International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8332-8337

Software Rejuvenation Model for Cloud Computing Platform

I M Umesh
System Analyst, Department of Information Science & Engineering,
Rashtreeya Vidyalaya College of Engineering, Mysuru Road, Bengaluru, Karnataka, India.
Orcid Id: 0000-0002-7595-5501

Dr. G N Srinivasan
Professor, Department of Information Science & Engineering,
Rashtreeya Vidyalaya College of Engineering, Mysuru Road, Bengaluru, Karnataka, India.
Orcid Id: 0000-0003-1059-5952

Matheus Torquato
Professor, Federal Institute of Alagoas (IFAL), Campus Arapiraca - AL, Brazil.
Orcid Id: 0000-0003-3211-7951

Abstract
Cloud computing has emerged as one of the most needed technologies that houses software systems and relevant functional entities resulting in complex, multiuser, multitasking and virtualized environments. However, reliability of such high performance computing systems depends both on hardware and software. Virtualization is the technology that many cloud service providers rely on for efficient management and coordination of the resource pool. The software systems, during operation accumulate errors or garbage leading to software aging which may lead to system failure and associated consequences. Software aging needs to be dealt with specialized techniques to facilitate continuous service delivery. To deal with software aging, a technique called software rejuvenation exists that reboots or re-initiates the software to avoid fault or failure. As cloud service run on virtual environments, impacts of aging in virtual environments needs to be addressed. The VMs run on a Virtual Machine Monitor (VMM) also called as hypervisor which is liable to suffer failures or hangs due to software aging. This paper explores how the aging of Virtual Machine monitors can be addressed. Once the aging patterns of VMM are detected, the Virtual Machines running on it are migrated to healthy VMM; aged VMM is rejuvenated and VMs are migrated back. After rejuvenation, the hypervisor supports the operation of VMs in an improved state.
Keywords: Cloud computing, Virtualization, Software aging, Rejuvenation

INTRODUCTION

One of the main issues in cloud services is software aging, be it Software as a Service (SaaS), Platform as a Service (PaaS) or Infrastructure as a Service (IaaS). Software aging is the type of software fault that reduces the performance of the software system. Software aging happens due to the accumulation of aging related bugs that consume resources leading to resource exhaustion. Aging effects are the result of error accumulation that leads to resource exhaustion. This status of softwares makes the system gradually shift from healthy state to failure prone state. The system performance metrics needs to be monitored in order to find the aging patterns while the system is running. The accurate prediction of time of aging needs to be forecasted to initiate the necessary actions to counter the aging effects. The software aging consequences can be avoided by using the technique called software rejuvenation.

Software rejuvenation is the proactive technique proposed to counter software aging. Software rejuvenation mechanism involves a series of steps such as periodically stopping the application and restarting it after cleaning the internal state. Software rejuvenation executes the actions that include garbage clearance, flushing of buffer queues, reinitializing of the internal kernel tables and cleaning up of the file systems. Intrinsically, it cleans and restores the application operating environment. The rejuvenation methods vary in the approach they employ for carrying out the rejuvenation. Some of the techniques use time-based approach which rejuvenates the system at pre-determined time interval. The other techniques in practice are inspection-based which involves the monitoring of aging indicator metrics and apply rejuvenation mechanism during the forecasted period. As rejuvenation mechanisms employ reboot type of activity, the service may be disrupted which in turn has business impact. Important research issue is to determine when and how often the software should be rejuvenated and the rejuvenation techniques to be followed to avoid the software aging effects.

