A Fault Repair Method for Workstation Cluster Based on Probabilistic Model Checking

Xi Wang¹, Ting Chen¹ and Chengtian OuYang¹

¹School of Information Engineering, Jiangxi University of Science and Technology, Jiangxi, China

*Corresponding author e-mail: wang_xi_happy@163.com, dustinchan123@163.com, oyct@163.com.

Abstract. To analyze the component modules and maintenance unit modules in the workstation cluster, a fault repair method for workstation cluster based on probabilistic model checking is proposed. In the proposed method, the queue model is introduced when workstations are waiting for repairing, and different priorities are assigned according to the importance of the component functions. The formal model of the system is established by an extended continuous time Markov chain, the attributes of the system are described by continuous stochastic logic, and the fault repair module is verified by PRISM. The experimental results show that the proposed method can greatly reduce the time required for the maintenance process, and improve the maintenance efficiency and the fault tolerance of the system.

1. Introduction

Cluster technology is characterized by reliability, availability and expansibility [1]. High reliability cluster systems require higher standards of reliability and disaster defense. When a node's computing or service fails or the system fails, the cluster system needs to spread the tasks running on the failed node to other normal nodes, continue to complete the current service or calculation, and can be repaired in a short time.

Model checking technology has been used in the analysis and verification of computer hardware, communication protocols, system’s safety design [2-3]. Probabilistic model checking usually builds a probabilistic model for the system, then uses the steady-state and instantaneous probability calculations to complete the quantitative verification [4, 13]. The model checking tool PRISM is usually adopted [5, 11]. Probabilistic model checking is also used to detect the correctness and performance of various stochastic distributed algorithms [6], service quality in communication networks and protocols [7-8], and performance analysis in areas such as power management systems [9].

Based on the research [10], this paper studies the fault repair module in high reliability workstation cluster, and proposes a fault repair method based on probabilistic model checking. In this method, component module and maintenance unit module in workstation cluster are analyzed. And they are formally modeled and analyzed by using model checking tool PRISM. Additionally sub algorithm FCR and sub algorithm REP are proposed. During the repair process, multiple faulty workstations waiting for repairing are put into the proposed queue model, then different priorities are assigned...
according to the importance of the function for the component. Finally, scheduling strategy of the maintenance unit is adjusted. The formal model of the system is established by an extended continuous time Markov chain. The attributes of the system are described by continuous stochastic logic, and the fault repair module is verified by PRISM. The experimental results show that the proposed method not only shortens the repair time, but also makes the system faster recover to normal working state. So the method provided in this paper is an available way to improve the maintenance efficiency and the ability of system’s fault tolerance.

2. Algorithm description
In this paper, the fault repair module of workstation cluster system is mainly composed of component module and repair module. Correspondingly, the proposed algorithm is mainly composed of sub algorithm FCR and sub algorithm REP.

2.1. Sub algorithm FCR
The continuous time Markov chain (CTMC for short) model [12] is extended, then extended continuous-time Markov chain (ECTMC for short) is used for modeling, and the properties of the workstation cluster are described by continuous stochastic logic (CSL for short). The definitions of CTMC and ECTMC are as follows.

**Definition 1** CTMC is a tuple \( C = (S, s_0, R, AP, L) \), where \( S \) is a finite set of states. \( s_0 \) is the initial state. \( R: S \times S \rightarrow R_{\geq 0} \) is the migration rate function. \( AP \) indicates an atomic proposition. \( L: S \rightarrow 2^{AP} \) is a state identification function.

**Definition 2** ECTMC is a tuple \( ECTMC = (W, S, s_0, R, Act, status, T) \), where \( S, s_0, R \) are the same as Definition 1. \( W \) represents the set of components in the system. \( Act = \{work, repair\} \), work indicates the normal working behavior of the system, and repair indicates the maintenance behavior of the system. \( status: S \rightarrow 2^{Act} \) represents the status identification of the components. \( T \) represents time.

