Research and Implementation of Distributed Database HBase Monitoring System

Lisi Guo[1,⋆], *Tao* Chen[2], *Yanming* Chen[2], and *Xinlong* Luo[1]

[1] *Beijing University of Posts and Telecommunications, Beijing, China*
[2] *China Mobile Group Design Institute Co.,Ltd, Beijing, China*

Abstract. With the arrival of large data age, distributed database HBase becomes an important tool for storing data in massive data age. The normal operation of HBase database is an important guarantee to ensure the security of data storage. Therefore designing a reasonable HBase monitoring system is of great significance in practice. In this article, we introduce the solution, which contains the performance monitoring and fault alarm function module, to meet a certain operator's demand of HBase monitoring database in their actual production projects. We designed a monitoring system which consists of a flexible and extensible monitoring agent, a monitoring server based on SSM architecture, and a concise monitoring display layer. Moreover, in order to deal with the problem that pages renders too slow in the actual operation process, we present a solution: reducing the SQL query. It has been proved that reducing SQL query can effectively improve system performance and user experience. The system work well in monitoring the status of HBase database, flexibly extending the monitoring index, and issuing a warning when a fault occurs, so that it is able to improve the working efficiency of the administrator, and ensure the smooth operation of the project.

1 Introduction

With the development of technology, nowadays the companies are producing rich types of data at an unprecedented speed and scale. Data storage increase dramatically.To meet the demand for the huge data, the NoSQL database came into being. The NoSQL database has good reading and writing abilities, and flexible database data model.In the existing NoSQL database, HBase is a highly reliable, column oriented, and scalable distributed database with high performance,gaining the superiority in offline data batch processing. As a distributed database, HBase , which disperses the data originally stored in a centralized database to multiple data storage node, have faster data access,greater scalability, and higher concurrent traffic. Thus, it can better meet current needs for data storage.

A well-running database play an important role in the era of big data, so database monitoring becomes a very important part of database usage. There are two general ways of database monitoring:artificial monitoring and software monitoring.

Artificial monitoring tends to be delayed in detection of malfunction.It's susceptible to data loss, abnormal operation of project and other problems.Database error or stop working even for a short period of time may cause huge losses.In addition, this approach requires a lot of labor,which is inability to cope with the increased management difficulty and workload of the database.The other method is software monitoring.The main open source monitoring software for large data analysis system is Ganglia, Nagios/Centreon, etc. Existing monitoring systems for large open source components is coarse-grained. Acquisition ability of monitoring index can't meet all the needs of big data platform system. They can not concisely show indicators the enterprise most want to see and meet the new monitoring needs constantly emerging in practical applications.

In summary, the development of an on-demand HBase database monitoring system has a very important practical significance. This article will provide relevant researchers with our design and implementation ideas of an on-demand extension of the HBase database monitoring system. On one hand, by means of modulating monitoring indicators of different type, this system can flexibly adding monitoring indicators by configuration file based on the actual needs of the project.On the other hand, It can detect alarm situation regularly and inform administrators immediately when problems occur, which can significantly improve the efficiency of management staff.Besides,Through the charts in front-end which show the performance of the latest data and historical data, managers can fully aware of the changes of database performance.

2 Research Objective and The logic Principle of Design

2.1 General Idea

The end-to-end monitoring to each node of the distributed database HBase is the premise to realize automatic opera-

⋆Corresponding author:greece_gls@163.com

tions and maintenance. Its implementation is based on collecting key data from the HBase sub-nodes and the server host where they reside.

There are three dimensions of key data information can be acquired: the database performance index (read and write traffic, downtime issues, etc.), the host performance metrics (CPU, memory, hard disk, etc.) and network connectivity (ping).

The monitoring of HBase database cluster operation can be achieved through managing, analyzing and presenting the collected data, thus guiding the storage of the new coming data according to the results. Therefore, the situation where the downtime and other serious consequences occur can be avoided, since the overloaded database or host will not continue to store data.

By setting the alarm rules and analyzing the collected data on the back-end, the risk management of the database can be realized. Once a server trigger a warning, the administrator would be notified by email to take reasonable steps to protect the normal operation of the database, ensuring the normal operation of the actual production system.

In all parts, data acquisition is the premise, centralized management and logical analysis of data is the core,and data application is the purpose. The implementation architecture of HBase monitoring system is shown in Figure 1.

