed to provide the technical support for the deep application of typical crop growth monitoring data resource.

3) The execution efficient of the cloud storage technology for typical crop growth monitoring data was analysed. Under the condition of the same data scale application, the mass data storage system has the characteristics of automatic expansion of the capacity and distributed concurrency computing in real-time, compared with the traditional relational database systems, it has more greater data throughput performance, more reliable operation, and can meet the challenges of mass data from typical crop growth environment monitoring.

Acknowledgments

This research was sponsored by the National science and technology support program (2013BAD15B04), the Beijing Natural Science Foundation (4122034) and major scientific and technological projects supported by Xinjiang Production and Construction Corps (2013AA004-1).

References

[1] Zeng X H, Wang Q 2011 General situation of agricultural information technology and network *China Agricultural Science Bulletin* **27**(8) 468-73

[2] Jia X F 2011 Research on the strategy for real-time information exchange of the mass data based on the Internet of things *E-Government* **4** 16-20

[3] Deng J G, Pan X H, Yuan H Q 2012 Research on cloud storage and its distributed file system *Journal of Dongguan University of Technology* **19**(5) 41-6 *(in Chinese)*

[4] Ning H S, Xu Q Y 2010 Research on global internet of things' developments and it's instruction in China *Acta Electronica Sinica* **38**(11) 2590-9

[5] http://www.enet.com.cn/cio/zhuanti/2012/bigdata

[6] Zou C H 2011 Research of adaptive local hyperplane K-nearest neighbour classification algorithm based on hadoop platform *South China University of Technology (in Chinese)*

[7] Yu X 2012 NoSQL: a rise in the great wave of big data *Network World* **4** 1-5

[8] Lu J H 2012 The Practice of Hadoop *Beijing: Machinery Industry Press (in Chinese)*

[9] Zhang H Q 2011 Relational database and NoSQL database, Computer Knowledge and Technology **7**(20) 4802-4

[10] Shen S 2012 Research on NoSQL Database Technology and Application *Nanjing: Nanjing University of Information Science and Technology (in Chinese)*

Authors

Yuansheng Wang, born on July 7, 1973, Wuhan, Hubei province, China

Current position, grades: associate professor of Beijing Research Centre for Information Technology in Agriculture.
University study: Ph.D. degree in cartography and geographic information systems, Beijing Forest University in 2008.
Scientific interest: agricultural and rural information resource integration, agricultural eco-environment, forest resources and forest fire prevention, agricultural LBS.
Publications: more than 10 journal papers since 1997
Experience: mainly engaged in the study of agricultural information technology since 2003.

Huarui Wu, born on July 21, 1975, Liaocheng, Shandong province, China

Current position, grades: professor of Beijing Research Centre for Information Technology in Agriculture.
University study: Ph.D. degree in computer applications, Beijing University of Technology in 2010.
Scientific interest: agricultural and rural information resource integration, agricultural Intelligent Systems, wireless sensor network technology.
Publications: more than 30 journal papers and co-authored several government reports since 2000.
Experience: mainly engaged in the study of agricultural information technology since 2001.

Lei Luo, born on August 10, 1987, Baoding, Hebei province, China

Current position, grades: engineer, Beijing Research Center for Information Technology in Agriculture.
University study: Master degree in computer application, Hangzhou Dianzi University, Hangzhou, China, in 2013.
Scientific interest: application of information technology in agriculture.
Experience: agricultural information technology engineer since 2013.

Weihong Ma, born on February 18, 1987, Xingtai, Hebei province, China

Current position, grades: Beijing Research Centre for Information Technology in Agriculture
University study: Master's degree in mechanical and electronic engineering, Beijing University of Technology in 2011.
Scientific interest: agricultural and rural information resource integration.
Experience: agricultural information technology engineer since 2011.

Kun Liu, born on December 18, 1990, Taian, Shandong province, China

Current position, grades: Software engineering student, Beijing Research Center for Information Technology in Agriculture.
University study: Master's degree, Shandong Agricultural University, Shandong, in 2014.
Scientific interest: research and application of information technology in agriculture.