An Associative Optimizing Method on Reliability and Cost in Clouds

SA MENG, YUANSHUN DAI, LIANG LUO, AND JIE LUO

1School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
2State Key Laboratory of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Corresponding author: Liang Luo (luoliang@uestc.edu.cn)

This work was supported in part by the Opening Foundation of SKLSDE under Grant SKLSDE-2019KF.

ABSTRACT Cloud computing enables users to use shared system resources in an efficient pay-as-you-go manner. However, the reliability of cloud computing is a challenging proposition. Cloud users cannot improve the reliability of their cloud environment from the hardware and system level. On the other hand, due to the cost and complexity of the system, improving the reliability of specific users using practical methods is not straightforward for the operators of cloud systems. In this paper, we attempt to optimize the reliability of the cloud system by using the correlation between cost and reliability. We conduct a comprehensive and detailed analysis of the reliability of clouds, extract vital features that can be used to improve system reliability and consider cost constraints. We propose the reliability model and virtual machine (VM) provisioning/request model; based on this model, we propose an SLA-oriented reliability assurance scheme to continuously optimize the reliability of clouds. The experimental results not only indicate the validity of the model and schemes proposed in this paper but also provide a new perspective for the continuous optimization of cloud system reliability.

INDEX TERMS Reliability, cloud computing, system optimization, SLA.

I. INTRODUCTION

The significant growth in cloud applications has caused a boom in cloud infrastructures. Each cloud infrastructure hosts computing and storage services with different requirements. Although the expansion of infrastructure resources has created reliability and security problems, identifying an effective way to increase the system reliability with a limited budget is challenging [1], [2].

The reliability of cloud services is defined in terms of service availability, failure of resources, or failure of services. In clouds, tasks submitted by users usually include requests for hardware, platform, and software services, and various tasks have different requirements regarding the performance, execution time, and reliability of cloud infrastructures. Typically, a service provider applies to a cloud platform for more than one VM to provide users with multiple application services, which utilize messaging among multiple VMs [3], [4]. These VMs often exchange data and occupy a lot of network load [5]. Research predicts that the global data center network traffic will reach 19.5 ZB in 2021 [6], [7]. A huge amount of data exchanges not only occupy the bandwidth of cloud infrastructure of physical infrastructure [8], [9] but also decrease the reliability of the entire cloud system [10]. According to Ponemon Institutes’ survey in 2016 [11], by long-term observations of 63 data centers, the total cost per minute or an unplanned outage in 2016 increased by 55.6% in 2010. According to Information Weekly, the annual loss of revenue from an IT service disruption exceeds $26.5 billion. A Service Level Agreement (SLA) is a clear expression of the rights and obligations between cloud service providers and users based on the Quality of Service (QoS). When SLA Violations occur, cloud service providers will have to face financial compensation, but even worse, the poor user experience will spread among social networks [12]. Rackspace, which is a well-known cloud service provider, receives a maximum fine of 30% of its generated consumer bills due to a slight drop in service reliability within a month.

To improve the reliability of the cloud system, numerous efforts have been made in past decades. Currently, most cloud systems improve system reliability via the redundancy of
hardware, software, data storage, and application services. In [13]–[16], variants of schemes based on the placement and migrations of VMs were proposed to migrate active VMs to highly reliable hardware. However, satisfying the real-time requirements of cloud services with these strategies is challenging because the overall migration of VMs often involves user data. Two possible scenarios exist: If a user’s data are also stored in the VM, the migration is often time-consuming and consumes a substantial amount of network bandwidth [17]. If the user’s data are stored outside the VM, the migration requires additional mechanisms to ensure the consistency between the data in the VM and the data in storage devices [18], [19]. Deploying redundant hardware is another common strategy to improve reliability. These redundancies include the whole or components of computing, storage, and network devices [20], [21].

A situation that cannot be disregarded is that redundancy will generate extra cost. In a typical computer system, two components with a reliability of 0.9 can achieve a reliability of 0.99 via a hot standby. However, these options are employed at the cost of overhead. Therefore, when considering improving the reliability of clouds, the cost is a factor that cannot be disregarded. However, the reliability and system costs often show a relationship of mutual restriction. In [19], [20], some existing reliability mechanisms can guarantee the reliability of cloud services to 99%. These methods require additional backup and storage hardware resources to store system and app logs and checkpoints to restore the system state caused by failures or interruptions. Increasing hardware redundancy is a direct way to improve system reliability but it creates a substantial cost burden to service operators. The increased hardware not only needs maintenance but also increases the energy consumption [22]. The severe energy consumption problem of the cloud has attracted extensive attention.

