

Software aging prediction and rejuvenation in cloud computing environment-a new approach

Shruthi P¹, Nagaraj G Choli²

¹Department of Computer Science and Engg, Global Academy of Technology, Bengaluru, India

²Department of Information Science and Engg, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Nov 1, 2020

Revised Mar 19, 2021

Accepted Mar 29, 2021

Keywords:

Rejuvenation
Software aging
Virtual machine

ABSTRACT

Service availability is one of the major requirements for user satisfaction. Several researches were conducted in recent years to find suitable infrastructure to enhance the availability. Even though both hardware and software are to be in good condition, in recent years, software faults are the major concern for service availability. Software aging is a type of software fault. Software aging occurs as a result of errors accumulation in the internal environment of the system leading to performance degradation. To manage software aging, technique used is software rejuvenation. There exist two kinds of approaches for studying software aging and deriving optimal software rejuvenation schedules. The two approaches are measurement based and model based. In model based approach, analytic models are built for capturing system degradation and rejuvenation process. In measurement based approach, attributes are periodically monitored and that may indicate signs of software aging. In this work, a prototype of measurement based model has been developed. The model captures the aging indicator metrics from cloud environment and rejuvenates once the system reaches aged status. The proposed model uses platform independent, non-intrusive technique for capturing metrics. The rejuvenation carried out after analysing the captured metrics, increases the availability of the service.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Shruthi P

Department of Computer Science and Engineering

Global Academy of Technology

Ideal Homes Township, Mysore Rd, Aditya Layout, RR Nagar, Bengaluru, 560098, Karnataka, India

Email: shrutip@gat.ac.in

1. INTRODUCTION

The phenomenon of software that continuously run for long time often exhibit an increasing failure rate or a degrading performance has been studied under the name of “software aging” [1]. This happens as a result of exhaustion of resources of operating system, accumulation of errors, data corruption, memory leaks, unreleased file locks, data fragments, unreleased database locks and so on. Software aging has been reported in cloud environment [2], web servers [3]-[4], enterprise clusters [5], online transaction processing systems [6], spacecraft systems [7] and military systems [8]. The presented work here focuses on software aging in virtualized environment as it provides the platform for cloud computing environment.

Internal error accumulation that happens because of software aging leads the software to failure prone state. The preventive methods like software rejuvenation cause direct or indirect costs because of system downtime. To mitigate the loss caused by rejuvenation, it is recommended to have optimal time for scheduling the rejuvenation which is based on system state prediction. If the system state can be predicted, rejuvenation of software can be better scheduled. Figure 1 shows the degradation process.

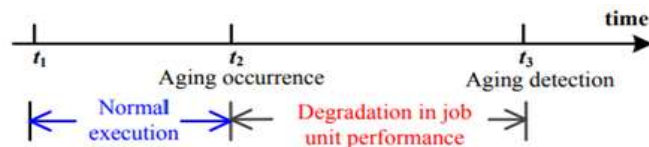


Figure 1. Performance degradation due to software aging source [9]

Several researchers have focused on time series or analytical models for analysis of software aging issues. Usage of machine learning algorithms for prediction of software aging is neglected. In this work, software aging prediction is done using machine learning. In the first part, data collected is pre-processed. Feature selection is done next for finding the subset of the collected metrics. In the end, machine learning algorithms are used for modelling the aging of software and prediction of aging.

2. RELATED WORK

Machine learning techniques can offer several applications in modelling of performance and evaluation. Defect diagnosis is one such application based on state prediction of a particular system. This can be achieved by monitoring the various parameters of the system. When the complexity of the problem is more and cannot be handled by the models created manually, machine learning models are preferred. Widely used algorithms are naïve bayes, decision trees, neural networks and support vector machines. Some of the previous studies have used tools like WEKA, and R. In some studies the researchers have implemented their own algorithms. It can be observed that the results were good wherever pre-processing strategies like filtering, feature selection, clustering, bagging and bootstrapping were used.

