Send Orders for Reprints to reprints@benthamscience.ae

Open Access

Polling Mechanism Based on Dynamic Priority Queue in Seismic Profession Cluster Devices Monitoring System

Wang Xiaoming[1,*], Du Lijuan[2], Ding Rui[1], Cai Changqing[1] and Li Ping[1]

[1]*Earthquake Administration of Shanghai Municipality, Shanghai, 200062, P.R. China*

[2]*School of air transportation, Shanghai University of Engineering Science, Shanghai 201620, P.R. China*

Abstract: This paper investigates the model and polling mechanism in cluster monitoring system for seismic profession. First, we propose a PULL monitoring system model suitable for seismic observation equipment cluster. Second, we modify the general FCFS polling mechanism which limits the polling counts of important nodes in cluster and pose a new polling mechanism of cluster node monitoring system based on dynamic priority queue (DPQ), which divides cluster nodes into dynamic and static queue. Nodes in dynamic queue are sorted by priority. The algorithm and flow chart of the DPQ polling mechanism are given. The experimental results show that, compared with the FCFS polling mechanism, DPQ increases polling counts for important nodes, and have little effect on the performance of central server, so the DPQ polling mechanism is suitable for seismic profession with great differences between cluster nodes.

Keywords: Cluster, dynamic priority queue, monitoring system, polling mechanism, static queue.

1. INTRODUCTION

Cluster is normally used to improve computing speed and reliability of one single computer. Through load balancing [1, 2] and parallel computing [3] technology the cluster achieves higher efficiency. Because of its special industry targets and tasks, the seismic profession needs to deploy the seismic monitoring nodes in various regions. With all kinds of servers, route and switch devices in the center, a seismic monitoring distributed cluster network is contributed to ensure the seismic monitoring data can be continuously and accurately transmitted to the center in real-time. Distributed cluster system possesses characteristics of multiple node types, so there are potential faults and anomalies [4] in hardware, operating system environment, internal network, cluster management and scheduling software, application and parallel computing environment, critical system services, storage file systems and so on. One of the basic mechanisms of distributed cluster monitoring system is the polling mechanism under which the center system monitors each node in the cluster either actively or passively, collects node information, triggers the alarm according to node state, notifies users through certain strategies.

Presently there are some researches on the cluster network polling mechanisms at home and abroad. The author in [5] studied a single-server cyclic polling system and reduced the average waiting time delay by setting priorities for visitors. A novel analytical model of the polling MAC protocol using M-gated services in discrete time was proposed in [6]. The author precisely analyzed the mean queue length, the mean polling period time and the mean packet delay. Considering the real time transmission and impartial service of information packets for Broadband Wireless Access network, the author proposed a novel and efficient model of two-level-polling system based on priority service in [7]. By the embedded Markov chain theory and the probability generating function methodology, the closed form expressions for obtaining the characteristic parameters of the system was built. In [8] a queuing model for the polling based service classes of WiMAX/IEEE 802.16 based wireless networks was presented. The author also gave operations for both single and multiple channels. In order to improve the throughput and reduce the average delay for UAV-DL (unmanned aerial vehicle data link), the author in [9] posed a new multi-link cooperative polling scheme based on Gilbert-Elliot channel, thus reducing the average delay and increasing the throughput.

In this paper we analyze the characteristics of cluster monitoring system, put forward the model of PULL monitoring which is suitable for seismic profession, and propose a polling mechanism of cluster node monitoring system based on dynamic priority queue (DPQ). DPQ polling mechanism uses dynamic and static queue. Dynamic queue, in which nodes are dynamically sorted by priorities, has the highest priority. Except for those in the dynamic queue, all cluster nodes, which have the same status, are contained in the static queue and polled by FCFS method. Finally, we build an experimental platform based on Nagios and Cacti through which comparisons are made to show that DPQ polling mechanism increases the counts of polling important node, and has little effect on the performance of central server.