In the workstation cluster system, the number of components is \( n \), and an ECTMC model is used to describe the process of failure, repairing, and reoperation of each component. In the ECTMC model, \( w(1, 2, \ldots, n) \) is the component queue, where \( w(i) \in W \) is the \( i \)th component. \( S = \{run, failure\} \) indicates that there are two kinds of operation states of components. \( s_0 \) is the initial state of the component. Rate \( \lambda_i \in R \) is the rate for state migration. State identification \( status; \{work, repair\} \in status \) represents whether the current component \( w(i) \) is working properly or needs to be maintained. \( t \in T \) represents the time. The input and output of the sub algorithm FCR are as follows.

Input: \( w(i), status; \{work, repair\}, T \).
Output: \( S, status(i) \).

Sub algorithm FCR starts from the current state of the component \( w(i) \). With the passage of time \( t \), \( w(i) \) fails at the migration rate \( \lambda_i \). \( status(i) \) and \( S \) are changed. The maintenance module runs after the failure occurs, if \( w(i) \) is repaired, then \( status(i) \) and \( S \) are changed again. Specific algorithmic description for FCR is shown in Figure 1.

2.2. Sub algorithm REP
Because there is only one maintenance unit, improving the efficiency of the maintenance unit can greatly improve the performance of the system. If multiple components fail at the same time, a reasonable scheduling strategy is needed to enable the system to return to normal state in a very short time. Therefore, this paper introduces queue model to maximize the efficiency of maintenance unit when multiple components fail. According to the functional distinction of system components, it can be divided into three priority levels:

1. workstation. The workstation sub cluster is consisted of multiple workstations.
2. switch. Switch components connect workstations.
3. backbone network. When a standby sub cluster is required, the backbone network connects the two sub cluster.
As a key component connecting two sub clusters and having only one, the backbone network has the highest priority of 3. Secondly, the priority of the two switches connected to the workstation is 2. There are usually a good many failures in the workstation due to more liable to failure. When a single workstation fails, it will not affect the operation of the whole system, so its priority is 1. When multiple workstations fail, they enter the queue to wait for repair. When they encounter higher priority component obstacles, the repair unit needs to repair the components with higher priority immediately after the current component has been repaired. The maintenance unit also has two states: idle and working. According to the different repaired components, the working state can be divided into four different kinds of states, which can be distinguished by different value of variable \( r \). \( r=0, 1, 2, 3 \) indicates idle, repairing workstation, repairing switch and repairing backbone network respectively. The queue length is represented by variable \( q \). The value of variable \( q \) cannot exceed the total number of workstations.

Input: \( q, r \), where the migration process is \{arrival, start\( (i) \), repair\( (j) \)\}; \( T \) represents time, it is synchronized with the time of component module in sub algorithm FCR. \( t \) is a time variable.

Output: Real-time output of the current repair module status \( r \) and queue length \( q \).

Starting from \( t=0 \), if the workstation components fail, the status of the maintenance module changes to the mode of repairing workstation, and the faulty workstation enters the queue and leaves the queue after repairing. If other components fail, the status of maintenance module changes to the mode of repairing other components, after the repair is completed, the next faulty component is repaired till \( t = T \). The specific algorithm is shown in Figure 2.

3. Experiment
This paper takes the scenario of workstation cluster [9] as a case study, and its scenario schematic diagram is shown in Figure 3.

The system is a highly reliable workstation cluster, which is connected by two sub clusters through the backbone network. The sub cluster is composed of several sub workstations, which are connected
by star topology. The switch serves as a bridge between the sub cluster and the backbone network. Each component is likely to collapse, but only one repair unit is responsible for repairing each component. Based on the characteristics of high reliability cluster, in order to minimize the time of service interruption and ensure the uninterrupted demand for computing services, it is necessary to maintain the cluster system at a relatively high level of quality of service (QoS for short).