Alonso, *et al*, [10] performed evaluation of machine learning algorithm that predicts failure time that happens because of dynamic software aging. A benchmark application called TPC-W was used for simulation of bookstore application execution. The application was installed using three-tier web J2EE application server. M5P is the algorithm used for classification. The model implemented using WEKA achieved acceptable prediction accuracy. This was done against scenarios with complexity with small data sets for training. Ohta and Hirota [11] developed a machine learning based method for estimation of number of computing nodes required in a server cluster to maintain satisfactory performance against changes in load. Httpperf was used to generate load to a small cluster. The results of the experimentation proved the effectiveness and feasibility of proposed method.

Hayashi and Ohta [12] developed a technique for detection of performance degradation by passive measurement of traffic between virtual machines. Various metrics related to traffic like packet rate, TCP SYN loss rate, connection rate, bit-rate and number of flows were monitored. The frequency of monitoring is 60 seconds by built-in program. The classifier used was C4.5 that constructed a decision tree. The traffic metrics were used to detect the performance. The ratio of error was 2.2% for 1000 instances of test data. Matheus Tarquato, *et al*, [13] presented a mechanism to support software aging and rejuvenation process. The researchers used an approach named SWARE which has phases Stress-Wait and REjuvenation. In the stress phase, workload was given to system to observe the impact in internal state of the system. Wait phase includes the study of system behaviour after generating the load. The work provided evidences for software aging and using live migration of VM helps to reduce the impact of software aging in virtualized environment.

Fumio Machida, *et al*, [14] state that in virtualized environment, both of the layers-virtual machines and virtual machine monitor are at the risk of software aging which happens because of bugs related to aging. This may lead to failures if not addressed. The researchers proposed a combined rejuvenation technique that performs rebooting of virtual machine and virtual machine monitor at once in order to rejuvenate both. By experiments, it was proved that the availability is enhanced along with high resource utilization. Kenichi Kourai [15] recommended a new approach for fast rejuvenation of VMMs. The procedure is called warm-VM reboot. The process involves rebooting of only VMM. During VMM reboot, VMs are suspended and resumed without saving the memory images to persistent storage.

H. Meng, *et al*, [16] worked to provide a cost-effective policy for rejuvenation to ensure high availability. This also mitigates the maintenance cost in cloud environment. In the proposed work, system is rejuvenated either at a predetermined time or after reaching threshold whichever happens first. M. Torquato *et al*, [17] opine that in virtualized systems, the significant issue is software aging which may degrade the system performance and later lead to failure. In this work, researchers proposed a security evaluation method that is based on availability model. This model includes migration of VMs as a rejuvenation method of virtualized environment. Software aging forecasting was done by researchers using time series model [18]. Rejuvenation in

cloud computing platform was attempted by researchers in [19]. Software aging and rejuvenation related works are attempted by various researchers as mentioned in [20]-[25]. By observing the previous works, it can be concluded that there is a need for platform independent, non-intrusive software aging detection method and also new approach for rejuvenation which increases the availability of cloud service.

3. THE PROPOSED MODEL

The proposed model building involves various steps such as setting up test bed, aging indicators identification, evaluation of overhead of used tools, collection of metrics, building model using machine learning framework, prediction of software aging and framework for software rejuvenation easily.

3.1. The test bed

The experimental setup has the infrastructure that include physical machine to host VMs, hypervisor, virtual machine, operating system, application, load generator, metrics collector, machine learning framework and physical machine (rejuvenation controller). The physical machine with configuration of Core i5 3.3 GHz processor, 8 GB RAM and 500 GB HDD. The hypervisor used is VMware ESXi. VMware ESXi is an hypervisor developed by VMware for deploying and serving virtual machines. ESXi runs on bare metal without running an operating system. The VMs created have the following configuration of Core i5 processor, 2 GB RAM, 40 GB HDD. Operating system installed for the test run is Windows Server. The application installed was online shopping demo site which was built on XAMPP platform.

