

# State-Control-Limit-Based Rejuvenation Modeling of Virtualized Cloud Server

Weichao Dang<sup>a,b,\*</sup> and Jianchao Zeng<sup>b,c</sup>

<sup>a</sup>*College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, China*

<sup>b</sup>*Division of Industrial and System Engineering, Taiyuan University of Science and Technology, Taiyuan, 030024, China*

<sup>c</sup>*School of Computer Science and Control Engineering, North University of China, Taiyuan, 030051, China*

---

## Abstract

Software rejuvenation modeling of the virtualized Cloud Server has been studied. A software rejuvenation policy on the virtual machines and the virtual machine monitor has been proposed in order to ensure high availability of the virtualized Cloud Server. The multi-component system, composed of the virtual machines and the virtual machine monitor, which are structurally dependent, has been reduced to the multiple two-component systems. The state-control-limit-based rejuvenation policy has been proposed and the stationary probability density of the two component system state has been derived. Furthermore, the stationary unavailability of the virtualized Cloud Server has been modeled. Numerical experiments have verified the correctness of the probability density function and the feasibility of the rejuvenation policy. The state-control-limit-based rejuvenation policy leads to lower unavailability of the virtualized Cloud Server in comparison with the lifetime-based rejuvenation policy.

**Keywords:** software rejuvenation; state-control-limit; virtualized Cloud Server; unavailability

(Submitted on November 15, 2017; Revised on January 21, 2018; Accepted on February 19, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

---

## 1. Introduction

Server virtualization becomes an essential software component for system infrastructure of various software services. Many enterprises introduce server virtualization to consolidate servers for enterprise applications and reduce the total cost of ownership. In recent years, server virtualization is also used for software infrastructure of cloud computing services, which provide computing resources over the Internet. By using a virtual machine (VM), the cloud service customer creates computing resources quickly on the cloud computing infrastructure. With the popularity of the server virtualization, availability management of the virtualized server system grows in importance [16,17,20].

Virtualization runs a software layer between the hardware and the operating system, known as a virtual machine monitor (VMM) or hypervisor, to virtualize hardware resources in physical servers and generate multiple execution environments called virtual machines (VMs). Inside each VM, the operating system (often called the guest OS) and applications run on virtual resources of the VM, such as virtual CPU, virtual network card, virtual RAM, and virtual disks. For long running software, such as OS and VMM, the aging-related bug is one of the major causes of software failures. Difficult to be removed completely in development and testing phases, aging-related bugs accumulate errors during software execution and may eventually cause unacceptable performance degradations or failures due to resource depletion [10]. The phenomena of software aging are observed in a wide variety of software, such as Unix OS, Linux, telecommunication system, Apache web server, Apache Axis web application server, and Xen [2,5,8,9,13].

As one of the promising techniques for assuring high-availability of the virtualized server system, software rejuvenation is a proactive recovery method that clears aging status by restarting or resetting the software execution environment [11,15]. To postpone or prevent system failures caused by software aging in VMM, software rejuvenation can be applied. Different

---

\* Corresponding author.

E-mail address: [dangweichao@tyust.edu.cn](mailto:dangweichao@tyust.edu.cn)

from software rejuvenation used for the mono-component, software rejuvenation applied to VMM involves additional cost due to the downtime of the hosted VMs running on the VMM. Therefore, such downtime needs to be taken into account in determining an appropriate rejuvenation schedule. It is necessary to study the rejuvenation policy on the virtualized server system with the VMM-VMs two-tier software systems.

Different from the above research, the present study uses the continuous state degradation model to describe the performance degradation process of VM and VMM accurately. Moreover, different from their consideration of the Migrate-VM rejuvenation, we focus on the rejuvenation policy on a single virtualized server system.

To design an effective rejuvenation policy on a virtualized server system, the following issues need to be considered:

- There are multiple VMs in a virtualized server system. Formalizing each VM as a variable will eventually result in an intractable high-dimensional optimization problem.
- Rejuvenating the VMM affects the running of all VMs hosted on top of it. The optimal maintenance cost is difficult to calculate when there are multiple VMs with a variety of downtime costs and rejuvenation costs.

To minimize system unavailability, we have proposed a state-control-limit-based rejuvenation policy on a single virtualized server system, composed of the VMs and the VMM, which are structurally dependent. The problem of multi-component system rejuvenation is reduced to the problem of multiple two-component systems. The two-component system evolution involving rejuvenation/recovery behavior is further modeled as an embedded Markov chain in continuous state space. The stationary law of the two-component system state is deduced. On this basis, the stationary unavailability model of the system is obtained. The particle swarm optimization algorithm is used to solve the model. The contribution of this paper lies in the state-based rejuvenation policy on the virtualized server system with structural dependence, and the probability density solution to the two-component system with continuous state degradation.

## 2. System description

### 2.1. Features of the virtualized server system

The cloud computing infrastructure consists of multiple network-connected servers. The server can be a workstation, a multi-core system, or a cluster. Cloud Server provides a virtualized environment through a VMM, instantiating one or more VMs on top of the VMM. The interface is responsible for loading prebuilt VM images from the repository and registering them with the VMM to schedule the user's VM construction request. Once instantiated, the VM starts executing and loads the operating system and applications to provide an isolated execution environment for the user. Performing proactive software rejuvenation on the VMs/VMM can improve availability of the Cloud Server.