Usually cloud services have virtualized environment for optimized maintenance. Hence software aging on virtualized environment needs to be addressed. There exists a need to rejuvenate the different layers of the virtualized environment using innovative approaches to enhance the service availability by reducing the downtime to zero. The virtualized

environment consists of different layers such as physical hardware, hypervisor (VMM), Virtual machines (VM), Operating System and Applications running on top of operating system. The aging issues in these layers are to be monitored and necessary rejuvenation action to be triggered in order to enhance the service availability and reduce the downtime. The Virtual Machine Monitor (VMM) also called as hypervisor is liable to suffer failures or hangs due to software aging [1]. This work addresses one such need i.e., rejuvenation of virtual machine monitor which hosts multiple virtual machines. The proposed rejuvenation mechanism uses genetic algorithm based migration method for migration of VMs before rejuvenating VMMs. The result indicates improved service availability.

The remaining sections present related work, aging forecasting techniques used and the proposed rejuvenation model. The results and discussions are done in the last section.

RELATED WORK

Researchers have employed different techniques for aging detection and rejuvenation. Previous studies have used measurement-based and model-based rejuvenation approaches. Measurement based approaches directly monitor system variables which are aging indicators and predict software aging patterns by analysing the collected runtime data statistically. Model-based studies can be distinguished by the type of stochastic process used to model the phenomenon such as Markov-based and Petrinets.

Alonso et al., [2] evaluated ML (machine learning) algorithms such as decision trees, K-nearest neighbour and Random forest using the R statistical language for aging prediction. The researchers used different sets of values when the ML algorithm had parameters to create different configurations of the same algorithm. The results indicated that Random Forest performs better than the rest of the models. Toshiaki Hayashi et al., [3] estimated the performance degradation by passively measuring the traffic exchanged by virtual machines. The authors have justified the selection of traffic characteristics as a performance information source by citing several advantages. This data along with the recorded traffic metrics was tested with the C4.5 machine learning classifier that constructed a decision tree to identify performance. This is a non-intrusive method of metrics collection as the traffic measurement is done on separate machine that is not a part of virtual environment. This facilitates the metrics extraction even under extreme performance degradation.

Jing Liu et al., [4] proposed an adaptive failure detection method. The parameters chosen for aging detection are CPU and memory usage; the delay in transmission of packet between service components and aging failure detection module. The collected metrics are encapsulated into a packet and sent to the aging detector module. The procedure used achieves two tasks, packet arrival time and the information it carries. Some failure probability is expected if the message does not arrive on time. The CPU usage and free memory available is used to detect the aging severity. The aging severity is divided into four levels i.e., from L1 to L4. Aging Degree Evaluator module inserts it into centralized failure event queue. Based on aging degree number, this queue is ordered and the top events will be rejuvenated inevitably.

The research work done by Yongquan Yan [5] indicates the significance of choosing the proper data set. The researcher proved that choosing the proper data set is more important than the method used to analyse the collected metrics. The work compares the resource utilization of a webserver that is not subjected to artificial load, but a true load which has aging patterns using linear and nonlinear methods. Lei Cui et al., [6] concentrated their work on finding the impact of software aging defect on virtual machine and physical machines. The aging rate in both the forms was calculated and compared. The outcome of this study is aging effects are more in virtual machines than physical machines which was caused due to aging effects in code of hypervisor or depletion due additional calls in VMM layer.

Rejuvenation strategies proposed so far include rejuvenation of software systems running on physical machines and virtual machines. Some models use rejuvenation mechanism that takes care of saving the status of virtual machine and reboot to clean its internal state. Some other mechanism used in virtual environments include migration of VMs, reset the status of VMMs etc.,

Kenichi Kourai et al., [7] opined that individual components of the virtual environment need to be rejuvenated independently. Whenever the aging patterns are observed on hypervisor or virtual machine layers, that specific layer only requires rejuvenation as the rejuvenation on other layers consumes computing resources. The researchers proposed a new technique in which VMM layer is enabled for reboot by suspend / resume mechanism of virtual machines running on it. XEN hypervisor was used for experiments and effectiveness has been observed through several runs. The proposed technique reduced the downtime by 74% and throughput was maintained as in pre reboot status.

