Early fault warning mechanism based-on association rules in server clusters management system

Yumei Ning*, Zhenguo Ding, Ping Zeng

School of computer science and technology, Xidian University, Xi'an 710071, Shaanxi, China

Received 1 June 2014, www.cmnt.lv

Abstract

With the rapid development of Internet and business internal information technology, the problem of server cluster fault early warning becomes important in server clusters management. To solve the problem, a server cluster fault early warning system (SCFEWS) based on event association rules is presented. This system is mainly based on the fault event association tree. First obtain event relationship and association rules. Construct early warning events associated tree with system logs and association rules, and store it in binary tree linked list. Then by using the warning event filtering algorithm, redundancy fault early warning events are filtered out and only source event for early-warning notification are kept. Experiment shows the proposed algorithm can effectively improve the accuracy of fault location for server clusters management.

Keywords: Server Cluster, Early Warning, Fault Association Tree, Warning Event Filtering Algorithm

1 Introduction

With the rapid development of Internet and business internal information technology, enterprise server clusters are gradually used to store a big data processing to meet growing business needs. The server clusters size and the number of servers upgrade and expand constantly with the development of enterprise business. But an increasing number of server components makes server clusters management difficulty and improves the probability of server clusters fault at the same time, which bring tremendous challenge to the entire server clusters manageability. So it has great significance to improve the cluster service availability about how to guarantee the stability of the server clusters system and how to find and promptly remove the hidden dangers in cluster system before fault occur.

Now various faults in server clusters are usually detected and located manually lacking server cluster fault warning management mechanism. So it is the priority problem monitoring real-time status for server cluster resources and giving corresponding warning to the administrator. With faults warning system, administrator can monitor server clusters comprehensively, detect server clusters anomalies timely, locate the fault timely and accurately, take action to avoid or reduce the risk and prepare for effective response [1]. Nowadays, although the major IT giants have their own server clusters management system, it does not open to all. And there is no universal solution for server cluster faults warning.

Early Warning Systems play a highly critical role in monitoring, prediction and reaction on upcoming disasters [2]. Early Warning Systems are found in many areas: natural catastrophic detection [3, 4] of tsunami, flood and earthquakes; detection of sudden and significant economic changes [5]; engineering and scientific area [1, 6], etc. In spite of variations between monitored domains, their main goal is similar: to reduce economic losses and mitigate the number of deaths from disasters by delivering information which allows people and organizations to prepare for emerging disasters [1]. So Early Warning Systems can also be used in server cluster management. Upcoming faults can be monitored and predicted by using Early Warning Systems.

This paper presents a server cluster management system based on the idea of Early Warning Systems. In a Server Clusters Fault Early Warning System (SCFEWS) put forwarded in this paper, fault warning module not only monitor real-time status for server cluster resources but also send early-warning information to the administrator before a fault occurs. It guarantees the reliable and stable running and the processing big data of the server clusters system.

In this paper, SCFEWS based on the warning event tree is used to filter out triggered fault early warning events and retain only source event for early warning notification. And the destination is avoiding the redundant warning events.

The remainder of this paper is organized as follows: Section 2 describes the formulation of the fault early-warning problem in server cluster system. Section 3 gives a precise problem statement and details our algorithm of the SCFEWS. Section 4 is the experimental evaluation in a few respects: its performance and the validity. And the Section 5 concludes the paper with a summary of our findings.

* ***Corresponding author*** e-mail: ssss@dddd.com

2 The proposed algorithm

This paper presents a server clusters fault early warning system in network resource management, which relies on client agents [7, 8] and provides early warnings against equipment failure such as CPU utilization, memory utilization, disk space, etc.

2.1 EARLY WARNING SYSTEM

The basic idea behind early warning [1] is that the earlier and more accurately we are able to predict potential risks, the more likely we will be able to manage and mitigate disasters.

An effective early warning system needs an effective communication system. Early warning communication systems are made of two main components:

(1) Communication infrastructure hardware that must be reliable and robust.