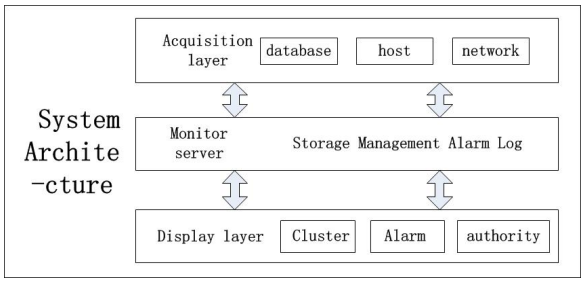

Figure 1. the architecture of HBase monitoring system

Based on the goal of building a distributed database HBase monitoring system and the actual operation and performance requirements of the project, the system is supposed to possess the following two capacity.

First, the ability to monitor the normal indexes of each sub-node of HBase database.

Second, the capabilities to manage and analyze data on server side.

2.2 The Monitoring Ability

The normal index of HBase database is the basis of monitoring system. The normal indexes of collected include physical hardware index, database performance index and network connectivity index.The normal indicators are shown in Figure 2.

2.2.1 physical hardware monitoring

CPU utilization, disk usage, memory usage and other hardware information on the server are monitored in the

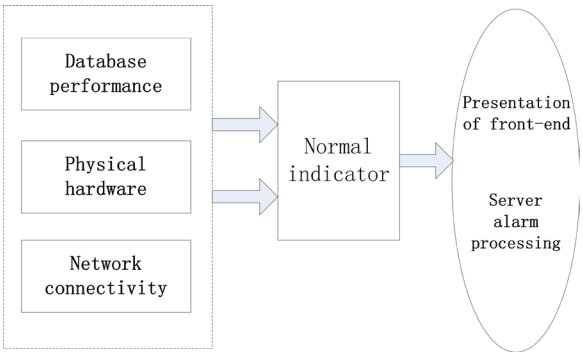

Figure 2. the normal indicators

physical hardware monitoring. The server is the foundation of all business operation. Once the physical hardware fails, the server can't provide services normally, and it even causes very serious consequences, such as server downtime.

2.2.2 database performance monitoring

These indicators include HRegionServer's operating status(lived or not), reading and writing visits and so on. These information presents the running state of the database, and provide services for database optimization.

2.2.3 network connectivity monitoring

The normal network is a prerequisite for the operation of a distributed database. Therefore, smooth network status is essential to the normal operation of the database.

2.3 The data management and analysis capabilities

The server mainly undertakes four tasks: data processing, fault alarm and logging.

2.3.1 data processing

After receiving the reported data, the server would store the data into the database respectively according to their different types. The front-end acquires the data from the server through the corresponding interface. The server would make a summary of the data per hour, obtaining the total average of that hour. In addition, to prevent database overloading, the system would remove data two days ago.

2.3.2 fault alarm

The server would initiate the alarm check task according to the specified frequency or time node. If the alarm threshold is reached, the system will automatically send a mail to notify administrator for subsequent processing.

2.3.3 Logging

The server would regularly record the daily operation of the database every day so that the administrator can check its running history. The server would record the time and details as long as an alarm occurs, otherwise record that the Hbase is running well today would be written.

3 Key Technologies and Key Issues

3.1 proxy implementation

The client-side deploys the proxy to collect data on a recurrent acquisition and initiative report basis.

The proxy is an extensible data acquisition module under modularized management, which can be used to expand the new functions according to the actual needs and configure the start-stop of various types of data in the configuration file. It adds types of indicator data to the task queue by different jobs, regularly initiating the data collection tasks and reporting to the server in accordance with the specified frequency or time node. The physical hardware information acquisition modules need to be deployed to the corresponding server. The specific process is shown in the Figure 3.

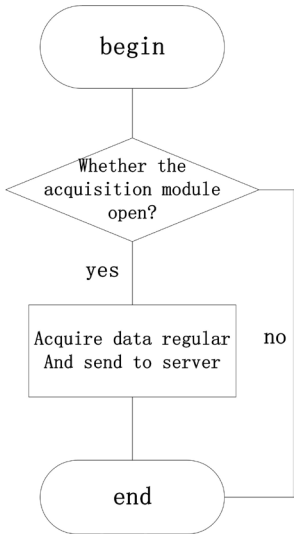

Figure 3. Data collection flowchart of the proxy

Among the normal indicators of the system, the physical hardware indicators are collected by Sigar API, database performance indicators are acquired through the HBaseAdmin interface, and the network connectivity is obtained through the ping connectivity detection.

3.2 the implementation of server

The server side adopts SSM architecture to develop. As a lightweight framework with simplicity, testability and loose coupling, the framework Spring has the features of Inversion of Control(IOC) and Aspect Oriented Programming(AOP). SpringMVC is more customizable because it separates the role of controller, model object, dispatcher and handler object. MyBatis provides the persistence layer framework which include SQL Maps and Data Access Object (DAO),almost eliminating the JDBC code and manual configuration parameters.It uses XML for configuration and original mappings, mapping the interface and common Java objects to records within the database, thereby unifying the database query management. Besides, the database adopt MYSQL in this system. The timing tasks use quartz for periodical summary, alarm detection and logging. Mail system uses the Postfix server, which can send the mails after finishing the configuration on corresponding server.

