HBase based storage system for the Internet of things

Yun Zheng[1, a], Chuanchang Liu[2, b]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China.

[a]zhengyun9006@163.com, [b] lcc3265@bupt.edu.cn

Keywords: Internet of things, storage system, HBase.

Abstract. Internet of things(*Iot*) continually produce a large amount of data , need to collect, process, storage, analysis and make use of these data. There are too many nodes in the Internet of things, the data is huge, and it needs a distributed data management system to manage and integrate. For the traditional distributed database system, the local fault and the data synchronization update brings the cost of the system performance, which makes the data quantity and the number of nodes are limited, and cannot meet the requirements of data management in the Internet of things. In view of the problem of the storage for the Internet of things, a two layer distributed storage structure is proposed, which uses the distributed database(*HBase*) to store the metadata of sensor data and *Mysql* to store the sensor data respectively. The experimental results show that the system has good scalability and query efficiency.

1. Introduction

With the development and expansion of the *Internet of things*, the data volume is growing, and the data storage becomes the key problem. One of the most important features of the *Internet of things* is the mass of nodes, in addition to the people and the server, goods, equipment, sensor networks, and other things are the composition of the node, its number is much larger than the Internet; at the same time, the data generated by the *Internet of things* is much higher than the *Internet*, such as the majority of sensor nodes in the full time working state. On the one hand, the data in the *Internet of things* is bound to require the backbone network to gather more data, the data transmission rate is higher; on the other hand, as the *Internet of things* is directly related to the real physical world, many cases need to access and control the corresponding nodes and equipment, so the high data transmission rate is needed to support the corresponding real-time.

With the rapid development of cloud computing technology [1], the research on mass data storage is becoming more and more extensive, distributed computing technology and *HBase* storage system based on *Hadoop* cluster [2], *HBase* is the open source implementation of *BigTable*. The sensor data type is monotonous, based on time sorting, so *HBase* is suitable for the information storage of the *Internet of things*. Building cluster storage management system with *Hadoop*, using *HBase* to store sensor data. To solve the problem of data storage in the cross region, design the two layer storage structure, according to the region to build the storage cluster, each region has a *HBase* cluster to storage data, and use *Mysql* to store storage information in various regions, and do not store the sensor data [3].

2. Key technologies

2.1 HBase.

HBase is a distributed, column oriented storage system that provides real-time read\write and random access to large data sets [4]. *HBase* automatically cut the table into different regions, and each region contains a subset of the rows of the table. *HBase* consists of a master node to coordinate one or more regional servers. *HBase* implementation relies on *Zookeeper* to coordinate management, *Zookeeper* is responsible for selecting a node for *Master*, the remaining nodes for region server. The *HBase* table is composed of rows and columns. The cells in the table are the intersection of rows and columns, and they are a version number. In default, the version number is the time stamp that is automatically assigned by *HBase* when the cell is inserted. The cell contents of the table are an array of non - interpreted bytes. Each column are grouped and form a column family, all the columns in the cluster members have the same prefix, group in the identifier to distinguish. Therefore, each column is represented as column family: qualifier.

2.2 Sensor network.

A sensor network has one or more sink nodes to collect data from sensor nodes, the interaction between the data and the application of the sink node through the gateway. Sensor network data storage is divided into internal data storage and external application storage, in this paper, we discuss the problem of data acquisition from the sensor, through the sink node, to external storage. The large scale of sensor network includes two aspects: on the one hand, sensor nodes are distributed in a wide geographical area; on the other hand, sensor nodes are deployed densely. In actual use, the sensor is widely distributed, large scale, the deployment is a regional, some sensors gathered in a region.

3. System architecture

According to the regional distribution of the sensor network, the data storage structure of the *Internet of things* is shown in Fig. 1. It contains three levels.

