Research on Automatic Management Model to Personal Computer

Yalin Song
Complex Intelligent Network Institute, Henan University, Kaifeng, P. R. China
Computing Center, Henan University, Kaifeng, P. R. China

Xin He
Complex Intelligent Network Institute, Henan University, Kaifeng, P. R. China
Computing Center, Henan University, Kaifeng, P. R. China

***Abstract*—This paper describes the roles and functions of autonomic computing for personal computer. Based on the technology of autonomic computing, this paper defines an architectural model for personal computer autonomic computing to solve some problems , which lead to the crisis of software complexity. Based on this model, one test program with self-monitoring, self-analyzing, self- healing characters is implemented. Some key values of knowledge base for automatic management is proposed in this paper after we discussed and analyzed the influence of utilizing rate of CPU and usage rate of RAM and usage rate of network bandwidth on operation system in personal computer. The results show that self-management is feasible for PC and is used in some application systems, and obtains good results.**

***Index Terms*—autonomic computing, automatic management model, self-monitoring, self- healing**

I. INTRODUCTION

Quality of services of information technology must be improved, while reducing the total cost of ownership of their operating environment. System deployment failures, hardware and software issues, and human errors can increasingly hamper effective system administration. Human intervention is required to enchance the performance and capacity of the components in IT system. But, with the development of the technology of science and information, human intervention leads to many problems. So, One new technology -- Autonomic computing is presented. It helps to address complexity by using technology to manage technology.

To complex system, autonomic computing has self-managing ability .it accomplish functions by taking an appropriate action based on one or more situations that they sense on environment. The automatic ability is a control loop that collects details from the system and acts accordingly. It includes self-monitoring, self- healing, and so on. Automatic computing leads to huge benefit to complex system management. But, existing studies main pay attention to server, the personal computer is ignored.

With the development of the technologies of electronic and microelectronics, personal computer is the main tool which is necessary in many fields such as routine, study, life etc. The computer system has become more and more big and the structure between hardware and software is also more complexity, but people have not capability enough to operate and manager this complex system. So we need urgently a kind of computer system having self-monitoring, self-configuring, self-analysis and self-controlling when some running programs on it exists bug and error. Only in this way can computer become more useful tool.

In this paper, the autonomic computing concept is introduced in section 2; in section 3, Design and implementation of architectural model of autonomic computing based on Personal computer; in section 4, performance of the autonomic computing model and algorithms are analyzed by experiments; in the last section, conclusion of this paper is given.

II. AUTONOMIC COMPUTING

Because the processing data is more complex and the number of accessing device is more and more, the traditional computing method (input data->execute computing->output result) has not to gratify our expectation now. Although we had added memory constantly and showed output result with graphic form and accelerated the speed of operation of basic component of computers, we also think that the important function of machine is computing and operating. We had not realized the real intention of the people using information that is not only computing but also hide complexity and reduce manual intervention.

Autonomic computing is a model of automatic management. The concept is from the immune system of human body, and gains this name. The autonomic computing system can control programs and system function of computer while will not bother users as well as human bodies' immune system. It is the important goal that we can create the system with self-running and advance function, and the complexity of system cannot be perceived by users [1].

IBM Corporation at first put forward the concept of autonomic computing in 2001[2]. The purpose is that will create a computer system to fulfill self- management. In this system computer can monitor voluntarily own

doi:10.4304/jsw.7.6.1227-1233

running-state and carry out the corresponding treatment operation about different state based manage tactic. It may reduce system complexity and the cost of manage because manager was free from the complex task for PC.

The main idea of autonomic computing is the function of automatic management with four characters [3]: self-controlling, self-configuring, self-optimizing and self-healing. (1) Self-controlling: the system knows about the current state and memory capacity and connecting devices of each element in it. It can detect and forecast and recognize some attack from everywhere to protect itself; (2) self- configuring: the system configuration will be completed automatically and it can adjust some parameters to keep the system work steadily and continuously; (3) self-optimizing: the computer system will dispatch some resources automatically in order to complete the goal of work normally; (4) self-healing: the computer system can take the measure of correcting according to the tactics to restore the running state of it if there are some problems, which is normal or accident, in PC. One autonomic computing system should include some parts: monitoring, analyzing, planning, executing and knowledge base, and these parts form a circulatory system that need operate and improve.