A VMM is a software similar to an operating system kernel, such as Xen, VMware ESXi, and Microsoft Hyper-V [3,21]. Since the VMM is long-running software and is seldom rebooted, software aging accumulates more easily than other components. Out-of-memory errors results in performance degradation or crash failure of the VMM [1,13], which in turn affects the VMs running on top of the VMM. Consequently, service interruption and data loss occur. The VMM failure is detected by the Failure Detector periodically inspecting the memory usage or the response time. When a failure is detected, the status of all the running VMs is saved and the VMM is repaired. As soon as the repair has been performed, the VMs can be reactivated. To improve the server availability, a software rejuvenation task can be triggered on the VMM by the Rejuvenation Manager. In this case, the running VMs are suspended in a safe way and the VMM is rebooted; thus, the aging state of the VMM is reset. Figure 1 shows the structure of a rejuvenation-enabled Cloud Server. The features of the virtualized server system are described as below:

- A Cloud Server contains a VMM on which  $n$  VMs are instantiated. The VMM provides the virtual resources (computing resources, storage resources, and network connections) needed for VM operations. The VM operating environment is isolated and independent of each other.
- The VMM failure is detected by the Failure Detector periodically inspecting the memory usage or the response time. The VMM degradation level is a random quantity that can be detected. It is expressed as a continuous random variable  $Y(t)$ . The VMM degradation process is represented by a stochastic process  $\{Y(k); k \in N\}$  where  $Y(t)$  is the deterioration level observed at the  $k$ th discrete inspection epoch  $t_k = kT$  ( $k \in N$ ) and  $T$  is the inspection interval. The VMs running on top of the VMM also degrade due to software aging. The  $i$ -th VM degradation

process is represented by a stochastic process  $\{X_i(k); k \in N, i \in [1, 2, \dots, n]\}$ , where  $X_i(k)$  is the deterioration level observed at the  $k$ th discrete inspection epoch  $t_k = kT$  ( $k \in N$ ) and  $T$  is the inspection interval.

- The degradation process of the Cloud Server is represented by  $\{(X_i(k), Y(k)); k \in N, i \in [1, 2, \dots, n]\}$ .

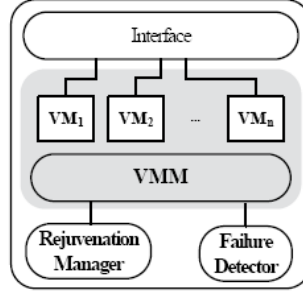


Figure 1. A rejuvenation-enabled Cloud Server

## 2.2. The state-control-limit-based rejuvenation policy

In the process of making the maintenance decision on the virtualized server, the VM (or the VMM) is rebooted or rejuvenated if it fails or requires preventive maintenance. If no failure or preventive maintenance is required, the VM (or the VMM) continues to run. The available decision set contains two available actions at each decision epoch, i.e.,  $A(S(k)) = \{0, 1\}$ , where  $S(k)$  represents the system state at the decision epoch  $k$ , 0 represents no maintenance, and 1 represents the maintenance. Under the assumption, the optimal rejuvenation policy is expressed as follows:

$$A(S(k)) = \begin{cases} 0, & S(k) < D_p^* \\ 1, & S(k) \geq D_p^* \end{cases}$$

where  $D_p^*$  is the rejuvenation threshold. This policy is called a state-control-limit policy, that is, rejuvenation is performed only if the state  $S(k) \geq D_p^*$  otherwise the system continues to run.

The rejuvenation threshold of the VMM and the  $VM_i$  is respectively represented by  $D_p^{(0)}$  and  $D_p^{(i)}$ . The failure threshold of the VMM and the  $VM_i$  is respectively represented by  $D_f^{(0)}$  and  $D_f^{(i)}$ . Periodical inspection and state-control-limit-based rejuvenation policy on the Cloud Server are described in detail in this section.

- The deterioration state of the VMM and each  $VM_i$  is inspected simultaneously at the detection epochs  $t_k = kT$  ( $k \in N$ ). Its value is  $(x_1, x_2, \dots, x_i, \dots, x_n, y)$ .
- If the deterioration state of the VMM is  $0 \leq y < D_p^{(0)}$ , then the VMM continues to degrade.
- If the deterioration state of the VMM is  $D_p^{(0)} \leq y < D_f^{(0)}$ , then rejuvenation is performed on the VMM. The rejuvenation duration is  $t_p^{(0)}$ .
  - If the deterioration state of the  $VM_i$  is  $0 \leq x_i < D_p^{(i)}$ , then the  $VM_i$  continues to degrade;
  - If the deterioration state of the  $VM_i$  is  $D_p^{(i)} \leq x_i < D_f^{(i)}$ , then rejuvenation is performed on the  $VM_i$ . The rejuvenation duration is  $t_p^{(i)}$ ;
  - If the deterioration state of the  $VM_i$  is  $x_i \geq D_f^{(i)}$ , then failure recovery is performed on the  $VM_i$ . The rejuvenation duration is  $t_f^{(i)}$ .
- If the deterioration state of the VMM is  $y \geq D_f^{(0)}$ , then failure recovery is performed on the VMM. The state of each  $VM_i$  is also restored to the initial healthy state. Since the recovery process for the  $VM_i$  can only be done after the recovery process of the VMM is implemented, the recovery duration of the VMM is  $t_f^{(0)}$  and the recovery duration of the  $VM_i$  is  $t_f^{(0)} + t_f^{(i)}$ .

