Acta Technica 61, No. 4B/2016, 291–300

The reliability model of cloud service system with recovery mechanism[1]

Dan Liu[2], Xin Sui[2,3], Huan Wang[2], Xu Di[2], Li Li[2,4]

Abstract. A variety of failure issues of CSS (cloud service system) is analyzed, and then proposed the FRRM reliability mode based on the failure nodes. In the model, the faults of CSS's soft/hardware and LAN communication are all discussed. Moreover, in order to reflect the influence of time on system reliability, the paper introduces time influence factor. For effective verification of FRRM reliability model, the paper makes a comparison experiment with the classical NHPP reliability model. The simulation results show that the FRRM reliability model is effective. It can make the reliability of cloud service system optimized.

Key words. Cloud service system, reliability model, modeling, faults.

1. Introduction

How to improve the reliability of cloud service system (CSS) had attracted the attention of many scholars both at home and abroad. Scholar He Li proposed a cloud service reliability measurement model which was based on checkpoint rollback strategy. The cloud service reliability model was established with the checkpoint rollback strategy over the changed system service reliability problems which were caused by failure nodes in CSS's virtual machine tasks. The theoretical deduction proved that the service reliability could be optimized when the test point setting time was increased by a certain index, but the failure of the calculation node was considered only in the study. Scholar Yadan Zhang proposed a hidden Markov

[1]This work was supported in part by the project of "12th Five-Year" planning of the Education Science of Jilin Province (GH14049), the science and technology project of "13th Five-Year" planning of the Education Department of Jilin Province (2016), the key project of "13th Five-Year" planning of the Education Science of Jilin Province (ZD16024), and Key Science and Technology Project of Jilin Province (20160204019GX).

[2]Changchun University of Science and Technology, College of Computer Science and Technology, 130022, China

[3]Jilin Provincial Institute of Education, 130022, China

[4]Corresponding author: Li Li, Changchun University of Science and Technology, College of Computer Science and Technology, 130022, China, ll@cust.edu.cn

model based on simulated annealing algorithm in the problem of faults detection. By solving the initial parameter of hidden Markov model, the detection accuracy was improved. Most scholars paid more attention to the faults of a single reason in the existing research. However, the existing reliability model could hardly solve the problem because of the failure of CSS's heterogeneity was diversified. From the viewpoint of system failures, the paper analyzes the failure types of CSS, and considers the influence of time on the system reliability. Finally, the reliability model called FRRM [1–4] was proposed.

2. Problem formulation

2.1. The architecture of CSS

CSS can provide services to the users because of the effective management, allocation and scheduling of virtual resources. The operation process is as follows: when cloud computing management platform (CCMP) receives users' service request, the task can be divided into multiple child-tasks, in order to improve the efficiency of the work. Then, according to the resource requirements of child-tasks, scheduling of some executive order or LAN-communication child-tasks which is based on each resource node can start. After that, task results commit into CCMP and are integrated, and the final result redrawn to the users. During the execution of child-tasks, CCMP nodes or LAN all may exhibit failure, which deteriorates the quality of the CCMP service [5–6]. So the paper established the reliability of the system services model based on the analysis of node failure.

2.2. The analysis of CSS node failure

In the actual cloud computing environment, due to the software, hardware and other reasons, the failure problem in the implementation process of the cloud service system is inevitable. Therefore, it is necessary to analyze the failure mechanism of the nodes according to the behavior characteristics of the nodes. The paper discusses the reliability of CSS from the perspective of resource allocation. Node failures may be caused by software or physical breakdown (including hardware or LAN fault), and they can be separated into recoverable failures and no-recoverable failures [7–9].

3. The proposed model

According to the definition of reliability in reliability engineering theory, cloud services platform can be defined as: CSS had the ability within the specified time to provide services to users in a given condition. CSS provides users with computing, storage, applications and other services in the form of service. All these services should be completed in the shortest possible time, and redraw the final result to the users. Based on the above situation, the paper makes the following assumptions: in the cloud service system, the virtual machine manager who is responsible for the task distribution is reliable, and the resource nodes are independent of each other. The child-tasks are independent of each other. All kinds of faults are independent during each child-tasks' implementation processes. It can be considered as a discrete probability distribution during the failure and recovery processes of each resource nodes and LAN.

