ELSEVIER
Available online at www.sciencedirect.com
SciVerse ScienceDirect
Physics Procedia 25 (2012) 860 – 867
Physics Procedia

2012 International Conference on Solid State Devices and Materials Science

Preliminary Research on Distributed Cluster Monitoring of G/S Model

Yang Wenhui, Meng Antai, Miao Fang, Tan Li

Key Lab of Earth Exploration & Information Techniques of Ministry of Education
Chengdu University of Technology
Chengdu, China

Abstract

In the G/S model, the distributed server cluster is responsible for providing spatial data management and information sharing. Monitoring is the main task of cluster management. This paper studied the implementation of current cluster monitoring systems and on this basis proposed solutions of G/S cluster monitoring, mainly make a new exploration on the storage of monitoring data – the monitoring information is stored in HGML files, and around the storage model achieved other functions the monitoring system needed, finally the prototype system was developed and the strategy was verified.

*Keywords:*G/S; distributed cluster; monitoring; HGML

1. Introduction

G/S(Geo-Browser/Distributed Spatial Data Servers) model is a new generation of Network connection model based on the request - aggregation - service mechanism, which provides diversity network services through the client-side dynamic aggregation of spatial information, Compare with the traditional C/S and B/S model, G/S model not only combines the advantages of both the C/S model making full use of client resources, processing client data efficiently and B/S model having unified Client side, showing data bounteously, but also improves the complexity of Information processing and efficiency of providing information service in B/S model.

In the G/S model, G side provides users with spatial data information display, calculation, service aggregation; S side as spatial information server side, also known as spatial information cloud, is the foundation of whole G/S model architecture, which provides background services for various applications

doi:10.1016/j.phpro.2012.03.169

of G side and uses the distributed server cluster technology by which stores and manages massive spatial information for achieving efficient data management and security classification.

In the S side, with cluster supervision by monitoring system, the administrator can grasp the current running status of cluster, detect system failure point and deal with system problems, monitoring information not only can be used as reference information for task scheduling and load balancing, but also can be analyzed for finding performance bottlenecks exist in the system and taking further optimization measures. In short, the S-side monitoring for improving cluster system performance, reliability and usability is very important and necessary.

2. Design

The main objective of cluster monitoring is to collect monitoring information from each node, such as CPU utilization, memory utilization, disk utilization, network traffic and etc. The design of cluster monitoring system is mainly focused on:

- Getting more types of monitoring information.
- Providing monitoring information storage for queries on historical data.
- Providing visual graphical interface to display real-time running state of cluster system.

2.1 monitoring system structure

View from the monitoring server side, cluster monitoring system is mainly composed by master monitoring system deployed in the monitoring server and monitoring agent deployed in the monitored nodes. According to different ways of monitoring agent gathering monitoring information, the structure of monitoring system can be divided into direct structure and hierarchical structure, these two styles of monitoring system structure is shown in Figure 1:

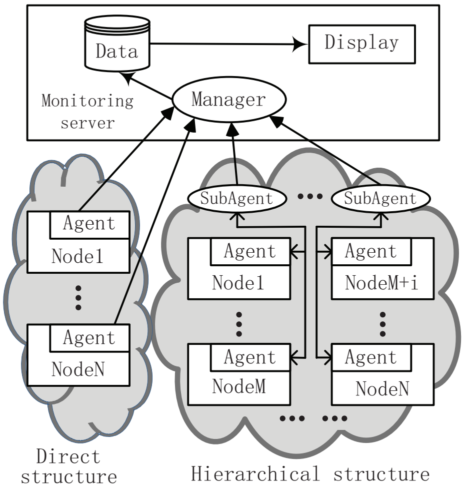


Figure 1. Two types of monitoring system structure

Monitoring server is responsible for collecting, storing and displaying monitoring information. In direct structure, each monitoring agent communicates with the monitoring server directly, monitoring agent reports node status information to monitoring server, receives instructions issued by monitoring server and triggers corresponding adjustment operation. In hierarchical structure, all of monitored nodes are divided into groups, there is an independent local monitor node as subagent in each group, the working mechanism of each group is similar to the way of direct structure, the subagent handles the affairs of its own group and communicates with higher level of subagent, the monitoring information and Management instructions are transmitted between monitoring server and monitored node through each level of subagent hierarchically. By doing so the master monitoring server should monitor the entire cluster system.

