Design and Implementation of Server Monitoring System Based on SNMP

Wenxian Zeng
College of Information & Technology
Hebei University of Economics & Business
Shijiazhuang, P.R.China
zengjmd@sina.com

Yue Wang
Information and Network Management Center
North China Electric Power University
Baoding, P.R.China
bdqy071100@163.com

***Abstract*—Considering servers being the core position in network, this paper introduces how to monitor servers through simple network management protocol (SNMP). We expand MIB resources by defining MIB objects to monitor the resources of sever, and use multi-threading technology to collect data and process them, which can improve the collection efficiency. The experimental results prove that it is a successful way of integrated monitor and control for servers based on SNMP.**

Keywords-snmp; server monitoring; net-snmp; winsnmp; multi-threading

I. INTRODUCTION

With the increasing expansion of the computer network and communication size, network has become essential to communicate for people's daily life. At the same time, the network monitoring has also developed rapidly. At present, the majorities of network management software are focused on the link and network equipment, but server should also be given adequate attention for being the carrier loading network services. In this paper, we introduce a server monitoring system, which not only monitors hardware and software of the system, but also monitors the security of the server's information.

This paper is organized into five sections: in the next section we introduce server monitoring based-on SNMP, in this section we briefly describe the SNMP management model, and analysis the monitoring context; in section 3 we design the system framework; in section 4 we present the system development in detail, including the agent expanding and the implementation of the data colleting module; in the last we give some conclusions of this paper.

II. SERVER MONITORING BASED-ON SNMP

At present, there are two types of network management protocol in computer network management field, which occupy the dominated position. One is common management information protocol and service (CMIP/CMIS), which is proposed by OSI organization. And the other is SNMP, which is put forward by IETF. The two protocols are corresponding to two different management programs. SNMP has become the most popular network management protocol for its simplicity and scalability. Almost all of the network equipment manufacturers support SNMP.

A. SNMP-based Management Model

The model of network management that is used for SNMP includes the following key elements [1]: Management station, Management agent, Management information base (MIB), Network management protocol (SNMP), which are shown in Fig. 1.

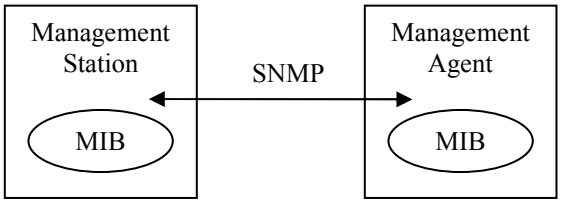

Figure 1. The SNMP-based management model

The management station serves as the interface for the human network manager into the network management system. The management station will have, at minimum: a set of management applications for data analysis, fault recovery, and so on; an interface by which the network manager may monitor and control the network; a protocol by which the management station and the managed entities exchange control and management information; a database of information extracted from the management databases of all the managed entities in the network. Only the last two elements are the subject of SNMP standardization.

The management agent responds to requests for information from a management station, responds to requests for actions from the management station, and may asynchronously provide the management station with important but unsolicited information.

In order to manage the resources in a network, these resources are represented as objects. Each object is, essentially, a data variable that represents one aspect of the managed system. The collection of objects is referred to as management information bases (MIB), which are written in a language called Structure of Management Information (SMI) [2]. SMI is a data-oriented language based on Abstract Syntax Notation 1 (ASN.1).The MIB functions as a collection of access points at the agent for the management station; the agent software maintains the MIB. A management station performs the monitoring function by retrieving the value of MIB objects.

978-0-7695-3615-6/09 $25.00 © 2009 IEEE
DOI 10.1109/JCAI.2009.34

The management station and agents are linked by a network management protocol, which includes five SNMP messages: GetRequest, GetNextRequest, SetRequest, GetResponse and trap. All the first three messages enables the management station to retrieve or set the values of objects at the agent, are acknowledged by the agent in the form of GetResponse message. In addition, an agent may issue a trap message in response to an event that affects the MIB and the underlying managed resources.

