International Journal of Grid Distribution Computing
Vol.7, no.3 (2014), pp.205-212
http://dx.doi.org/10.14257/ijgdc.2014.7.3.21

Method of Fault Detection in Cloud Computing Systems

Ying Jiang, Jie Huang, Jiaman Ding and Yingli Liu

Yunnan Key Lab of Computer Technology Application, Kunming University of Science and Technology
Faculty of Information Engineering and Automation, Kunming University of Science and Technology
jy_910@163.com, hj_0118@163.com, tjom2008@163.com, lyl2002@126.com

Abstract

Cloud computing integrates a large number of virtual resources and provides flexible service to users by pay-as-you-need. Compared with the traditional software systems, the fault detection is more complex and difficult in cloud computing systems. The fault model is helpful for people to find and distinguish the software fault. However, the current researches mostly focus on software fault model under the traditional environment. Based on the research of software fault model, this paper proposed the fault models in cloud computing systems combining with the characteristics of cloud computing. Then an improved C4.5 decision algorithm is used to distinguish the cloud fault. The experiment results showed that the fault model and improved algorithm is effective.

Keywords: *cloud computing systems, fault model, fault detection*

1. Introduction

Software testing is one of the most important technologies to assure software quality. The goal of software testing is to find faults as much as possible. Cai Kai-Yuan gave the definitions for software error, software defect and software fault [1]. Software error refers to the human error that is undesired or unacceptable in software life cycle. Software defect means the deviation exists in software, which is undesired or unacceptable. Software fault refers to a kind of undesired or unacceptable internal state that occurs in the running process. In IEEE Standard Glossary, the fault is defined as the incorrect step, processing or data definition in a computer program [2]. In ISO/IEC 15504 standards, software fault means that the software did not show the expected results [3].

It is worth noting that the difference is very small among software error, software defect and software fault. A clear distinction can not be given in many papers. Therefore, the fault in this paper means all the errors mentioned above. That is to say, all the errors that might appear in the software life cycle can be regarded as software fault.

There are many types of software fault, at the same time they are changing according to the software development. Actually, the research on fault classification and fault model attracted more and more attention. The fault model is a combination of some basic faults, and should be an abstraction of software physical errors. It is the first step to establish the fault model while detecting the faults of systems.

The remainders of this paper are organized as follows. Related works are summarized in Section 2. In Section 3, fault models in cloud computing systems are presented. In Section 4, improved C4.5 decision algorithm is described to detect the faults. The experiments are described in Section 5. In Section 6, the conclusion and future work are proposed.

ISSN: 2005-4262 IJGDC

2. Related Works

For the traditional software, the researches about the failure classification and the establishment of fault model have been relatively mature. There are several classifications from different aspects, including fault phase, consequences of fault, nature of the error and type of errors. Zhu Rong, *et al.*, established six fault models to manage and standardize fault occurred in scientific computing program, which can cover an amount of faults [4]. Nie Lin-Bo, *et al.*, proposed a defect classification method based on processing [5]. Zeng Fu-Ping et al. introduced "hierarchy theory" in defect mode classification to partition the defect mode classification by anglicizing software defect data, then proposed the defect mode in software requirements analysis phase, design phase and coding phase [6].

The above software fault models are aimed at the traditional application environment, which are the abstraction of the traditional software fault. However, under the cloud computing environment, the classification and detection of fault would be changed. In addition, some faults may just appear when software is running in cloud.

According to the definition given by the USA National Institute of standards and Technology (NIST) [7], cloud computing is a computing model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (*e.g.*, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Using cloud computing, the user can apply or release resources quickly according to the business load and pay for the used resources by pay-as-you-need, which can reduce operational cost while improving service quality.

Leah Muthoni Riungu, *et al.*, discussed the influence of cloud computing to software testing using grounded theory as research basis [8, 9]. Rao Xiang, *et al.*, presented an extraction method for fault feature called Companion State Tracer to detect fault in cloud computing system [7]. Although people begin to study the software fault of online testing under cloud computing environment, they fail to exploit the characteristics of fault model.