The choice of direct structure or hierarchical structure in applications should be based on various factors, these factors include the type of monitoring management , monitoring data sampling frequency, monitoring data quantity, the Internet throughput and the type of monitoring data, Figure 2 describes how to judge what kind of structure should be used:

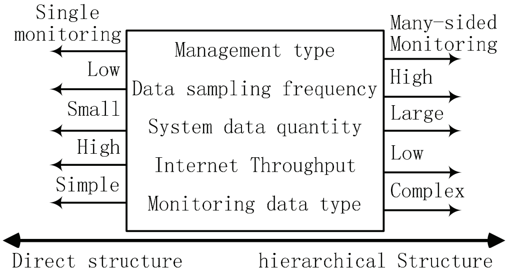


Figure 2. Reference factors of structure options

In this paper, because we are doing small-scale experimental applications at elementary step, we choose direct structure to develop a prototype system quickly to test and see the effect, when in the subsequent complex application of large-scale we will use the hierarchical structure to achieve good monitoring results.

2.2 monitoring data acquisition

In the Linux/UNIUX system, there are many methods for monitoring information acquisition, each sampling method has its advantages and disadvantages, but commonly used method for monitoring information acquisition are discussed as follows:

- The acquisition method based on Shell command. Linux/UNIX operating systems provide Shell command to user for viewing system information and executing the script, so the monitoring agent executes corresponding command with Shell and gets result the command returned, then analyzes the results and obtains the required monitoring parameter, this method is simple and there are no incompatibilities between different operating systems, the main flaw is that when a Shell command is executed, the operating system will create a new processes to deal with, which invoke a new system call, so this method increases a certain expenses on the monitored node, frequent use of Shell command will directly affect the performance of machines and result in efficiency reduction in cluster system.
- The acquisition method based on /proc file system. /proc file system is a virtual file system dynamically maintained and updated by Linux/UNIX kernel, running in the core kernel mode, it loaded in the /proc directory by operating system kernel when the operating system started. The /proc file system includes many system information, such as CPU running status, memory usage conditions, file system usage status, network status, process status and other parameters about machine, so by reading the corresponding information files can get the current system parameter information. The advantage of this method is the system overhead is small and there is less influence on cluster system. The disadvantage is that the format of /proc file system is related with the operating system kernel, it is difficult to develop a common monitoring agent.

Because the operating system kernel of S-side server node in G/S model is impossible to be consistent, in general conditions, we selected the way of Shell command to collect monitoring data. However, when the number of cluster node using the same operating system kernel is large, we can select the method based on the /proc file system to reduce the expenses of monitored node.

2.3 Monitoring data transmission

There are two type of communication protocols can be chosen for transmitting monitoring data from monitoring agent to master monitoring, they are described as follows:

- Pull protocol. Monitoring server sends collection command to each monitored nodes when monitoring server needs monitoring information of monitored nodes, once monitored node receives command, it sends corresponding monitoring information back to the monitoring server immediately. This model of communication protocol is simple to design and has good robustness in use, if one side of communication doesn't response to the communicating message as far as time-out error occurs, the retransmission mechanism will ensure the communication successfully. But to complete a monitoring information communication needs to transfer two kinds of information, one is the request command for information, the other is the response message monitoring system needed, because just the response information is valid, this model will bring a certain delay, and take up unnecessary Internet bandwidth.
- Push protocol. In contrast with the pull protocol, the monitored nodes push their node parameter proactively to the monitoring server. This communication model improves the real-time of information exchange and reduces the amount of network traffic. The disadvantage is that monitoring server is in a passive state, at some point it needs to accept a great deal of monitoring information coming from different monitored nodes, which can cause large resource consumption in monitoring server and be prone to a monitoring information loss situation.

Both methods have advantages and disadvantages, so we choose different transmission protocols for different information types. In this paper, push protocol is applied to periodic monitoring data such as node status information which monitored node periodically reports to the monitoring server, and pull protocol is used for data which monitoring system needs to request to get from monitored node. In this way, the system will become more efficient and occupy a small network bandwidth.

