How the Time-Before-Failure Reacts to Periodic Rejuvenation

Shah Asaduzzaman and Muthucumaru Maheswaran
School of Computer Science, McGill University
Montreal, QC, H3A 2A7, Canada
Email: {asad,maheswar}@cs.mcgill.ca

***Abstract*— Rebooting is one of the commonly used approaches to recover from undesired crash or performance degradation in software systems. Recently, however, planned and periodic restart or rejuvenation has been proposed as a reliability management tool for avoiding unwanted failure of long-running systems. This paper presents an interesting observation that periodic rejuvenation alters the lifetime distribution of a system in a generic way irrespective of the original distribution.**

I. Introduction

Rebooting is frequently used as a last resort to get around unexplained crash or performance degradation of long-running software systems. Often, such crashes are preceded by a gradual degradation of system performance or *aging*, due to some hidden software bugs [1], [2]. If the system is restarted in a planned manner along with checkpointing of the present state instead of restarting as a reaction after the failure has occurred, the internal data-structures of the system can be re-initialized without experiencing the undesired effects of a crash. Such a planned restarting idea, called *rejuvenation*, was first formalized by Huang et al. in 1995 [3] and has been applied to several practical systems [4].

Although planned rejuvenation saves us from unpredictable losses, there are certain overheads such as cost of checkpointing and server downtime. Two approaches are generally followed to find the optimal time for rejuvenation that minimizes the overheads – *open loop* approach, where a mathematical model of the software aging process is developed to find an optimal time interval for periodic rejuvenation, and *closed loop* approach, where the actual runtime state of the software is continuously monitored by probing system resources and rejuvenation is triggered when the gradual degradation of the state goes beyond a certain threshold.

In this paper, we explore the effect of open-loop periodic rejuvenation over the probability distribution of execution lifetime of a software before is faces an unplanned failure. Several methods have been proposed to model the software execution life cycle under rejuvenation [3], [5]. However, in all of them, the execution life is divided into couple of discrete states with different probabilities of transition into a failed state. In reliability engineering, it is more common to model the system lifetime as a continuous random variable of certain distribution. We observed that periodic rejuvenation alters the running lifetime in a general way independent of the underlying probability distribution. Interestingly, the hazard rate of the rejuvenated system becomes approximately constant. The observations are generic and applicable to any system that is amenable to rejuvenation, not only software systems. In the following section, we present these observations with formal proofs and numerical results.

II. Lifetime distribution under rejuvenation

Let t be the global clock time that is initialized to 0 when the system starts for the first time. We define the *total lifetime* as the time the system passes after starting execution before any failure or unexpected crash. The time the system passes after a rejuvenation until any failure is defined as the *rejuvenated lifetime*. Let the random variable X denotes the total lifetime of the system when no rejuvenation is applied, and has a probability distribution defined by probability density function (PDF) $f(t)$, cumulative distribution function (CDF) $F(t)$, reliability or survival function $R(t)$ and hazard rate function $z(t)$ [6]. Say, the random variable X_r denotes the total lifetime of the system when a periodic rejuvenation with interval T is applied. Let $f_r(t)$, $F_r(t)$, $R_r(t)$ and $z_r(t)$ denote the PDF, CDF, reliability and hazard rate of X_r, respectively. Let $t_i (i \in Z^+)$ denotes the rejuvenated system clock time after ith rejuvenation given the system survives till then, i.e., $t_i = t - iT$. Following two theorems describe the observations on X_r.