(2) Appropriate and effective interactions among the main actors of the early warning process.

Communication infrastructure for SCFEWS uses intelligent mechanisms [9] for communication and scatter of fault alerts through agents on which the server clusters can be remotely monitored and managed. SCFEWS use association rules tree for fault prediction. With them SCFEWS can better guarantee the server clusters running stable and reliable.

2.2 EXPERT SYSTEM

Expert system is intelligent computing Machine application system that has a lot of specialized knowledge and experience. Expert System contains the following modules [10].

(1) The knowledge base, where the knowledge of experts is represented using rules, frames, semantic networks, and first-order logic based methods, etc.

(2) The working memory module that stores the input data and the information generated by the processing of rules.

(3) The inference engine where the processing of the rules and the reasoning of the Expert system take place.

(4) The user interface module facilitates the interaction between the user and the Expert system.

(5) The knowledge acquisition facility provides the user with appropriate useful tools during knowledge acquisition procedures.

(6) The explanation module allows the Expert system to present its reasoning regarding its conclusions.

The main components involved in Expert system are the knowledge base and the domain expert.

2.3 EVENT CORRELATION

Event Correlation is an important fault localization strategy. The basic idea is to correlate multiple events to a single concept to filter out unnecessary events and unrelated events, reduce the amount of information presented to the network administrator, and increase the semantic content of the information represented by the association process.

2.4 ALGORITHM DESCRIPTION

The thought of the SCFEWS is showed below. Please see Figure 1. Initial stage, there is small amount of the managed server clusters monitored by management centre. With the expansion of the managed server cluster scale, the probability of server cluster faults is increasing accordingly. Therefore, corresponding early warning notification should be sending out before a fault occurs. It is important finding usable system early-warning information from all monitoring information. Thus, with the warning notification system administrator can take early preventive measures rather than remedial measures before a server fault occurs. And that is the processing of SCFEWS. Thus, the system robustness and the scalability to manage server cluster fault are improved.

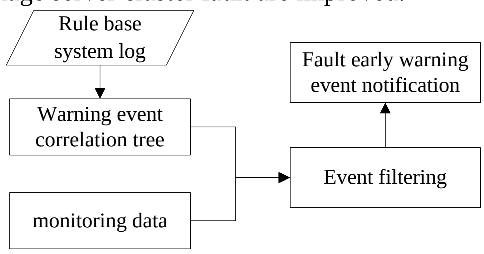

FIGURE 1 Structure chart of the SCFEWS

There are four mainly components in SCFEWS: the monitoring database, the warning event tree, event filtering module, and fault early-warning event notification module.

The monitoring data is comprised of server hardware such as temperature, fan, voltage, and power etc., system resources such as CPU utilization, memory utilization, disk space, and network speed etc., service resources such as processes, system services and application services etc. and network connectivity etc.

The warning event correlation tree is used to store the relationships between the various fault phenomena and is dynamically maintained for there is a certain correlation between various fault phenomena. The relationships data come mainly from the association rules for warning event and system logs.

Event filtering module: in server clusters, the generation of fault is often not a single fault phenomenon, but multiple fault phenomena occurring at the same time, that is to say cascading fault phenomenon caused by single fault. There is a certain correlation between fault phenomena, so event filtering module need to filter out triggered fault early-warning events and try to keep only source event for early-warning notification.

Fault early-warning event notification module sends the early-warning notification to system administrator for rapid positioning faults. And early preventive measures are taken to prevent fault risk caused by server clusters before a server fault occurs.

3 Problem solution

The Server Clusters Fault Early Warning System (SCFEWS) aims to filter fault early-warning events according to generated warning event tree and try to keep only source of information for early-warning event for notification. The system consists of a server-side proxy equipment, communication modules and early warning systems management console.