Due to the large number of users and complex structure of a cloud computing system, the work of reliability assurance produces many challenges: The reliability requirements of such an extensive range of users differs, and the existing SLA cannot cover the actual reliability needs of users. On the other hand, due to cost constraints, service providers, or cloud data center operators cannot improve the reliability of each user via redundancy. Therefore, obtaining cost-effective solutions to improve the reliability of cloud systems has become an urgent problem for both academia and industry.

Due to the existing situation, this paper proposes two valid initial reliability models for VMs, servers, and relevant models of reliability, energy consumption, and cost. To provide a reliability-aware energy efficiency cloud service, a set of optimization strategies for reliability assurance and efficiency enhancement, including an SLA-oriented reliability assurance scheme and physical machine decision-making algorithm, are proposed. According to these proposed models and algorithms, detailed simulation experiments are performed, and the real cloud’s operation data are introduced to verify the effectiveness of the model and algorithm. The experimental results are verified and analyzed in detail.

The remainder of this paper is structured as follows. The next section describes the system and related models. Section 3 is devoted to the analysis and construction of SLA-oriented reliability assurance schemes and cost evaluations. Section 4 gives specific example solutions and investigates the effect of different model parameters on the reliability, survivability and cost of a cloud system. Section 5 concludes the paper and outlines directions for future work.

II. SYSTEM DESCRIPTION AND RELIABILITY MODELS

A. SYSTEM DESCRIPTION OF CLOUD WITH RELIABILITY CONCERN

For a running cloud, with an increase in running time, the reliability of the whole system is slowly decreased. This phenomenon is related to the aging of VM components, the continuous state change in task scheduling, resource allocation, and network topology of the virtual environment, etc. Cloud systems often encounter various failures, such as queue overflow, request timeout, loss of data sources, software failure, database inaccessibility, hardware failures and network failures [21]. As shown in Figure 1, the infrastructure of a cloud system is often composed of multiple distributed data centers with similar structures.

**FIGURE 1. Multi-tier cloud infrastructure architecture.**

Similar to standard computers, cloud systems use redundancy and backup to improve the system reliability. To ensure the practicability of the models provided in this paper, we adopt the extensively employed backup methods:

**Cold Spare (CSP):** The cold standby node acts as the backup of the primary node, which will be active and configured only when the primary node fails for the first time. Subsequently, when the primary node fails, the standby node is started, and the data are restored before the failed component causes a system failure. The data of the primary node can be backed up on the storage system and restored on the standby node when needed. This action usually provides several hours of recovery time.
Warm Spare (WSP): Install application, service, and software components and their running environment, which is consistent with the primary node on the standby node. The warm standby node is sleeping. When the primary node fails, the warm standby node enters the running state, and related applications, services, and the software will start on the warm standby node. This process is usually automated using cluster managers. Data and warm backup nodes are synchronized periodically using disk-based replication or a shared disk. This action usually provides a few minutes of recovery time.

Hot Spare (HSP): Install applications, services, and software and their running environments, which are consistent with the primary node in hot standby mode. Hot standby nodes and related applications, services, and software components are running but do not process data or requests. Data almost comprise real-time synchronization, and both systems will have the same data. Data replication is usually accomplished via software functions. This process usually provides a few seconds of recovery time. Software components are running but they do not process data or requests.

Enhanced redundancy and backup strategies require a substantial amount of system operation, which increases the cost. Different redundancy backup strategies have different levels of reliability, energy consumption, and cost. In clouds, a cold backup means that cold standby nodes exist as VM images rather than active VM instances. Data between primary nodes and standby nodes are synchronized periodically according to the specified schedule. Since the CSP is not running and does not bear any workload, its failure rate is considered to be 0, that is, its reliability is 1. CSP has the smallest demand for energy consumption; thus, the corresponding cost is also considered to be the smallest. Hot backup and warm standby nodes are VM instances, which means that warm standby and hot standby nodes must be installed and deployed and must be available immediately when the primary node fails. For the warm backup node, most of the time is in the sleep state. We consider that this node has high reliability, moderate energy consumption demand, and moderate energy consumption cost. A hot standby node is deployed and run with the primary node, although it usually does not require any workload to process user requests. To ensure fault tolerance, the critical data of the hot standby node are mirrored almost in real-time from the primary VM instance (for example, in the 200 microseconds range). Thus, we can conclude that the reliability of the hot standby node is lower than that of the warm standby node, and its energy consumption and corresponding energy consumption cost are higher than that of the warm standby node. Generally, the reliability of the cold standby node, warm standby node, hot standby node, and primary node is decreasing, and their energy consumption and corresponding energy consumption cost are increasing.