Cloud computing has its own features: (1) Flexible service. All the resources in cloud computing are used as a service provided to the user. The scale of the service can quickly adjustable to adapt to the dynamic change of load. (2) Resource pooling. Resources are managed in the form of a shared resource pool. (3) On-demand service. A consumer can unilaterally provision computing capabilities according to own requirements. (4) Measured Service. Cloud computing monitors user’s resource usage and users pay for the resources they used.

Compared with the existing work, the contributions of the paper are the following: (1) Based on the research of software fault model, software fault model in cloud computing systems is proposed combining with the characteristics of cloud. (2) In order to detect the faults in cloud computing, an improved C4.5 decision algorithm is proposed. (3) Cloud computing fault data are simulated to verify the correctness of the algorithm and the effectiveness of the proposed fault model.

3. Fault Models in Cloud Computing

Fault models can be viewed as the summary and abstraction of the software tester’s experience. According to the characteristics of cloud computing environment, we summarize the essential condition of the fault model in cloud. (1) The faults in the model are practical. That is to say, the faults defined in the model are real and representative of cloud computing systems. (2) The faults in the model can be detected. There is an algorithm which can be used to detect these faults.

In the studies of traditional software fault model, there are many kinds of methods to classify the software faults. Each method has different point of view and benchmark. Aimed to describe a fault from multiple aspects and match the characteristics of cloud computing, we propose fault models under cloud computing from the following four aspects.

3.1. Level

Cloud computing system can be regarded as a collection of a set of services. These services are divided into three layers.

- Infrastructure layer. The infrastructure layer mainly includes computing resources, storage and network resources, *etc.*, the entire infrastructure is provided to the user as a service, namely IaaS (infrastructure as a service).
- Platform layer. The platform layer is mainly designed for application developers. Application developers don't have to worry about resources needed when running applications. It can provide all the platform resources needed in application running and maintenance, which called PaaS (platform as a service).
- Application layer. It provides simple software application services and user interaction interface to users. This layer is called software as a service, namely SaaS (software as a service).

The three levels of cloud computing have different content, resource and function. Therefore, we divided the faults into IaaS layer fault, PaaS layer fault and SaaS layer fault. Any happened fault may only appear in one level, which cannot appear in multiple layers at the same time.

3.2. Service Stage

One of the outstanding characteristic under cloud computing is on-demand service. Users send request to the cloud computing providers according to their own needs. Then providers provide appropriate resources to users according to their requirements. Moreover, users pay for the resources. According to service stage when faults occur, we can divide the faults into service lookup stage fault, service request stage fault and service provide stage fault. This classification method fully embodies the characteristic of providing uninterrupted service. As long as the software running under the cloud, we can find which service stage the fault belongs to if the software fails.

3.3. Influence Range

Traditional software fault can be divided into negligible fault, general fault and serious fault according to its severity. For the super large-scale cluster resources and complicated scheduling policies in cloud, software fault under cloud computing environment is not a simple traditional fault. A small fault may cause big problem. According to the severity of a fault, the faults in cloud computing systems can be divided into only own influence fault, other parts influence fault and service result influence fault. Among the above faults, only own influence fault has the lowest severity. The more serious fault will affect the other components in cloud computing. Because cloud computing is providing service to the user, the most serious fault is service result influence fault.

3.4. Source

Resources in cloud are various and complex. Therefore, the fault may come from many sources. Using the hierarchical structure, the source of the fault classification is shown as Figure 1.