Using load generators like Jmeter, several users load can be simulated simultaneously. The tool generates load which enables performance analysis under different conditions. In this work, test runs were done varying the number of users from 5 to 300. To collect the metrics, PRTG (paessler router traffic grapher) tool has been used.

Machine learning enables the computing devices to learn without being programmed explicitly. It is a process of development of programs that learn themselves. The algorithms produce result when exposed to new data. It is similar to data mining. But, instead of searching for patterns in data, machine learning uses the data to pattern detection and adjusts the program accordingly. The capability of machine learning algorithms is exploited here for forecasting of software aging.

Weka is an open source tool for data mining tasks. The algorithms can be used directly with a data set or can be called from java code. Weka has tools for classification, pre-processing, clustering, regression and visualization. The tool is well suited for development of new machine learning methods. The rejuvenation controller machine includes metrics collector and aging forecaster application.

3.2. Aging indicators identification

The variables used to identify the system status either individually or in combination are called aging indicators. These can be classified in two general classes as per the granularity- application specific and system-wide indicators. Information pertaining to sub-systems is provided by system-wide indicators. Examples of such aging indicators are memory usage, CPU usage, file table size, used swap space and system load. The information related to specific application is provided by application-specific indicators. Examples of application specific aging indicators are application response time, and Java VM heap size.

The virtualized environment consists of various layers such as physical hardware, virtual machine monitor also called as hypervisor, virtual machine, operating system, applications and application component. The status of aging indicators from all layers need to be collected to estimate the probable hang or crash of that particular layer and relationship between them needs to be analysed to build a robust aging forecasting model. The aging indicators from all levels identified are explained here. Monitoring multiple metrics is necessary as it reflects broader resource utilization. The metrics identified for this research work and justification is given in Table 1.

Table 1. Aging indicators

Level	Aging indicator	Justification
Application	Application response time	Response time of application in milliseconds indicate the response status.
Operating System /	CPU load	CPU load indicator in percentage. This metric helps to manage CPU usage.
Virtual Machine	Memory Availability	Available memory indicator that helps to manage memory usage.
Virtual Machine	CPU Load	Environment variable that indicate CPU load of VMM.
Monitor	Memory Consumption	Environment variable that indicate memory status of VMM.

3.3. Overhead of metrics collector

It is required to collect the metrics related to performance without affecting the physical and virtual machines performance during run time. The metrics collected indicate the resource usage status and hence non-intrusive technique needs to be used. Considering this requirement, network monitoring tool called PRTG has been used that uses programming interfaces of each device monitored. This setup keeps the devices free of additional performance overhead. For testing the resource consumption by PRTG, the application executing on the virtual machine was stopped and CPU load on the machine is observed. It can be noticed from Figure 2 that the resource utilization by metrics collector is negligible.

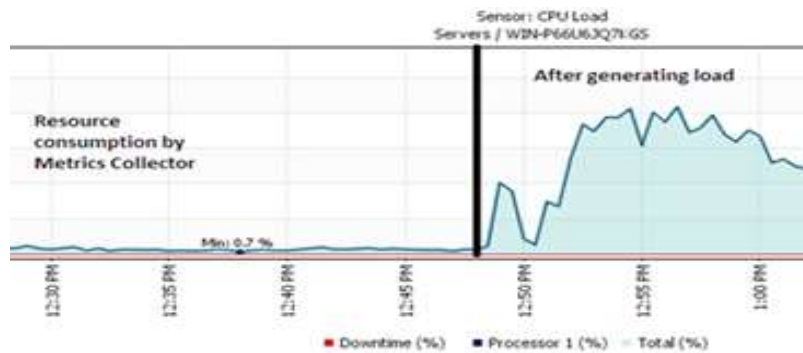


Figure 2. Performance overhead of metrics collector

3.4. Metrics collection

The aging indicator metrics are collected for specific period of time for all the layers of virtualized environment. The application was given a varying load as mentioned in Table 2. Figure 3 depicts the screenshot captured during load simulation.