The rest of the paper is organized as follows. In Section 2, we discuss the cluster monitoring system characteristic and

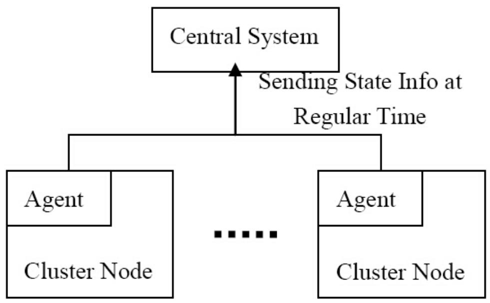

Fig. (1a). PUSH Monitoring.

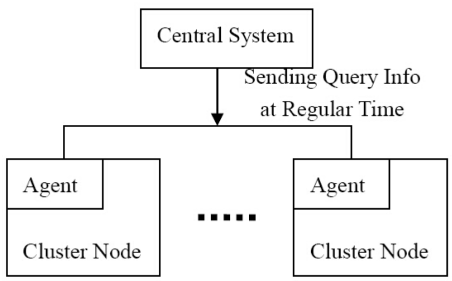

Fig. (1b). PUULL Monitoring.

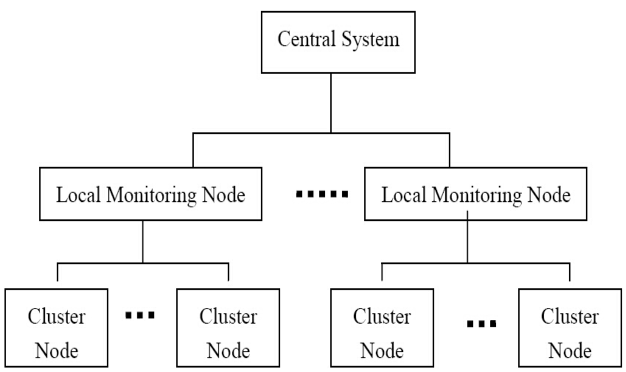

Fig. (1c). DISTRIBUTE Monitoring.

give a specific model for seismic profession. Section 3 proposes DPQ polling mechanism for seismic profession with great differences between cluster nodes. We build experimental platform to check the results of DPQ polling mechanism in section 4. Finally, we conclude the paper in section 5.

2. CLUSTER MONITORING SYSTEM MODEL

2.1. Running Model [10]

2.1.1. PUSH Monitoring

Each node in the cluster is deployed by an agent (hardware or software) that collects node information and sends it to central system. The central system judges node state by the information sent by node actively, and delivers feedback to user according to system alarm mechanism as shown in Fig. (**1a**). In this model, detailed node information is monitored, so correct judgment of node state is more likely to be made. The drawbacks of this model are high demands for agent performance, great costs and influences by the number of cluster nodes. The more nodes in cluster, the greater loads of central system and the more cost in communication links. The number of monitored nodes is extremely big and the node attribute information is single, so this model is not suitable for seismic profession.

2.1.2. PULL Monitoring

According to the node attributes and monitoring requirements, agent is deployed in each cluster node. As shown in Fig. (**1b**), the central system judges node state by the information sent by node passively, and delivers feedback to user according to system alarm mechanism. In this model, central system can reasonably set polling and alarm mechanisms according to its performance, thus assuring system reliability, decreasing costs in communication links and its loads. The disadvantage of this model is that agent needs to be deployed in each cluster node, thus increasing costs, furthermore, polling and alarm mechanisms of central system are highly required in PULL monitoring model. It is must be considered how to achieve a compromised load balancing degree. Monitored nodes in seismic profession are servers, routers, switchers, data acquisition devices, UPS equipments, Geoelectrical Observation Apparatus, Vertical Pendulum Tiltmeters, Water Temperature Recorders and so on. Among all those devices, most of them, except for some servers, UPS devices and other additional equipments such as power controller [11], do not have to be deployed with agents as they are only needed to be monitored on-off state through IP address, so only polling and alarm mechanisms of monitoring system are to be concerned. This model is suitable for monitoring system in seismic profession.