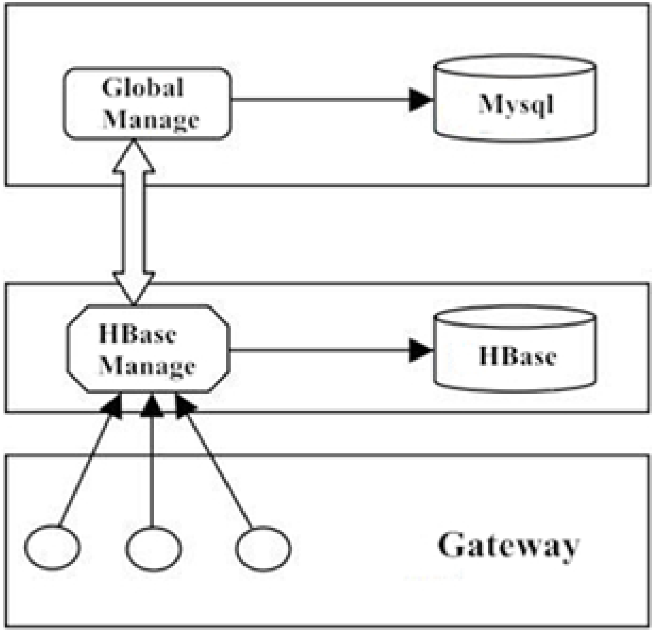

Fig. 1 System architecture

Gateway Layer: sensor data access layer, different sensor types have the corresponding gateway to receive and process sensor data. Application through the gateway, access to data collected by the sensor.

HBase Manage Layer: used to manage the storage of sensor data, acquisition of sensor data from the gateway, and stores the data in real time.

Global Manage Layer: manage and record the global information, the location of the sensor data distribution.

Due to large scale sensor distribution in the presence of regional, if all the area sensor data is stored in a *HBase* cluster database, it will lead to the consumption of network resources, and the result of long time consuming. Therefore, the data stored in each region is stored in the regional database server to reduce the consumption of network resources, improve data storage and access in real time. In this paper, the gateway is divided into regions, each region has its own *HBase* to store the gateway to collect real-time sensor data, *Mysql* to save the global information, records of all region *HBase*, gateway and sensor information.

4. System implementation

4.1 Table structure design.

The storage architecture designed in this paper needs to store two kinds of data tables: *Global* information table and sensor data table. Among them, the global information table includes all of the region *HBase* table, all the gateway information as gateway tables, and all the sensor information as sensor tables. These tables have only one column family, and qualify is the value of the attribute values. In each region *HBase* table, gid represents the only identifier, ip represents master address, port represents master port, capacity indicates the available storage space, location indicates the location of the location. In gateway table, gid represents the only identifier, name represents the name of gateway, location indicates the location of the location, *Rgid* represents region *HBase* identifier for storing gateway data. In sensor table, gid represents the only identifier, id indicates the local number of nodes, location indicates the location of the location, sensor type represents the type of sensor, data type represents the type of sensor data, sample rate indicates the sampling frequency, storage table represents table name for storing data, *Ggid* represents gateway identifier. Each sensor has a table to store the collected data, called sensor data table.

4.2 Data writing process.

The operation of the various tables in *HBase* requires the object of the *HTable* class, use put/get method to complete the insertion and read data operation. Use the *HBaseAdmin* object to complete the new table, delete and other operations. Data write is a collection of data collected by the sensor to the gateway, need to use *HBase Manage Layer* write operation to written to the region *HBase* specified table. Each sensor needs to allocate a table to store the data, the distribution of the table is applied to the *Global Manage Layer* by *HBase Manage Layer*, application form and table structure. *HBase Manage Layer* receives sensor data from each gateway data, need to store these data in a *Global Manage Layer* table, as shown below.