3.1. The reliability model of CSS

3.1.1. Brief introduction of modeling method. In essence, cloud computing is a fusion of distributed processing, parallel computing and virtualization technology. It is a new computing model of information system development and evolution, it has a significant scalability, flexible and easy to configure. In CSS, unavailable computing resources, software failure, link failure, etc. are likely to hinder the completion of the task and the reliability of cloud model. Therefore, as a focal point of node failure, the paper proposes an abstract description of the reliability of the service system and pushes out a reliability system model. In the process, the reliability of cloud computing service system can be expressed as

$$R(t) = 1 - F(t)\,, \tag{1}$$

where $R(t)$ stands for the CSS reliability function and $F(t)$ denotes its distribution function.

3.1.2. The establishment of reliability model. CSS divides the task Q submitted by user into a number of m child-tasks, and names the set of child-tasks as $Q = q_1, q_2, q_3, ..., q_m$. Then, m child-tasks are assigned to the l resource nodes; there is an inevitable redundancy allocation in the allocation process. And the handle time representing the allocation time of kth child-task to node i can be denoted as $\tau_{i,k}$.

$$\tau_{i,k} = \frac{W_i}{S_k}\,, \tag{2}$$

where W_i stands for ith child-task workload and S_k is the computing speed of the kth node.

Due to CSS usually adopts the fault-tolerant technique. Therefore, in addition to the execution time $TE_{i,k}$ and total failure recovery time $TR_{i,k}$ the total LAN communication time $TS_{i,k}$ may also be added, and this is the actual total execution time T of child-task. In interval $(0, T]$ it is assumed that the total number of recoveries of kth node N_k is a random variable, such that $N_k = n > 1$. In the course of ith child-task execution, two node failure cases happen: one is a recoverable failure, and the other is a non-recoverable failure. So, the the total time T of the ith child-task execution on node k can be written as

$$T = \{ \begin{array}{cc} \tau_{i,k} + TS_{i,k} & N_k = 0 \\ TE_{i,k} + TR_{i,k} + TS_{i,k} & N_k > 0 \end{array} . \tag{3}$$

From the hypothesis from the last section, the failures of soft/hardware and LAN communication obey certain Poisson distribution. And during the execution of the process, the failure distribution parameters are constant. In many kinds of failures, only the hardware failure is recoverable. Therefore, in the time interval $(0-T]$ there holds

$$TE_{i,k} = \sum_{1}^{n+1} TE_{i,k}^{(j)} . \tag{4}$$

and

$$TR_{i,k} = \sum_{1}^{n+1} TR_{i,k}^{(j)} . \tag{5}$$

where $TE_{i,k}^{(j)}$ stands for the jth execution time of the ith child-task at the kth node, $j \in [1, n+1]$, and $TR_{i,k}^{(j)}$ denotes the jth failure recovery time.

It can be assumed that $TE_{i,k}^{(j)}$ is independent and identically distributed random variable, and the parameter of the exponential distribution is λ_k. The value of $TR_{i,k}^{(j)}$ is also independent and identically distributed random variable, and the parameter of the exponential distribution of μ_k, which is called recovery rate. Finally, $TS_{i,k}^{(j)}$ is also independent and identically distributed random variable, and the parameter of the corresponding exponential distribution is ν_k. In order to describe the failure recovery capability, the random variable $X_k^{(j)}$ is defined, which is used to express whether the node k is recoverable or not. If the kth node becomes non-recoverable, it is expressed by $X_k^{(j)}$. Then, the child-task distributed at the failure node is stopped. So, when $g > j$, then $TE_{i,k}^{(g)} = 0$ and $TR_{i,k}^{(g)} = 0$. When the node k becomes recoverable, it is expressed by $X_k^{(j)} = 0$.

In CSS, the failure recovery has a certain probability. So it should be divided into different situations to discuss. It should be also considered the influence of reliability time when discussing the system reliability model. Therefore, the paper introduces the time influence factor $\delta^{(t-\tau_{i,k})}$, where $\delta \in (0, 1)$.

All the faults belong to non-recoverable failures except for hardware failures. In interval $(0-T]$, it is assumed that the ith child-task is executed at the kth node and follows the exponential distribution of the parameter λ_k. Then, the probability of software failure is

$$F_{\mathrm{s}} = \delta^{(t-\tau_{i,k})} \mathrm{e}^{-\lambda_k \tau_{i,k}} . \tag{6}$$

Likewise, the probability of LAN failures of system communication is

$$F_{\mathrm{c}} = \delta^{(t-\tau_{i,k})} \mathrm{e}^{-\lambda \nu_k \tau_{i,k}} . \tag{7}$$

Then, it is analyzed the probability of hardware failure denoted as F_{h}. Because of certain recoveries of the hardware failure, not only the probability of non-failure node should be considered but also the probability of failure at the recovered node. So, $F_{\mathrm{h}} = F_{\mathrm{tf}} + F_{\mathrm{fr}}$.