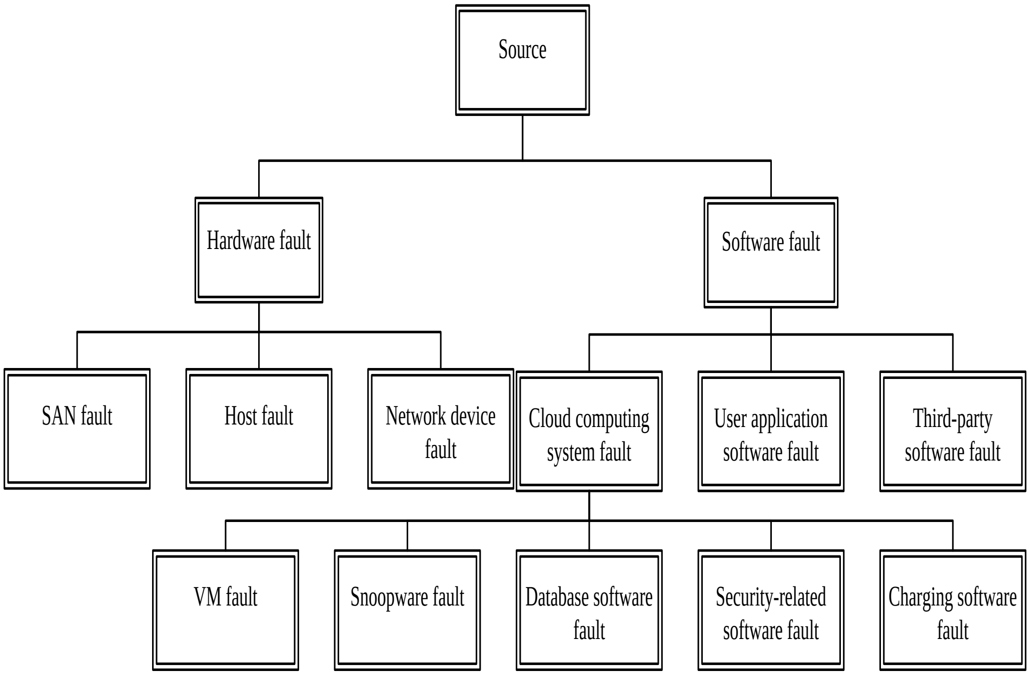

Figure 1. Source of Fault Classification

The traditional software fault classification is dividing faults into hardware fault and software fault. In cloud computing systems, we still can divide hardware fault into SAN fault, Host fault and Network device fault. The software fault can be divided into Cloud computing system fault, User application software fault and Third-party software fault. In order to achieve high quality service, there is much assistant software in cloud computing system. If this software has problems, it will produce a direct or indirect effect on users. Therefore, these faults include VM fault, Snoop ware fault, Database software fault, Security-related software fault and Charging software fault.

4. Improved C4.5 Decision Algorithm

Decision tree is a simple, effective and widely accepted classification technique. It is usually used to solve multiple attribute data classification problem. C4.5 algorithm is an improved algorithm of ID3, which was proposed by J. R. Quinlan in 1993 [10]. C4.5 algorithm uses information gain rate to select attributes, which avoids the phenomenon of choosing more attribute values when select attributes. In addition, it can deal with continuous attributes. C4.5 decision tree algorithm is end up with a decision tree, whose leaf nodes represent different categories respectively.

However, the format of instance used in C4.5 and that of used in our paper are different. For example, the instance format used in C4.5 algorithm is (Attrib1, Attrib2, Attrib3, Attrib4, Attrib5, Attrib6, Class attrib). The instance format of our paper is (Attrib1, Attrib2, Attrib3, Attrib4, Attrib5, Attrib6, Class attrib1, Class attrib2, Class attrib3, and Class attrib4).

From the above two instances, it can be seen that there is only one class attribute in C4.5 algorithm. The classification is carried out from one point of view, and then divided into different categories. According to the fault models proposed in Section 3, a fault can be classified from four aspects. Therefore, the instances used in this paper have four class attributes. As a result, using C4.5 algorithm cannot get complete classification results. So we improved C4.5 algorithm in this paper. The flow chart of the improved algorithm is shown in Figure 2.