![Figure 3. Scenario of workstation cluster.](image)

![Figure 4. Probability of encountering the same problem.](image)

Therefore, the following constraints should be followed: i) in order to provide the lowest quality of service \((\text{QoS}_{\text{min}}\text{for short})\), at least \(k (k \geq (3/4)\ast N)\) workstations are required to maintain normal working, and \(k\) workstations are connected. If the current number of sub cluster workstations is less than \(k\), it is necessary to connect another sub cluster through the backbone network to ensure that the number of workstations working normally is larger than \(k\). Otherwise, lower than \(\text{QoS}_{\text{min}}\) cannot ensure the normal operation of the whole system. The whole system can be restarted only when the number of workstations is higher than that of \(\text{QoS}_{\text{min}}\) after repairing the faults by the maintenance unit; ii) in order to provide high quality of service \((\text{QoS}_{\text{max}}\text{for short})\), at least \(h (k < h \leq N)\) workstations are required to keep working normally.

The system can be divided into six main components: two sub clusters, two switches, backbone network and maintenance unit. Workstation module, switch module, backbone module and maintenance module can be modeled in PRISM, according to the different functions of components. These modules can interact with each other, and the parallel combination of each element constitutes the whole system.

According to the sub algorithm FCR and sub algorithm REP proposed in this paper, each component of the system is modeled separately. The attributes of the system are verified by PRSIM.

| Parameter    | Meaning                                | Parameter value |
|--------------|----------------------------------------|-----------------|
| \(N\)        | Preset value of workstation number     | actual quantity |
| left\_mx     | Number of workstations in the left sub-cluster | \(N\)           |
| right\_mx    | Number of workstations in the right sub-cluster | \(N\)           |
| \(n\)        | Capacity of queue                      | \((0.5\sim2)\ast N\) |
| \(k\)        | \(\text{QoS}_{\text{min}}\)            | \((3/4)\ast N\)  |
| \(h\)        | \(\text{QoS}_{\text{max}}\)            | \(N\)           |
| ws\_fail     | Average failure probability of single workstation | \(1/500\)       |
| switch\_fail | Average failure probability of switch   | \(1/4000\)      |
| line\_fail   | Average failure probability of backbone network | \(1/5000\)      |

If the value of \(\text{QoS}_{\text{min}}\) is \((3/4)\ast N\), neither of the workstation sub clusters is working properly, then the minimum value of queue length is the total number of workstations that failed in the whole system,
i.e. \((1/2)*N\), the maximum value of queue length is the total number of workstations. Consequently, capacity of queue is \((1/2)*N \leq n \leq 2*N\).

In the experiment, the probabilistic model checking tool PRISM is used to verify and analyze the attributes of the system, and the results are compared with those in literature [10]. The experimental process and results are as follows. In Figure 4-6, the solid line represents the experimental results in literature [10], and the dotted line represents the experimental results in this paper.

(1) Attribute 1: \(P=\text{true U}[T,T] \text{"minimum"} \{\text{"minimum"}\}\{\text{max}\}\]

Attribute 1 describes the maximum probability for encountering the same problem again after \(T\) time units when the number of workstations working normally in the system is less than QoS min, as shown in Figure 4.

In Figure 4, the green and blue solid lines represent respectively the different number of workstations, and the probability changes with time. Two dotted lines represent the probability trend after queue introduction under the same conditions. It is found that the probability of encountering the same fault again decreases greatly in the same time after queue introduction. Therefore, the introduction of queue can effectively reduce the probability of encountering the same fault.

(2) Attribute 2: \(P=?\ \text{true U}\\leq T \text{ "premium"} \{\text{"minimum"}\}\{\text{min}\}\]

Attribute 2 describes the probability of the system from QoS min to QoS max within \(T\) time units. In attribute 2, the label "premium" represents that the formula satisfies QoS max, and the label "minimum" represents that the formula satisfies QoS min. The experimental results are shown in Figure 5.

From Figure 5, the solid line shows the trend of probability change of different workstation number \(N\) over time. With the increasing of workstation number, the more time it takes to reach the same probability \(P\), that is, the number of workstations will affect the time when low-level QoS is restored to high-level QoS. Under the same experimental conditions, the dotted line represents the change of probability after the queue is introduced. Thus, after introducing queue model, the time required for the system to recover from the lowest QoS to the high-level QoS is greatly reduced under the same probability \(P\). That is to say, the introduction of queue in this paper improves the efficiency of maintenance, reduces the interaction between component modules and repair unit, and improves the disaster resistance of the whole system. Even if there is a fault, it can recover quickly and return to normal working condition.

