Availability Enhancement for Cloud Services by Migration based Rejuvenation: Analytical Modeling

Pa Pa Hlaing, and Thandar Thein

Abstract— Virtualization is a core technology in cloud computing and virtual machine migration is a powerful tool to facilitate system maintenance, load balancing, fault tolerance, and power-saving. As cloud services have been widely used and most of the cloud services are running on top of the virtual machine (VM), software aging in VM is a challenging issue and high availability assurance of VMs becomes a significant concern. When an application goes continuously, VM performance will degrade and failure rate will increase due to software aging. VM rejuvenation, VM migration is a promising technique for enhancing the availability of cloud services as it can postpone or prevent the software aging in VM. When application running on VM needs to be rejuvenated the hosted VM can be migrated to another VM on other hosts using VM migration and continue to provide cloud services. In this paper, the effectiveness of VM rejuvenation is investigated by Markovian modeling. Numerical examples are presented to illustrate the applicability of the model.

Keywords— Availability, Cloud Services, Markov Model, Rejuvenation, Software Aging, VM Migration

I. INTRODUCTION

NOWADAYS, Cloud computing services are becoming the primary source of computing power for both enterprises and personal computing applications. A cloud computing platform can provide a variety of resources, including infrastructure, software, and services, to users in an on-demand fashion [13]. To access these resources, a cloud user submits a request for resources. The cloud provider then provides the requested resources from a common resource pool and allows the user to use these resources for a required time period. Compared to traditional approaches, cloud computing services eliminate the costs of purchasing and maintaining the infrastructures for cloud users [7].

Virtualization technology plays a key role in cloud computing platform since it makes it possible to significantly reduce the number of physical servers in cloud data centers by having each server host multiple independent virtual machines (VMs) managed by a Virtual Machine Monitor (VMM) often referred to as a Hypervisor. Virtualization brings some benefits like better utilization of resources and fault tolerance [4].

An important feature for virtualized cloud systems is the ability to move virtual machines (VMs) from one physical host to another. This characteristic is called VM migration [3].

Many organizations and businesses which rely on cloud computing platforms require nearly uninterrupted service. Therefore system availability is an important concern for cloud platforms. In this context, software rejuvenation is an auspicious technique to achieve high availability [5]. In cloud environments the VMs run on hypervisor and applications run on hosted VMs. These components are liable to suffer failures or hangs due to software aging. Several studies have reported that the unavailability of servers more often originates from software faults rather than hardware faults. When software applications executing continuously for a long period of time, their performance degraded in rate and increased occurrence rate of hang/crash failures [16]. In this situation, VM rejuvenation mechanism can be performed as a fault prevention action and has been widely used to avoid the occurrence of unplanned failures.

For assuring high availability and reliability of systems from structural perspective, model-based assessment has been applied to many engineering domains. State-space models such as Continuous Time Markov Chain (CTMC), semi-Markov process and Stochastic Petri Net (SPN) have been used widely for evaluating the performance [10], reliability/availability [1], and performability [8] of computer systems. In this paper, we construct availability model to assess the steady state availability of the virtualized cloud system considering the effect of VM rejuvenation.

The rest of the paper is organized as follows. In section II we discuss the related work. Section III describes the architecture of the virtualized cloud system analyzed in this paper, whereas Section IV presents the availability modeling. Model analysis through the numerical results discuss in Section V. Finally we conclude our paper in Section VI.

II. RELATED WORK

There have been a lot of research works to assess the availability of the system with rejuvenation. Matheus et al. [12] proposes a comprehensive availability model of a cloud...
computing environment with time-based rejuvenation supported by the live migration mechanism. They evaluate the impact that different rejuvenation policies based on live migration produced on the steady-state availability.

F. Salfner and K. Wolter [6] investigated the effect of three time-triggered system rejuvenation policies on service availability using a queuing model. They defined a metric for steady-state availability using combination of simulation and analytical reasoning. They analyzed time-to-failure of systems with rejuvenation.