2.1.3. DISTRIBUTE Monitoring

Too many nodes in cluster may cause low efficiency in PUSH and PULL models. The DISTRIBUTE model divides nodes into groups, each of which sets up a local monitoring node which is responsible for the collection of nodes state information in this group. Central system gathers aggregated information from each local node through PUSH or PULL models, as is shown in Fig. (**1c**). By grouping the cluster nodes, DISTRIBUTE monitoring model solves the cluster nodes population problem, but complex deployment mode contributes to its limitations. Central system indirectly gathers state information from cluster nodes, which may lead to lag of updated information, thus reducing the reliability and timeliness of monitoring system. In addition, performances of local nodes are highly demanded in this model, so it is improbable for seismic profession to use DISTRIBUTE monitoring model.

2.2. Node Properties

According to functional properties division in seismic profession, monitoring devices in cluster are divided into three types: station monitoring devices, routers and switchers, application servers. The first two types only need to be monitored with IP address and static port on-off state by ping

Table 1. Alarm thresholds for different types of link states.

Monitoring results / Devices states / Link types	Normal	Warning	Critical
SDH	delay＜500 ms packet loss＜20%	500ms≤delay＜1000 ms 20%≤packet loss＜60%	delay≥1000 ms packet loss≥60%
Satellite	delay＜1100 ms packet loss＜30%	1100ms≤delay＜2000 ms 30%≤packet loss＜60%	delay≥2000 ms packet loss≥60%
GPRS	delay＜1000 ms packet loss＜30%	1000ms≤delay＜1800 ms 30%≤packet loss＜60%	delay≥1800 ms packet loss≥60%
ADSL	delay＜500 ms packet loss＜20%	500ms≤delay＜1000 ms 20%≤packet loss＜60%	delay≥1000 ms packet loss≥60%
3G	delay＜800 ms packet loss＜30%	800ms≤delay＜1200 ms 30%≤packet loss＜60%	delay≥1200 ms packet loss≥60%

command, so it is not necessary to deploy agents. While to application servers, agents should be installed in order to monitor operating systems as well as information distinguished by operating system and application types. The basic factors affecting the performance of the Windows systems include CPU usage, memory usage, disk usage, which is different from Linux systems that the latter's underlying architecture is opposite to the former's. Linux systems take maximum use of hardware resources as objective, so it is meaningful to calculate average CPU and memory usages in some periods. Central system must take these factors into account when monitoring such nodes. For some critical application servers such as ORACLE servers, users care more about instance state, process state, table space utilization rate and other information, so we must consider these monitoring requirements in the cluster monitoring system deployment.

2.3. Link Status

According to technology types, cluster nodes in seismic profession can be divided into SDH, CDMA (GSM, GPRS, 3G), satellite links. The requirements for nodes monitoring information in different links are also different due to the different types of transmission mediums in different communication links, the different transmission distances and the different communication technologies used, consequently, the node polling mechanisms to be discussed in section 3 will be influenced. Taking seismic monitoring devices cluster in Shanghai Earthquake Administration as an example, alarm thresholds for different types of link states are set in Table **1**.

3. ALGORITHM AND IMPLEMENTATION

3.1. DPQ Algorithm Description

There is the basic polling mechanism of cluster monitoring system: Central system reads required information from the cluster nodes according to the configuration files, while nodes send status feedbacks to central system either directly or through agents. That kind of mechanism will lead to the following problems: First, important nodes will be ignored as all cluster nodes have equal status. In seismic profession, stationary(especially near the fault zone) real-time observation devices and core databases are of most important so they should be set with the highest priorities; Second, the ordinary cluster nodes' states are normal over a long period of time. Based on classical trend analysis algorithms such as Linear Regression Algorithm and Exponential Regression Algorithm [12], such nodes' priorities should be lowered. To solve the above two problems, we propose the DPQ polling mechanism for cluster nodes monitoring, which divides cluster nodes into dynamic and static queues. Dynamic queue, in which the polling sequence of each node is assigned by dynamic priorities, has the highest priority. Static queue, in which all cluster nodes are included and each of them has the same position, is polled by FCFS method. In one polling cycle, central system polls according to the nodes' priorities in dynamic queue. After the polling, nodes' priorities decrease. Theoretically there are two situations: First, dynamic queue is not polled through in one polling cycle. In this case, the dynamic queue waits for the next polling after adjustment of nodes' priorities while the static queue is not polled. Second, dynamic queue is polled through in one polling cycle. In this case, the static queue is polled subsequently until the end of the polling cycle. In practice, we set polling cycle to 5 minutes considering the performance of server and the fairness of polling (to be discussed in section 4). Dynamic queue is initialized with a certain length. The node with highest priority ranks in the head, while the lowest in the end. During the polling, Binary Insertion Algorithm [13] is selected for its relatively low complexity and high stability. Static queue is initialized according to the FCFS algorithm and new node is inserted into the queue in turn. To prevent the initialized priority being too low so as not to be polled in a long term, node priority minus one after it is polled until it is less than a threshold, then it is reinitialized. Meanwhile, in