### 3. Modeling the unavailability of the Cloud Server

The Cloud Server degradation process can be represented by a continuous non-decreasing stochastic process  $\{(X_i(t), Y(t)), t > 0, i = 1, 2, \dots, n\}$ . A discrete-time state sequence  $\{(x_1(k), x_2(k), \dots, x_i(k), \dots, x_n(k), y(k)), k = T_0, T_1, T_2, \dots, T_n\}$  is obtained through the discrete-time periodic state inspection and the implementation of the rejuvenation policy described in section 2.2. If the corresponding rejuvenation behavior of the  $VMM$  and the  $VM_i$  can be completed within the inspection cycle, then the state of the system at the next inspection epoch is related only to the state of the system at the last inspection epoch; therefore, the stochastic process can be modeled as an embedded Markov chain in continuous state space. According to the characteristics of the virtualized server system, that is, each VM is independent of each other and its state evolution is only affected by the  $VMM$ . The multi-component system as shown in Figure 1 can be seen as the multiple two-component series systems, each composed of one  $VMM$  and one  $VM$ . The virtualized server unavailability is determined by the unavailability of each  $VM$ . The inspection cycle consists of two phases: the rejuvenation/recovery phase and the operating phase. The  $VM_i$  does not provide services in the rejuvenation/recovery phase, and the unavailability duration is represented with  $D_m^{(i)}(t)$ . In the operating phase, a scenario needs to be considered in which a soft failure occurs before the inspection epoch; the  $VM$  degradation level has exceeded the failure threshold and performance of the  $VM$  is very poor. The  $VM$  is also considered unavailable at that time and the unavailability duration is represented with  $D_u^{(i,0)}(t)$ . The stationary unavailability of the virtualized server can be approximated by the average unavailability of the  $VMs$  in one inspection cycle.

$$\bar{A}_\infty = \frac{\sum_{i=1}^n \lim_{t \rightarrow \infty} \frac{D_m^{(i)}(t) + D_u^{(i,0)}(t)}{t}}{n} \approx \frac{\sum_{i=1}^n [E(D_m^{(i)}(T)) + E(D_u^{(i,0)}(T))]}{nT} \quad (1)$$

#### 3.1. The stationary law $\Omega(x_i, y)$ of the $VM$ - $VMM$ state

Once the multi-component system of virtualized Cloud Server is regarded as the multiple two-component systems, the study of the state evolution of the Cloud Server can be reduced to the study on the two-component  $VM$ - $VMM$  state evolution problem. Derivation of the stationary law  $\Omega(x_i, y)$  of the  $VM$ - $VMM$  state is described in this section.

The discrete time state sequence  $\{(X_i(k), Y(k)), k = T_0, T_1, T_2, \dots, T_n\}$  is obtained by periodic state inspection at discrete epochs. Obviously, due to stochastic degradation, the state of the two-component system at each inspection epoch is a two-dimensional random variable. For convenience of description, each inspection epoch is called a semi-renewal point, and the duration between adjacent inspection epochs is called a semi-renewal cycle. The sequence is modeled as an embedded Markov chain in continuous state space. According to the stationary probability density formula of the irreducible, ergodic Markov chain  $\Omega(x_i, y) = \int_{(x'_i, y')} \Omega(x'_i, y') dx'_i dy' k((x'_i, y')|(x_i, y))$ , the stationary probability density of the degradation process of the  $VM_i$  and the  $VMM$  can be derived. The stationary law of the degradation state is represented by  $\Omega(x_i, y)$ . The state of the  $VM_i$  and the  $VMM$  at the beginning of the semi-renewal cycle is  $x'_i$  and  $y'$  respectively, and the joint state is denoted by  $(x'_i, y')$ . The state of the  $VM_i$  and the  $VMM$  at the end of the semi-renewal cycle is  $x_i$  and  $y$  respectively, and the joint state is denoted by  $(x_i, y)$ .  $k(\cdot|\cdot)$ , also called the transition probability kernel ([18], p.112), is conditional probability density from the state  $(x'_i, y')$  to the state  $(x_i, y)$  at the beginning and the end of the semi-renewal cycle.

The state degradation increment of the  $VM_i$  in unit time is represented with  $\Delta X_i$ , the probability density function of which is represented by  $f_i(x)$ ,  $i = 1, 2, \dots, n$ . The state degradation increment of the  $VMM$  in unit time is represented with  $\Delta Y$ , the probability density function of which is represented by  $f_0(y)$ .

The scenarios of degradation state  $(x'_i, y')$  of the  $VM_i$  and the  $VMM$  at the beginning of the semi-renewal cycle are analyzed and the probability of occurrences of various maintenance decisions are obtained through the integration. The degradation process of the  $VM_i$  and the  $VMM$  is further considered after the corresponding maintenance occurs. The trend of degradation of the  $VM_i$  and the  $VMM$  after different maintenance combinations is finally obtained. The possible scenarios are as follows:

- 1) If  $0 < x'_i < D_p^{(i)}$ ,  $0 < y' < D_p^{(0)}$  at the beginning of the semi-renewal cycle and  $x_i < D_p^{(i)}$ ,  $y < D_p^{(0)}$  at the end of the semi-renewal cycle, i.e.  $0 < x'_i < x_i < D_p^{(i)}$ ,  $0 < y' < y < D_p^{(0)}$ , then the probability of such a scenario is  $P \{0 < x'_i < x_i < D_p^{(i)}, 0 < y' < y < D_p^{(0)}\} = \int_0^{x_i} \int_0^y \Omega(x'_i, y') dx'_i dy'$ . If  $x_i \geq D_p^{(i)}$ ,  $y \geq D_p^{(0)}$  at the end of the semi-renewal cycle, i.e.  $0 < x'_i < D_p^{(i)} \leq x_i$ ,  $0 < y' < D_p^{(0)} \leq y$ , then the probability of such a scenario is

$\int_0^{D_p^{(i)}} \int_0^{D_p^{(0)}} \Omega(x'_i, y') dx'_i dy'$ . The probability of  $0 < x'_i < D_p^{(i)}, 0 < y' < D_p^{(0)}$ , therefore, is expressed as  $\int_0^{\min(x_i, D_p^{(i)})} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T)}(x_i - x'_i) f_0^{(T)}(y - y')$ , where  $f_i^{(T)}(x_i - x'_i)$  is the  $T$ -th convolution of  $f_i(x_i - x'_i)$  and  $f_0^{(T)}(y - y')$  is the  $T$ -th convolution of  $f_0(y - y')$ .