In the model proposed by Aye Myat Myat Paing and Ni Lar Thein [8], a high availability cluster was configured between virtual machines running on different physical machines. In the event of failure, the standby physical server takes over the services using live migration of VMs. The down time of a VM caused by live VM migration is very small and the VM continues the execution even while the original host is down. The researchers have presented possible combinations of server virtualization technology, high availability cluster and time-based software rejuvenation which have been designed to use over any server or service.

Accelerated Life Tests were used by Jing Zhaoa [9] to study the application failures due to memory leaks by injecting

memory leaks. The researchers obtained the accelerated life test results by injecting memory leaks. The conclusion is that the optimal rejuvenation by parametric method is stable, non-parametric estimation adapt to changes and eventually converge.

Payal Kulkarni [10] proposed a method of shifting of application on number of virtual machine to avoid workload. If memory occupied on VM -1 is greater than threshold memory, then find out the application occupying max memory. Initiate look method to find any other VM in same network having (total memory occupied + memory of considered application) less than its threshold memory. If found, send 'application close' command to VM-1 and 'application start' command to VM-2 (shift application from VM -1 to VM -2).

The conclusion of the rejuvenation studies is that there is a need for improvement in rejuvenation strategies applied to virtualized environment. The multi layered virtual environment needs an optimal rejuvenation model that dynamically rejuvenates aged components improving the service availability and reducing the downtime. The presence of proper rejuvenation mechanism increases the trust-worthiness and reliability of the cloud based systems that leverage virtualization technology.

Software Aging Forecasting

This section discusses the software aging forecasting process. Accumulation of aging related bugs leads to resource exhaustion. Hence, to prevent the system crash, it is necessary to predict resource exhaustion time. As the current work is focused on addressing the rejuvenation of VMM, it is necessary to monitor multiple metrics that reflect broader utilization of the resources of VMMs. The metrics collected here are CPU usage, memory usage, network bandwidth and disk access data as the aging of these resources directly impact the performance.

One month data was captured from different physical hosts. In order to predict software aging patterns of VMMs, analysis was done for the collected data to find whether software aging trend exists, which is indicated by exhaustion of resources. Figure 1 depicts the memory utilization graph of one month.

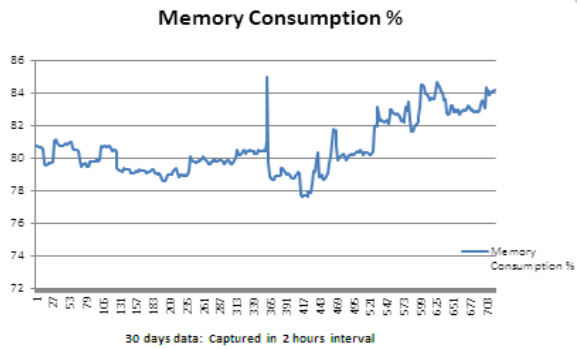

Figure 1: Memory Consumption trend of VMM

Resource exhaustion can be predicted by using the Machine Learning (ML) algorithms. The aging forecasting is done using time series forecasting. The basic learner used is Linear Regression. The captured data has been used to forecast the aging patterns using machine learning framework WEKA. In addition to time series forecasting, REPTree is also used to estimate the resource consumption. REPTree which is a fast decision tree learner that builds a decision using information gain/variance and prunes it using reduced-error pruning (with backfitting).

The parameters of model as generated by weka tool are shown in Table 1.

Table 1: REP tree model parameters

Parameter	Value
Tree size	67
Absolute error	2.4
Relative absolute error	2.69

This model is converted to java code to be included within the simulation environment. Figure 2 indicates the prediction of memory consumption metric.