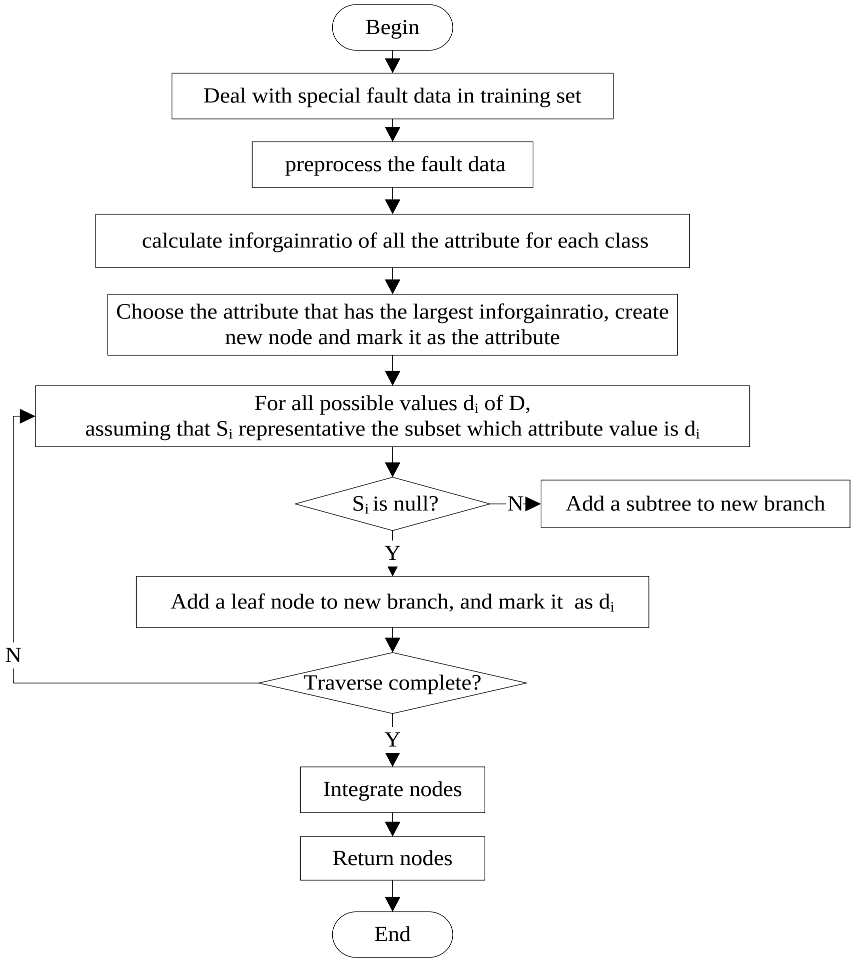
Begin
Deal with special fault data in training set
preprocess the fault data
calculate inforgainratio of all the attribute for each class
Choose the attribute that has the largest inforgainratio, create new node and mark it as the attribute
For all possible values d_i of D,
assuming that S_i representative the subset which attribute value is d_i
S_i is null?
N
Add a subtree to new branch
Y
Add a leaf node to new branch, and mark it as d_i
N
Traverse complete?
Y
Integrate nodes
Return nodes
End

Figure 2. The Control Flow Chart of the Improved Algorithm

5. Experiment Analysis

In order to verify the above method, the experiments are executed using the simulated fault data in cloud. There are six attributes in sample data, which represent six fault features of fault in cloud. The six attributes are MIPS usage, memory usage, RAM usage, bandwidth usage, response time, and the average rate of node load. The four class attributes respectively represent level class, source class, service stage class and influence range class.

Part of the sample data in training set is shown in Table 1.