- 2) If  $D_p^{(i)} \leq x'_i < D_f^{(i)}, 0 < y' < D_p^{(0)}$  at the beginning of the semi-renewal cycle, then the VM is restored to the pristine healthy state after rejuvenation and the pdf at the end of the semi-renewal cycle is  $f_i^{(T-t_p^{(i)})}(x_i)$ , and the VMM continues to degrade to  $y(y < D_p^{(0)})$  or  $D_p^{(0)}$ . The probability of  $D_p^{(i)} < x'_i < D_f^{(i)}, 0 < y' < D_p^{(0)}$ , therefore, is expressed as  $\int_{D_p^{(i)}}^{D_f^{(i)}} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_p^{(i)})}(x_i) f_0^{(T)}(y - y')$ .
- 3) If  $x'_i \geq D_f^{(i)}, 0 < y' < D_p^{(0)}$  at the beginning of the semi-renewal cycle, then the VM is restored to the pristine healthy state after failure recovery and the pdf at the end of the semi-renewal cycle is  $f_i^{(T-t_f^{(i)})}(x_i)$ , and the VMM continues to degrade to  $y(y < D_p^{(0)})$  or  $D_p^{(0)}$ . The probability of  $x'_i \geq D_f^{(i)}, 0 < y' < D_p^{(0)}$ , therefore, is expressed as  $\int_{D_f^{(i)}}^{\infty} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_f^{(i)})}(x_i) f_0^{(T)}(y - y')$ .
- 4) If  $0 < x'_i < D_p^{(i)}, D_p^{(0)} \leq y' < D_f^{(0)}$  at the beginning of the semi-renewal cycle, then the VMM is restored to the pristine healthy state after rejuvenation and the pdf at the end of the semi-renewal cycle is  $f_0^{(T-t_p^{(0)})}(y)$  and the VM state remains unchanged during VMM rejuvenation. After VMM rejuvenation is completed, the  $VM_i$  continues to degrade to  $x_i(x_i < D_p^{(i)})$  or  $D_p^{(i)}$ . The probability of  $0 < x'_i < D_p^{(i)}, D_p^{(0)} < y' < D_f^{(0)}$  is expressed as  $\int_0^{\min(x_i, D_p^{(i)})} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_p^{(0)})}(x_i - x'_i) f_0^{(T-t_p^{(0)})}(y)$ .
- 5) The probability of  $D_p^{(i)} \leq x'_i < D_f^{(i)}, D_p^{(0)} \leq y' < D_f^{(0)}$  is expressed as  $\int_{D_p^{(i)}}^{D_f^{(i)}} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_p^{(i)}-t_p^{(0)})}(x_i) f_0^{(T-t_p^{(0)})}(y)$ .
- 6) The probability of  $x'_i \geq D_f^{(i)}, D_p^{(0)} \leq y' < D_f^{(0)}$  is expressed as  $\int_{D_f^{(i)}}^{\infty} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of the state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_f^{(i)}-t_p^{(0)})}(x_i) f_0^{(T-t_p^{(0)})}(y)$ .
- 7) The probability of  $y' \geq D_f^{(0)}$  is expressed as  $\int_{-\infty}^{\infty} \int_{D_f^{(0)}}^{\infty} \Omega(x'_i, y') dx'_i dy'$  and the conditional probability density of state  $(x_i, y)$  at the end of semi-renewal cycle is  $f_i^{(T-t_f^{(i)}-t_f^{(0)})}(x_i) f_0^{(T-t_f^{(0)})}(y)$ .

Integrating the above-mentioned analyses in 1-7, we can get

$$\begin{aligned}
 \Omega(x_i, y) = & \int_0^{\min(x_i, D_p^{(i)})} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy' f_i^{(T)}(x_i - x'_i) f_0^{(T)}(y - y') + \int_{D_p^{(i)}}^{D_f^{(i)}} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_p^{(i)})}(x_i) f_0^{(T)}(y - y') \\
 & + \int_{D_f^{(i)}}^{\infty} \int_0^{\min(y, D_p^{(0)})} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_f^{(i)})}(x_i) f_0^{(T)}(y - y') + \int_0^{\min(x_i, D_p^{(i)})} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_p^{(0)})}(x_i - x'_i) f_0^{(T-t_p^{(0)})}(y) \\
 & + \int_{D_p^{(i)}}^{D_f^{(i)}} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_p^{(i)}-t_p^{(0)})}(x_i) f_0^{(T-t_p^{(0)})}(y) + \int_{D_f^{(i)}}^{\infty} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_f^{(i)}-t_p^{(0)})}(x_i) f_0^{(T-t_p^{(0)})}(y) \\
 & + \int_0^{\infty} \int_{D_f^{(0)}}^{\infty} \Omega(x'_i, y') dx'_i dy' f_i^{(T-t_f^{(i)}-t_f^{(0)})}(x_i) f_0^{(T-t_f^{(0)})}(y)
 \end{aligned} \tag{2}$$

### 3.2. The mean maintenance time $E(D_m^{(i)}(T))$ and the mean unavailability time $E(D_u^{(i,0)}(T))$

The scenarios of the degradation state of the  $VM_i$  and the  $VMM$  are analyzed at the beginning of the semi-renewal cycle and the probability of the occurrences of various maintenance decisions are obtained through the integration. The maintenance duration time of the  $VM_i$  and the  $VMM$  is further considered after the corresponding maintenance occurs. The mean maintenance time  $E(D_m^{(i)}(T))$  can be finally obtained.

