Software Quality Journal (2020) 28:3–5
https://doi.org/10.1007/s11219-020-09496-0

Guest editorial: special issue on modeling and mitigation techniques for software aging

Zheng Zheng[1] · Kishor S. Trivedi[2]

Published online: 26 February 2020

Software aging is a problem of progressive degradation of performance and dependability in computer programs, especially those executing for long period of time. This phenomenon has been extensively studied for more than 20 years, as it affects many systems, from embedded devices to server software to critical systems. The complexity of modern software systems poses significant challenges to model and design mitigation techniques for software aging.

Techniques and tools for efficiently alleviating software aging problems have been the subject of research in recent years. Software rejuvenation, known as the proactive restart of applications (components/threads/tasks) or the reboot of VMs/operating systems, is the most prominent approach to combatting software aging. A parallel track addressing problems like testing, locating, and repair has applied specifically to aging-related bugs. However, the software aging problem is far from being solved, and the relationship between methods for modeling software aging and those for mitigation techniques deserves further study.

The aim of this special issue is to gather the recent advances in modeling and mitigation techniques for software aging. The special issue presents an opportunity to explore the synergies between the modeling of software aging and the testing and debugging of programs with aging-related bugs and mitigation techniques during software development and operating.

The special issue has gone through an open call for papers and rigorous peer-review, where 9 articles from 16 submissions have been selected as representatives of ongoing research and development activities. The articles are briefly discussed as follows:

The first paper by Roberto Pietrantuono et al. on “A Survey on Software Aging and Rejuvenation in the Cloud” reviewed the effort conducted so far by the software aging and rejuvenation (SAR) community in the cloud domain. A set of 105 papers has been examined from three source digital libraries in order to have a clear view of the state of the art.

✉ Zheng Zheng
zhengz@buaa.edu.cn

Kishor S. Trivedi
kst@ee.duke.edu

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

[2] Electrical & Computer Engineering, Duke University, Durham, NC 27708-0291, USA

The second paper by Xiaoting Du et al. on “Cross-project Bug Type Prediction Based on Transfer Learning” proposed a cross-project bug type prediction framework which significantly improves the cross-project bug type prediction results. In addition, the factors that influence the prediction results were studied.

The third paper by Matheus Torquato et al. on “Availability and Reliability Modeling of VM Migration as Rejuvenation on a System under Varying Workload” proposed a set of models for availability and reliability evaluation of a virtualized system with VMM software rejuvenation enabled by VM migration scheduling.

The fourth paper by Jianwen Xiang et al. on “Software Aging and Rejuvenation in Android: New Models and Metrics” argued that the user experience with fluent UI responses should be addressed for mobile users in addition to traditional dependability metrics. It proposed proactive rejuvenation strategies by considering both aging status and usage behavior to achieve the best user experience and the least user interference.

The fifth paper by Fangyun Qin et al. on “An Empirical Study of Factors Affecting Cross-project Aging-Related Bug Prediction with TLAP” studied three factors including normalization methods, kernel functions, and machine learning classifiers that impact the prediction performance of cross-project aging-related bug prediction with TLAP and discussed the influence with single-factor, bigram, and triplet patterns.

The sixth paper by Vasilis Koutras et al. on “On the Performance of Software Rejuvenation Models with Multiple Degradation Levels” modeled software systems’ overall performance capacity by assigning a performance capacity level at each of the possible states that it can be in, using a continuous time Markov process.

The seventh paper by Junjun Zheng et al. on “A Transient Interval Reliability Analysis for Software Rejuvenation Models with Phase Expansion” focused on the phase expansion approach for solving the transient solutions for the basic software rejuvenation models. In numerical examples, the authors discussed the accuracy of the phase expansion and revealed quantitative properties of the interval reliability measures.

The eighth paper by Xiaoxue Wu et al. on “Invalid Bug Reports Complicate the Software Aging Situation” used performance bugs that are highly related to software aging as an example to construct a binary classification model for bug report classification, utilizing invalid bug reports (IBRs) to capture software aging signals.

The ninth paper by Shunkun Yang et al. on “Anti-aging Analysis for Software Reliability Design Modes in the Context of Single-event Effect” explored the anti-aging effects and rules of software reliability design modes, including triple modular redundancy (TMR) and logical partitioning. The simulation and theoretical results showed that the reliability design mode can alleviate software aging.

The special issue was preceded by the 10th International Workshop on Software Aging and Rejuvenation (October 2018 in Memphis, TN, USA). The articles have undergone rigorous peer-review according to the journal’s high standards.

Collectively, these 9 papers provide a detailed compilation of the diverse range of issues currently being investigated in the field of Software Aging and Rejuvenation. To conclude this special issue, we would like to thank all reviewers who have helped in the paper review process, and the authors for their contribution and efforts to complete the articles with high quality.

Zheng Zheng and Kishor Trivedi
December 19, 2019

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Zheng Zheng received the Ph.D. degree in computer software and theory from the Chinese Academy of Science, Beijing, China, in 2006. In 2014, he was a Research Scholar with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. He is currently a Full Professor in Control Science and Engineering with Beihang University, Beijing, China. His research interests include software dependability, unmanned aerial vehicle path planning, artificial intelligence applications, and software fault localization.

Kishor S. Trivedi (LF'17) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology Mumbai, Mumbai, India, in 1968, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1972 and 1974, respectively. He is currently the Fitzgerald Hudson Chair with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. His research interests are in reliability, availability, performance and survivability of computer and communication systems and in software dependability. Dr. Trivedi is a Golden Core Member of the IEEE Computer Society. He was the recipient of the IEEE Computer Society Technical Achievement Award.