The author, Rinsaka [9] described a fault-tolerant software system with two versions of redundant structure and random rejuvenation schedule, and evaluated quantitatively a dependability measure like the steady-state system availability. They developed CTMC model with redundancy and rejuvenation, by taking into account of the failure correlation on the failure property between two software systems.

The authors [15] presented a mixed software rejuvenation policy for an operational software system with multiple degradation states, which considered both the history information and the current running state. By this policy, the system was rejuvenated when it achieved to a degradation threshold or it came to pre-determined rejuvenation interval.

Some studies incorporated software rejuvenation for VM into availability model and computed the down time cost or steady state availability of the system [2], the research paper [14] provided stochastic process based models to evaluate availability of the system in case of without virtualization technology and in case when virtualization and software rejuvenation were used.

In this paper, we discussed analytical models for evaluating the effectiveness of software rejuvenation in virtualized cloud system which experience software aging. The aim of the analytical modeling approach was to access the steady state availability determining the times to trigger rejuvenation incorporate with VM migration.

III. SYSTEM ARCHITECTURE

This study considers a system with n-physical machines. Physical machine (PM) are sometime called physical hosts and each host contain a VMM which runs the VMs with desired applications, one management server which is a component responsible for controlling the entire cloud environment by means of cloud management tool and management server needs to be up and running, because it controls the whole environment. There is a remote storage volume which is accessed by the VMs and managed by the management server. The virtualized cloud system architecture is presented in the Fig. 1.

Since software applications on VMs execute continuously for long period of time the processes corresponding to the software in execution age or slowly degrade their performance. When one of the VM on the physical host degrades performance because of execution age, rejuvenation activities will be scheduled. If performance degradation rate is slow, VM is in application level rejuvenation state and at that state the VM will be rejuvenate with some activities such as clean the internal state or service restart. If performance degradation rate is high, its state reaches system level rejuvenation state, that we refer migration state and at that state VM should be migrated to another host to become a healthy one. We assume that migration decision such as which host VM should be migrated is decided by management server.

Fig. 1 Virtualized cloud system architecture for cloud services

IV. AVAILABILITY MODELING

In this section, we present our proposal to enhance cloud service availability by applying VM rejuvenation mechanism. Long running cloud application can occur Software aging. First, we study the aging behavior of the cloud service applications which are running on VMs. Then we construct a state transition model to measure the availability of the system. The markov chain state transition diagram for the system is shown in Fig. 2. In the model, there are five states: Up State (U), Degradation State (D), Application Level Rejuvenation State (R), System Level Rejuvenation State (Migrate state) (M) and Failure State (F).

Initially the VM is functioning in the initial Up State, U. As time progresses, VM performance will be degraded and state may change from the U state to D state with rate λU. If VM performance degradation is low, its state reaches R state with triggering rate λR. At that state, application level rejuvenation activities are performed and state change form R state to U again with rate µ. When VM performance degradation rate is very high, state will enter M state with rate λM and VM will be migrated from one physical host to another with rate λM. There is no physical host that can accept the VM, all services running on the VM will enter failure state with rate λ. After the VM has been recovered with rate µ, it will become Up State again.

As a model assumption, migration probabilities are calculated by management server based on their capacity of physical hosts. We also assume that Sojourn time in all the state of the system is exponentially distributed.
We define the steady-state probabilities of the system as follows:
- Probabilities in the up state: \( P_{U_i} \);
- Probabilities in the degradation state: \( P_{D_i} \);
- Probabilities in the application level rejuvenation state: \( P_{R_i} \);
- Probabilities in the system level rejuvenation state or migration state: \( P_{M_i} \);
- Probabilities in the failure state: \( P_{F} \);

where \( i \) is the number of operational VMs.

We compute the steady-state probability by writing down the steady-state balance equations as follows.