The follow is the specific approach. Real-time monitoring various state of system equipment via proxy server clusters, storing all kinds of the collected information, judging its status data according to the threshold value. When monitoring event is judged to be early warning event, it will be encapsulated into object-oriented fault early warning event and delivered to early warning filtering module. Warning filtering module will filter out triggered fault early warning events according to generated warning event tree, and only retain source early warning event for notification of early warning event.

The emphasis of the SCFEWS is early warning filtering algorithm and the warning event tree with which early warning events is handled with.

3.1 ALGORITHM DESCRIPTION

Usually there are three main methods for early warning judging. They are threshold judging, Status Judgment and shazam judging. This paper uses threshold judging and shazam judging. Early warning judging [1] model is mainly responsible for processing the data obtained from the condition monitoring model. It judges the data according to the threshold value and quick change. When monitoring event is judged to be early warning event, the detailed event information will be encapsulated into fault event and delivered to memory fault events waiting for subsequent processing by early warning filtering module.

Adopt object-oriented method [11] and construct quaternion form $E=(U,S,O,D,T)$ for storing early warning event-related information to descript the fault event in this paper. And symbol definition is as follows:

$U(U\text{ei})$: Identification of early warning events means such events is monitoring by the corresponding event listener for subsequent processing.

S=(Source) : Source of early warning events.

$O=(Object)$: Object of fault event. That is the fault device type.

$D=(D\text{escription})$: Fault description in detail, including fault identification, fault monitor and fault description, etc.

$T=(T\text{ime})$: Time for early warning events.

Early warning judging process is as follows:

(1) Get effectively monitor data from status monitoring model.

(2) Select the appropriate judging method based on monitoring data type.

(3) If the monitor data exceeds a threshold value or if it meets the fast changing limit, the monitor data is defined as early warning information and stored, else go to steps 1.

3.2 EARLY WARNING ASSOCIATION RULES TREE

When a fault event occurs in server clusters, it may cause other fault events closely related. There may be a relationship between them, such as time correlation, physical correlation [12], etc. Fault event sequence [13] is the combination of some fault events according to certain rules in a certain time interval.

Fault events associated tree is constructed based on the Fault event sequence in the rule base.

3.2.1 Get association rules

System administrators formulate relationship between warning events before server cluster fault early warning system is running, using algorithms inference log data and obtain relationship between warning events.

Association rules reveal the unknown dependencies between data items. According to the mining associated relationship, information from a data object can be used to infer the information about another data object [12, 13]. Describe as following.

Set $I=\{i_1,i_2,...,i_n\}$ is a project collection. X and Y represent item sets. X is the precursory item event and Y is the subsequent item event. Then Association rules is defined an implication relationship of the following forms: $X \to Y, where X \subset I, Y \subset I, and X \cap Y = \varnothing$.

3.2.2 Construct early warning events associated tree

Read the warning events association rules in rule base, according binary tree structure to construct and store the structure of early warning events associated tree. When management control starts, binary tree linked list is to be constructed and the warning events associated tree is to be generated according to the algorithm. Once relationship between early warning events in rule base change, the warning events associated tree need to be re-generated.

The node of early warning events is mainly composed of four components.

String *id* represents early warning event.

EventNode *child* represents the child node of early warning event.

EventNode *brother* represents the brother node of early warning event.

Variable *occur* represents whether the early warning event occurred. See Definition (1).

$$occur = \begin{cases} 0, \text{event does not occur} \\ 1, \text{event occur} \end{cases}. \quad (1)$$

Binary Tree linked list for the early warning event is structured as follows in Figure 2.

```
public class WarningNode{
    String id;
    WarningNode child;
    WarningNode brother;
    int occur;
}
```

FIGURE 2 Warning node class

3.2.3 The algorithm of warning event correlation tree

Construct the algorithm of warning event correlation tree depending on the warning events association rules in rule base. Define *endFlag* to be the tag of scanning rule database. See Definition (2).

$$endflag = \begin{cases} 0, \text{there exist unprocessed event} \\ 1, \text{scan end} \end{cases}. \quad (2)$$

The algorithm process is as follows:

(1) Generate Root node, set *endFlag* value to 1, and began to build children brothers linked list.

