A Realtime Monitoring Method for Cluster System Running State Based on Network

Wanqing Chi¹ and Wei Zhou²

¹School of Computer, National University of Defense Technology, No. 109 Deya Road, Kaifu District, Changsha Hunan, China 410073
Email: chiwq@nudt.edu.cn
²College of Foreign Languages, Changsha University, No. 98 Hongshan Road, Kaifu District, Changsha Hunan, China 410003
Email: jovialzwh@sina.com

Abstract. High Performance Computing (HPC) system running state monitoring is an important task. Small-scale cluster systems exist separately as one whole system or exist as a part of a large-scale HPC system. Such cluster systems are often the core of the entire system. Therefore, for such systems, there should be more timely and accurate monitoring of running state. However, various HPC system monitoring softwares currently focus on the aggregation of overall running and resource state, and cannot provide realtime monitoring, detailed information, and process control. Based on the existing single-machine real-time monitoring tools, this paper adds network-based information exchange protocol, multi-threaded parallel information acquisition service, extends information representation and display output style, adds remote process control, then carries out related tests, which implements a more efficient realtime running state monitoring tool for small-scale cluster systems.

1. Introduction
In the High Performance Systems, it is a necessary operation and maintenance management work to monitor running state for HPC system. Then there are some system softwares for monitoring HPC, such as Ganglia, nagios, cacti, Zabbix, and so on [1]. These softwares have good scalability and can be adopted according to the scale of system. But they mainly focus on the whole states of system resources, and the information during longer time, not realtime, for example, 10 seconds or longer. And the information is showed in the form of HTML page, lacks interactivity. In the fact the scale of HPC is often flexible, in addition to the conventional large-scale cluster system with more than hundreds or thousands of nodes, the small-scale cluster system (usually no more than 10 nodes) is more flexible. Such as the experimental system, small-scale business system, cloud computing platform, data processing platform, and even the subsystems of large-scale system, such as management services, login server cluster, file system service cluster and so on. This kind of system is small in scale, but it needs more real-time monitoring. System maintainers hope that they can quickly and real-time obtain the main system operation information in the cluster, such as system available resources, system load, number and status of system processes, and the use of system resources, etc., and even need timely control (such as termination) the processes in the system. In fact, for stand-alone systems, there is a very good real-time monitoring and management tool, such as the Procs toolkit [4]. It provides a series of functions such as real-time state monitoring, detailed information provision and process control for the system and process through the proc file system.
The monitoring work of this paper is to show the state of the system and the process that affects it in real time, which is conducive to the system administrator's first understanding and control, and make decisions when necessary.

2. System State and Resources
The resources of High Performance Computer system can be divided into hardware resources and software resources. Hardware resources include CPU, memory, network, storage and so on. Software resources include operating systems, system libraries, and system applications and so on. The function of software resources, especially the operating system, is to manage all hardware resources and abstract them, provide them to the upper software through the agreed unified interface, and ultimately use them to the users. Users use system resources through the process to complete the computation.

Process is the most basic and important concept in the operating system. It is a running activity of a program with independent functions about a data set. Its activities need to apply for and own system resources. It is a dynamic concept and an active entity. Processes need to use CPU, memory, network, storage and other system resources. The use of resources by processes changes the state of the system [2].

System state is a comprehensive reflection of various resource changes that have been changing during the operation and use of the system, which reflects the availability and stability of the current system. The monitoring work of this paper is to show the state of the system and the process that affects it in real time, which is conducive to the system administrator's first understanding and control, and make decisions when necessary.

3. System Structure
The monitoring system adopts distributed C/S architecture, which consists of two parts: monitoring service daemon process and local customer monitoring tools, as shown in Figure 1. Each server deploys a set of monitoring tools separately.

Monitoring service process is a server daemon running in the background. It supports multi-thread concurrency and can process multiple requests at the same time. It is mainly responsible for receiving and processing requests from other client monitoring tools. Its main functions include collecting and sending real-time running status of local servers and controlling local processes.

The monitoring tool directly provides the real-time running status of the cluster, sends requests to other service programs in the cluster, collects the system status, and displays it in real time in an intuitive and concise way, providing the function of controlling remote processes.

![Figure 1. System Structure](attachment:system_structure.png)

4. Key Technologies
The whole monitoring and management system need to solve the problems of resource representation, collection, transmission, synchronization, process control, display and a series of key technologies.

4.1. Resource Representation
The monitoring system needs to display the real-time running status of the cluster, including system load, the total number and status of current processes, CPU number, CPU load and other information,
memory usage. The process information is the core information, and other information reflects the overall state of the system. Then the monitoring system defines two main data structures as follows:

The general information structure summary_s can be divided into the following categories:

a) **System load**: average load of the system within the last 5, 10 and 15 minutes;
b) **Overall status of processes**: total number of processes; number of running processes; number of suspended processes; number of suspended processes; number of zombie processes and so on;
c) **CPU information**: the number of logical CPUs, the percentage of CPUs occupied by users, and the percentage of CPUs occupied by kernels;
d) **Physical memory information**: all available memory, used memory, free memory, buffer memory;
e) **Swap space**: all, used, idle and buffered;

The process details structure proc_d_s mainly includes the following information:

a) User name, process ID, priority, virtual memory size, occupied physical memory size, shared memory size and percentage, CPU usage percentage and time, process status, name (command or program name).
b) Process states include: D: uninterruptible sleep state; R: running state; S: sleep state; T: tracked or stopped; Z: zombie state.