B. RELIABILITY MODELS FOR CLOUDS

The majority of existing research uses Markov models, analytic models, and fault tree models to study the reliability of distributed systems. For practical reasons, we will use fault tree models to describe the reliability performances of physical servers and VMs in clouds. To make the model easy to understand, we make the following assumptions:

1) Let the set of VMs in the cloud system consist of all VMs (referred to as “VM system”) and the failure parameters of each VM obey an exponential distribution [22] and the failure of different VMs are relatively independent and non-recoverable. The failure rate of all VMs is constant.

2) Let all physical machines in the cloud system (referred to as “physical machine system”) consist of a set of physical machines. The failure of each physical machine obeys a Poisson distribution [22], and the failure of each PM is relatively independent and non-recoverable. The failure of a PM will cause the VM on it to fail.

The Poisson process is usually used to describe the random process of the number of requests that arrive in the service system. In this paper, we also use the Poisson process to describe the number of VM requests that arrive in the cloud and construct the VM Provisioning Request Model (VMPRM) in the cloud.

Assuming that the class l VM supply request satisfies the Poisson process with the strength λl, the VMs are independent of each other and the exponential distribution of the running time service of the VMs in the data center. If the expected running time is 1/µl, the expected number of class l VMs in the steady-state system is λl/µl.

VMs are divided into two types: Computing Server (CS) and Storage Server (SS). Assuming that the supply requests of CSs and SSs satisfy the Poisson processes of λcs and λss, respectively, the expected running time is 1/µcs and 1/µss, respectively, and the total number of VMs that are contained in the system in a steady state can be expressed as follows:

\[ m = \frac{\lambda_{cs}}{\mu_{cs}} + \frac{\lambda_{ss}}{\mu_{ss}} \]  

When a cloud does not have a redundancy mechanism, and if a VM fails, the cloud cannot provide services. This situation is referred to as VM failure. In the previous figure, VM failure is regarded as the top event of the fault tree. Any VM failure can produce a top event when no redundancy exists. Since the failure event of VM vi obeys an exponential distribution with the parameter λvi, the probability of the top event occurrence can be expressed as follows:

\[ \Pr(VM\text{Failure}) = 1 - \prod_{i=1}^{m} (1 - \Pr(v_{i,\text{failure}})) \]

\[ = 1 - \prod_{i=1}^{m} (1 - \int_{0}^{t} \lambda_{i}e^{-\lambda_{i}t} dt_{i}) \]

\[ = 1 - \prod_{i=1}^{m} e^{-\lambda_{i}t} \]
Thus, the VM reliability function \( R(t) \) of the cloud system can be expressed as follows:

\[
R(t) = \prod_{i=1}^{m} e^{-\lambda_i t}
\]  

(3)

We discuss the reliability of VMs with a hot backup mechanism. Assume that the failure rates of the main component M and hot spare part N are \( \lambda_M \) and \( \lambda_N \), respectively, if M works normally, N does not handle the task, which means that \( \lambda_N \) is normally less than \( \lambda_M \). When M fails, N assumes all of M’s task load and becomes the main component, which is denoted as \( N^* \). However, because the configuration of M and N may not be possible, \( \lambda_M \) and \( \lambda_N \) may not be equal.

As shown in figure 2, the state transition of VMs can be regarded as a Markov chain with HSP gates, which includes four states: MN, M, \( N^* \), and Failure. The state MN indicates that both the main component and the hot spare parts are in normal operation; the state M means that the hot spare parts fail before the main component; the state \( N^* \) shows that the main component fails before the hot spare parts; and the state Failure indicates that both the main component and the hot spare parts fail simultaneously, and the HSP gate fails. Note that all the state transitions are real-time.