(3) Attribute 3: \(P=? \{\text{"minimum"} U\\geq T \text{ "minimum"} \{\text{"minimum"}\}\{\text{max}\}\]

Attribute 3 describes the probability that it takes more than \(T\) time units to recover to normal operation when the QoS of system is below standard. The experimental results are shown in Figure 6.

In Figure 6, the solid line reflects that the time required for different number of workstations returning to normal state increases with the number of workstations increasing. The dotted line indicates that the time required after queue being introduced is much less than that before. From the experimental analysis of Figure 2-4, it can be seen that when the number of fault workstations is large,
due to the introduction of queue model in the algorithm proposed in this paper, the maintenance scheduling of fault components becomes more orderly, and the data interaction among components is reduced, so that maintenance units do not need to spend more time to detect other components. It improves the efficiency of maintenance unit, thus improving the reliability and disaster resistance of the system.

4. Conclusion
Taking the scenario of workstation cluster as a case study, this paper proposes a workstation cluster fault repair method based on probabilistic model checking. By defining the extended continuous-time Markov chain model and the proposed sub algorithm FCR and sub algorithm REP, the operation scenario of workstation cluster is formalized as the ECTMC model. The attributes of the system are described by CSL, and the attributes of the system are verified and analyzed by the probabilistic model checking tool PRISM. The experimental results show that in the interaction between component modules and maintenance unit, the queue model is used in the proposed method. When the number of workstations with faults is large, they enter the queue to wait for repair. According to the importance of component function, different priorities are divided and the scheduling strategy of repair unit is adjusted. This method reduces the interaction time between each functional component and repair unit, which greatly cuts down the repair time, further improves the maintenance efficiency. In addition, the proposed method enables the whole system to return to normal work faster, and improves the capability for fault tolerance and disaster resistance of the system. Compared with the existing algorithms, the proposed method has some advantages and provides a better solution for improving system performance and reliability.

Acknowledgment
This work has been supported by the national natural science foundation of China (No. 61462034, No. 61561024), and Scientific Research Fund of Jiangxi Provincial Education Department, China (No. GJJ160632, No. GJJ170517).

References
[1] Zhang X, Hu Z, Zheng J and Tang S 2003 Comput. Eng. 4 26
[2] Clarke E M, Grumberg O, Hiraishi H, Jha S, Long D E, McMillan K L and Ness L A 1993 Proc. of Chdl’93 (Ottawa) vol A (Amsterdam: Elsevier Science Publishers) pp 15-30
[3] Lahtinen J, Valkonen J, Björkman K, Friis J, Niemela I and Heljanko K 2012 Reliab. Eng. Syst. Safe. 105 104
[4] Ji M 2014 Research on the Model Checking Method for Complex Stochastic Systems (Harbin: Harbin University of Engineering)
[5] Kwiatkowska M, Norman G and Parker D 2011 Proc. of CAV 2011 (Snowbird) vol 6806 (Berlin: Springer) pp 585-591
[6] Feng C, Zhang H, Yan S, Fu Y and Bao X 2017 Proc. of ICRSE(Beijing) vol 1 (Piscataway, N J: IEEE) pp 1-6
[7] Guo X, Yang Z 2016 Proc. of ICSESS(Beijing) vol 1 (Piscataway, N J: IEEE) pp 564-567
[8] Sesić A, Dautovic S and Malbasa V 2008 IEEE.T.Comput.Aid.D. 27 403
[9] Haverkort B R, Hermanns H and Katoen J P 2000 Proc. of SRDS’00(Erlangen) vol 1 (Piscataway, N J: IEEE) p 228
[10] Workstation Cluster on http://www.prismmodelchecker.org/casestudies/cluster.php
[11] Liu Y, Li X, Ma Y and Wang L 2015 Chinese Journal of Computers 11 2145
[12] Zhou C 2014 Theory and Application of Random Model Checking(Beijing: Science Press)
[13] Gu Y, Zhu F and Tang S 2012 Journal of Jiangxi University of Science and Technology 33 51