For state \( P_{U1} \),

\[
\lambda_s P_{U1} = \left( \sum_{i=2}^{n} \lambda_m P_{M_i} \right) + \mu_r P_{R_1} + \mu P_{F}
\]  

(1)

For state \( P_{U_i} \) (\( i = 2, 3, \ldots, n \)),

\[
\lambda_s P_{U_i} = \left( \sum_{i=2}^{n} \lambda_m P_{M_i} \right) - \lambda_{m_i} P_{M_i} + \mu_r P_{R_i}
\]  

(2)

For state \( P_{D_i} \) (\( i = 1, 2, \ldots, n \))

\[
\lambda_s P_{D_i} = \lambda_d P_{U_i}
\]  

(3)

For state \( P_{R_i} \) (\( i = 1, 2, \ldots, n \))

\[
(\lambda_s + \mu_r) P_{R_i} = \lambda_s P_{D_i}
\]  

(4)

For state \( P_{M_i} \) (\( i = 1, 2, \ldots, n-1 \))

\[
\lambda_s P_{R_{i-1}} = (i-1) \lambda_m P_{M_{i-1}}
\]  

(5)

For state \( P_{M_n} \) (\( i = n \))

\[
\lambda_s P_{R_{n-1}} = (i-1) \lambda_m P_{M_{n-1}} + \lambda P_{M_i}
\]  

(6)

For state \( P_{F} \)

\[
\lambda P_{M_i} = \mu P_{F}
\]  

(7)

The conservation equation of Fig. 2 is obtained by summing the probabilities of all states in the system and the sum of equation is 1.

\[
\sum_{i=1}^{n} P_{U_i} + \sum_{i=1}^{n} P_{D_i} + \sum_{i=1}^{n} P_{R_i} + \sum_{i=1}^{n} P_{M_i} + P_{F} = 1
\]  

(8)

Combining the above mentioned balance equations with the conservation equation, and solving these simultaneous equations, we acquire the closed-form solution for the system.

\[
P_{D_i} = \frac{\lambda_d}{\lambda_s} P_{U_i}
\]  

(9)

\[
P_{R_i} = \frac{\lambda_s}{\lambda_m + \mu_r} P_{U_i}
\]  

(10)

\[
P_{M_i} = \frac{\lambda_s}{2\lambda_m - \lambda_s + \mu_r} P_{U_i}
\]  

(11)

\[
P_{F} = \frac{\lambda_s}{\mu} P_{M_i}
\]  

(12)

\[
P_{U_i} = \left[ \left( i + \frac{\lambda_d}{\lambda_s} \right) + \frac{\lambda_d}{\lambda_s + \mu_r} + \frac{\lambda_s}{\lambda_m + \lambda_s + \mu_r} \right]^{-1}
\]  

\[
\frac{\lambda_s}{2\lambda_m + \lambda_s + \mu_r} + \frac{\lambda_s}{\mu (2\lambda_m + \lambda_s + \mu_r)}
\]  

(13)

V. MODEL ANALYSIS

A. Availability and Downtime Analysis

Availability is a probability of a system which provides the services in a given instant time. In our model, services are not
available when VM is in application level rejuvenation state (R), system level rejuvenation state, migration state, (M) and fail state (F).

\[
Availability = 1 - Unavailability
\]  
(14)

\[
Availability = 1 - \left( \sum_{i=1}^{n} P_{R_i} + \sum_{i=1}^{n} P_{M_i} + P_F \right)
\]  
(15)

Downtime is the expected total downtime of the application with rejuvenation in a T time units is

\[
Downtime = T \times \left( \sum_{i=1}^{n} P_{R_i} + \sum_{i=1}^{n} P_{M_i} + P_F \right)
\]  
(16)

