Reliability Measures of a Computer System with Different Repair and Redundant Policies for Components

V.J. Munday* and S.C. Malik

Department of Statistics, M.D. University, Rohtak-124001 (India)

Email: vjmunday@rediffmail.com and sc_malik@rediffmail.com

Received on 15 Jan., 2014: Revised on 25 March, 2014: Accepted on 30 April, 2014

*Corresponding Author

Abstract

Reliability measures of a computer system with independent constant failure of hardware and software components have been evaluated by using semi-Markov process and regenerative point technique. A single server is provided immediately to repair the system at hardware failure while software is up-graded whenever it fails to meet out the desired functions properly. Hardware repair and software up-gradation are perfect. The random variables are statistically independent. The failure times of hardware and software components follow negative exponential distribution whereas the distributions for their repair and up-gradation times are taken as arbitrary with different probability density functions. The graphical behaviour of some measures of system effectiveness has been observed for arbitrary values of various parameters and costs. The profit of the present model has also been compared with the models developed under component wise redundancy.

Keywords: Computer System, Independent Failure, Reliability Measures and Profit Comparison.

1. Introduction

The demand of computer systems has been increasing day by day in most of the business sectors due to their ability to finish the jobs timely and efficiently. In spite of this increasing demand, a little work has been done to the assessment of reliability and economic measures of computer systems with independent hardware and software failures. However, scientists and engineers are trying to develop such computer systems that can provide better services to the users for a reasonable period. And, they have achieved it in some cases by adopting better repair policies and redundancy techniques. In the past few decades, the technique of redundancy for the whole system has been used by the researchers including Cao and Wu [3], Lam [7] and Yadavalli et al. [6]. Also, Malik and Anand [2, 4] and Kumar et al. [1] developed reliability models for the computer systems with cold standby redundancy for the
whole system. Recently, Malik and Munday [5] analyzed a computer system by providing hardware redundancy in cold standby. In most of these studies, the utility of redundancy and repair strategies has not been examined properly.

The purpose of present study is not only to obtain reliability measures of a computer system but also to make comparison of its profit with that of the computer system models developed under component wise redundancy. Initially, reliability measures for a computer system model are obtained by considering independent hardware and software failure with some probabilities. A single server is provided immediately to repair the system at hardware failure while software is up-graded when it fails to meet out the desired functions properly. Hardware repair and software up-gradation are perfect. The random variables are assumed as statistically independent. The switch devices are fault free. The failure time of the hardware and software components follow negative exponential distribution whereas the distributions for hardware repair and software up-gradation times are taken as arbitrary with different probability density functions. The results for some measures of system effectiveness such as transition probabilities, mean sojourn times, mean time to system failure (MTSF), availability, busy period of the server due to hardware repair and software up-gradation, expected number of hardware repairs and expected number of software up-gradations are evaluated in steady state by using semi-Markov process and regenerative point technique. Graphs are drawn to depict the behaviour of some measures of system effectiveness for arbitrary values of various parameters and costs. The profit of the present model has also been compared with the models developed under component wise redundancy.

2. Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>Set of regenerative states</td>
</tr>
<tr>
<td>$\bar{E}$</td>
<td>Set of non-regenerative states</td>
</tr>
<tr>
<td>$O$</td>
<td>Computer system is operative</td>
</tr>
<tr>
<td>$a/b$</td>
<td>Probability that the system has hardware / software failure</td>
</tr>
<tr>
<td>$\lambda_1$ (\lambda_2)</td>
<td>The constant hardware/software failure rate</td>
</tr>
<tr>
<td>$P_1$</td>
<td>Profit of the system model with hardware redundancy in cold standby</td>
</tr>
<tr>
<td>$P_2$</td>
<td>Profit of the system model with software redundancy in cold standby</td>
</tr>
<tr>
<td>HFUr</td>
<td>The hardware is failed and under repair</td>
</tr>
<tr>
<td>SFUg</td>
<td>The software is failed and under up-gradation</td>
</tr>
</tbody>
</table>
\[ g(t)/G(t) : \text{pdf/cdf of hardware repair time} \]
\[ f(t)/F(t) : \text{pdf/cdf of software up-gradation time} \]