$$\begin{aligned} E(D_m^{(i)}(T)) = & \int_{D_p^{(i)}}^{D_f^{(i)}} \int_0^{D_p^{(0)}} \Omega(x_i, y) dx_i dy * t_p^{(i)} + \int_{D_f^{(i)}}^{\infty} \int_0^{D_p^{(0)}} \Omega(x_i, y) dx_i dy * t_f^{(i)} \\ & + \int_0^{D_p^{(i)}} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dx_i dy * t_p^{(i)} + \int_{D_p^{(i)}}^{D_f^{(i)}} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dx_i dy * (t_p^{(0)} + t_p^{(i)}) \\ & + \int_{D_f^{(i)}}^{\infty} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dx_i dy * (t_p^{(0)} + t_f^{(i)}) + \int_0^{\infty} \int_{D_f^{(0)}}^{\infty} \Omega(x_i, y) dx_i dy * (t_f^{(0)} + t_f^{(i)}) \end{aligned} \quad (3)$$

If the  $VM_i$  degradation level has reached  $D_f^{(i)}$  (a soft failure occurs) at the epoch  $t_f^i ((n-1)T < t_f^i \leq nT)$  before the inspection epoch, then the  $VM_i$  is not available between  $t_f^i$  and the inspection epoch. The unavailable time is denoted by  $D_u^{(i)}(T)$ . Obviously,  $t_f^i \in (t_k, t_{k+1})$  and  $k+1 \leq T$ .  $t_k$  is used to approximate the time at which the failure occurs, so  $k$  is assigned the value of each integer in the inspection cycle, i.e.  $k \in [1, T]$ . Similarly, the unavailable time of the  $VMM$  due to the soft failure occurrences is denoted by  $D_u^{(0)}(T)$ .

According to the definition of the expectation, we can write

$$\begin{aligned} E(D_u^{(i,0)}(T)) = & \sum_{k=1}^T (T-k) * P(D_u^{(i,0)}(T) = k) = \sum_{k=1}^T (T-k) \\ & * [P(D_u^{(i)}(T) = k) + P(D_u^{(0)}(T) = k) - P(D_u^{(i)}(T) = k, D_u^{(0)}(T) = k)] \end{aligned} \quad (4)$$

where  $P(D_u^{(i,0)}(T) = k)$  indicates the probability of failure occurrences of the  $VM_i$  or the  $VMM$  at time  $k$ .  $P(D_u^{(i,0)}(T) = k)$  indicates the probability of failure occurrences of the  $VM_i$  at time  $k$ .  $P(D_u^{(0)}(T) = k)$  indicates the probability of failure occurrences of the  $VMM$  at time  $k$ .  $P(D_u^{(i)}(T) = k, D_u^{(0)}(T) = k)$  indicates the probability of simultaneous failure occurrences of the  $VM_i$  and the  $VMM$  at time  $k$ .

The event  $D_u^{(i)}(T) = k$  is equivalent to the composite event: the  $VM_i$  runs during the first  $k$  time units of the cycle and fails between  $t_k$  and  $t_{k+1}$ ; and the  $VM_i$  remains in the failure state during the last  $l-k$  periods of the cycle. Let  $H_i(k|z; l)$  be the probability that the availability time equals  $k$  time units given that the deterioration value after maintenance equals  $z$  and the semi-renewal cycle is  $l$  time units. If the  $VM_i$  is not maintained before the semi-renewal cycle, then  $z = x_i$ . If the  $VM_i$  is maintained before the semi-renewal cycle, then  $z = 0$ .  $H_i(k|z; l) = \begin{cases} f_i^{(k)}(D_f^{(i)} - x_i) & \text{no maintenance, } l \geq k \\ f_i^{(k)}(D_f^{(i)}) & \text{after maintained, } l \geq k, z = 0 \end{cases}$ , where  $f_i^{(k)}(\cdot)$  is the  $k$ -th convolution of  $f_i(\cdot)$ .

The probability  $P(D_u^{(i)}(T) = k)$  is calculated as follows.

$$\begin{aligned} P(D_u^{(i)}(T) = k) = & \int_0^{D_p^{(i)}} \left( \int_0^{D_p^{(0)}} \Omega(x_i, y) dy \right) H_i(k|x_i; T) dx_i + \int_{D_p^{(i)}}^{D_f^{(i)}} \left( \int_0^{D_p^{(0)}} \Omega(x_i, y) dy \right) H_i(k|0; T - t_p^{(i)}) dx_i \\ & + \int_{D_f^{(i)}}^{\infty} \left( \int_0^{D_p^{(0)}} \Omega(x_i, y) dy \right) H_i(k|0; T - t_f^{(i)}) dx_i + \int_0^{D_p^{(i)}} \left( \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dy \right) H_i(k|x_i; T - t_p^{(0)}) dx_i \\ & + \int_{D_p^{(i)}}^{D_f^{(i)}} \left( \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dy \right) H_i(k|0; T - t_p^{(0)} - t_p^{(i)}) dx_i + \int_{D_f^{(i)}}^{\infty} \left( \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) dy \right) H_i(k|0; T - t_p^{(0)} - t_f^{(i)}) dx_i \\ & + \int_0^{\infty} \left( \int_{D_f^{(0)}}^{\infty} \Omega(x_i, y) dy \right) H_i(k|0; T - t_f^{(0)} - t_f^{(i)}) dx_i \end{aligned} \quad (5)$$

Let  $H_0(k|z; l)$  be the probability that the availability time equals  $k$  time units given that the deterioration value after maintenance equals  $z$  and the semi-renewal cycle is  $l$  time units. If the  $VMM$  is not maintained before the semi-renewal

cycle, then  $z = y$ . If the VMM is maintained before the semi-renewal cycle, then  $z = 0$ .

$$H_0(k|z; l) = \begin{cases} f_0^{(k)}(D_f^{(0)} - y) & \text{no maintenance, } l \geq k \\ f_0^{(k)}(D_f^{(0)}) & \text{after maintained, } l \geq k, z = 0 \end{cases}, \text{ where is the } k\text{-th convolution of } f_0(\cdot).$$

Similarly, the probability  $P(D_u^{(0)}(T) = k)$  and the probability  $P(D_u^{(i)}(T) = k, D_u^{(0)}(T) = k)$  are calculated as follows.