![FIGURE 2. State transition of VMs.](image)

Let \( P_i(t) \) denote the probability that the cloud is in state \( i \) at time \( t \), and \( P_i(dt) = P[X(t + dt) = j|X(t) = i] \) denote the transition probability of the random variable \( X(t) \). \( [P_{ij}(dt)] \) represents the one-step transition matrix of the Markov chain. Given the initial probability, the transition matrix can describe the whole state transition process.

\[
[P_{ij}(dt)] = \begin{bmatrix}
1 - (\lambda_M + \lambda_N)dt & \lambda_N dt & \lambda_M dt & 0 \\
0 & 1 - \lambda_M dt & 0 & \lambda_N dt \\
0 & 0 & 1 - \lambda_N dt & \lambda_N dt \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

We note that

\[
P_1(t) = e^{-\lambda_M t}
\]

\[
P_2(t) = e^{-\lambda_M t} - e^{-(\lambda_M + \lambda_N)t}
\]

\[
P_3(t) = \frac{\lambda_M}{\lambda_M + \lambda_N - \lambda_N^*} (e^{-(\lambda_N^* - \lambda_N)t} - e^{-(\lambda_M + \lambda_N)t})
\]

(4)

In state MN, M, and \( N^* \), the cloud system is running normally. The reliability function \( R(t) \) is expressed as follows:

\[
R(t) = P_1(t) + P_2(t) + P_3(t)
\]

\[
= e^{-\lambda_M t} + \frac{\lambda_M}{\lambda_M + \lambda_N - \lambda_N^*} (e^{-(\lambda_N^* - \lambda_N)t} - e^{-(\lambda_M + \lambda_N)t})
\]

(5)

By using the HSP strategy, the main component of each VM and its hot spare parts act as an input of the HSP gate. The probabilities of the top events is determined by the output of the multiple HSP gates as the input of the gates. The output of the HSP gates can be expressed as \( S_1, S_2,\ldots, S_m \), and their probabilities of occurrence are \( U_{S_1}, U_{S_2},\ldots, U_{S_m} \), respectively. The probability of VM failure is as follows:

\[
Pr(VM \text{ Failure}) = 1 - \prod_{i=1}^{m} (1 - U_{S_i})
\]

\[
= 1 - \prod_{i=1}^{m} \frac{\lambda_i}{\lambda_i + \lambda_i^*} (e^{-(\lambda_i + \lambda_i^*)t} - e^{-(\lambda_i^*)t})
\]

(6)

From these formulas, the reliability of VM decreases with an increase in time. Adopting the hot spare strategy can enhance the reliability of VMs.

### III. SLA-ORIENTED RELIABILITY ASSURANCE SCHEME AND COST EVALUATIONS

This section discusses the SLA-Oriented Reliability Assurance Scheme (SLA-ORAS). According to the SLA, IaaS service providers and users/agents agree regarding service reliability thresholds. The service provided by IaaS service providers should always be higher than the reliability threshold. In cloud systems, VMs directly address users’ workloads. To ensure that the reliability of service is higher than the reliability threshold, that is, the reliability of VMs should be higher than the threshold. Because the reliability of VMs will decrease with time, the reliability of the VM system should be periodically restored to an established reliability threshold. The figure 3 shows the process of the SLA-oriented reliability assurance scheme.

To adapt to the task of some reliability-sensitive users, IaaS service providers built a VM system with a full hot spare backup strategy. However, with an increase in running time, the reliability of the cloud system will slowly decrease. For reliability-sensitive users, the cloud system needs a method that can dynamically restore the reliability of VMs. Therefore, the VM reliability model, which is based on the HSP Strategy and CSP strategy, is needed.
A threshold is established for the VM reliability of the cloud system. When the reliability of VMs reaches the threshold, cloud systems need to take action to improve the system reliability. To better model and analyze the reliability of VMs, the running state of VMs can be divided into two stages: in stage 1, the VMs are reliable enough, that is, the reliability of the VMs is higher than the threshold. This stage only focuses on the reliability of the main components and hot spare parts of the cloud without considering the cold spare parts. The cold spare parts cannot take over the task load of the system immediately after the main components and hot spare parts fail. In Phase 2, with the long-running of VMs in clouds, the reliability decreases to the threshold, and the cold spare parts fail. In Phase 2, the VM system immediately after the main components and hot spare parts fail. In Phase 2, with the long-running of VMs the system immediately after the main components and hot spare parts fail. In Phase 2, the VM system immediately after the main components and hot spare parts fail. In Phase 2, with the long-running of VMs in clouds, the reliability decreases to the threshold, and the system enters this phase to restore the system reliability above the threshold.

Set the VM to VM = [v1, v2, ..., vm], and set the hot spare VM to VM∗ = [v′1, v′2, ..., v′m]. Configure the cold spare VMs VM∗ = [v′1, v′2, ..., v′m] and VM∗∗ = [v′′1, v′′2, ..., v′′m] for VM and VM∗ respectively. A cold spare VM exists in the form of a VM image, which occupies only a part of the disk space and does not occupy resources such as processors and memory. The fault tree modeling process of Phase 1 and Phase 2 is discussed.