Ganek presents an overview of IBM’s autonomic computing initiative in paper [14]. It examines the genesis of autonomic computing, the industry and marketplace drivers, the fundamental characteristics of autonomic systems, a framework for how systems will evolve to become more self-managing, and the key role for open industry standards needed to support autonomic behavior in heterogeneous system environments.

Kephart introduces a unified framework that interrelates three different types of policies that will be used in autonomic computing systems: Action, Goal, and Utility Function policies in paper [15]. These policy framework is based on concepts from artificial intelligence such as states, actions, and rational agents. It shows how the framework can be used to support the use of all three types of policies within a single autonomic component or system, and use the framework to discuss the relative merits of each type.

System and network security are vital parts of any autonomic computing solution. The ability of a system to react consistently and correctly to situations ranging from benign but unusual events to outright attacks is key to the achievement of the goals of self-protection, self-healing, and self-optimization. Because they are often built around the interconnection of elements from different administrative domains, autonomic systems raise additional security challenges, including the establishment of a trustworthy system identity, automatically handling changes in system configuration and interconnections, and greatly increased configuration complexity. On the other hand, the techniques of autonomic computing offer the promise of making systems more secure, by effectively and automatically enforcing high level security policies. Chess discuss these and other security and privacy challenges posed by autonomic systems and provide some recommendations for how these challenges may be met in paper [16].

Brittenham designs one autonomic computing framework for IT service management. It defines a set of best practices to align information technology (IT) services to business needs by using the IT Infrastructure Library[17].This framework helps organizations manage IT services using standard design patterns and the requisite customization. He discuss critical contributions that autonomic computing offers to the definition and implementation of an ITSM architecture and infrastructure. He first introduces key architectural patterns and specifications of autonomic computing as they relate to an ITSM logical architecture. Then he shows how autonomic computing delivers value through a set of ITSM-based case studies that address problem determination, impact assessment, and solution deployment.

Because computing systems have become so complex that the IT industry recognizes the necessity of deliberative methods to make these systems self-configuring, self-healing, self-optimizing and self-protecting. Architectures for system self-management, srivastava explores the planning needs of Autonomiac computing, its match with existing planning technology and its connections with policies and planning for web services and scientific workflows (grids) inpapter [18]. Then, he shows that planning is an evolutionary next step for AC systems that use procedural policies today.

In addition, application-layer networks (ALN) are software architectures that allow the provisioning of services requiring a huge amount of resources by connecting large numbers of individual computers, e.g. Grids and P2P-Networks. Self-organization, like proposed by the Autonomic Computing concept, might be the key to controlling these systems. So, by CATNET project[19], eymann evaluates a decentralized mechanism for resource allocation in ALN, based on the economic paradigm of the Catallaxy. The economic model is based on self-interested maximization of utility and self-interested cooperation between software agents, who buy and sell network services and resources to and from each other.

The above concept, however, is put foreword to resolve some complex problems about sever firstly. As to server, the personal computer has its own characteristics, such as lower hardware requirement, cheaper price, large quantity, wide application range etc. Furthermore, the people who operated personal computer almost are ordinary users that did not participate in specialized training; the hardware and software configuration often change as not well as sever computer that can improve fault-tolerant through the technology of hardware redundant. With constantly updating software and hardware peoples operated personal computer feel the more and more difficult in handling and maintaining. In a word, it become an urgent problem how do apply the technology of autonomic computing into the personal computer to simplify people's operation and management.

For this in this paper we define an architectural model for personal computer autonomic computing to solve some problems which lead to the crisis of software complexity. Based on this model, one program with self-monitoring, self-analyzing, self-healing characters is implemented in dot net framework.

III. Design and implementation of architectural model of autonomic computing based on PC

The paper has been implemented automatic management about personal computer to reduce the complexity of managing personal computer through integrating the technology of autonomic computer and personal computer. According to IBM's model of autonomic computing we defines an architectural model for the personal computer's autonomic Computing whose main thought is that implemented one model with the function of automatic management. It has two parts: self-manager and managed resource. See figure1. Managed resource is said to the hardware and software resource. And self-manager is made up from controlling model, analyzing model, planning model, executing model, communication model, etc. They shared a knowledge base and formed a circle of controlling in order to control managed resource.