2.4 Monitoring data storage

After the monitoring server obtained monitoring information, monitoring information needs to be stored for historical data query and report forms generation. Study throughout the current implementation of monitoring system, the method of monitoring data storage is diverse: using traditional relational database, using self-developed in-memory database, using RRDtool (round-robin database tool) database, for instance. In this paper, by using HGML file to store monitoring data, combined with the function of the file system of operating system, we realized the mechanism of monitoring information organization and monitoring information maintenance, because the monitoring information of each monitored node is stored in respective HGML file, it has a great convenience and flexibility on information integration, sharing, exchange and etc.

3. Implementation of Prototype System

Through the above analysis, refer to the direct structure of cluster monitoring in figure 1, we realized a prototype system of S-side cluster monitoring in G/S model, The specific implementation of each module are as follows:

3.1 Agent Module

Agent module every 2 seconds to do the followings: calls the relevant Shell command, gets resulting string Shell terminal returned, extracts useful monitoring parameters by analyzing the format of string, combines all the monitoring parameters into a message, and then sends the message to the specific port of the manager module by TCP protocol.

3.2 Manager module

Manager module is responsible for monitoring information storage and maintenance, mainly composed by DataCollect and DataJudge two functional modules:

1) DataCollect Module

DataCollect module receives monitoring information from each Agent module, each node information is stored in a single HGML file which is named as the monitored node's IP address, take file 192.168.72.139.hgml for example, this file represents the information of monitored node whose IP address is 192.168.72.139.

The operation procedure of this module is as follows:

- The main progress listens on a specific port using Socket communication protocol, once finds that there is a Agent requesting to establish TCP connection with DataCollect module, agrees connection and starts the storage thread immediately which runs in a child progress to deal with communication and store the monitoring information, mean while, the main progress as parent progress returns and still waits for other request of connection monitored node launched. So the parent progress is spatial for treating all of communication request launched by monitoring agents incessantly, and the specific information storage of each monitored node is entrusted to a single child progress.
- In the storage thread, firstly, it establishes TCP connection with the Agent, obtains the IP address of Agent, receives monitoring message and then disconnects connection, secondly, stores monitoring information by following steps: looks for whether there is corresponding file IP.hgml matched with the IP address of monitored node in the file directory, if exists, then updates file with receiving information; else If only the file IP_stop.hgml exists, it indicates that the corresponding monitored node was out of service formerly and is on the line again now, so deletes file IP_stop.hgml, creates a new file IP.hgml and saves information; apart from above two conditions, that is to say, there is no related file exist with current IP address, it indicates that the host reporting the monitoring information is new on-line and in the range of monitored targets, creates file and saves information.

Through the above mechanism we completed to receive and store monitoring information, moreover, we can ensures the real time property of monitoring system, the specific flow chart of this module is shown in Figure 4:

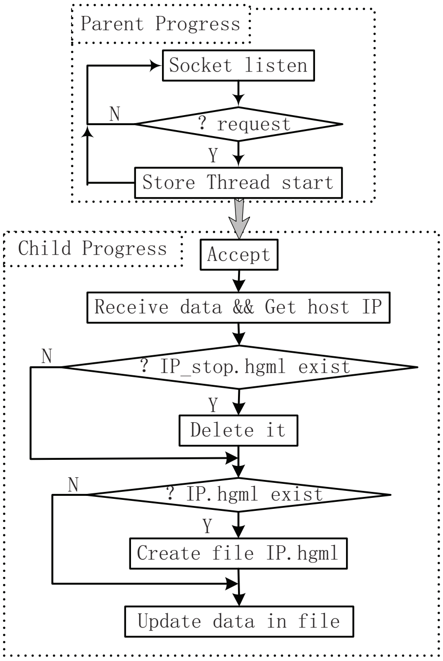


Figure 3. the workflow of DataCollect module

2) *DataJudge Module*

DataCollect module just carries out the monitoring information storage and update, when a monitored node doesn't send monitoring information to DataCollect module, it is probably by monitored node failure, in this situation, we need to mark the failure node, and so is DataJudge module's task. The general solution idea is that: to judge whether an IP.hgml file is updated recently we can know whether the corresponding monitored node works normally at current time. Therefore, DataJudge module mainly realizes the following functions and requirements:

- Regularly scans all of the modification time of IP.hgml file. Because each Agent module sends monitoring information every 2 seconds, taking the delays of communication and writing files into account, scan interval should be greater than 2 seconds to ensure that each scanned file is updated and not be greater than double monitoring data transmission interval (slightly larger than 4 seconds) to ensure that if the file was updated, it should be scanned, so the system uses 3 seconds as the scanning interval.
- on the assumption that the current scan time point is t (in seconds), if there is IP.hgml file whose modification time isn't in [t-3, t] period of time, can assert that the corresponding monitored node has failed and renamed IP.hgml file as IP_stop.hgml, such as 192.168.72.69_stop.hgml indicates that the monitored node whose IP address is 192.168.72.69 is currently unavailable.