(2) Scan the rule base, add early warning event node whose precursor event is empty and subsequent events is not empty to the second layer of early warning events associated tree, and then set *occur* value to 1.

(3) Set *endFlag* value to 1, scan the rule base.

(4) If there is the early warning event node whose precursor event is not empty and *occur* value is 0, add the node to the sub-linked list of precursory early warning events node and set *occur* value to 1. If the node does not exist, the *endFlag* value will to be set 0 waiting for next scan processing.

(5) If *endFlag* value is 1, set all the warning events *occur* value to 0 within the linked list, scan end and the algorithm end. Go to steps (3) if *endFlag* value is 0.

3.3 EARLY WARNING EVENT FILTERING ALGORITHM

The early warning filtering algorithm need to depend on early warning association rules tree, and it mainly adopts the thoughts of expert systems. The aim of the algorithm is obtaining only source of information for early warning event for notification. Relying on early warning association rules tree, the algorithm filters all generated early warning events stored in filtering database, and eliminates redundant warning events.

The flow sheet for early warning filtering algorithm is shown in Figure 3.

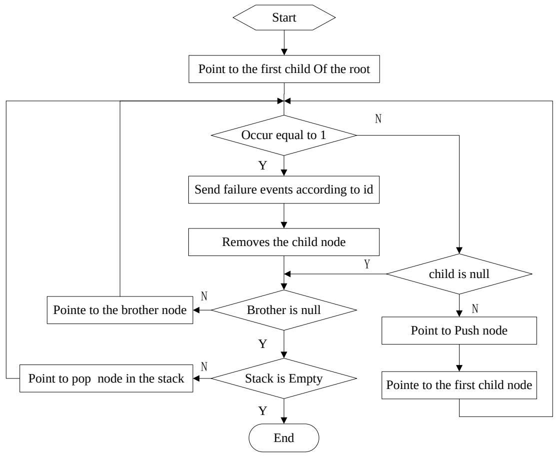

FIGURE 3 Flow sheet for filtering algorithm

The process of the early warning filtering algorithm description:

(1) Determine whether the event data in rule base changes.

(2) Call the early warning association rules tree algorithm and re-construct the warning event correlation tree, if the change exists. Call the warning event correlation tree constructed above, if the change does not exist.

(3) Traverse the warning event correlation tree, mark the warning event already stored in storage. Set variable *happen* value to 1.

(4) Traverse the warning event correlation tree using depth-first traverse principles. If there is a warning event whose precursor event does not occur or is empty, deliver it to early warning notification module. At the same time, if it has sub-event, remove all of its sub-events.

This paper uses a timer to trigger the early warning filtration module. The module will automatically carry out once filtering operation at regular intervals, and the administrator can set the interval. Finally the early warning event filtered will to be send to the warning notification module and there to be dispatched to administrator. Please see Figure 4.

```
LinkedList<WarnRule> events = new LinkedList<WarnRule>();
for(WarnNode w : events)
    markOccor(w);
Iterator<WarnEventRule> iterator = warnEventList.iterator();
filterEvent(iterator);
public void filterEvent(iterator<WarnRule> iterator){
    while(iterator.hasNext()){
        WarnRule occorRule=iterator.brother();
        if(occorRule.occor == true)){
            getWarnNodeSender().SendNode(changeWarnNode(occorRule));
            occorRule.child=null;
        }else  if(occorRule.child!=null){
            iterator=occorRule.child.iterator();
            filterEvent(iterator);
        }
    }
}
```

FIGURE 4 The early warning filtering algorithm

The early warning filtering algorithm can not only ensure the manageability of SCFEWS, but also ensure redundancy warning information not to be sent out.