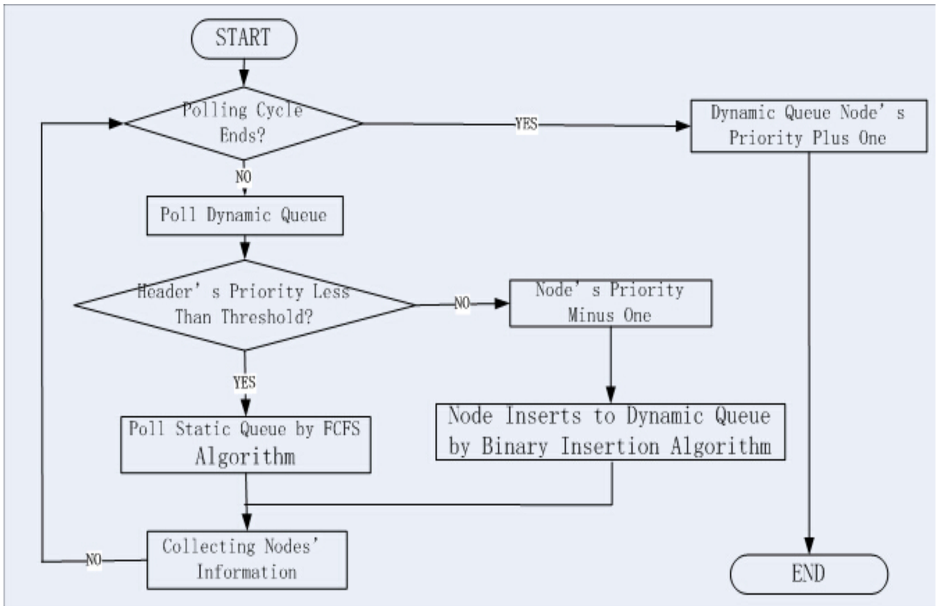

Fig. (2). DPQ algorithm flow chart.

order to ensure the fairness of polling, we set the priority threshold in the dynamic queue. When the head node's priority is below the threshold, central system directly polls static queue rather than dynamic. Header priority may be always lower than threshold; in this case, dynamic queue will be never polled. In order to prevent such situation, each node's priority plus one when one polling cycle ends. Fig. (**2**) shows the algorithm flow.

3.2. DPQ Algorithm Implementation

Assuming that the lengths of dynamic(LDynamic) and static(LStatic) queue are m and n respectively, the polling cycle is T, the dynamic queue priority threshold is TH, the priority of node I in dynamic queue is P[i], Binary Insertion and FCFS Algorithms are described respectively as BinaryInsert(N[i],L-N[i]) and FCFSInsert(N[i],L-N[i]). The Pseudo codes of DPQ polling mechanism algorithm are as follows:

```
Function PollDynamic(L)
// Dynamic queue polling function
BEGIN
  For i:=0 To m-1
  {
   P[i]--;
   BinaryInsert(N[i],L-N[i]);
   status information collecting;
   TimeEnd:=GetLocalTime();
   If((TimeEnd-TimeStart)>T)
   Then return;
  }
  return;
END PollDynamic

Function PollStatic(L)
// Static queue polling function
BEGIN
  For i:=0 To n-1
  {
   FCFSInsert(N[i],L-N[i]);
   status information collecting;
   TimeEnd:=GetLocalTime();
   If((TimeEnd-TimeStart)>T)
   Then return;
  }
  return;
END PollStatic

Function AddPriority(L)
// Dynamic queue node priority increasing function
BEGIN
For i:=0 To m-1
    P[i]++;
  return;
END AddPriority

Main()  //Main function
BEGIN
  TimeStart:=GetLocalTime();
  TimeEnd:=GetLocalTime();
  While((TimeEnd-TimeStart)>T) Do
  {
   If(P[0])>=TH)
    Then PollDynamic(LDynamic);
    Else PollStatic(LStatic);
  }
  AddPriority(LDynamic);
```

Table 2. The experimental parameters.

