Inspection-Based Preventive Maintenance in Software Systems

Kalyanaraman Vaidyanathan, Dharmaraja Selvamuthu and Kishor S. Trivedi
Dept. of Electrical & Computer Engineering, Duke University
Durham, NC 27708, USA
\{kv,dharmar,kst\}@ee.duke.edu

1. Introduction

Recently, the phenomenon of “software aging”, one in which the state of the software system degrades with time, has been reported [1]. This may eventually lead to performance degradation of the software or crash/hang failure. Preventive maintenance of operational software systems is used specifically to counteract this phenomenon. However, preventive maintenance incurs an overhead in terms of downtime and this must be traded off with the downtime due to failures to obtain maximum benefits. In this paper, we present an analytical model of a software system employing inspection-based preventive maintenance, through a Markov Regenerative Process (MRGP) with a subordinated semi-Markov reward process. Furthermore, we consider preemptive-resume type transitions. The model is solved for steady state availability and the expected downtime over an interval. A numerical example is presented to illustrate the applicability of the model. With the help of this model, optimal strategies for preventive maintenance techniques such as “software rejuvenation” [1] could be formulated.

2. The System

Consider a software system that starts in a “robust” (or new) working state, $D_0$ (Figure 1). As time progresses, it transits through several exponential stages ($D_1$ through $D_k$) and eventually suffers a major failure (state $F$). The degradation rate in state $i$ is $\lambda_i$. This is a generalization of the 2-stage failure model in [1]. Each exponential stage, $D_i$, is a deterioration failure and a series of deterioration failures leads to a major failure. Thus the time to failure for the software system starting from the initial state (ignoring Poisson failures and preventive maintenance) is hypo-exponentially distributed. From the failure state $F$, a full restart (hardware reboot) is required to bring the system back to the “robust” state, $D_0$. The distribution of the time taken for the full restart is $F_R(t)$. The system can also experience Poisson failures (constant failure rate, $\lambda_p$) from any stage (states $D_0$ through $D_k$) which occur abruptly unlike the gradually worsening deterioration failures and lead the system to the corresponding states, $F_0$ through $F_k$. In a real software system this could correspond to sudden failures from causes other than software aging. A deterioration failure is a “soft” failure in which the system can still provide service (possibly at a degraded level), whereas a Poisson failure is a “hard” failure in which the system cannot do any useful work. A simple retry brings back the system from the Poisson failure to the deterioration stage from which it failed. $F_i(t)$ denotes the distribution of time to perform a simple retry. Two kinds of preventive maintenance are performed on this system. A minimal maintenance (corresponding to a partial system cleanup) from a deterioration stage restores the system to the previous deterioration stage and a major maintenance (full cleanup/reboot) from any of the deterioration stages restores to the system to the “as good as new” state. The system is inspected with the period between two inspections assumed to be generally distributed with CDF $F_I(t)$. States $I_0$ through $I_k$ denote the states where the actual inspection takes place and the time to perform the inspection is generally distributed ($F_{ins}(t)$). At the end of inspection, one of two kinds of preventive maintenance of system is carried out based on the following inspection-based policy. We assume that there are $k$ deterioration stages in the system and that the current deterioration stage, $i$, is observable through some system parameter(s). Thresholds $g$ and $b$ are established so that no maintenance is done if the inspection finds the system in state $D_i$, $i \leq g$. A minimal maintenance (CDF $F_{m}(t)$) is performed when $g < i \leq b$. In this case, the system is restored to degradation stage $i - 1$. A major maintenance (CDF $F_{M}(t)$) is performed when $b < i \leq k$, whereupon, the system is restored to the initial state $D_0$. States $m_n$ correspond to states where a minimal maintenance is being performed and states $M_n$ correspond to states where a major maintenance is being performed. The system is operational only in states $\{D_n,n = 0,1,\ldots,k\}$. We solve the MRGP model to obtain the steady state availability. We then determine the optimal value of the inspection interval

©2002, K. Vaidyanathan
which minimizes expected downtime for a given set of parameter values.

3. Results and Conclusions

We illustrate the applicability of our model through a numerical example. We consider an inspection-based maintenance model with 4 deterioration stages (i = 0, . . . , 3). No maintenance is done in stages 0 and 1. A minimal maintenance is done when the system is in deterioration stage 2 and a major maintenance is done in stage 3. Therefore, the parameters g, b and k are 1, 2 and 3 respectively (Section 2). All failure, reboot, retry and maintenance times and the time taken for actual inspection are considered exponentially distributed. The inspection interval δ, is considered deterministic. The values for parameters λi and λp are 0.1/hr and 0.05/hr respectively. The mean times for retry, full reboot, minimal maintenance, major maintenance and actual inspection are 2min, 1hr, 5min, 15min and 40sec respectively.

The steady state measure computed is expected downtime, D, over an interval [0, 1000]. Figure 2 shows the plot for expected downtime versus inspection interval. The solid line shows the plot when the inspection interval is deterministic and the dashed line shows the plot when the deterministic interval is approximated using an exponential distribution with mean equal to the deterministic interval. If the inspection interval is close to zero, the system is performing preventive maintenance very often and thus incurs high downtime. As the inspection interval increases, the expected downtime decreases and reaches an optimum value. If the inspection interval goes beyond the optimal value (i.e., we perform preventive maintenance less and less frequently), the system failure has more influence on the expected downtime than maintenance. Hence the expected downtime begins to increase. For the deterministic inspection interval case, the minimum for the expected downtime (13.6119 hours) occurs at 4.85 hours. In the case of the exponential approximation, the minimum value of downtime is 15.7475 hours which occurs when the inspection interval is 3.35 hours. Hence if we were approximating our deterministic interval using an exponential distribution, we would be incurring a higher downtime (that could be avoided) by inspecting the system every 3.35 hours.

![Figure 1. MRGP model of the system](image)

![Figure 2. Expected downtime vs insp. interval](image)

References