4 Results

In order to verify the reliability of the Server Clusters Fault Early Warning System (SCFEWS), we programmed and simulated in the Network experimental Environment as shown in Figure 5. Management console was mainly consisted of management server and database server. Managed devices included five sets of managed servers, four managed PC and two mobile terminals. They were Lenovo IdeaCentre, AMOI N820 and IBM X5 respectively. The testing and analysis is only for server cluster fault warning system functional modules.

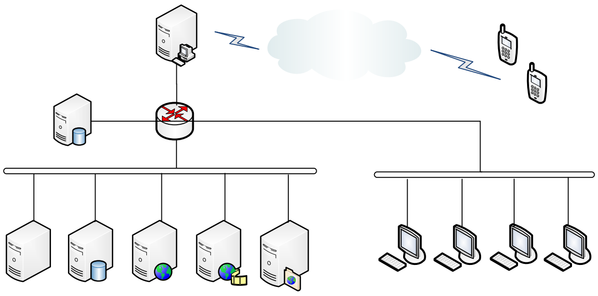

FIGURE 5 Network experimental Environment

For the convenience of testing the rationality and effectiveness of the model algorithm, the warning event correlation rules for the experimental system is part of preset in this paper as shown in Table 1. Program algorithm is omitted here.

The model algorithm is analysed from the performance, the warning filtration effect and the network utilization in the network experimental environment above.

TABLE 1 Correlation rules definition

No	Early warning UEI	Early warning description	precursory early warning	precursory early warning description
1	AgentDown	Client Agent shut down	DeviceDown	Server shut down
2	GetDataFail	Fault to getting data	AgentDown	Client Agent shut down
3	ServiceClosed	Host server is shut down	DeviceDown	Server shut down
4	DeviceUnreachable	Host network is unreachable	DeviceDown	Server shut down
5	ApacheDown	Apache Application stops	DeviceDown	Server shut down

Process is as follows: After the Device Agent of Managed device is launched, its various types of information is collected and incorporated into the monitoring database. If the monitoring data exceeds the defined threshold, then the event data is encapsulated into early warning and early warning is transmitted to the early warning filter module for pre-process based on the warning event correlation rules. And the early warning notifications being filtered are sent to the management client.

Experimental results show that the system is running continuously seven days in monitoring, during which the system produced a total of 44 early warning events. There are only 25 early warning events being warning information after filtering module filters the 44 early warning events, which is basically close to the original number of early warning events. Thus early warning filtration achieves a desired effect and the effectiveness of the warning filtering algorithm is proved in this paper. Please see Figure 6.

The following is the SCFEWS performance experimental.

Server is usually used to provide services. Because the proxy client software runs on the server equipment, and communicates with the management console, there must be low resource occupancy not to impact on performance of the server. So the CPU usage of proxy client and procedures network occupancy rate is experimented.

Server device can obtain CPU usage via top and other procedures when the proxy client software is installed on the managed device and turned on in the daemon. By setting the management console monitor polling cycle and sampling the CPU utilization, the CPU utilization of proxy client server is measured. Please see Figure 7.