Here, F_{tf} represents the failure of node, and the exponential distribution of the parameter was ρ_k, so that $F_{\mathrm{tf}} = \mathrm{e}^{-\rho_k \tau_{i,k}}$. Symbol F_{fr} stands for the failure at the recovery node, so that $F_{\mathrm{fr}} = F_{\mathrm{fr}}^{(1)} + F_{\mathrm{fr}}^{(2)} + F_{\mathrm{fr}}^{(3)} + \ldots + F_{\mathrm{fr}}^{(n)}$. Generally, $F_{\mathrm{fr}}^{(n)}$ represents the recoverable failure to ith execution of the child-task on kth node and the number of this failure is n. So, it can be written as

$$F_{\mathrm{fr}}^{(n)} = F_{\mathrm{fr}}\{\sum_{j=1}^{n} TE_{i,k}^{(j)} < \tau_{i,k}\,,\ \sum_{j=1}^{n} TE_{i,k}^{(j)} + TE_{i,k}^{(n+1)} > \tau_{i,k}\,,\ \sum_{j=1}^{n} X_k^{(j)} = 0\}\,. \quad (8)$$

As the times $TE_{i,k}^{(j)}$, $(j = 1, 2, \ldots n)$ are independent of each other, it can be taken as the Erlang distribution of random variables which follows parameters n, ρ_k. According to this, $\sum_{j=1}^{n} TE_{i,k}^{(j)}$ and $TE_{i,k}^{(n+1)}$ are independent of each other. So, the joint density can be written as

$$f_{i,k}^{(n)} = \begin{cases} \mathrm{e}^{-\rho_k x - \rho_k y \frac{\rho_k^2 (\rho_k x)^{n-1}}{(n-1)!}} & x \geq 0, y \geq 0 \\ 0 & \text{elsewhere} \end{cases}\,, \quad (9)$$

$$F_h = \delta^{(t-\tau_{i,k})} \sum_{n=1}^{\infty} F_{\mathrm{fr}}^{(n)} \cdot \mathrm{e}^{-\rho_k \tau_{i,k}} = \delta^{(t-\tau_{i,k})} \mathrm{e}^{(x_k - 1)\rho_k \tau_{i,k}}\,. \quad (10)$$

Therefore, in interval $(0, T]$, the probability F of total failure of the ith child-task at the kth node is:

$$F = F_{\mathrm{s}} F_{\mathrm{c}} F_{\mathrm{h}} = \delta^{3(t-\tau_{i,k})} \mathrm{e}^{-\lambda_k \tau_{i,k} - \lambda \nu_k \tau_{i,k} + (x_k - 1)\rho_k \tau_{i,k}}\,. \quad (11)$$

The above paragraph just discussed the failure probability of ith task at kth node. In the actual system, on must consider the running mode of parallel execution of multi-tasks and multi-nodes. In the process of distribution and execution, not only redundant allocation should be considered, but also the influence of time on the system reliability. Therefore, the reliability R of CSS should be written as

$$R = \prod_{i=1}^{m} [1 - \prod_{k=1}^{l} (1 - \delta^{3(t-\tau_{i,k})} \mathrm{e}^{-\lambda_k \tau_{i,k} - \lambda \nu_k \tau_{i,k} + (x_k - 1)\rho_k \tau_{i,k}})]\,. \quad (12)$$

4. Results

In order to verify the validity of the FRRM model, the paper designes the simulation experiment for it. The authors research the impact of node failure recovery mechanism on the reliability of CSS. In the same experimental conditions, FRRM is compared with the model of non-homogeneous Poisson process (NHPP).

The initial conditions: in the simulation experiment, the task number is set 0∼100, and the allocation of resource nodes for CSS is 0 ∼ 200. The value of $\lambda_k \in [0.001, 0.004]$, $\mu_k \in [0.001, 0.003]$, $\nu_k \in [0.002, 0.003]$. All the initial conditions of the experiment followed these conditions.