Phase 1: The components involved in the reliability modeling in this phase are A and B. The state of the VM system is the same as that for the HSP redundancy strategy.

Phase 2: With an increase in service time, the reliability of the cloud system decreases gradually. When the reliability of VMs drops below the threshold, the cloud system enters the second stage. The cold spare parts start to run and take on the processing of the workload of the VM system. When the workload of VM is transferred to VM∗, VM∗ and VM∗∗ completely replace VM and VM∗, respectively, for task processing. VM and VM∗ will be destroyed, and resources will be released. During this period, the cloud system is in a high-reliability state. When the replacement is completed, the reliability of the system is equivalent to the initial state with the HSP redundancy strategy. Periodically putting the VM system in phase 1 and phase 2 can position the VM system above the set reliability threshold. From the overall view of the system, the reduced reliability of the VM system has been restored.

As shown in figure 4, each VM master component of the cloud system acts as an input to an HSP gate with its hot spare parts. Simultaneously, the corresponding cold spare parts will be activated, which is also an input of an HSP gate. Concurrently, they act as two inputs of the AND gate. The output of the M AND gate is the input of the OR gate. The output of the OR gate is the probability of the top event, or the output of the gate is the probability of the top event, that is, the failure probability of the cloud system. We express the output of the gate as S1, S2, ..., Sm and the probability of occurrence is U_S1, U_S2, ..., U_Sm. The probability of the top event occurrence is

$$Pr(VM\ Failure) = 1 - \prod_{i=1}^{m} (1 - U_{S_i})$$ (7)

Consider $S_i$, for example. The output of the two HSP gates are $Q_1$ and $Q_2$ respectively. The output $S_i$ is obtained from the fault tree and gate properties when $Q_1$ and $Q_2$ occur simultaneously. Thus,

$$U_{S_i}(t) = Q_1(t) \cdot Q_2(t)$$

$$= \left[ 1 - e^{-l_1 t} + \frac{l_1}{l_1 + l_2} (e^{-l_1 t} - e^{-l_2 t}) \right] \cdot \left[ 1 - e^{-l_1 t} + \frac{l_2}{l_2 + l_3} (e^{-l_1 t} - e^{-l_3 t}) \right]$$ (8)

In the process of transferring the workload of the VM system to VM∗∗, the reliability function R(t) of the VM system is expressed as follows:

$$R(t) = 1 - Pr(VMs\ Failure) = \prod_{i=1}^{m} (1 - U_{S_i})$$ (9)

After Phase 2, the VMs are in a high-reliability state. When the replacement is completed, the reliability of the system is equivalent to the initial state with the HSP redundancy strategy, and the VM system is restored to Phase 1.

IV. EXPERIMENT

Because the cloud computing system is extensive and has many users, experimenting with the actual cloud system is
difficult, and the limited laboratory software and hardware environment hinders presenting the characteristics of the actual cloud system. In this paper, we use the CloudSim [12] cloud system simulation framework. We flexibly configure a variety of data centers, hosts, VMs, and task types. To make task requests and data center configuration more realistic and rigorous, we employ Google Trace, which contains the running logs of an actual cloud computing environment, as the experimental data set.

To validate the model and method proposed in this paper, CloudSim needs to be extended in two aspects: the first aspect is to simulate various host failures, VM failure, and other failure events and add simulation implementation classes, such as exponential distribution and Poisson distribution. The second aspect is to integrate Google Trace into CloudSim, add Classes to process Google Trace data and re-implement classes that simulate the hosts and tasks of a Google cluster.

Google Trace data contains six sets of data, including physical machine resources, physical machine events, task events, task resource requests, and task constraints, etc. The total data size is approximately 41 GB. Three main categories exist:

*Machines:* Contains physical machine attribute data and event data. According to the different attributes of the core and clock frequency of the physical machine, the attributes of the physical machine in the cluster are recorded. Physical machine event data record the status information of the physical machine in the cluster at different times, which is added, removed, and updated in the cluster.

*Jobs:* Records different workloads at different times in the cluster, which consists of multiple tasks. For example, Map/Reduce consists of Map job and Reduce job. Map job and Reduce job are composed of several tasks, respectively.

*Tasks:* Record the specific tasks performed at different times in the cluster. This category involves three kinds of data: task events, task resource requests, and task constraints.