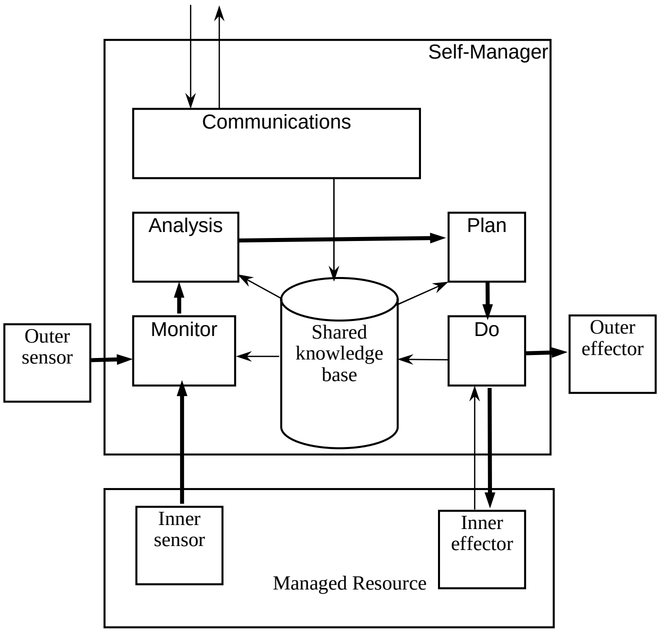


Figure1. Self-management model

As following is the step of implement about automatic management module.

A. Building shared knowledge base

In order to save tactics, logs and performance information of system we built shared knowledge base that gets information from monitor module and supported by users. When some condition is satisfied the role will be triggered and then lead to some corresponding action.

This paper used extensible markup language (XML) to save the shared knowledge base because XML language has some characteristics that are easy and internationalized standard etc. And the operation of reading and writing is very convenient. Here is a format about this file.

```
<CTRLContent>
  <ID>1</ID>
  <TypeName>CPU</TypeName>
  <UValue>50</UValue>
  <Unit>%</Unit>
  <Duration>3000</Duration>
</CTRLContent>
```

The Node "TypeName" is the type of monitoring; "UValue" is upper limit value of "TypeName"; "Unit" is the unit of "UValue"; "Duration" is the interval of current value more than UValue, whose unit is millisecond.

B. Design Operation module of knowledge base

The shared knowledge base needs update constantly to support more efficient management tactics. The paper designed an updating module to update the base. All data gained by the monitoring module and input by users update the knowledge base through the module.

According to the above file format, this paper use DataTable class of dot net framework to read the base file, and quickly convert this XML file to inner data table of .net that can be read and written conveniently and fastly. Here is the main code.

```
DataTable dtCTRL=new DataTable();
dtCTRL.ReadXml(xmlPath);
Read Data：
    dtCTRL.Rows[0]["TypeName"]
    //get the name monitored in knowledge
    dtCTRL.Rows[0]["UValue"]
    // get the upper value monitored in knowledge
    dtCTRL.Rows[0]["Duration"]
    // the interval of current value more than UValue
……
Write Data:
    dtCTRL.Rows[0]["UValue"]=value;
    dtCTRL.Rows[0]["Duration"]=value;
    dtCTRL.WriteXml(xmlPath);
```

In addition, we also reference Xml namespace to Read/Write Xml files that of main class is Xml Document.

C. Design monitoring and analyzing module

Because of some reasons such that required resources are not satisfied and connected net time out etc. some processes has been run in computer could be suspended and do not accept information from users. The operation of user controlling computer will be influence and computer's running-state will become unsteady. The entire computer system maybe breakdown with the more and more the above processes. To some extent, maybe people find these processes suspended and kill these processes through the task manager of windows, maybe we could not do anything on PC and finally computer system only is restart.

The paper designed monitoring module and analyzing module based the model of automatic management. This program will collect performance information of system that included the occupancy rate and utilizing rate of CPU, the utilizing rate of memory and net connecting state etc through sensor controlling. The information is main parts of the shared knowledge base and will be analyzed in order to check if the current action of computer system is correct by the operating module of the base. If abnormal behavior of PC was tested the computer system would try its best to do something to avoid reboot the machine.