4.2. Resource Information Acquisition Technology

The system is developed and operated on Linux system, and all information is acquired on the basis of process file system (proc). Proc is a pseudo-file system, which stores a series of special files, such as the current running state of the kernel. Users can view the system hardware, software information and the current running process information through these files, and even change some of them in order to change the running state of the kernel. These files include the core state information of the system, such as load, process, interrupt, device, memory and so on. Each process in the system corresponds to a directory named PID under proc, in which the information of the process is output as a file, mainly the file status. [5]

According to the system information and the format of status file in the proc directory, the monitoring software can obtain all kinds of process system information mentioned above by using standard file access interface and input and output functions.

4.3. Protocol Design

Client and service programs exchange information and control remote processes through the network, which is the important improvement of this monitoring system. According to the process of information and exchange, the following control protocols are designed:

- **Request 0**: The client shakes hands with the server for initialization.
- **Request 1**: Exchange of general information. The client sends out requests, and the service daemon collects the overall information and sends it back to the client. The unit of information exchange is the summary_s structure (mentioned in 4.1).
- **Request 2**: Exchange process information. The client sends out requests, and the service daemon collects all process details and sends them back to the client. The basic unit of information exchange is proc_d_s structure (mentioned in 4.1).
- **Request 3**: Control the remote process. The client sends a request, including the PID and control signals. The server sends the control signals to the specified process and returns the execution results.

In the information exchange protocol, we can choose the following communication protocols: connected TCP protocol, reliable transmission, long time; connectionless UDP protocol, fast speed, poor transmission reliability. It can be used according to the responsibility of the system and network status and monitoring requirements. In terms of reliability, the handshake protocol is designed by the monitoring software to ensure the integrity and correctness of the transmission information when the access to important information and the network communication of the system is unstable.
4.4. Configuration Management Technology
Flexible configuration management is an important means to improve the availability and versatility of complex systems. The configuration of this system mainly includes the following categories:
1) Applicability Configuration: This type of configuration for system deployment is an adaptation to the deployment environment, such as the list of servers in the cluster.
2) Running Configuration: This kind of configuration is used to adjust the working mode of the system, such as the protocol of information exchange (TCP/UDP), whether to shake hands, update timeout time, cycle interval time, etc.
3) Information Configuration: This kind of configuration is used to adjust the content of information exchange. Because this software is real-time monitoring and exchanges more information, in order to improve efficiency and flexibility, we should select the information we need to obtain according to actual needs, and design the function of information configuration selection and process neglect. Information configuration selection refers to the choice of exchanging general information and process information. Process neglect function refers to shielding some processes that are not concerned, reducing information transmission, improving real-time performance and reducing the impact on system load.

4.5. Information Aggregation Sorting Technology
The core activity object of the system running state is the process. The most information processed by this software is the process information. In the system implementation, all process information, including local machine, is stored in a huge array of structures. Each time the output is displayed, these processes need to be sorted according to the sorting basis of display, and then displayed. Generally speaking, there are more than 200 processes in a server, and the number of processes in several servers is easily close to 1000. Normally, this sort of scheduling consumption and time is acceptable, but for real-time monitoring of clusters, time consumption and system interference should be avoided as much as possible. The software adopts partial sorting method, that is, according to the maximum number of process information displayed in the current window, the process that meets this number is sorted out, that is, the sorting operation is stopped, which greatly reduces the time and system consumption.

4.6. Output Display Technology
The monitoring system finally provides concise and intuitive information display. Based on the original design intention of the system, in order to maintain the manager to quickly and timely obtain the system operation status and implement the necessary control, the system implements the output interface of text terminal window based on ncurses library, which can provide rich information in limited windows and facilitate operation and control. Ncurses is a function library that manages the display of application programs at character terminals. It provides users with a flexible and efficient API by encapsulating the original terminal control code (escape sequence). It provides a set of functions to control cursors, create windows, change the colour of foreground and background, and handle mouse operations. It is helpful for users to write applications under character terminal [6].

5. Operation of the System
The monitoring client cooperates with several service monitoring programs to monitor the running state of the cluster system.

5.1. operation of Monitoring Client
The client workflow consists of the following steps, as shown in Figure 2.
1) **Initialization**: Initialize the relevant data structure according to the configuration file, determine the composition of the cluster, initialize other configurations, and shake hands with other service monitors to confirm whether the other party is functioning properly;
2) **Local information acquisition**: according to the configuration file or command options, based on the proc file system, read the local overall information, process information;
3) **Remote information acquisition**: according to the configuration file, it communicates with other service daemons in turn to obtain relevant information and record it locally;

4) **Display output**: Display server names and overall status information sequentially. The client sorts all process information by default way, and determines the number of processes displayed according to the current window size. Then it selects the corresponding number of process structures in all process information structures, and output the main information in accordance with the format, including: server names, PIDs, user, CPU utilization and memory used, and so on. According to the window size it can hide the overall status information and process details separately.