3.2.1 database table design

The database includes the following four tables:Monitordata, Monitordatahour, Cluster and Alarmdata.The table Monitordata storages the all monitor data.The table Monitordatahour makes summary for the monitor data per hour.The table Cluster storages the information of cluster.And table Alarmdata records the information of alarm data.Details are as shown from Table1 to Table4:

Table 1. the monitordata

column name	column discription	example
eventid	id for event	5365
objetid	type of monitoring data	hbase
timestamp	monitoring timestamp	2017-07-5 11:23:30
location	IP of monitoring location	10.254.201.204
content	monitoring value	{”serverName”:”NBG-204”, ”serverStatusSign”:”on”, ”memoryMap”: {”total”:23414,”used”:6666}, ”visits”:0.0, ”ReadRequestCount”:43457384, ”WriteRequestCount”:357045, ”hbaseTest”:”true”}

Table 2. the monitordatahour

column name	column discription	example
eventid	id for event	5365
objetid	type of monitoring data	hbase
timestamp	monitoring timestamp (that hour)	2017-07-5 11:00:00
location	IP of monitoring location	10.254.201.204
content	monitoring value	{”serverName”:”NBG-204”, ”serverStatusSign”:”on”, ”memoryMap”: {”total”:23414,”used”:6642}, ”visits”:0.0, ”ReadRequestCount”:43275952, ”WriteRequestCount”:357045, ”hbaseTest”:”true”}

Table 3. the cluster

column name	column discription	example
id	id for event	1
hostip	IP of server	10.254.201.201
role	the role that server contains	hbase
hostname	name of the server	”NBG-201”

Table 4. the alarmdata

column name	column discription	example
eventid	id for event	5333
objetid	objective of alarm data	cpu
timestamp	alarm timestamp	2017-07-5 11:05:30
location	host IP of alarm occured	10.254.201.204
alarmname	alarm name	high cpu usage
alarmdetail	alarm details	cpu usage over 90%
level	alarm level	high
status	alarm processing status	send email

3.2.2 The key technologies of the SSM architecture

The control layer (Controller), which is responsible for data management, is the core of server-side. The main tasks of the control layer are: Parsing and storing received data into the database, and directly using corresponding Java interface of the mapper layer to process data in the database; Providing data interface to the front-end. The control layer passes data to the front-end by post. It sends parameterless request when the front-end requests for loading the first page data, while it sends request with parameters when the front-end requests for certain types of data.

The Service is a processing layer for business logic, which is mainly responsible for alarm checking and logging. Alarm detection and log records will be triggered regularly. Basic warning rules are listed below. 1. Hardware: the CPU,memory or disk is too high. The threshold can be set by the administration.2. HBase: Once down, the alarm will be sent. 3. Network connectivity: A warning will be sent when there is a host fail to ping. Warning rules can also be increased on the front-end according to actual demands. Log records are carried out daily. The time and details would be recorded when alarms occur. Otherwise it records the good condition that day.

The Mapper layer provides the mapping between Java interfaces and SQL, and the original SQL queries are all defined in the Mapper’s XML file.

3.3 The resolution to the key issues

The rendering speed of the web front end has an extremely important impact on user experience. In the actual development of a project, excessive SQL queries will be time-consuming, resulting in slow page rendering and poor user experience. Therefore, it is of great importance to reduce SQL query statements to improve system performance and user experience.

In this system, because the monitored database is a distributed database, three different sets of data exist if there are three server hosts. Each group of data is divided into multiple types of data, such as host data, HBase data and ping data. Supposing that different types of data were acquired by SQL queries respectively, getting all the information from the latest database would go through 3 * 3 times SQL queries. If the cost of one query is 0.25s, it will cost 0.25 * 9s when the front end send requests to the server asking for data.Excessive SQL queries bring too much time to consume. Therefore, by using one SQL query to get all types of data from one host and distinguishing various types on the server side, we can reduce the time overhead. The time overhead this method requires is 0.25 * 3s, which is one third of the previous one. The front-end takes less than one second to render, while it costs more than two seconds before, thus the user experience is greatly improved.

4 Results

Among three months of actual operation, the system worked well to detect the latest state of the host, the database, and the network. It can also longitudinally perceive the changes of the database operation by checking the historical data. Besides, the system have done well in alarm warning, which effectively improved the working efficiency of the administrator. Part of the front-end interface is shown in the Figure 4,Figure 5 and Figure 6.