In the dot net framework supported powerful operating class: Process. It can get some useful information: the occupancy rate of CPU and the utilizing rate of memory and so on. The main code is:

In the dot net framework supported some powerful operating classes: Process. They can get some useful information at ease: the occupancy rate of CPU and the utilizing rate of memory and so on[4]. The main code is:

```
Process[] procs = Process.GetProcesses();
long ram;
TimeSpan tCPU;
DateTime dtStart;
bool bResFlag;
foreach (Process myProc in procs)
   //get all processes opened by windows
{
        ram = myProc.WorkingSet64;
        //get the utilizing rate of Memory
        tCPU =myProc.TotalProcessorTime;
        //get the duration of process
        dtStart =myProc.StartTime;
        //the time of starting this process
        bResFlag =myProc.Responding; //respond Flag
 }
```

We can get the occupancy rate of CPU from the current time, the value CPU and dtStart's value.

D. Designed planning and executing module

When some problems was found the module will analyze these troubles to deal with it automatically based on the controlling results and strategies in knowledge base. Finally, the system will call effectors to do it.

E. Designed sensor and effecter module

The main function of sensor is that collected managed resource's information, and it has two parts: inner sensor and outer sensor [5]. The inner sensor could gain inner resource of computer and outer sensor module may get some information not in PC through net environment, so it can manage others computers and achieve the goal of extending self model.

The effector does process for managed resource and divides two parts: inner effector and outer effector. Inner effector can deal with internal resources of computer while outer effector do other outers' through net connection. In addition, we also use peer to peer technology to search resources to implement remote cooperative between computers that could extend the function of sensor and effector.

In this paper implemented effector will terminate some processes to restore the normal state of computer, which shows the function of self-healing. We may use the Kill() method of Process class to achieve this goal in dot net environment. The main step is:

```
  Process
myprocess=process.GetProcesses(ProcessName);
  myprocess.Kill();
```

IV. Performance analysis

The Methods above are realized by programming in VC#2005 and running on the computer, whose processor is P4 2.4G and Memory is 1G and operating system is windows XP. The implemented programs included all parts of self-manager and monitor the whole system through monitoring module and use analyzing module to judge whether some processes was suspended. If processes suspended were found, the program will terminate them.

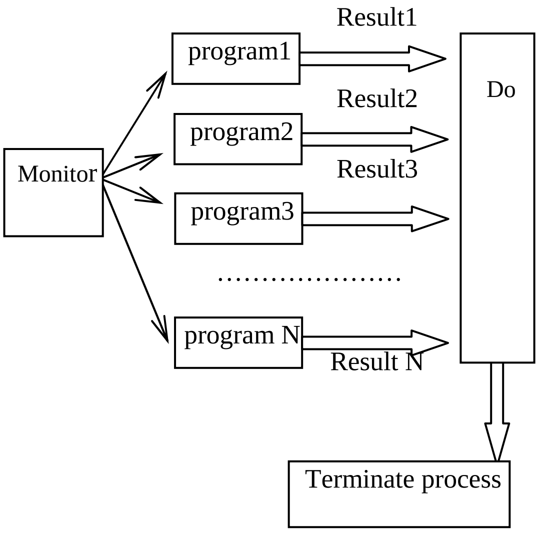
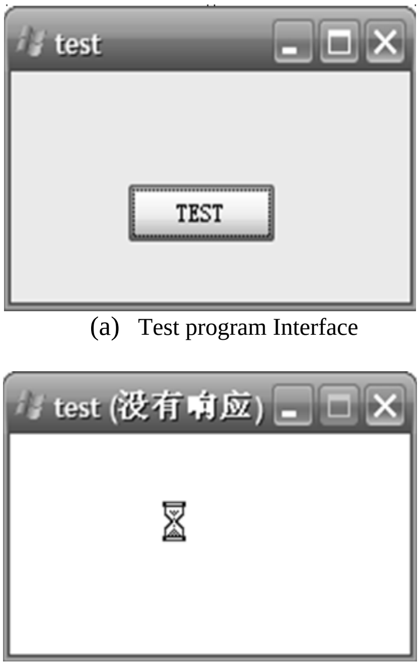


Figure2. Self-manager illustrative diagram

All parts above depend on one another and formed a circle of controlling that is used as roles of judgment condition, such as judged the suspending factor, the interval of sending message etc. All information was saved in the knowledge based. These roles can be certainly updated by the communication interface. Here is the experiment illustrative diagram in figure 2.