$$P(D_u^{(0)}(T) = k) = \int_0^{D_p^{(0)}} \left( \int_0^{D_f^{(i)}} \Omega(x_i, y) dx_i \right) H_0(k|y; T) dy + \int_{D_p^{(0)}}^{D_f^{(0)}} \left( \int_0^{D_f^{(i)}} \Omega(x_i, y) dx_i \right) H_0(k|0; T - t_p^{(0)}) dy + \int_{D_f^{(0)}}^{\infty} \left( \int_0^{\infty} \Omega(x_i, y) dx_i \right) H_0(k|0; T - t_f^{(0)}) dy \quad (6)$$

$$P(D_u^{(i)}(T) = k, D_u^{(0)}(T) = k) = \int_0^{D_p^{(i)}} \int_0^{D_p^{(0)}} \Omega(x_i, y) H_i(k|x_i; T) dx_i H_0(k|y; T) dy + \int_{D_p^{(i)}}^{D_f^{(i)}} \int_0^{D_p^{(0)}} \Omega(x_i, y) H_i(k|0; T - t_p^{(i)}) dx_i H_0(k|y; T) dy + \int_{D_f^{(i)}}^{\infty} \int_0^{D_p^{(0)}} \Omega(x_i, y) H_i(k|0; T - t_f^{(i)}) dx_i H_0(k|y; T) dy + \int_{D_p^{(i)}}^{D_f^{(i)}} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) H_i(k|0; T - t_p^{(0)} - t_p^{(i)}) dx_i H_0(k|0; T - t_p^{(0)}) dy + \int_{D_f^{(i)}}^{\infty} \int_{D_p^{(0)}}^{D_f^{(0)}} \Omega(x_i, y) H_i(k|0; T - t_p^{(0)} - t_f^{(i)}) dx_i H_0(k|0; T - t_p^{(0)}) dy + \int_0^{\infty} \int_{D_f^{(0)}}^{\infty} \Omega(x_i, y) H_i(k|0; T - t_f^{(0)} - t_f^{(i)}) dx_i H_0(k|0; T - t_f^{(0)}) dy \quad (7)$$

Substitute Equation (5), Equation (6), and Equation (7) into Equation (4). The mean unavailable time  $E(D_u^{(i,0)}(T))$  can be obtained. Then, after Equation (3) is substituted into Equation (1), the unavailability of the virtualized Cloud Server can be obtained.

### 3.3. The maintenance decision model

The inspection intervals and the rejuvenation thresholds of the VMM and the VM need to be adjusted in order to minimize the stationary unavailability, hence influencing the probability of rejuvenation and recovery maintenance decisions. The rejuvenation problem under discussion can then be modelled as Equation (8), which describes a single-objective problem with constraints, namely,

$$\begin{aligned} \min & \bar{A}_{\infty}(T, D_p^{(0)}, D_p^{(i)}), i = 1, 2, \dots, n \\ \text{s.t. } & T \in \mathbb{R}^+ \\ & 0 < D_p^{(0)} < D_f^{(0)} \\ & 0 < D_p^{(i)} < D_f^{(i)} \end{aligned} \quad (8)$$

This is a multivariate continuous optimization problem that can be solved through using the optimization algorithm such as evolutionary algorithm, swarm intelligence algorithm, biological heuristic algorithm, ecological algorithm, and complex adaptive algorithm. As a stylized representation of the movement of organisms in a bird flock or fish school, Particle Swarm Optimization (PSO) is originally proposed by Kennedy and Eberhart and was first intended for simulating social behaviour [6,12]. Compared with the traditional optimization algorithm, PSO algorithm does not depend on the characteristics of the problem and the convergence speed of PSO algorithm is fast with only a few parameters needing to be adjusted [7,14]. Therefore, the PSO algorithm is chosen to solve the Cloud Server maintenance decision model.

## 4. Numerical experiments

If the degradation process satisfies strict monotonous and incremental stationary independent features, it can be modeled as a gamma process [4,19]. We chose gamma distribution as the experimental distribution of the VM and the VMM degradation processes. If we suppose that the degradation increment accords with  $\Gamma(\alpha, \beta)$  within unit time and the pdf is  $f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$ ,  $x > 0$ , where  $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$ ,  $\alpha > 0$ , then the degradation increment accords with  $\Gamma(t\alpha, \beta)$  within  $t$  time units. Performance degradation increment of the VM accords with  $\Gamma(5, 0.02)$  and performance degradation

increment of the  $VMM$  accords with  $\Gamma(5, 0.01)$ . The failure threshold of the VM and the VMM is respectively normalized as  $D_f^{(i)} = 1$  and  $D_f^{(0)} = 1$ . The mean rejuvenation time is  $E(t_p^{(i)}) = 0.15$  and  $E(t_p^{(0)}) = 0.25$  respectively. The mean recovery time is  $E(t_f^{(i)}) = 0.75$  and  $E(t_f^{(0)}) = 1.25$  respectively. The state-control-limit-based rejuvenation decision optimization model is verified in this section.

#### 4.1. Stationary law $\Omega(x_i, y)$ of the degradation state

The probability density function  $\Omega(x_i, y)$  is numerically solved and the MATLAB scripts are coded to solve Equation (2). Considering the time complexity and computational accuracy, we choose the 1.5 times state space upper limit, at which the cumulative probability of the gamma distribution equals 1, to truncate the upper limit  $\infty$  of the probability density. The state space is divided into 100 equal parts, and the inspection interval is set as  $T \in [1, 5]$ . Figure 2 illustrates the stationary probability density surface when the inspection interval  $T = 2.89$  and the degradation state of the VM and the VMM is  $D_p^{(1)} = 0.97$  and  $D_p^{(0)} = 0.90$  respectively.