In the experiments, different kinds of physical hosts are simulated in CloudSim according to the attribute data of Machines, and the resource usage of physical hosts at different times is simulated according to the event data of Machines, as well as adding and removing events in the data center at different times. Additional events of physical hosts, which can be considered physical hosts, have been repaired and can be reused to join the cluster. Removal events can be considered as hardware failure of physical hosts and cause physical hosts to be removed from the cluster. According to the Jobs and Tasks data, simulation task requests reach the data center as resource use requests. Large-scale task requests are imported into the CloudSim to carry out simulation experiments on reliability and energy consumption, which can more truly reflect the real situation, verify the correctness of reliability and energy consumption models, and ultimately guide the optimization of the reliability and efficiency of large-scale cloud systems.

### A. Analysis of Reliability Assurance Scheme for SLA

To provide an SLA-oriented reliability assurance scheme, ensuring that the reliability of the cloud system is above the agreed reliability threshold in the SLA is necessary.

As mentioned in section 3, reliability assurance consists of two stages: in stage 1, the cloud system is reliable enough, that is, the system reliability is higher than the preset threshold. This stage only focuses on the reliability of the main components and hot spare parts in the system, without considering the cold spare parts that have not been started. In Stage 2, with the long-running of VM system, the reliability of the system decreases to the preset threshold. By opening the cold spare parts, the system can enter this stage to restore the system reliability above the threshold.

Assume that the cloud system $VM = \{v_1, v_2, \ldots, v_m\}$ has 100 VMs. We consider the same failure rate and different failure rate, respectively, of VM components. First, considering the same situation, assuming that the failure rate of the main component VM $v_i$ is $\lambda_i = 0.0004$ and the failure rate $\lambda'_i = 0.00025$ is the hot standby component VM $v'_i$, the reliability function of the system can be obtained by the following:

$$R(t) = \left(e^{-0.0004t} - 1.6 \left(e^{-0.00065t} - e^{-0.0004t}\right)\right)^{100}$$

The reliability of the VM system periodically decreases the reliability threshold (99%) every 27 days. By starting the cold spare parts, the VM system is in a high-reliability state. Assume that approximately 1 hour, or 0.04 days, is needed to start the cold spare parts and synchronize the data of the main components and the hot spare components with the start-up cold spare parts. During this period, the reliability of the VM system is close to 100%. When the data synchronization is finished, the original main components and thermal insulation spare components will be destroyed, and the cold spare parts will turn into new main components and hot spare components. In this cycle, the VM system is periodically restored every 27 days. Its reliability curve is shown in figure 5.

**FIGURE 5.** Periodic reliability recovery diagram of isomorphic VMs.

We also consider the different failure rates of VMs in cloud systems. Assume that two kinds of VMs exist in cloud
systems: Computing Server VM (CS) and Storage Server VM (SS). The failure rate of the CS’s main component VM is 0.0004 and the hot standby component VM is 0.00025. The reliability function of the system can be obtained by formula 9 as follows:

\[
R(t) = \left[ \frac{e^{-0.0004t} - 1.6 \cdot (e^{-0.00065t} - e^{-0.0004t})}{50} \cdot \left[ e^{-0.0005t} - 1.6 \cdot (e^{-0.00008t} - e^{-0.00005t}) \right]^{50} \right]
\]

In a follow-up experiment, there are two types of VMs: CS and SS. The period of stage 1 is 23 days. The reliability of the VM system will be lower than the threshold (99%) from the 24th day. Thus, approximately one hour is needed to complete the data synchronization in Phase 2 and the destruction of the original hot spare parts. The process is then cyclized periodically, as shown in figure 6 and figure 7.

For the local reliability recovery of VM, the relationship between system reliability and time is shown in Figure 6. The local reliability recovery process will be performed 8 times in 115 days, while the global reliability recovery process will be performed 10 times. Assuming that the cost of each reliability recovery process is the same, the local reliability recovery method can reduce the cost by 20% compared with the global recovery. The failure rate of different kinds of VMs is the fundamental reason for the difference in the local reliability. In reality, the types of VMs can also be subdivided, and the use of local reliability recovery will yield more significant cost savings.

B. SIMULATIONS AND EVALUATIONS OF COST-EFFECTIVE RELIABILITY OPTIMIZATION METHODS

To simulate and analyze the cost-effective reliability optimization methods, considering how to reliably distribute VMs to physical machines and simultaneously achieve an energy-saving effect is necessary. This approach involves the reliability and energy consumption of VM systems and physical machine systems. Based on the energy consumption model obtained in the previous section, we employ the first Fit (FF) to evaluate the RERA algorithm.