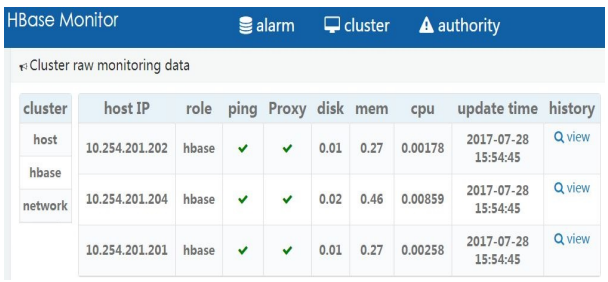

Figure 4. host information

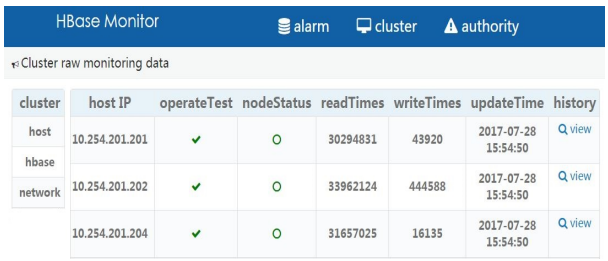

Figure 5. HBase information

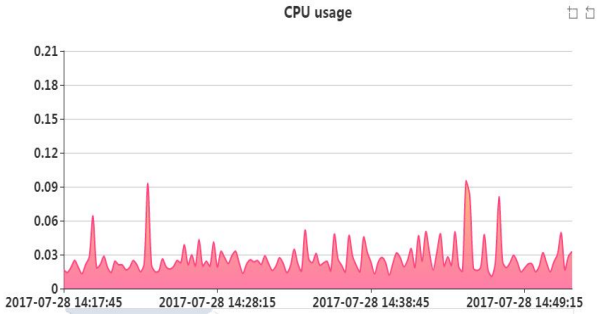

Figure 6. the historical cpu usage

5 Conclusion

In recent years, with rapid growth of data quantity, the distributed database have begun to play an important role. Monitoring database becomes important means to assure the safety of data storage and smooth operation of project. This paper dealt with a certain operator's actual demand of the distributed database HBase in its projects. We have designed and realized an on-demand HBase monitoring system which can extend monitoring indexes flexibly. In addition, given the low loading speed in the process of practical operation, we work out a solution to reduce the SQL query, providing a reference for related researchers. Practice shows that reducing the SQL queries can effectively improve system performance and user experience. The system works well in monitoring the HBase database's running state and sending alarms in time, which improve the efficiency of the administrators and ensure the smooth running of the projects. Nevertheless, this paper only involves data collection, management and presentation. The application of data remains further research. Studying the historical monitoring data and guide the fault prediction would be a new direction which is worth exploring.

References

[1] Bartosz Balis,Marian Bubak,Daniel Harezlak,Piotr Nowakowski,Maciej Pawlik,Bartosz Wilk. "Towards an operational database for real-time environmental monitoring and early warning systems[J]," Procedia Computer Science,2017,108.

[2] Jun Wu Xu,Jun Ling Liang. "Research on a Distributed Storage Application with HBase[J]," Advanced Materials Research,2013,2200(631):

[3] Mavromichalaki, H.,Papaioannou, A.,Plainaki, C. et al."Applications and usage of the real-time Neutron Monitor Database[M]," Advances in Space Research: The Official Journal of the Committee on Space Research(COSPAR),2011,47(12):2210-2222.

[4] Seung Kim,Nam Wook Cho,Young Joo Lee,Suk-Ho Kang,Taewan Kim,Hyeseon Hwang,Dongseop Mun. "Application of density-based outlier detection to database activity monitoring[J]," Information Systems Frontiers,2013,151:.

[5] Bartosz Balis, Robert Brzoza-Woch, Marian Bubak, Marek Kasztelnik, Bartosz Kwolek, Piotr Nawrocki, Piotr Nowakowski, Tomasz Szydlo, Krzysztof Zielinski. "Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities", Future Generation Computer Systems (2016) In press.

[6] Anonymous. Noble Gases; Application Security, Inc." Announces Database Activity Monitoring for DB2 on IBM z/OS[J]," Computers, Networks & Communications,2011.

[7] GaoYong-ping,Guan Fen-fen."Based on MVC to Monitor the Performance of Oracle Database[C]," Proceedings of 2012 international conference on electronics information and electrical engineering.2012:325-327

[8] Chun-ying WU,Li-yan ZHANG,LieCAO et al."Distributed Spinning Frame Monitor System Based on DSP and PROFIBUS[C],"//2010 2nd International Conference on Computer Engineering and Technology (ICCET 2010). v.3.2010:1872-1875.