\[ q_{ij}(t) / Q_{ij}(t) : \text{pdf / cdf of first passage time from regenerative state } S_i \text{ to a regenerative state } S_j \text{ or to a failed state } S_j \text{ without visiting any other regenerative state in } (0, t] \]

\[ M_i(t) : \text{Probability that the system up initially in state } S_i \in E \text{ is up at time } t \text{ without visiting to any regenerative state} \]

\[ W_i(t) : \text{Probability that the server is busy in the state } S_i \text{ up to time ‘}t\text{’ without making any transition to any other regenerative state or returning to the same state via one or more non-regenerative states.} \]

\[ \mu_i : \text{The mean sojourn time in state } S_i \text{ which is given by} \]
\[ \mu_i = E(T) = \int_0^\infty P(T > t) \, dt = \sum_j m_{ij}, \]
where \( T \) denotes the time to system failure.

\[ m_{ij} : \text{Contribution to mean sojourn time (} \mu_i \text{) in state } S_i \text{ when system transits directly to state } S_j \text{ so that} \]
\[ \mu_i = \sum_j m_{ij} \text{ and } m_{ij} = \int_0^\infty tdQ_{ij}(t) = -q_{ij}^{-1}(0) \]

\& /© : Symbol for Laplace-Stieltjes convolution/Laplace convolution

\*/** : Symbol for Laplace Transformation (LT)/Laplace Stieltjes Transformation (LST)

The transition between different states of the system model has been shown in Fig. 1.

**State Transition Diagram**

Fig. 1
3. Transition Probabilities and Mean Sojourn Times

Simple probabilistic considerations yield the following expressions for the non-zero elements:

\[ p_{ij} = Q_{ij}(\infty) = \int_0^\infty q_{ij}(t)dt \]

\[ p_{01} = \frac{a\lambda_1}{a\lambda_1 + b\lambda_2}, \quad p_{02} = \frac{b\lambda_2}{a\lambda_1 + b\lambda_2} \]

\[ p_{10} = g^*(0) \quad p_{20} = f^*(0) \]

For \( g(t) = ae^{-\alpha t} \) and \( f(t) = \theta e^{-\theta t} \) we have

But, \( f^*(0) = g^*(0) = 1 \) and \( p + q = 1 \)

It can be easily verified that \( p_{01} + p_{02} = p_{10} = p_{20} = 1 \)

The mean sojourn times \( (\mu_i) \) is the state \( S_i \) are

\[ \mu_0 = \frac{1}{a\lambda_1 + b\lambda_2}, \quad \mu_1 = \frac{1}{\alpha}, \quad \mu_2 = \frac{1}{\theta} \]

Also \( m_{01} + m_{02} = \mu_0, \quad m_{10} = \mu_1, \quad m_{20} = \mu_2 \)

4. Reliability and Mean Time To System Failure (MTSF)

Let \( \phi(t) \) be the cdf of first passage time from regenerative state \( S_i \) to a failed state.

Regarding the failed state as absorbing state, we have the following recursive relations for \( \phi(t) \):

\[ \phi_0 = Q_{01}(t) + Q_{02}(t) \]

Taking LST of above relation (1) and solving for \( \phi_0^*(s) \).

We have

\[ R^*(s) = \frac{1 - \phi_0^*(s)}{s} \]

The reliability of the system model can be obtained by taking Laplace inverse transform of the above equation.

The mean time to system failure (MTSF) is given by

\[ MTSF = \lim_{s \to 0} \frac{1 - \phi_0^*(s)}{s} = \mu_0 \]
5. Steady State Availability

Let \( A_i(t) \) be the probability that the system is in up-state at an instant ‘t’ given that the system entered regenerative state \( S_i \) at t=0. The recursive relations for \( A_i(t) \) are given as:

\[
\begin{align*}
A_0(t) &= M_0(t) + q_{01}(t) \odot A_1(t) + q_{02}(t) \odot A_2(t) \\
A_i(t) &= q_{10}(t) \odot A_0(t) \\
A_2(t) &= q_{20}(t) \odot A_0(t)
\end{align*}
\]  