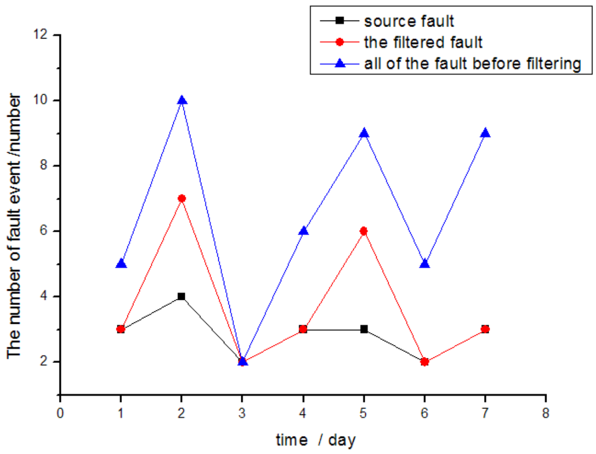

FIGURE 6 Comparison of early warning event

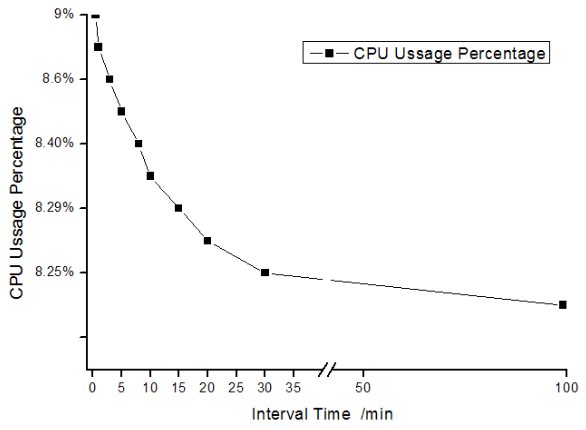

FIGURE 7 CPU utilization of server proxy client

It can be seen from the above figure that the monitoring polling cycle time is inversely proportional to the CPU utilization. Once polling cycle time shortens, CPU occupancy rate goes up. When proxy software is not running, CPU occupancy rate maintains at about 8.3%. When the proxy software is opened and the polling cycle is set to be thirty minutes, at this time, although management client does not send the command, the CPU utilization increases slightly for activity agent software monitoring ports and the activity agent process taking some of the resources. Along with shortened polling cycle, proxy software is more and more frequent to get early warning surveillance system data, so the value is also gradually increasing in CPU utilization. When polling cycle is set to half a minute, the CPU utilization maintains at about 9.0%. Thus the agent software on server equipment occupies very little CPU resources, and it will not have big impact on the server computing performance.

Server device can obtain the network rate via iftop and other procedures when the proxy client software is installed on the server device and turned on in the daemon.

Sampling the network speed during the agent software opening and the management end different monitoring polling cycle stetted, server agent can be analysed terminal network rate which is showed in Figure 8.

It can be seen from the above figure that the monitoring polling cycle time is inversely proportional to the network traffic. Once polling cycle time shortens, CPU occupancy rate goes up gradually. When proxy software is not running, network traffic maintains at about 0.28KB/S. When the proxy software is opened and the polling cycle is set to be thirty minutes, the network traffic decreases slightly around 0.31KB/S. Along with shortened polling cycle, network traffic is also gradually increasing. When polling cycle is set to half a minute, the network traffic maintains at about 0.8KB/S. Thus, the software routine has slight impact on the server computing performance when interval time is set a certain value.

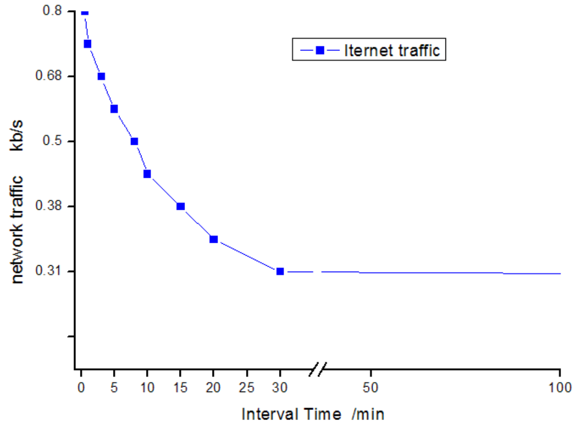

FIGURE 8 The network traffic of server proxy client

5 Conclusion

It can be seen from the above experimental data that the warning model algorithm presented in this paper has certain advantages in fault advance warning. The Server Clusters Fault Early Warning System (SCFEWS) designed in this paper integrating hardware status monitoring, system resource monitoring, early warning service monitoring, device control and device resource management has achieved the early warning and management of server clusters, also has not a major impact on the server computing performance. It greatly facilitates the operation and maintenance of the server clusters management. The algorithm has high value for reference to large-scale server clusters management.