Table 2. Generated load

Load generation details	Duration/Type
Concurrent users	10-300
Frequency of load firing	Once in 30 minutes
Duration	4-24 hours
Load type	Application Access



Figure 3. A random CPU workload graph

3.5. The aging forecasting process

The long running web application is considered for studying the software aging process. The metrics which are aging indicators are collected for a specific period of time as indicated in previous section. Using the collected metrics, aging detection model has been built using WEKA machine framework. An algorithm, Time

Series is used to develop a model for predicting software aging. Figure 4 shows the aging forecasting model. Resource consumption metrics is extracted and using developed forecasting model, prediction is done. The learners used in this case are gaussian processes and linear regression. The analysis of forecasted values is done for finding the patterns of aging.



Figure 4. Aging forecasting model

3.6. The aging forecasting process

Once the aging of virtual machine and virtual machine monitor are detected, the rejuvenation is performed. The rejuvenation is achieved by live migration of VMs from aged VMMs to healthy VMMs. This is depicted in Figure 5. The rejuvenation mechanism is summarised in Table 3.

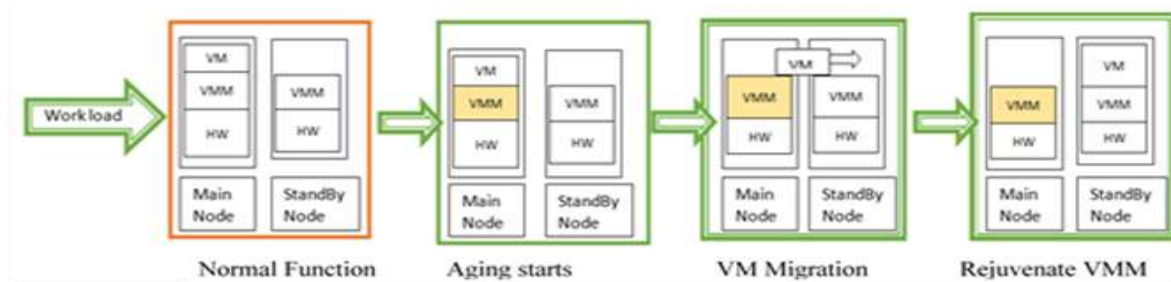


Figure 5. The proposed model

Table 3. Rejuvenation methodology

Activity	Rejuvenation process
<ul style="list-style-type: none"> • Detect overloaded hosts and migrate some VMs based on Minimum Migration Time policy to other hosts. • Detect under loaded hosts and migrate all VMs to other hosts. • Rejuvenate under loaded hosts after migration of all VMs. 	<ul style="list-style-type: none"> • The live migration does not have any significant impact on the internal states of VM. VMs migrated from aged hosts land on less aged VMM thus avoiding further aging accumulation • Hosts get rejuvenated, reinitiates VMM leading to fresh state and ready to receive migrated VM. • Depending on the aging indicators of different granularity, rejuvenation is executed at different levels like application level or VM level.

Migration of VMs is performed on the basis of time required to execute complete migration of a particular VM housed on a host compared to other VMs on that host. The migration time is estimated in terms of RAM being utilized by the VM divided by the additional network bandwidth accessible for that VMM. For example, V_j is a set of VMs housed on host j . The MMT policy determines a VM v by means of fulfilling the following conditions. This is shown in (1).

$$v \in V_j | \forall a \in V_j, \frac{RAM_u(v)}{NET_j} \leq \frac{RAM_u(a)}{NET_j} \tag{1}$$

Where $RAM_u(a)$ is the amount of RAM being utilized by the VM a ; and NET_j indicates the available bandwidth for migration from the host node.