4.1. The influence of node failure recovery rate on reliability

In the simulation experiment, it can be obviously seen that the probability of successful execution of the FRRM model is higher than that of the NHPP model under the same failure recovery rate. When the failure recovery rate approaches 0, it indicates that the failure recovery mechanism has no effect, and the successful execution probability is low; when the failure recovery rate increases, the probabilities of successful executions of both models also increase. When the failure recovery rate approaches 1, both probabilities of successful executions of the two model reach the maximum values. The process shows that the failure recovery mechanism plays an important role as the probabilities of successful executions of both models do not reach 100 %. It describes that two failures of software and LAN belonging to no-recoverable faults play a role, as shown in Fig. 1.

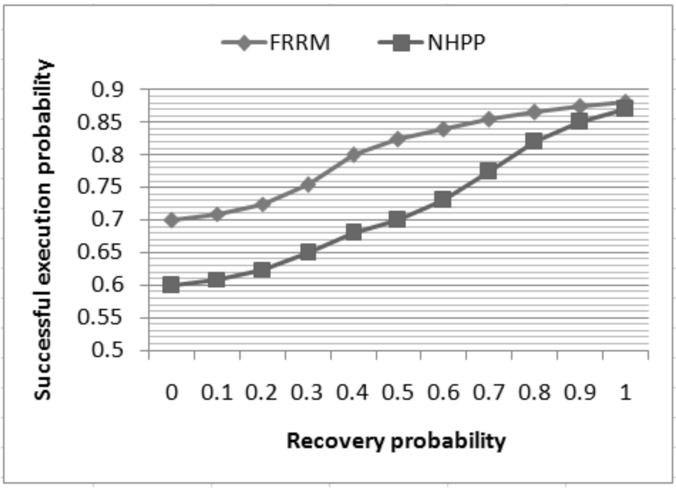

Fig. 1. Relationship between recovery probability and probability of successful execution

4.2. Impact of number of tasks requested by user to system reliability

Through simulation results, with the increase of the number of tasks requested by the user, the successful execution probability is reduced. In the process of experiment, when the number of requests increased from 10 to 50, the system reliability of both the FRRM model and NHPP model decreased, and the downward trend was relatively slow. But when the number of requests increased from 50 to 100, the system reliability of the two models significantly decreased. The successful execution probability of the FRRM model was reduced by nearly 8 % and that of the NHPP model was reduced by 25 %. In contrast, the probability of successful execution of the FRRM model is higher than that of the NHPP model, and this shows that the

FRRM model performs better, as is shown in Fig. 2.

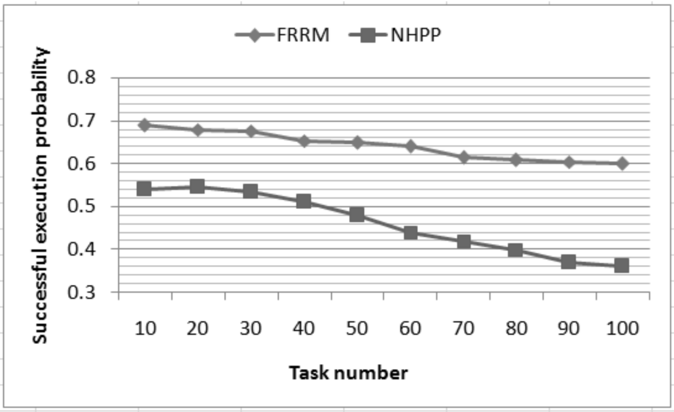

Fig. 2. Relationship between number of requested tasks and probability of successful execution

4.3. *Impact of number of CSS resources on system reliability*

Through simulation results, the probability of successful execution was also increased with the increase of the number of system resources. In the process of simulation experiment, the reliability of the cloud service system under the two models was both improved under the increasing number of available resource nodes. In contrast, the probability of successful execution of the FRRM model was higher than that of the NHPP model. Especially when the number of resources was between from 0 to 80, the probability of successful execution increased by nearly 21 %, as is shown in Fig. 3.

4.4. *Verification of time influence factor on system reliability*

In the FRRM model, the time influence factor $\delta^{(t-\tau_{i,k})}$ was added. The experiment compared the reliability model before and after adding the attenuation factor in the simulation experiment, as shown in Fig. 4.