A. Implementation Process

To test this model we developed a small program, called "test.exe". It has a main form user-interface and a command button. As clicked the button, the program will enter the state of endless loop. Furthermore, the program will appear a serious bug, which is the memory leak, if it has not been end. The program can not be operated in this time and even lead to breakdown. See figure 3a and figure 3a.



(a) Test program Interface

(b) Test program run state

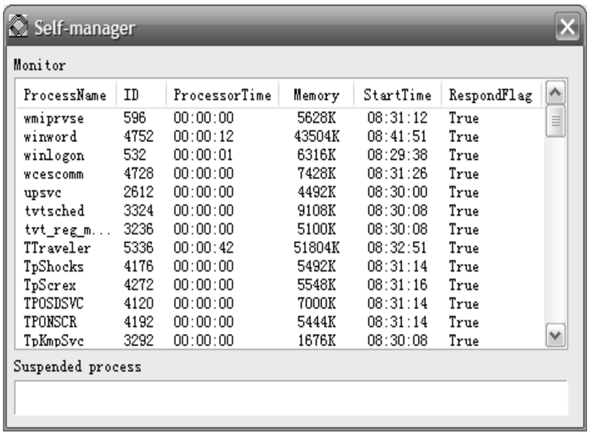
Figure3. Test program state

(1) Build the knowledge base file. We create a new XML file to store the knowledge base, which includes some control roles of CPU, RAM and Net. See figure 4.

```
<roleBase>
<CTRLContent>
   <ID>1</ID>
   <TypeName>CPU</TypeName>
   <UValue>50</UValue>
   <unit>%</unit>
   <duration>30000</duration>
</ CTRLContent >
< CTRLContent >
   <ID>2</ID>
   <TypeName>RAM</TypeName>
   <UValue>600</UValue>
   <unit>MB</unit>
   <duration>30000</duration>
</ CTRLContent >
< CTRLContent >
   <ID>3</ID>
   <TypeName>Net</TypeName>
   <UValue>1000</UValue>
   <unit>KB</unit>
   <duration>30000</duration>
</ CTRLContent >
</roleBase>
……
```

Figure4. The format of Shared knowledge base

(2) Run automatic management system selfManager.exe, and this system will monitor all processes, see figure 5.

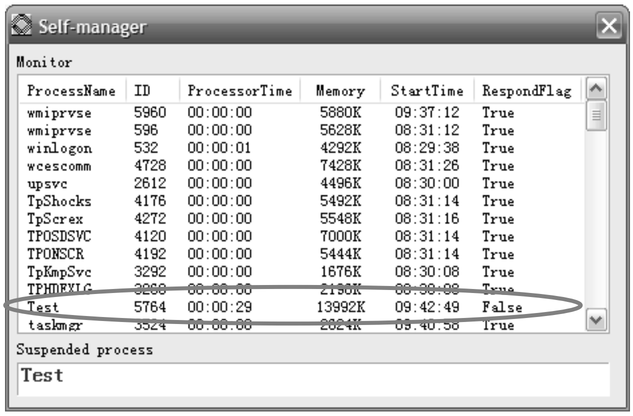


ProcessName	ID	ProcessorTime	Memory	StartTime	RespondFlag
wmiprvse	596	00:00:00	5628K	08:31:12	True
winword	4752	00:00:12	43504K	08:41:51	True
winlogon	532	00:00:01	6316K	08:29:38	True
wcescomm	4728	00:00:00	7428K	08:31:26	True
upsvc	2612	00:00:00	4492K	08:30:00	True
tvtsched	3324	00:00:00	9108K	08:30:08	True
tvt_reg_m...	3236	00:00:00	5100K	08:30:08	True
TTraveler	5336	00:00:42	51804K	08:32:51	True
TpShocks	4176	00:00:00	5492K	08:31:14	True
TpScrex	4272	00:00:00	5548K	08:31:16	True
TPOSDSVC	4120	00:00:00	7000K	08:31:14	True
TPONSCR	4192	00:00:00	5444K	08:31:14	True
TpKmpSvc	3292	00:00:00	1676K	08:30:08	True

Figure5. Self-management system interface

We can find that there are some important information, such as memory usage, CPU state and so on.