(3)

Where

\[
M_0(t) = e^{-(a_4+b_4)t}
\]

Taking LT of relations (3) and solving for \( A_0^*(s) \), the steady state availability is given by

\[
A_0(\infty) = \lim_{s \to 0} sA_0^*(s) = \frac{N_2}{D_2}
\]

(4)

Where \( N_2 = \mu_0 \) and \( D_2 = \mu_0 + p_{01}\mu_1 + p_{02}\mu_2 \)  

(5)

6. Busy Period of the Server

(a). Due to Hardware Repair

Let \( B_i^H(t) \) be the probability that the server is busy in repairing the unit due to hardware failure at an instant ‘t’ given that the system entered state \( S_i \) at t=0. The recursive relations for \( B_i^H(t) \) are as follows:

\[
\begin{align*}
B_0^H(t) &= q_{01}(t) \odot B_1^H(t) + q_{02}(t) \odot B_2^H(t) \\
B_1^H(t) &= W_1^H(t) + q_{10}(t) \odot B_0^H(t) \\
B_2^H(t) &= q_{20}(t) \odot B_0^H(t)
\end{align*}
\]

(6)

where

\[
W_1^H(t) = \overline{G(t)}dt
\]

(b). Due to software Up-gradation

Let \( B_i^S(t) \) be the probability that the server is busy due to up-gradation of the software at an instant ‘t’ given that the system entered the regenerative state \( S_i \) at t=0. We have the following recursive relations for \( B_i^S(t) \):

\[
\begin{align*}
B_0^S(t) &= q_{01}(t) \odot B_1^S(t) + q_{02}(t) \odot B_2^S(t) \\
B_1^S(t) &= q_{10}(t) \odot B_0^S(t)
\end{align*}
\]
\[B_t^S(t) = W_t^S(t) + q_{20}(t) \square B_0^S(t) \quad (7)\]

where
\[W_t^S(t) = \overline{F(t)} dt\]

Taking LT of relations (6) & (7), solving for \(B_0^{H^*}(s)\) and \(B_0^{S^*}(s)\). The time for which server is busy due to repair and up-gradation respectively are given by
\[B_0^H = \lim_{s \to 0} sB_0^{H^*}(s) = \frac{N_3^H}{D_2} \quad (8)\]
\[B_0^S = \lim_{s \to 0} sB_0^{S^*}(s) = \frac{N_3^S}{D_2} \quad (9)\]

Where
\[N_3^H = p_0 W_1^{H^*}(0), \quad N_3^S = p_{02} W_1^{S^*}(0)\]

and \(D_2\) is already mentioned.

7. Expected Number of Hardware Repairs

Let \(NHR_i(t)\) be the expected number of hardware repairs by the server in \((0, t]\) given that the system entered the regenerative state \(S_i\) at \(t=0\). The recursive relations for \(NHR_i(t)\) are given as:
\[NHR_0(t) = Q_{01}(t) + [1 + NHR_1(t)] + Q_{02}(t) & NHR_2(t)\]
\[NHR_1(t) = Q_{10}(t) & NHR_0(t)\]
\[NHR_2(t) = Q_{20}(t) & NHR_0(t) \quad (11)\]

Taking LST of relations (11) and solving for \(NHR_0^{**}(s)\). The expected number of hardware repairs are given by
\[NHR_0(\infty) = \lim_{s \to 0} sNHR_0^{**}(s) = \frac{N_4}{D_2} \quad (12)\]

Where
\[N_4 = p_{01}\] and \(D_2\) is already mentioned.