B. Analysis of Monitoring Context

Server monitoring context includes the following four areas: static information (hardware description, software description, administrator, physical location, etc.), dynamic information (interface traffic, usage of CPU, memory and disk, etc.) network services (HTTP, FTP, DNS, SMTP, POP3, SQL Server database, etc.) and network performance.

MIB-II and host resources MIB can monitor the state of hardware and software, and the running state of the system. We also want to achieve a comprehensive server performance monitoring, but the standard MIB can not meet the need. So we can expand the agent by adding MIB database files which are defined by ourselves according to the SMI standard, in order to expand the MIB.

III. FUNCTION DESIGN OF SYSTEM

The monitoring system is based on SNMP's manager-agent model [3, 4]. We use layered structure method to design the system according to the different functions of the system. The system should include the following modules, which are shown in Fig. 2.

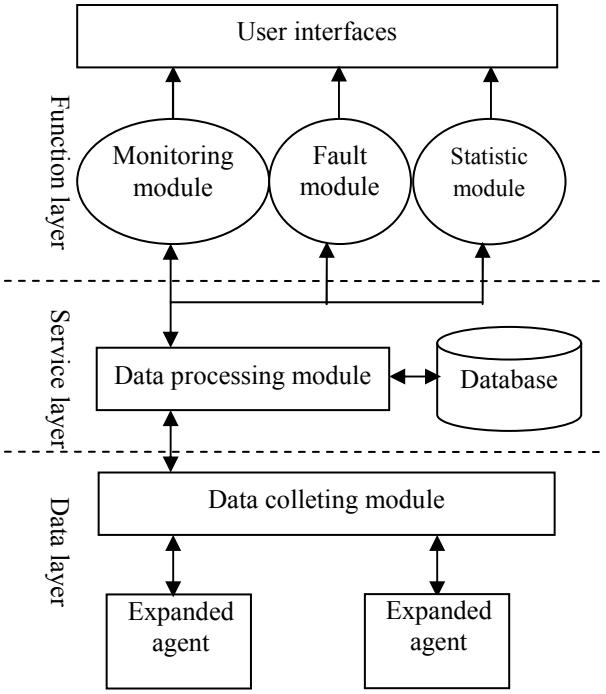

Figure 2. Framework of the system

We design the system into three layers: data layer, service layer and function layer.

The data layer at the bottom is responsible for the communication between manager and agent, require and set the information in the MIB. Here we use two ways to collect data. One is real-time collection, which collets the real-time information that need to be displayed and sending them to the upper layer in time. Another way is timing polling, which collects information regularly during an interval and sends them to the upper layer.

The service layer in the middle layer is responsible for dealing with the collected information. If the collected information needs to display timely, it will be directly send to the upper layer to display, or it will be stored into the database for querying the history information.

The function layer in the upper layer is the interface for the administrator operating the system. It can show the monitored server's configuration information, performance information, and fault information in a visual and graphical interface.

IV. IMPLEMENTATION OF THE SYSTEM

The development environment is Windows 2000 server, which encapsulates the implementation of the SNMP protocol, and provides a set of interfaces for developing network management programs based on SNMP, called WinSNMP API [5]. We use the SQL Server 2000 as our database and choose VC++ as the development tool.

A. Agent Expansion

We choose net-snmp agent to expand the MIB. We can define the MIB objects according to SMI standards and add them into net-snmp agent. The detail steps are as follows.

First of all, we design our own MIB files according to SMI standards. Secondly, we compile the files, and use mib2 tool to generate the procedure framework, and then improve the framework program. At last, we recompile the files, install the new agent and run it.

B. Data Collecting Module

The collecting program runs on a single manager station (a computer or workstation). The colleting process creates four threads. They are sending thread, receiving thread, preprocessing thread and storing thread. The four threads are controlled by a main controlling thread. Sending thread is responsible for sending SNMP request message. Receiving thread is responsible for receiving SNMP response message. Preprocessing thread is responsible for filtering and integrating the collected flow information. Storing thread is responsible for storing data into database, which are preprocessed by preprocessing thread.

This module is implemented by using WinSNMP API. The programming steps are as follows [6, 7].