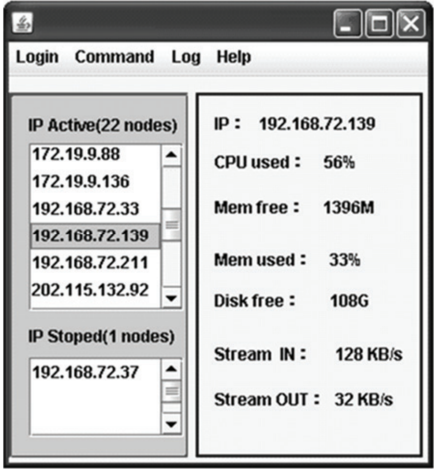
Through above mechanism, at any moment, when to gather the statistics of current running state of cluster system, all of monitored nodes can be sorted into normal or abnormal working group easily by scanning all filename of HGML because of the characteristic of filename we have defined.

3.3 Display Module

Display module shows cluster monitoring information to people and by which the administrator carries out operations to manage the cluster system, in this system, information display is only relevant to all of HGML file and its main achievement is as follows:

- Every 3 seconds to scan all of HGML filename, obtains the classified lists of IP addresses by judging whether filename is IP.hgml or IP_stop.hgml, displays two kinds of monitored node's IP address in interface panel.
- When the administrator needs to see real-time monitoring information of a monitored node, such as clicking the node's IP address on the graphic interface with mouse, program will look for the corresponding IP.hgml file to read context and display the monitoring data every 3 seconds.

Final design of a simple control interface is shown in Figure 4:



IP Active(22 nodes)

172.19.9.88
172.19.9.136
192.168.72.33
192.168.72.139
192.168.72.211
202.115.132.92

IP Stoped(1 nodes)

192.168.72.37

IP : 192.168.72.139
CPU used : 56%
Mem free : 1396M
Mem used : 33%
Disk free : 108G
Stream IN : 128 KB/s
Stream OUT : 32 KB/s

Figure 4. the running interface of Display module

4. Summary and Outlook

This article introduces the implementations of cluster monitoring prototype system in G/S model, preliminarily realized the mechanism of monitoring information collection, information storage and information maintenance. Using HGML document to store and exchange data, greatly reduces the coupling between various applications, and is beneficial to expend new functions for each module. In the future applications, we should strengthen the program robustness and fault tolerance of system, on this basis, it should provide automatic alarm mechanism, and we plan to develop cluster scheduling system based on monitoring information stored by HGML document.

5. Acknowledgment

The authors wish to acknowledge with appreciation the financial support (Grant No. NSFC 61071121) provided by the National Nature Science Foundation of China.

References

[1] Miao Fang. Research on Spatial Information Network Service Model of Service integrated on Client [R]. National Natural Science Fund Application, 2010.

[2] Miao Fang, Ye Chengming; Liu Rui. Discussion on digital earth platform and the technology architecture of digital China, Science of Surveying and Mapping, Vol. 32, pp. 157-158, November 2007.

[3] Fan Juntao, Li Guoqing. Practical cluster monitoring system. Computer Engineering and Design, vol. 29, no. 2, pp. 190-192, January 2008.

[4] Xue Zhenghua; Dong Xiaoshe; Wu Weiguo; HU Leijun. Design and Implementation of an Adaptive Monitoring System for Large-Scale Server Clusters. Journal of Xi'an JiaoTong University, vol. 42, no. 4, pp. 399-403, April 2008.

[5] Zeng Yalin; Guo Bin. Design and Implementation of GUI Software Based on Java [J]. Computer and Modernization, no. 10, pp. 116-117, October 2005.