Table 1. Part of the Sample Data in Training Set

MIPS%	Mem%	RAM%	BW%	RTime(s)	Node%	Class 1	Class 2	Class 3	Class 4
0.7	0. 7	0.3	0.2	35	0.6	IaaS fault	Host fault	service provide fault	effect result fault
0.4	0. 6	0.7	0.5	56	0.25	IaaS fault	VM fault	service provide fault	effect other fault
0.3	0. 62	0.9	0.4	24	0.2	PaaS fault	Host fault	service provide fault	effect result fault
0.5	0. 5	0.3	0.92	120	0.7	SaaS fault	User application fault	service request fault	effect result fault
0.7	0. 56	0.8	0.1	45	0.4	IaaS fault	Host fault	service provide fault	effect other fault
0.3	0. 9	0.6	0.78	80	0.7	SaaS fault	Network device fault	service provide fault	effect result fault
0.45	0. 8	0.7	0.6	14	0.17	IaaS fault	VM fault	service provide fault	effect result fault
0.56	0. 53	0.8	0.4	48	0.9	SaaS fault	Host fault	service request fault	effect result fault
0.3	0. 4	0.43	0.6	34	0.5	IaaS fault	VM fault	service provide fault	effect result fault
0.2	0. 85	0.9	0.9	95	0.45	IaaS fault	User application fault	service provide fault	effect result fault
0.18	0. 3	0.6	0.54	53	0.5	IaaS fault	Host fault	service provide fault	effect other fault
0.5	0. 44	0.5	0.79	86	0.84	SaaS fault	User application fault	service request fault	effect result fault
……	……	……	……	……	……	……	……	……	……

MIPS%: MIPS usage. Mem%: memory usage. RAM%: RAM usage. BW%: bandwidth usage. RTime(s): response time. Node%: the average rate of node load.

Based on the experimental data in Table 1, the decision graph is gained using the improved C4.5 algorithm, which is shown in Figure 3.

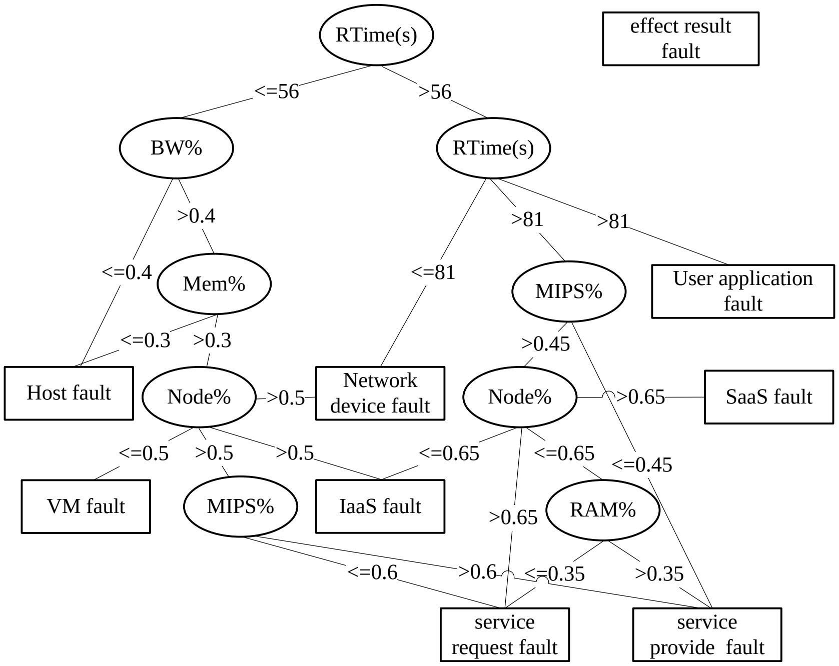

Figure 3. Decision Graph

As shown in Figure 3, improved C4.5 algorithm can realize the classification of faults from four aspects and produce a decision graph. We can classify new fault data by traversing the graph.