(3) Execute test.exe, then click TEST button. The program entered the state of endless loop. Observe the state of "text.exe".

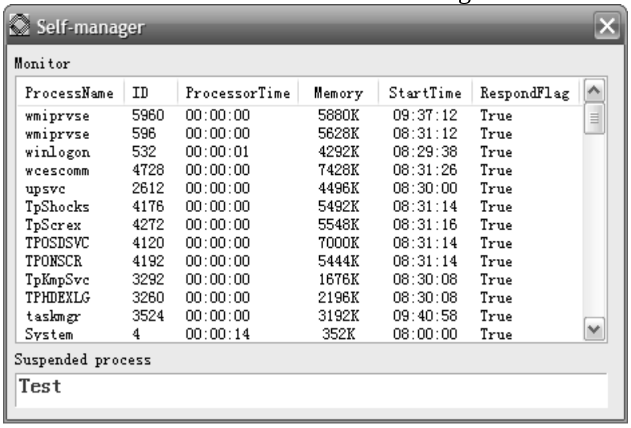


ProcessName	ID	ProcessorTime	Memory	StartTime	RespondFlag
wmiprvse	5960	00:00:00	5880K	09:37:12	True
wmiprvse	596	00:00:00	5628K	08:31:12	True
winlogon	532	00:00:01	4292K	08:29:38	True
wcescomm	4728	00:00:00	7428K	08:31:26	True
upsvc	2612	00:00:00	4496K	08:30:00	True
TpShocks	4176	00:00:00	5492K	08:31:14	True
TpScrex	4272	00:00:00	5548K	08:31:16	True
TPOSDSVC	4120	00:00:00	7000K	08:31:14	True
TPONSCR	4192	00:00:00	5444K	08:31:14	True
TpKmpSvc	3292	00:00:00	1676K	08:30:08	True
TPHDEXLG	3260	00:00:00	2196K	08:30:08	True
Test	5764	00:00:29	13992K	09:42:49	False
taskmgr	3524	00:00:00	2824K	09:40:58	True

Test

Figure6. Monitoring result

(4) Terminate "test.exe". The "test.exe" was terminated as the role of the knowledge base.



ProcessName	ID	ProcessorTime	Memory	StartTime	RespondFlag
wmiprvse	5960	00:00:00	5880K	09:37:12	True
wmiprvse	596	00:00:00	5628K	08:31:12	True
winlogon	532	00:00:01	4292K	08:29:38	True
wcescomm	4728	00:00:00	7428K	08:31:26	True
upsvc	2612	00:00:00	4496K	08:30:00	True
TpShocks	4176	00:00:00	5492K	08:31:14	True
TpScrex	4272	00:00:00	5548K	08:31:16	True
TPOSDSVC	4120	00:00:00	7000K	08:31:14	True
TPONSCR	4192	00:00:00	5444K	08:31:14	True
TpKmpSvc	3292	00:00:00	1676K	08:30:08	True
TPHDEXLG	3260	00:00:00	2196K	08:30:08	True
taskmgr	3524	00:00:00	3192K	09:40:58	True
System	4	00:00:14	352K	08:00:00	True

Test

Figure7. Operation result

B. Control Threshold Analysis

It is well known that the “not responding” phenomenon of the running program occasionally appeared in the operating system of Windows. Based on studying of the design principles of operation systems and the correlation rules of job scheduling, we find that the not responding problem of running program has some reasons: 1) one running program occupied operation capacity of CPU long time; 2) one running program used a lot of storage space; 3) one running program toke up a large number of network bandwidth; 4) a appearance of a complicated concatenation of circumstances. The operation system will appear some serious results, such as memory leak, system crashes and so on if the running problem program has not been end in time. And the program can not be operated during this time and even lead to breakdown.

We tested and analyzed the various factors and draw a conclusion. Here are some charts.