In the RERA algorithm, we seek a physical machine that satisfies both reliability constraints as the VM placement object. When the reliability constraints are satisfied, the common cause failure caused by the failure of some physical unit combination will not affect the regular service provided to users. In the FF algorithm, we do not consider the reliability constraints and minimum energy consumption optimization of VM placement. We use the energy consumption of the two systems and the number of failures that cause the system to provide services to users as evaluation indicators, that is, better energy efficiency and fewer faults are the goals of the algorithm.

In Google Trace’s Task Events data, each record describes the agent from which the task originated. To evaluate the RERA algorithm, ten agents were selected in the experiment, and only one agent was tested at a time. Establish a data center with 500 available hosts. For the configuration of the data center, the resource configuration of the host is the maximum. According to the data of Machine Events of Google Trace, the state of the physical machine is defined. When an additional event arrives, it is equivalent to “booting up” an available host and joining the cluster. The host configuration is simultaneously adjusted according to the available resources described by the arriving host.

The simulated cloud system consists of 2000 available VMs, which are used as the main spare parts of VMs and hot spare parts of VMs. The cold spare VM will only be active when the reliability recovery reaches stage 2, which shows that the idle, available VM will be converted to a working state. The main spare parts and hot spare parts of VMs that are destroyed are re-converted to an idle VM. The reliability of the idle VM is 100%; only the failure distribution function of the working VM is calculated. Establishing the failure rate of the VM simulates a VM system with a redundant backup strategy.

In the simulated clouds, only one agent’s task is handled at the same time. We set the simulation time to 864,000 seconds or ten days. In this process, all tasks of the same agent in the Task Events data are processed. The experimental results are shown in table 1. In the table, power represents the average power of the whole simulation process, and the energy consumption represents the total energy consumption of the whole simulation process.
The analysis of the experimental data reveals that the average power required by the RERA algorithm is much lower than that of the FF algorithm in performing the task of the same agent in Task Events data.

V. CONCLUSION

The continuous reliability optimization of clouds is a challenging issue. Theoretical reliability models are often used to evaluate the system reliability of clouds. However, existing reliability models for clouds is limited to its applicability. Many users are reluctant to pay more to improve the reliability of the cloud environment they use. On the other side, it is also difficult for cloud operators to degrade reliability for specific user due to the cost and complexity of the system. This paper attempts to optimize the reliability of the clouds by balancing the correlation of cost and reliability. We built a reliability model of the VM system and a reliability aware cost model by using fault tree analysis methods. We also proposed a set of adjustment strategies that are integrated into the SLA-ORAM for optimizing reliability under the constraint of cost. We also carried out several simulation experiments using Google Trace data. The experimental results not only verify the effectiveness of the proposed model and algorithm but also provide practical strategies for improving the reliability of the cloud system.

REFERENCES

[1] R. Muñoz, R. Vilalta, N. Yoshikane, R. Casellas, R. Martínez, T. Tsuritani, and I. Morita, “Integration of IoT, transport SDN, and edge/cloud computing for dynamic distribution of IoT analytics and efficient use of network resources,” J. Lightw. Technol., vol. 36, no. 7, pp. 1420–1428, Apr. 1, 2018.

[2] Y. Zhu, D. Huang, C.-J. Hu, and X. Wang, “From RBAC to ABAC: Constructing flexible data access control for cloud storage services,” IEEE Trans. Services Comput., vol. 8, no. 4, pp. 601–616, Jul./Aug. 2015.

[3] X. Xu and H. H. Huang, “Exploring data-level error tolerance in high-performance solid-state drives,” IEEE Trans. Rel., vol. 64, no. 1, pp. 15–30, Mar. 2015.

[4] A. Vishwanath, F. Jalali, K. Hinton, T. Alpcan, R. W. A. Ayre, and R. S. Tucker, “Energy consumption comparison of interactive cloud-based and local applications,” IEEE J. Sel. Areas Commun., vol. 33, no. 4, pp. 616–626, Apr. 2015.

[5] M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, “Managing data transfers in computer clusters with orchestra,” ACM SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. 98–109, Oct. 2011.

[6] S. U. R. Malik, S. U. Khan, and S. K. Srivivasan, “Modeling and analysis of State-of-the-art VM-based cloud management platforms,” IEEE Trans. Cloud Comput., vol. 1, no. 1, Jan. 2013.