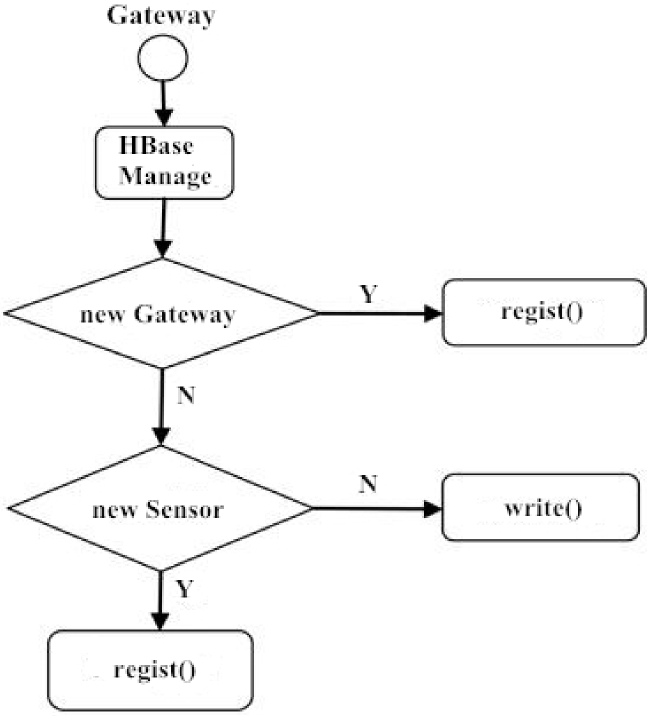

Fig. 2 Data writing process

4.3 Data access process.

Data access is the data information of the client visiting a region, *Global Manage Layer* maintains global information, the client needs to access the database address of the data in the *Global Manage Layer* query, then *Global Manage Layer* to the corresponding database server to submit the request to get data and return data.

5. Experimental results

In order to verify the efficiency of storage and query, and the scalability of *HBase*. use 25 sensors. The write and read time of the cluster is compared with the single machine and the configuration of 5 nodes. As can be seen from the graph, when the data volume is relatively small, the gap between the single and the cluster is small. With the increase of the number of sensors, the advantage of the cluster is more prominent, the growth rate of the writing time is significantly smaller than that of the single unit. *HBase* cluster has obvious advantages for large storage size data.

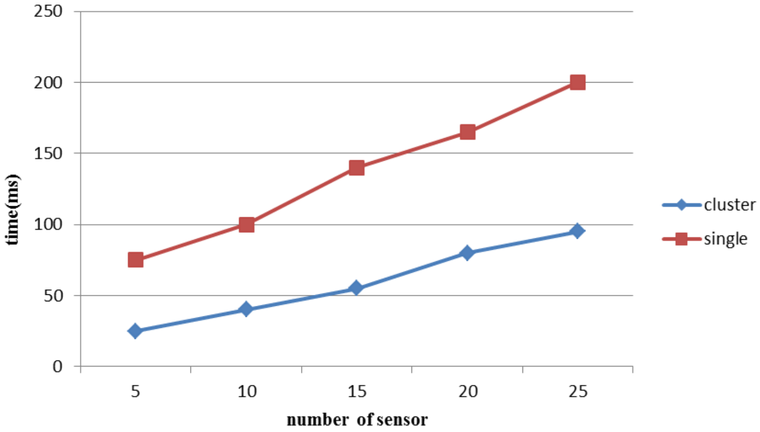

Fig. 3 cluster and single

6. Summary

This paper mainly analyzes the storage problem of large scale data in wireless sensor networks, due to the cross regional characteristics of the sensor distribution, a hierarchical storage architecture based on *HBase* is proposed. Through experiments, the *HBase* has good scalability, high efficiency and efficiency.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (Grant No.U1536112).

References

[1]. Zhang S, Zhang S, Chen X, et al. Cloud computing research and development trend[C]//Future Networks, 2010. ICFN'10. Second International Conference on. Ieee, 2010: 93-97.

[2]. Taylor R C. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics[J]. BMC bioinformatics, 2010, 11(Suppl 12): S1.

[3]. Ronstrom M, Thalmann L. MySQL cluster architecture overview[J]. MySQL Technical White Paper, 2004.

[4]. Shvachko K, Kuang H, Radia S, et al. The hadoop distributed file system[C]//Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010: 1-10.

[5]. Tudorica B G, Bucur C. A comparison between several NoSQL databases with comments and notes[C]//Roedunet International Conference (RoEduNet), 2011 10th. IEEE, 2011: 1-5.