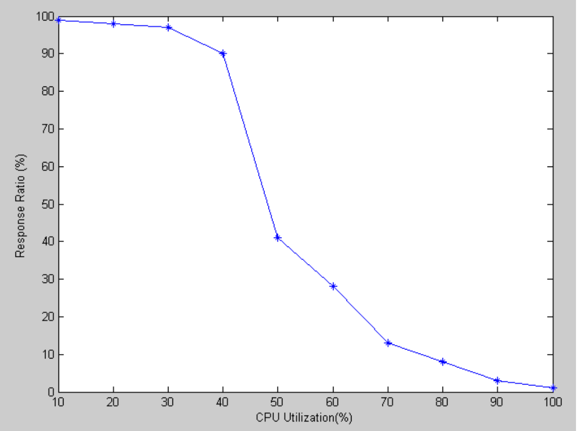


Figure8. CPU utilization and response ratio

From this chart we found that with the utilizing rate of CPU continuously rise lock-up occurrence probabilities greatly increased. In particular, it appeared “not responding” problem as the utilizing rate of CPU reached or excessed 50%.

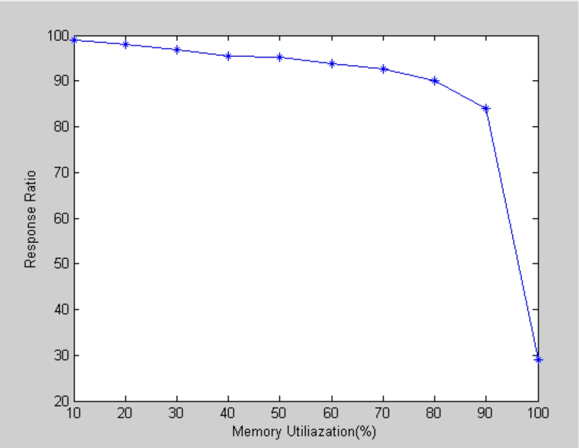


Figure9. Memory utilization and response ratio

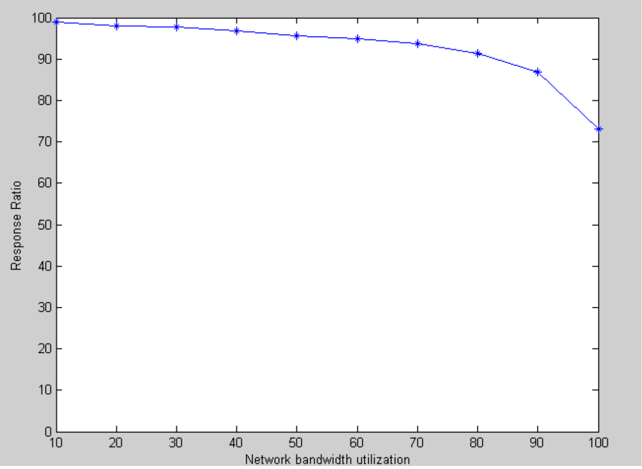


Figure10. Network bandwidth utilization and response ratio

Although we may use hard disk to set aside for virtual memory for Microsoft Windows, the response rate of computer will decline sharply when the usage of physical memory exceed a certain limit. According to the testing result it is obvious that the performance starts to slow as the utilizing rate of RAM reached or over 60%. And then “not responding” problem appears when the utilizing rate over 80%. From the following image we know that with the usage rate of memory continuously rises the response ratio of PC greatly declined.

A large number of network bandwidth was occupied when PC has been attacked by more and more intruders and worms. It appears some bad results, such as no surfing the Internet, slow response and breakdown. From this chart we found that with a lot of network bandwidth was occupied it continuously raise the response ratio of PC greatly declined..

C. Choose keys of knowledge database

The influence of utilizing rate of CPU and usage rate of RAM and usage rate of network bandwidth was discussed and analyzed on operation system in PC in this paper. And the key values of knowledge case were provided. The main reason of slow response ratio is the CPU utilization. The memory utilization and the network bandwidth utilization only play a secondary or supplementary role. So 50 percent is selected for the value of the utilizing rate of CPU that the response speed is very slow. If the duration is more than 30s the problem grogram is stopped. The key value of memory is about 60% physical memory, which is 600MB in this paper, and the key of duration is 30s.