5) Starting from step 2, the periodic cycle is executed.

![Figure 2. Workflow of Client Monitor](image)

![Figure 3. Workflow of Monitoring Server](image)

5.2. **Operation of Monitoring Server**

The server workflow consists of the following steps, as shown in Figure 3.

1) **Initialization**: Initialization of relevant data structures according to configuration files;

2) Enter the loop and start network listening at the agreed port, waiting to receive external service requests. When receiving an external request, server starts the service thread and continues to listen;

3) The service thread establishes communication with the requester and completes one of the following tasks according to the protocol each time:

   a) Obtain and send the overall state information of the machine;
   
   b) Get and send details of the local process;
   
   c) According to the received PID and signal, the specified signal is sent to the PID process.
   
   d) When receiving the signal of the network shutdown, the service thread terminates.

6. **Experiments and Analysis**

6.1. **experimental Environment**

This system mainly faces small-scale cluster, so a test environment composed of 3 virtual machine servers is constructed. Each server CPU has a main frequency of 2.3GHz, including four threads,
30GB memory, 2197MB switching partition, Gigabit Network card, and RHEL 7.3 operating system. The names of the three servers are svr1, svr2 and svr3.

6.2. Function Test and Analysis

Test 1: As shown in Figure 4, the real-time running status of the cluster can be acquired and displayed correctly. Run the monitoring command on svr1. From the output, we can see that the monitoring client shows the overall status information of the servers in the cluster. In the process details section, the process is sorted according to the utilization rate of CPU. Besides xfsaild and sysmon, the monitoring client also lists the busy processes on svr2 and svr3. The monitoring service daemon process sysmond is to access the proc directory of the host to collect system status and process information in response to the request of the monitoring client of svr1, and send it back to svr1 through the network.

Test 2: Remote control process. By monitoring the client, you can show the process that occupies the most resources in the cluster. You can also stop the process on a server by using this tool. In the interface, you can input K, then give the server name, process PID, signal. If it is a local process, you can send the input signal to the process, if it is on other servers. The process notifies the service monitoring program on the corresponding server to implement the operation. After testing, the process on the local machine and other servers can be aborted correctly.

6.3. Performance Test and Analysis

Test 1: It tests the CPU occupancy and memory usage of the monitoring service daemon and client program. Several clients are started on svr1, and communicate with server daemon on other servers. The client on svr1 can get the information of those other service daemons on the local computer. After testing, it is found that the peak CPU share and memory usage of service monitors are related to the number of clients running at the same time, but both of them have a relatively short peak. From the
analysis of its operation process, it is mainly the time of acquiring system resources and sending them. After measurement, this time is generally within 0.5 seconds, the peak CPU utilization rate is 9%, but the maintenance time is also very short. According to observation, the stability value is generally 1%-3%, the average is 1.2%, which has little impact on the system. This is also need of the monitoring system. In contrast, when the tests are carried out without network, the CPU usage is about 6%.

**Test 2:** It tests the response time of the test client to obtain information from other servers. In this test environment, the average delay of the current monitoring client to obtain information from the other two servers is about 1.1 seconds.

### 7. Conclusion

In this paper, we design and implement a real-time monitoring tool based on network communication for small-scale cluster based on single-computer monitoring tool. It makes full use of the functions of single-computer monitoring tool, adds the functions of network communication and protocol design, and consumes little resources. It has little impact on the system, and it is convenient and effective to be deployed, and has the advantages of high rate, good real-time, concise and rich output information. The next step is to optimize resource reading and information transmission, optimize the display interface, refine information display and enrich management functions.

### 8. Acknowledgments

This work is partially supported by the National Key Research and Development Program of China (2016YFB0200401), by program for New Century Excellent Talents in University, by National High-level Personnel for Defense Technology Program (2017-JCJQ-ZQ-013), by the HUNAN Province Science Foundation 2017RS3045.

### 9. References

[1] Li C 2014 *Linux Enterprise Application Case Elaboration (2nd Edition)* (Beijing: Tsinghua University Press)

[2] Peng Y 2016 *Linux Operating System Case Course* (Beijing: Machinery Industry Press)

[3] Jones M 2006 Using the proc file system to access the contents of the Linux kernel ([https://www.ibm.com/developerworks/cn/linux/l-proc.html](https://www.ibm.com/developerworks/cn/linux/l-proc.html))

[4] Sourceforge 2009 Procps - The /proc file system utilities ([http://procps.sourceforge.net/faq.html](http://procps.sourceforge.net/faq.html))

[5] Qiu T 2016 *Course on Application and Development of Linux Operating System* (Tsinghua University Press)

[6] Gu Y et al. 2010 *NCURSES Programming HOWTO Chinese Version 2* (Urumqi: Software College of Xinjiang University)