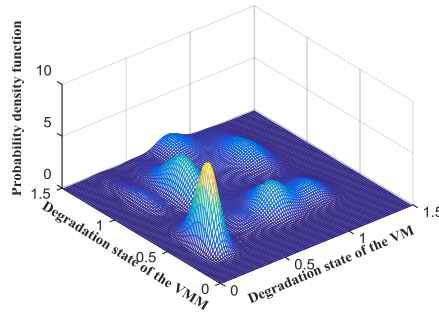


Figure 2. Representation of the probability density function  $\Omega(x, y)$

#### 4.2. Performance evaluation of the rejuvenation policy

On the basis of the joint probability density of the two-component degradation state, the effects of the inspection intervals and the rejuvenation thresholds on the stationary unavailability of the system are observed respectively. Figure 3(a) shows the effects of different inspection intervals on the stationary unavailability of the virtualized server system when the number of the VM is 1 and the rejuvenation threshold of the VM and the VMM is  $D_p^{(1)} = 0.66$  and  $D_p^{(0)} = 0.80$  respectively. Figure 3(b) shows the effects of different rejuvenation thresholds on the stationary unavailability of virtualized server systems when the number of VM is 1 and the inspection interval is  $T = 2.20$ . The surface in Figure 3(b) is convex shaped, which has minimum unavailability 0.0371148 when  $D_p^{(1)} = 0.66$  and  $D_p^{(0)} = 0.80$ . It is consistent with the PSO algorithm optimization result (Figure 4(n=1)). In our experiments, the average unavailability of the virtualized server under the lifetime-based policy is estimated by the MATLAB simulation program, and the minimum unavailability of the virtualized server is 0.069767( $n = 1$ ).

The PSO algorithm is used to optimize the multivariate continuous optimization problem of Equation (8) and the number of the VMs is 1, 2, 3, and 4, respectively. PSO algorithm parameters are set as follows: the particle swarm size is 30, and the iterative process is ended if the fitness value is not improved after the last 50 generations. Figure 4 shows the iterative evolution of the PSO running under a given parameter using the state-control-limit-based rejuvenation policy proposed in this paper. The X-axis represents the number of iterations, and the Y-axis indicates the fitness value, which is the average unavailability of the virtualized server. We can see that the number of iterations in each run is about 100 generations, and the iterative process converges to the optimal value. It is not difficult to find that the stationary unavailability of the system is significantly reduced by using the state-control-limit-based rejuvenation policy described in this paper.



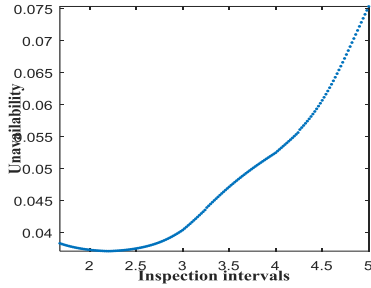


Figure 3(a). Effects of inspection intervals on the stationary unavailability

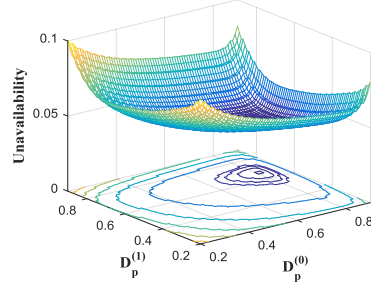


Figure 3(b). Effects of rejuvenation thresholds on the stationary unavailability

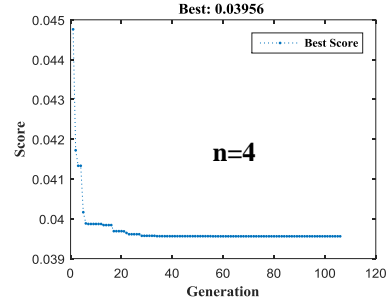
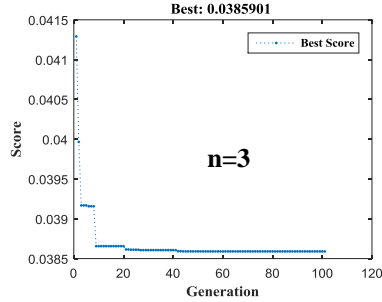
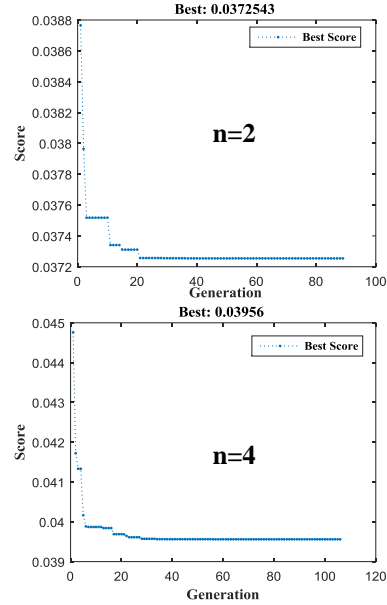
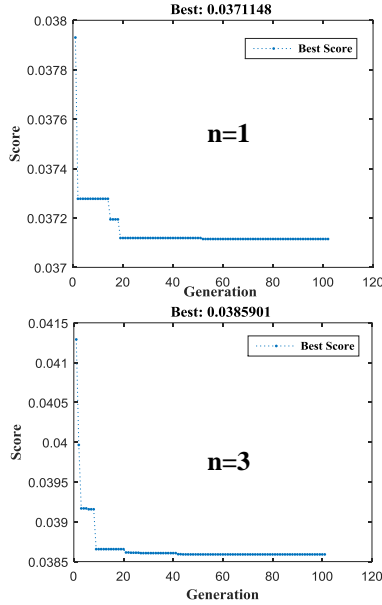


Figure 4. Iterative evolution of PSO

## 5. Conclusions

In order to guarantee the minimum unavailability of the virtualized server, the aging state of the VMM and the VMs is periodically inspected, and the state-control-limit-based rejuvenation policy is proposed. The evolution process of the virtualized server involving software rejuvenation behavior is expressed as the embedded Markov chain in continuous state space. The joint state of the VM and the VMM is analyzed at the inspection epochs and the joint probability density of the degraded state of the two-component VM-VMM is derived. On this basis, the system unavailability optimization model is obtained. The inspection interval and the rejuvenation threshold are the decision variables and the PSO optimization algorithm is used to find the approximate optimal solution of the optimization model. The experimental results show that in comparison with the lifetime-based maintenance policy, the state-control-limit-based rejuvenation policy facilitates lower system unavailability.