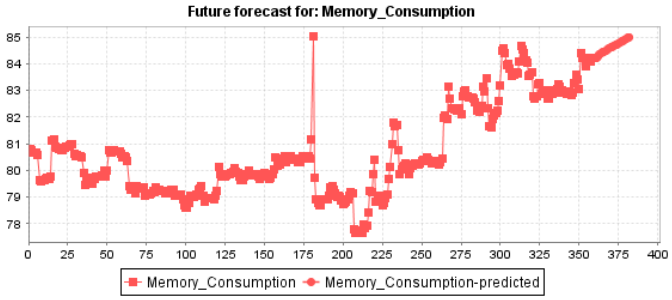

Figure 2: Memory Consumption prediction

The resource utilization forecasting is done for the aging indicator metrics such as CPU usage, Memory usage and the probable time of aging is given as input to rejuvenation process. As the present work is focussed on rejuvenation technique, the discussion on aging detection process is limited. Figure 3 indicate the aging prediction process flow.

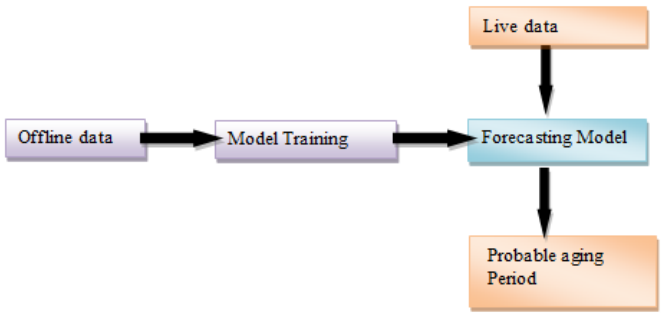

Figure 3: Aging prediction process

Proposed Rejuvenation Model

The architecture of the proposed model is indicated in the figure 4. The VMs are hosted on n number of hosts. The software aging of VMMs is estimated using the procedure mentioned in previous section. Once the aging is detected, VMs hosted on aged host are migrated to healthy host and rejuvenation of aged host is carried out. During migration, the genetic algorithm based procedure is used to identify the healthy host in optimal manner. After rejuvenation, VMs are migrated to original host. The rejuvenated host are free of aging related bugs and hence service availability can be enhanced.

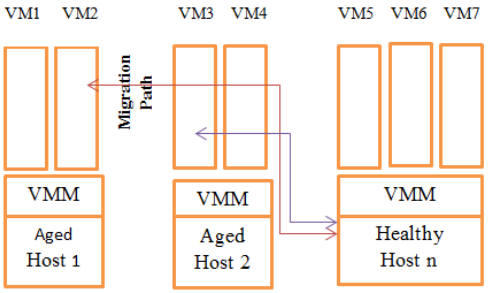

Figure 4: Proposed Model

The entire scenario of aging detection and VM migration has been simulated using the CloudSim, a simulation tool. The proposed model has been simulated using the real life workload traces from CoMon project, a monitoring infrastructure for Planet Lab. The resource utilization data is of more than a thousand virtual machines from servers located from five hundred different places all around the world. The data is randomly chosen for the simulation runs.

Once the software aging is detected, rejuvenation of aged VMMs is to be carried out. This can be achieved by rebooting the VMMs. But, before reboot, all VMs need to be migrated to healthy VMM. Live migration of virtual machines is the technique that migrate VM from one physical machine to another physical machine without causing outages.

VM Migration rule

After performing the aging detection, VMMs selection algorithm is executed to offload the VM from the aged host and place it to the healthy host, so as to reduce downtime probability and inaccessibility. We have applied least migration time (LMT) policy that exhibits VM (say v) migration while ensuring least time to migrate or place v onto a healthy host. In our proposed model, the migration time is estimated in terms of the resources being utilised by v divided by the total resources accessible for the host. Let us consider that VM_p be the VM connected with the host node p. Then, the selection of v for migration is made based on following equation

$$R_r(v)(P) < R_a(K) - R_r(K) \quad \text{---------} \quad (1)$$

Where $R_r(v)(P)$ represents the resources required for VM v hosted on VMM P, $R_a(K)$ is the available resource in host K and $R_r(K)$ is the required resource for VM v in VMM K.