Parameters	**Value**	**Descriptions**
Central Server CPU	**Intel(R)Xeon(R) 5130@2.00GHz**	
Central Server Memory	4G	
Numbers of cluster nodes	248	Total numbers of all station devices, central servers, routers and switchers, virtual machines
Dynamic queue length(m)	78	Total numbers of core database, router and switcher, part of important station devices
Static queue length(n)	170	Total numbers of other nodes
Polling Cycle(T)	5 minutes	Time using for monitoring system polling once, set 5 minutes based on experience
Dynamic queue priority Threshold(TH)	80	Dynamic queue polling threshold is set for fairness, and only when the priority of header is not lower than threshold, dynamic queue will be polled. It is set as 80 based on experiment
Span of experimental data acquisition	30 days	In order to meet the accuracy requirements, we collected 30 days monitoring results, which are from 05/01/2014 to 05/31/2014（FCFS） and from 06/01/2014 to /07/01/2014（DPQ）

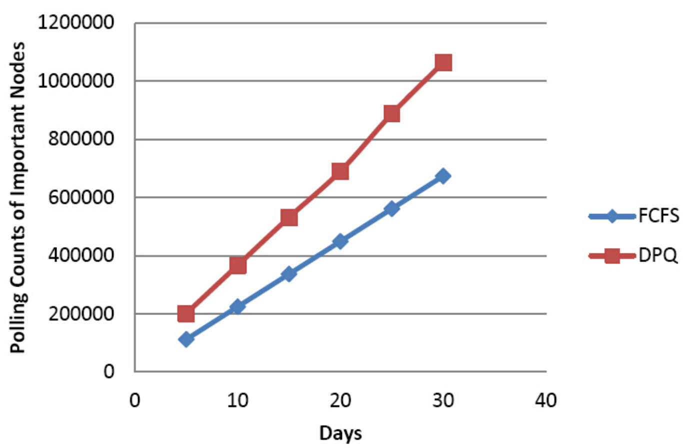

Fig. (3). Comparison of important nodes polling counts between two polling mechanisms.

return;
END Main

4. EVALUATION

We take central and stationary devices in Shanghai Earthquake Administration as samples and build an experimental platform based on Nagios [14], Cacti [15] monitoring systems to implement DPQ polling mechanism. Parameters are shown in Table **2**.

Experiments make comparisons between two polling mechanisms about polling counts of important nodes and influence on central server performance according to 30 days' results respectively.

Fig. (**3**) shows that important nodes polling counts are approaching nearly linear over time under both FCFS and DPQ polling mechanisms. Important nodes polling counts under DPQ poling mechanism are higher than that of FCFS. This is because all nodes in dynamic priority queue have higher priorities under DPQ polling mechanism and will be polled by central system first, meanwhile, each polled node is reinserted into dynamic queue according to Binary Insertion Algorithm, which ensures the priorities of important nodes. In addition, the threshold ensures fairness for polling nodes in static queue. As a result, the DPQ polling mechanism is suitable for seismic profession with great differences between cluster nodes.

Fig. (**4**) shows statistics of influence on server CPU and memory usages under FCFS and DPQ polling mechanisms respectively over 30 days. From comparison between Fig. (**4a** and **4b**), we can see that two polling mechanisms have little influence on the server CPU and memory usages. Within 30 days, the average server CPU usage under FCFS is 11.24%+5.11%=16.35% and DPQ is 11.16%+5.10-%=16.26%, which means no significant effect on CPU usage.