4. CONCLUSION

In this work, a new method has been proposed for forecasting of aging. For this, aging indicator metrics are used. After analysing the aging patterns, the time of system degradation or failure is estimated. The developed aging prediction model is non-intrusive and platform independent that can be used for any cloud environment. For rejuvenation, VM migration technique has been used.

REFERENCES

- [1] Michael Grottko and Kishor S Trivedi, "Software faults, software aging and software rejuvenation," *Journal of Reliability Engineering Association of Japan*, vol. 27, no. 7, pp. 425-438, 2005, doi: 10.11348/reaajshinrai.27.7_425
- [2] D. J. Dean, H. Nguyen, and X. Gu, "Ubl: unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems," *Proceedings of the 9th international conference on Autonomic computing*, pp. 191-200, 2012, doi: 10.1145/2371536.2371572
- [3] J. Zhao, K. S. Trivedi, and M. Grottko, *et al.*, "Ensuring the performance of Apache HTTP server affected by aging," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 2, pp. 130-141, 2014, doi: 10.1109/TDSC.2013.38
- [4] H. Meng, X. Hei, and J. Zhang, *et al.*, "Software Aging and Rejuvenation in a J2EE Application Server," *Quality and Reliability Engineering International*, vol.32, pp. 89-97 2014, doi: 10.1002/qre.1729
- [5] V. Castelli, R. E. Harper, and P. Heidelberger, *et al.*, "Proactive management of software aging," *IBM Journal of Research and Development*, vol. 45, no. 2, pp. 311-332, 2001, doi: 10.1147/rd.452.0311
- [6] K. J. Cassidy, K. C. Gross, and A. Malekpour, "Advanced pattern recognition for detection of complex software aging phenomena in online transaction processing servers," *Proceedings. International Conference on Dependable Systems and Networks*, pp. 478-482, 2002, doi: 10.1109/DSN.2002.1028933
- [7] E. Marshall, "Fatal error: how Patriot overlooked a Scud," *American Association for the Advancement of Science*, vol. 255, no. 5050, 1992.
- [8] A. Avritzer, R. G. Cole, and E. J. Weyuker, "Methods and opportunities for rejuvenation in aging distributed software systems," *Journal of Systems and Software*, vol. 83, no. 9, pp. 1568-1578, 2010, doi: 10.1016/j.jss.2010.05.026
- [9] J. Bai, X. Chang, F. Machida, K. S. Trivedi and Z. Han, "Analyzing Software Rejuvenation Techniques in a Virtualized System: Service Provider and User Views," in *IEEE Access*, vol. 8, pp. 6448-6459, 2020, doi: 10.1109/ACCESS.2019.2963397
- [10] Alonso, J., Torres, J., Berral, J., L., and Gavaldà, R., "Adaptive on-line software aging prediction based on machine learning," *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pp. 507-516, June 28 2010-July 1 2010, doi: 10.1109/DSN.2010.5544275
- [11] Ohta, S. and Hirota, T., "Machine Learning Approach to the Power Management of Server Clusters," *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*, pp. 571-578, 2011, doi: 10.1109/CIT.2011.49
- [12] Hayashi, T. and Ohta, S., "Performance Management of Virtual Machines via Passive Measurement and Machine Learning," *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*, pp. 533-538, 2012.
- [13] Matheus Torquato *et al.*, "SWARE: A Methodology for Software Aging and Rejuvenation Experiments," *Journal of Information Systems Engineering & Management*, 2018, doi: 10.20897/jisem.201815
- [14] F. Machida, J. Xiang, K. Tadano, and Y. Maeno, "Combined Server Rejuvenation in a Virtualized Data Center," *9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*, Fukuoka, pp. 486-493, 2012, doi: 10.1109/UIC-ATC.2012.52
- [15] K. Kourai and S. Chiba, "Fast Software Rejuvenation of Virtual Machine Monitors," *IEEE Transactions on Dependable and Secure Computing*, Vol. 8, no. 6, pp. 839-851, 2011, doi: 10.1109/TDSC.2010.20
- [16] H. Meng, X. Zhang, L. Zhu, L. Wang and Z. Yang, "Optimizing software rejuvenation policy based on CDM for cloud system," *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Siem Reap, pp. 1850-1854, 2017, doi: 10.1109/ICIEA.2017.8283139
- [17] M. Torquato, P. Maciel and M. Vieira, "A Model for Availability and Security Risk Evaluation for Systems With VMM Rejuvenation Enabled by VM Migration Scheduling," in *IEEE Access*, vol. 7, pp. 138315-138326, 2019, doi: 10.1109/ACCESS.2019.2943273
- [18] I. M. Umesh, G. N. Srinivasan, and Matheus Torquato, "Software Aging Forecasting Using Time Series Model," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 8, no. 3, pp. 589-596, 2017, doi: 10.11591/ijeecs.v8.i3.pp589-596
- [19] I. M. Umesh, G. N. Srinivasan, and Matheus Torquato, "Software Rejuvenation Model for Cloud Computing Platform," *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 8332-8337, 2017.
- [20] M. Torquato, J. Araujo, I. M. Umesh, and P. Maciel, "SWARE: A methodology for software aging and rejuvenation experiments," *Journal of Information Systems Engineering & Management*, vol. 3, no. 2, pp. 15-27, 2018, doi: 10.20897/jisem.201815
- [21] Umesh I. M. and Srinivasan G. N., "Dynamic software aging detection-based fault tolerant software rejuvenation model for virtualized environment," In: Satapathy S, Bhateja V, Joshi A, editors. *Proceedings of the international conference on data engineering and communication technology. Advances in intelligent systems and computing*, vol. 469. Singapore: Springer, pp. 779-787, 2017, doi: 10.1007/978-981-10-1678-3_75