Theorem 1: $z_r(t) = z(t_i) = z\left(t - \lfloor t/T \rfloor T\right)$

Proof: Let a random variable $X_{r(i)}$ denotes the rejuvenated lifetime of the system after ith rejuvenation. So, $X_r = t \Leftrightarrow X_{r(i)} = t_i \Leftrightarrow X_{r(i)} = t - iT$. Since rejuvenation re-initializes the system state, within the corresponding intervals $iT < t \leq (i+1)T$, all the $X_{r(i)}$'s are identically distributed as the original lifetime X. Now, by definition,

$$\lim_{\epsilon \to 0} P\{X_r \in (t, t+\epsilon) | X_r > t\} = \epsilon z_r(t) \qquad (1)$$

For any arbitrary t, w.l.o.g. we can assume that $t \in (iT, (i+1)T]$. So, $X_r > t \Rightarrow X_{r(i)} > t - iT$. Since $X_{r(i)}$'s are identically distributed as the original life time X, from Eqn. 1, it follows –

$$\begin{aligned} \epsilon z_r(t) &= P\{X_{r(i)} \in (t - iT, t - iT + \epsilon) | X_{r(i)} > t - iT\} \\ &= P\left\{X \in (t - \lfloor t/T \rfloor T, t - \lfloor t/T \rfloor T + \epsilon)\right. \\ &\qquad \left. | X > t - \lfloor t/T \rfloor T\right\} \\ &= \epsilon z\left(t - \lfloor t/T \rfloor T\right) \end{aligned}$$

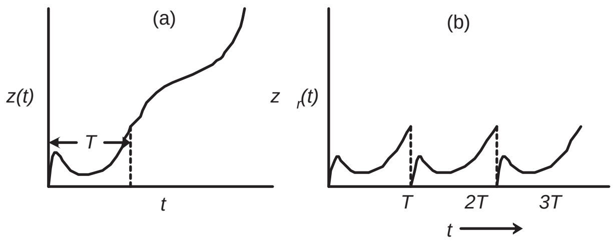

Fig. 1. Rejuvenated hazard rate

Hence, the theorem follows. Fig. 1 illustrates this effect. ■

Theorem 2: In the long run, X_r can be approximated by an exponential distribution with a constant hazard rate $\lambda = \frac{1}{T}\int_0^T z(x)dx$, regardless of the distribution of original lifetime X.

Proof: By definition, $R_r(t) = e^{-\int_0^t z_r(x)dx}$. To evaluate the integral $\int_0^t z_r(x)dx$, we observe from Fig. 1 and Theorem 1 that the area under the curve $z_r(x)$ for the interval $[0,t]$ can be expressed as –

$$\lfloor t/T \rfloor \int_0^T z(x)d(x) + \int_0^{t-T\lfloor t/T \rfloor} z(x)dx$$

When $t \gg T$, or $t - T\lfloor t/T \rfloor$ is very small compared to t, we can approximate –

$$\int_0^{t-T\lfloor t/T \rfloor} z(x)dx \approx (t - T\lfloor t/T \rfloor)\frac{1}{T}\int_0^T z(x)d(x) \qquad (2)$$

$$\Rightarrow \quad \int_0^t z_r(x)dx \approx \frac{t}{T}\int_0^T z(x)dx$$

$$\Rightarrow \quad R_r(t) \approx e^{-Kt},\ K = \frac{1}{T}\int_0^T z(x)dx, \text{ a constant}$$

So, the CDF $F_r(t)$, the PDF $f_r(t)$ and the Hazard rate $z_r(t)$ becomes –

$$\begin{aligned} F_r t &= 1 - R_r(t) \approx 1 - e^{-Kt} \\ f_r(t) &= \frac{d}{dt}F_r(t) \approx Ke^{-Kt} \\ z_r(t) &= \frac{f_r(t)}{R_r(t)} \approx K = \frac{1}{T}\int_0^T z(x)dx \end{aligned}$$

This proves X_r to be exponentially distributed, with a constant hazard rate $\lambda = \frac{1}{T}\int_0^T z(x)dx$. The mean time before failure (MTTF) $= E(X_r) \approx \frac{1}{K}$ ■