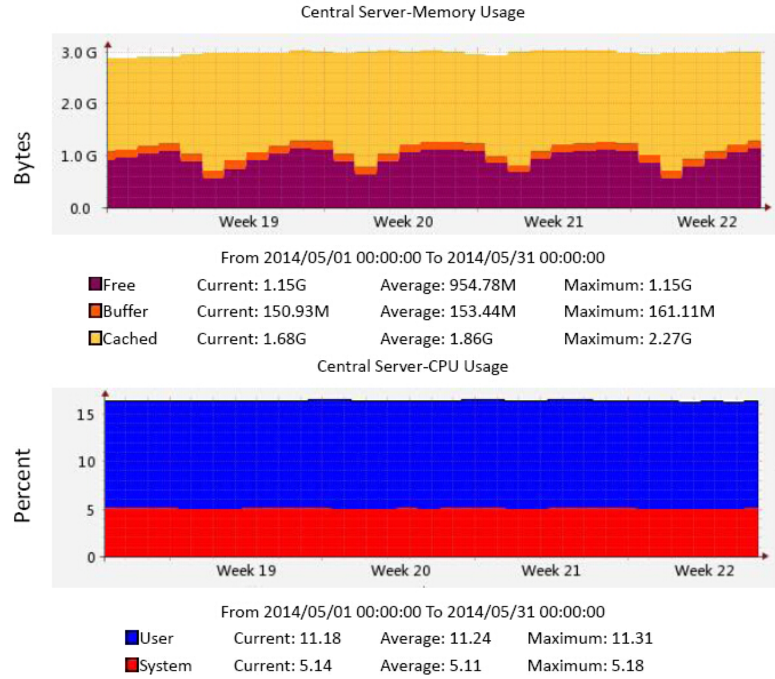

Fig. (4a). Effect of FCFS polling mechanism on server performance.

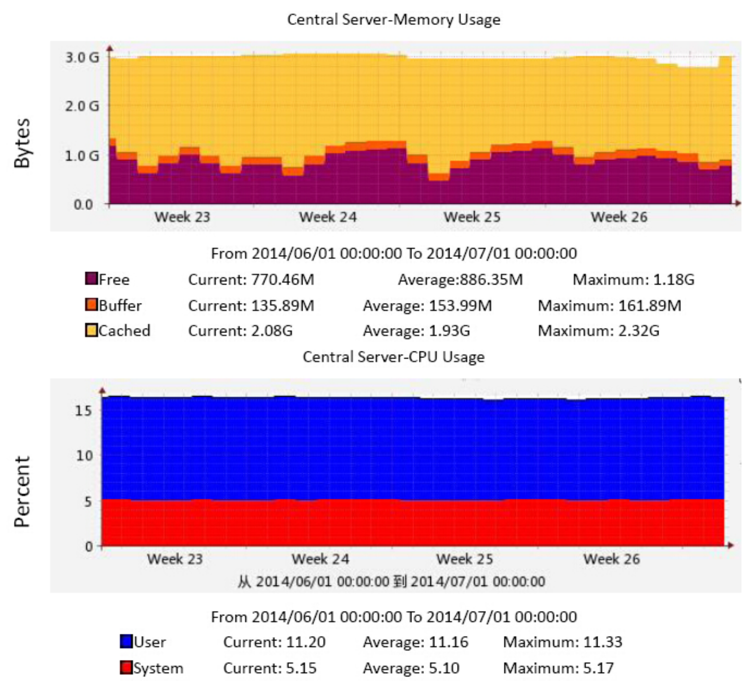

Fig. (4b). Effect of DPQ polling mechanism on server performance.

Within 30 days, the average server memory available under FCFS is 954.78 MB+153.44 MB+1.8-6 GB=3012.86 MB, and DPQ is 886.35 MB+153.99 MB-+1.93 GB=3016.66 MB, which means no significant effect on memory usage. Both FCFS and DPQ polling mechanisms can meet requirements for normal operation of the system, thus new DPQ polling mechanism will not cause excessive load of central monitoring system.

CONCLUSION

This paper discusses three modes of cluster monitoring system, and proposes PULL monitoring model for cluster monitoring system according to the characteristics of seismic profession. In this model, we analyze the shortcomings of the original FCFS polling mechanism and pose a DPQ polling mechanism. New mechanism solves the problem of great differences between cluster nodes by initializing dynamic priority queue, and ensures fairness of polling cluster nodes by setting the threshold. The flow chart and algorithm of DPQ are given in this paper. The results from experimental monitoring platform we build show that, under DQP polling mechanism, polling counts of important nodes are increased without increasing load of central server, so this mechanism is suitable for seismic profession with great differences between cluster nodes.