Genetic Algorithm based migration technique

The genetic algorithm is a method for solving optimization problems that is based on biological evolution style of natural selection. In this work, genetic algorithm based migration technique has been employed to identify the target host for migrating the aged VMs. The proposed model uses the following steps for VM migration.

Table 2: VM Migration steps

Step		Activity
1		Forecast software aging probability based on aging indicator metrics
2		Identify the aged and healthy VMMs
3		Migrate VMs on aged VMM to healthy VMM using Genetic Algorithm based method.
	3.1	Map aged VMMs and healthy VMMs to use it as initial population for GA.
	3.2	Sort all VMMs on the basis of its aging probability.
	3.3	Calculate the resource requirements of VMs running on aged VMMs.
	3.4	Select the healthy hosts with minimal workload and required resources. Fitness Value = Total resources of VMM –(Resources being used + Required resources for VM to be migrated)
	3.5	Perform mapping for aged VMMs and healthy VMMs and obtain proposed partner chromosome
	3.6	Perform cross over between aged VMM and VMM with higher fitness value
	3.7	Execute migration between selected chromosomes
4		Rejuvenate aged VMMs
5		Migrate back the VMs to original VMM
6		Repeat the process as per the policy*

* The policy employed for rejuvenation schedule is based on input obtained dynamically from aging detection process. If the aging status of VMM is critical, then the rejuvenation is executed immediately, else rejuvenation scheduling will be done after more VMMs are reported aged.

The assumption here is that all the VMMs use the shared storage. Hence, live migration is possible. Live migration enables the users to continued access even during migration.

RESULTS AND DISCUSSIONS

The LMT selection policy was executed that prepares mapping for aged VM migration, which has been followed by proposed Genetic algorithm based rejuvenation using VM migration approach. In order to evaluate the proposed model, the VM migration is also done using other popular techniques like Ant Colony Optimization (ACO) and best fit decreasing (BFD) based placement schemes. The key performance parameters such as SLAV downtime, SLA caused performance degradation and steady state system availability has been estimated.

Live migration is primarily used to achieve zero downtime during maintenance activities. The downtime needs to be within the SLA (Service Level Agreement) when executing migration of VMs.

Figure 5 indicates the downtime, wherein it clearly evident that the proposed technique outperforms other heuristics.

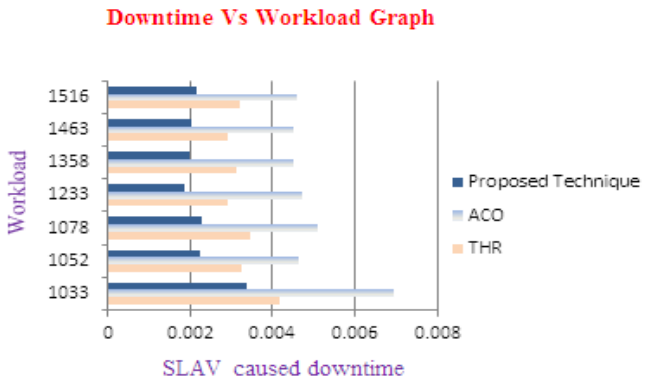

Figure 5: Downtime caused by different techniques

One of the main objectives of the proposed technique is to ensure the higher service availability during rejuvenation process. Figure 6 indicates the increased resource availability up to 2% higher than the other techniques.

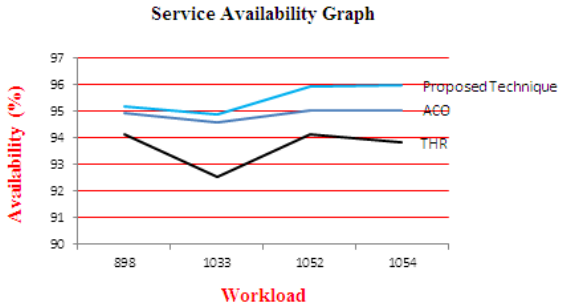

Figure 6: Service Availability Comparison

Figure 7 illustrates the reduced performance degradation that happens during rejuvenation process using the proposed technique.