[7] J. Lloret, M. Garcia, J. Tomas, and J. J. P. C. Rodrigues, “Architecture and protocol for intercloud communication,” Inf. Sci., vol. 258, pp. 434–451, Feb. 2014.

[8] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, “To offload or not to offload? The bandwidth and energy costs of mobile cloud computing,” in Proc. IEEE INFOCOM, Turin, Italy, Apr. 2013, pp. 1285–1293.

[9] X. Fu, J. Chen, S. Deng, J. Wang, and L. Zhang, “Layered virtual machine migration algorithm for network resource balancing in cloud computing,” Frontiers Comput. Sci., vol. 12, no. 1, pp. 75–85, Feb. 2018.

[10] J. Zhao, Y. Xiang, T. Lan, H. H. Huang, and S. Subramaniam, “Elastic reliability optimization through Peer-to-Peer checkpointing in cloud computing,” IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 2, pp. 491–502, Feb. 2017.

[11] Ponemon Institute and Emerson Network Power. 2016 Cost of Data Center Outages. Accessed: Jan. 19, 2016. [Online]. Available: https://www.ponemon.org/blog/2016-cost-of-data-center-outages

[12] R. Buyya, K. Ramamohanarao, C. Leckie, R. N. Calheiros, A. V. Dastjerdi, and S. Versteeg, “Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions,” in Proc. IEEE 21st Int. Conf. Parallel Distrib. Syst. (ICPADS), Melbourne, VIC, Australia, Dec. 2015, pp. 75–84.

[13] A. N. Tantawi, “A scalable algorithm for placement of virtual clusters in large data centers,” in Proc. IEEE 20th Int. Symp. Modeling, Anal. Simulation Comput. Telecommun. Syst., Washington, DC, USA, Aug. 2012, pp. 3–10.

[14] N. Trung Hieu, M. Di Francesco, and A. J. Jässki, “A virtual machine placement algorithm for balanced resource utilization in cloud data centers,” in Proc. IEEE 7th Int. Conf. Cloud Comput., Anchorage, AK, USA, Jul. 2014, pp. 474–481.

[15] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, “An energy efficient ant colony system for virtual machine placement in cloud computing,” IEEE Trans. Evol. Comput., vol. 22, no. 1, pp. 113–128, Feb. 2018.

[16] M. Block, R. Emery, K. Lank, and J. Ryan, “Implant placement accuracy using dynamic navigation,” Int. J. Oral Maxillofacial Implants, vol. 32, no. 1, pp. 92–99, Jan. 2017.

[17] R. A. M. Razali, R. A. Rahman, N. Zaini, and M. Samad, “Virtual machine migration implementation in load balancing for cloud computing,” in Proc. 5th Int. Conf. Intell. Adv. Syst. (ICIAS), Kuala Lumpur, Malaysia, Jun. 2014, pp. 1–4.

[18] A. Sallam and K. Li, “A multi-objective virtual machine migration policy in cloud systems,” Comput. J., vol. 57, no. 2, pp. 195–204, Feb. 2014.

[19] M. Arif, A. K. Kiani, and J. Qadir, “Machine learning based optimized live virtual machine migration over WAN links,” Telecommun. Syst., vol. 64, no. 2, pp. 245–257, Feb. 2017.

[20] E. Sousa, F. Lins, E. Tavares, and P. Maciel, “Cloud infrastructure planning and local applications,” J. Lightw. Technol., vol. 36, no. 7, pp. 1420–1428, Apr. 1, 2018.
YUANSHUN DAI (Senior Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2000, and the Ph.D. degree from the National University of Singapore, Singapore, in 2003. His current research interests include reliability modeling and optimization, and intelligent self-improvement in large-scale computing systems. He is a Chapter Chair of the Reliability Society.

LIANG LUO (Member, IEEE) received the B.Sc. degree in computer science from Beihang University, China, in 2004, the M.Sc. degree in software engineering from the Beijing University of Technology, China, in 2006, and the Ph.D. degree in computer science from Beihang University, in 2014. He is currently an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. His research interests include cloud computing, system reliability, energy efficiency, and optimization. He is a member of the IEEE Reliability Society.

JIE LUO (Member, IEEE) received the Ph.D. degree in computer science from the School of Computer Science and Engineering, Beihang University, China, in 2012. He is an Associate Professor with the School of Computer Science and Engineering, Beihang University. His research interests are knowledge graph construction and inference.