B. Numerical Results

In order to analyze the availability of the system, we perform numerical analysis using the following parameter values shown in TABLE I.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value (hr⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda_d)</td>
<td>VM degradation rate</td>
<td>1 time / week</td>
</tr>
<tr>
<td>(\lambda_r)</td>
<td>VM rejuvenation trigger rate</td>
<td>1 time / 3 days</td>
</tr>
<tr>
<td>(\lambda_m)</td>
<td>VM migration trigger rate</td>
<td>1 time / day</td>
</tr>
<tr>
<td>(1/\rho_{mi})</td>
<td>migration time</td>
<td>30 sec</td>
</tr>
<tr>
<td>(1/\rho_i)</td>
<td>rejuvenation time</td>
<td>10 sec</td>
</tr>
<tr>
<td>(\lambda_f)</td>
<td>failure rate</td>
<td>3 time / month</td>
</tr>
<tr>
<td>(\mu)</td>
<td>repair rate</td>
<td>1 time / hrs</td>
</tr>
</tbody>
</table>

For example, we assume that there are 3 physical hosts in our system and the state transition diagram of 3 physical hosts system is modeled in Fig. 3.

Fig.3 State Transition Diagram of 3 Physical Hosts System

Steady-state probabilities of 3 physical hosts system are as follows:

- Up state: \(P_{U1} + P_{U2} + P_{U3}\);
- Degradation state: \(P_{D1} + P_{D2} + P_{D3}\);
- Application level Rejuvenate state: \(P_{R1} + P_{R2} + P_{R3}\);
- Migration state: \(P_{M1} + P_{M2} + P_{M3}\);
- Failure state: \(P_F\);

The steady state availability and downtime analysis of 3 physical hosts system are shown in the following Figures. The results derived from numerical equations are validated with SHARPE tools. SHARPE [11] is Symbolic Hierarchical Automated Reliability and Performance Evaluator and it is a well known package in the field of reliability and performability analysis of the system.

Fig. 4 illustrates the availability changes for the proposed model with 3 physical hosts system. The influence of VM degradation rates and rejuvenation trigger rates on availability is shown. The rejuvenation transition firing rates \(\lambda_r\) are assumed 1 time/3 days and 1 time/4 days. It can be observed that the rejuvenation trigger rate increases for VM, the higher availability can be achieved.

Fig. 5 plotted the downtime as a function of the VM degradation rates and rejuvenation trigger rates. For the system with higher VM degradation rate, it can be shown that the rejuvenation trigger rate increase for VM, the lower downtime can be achieved.
The availability changes for the model with VM degradation rates and rejuvenation rates are shown in Fig. 6. The rejuvenation time $\mu_r$ are assumed 20 seconds and 10 seconds. It can be observed that the quicker rejuvenation rate for VM, the higher availability can be achieved.

The differences in downtime with different VM degradation time and different rejuvenation time are shown in Fig. 7. From the result, it is apparent that the quicker rejuvenation time for VM can enhance the availability and reduce the downtime.

In Fig. 8, we plot the steady-state availability on different physical hosts. When there is one physical host in the system, the operational VM on the host can’t be migrated to other physical host. The amount of availability increment from 1 physical host to 2 or more physical hosts is significant because there are more opportunities for the operational VM in the two or more physical host system to be migrated from one physical host to another.

We analyze the availability on different physical hosts as a function of different migration rates. The change in the availability of system with the different numbers of physical hosts and different migration rates is plotted in Fig 9. The more physical hosts in the system, the more chance the operational VM to be migrated. We also observe that the amount of availability increment depends on migration rates. The faster migration rate for the VM can enhance the availability.

VI. CONCLUSION

In this paper, we have presented an approach to study the availability analysis on virtualized cloud system for cloud services with VM migration is as a rejuvenation action. It is found that VM migration is very helpful for system level rejuvenation process of VM. We have also shown that how application level rejuvenation and system level rejuvenation can enhance the availability of the cloud services and can reduce the downtime. The feasibility and correctness of our approach is evaluated with SHARPE tools and numerical derivations. According to the evaluation analysis, the proposed migration based rejuvenation model provides the availability enhancement for cloud services.

REFERENCES


