A Comprehensive Approach to Software Aging and Rejuvenation on a Single Node Software System

P.K. Saravakos¹, G.A. Gravvanis¹, V.P. Koutras², A.N. Platis²

¹ Department of Electrical and Computer Engineering, School of Engineering, Democritus University of Thrace, 12, Vas. Sofias street, GR 67100 Xanthi, Greece; Email: ggravvan@ee.duth.gr; psaravak@ee.duth.gr
² Department of Financial and Management Engineering, Business School, University of the Aegean, 31, Fostini street, GR 82100 Chios, Greece; Email: v.koutras@fme.aegean.gr; platis@aegean.gr

ABSTRACT

The technological innovations in software technology contributed significantly in the establishment of complicated software structures and modules which gradually replaced obsolete simpler counterparts. However the ever growing use of intricate software structures revealed an alarming phenomenon stated as software aging. Software aging refers to the degradation of the operational environment of the software and has been the rationale of several software failures, which entailed an increased downtime cost and in several occasions the loss of human lives. Over the past years, researchers achieved to counteract software aging by proposing a diversity of techniques. Software rejuvenation is a proactive technique employed to counterbalance software aging by periodically restoring continuously running applications to a robust state with the aim of preventing future failures. The definite rejuvenation technique applied on a software system is been decided in the software rejuvenation policy, where the specifics of the methods applied on the system in order to counteract software aging are been determined. The most important factor in determining the rejuvenation policy consists of a tradeoff between the system’s reliability and availability.

In this paper, we examine the software aging effect on two separate insulated systems as well as the behavior of the more contemporary of the two examined systems, after the implementation of a software rejuvenation technique. The behavior of both systems, functioning under a critical workload, is been examined rooted on the overall resource consumption. The resource consumption and hence the resource availability indicates the effect of software aging on each system during its course to failure. Software rejuvenation is been incorporated in the behavioral model of one of the two systems and entails killing and restarting active processes in the user’s domain and the initiation of several maintenance procedures with the intention of restoring the system to a robust, low resource consumption state.

Our work is focused on firstly establishing a comprehensive methodology to monitor and model a software system, secondly determining an appropriate rejuvenation policy and finally developing software rejuvenation software which shall effectively realize the aforementioned policy. In particular, both systems’ behavior is been approached based on Continuous Time Markov Chains and modeled with ARIMA models. In addition to that several statistical metrics are calculated to estimate the overall performance of the system. The rejuvenation policy takes the individual characteristics of the software system examined under consideration with the aim of enhancing the overall reliability of the system. Moreover the realization of software rejuvenation entails the use of batch files for various...
tasks, from alerting resource exhaustion to instigating the maintenance procedure. Batch files have been selected with the purpose of maintaining the effect of the software rejuvenation software as discrete as possible and thus restricting the contribution of the before mentioned software to an impending failure. To conclude, this paper supplies an axis to monitor and model a software system under the effect of software aging, including a software rejuvenation technique.

REFERENCES