The SCFEWS designed in this paper need to deal with system fault human involvement after the early warning events occur. Failover automatically and automatic downgrade functionality will to be introduced into early warning in the next research, trying early warning automated processing and restore.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (No.JY10000917003).

References

[1] Grasso V 2012 *Early Warning Systems: State-of-art Analysis and Future Directions* UNEP
[2] López V F, Medina S L, Juan F de P 2012 Taranis: Neural Networks and Intelligent Agents in the Early Warning against Foods *Expert Systems with Applications* **39** 10031-7
[3] Hong Y, Adler R F 2007 Towards an Early-warning System for Global Landslides Triggered by Rainfall and Earthquake *International journal of remote sensing* **28**(16) 3713-9
[4] Bartosz Balisa, Marek Kasztelnikb, Marian Bubaka 2011 The UrbanFlood Common Information Space for Early Warning Systems *Proc. Conf. on Computational Science* Japan 96-105

[5] Ng G S, Quek C, Jiang H 2008 FCMAC-EWS: A Bank Failure Early Warning System based on a Novel Localized Pattern Learning and Semantically Associative Fuzzy Neural Network *Expert Systems with Applications* **34**(2) 989-1003

[6] Nasonov D, Nikolay B 2014 Hybrid Scheduling Algorithm in Early Warning Systems *Proc. Conf. on Computational Science* Nevada, USA 1677–87

[7] Corchado J M, Tapia D I, Bajo J 2012 A Multi-agent Architecture for Distributed Services and Applications *International Journal of Innovative Computing, Information and Control* **8**(4) 2453-76

[8] Andreas P, Martin F 2004 Secure Network Management within an Open Source Mobile Agent Framework *Journal of Network and Systems Management* **12**(1) 9-31

[9] Dokas I M, Karras D A, Panagiotakopoulos D C 2009 Fault Tree Analysis and Fuzzy Expert Systems: Early Warning and Emergency Response of Landfill Operations *Environmental Modelling and Software* **24**(1) 8–25

[10] Turban E 1997 *Decision Support and Expert Systems: Management Support Systems* Prentice-Hall

[11] Rajkumar Buyya 2000 PARMON: a Portable and Scalable Monitoring System for Clusters *Software: Practice and Experience* **30**(7) 723-39

[12] Liu G, Mok A K, Yang E J 1999 Composite Events for Network Event Correlation *Proc. Conf. on Integrated Network Management of the Sixth IFIP/IEEE* Boston, MA, USA 247-60

[13] Xu J P, Zeng Z Q 2011 Applying Optimal Control Model to Dynamic Equipment Allocation Problem: Case Study of Concrete-faced Rock Fill Dam Construction Project *Journal of Construction Engineering and Management* **137**(7) 536-50

Authors

Yumei Ning, born in December, 1978, China

Current position, grades: working toward the PhD degree in the School of Computer Science and Technology at the Xidian University of Xi'an.
University studies: master degree in information science from the Xidian University, Xi'an, China, in 2006.
Scientific interests: computer network management technology and computer image processing.

Zhenguo Ding, born in October, 1959, China

Current position, grades: Professor in the School of Computer Science and Technology at the Xidian University, Xi'an, China. And he is the education technology academic leaders in Xidian University.
University studies: PhD degree in the School of Computer Science and Technology from the Xidian University, Xi'an, China, in 2005.
Scientific interests: range from computer network technology to remote education technology. He has published more than 40 technical papers in these areas.

Ping Zeng, born in February, 1956, China

Current position, grades: Professor and PhD supervisor in the School of Computer Science and Technology at the Xidian University, Xi'an, China. He is a senior member of Chinese Institute of Electronics.
Scientific interests: computer graphics, computer image processing, color management technology and computer peripherals technology.
Publications: more than 70 technical papers in these areas.