ABOUT THE AUTHORS

First Author Wang Xiaoming, Earthquake Administration of Shanghai Municipality, engineer, studying for PhDs in DongHua University. The author's major is intelligent vehicle communication.

Second Author Cai Changqing, Earthquake Administration of Shanghai Municipality, senior engineer. The author's major is Computer Since and Technology.

Third Author Du Lijuan, School of air transportation, Shanghai University of Engineering Science, lecture. The author's major is English Linguistics and Aviation English.

Forth Author Ding Rui, Earthquake Administration of Shanghai Municipality, engineer. The author's major is Computer Software.

Fifth Author Li Ping, Earthquake Administration of Shanghai Municipality, senior engineer. The author's major is Geophysical.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by Science for Earthquake Resilience (XH14020Y).

REFERENCES

[1] C. Zhang, and J. H. Gu, "A load balancing strategy in virtual machine cluster based on live migration of virtual machine", *Microelectronics and Computer*, vol. 31, no. 4, pp. 79-82, 2014.

[2] X. M. Wang, and H. W. Qin, "The conception and implementation of desktop virtualization technology in Earthquake Administration of Shanghai Municipality", *Seismological and geomagnetic observation and research,* vol. 34, no. 3/4, pp. 238-244, 2013.

[3] S. G Yin, Z. D. Xia, C. Yu, and Q. F. Zhu, "Study and application of parallel computing of MCNP code based on rocks", *Nuclear Electronics and Detection Technology*, vol. 33, no. 12, pp. 1511-1514,2013.

[4] Y. Zhang, L. Chen, and J. Pang, "Fault monitoring and management system for multiple computing clusters", *Computer Engineering and Science*, vol. 35, no. 11, pp. 54-61, 2013.

[5] M. A. A. Boon, I. J. B. F. Adan, and O. J. Boxma, "A polling model with multiple priority levels", *Performance Evaluation*, vol. 67, no. 1, pp. 468-484, 2010.

[6] D. F. Zhao, H. W. Ding, Y. F. Zhao, and M. G. Wang, "An analytical Model of a Discrete-Time Polling MAC Protocol for Wireless LANs Using M-Gated Services", *Acta Electronica Sinica*, vol. 38, no. 7, pp. 1495-1500, 2010.

[7] Q. L. Liu, D. F. Zhao, and Y. F. Zhao, "Performance analysis of two-level polling system based on priority service", *Journal of PLA University of Science and Technology (Natural Science Edition)*, vol. 12, no. 3, pp. 223-228, 2011.

[8] R. Iyengar, and B. Sikdar, "A Queueing Model for polled service in WiMAX/IEEE 802.16 networks", *IEEE Transactions on Communications*, vol. 60, no. 7, pp. 1777-1781, 2011.

[9] Y. Q. Mao, S. J. Li, B. Li, and Q. R. Zheng, "A new multi-link cooperative polling scheme", *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 23, no. 5, pp. 585-592, 2011.

[10] Y. F. Wu, Q. Gui, M. Y. Luo, W. J. Cai, and M. Xu, "Monitoring technology of computer in cluster", *Computer and Modernization*, no. 5, pp. 218-222, 2013.

[11] X. M. Wang, J. Chen, and Y. Zhang, "The implementation of remote management unit of state power", *Northwestern Seismological Journal*, vol. 35, pp. 158-161, 2013.

[12] Y. H. Liang, Y. C. Liu, "Trend analysis method for monitoring metrics in monitor system", *Computer Engineering and Applications*, vol. 49, no. 12, pp. 218-222, 2013.

[13] H. Ellis, and S. Sartaj, Anderson-Freed Susan, Fundamentals of Data Structures in C, Silicon Press, 2008, pp. 8-11.

[14] Nagios Software(http://www.nagios.org).

[15] Cacti Software(http://www.cacti.net).

Received: September 22, 2014 Revised: November 03, 2014 Accepted: November 06, 2014