To further visualize the effects of rejuvenation, we applied periodic rejuvenation at 100hr intervals to aging systems having MTTF of 1000hrs and certain lifetime distributions (exponential and Weibull). We gathered statistics on lifetime of the system over 10000 runs. When the original lifetime distribution is exponential, the lifetime after rejuvenation has almost the same distribution (Fig. 2a,b). When the original lifetime has Weibull distribution, with scale parameter $\alpha = 1128$, shape parameter $\gamma = 2$ and MTTF$= 1000$hrs and the system is rejuvenated at the same intervals, the new lifetime becomes exponential (Fig. 2c), the hazard rate becomes constant (Fig. 2d) and the MTTF becomes 12441hrs, approximately. When the shape parameter γ of the Weibull distribution for original lifetime is set to 0.5, the original hazard rate becomes decreasing function of time. If we apply the periodic rejuvenation on this system, the resultant lifetime becomes exponential with constant hazard rate, and the MTTF actually decreases from 1000hrs to 206hrs. All these simulation results approximately matches the results predicted by the theorems. In fact, the exact density and hazard rate functions oscillate along with the rejuvenation, and the approximation in Eqn. 2 smoothes out the oscillation and gives an average value of the functions. Moreover, the MTTF results comply with the intuition that periodic rejuvenation is beneficial only when the hazard rate of the original system is an increasing function of time.

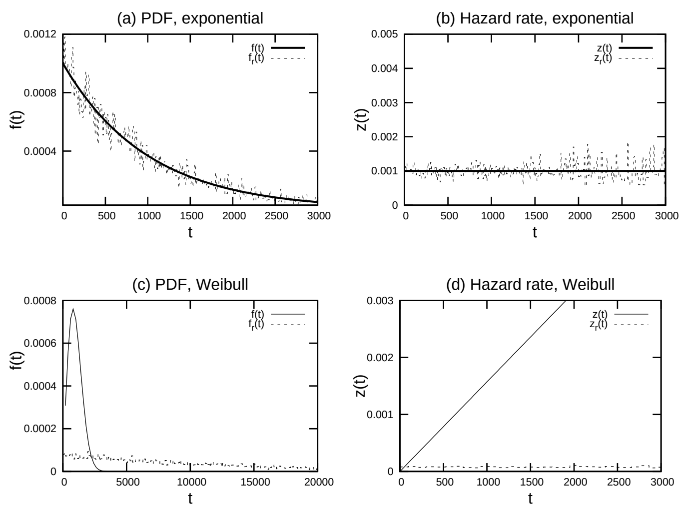

Fig. 2. Simulation results

III. Conclusion

The analysis presented in this paper will be useful in development of stochastic models of aging systems under rejuvenation. To keep our analyses simple, we did not consider the rejuvenation overhead. However, this does not affect the main conclusion that periodic rejuvenation modulates the lifetime distribution into an exponential one, irrespective of the original lifetime distribution.

References

[1] S. Garg, A. van Moorsel, K. Vaidyanathan, and K. S. Trivedi, “A methodology for detection and estimation of software aging,” in *proc. 9th ISSRE*, Nov. 1998, pp. 283–292.

[2] M. Grottke, L. Li, K. Vaidyanathan, and K. S. Trivedi, “Analysis of software aging in a web server,” *IEEE Trans. Reliability*, vol. 55, no. 3, pp. 411–420, 2006.

[3] Y. Huang, C. Kintala, N. Kolettis, and N. Fulton, “Software rejuvenation: analysis, module and applications,” in *proc. FTCS-25*, Jun. 1995, pp. 381–390.

[4] E. Marshall, “Fatal error: how patriot overlooked a scud,” *Science*, vol. 255, p. 1347, 1992.

[5] K. Vaidyanathan and K. S. Trivedi, “A comprehensive model for software rejuvenation,” *IEEE Trans. Dependable and Secure Comp.*, vol. 2, no. 2, pp. 124–137, 2005.

[6] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, 2nd ed. John Wiley and Sons, 2001.