8. Expected Number of Software Up-gradations

Let \(NSU_i(t)\) be the expected number of software up-gradations in \((0, t]\) given that the system entered the regenerative state \(S_i\) at \(t=0\). The recursive relations for \(NSU_i(t)\) are given as follows:
\[NSU_0(t) = Q_{01}(t) & NSU_1(t) + Q_{02}(t) & \{1 + NSU_2(t)\}\]
\[NSU_1(t) = Q_{10}(t) & NSU_0(t)\]
\[ NSU_2(t) = Q_{20}(t) \& NSU_0(t) \]  

Taking LST of relations (14) and solving for \( NSU^*_0(s) \). The expected number of software up-gradations are given by

\[ NSU_0(\infty) = \lim_{s \to 0} sNSU^*_0(s) = \frac{N_5}{D_2} \]  

Where

\[ N_5 = p_{02} \text{ and } D_2 \text{ is already mentioned.} \]  

9. **Profit Analysis**

The profit incurred to the system model in steady state can be obtained as:

\[ P = K_0A_0 - K_1B_0^H - K_2B_0^S - K_3NHR_0 - K_4NSU_0 \]  

Where

\[ K_0 = \text{Revenue per unit up-time of the system} \]

\[ K_1 = \text{Cost per unit time for which server is busy due to hardware repair} \]

\[ K_2 = \text{Cost per unit time for which server is busy due to software up-gradation} \]

\[ K_3 = \text{Cost per unit repair of the failed hardware} \]

\[ K_4 = \text{Cost per unit up-gradation of the failed software} \]

and \( A_0, B_0^H, B_0^S, NHR_0, NSU_0 \) are already defined.

10. **Particular Case**

Suppose \( g(t) = ae^{-at} \) and \( f(t) = \theta e^{-\theta t} \)

We can obtain the following results:

\[ MTSF(T_0) = \mu_0 \]

\[ Availability(A_0) = \frac{N_2}{D_2} \]

\[ Busy \ period \ due \ to \ hardware \ failure \ (B_0^H) = \frac{N_3^H}{D_2} \]

\[ Busy \ period \ due \ to \ software \ failure \ (B_0^S) = \frac{N_3^S}{D_2} \]

\[ Expected \ number \ of \ repair \ at \ hardware \ failure (NHR_0) = \frac{N_4}{D_2} \]

\[ Expected \ number \ of \ up-gradation \ at \ software \ failure (NSU_0) = \frac{N_5}{D_2} \]

Where
\[ N_2 = \frac{1}{a\lambda_1 + b\lambda_2}, \quad D_2 = \frac{\alpha\theta + \theta a\lambda_1 + ab\lambda_2}{a\theta(a\lambda_1 + b\lambda_2)}, \quad N_3^H = \frac{a\lambda_1}{\alpha(a\lambda_1 + b\lambda_2)} \]

\[ N_3^S = \frac{b\lambda_2}{\theta(a\lambda_1 + b\lambda_2)}, \quad N_4 = \frac{a\lambda_1}{a\lambda_1 + b\lambda_2}, \quad N_5 = \frac{b\lambda_2}{a\lambda_1 + b\lambda_2} \]

11. Conclusion

The graphical behaviour of some reliability measures of the system model has been observed for a particular case as shown in figures 2, 3 and 4. It is observed that mean time to system failure (MTSF), availability and profit function keep on decreasing when hardware and software failure rates \((\lambda_1, \lambda_2)\) increase while their values increase with the increase of hardware repair rate \((\alpha)\) and software up-gradation rate \((\theta)\). Thus a computer system can be made more available by using components of low failure rates instead of increasing the repair and up-gradation rates at their failures.

The profit of the present model has also been compared with the models developed for a computer system by giving cold standby redundancy to hardware \([5]\) and software components separately. It is concluded that the idea of hardware and software redundancy in cold standby is useful in making the system more profitable provided the system has more chances of hardware failure. However, a computer system with hardware cold standby redundancy is more profitable as compared to that of having software cold standby redundancy. It is interesting to note that the idea of software cold standby redundancy in a computer system would not be profitable if system has more chances of software failure. The comparison of profits of the system models are also shown in figures 5 and 6.

12. References


Profit Vs Hardware Failure Rate ($\lambda_1$)

$K_1 = 15000, K_2 = 1000, K_3 = 700, K_4 = 1500, K_5 = 1200$

P1-P Vs Hardware Failure Rate ($\lambda_1$)

P2 - P Vs Hardware Failure Rate ($\lambda_1$)

Fig. 4

Fig. 5

Fig. 6