5. Discussion

During the simulation experiment, NHPP was chosen to compare with the FRRM model. As the modeling object of NHPP was similar to the research object described in the, they were both targeted by modeling with “discrete event”. The experiment results showed that FRRM model could effectively improve the software reliability of CSS and had an advantage over the others. But in the experiment, it was found

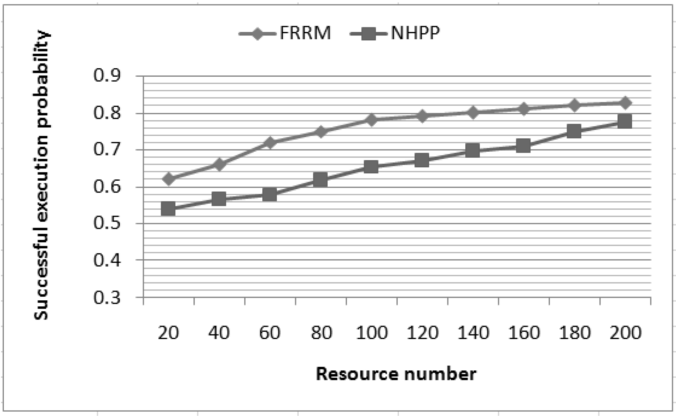

Fig. 3. Relationship between number of requested resources and probability of successful execution

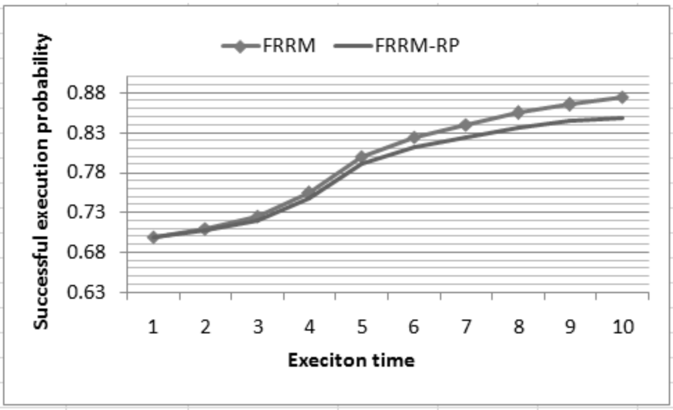

Fig. 4. Influence of time influence factor on system reliability

that the convergence of the algorithm was not very well. The optimization algorithm will be researched in further periods of work in the field.

6. Conclusion

The paper mainly researched the reliability of CSS and established the reliability model by analyzing the failure mechanism of CSS. In the process of model building, the relationship between the operation time and system reliability was also considered. In order to verify the validity of the model, a simulation experiment was carried out. In the experiment, the proposed model proved to improve the reliability of CSS.

References

[1] G. Levitin, Y.-S. Dai: *Service reliability and performance in grid system with star topology.* Reliability Engineering and System Safety *92* (2007), No. 1, 40–46.

[2] A. Dogan, F. Ozquner: *Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing.* IEEE Trans. Parallel and Distributed Systems *13* (2002), No. 3, 308–323.

[3] Y. S. Day, M. Xie, K. L. Poh: *Reliability of grid service systems.* Computers and industrial Engineering *50* (2006), Nos. 1–2, 130–147.

[4] N. Yadav, V. B. Singh, M. Kumari: *Generalized reliability model for cloud computing.* IJ Computer Applications *88* (2014), No. 14, 13–16.

[5] H. R. Faragardi, R. Shojaee, H. Tabani, A. Rajabi: *An analytical model to evaluate reliability of cloud computing systems in the presence of QoS requirements.* Proc. IEEE/ACIS IC Computer & Information Science, 16–20 June 2013, Niagata, Japan, *446*, 315–321.

[6] J. Zhao, L. Hu, Y. Ding, G. Xu, M. Hu: *A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment.* Plos One *9* (2014), No. 9, paper e108275.

[7] W. Wang, G. Zeng, D. Tang, J. Yao: *Cloud-DLS dynamic trusted scheduling for cloud computing.* Expert Systems with Applications *39* (2012), No. 3, 2321–2329.

[8] X. Qiu, Y. Dai, Y. Xiang, L. Xing: *A hierarchical correlation model for evaluating reliability, performance, and power consumption of a cloud service.* IEEE Trans. Systems, Man & Cybernetics Systems *46* (2016), No. 3, 401–412.

[9] J. Rahme, H. Xu: *A software reliability model for cloud-based software rejuvenation using dynamic fault trees.* IJ SW Engrg. & Knowledge Engrg. *25* (2015), No. 10, 1491–1513.

Received November 16, 2016