1) Initialization of WinSNMP

a) Calling SnmpStartup function to start WinSNMP

b) Calling SnmpCreateSession function to create a session

c) Calling SnmpSetRetransmitMode function to set retransmission mode

d) Calling SnmpSetRetry function to set retransmission times

e) Calling SnmpSetTimeout function to set timeout time

2) Creating varbindlists

a) Calling SnmpStrToOid function to create OID, which we want to request

b) Calling SnmpCreateVb1 function to create varbindlists

c) Calling SnmpSetVb function to add variable into the varbindlists

3) Creating PDUs

We should call SnmpCreatePdu function to create PDUs, when the varbindlists has been created. We must set error_index parameter, error_status parameter, and request_id parameter. They are necessary.

4) Sending message

a) Calling SnmpStrToContext function and SnmpStrToEntity function to create community and entity.

b) Calling SnmpSendMsg function to send message

5) Receiving and processing message

a) Calling SnmpRecvMsg function to receive data

b) Calling SnmpGetPduData function to get the data from PDU

c) Calling SnmpCountVb1 to get the number of variables in varbindlists

d) Calling SnmpGetVb function to obtain variable's OID and its value in varbindlists

e) Calling SnmpOidToStr function to convert OID into string

f) Converting smiVALUE type data into many common data types data, such as int, char and so on, in accordance with their syntax member variables.*

6) Releasing resources

a) SnmpFree<xxx> function: releasing Entity, Context, Pdu, Vb1, Descriptor

b) SnmpClose function: closing a session

c) SnmpCleanup function: releasing all of the resources, it must be called before the end of the process.

The system running state is shown in Fig. 3. It shows the CPU utilization of one host. We can notice the original value of the CPU utilization at every acquiring moment and its changing rate.

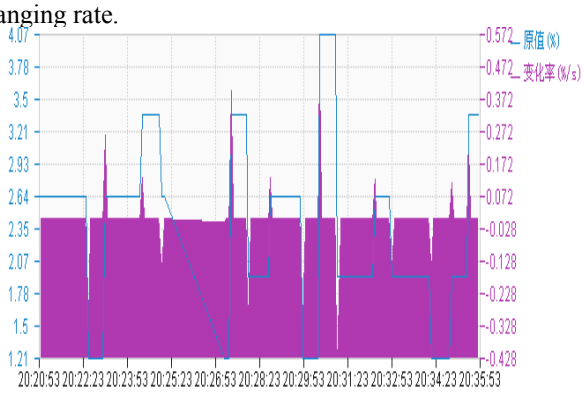

Figure 3. CPU utilization test figure

V. CONCLUSIONS

Considering the advantages of SNMP, such as simple, flexible, small network load, and its strong expansion, we develop the system which can monitor and control servers under the condition of not affecting the server's load and its service performance. It also can make alarm to the administrator immediately when any abnormality occurs. The system runs well so far. It is very convenient for administrator to see the monitoring results because of the graphics display. So it is very meaningful to develop the server monitoring system.

REFERENCES

[1] Cen Xiandao,An Changqing. Network Management Protocol and Development [M].Beijing: Tsinghua University Press,1998.

[2] M. Rose and K. McCloghrie, Structure and identification of management information for TCP/IP-based Internets, RFC 1155, May 1990.

[3] Wang Peiyu, Yang Jiahai, et al. Using RDBMS in Network Data Colleting and Management. Computer Science.2000,27(12):36-40.

[4] Zhao Wanping, Lu Qing, Li Chunsheng. Design and Implementation of Whole Network Flow Analysis System Based on SNMP Protocol. Journal of Shanxi University of Technology. 2006, 22（4）: 63-65.

[5] He Peng, Qiu Jianlin, Gu Xiang. Design and Implementation of Remote Host Monitoring System Based on SNMP. Computer Engineering and Design. 2008, 29(13):3303-3312.

[6] Zeng Wei. Research on the Network Management Technology Based on WinSNMP. Journal of Wuhan University of Technology (Information & Management Engineering), 2007,29 (12):15-18.

[7] Yu Xiaogao. The Research on The Program Technique of Implementing NMS in WinSNMP API. Computer Applications and Software, 2007,24(1): 121-122,133.