## Acknowledgements

This research was supported in part by the Chinese National Natural Science Foundation under Grant No. 61573250, the Key Research and Development Program of Shanxi Province under Grant No. 201703D121042-1, the Key Science and Technology Program of Shanxi Province under Grant No. 20130321006-01, the Youth Foundation of Shanxi Province under Grant No. 201601D021065 and the PhD Research Start-up Foundation of TYUST under Grant No. 20152021.

## References

1. J. Araujo, R. Matos, V. Alves, P. Maciel, F.V.D. Souza, R. Matias, and K.S. Trivedi, "Software Aging in the Eucalyptus Cloud Computing Infrastructure: Characterization and Rejuvenation," *Acm Journal on Emerging Technologies in Computing Systems*,

- vol. 636, no. 8, pp. 1557-1564, 2014
2. A. Avritzer and E.J. Weyuker, "Monitoring Smoothly Degrading Systems for Increased Dependability," *Empirical Software Engineering*, vol. 2, no. 2, pp. 59-77, 1997
3. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," *Acm Sigops Operating Systems Review*, vol. 37, no. 5, pp. 164-177, 2003
4. A. Bobbio, M. Sereno, and C. Anglano, "Fine Grained Software Degradation Models for Optimal Rejuvenation Policies," *Performance Evaluation*, vol. 46, no. 1, pp. 45-62, 2001
5. D. Cotroneo, R. Natella, R. Pietrantuono, and S. Russo, "Software Aging Analysis of the Linux Operating System," 71-80, 2010
6. R. Eberhart and J. Kennedy, "A New Optimizer Using Particle Swarm Theory," in *International Symposium on MICRO Machine and Human Science*, pp. 39-43, 1995
7. V.D.B. Frans and A.P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225-239, 2004
8. S. Garg, A.V. Moorsel, K. Vaidyanathan, and K.S. Trivedi, "A Methodology for Detection and Estimation of Software Aging," in *International Symposium on Software Reliability Engineering, 1998. Proceedings*, pp. 283-292, 1998
9. M. Grottke, L. Li, K. Vaidyanathan, and K.S. Trivedi, "Analysis of Software Aging in a Web Server," *Discussion Papers*, vol. 55, no. 3, pp. 411 - 420, 2005
10. M. Grottke and K.S. Trivedi, "Fighting Bugs: Remove, Retry, Replicate, and Rejuvenate," *Computer*, vol. 40, no. 2, pp. 107-109, 2007
11. Y. Huang, C. Kintala, N. Kolettis, and N.D. Fulton, "Software Rejuvenation: Analysis, Module and Applications," in *proceedings of 25 International Symposium, Fault-Tolerant Computing, FTCS-25*, pp. 381-390, 1995
12. J. Kennedy and R. Eberhart, "Particle Swarm Optimization." Springer US, 2011
13. K. Kourai and S. Chiba, "Fast Software Rejuvenation of Virtual Machine Monitors," *IEEE Transactions on Dependable & Secure Computing*, vol. 8, no. 6, pp. 839-851, 2010
14. J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281-295, 2006
15. F. Machida, S.K. Dong, J.S. Park, and K.S. Trivedi, "Toward Optimal Virtual Machine Placement and Rejuvenation Scheduling in a Virtualized Data Center," in *IEEE International Conference on Software Reliability Engineering Workshops, 2008. ISSRE Wksp*, pp. 1-3, 2008
16. F. Machida, S.K. Dong, and K.S. Trivedi, "Modeling and Analysis of Software Rejuvenation in a Server Virtualized System with Live VM Migration," *Performance Evaluation*, vol. 70, no. 3, pp. 212-230, 2013
17. F. Machida, J. Xiang, K. Tadano, and Y. Maeno, "Combined Server Rejuvenation in a Virtualized Data Center," in *International Conference on Ubiquitous Intelligence & Computing and International Conference on Autonomic & Trusted Computing*, pp. 486-493, 2012
18. S. Meyn and R.L. Tweedie, "Markov Chains and Stochastic Stability." Springer-Verlag, 1993
19. J.M.V. Noortwijk, "A Survey of the Application of Gamma Processes in Maintenance," *Reliability Engineering & System Safety*, vol. 94, no. 1, pp. 2-21, 2009
20. A. Rezaei and M. Sharifi, "Rejuvenating High Available Virtualized Systems," in *Ares '10 International Conference on Availability, Reliability, and Security*, pp. 289-294, 2010
21. J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues," in *Second International Conference on Computer and Network Technology*, pp. 222-226, 2010

**Weichao Dang** received a M.S. degree in Computer Application Technology at Taiyuan University of Science and Technology in 2004. He is an Associate Professor of Taiyuan University of Science and Technology. He is now pursuing his PhD degree in Control Theory and Control Engineering at Lanzhou University of Technology. His current research interests include software reliability, maintenance scheduling and cloud computing.

**Jianchao Zeng** is a Professor and a PhD supervisor at North University of China. He received his PhD degree in System Engineering at Xi'an Jiaotong University in 1990. He is now the Vice President of North University of China. His current research focuses on modeling and control of complex systems, intelligent computation, swarm intelligence and swarm robotics.