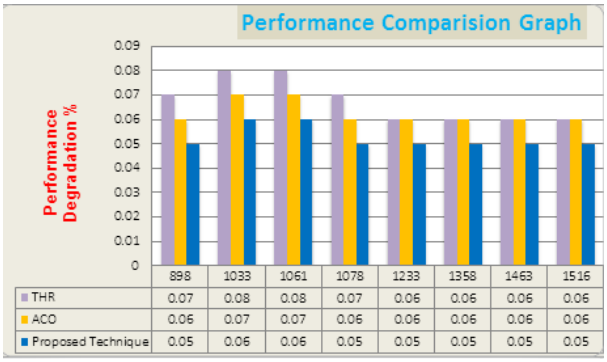

Figure 7: Workload vs Performance Degradation comparison

CONCLUSION AND FUTURE WORK

The proposed model accomplishes better model for rejuvenation of virtual machine monitors. The increased service availability and reduced performance degradation enables the proposed technique to be more realistic and responsive. The proposed rejuvenation model is potentially a better solution for cloud infrastructures as well as online/offline application softwares that are hosted on virtual machines.

As a result of implementation of the proposed technique, there was a considerable improvement in the performance indicators viz., service availability, performance as discussed above. This work can be enhanced by implementing a more advanced method for accommodating the migrated VMs to any other host instead of moving them to original host. The method can be developed that reduces consumption of computing resources and enhances service availability to large extent.

REFERENCES

[1] Melo, Matheus, Paulo Maciel, Jean Araujo, Rubens Matos, and Carlos Araujo. "Availability study on cloud computing environments: Live migration as a rejuvenation mechanism", 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp 1-6, 2013.

[2] Alonso, J. Belanche, L. Avresky, .R. "Predicting Software Anomalies Using Machine Learning Techniques", 10th IEEE International Symposium on Network Computing and Applications (NCA), pp 163-170, 2011.

[3] T. Hayashi and S. Ohta, “Performance Degradation of Virtual Machines via Passive Measurement and Machine

Learning", International Journal of Adaptive, Resilient and automates systems, pp 40-56, 2014.

[4] Jing Liu, Jiantao Zhou, Rajkumar Buyya, "Software Rejuvenation based Fault Tolerance Scheme for Cloud Applications", IEEE 8th International Conference on Cloud Computing, pp 1115-1118, 2015.

[5] Yongquan Yan, "A Practice Guide of Predicting Resource Consumption in a Web Server", Review of Computer Engineering Studies, pp 1-8, 2015

[6] Lei Cui, Bo Li, Jianxin Li, James Hardy, and Lu Liu, "Software Aging in Virtualized Environments: Detection and Prediction", IEEE International Conference on Parallel and Distributed Systems, pp 718-719, 2012.

[7] Kenichi Kourai, Shigeru Chiba, "Fast Software Rejuvenation of Virtual Machine Monitors", IEEE Transactions on Dependable and Secure Computing, 8(6):839 – 851, 2011.

[8] Aye Myat Myat Paing and Ni Lar Thein, "Stochastic Reward Nets Model for Time based software Rejuvenation in Virtualized Environment", International Journal of Computer Science and Telecommunications, pp 1-10, 2012.

[9] Jing Zhaoa, YanBin Wang, GaoRong Ning, Kishor S. Trivedi, Rivalino Matias Jr., Kai- Yuan Cai, "A comprehensive approach to optimal software rejuvenation", Performance Evaluation, ELSEVIER, pp 917-933, 2013.

[10] Payal Kulkarni, "Software Rejuvenation and Workload Distribution in Virtualized System", International Journal of Innovative Research in Computer and Communication Engineering, pp 5966-5973, 2015.