In order to verify the correctness of the improved C4.5 algorithm, new data are simulated. The verification result is shown in Table 2:

Table 2. Verification Result

MIPS%	Mem%	RAM%	BW%	RTime(s)	Node%	Class 1	Class 2	Class 3	Class 4
0.55	0.57	0.3	0.5	90	0.4	IaaS fault	User application fault	service request fault	effect result fault
0.6	0.4	0.42	0.6	85	0.9	SaaS fault	User application fault	service request fault	effect result fault
0.3	0.4	0.7	0.9	54	0.65	IaaS fault	Network device fault	service request fault	effect result fault
0.7	0.6	0.34	0.7	45	0.8	IaaS fault	Network device fault	service provide fault	effect result fault
0.6	0.3	0.8	0.5	38	0.1		Host fault		effect result fault

As shown in Figure 3, all faults belong to “effect result fault”. Therefore, the values of class 4 in Table 2 are filled in “effect result fault”. Then the RTime(s) is the first attribute should be considered. For example, the RTime(s) value of the first fault data in Table 2 is 90. The branch of >56 will be selected. After considering the attribute RTime(s), which value is 90, the branches of >81 are selected. Therefore, the class “User application fault” is filled in class 2. Then MIPS% should be considered. As the value of MIPS% is 0.57, the branch of >0.45 will be selected and the value of Node% will be considered. According the 0.4 of Node% in Table 2, the branches of <=0.65 are selected, so “IaaS fault” can be filled in class 1 and the value of RAM% should be considered. Finally, according to the 0.3 of RAM% value and the branch of <=0.35, the class “service request fault” is gained and filled in class 3.

6. Conclusion and Future Work

In order to deal with the faults in cloud computing systems, this paper proposes fault models about cloud from four aspects. Then an improved C4.5 algorithm is implemented to detect the fault. The experiments prove that our model and algorithm is effective. In the future, the new fault will be identified compliance with cloud computing systems.

Acknowledgements

This research is sponsored by the National Science Foundation of China No. 60703116 and 61063006, and the Application Basic Research Plan in Yunnan Province of China No. 2013FZ020.

References

[1] K. Y. Cai, “The Foundations of Software Reliability Engineering”, Tsinghua University Press, Beijing, **(1995)**.

[2] IEEE. IEEE Std. IEEE Standard Glossary of Software Engineering Termiology, **(1990)**.

[3] K. El-Emam and I. Garro, “ISO/IEC 15504”, International Organization for Standardization, **(1999)**.

[4] R. Zhu and S. Y. Xu, “Establishing the fault model of software testing”, Computer engineering and Applications, vol. 17, **(2003)**, pp. 67-92.

[5] L. B. Nie and M. R. Liu, “Research on software defects classification”, Application Research of Computers, vol. 7, **(2004)**, pp. 84-86.

[6] F. P. Zeng, H. L. Jin, M. Y. Lu. Research on software defect mode. Computer Science, vol. 38, no. 2, **(2011)**, pp. 127-130.

[7] X. Rao, H. M. Wang, Z. B. Chen, Y. F. Zhou, H. Cai, Q. Zhou and T. T. Sun, “Detecting Faults by Tracing Companion States in Cloud Computing Systems”, Chinese Journal of Computers, vol. 35, no. 5, **(2012)**, pp. 856-870.

[8] M. R. Leah, T. Ossi and S. Kari, "Research Issues for Software Testing in the Cloud", 2nd IEEE International Conference on Cloud Computing Technology and Science, **(2000)** November 30 -December 3, pp. 557-564.

[9] M. R. Leah, T. Ossi and S. Kari, "Software Testing as an Online Service: Observations from Practice', Software Testing. 2010 Third International Conference on Verification, and Validation Workshops, **(2000)** April 6-10, pp. 418-423.

[10] J. R. Quinlan, "C4. 5: Programs for Machine Learning", San Mateo, CA: Morgan Kaufmann, **(1993)**.

Authors

Ying Jiang, Ph. D. and professor. Her research interests include software engineering, software testing and service computing.

Jie Huang, Master candidate. Her research interests include software testing technology and service computing.

Jiaman Ding, Master and lecturer. His research interests include software engineering and cloud computing.

Yingli Liu, Master and lecturer. Her research interests include software engineering and cloud computing.