The automatic management system can quickly handle these errors based on the key values of knowledge base to ensure normal system environment.

V. Conclusion

The paper describes the roles and functions of autonomic computing for personal computer, and defines an architectural model for PC-based autonomic computing after analyzing the requirement of personal computer field. According to this model, one program with self- monitoring, self-analyzing, self- healing characters is implemented in dot net framework. It shows that automatic management is feasible for personal computer. But this paper concern is the automatic management about single personal computer. As for the problem of automatic management about between different computers, further research is needed.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments. This work is supported by National High Technology Research and Development Program of China (No.2008AA01Z410), National Natural Science Foundation of China (No. 60873071) and the science and technology development project of Shaanxi province (No. 2007K04-05).

REFERENCES

[1] Kephart J, Chess D. The Vision of Autonomic Computing [R].IEEE Computer Society, January 2003;41- 50.

[2] IBM corporation. An architectural blueprint for autonomic computing. http://www-03.ibm.com/autonomic. 2003.

[3] Steve R. White, James E. Hanson,Ian whalley. An Architectural Approach to Autonomic Computing [C]. Proceedings of the International Conference on Autonomic Computing (ICAC'04), 2004.

[4] Christian Nagel Bill Evjen Jay Glynn. Professional C#. 4rd[M]. 2006.

[5] Roy Sterritt, David F. Bantz. PAC-MEN: Personal Autonomic Computing Monitoring Environment [C]. Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), 2004.

[6] Cai ming, Yu wei. An improved method of resource location based on Peer-to-Peer[J]. Control and Automation, 2006,3-3:108-109.

[7] Neroni, Massimo. Automatic management system for instrumentation maintenance. Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings, v 11 pt 5, p 1595, Nov 1989.

[8] Shivers, Olin. Automatic management of operating-system resources. ACM SIGPLAN International Conference on Functional Programming, ICFP, p 274-279, 1997.

[9] Cytron, Ron. Automatic management of programmable caches. the International Conference on Parallel Processing, v 2, p 229-238, 1988.

[10] Wang, Zhi-Li. Test result automatic judgment in network management interface testing. Beijing Youdian Daxue Xuebao/Journal of Beijing University of Posts and Telecommunications.

[11] Jo, Sun-Moon. Access control technique for XML documents. The 2005 International Conference on Internet Computing.

[12] Li Qiang; Hao Qin-Fen; Xiao Li-Min .An Integrated Approach to Automatic Management of Virtualized Resources in Cloud Environments. COMPUTER JOURNAL , 2011,54(6): 905-919.

[13] Autonomic computing white Paper. An architectural blueprint for autonomic computing, 2005,Third Edition.

[14] A. G. Ganek, A. G. Ganek. The dawning of the autonomic computing era[J], IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003,5-18.

[15] Jeffrey O. Kephart , William E. Walsh. An Artificial Intelligence Perspective on Autonomic Computing Policies[C], Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks: 2004.

[16] D. M. Chess, C. C. Palmer, S. R. White. Security in an autonomic computing environment[J], IBM SYSTEMS JOURNAL, VOL 42, NO 1, 2003.

[17] P. Brittenham, R. R. Cutlip, C. Draper, B. A. Miller, S. Choudhary, M. Perazolo. IT service management architecture and autonomic computing[J], BM SYSTEMS JOURNAL, VOL 46, NO 3, 2007.

[18] Biplav Srivastava, Subbarao Kambhampati. The case for automated planning in autonomic computing[C],ICAC 2005.

[19] T. Eymann, M. Reinicke O. Ardaiz, P. Artigas, F. Freitag, L. Navarro. Self-Organizing Resource Allocation for Autonomic Networks[C], Proceedingo n 14th International Workshop on Database and Expert Systems Applications, 2003.

Yalin Song was born in Henan, China, 1977. He received master degree of science in computer science and information technology from Yunnan Normal University, China. Currently, his research interests include automatic computing and image processing.

Xin He was born in Henan, China, 1974. He received the PhD degree of engineering in computer science and technology from xi'an Jiaotong University, China, in 2011. He is assistant Professor, master supervisor. Currently, his research interests include mobile computing, automatic computing and dynamic trust management.