- [22] Umesh, I. M and Srinivasan, G. N, "Optimum software aging prediction and rejuvenation model for virtualized environment," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 3, no. 3, pp. 572-578, 2016, doi: 10.11591/ijeecs.v3.i3.pp572-578
- [23] Torquato, M., Umesh, I. M., and Maciel, P, "Models for availability and power consumption evaluation of a private cloud with VMM rejuvenation enabled by VM live migration," *The Journal of Supercomputing*, vol. 74, no. 9, pp. 4817-4841, 2018, doi: 10.1007/s11227-018-2485-4
- [24] Torquato M, Maciel P, Araujo J, and Umesh I. M, "An approach to investigate aging symptoms and rejuvenation effectiveness on software systems," *Proceedings of 12th Iberian Conference on Information Systems and Technologies (CISTI). IEEE*, pp 1-6, 2017, doi: 10.23919/CISTI.2017.7975806
- [25] Payal Kulkarni, "Software Rejuvenation and Workload Distribution in Virtualized System," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 3, no. 6, pp. 5966-5973, 2015, doi: 10.1109/TC.2013.30

BIOGRAPHIES OF AUTHORS



Shruthi P is an Information Science and Engineering graduate. She also holds the M Tech degree in Computer Science and Engineering from VTU. She is pursuing PhD under VTU in the area of Software aging and rejuvenation. She presently works as Assistant Professor in the Department of Computer Science and Engineering, Global Academy of Technology, Bengaluru, Karnataka. She has a total of 7 years of experience in teaching and industry. She has published several papers in International journals. She is active in research and also a life member CSI society.



Dr. Nagaraj G Cholli is a Computer Science and Engineering graduate. He also holds the M Tech degree in Computer Science & Engineering from IIT-Roorkee. He obtained PhD from VTU in the year 2016 in the area of Software aging and rejuvenation. He presently works as Associate Professor at Department of Information Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka, India. He has a total of 14 years of experience in teaching, research and industry. He has published several research articles in International journals and presented papers at Conferences. He is active in research, has filed patents and guiding several PhD scholars. He is also a life member of ISTE and CSI society.