Correlation in Multiversion Software

by

Toke Jayachandran

October 1996

Approved for public release; distribution is unlimited.

Prepared for: Naval Postgraduate School
Monterey, CA 93943-5000
This report was prepared in conjunction with research conducted for the Naval Postgraduate School and funded by the Naval Postgraduate School.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Toke Jayachandran
Professor, Department of Mathematics

Reviewed by:

WALTER M. WOODS
Chairman

Released by:

DAVID W. NETZER
Dean of Research
**Title:** Correlation in Multiversion Software

**Author:** Toke Jayachandran

**Performing Organization:** Naval Postgraduate School, Monterey, CA 93943-5000

**Abstract:**

It has been established both theoretically [1] and experimentally [2], that independently developed redundant software versions fail dependently. Several probability models that account for this phenomenon of concurrent failures have appeared in the literature. Tomek et al., [3] proposed an intensity distribution that introduced a specific type of correlated failure pattern viz., pairwise correlation between software modules. They derived the intensity pmf for \( N = 2 \) and 3 modules and indicated the desirability of an efficient algorithm to compute the pmf for larger values of \( N \). This paper contains an easily programmable algorithm to generate the pmf for any choice of \( N \).
Correlation in Multiversion Software

Toke Jayachandran
Code MA/Jy
Naval Postgraduate School
Monterey, CA 93943

September, 1996

ABSTRACT

It has been established both theoretically [1] and experimentally [2], that independently developed redundant software versions fail dependently. Several probability models that account for this phenomenon of concurrent failures have appeared in the literature. Tomek et al., [3] proposed an intensity distribution that introduced a specific type of correlated failure pattern viz., pairwise correlation between software modules. They derived the intensity pmf for N = 2 and 3 modules and indicated the desirability of an efficient algorithm to compute the pmf for larger values of N. This paper contains an easily programmable algorithm to generate the pmf for any choice of N.

1 INTRODUCTION

The two principal techniques for software redundancy are N-version programming [4] and the recovery blocks [5]. Both require multiple independently developed software versions to achieve high reliability in software systems. Initially, it was believed that failures in independently produced software occur dependently; and system reliability computations were based on this premise. It was subsequently demonstrated, both theoretically [1] and experimentally [2] that multiple versions can fail simultaneously for some choices of inputs. As a result, reliability estimates assuming independent failures can be overly optimistic. Several papers introducing probability models that allow for concurrent failures have appeared in the literature recently. Nicola and Goyal [6] proposed a model for simultaneous failure of independent software modules and the model has been shown to provide a good fit to the experimental data in [2]. Tomek et al., [3] introduced another model for generating the probability distribution (intensity distribution) of the number of modules (in an N-version system) that fail concurrently for a randomly selected input. The latter model incorporates the correlated failure syndrome into the intensity pmf through a parameter K that represents the probability that a pair of modules will produce identical outputs. They derived explicit
expressions for the pmf for N=2 and N=3 module software systems, and suggested that an efficient algorithm is needed to derive the pmf for larger values of N. This paper presents such an algorithm for generating the intensity pmf for different choices of the parameters N and K. The algorithm is easily programmable and is particularly suited for use with symbolic computation packages such as MAPLE\(^\text{\textcopyright 1}\).

The Tomek et al., [3] model for correlated failure is described in Section II and the algorithm for deriving the intensity pmf for chosen values of N and K is presented in Section III. A MAPLE program for generating the pmf and the output of the program for N=5 and K=.1 are included in the Appendix.

\section{A PROBABILITY MODEL FOR CORRELATED FAILURES}

Consider a redundant software system with N independently developed modules. Let \(\Theta_N(X)\) be the proportion of modules (out of N) that fail (produce an incorrect output) for a randomly chosen input X. Then \(\Theta_N(X)\) is a random variable assuming the values \(\{0, 1/N, 2/N, \ldots, 1\}\). \(\Theta_N(X)\) is called the intensity function and its probability distribution is referred to as the intensity distribution. For their probability based correlated failures model, Tomek et al. [3] assume that for each pair of modules, a proportion K of all possible inputs, will always generate identical outputs for the two modules. It is possible for two different pairs of modules to have identical inputs on two different sets of inputs, albeit the proportion of such inputs K is the same for all pairs. It is further assumed that a module will produce an incorrect output with probability \(p\). For N=2 modules, the space of all possible inputs is comprised of two subsets \(R\) and its complement \(R'\). \(R\) is the set of inputs for which the two modules will produce identical results, and for inputs from \(R'\) the module outputs are independent. The intensity function \(\Theta_2(X)\) assumes the values 0, 1/2, 1 and

\[
\begin{align*}
\Pr[\Theta_2(X) = 0] &= \Pr[X \in R].\Pr[\text{both module outputs are correct}|X \in R] \\
&\quad + \Pr[X \in R'].\Pr[\text{both module outputs are correct}|X \in R'] \\
&= K(1-p) + (1-K)(1-p)^2;
\end{align*}
\]

\[
\begin{align*}
\Pr[\Theta_2(X) = 1/2] &= \Pr[X \in R'].\Pr[\text{exactly one output is correct}|X \in R'] \\
&= 2(1-K)p(1-p);
\end{align*}
\]

\[
\begin{align*}
\Pr[\Theta_2(X) = 1] &= \Pr[X \in R].\Pr[\text{both module outputs are correct}|X \in R] \\
&\quad + \Pr[X \in R'].\Pr[\text{both module outputs are correct}|X \in R'] \\
&= Kp + (1-K)p^2;
\end{align*}
\]

\(^1\)MAPLE is a registered trademark of Waterloo Maple Software
In the case of $N = 3$ modules, the input space is partitioned into 3 types of subsets $R_1$, $R_2$, and $R_3$ where $R_i$ ($i = 2, 3$) is the set of inputs for which exactly $i$ modules will produce identical results; $R_1$ is the set of inputs for which the module outputs are independent. There will be three subsets of the type $R_2$ and just one subset each of the types $R_1$ and $R_3$. The probabilities for the selection of an input from these subsets are $K^2$ for $R_3$, $3K(1 - K)$ for $R_2$ and $1 - K^2 - 3K(1 - K) = (1 - K)(1 - 2K)$ for $R_1$ and

$$Pr[\Theta_3(X) = j/3] = \sum_i Pr[X \in R_i].Pr[\text{exactly } j \text{ outputs are correct}|X \in R_i] \quad j = 0 \ldots 3.$$ 

Therefore

$$\begin{align*}
Pr[\Theta_3(X) = 0] &= K^2(1 - p) + 3K(1 - K)(1 - p)^2 + (1 - K)(1 - 2K)(1 - p)^3; \\
Pr[\Theta_3(X) = 1/3] &= K^2.0 + 3K(1 - K)p(1 - p)^2 + (1 - K)(1 - 2K)p(1 - p)^2; \\
Pr[\Theta_3(X) = 2/3] &= K^2.0 + 3K(1 - K)p(1 - p) + (1 - K)(1 - 2K)p^2(1 - p); \\
\end{align*}$$

(2)

The calculation of the probabilities of selecting an input from the subsets partitioning the input space, and the conditional pmf of $\Theta_N(X)$ becomes increasingly more difficult as the number of modules $N$ increases. An efficient algorithm that will perform the needed book keeping in a systematic fashion is presented in the next section.

3 AN ALGORITHM FOR GENERATING THE INTENSITY DISTRIBUTION

For an $N$ module software system, the input space is partitioned in $N$ types of subsets $R_i$, $i = 1, 2, \ldots, N$. Inputs from subset type $R_i$ will result in identical outputs from $i$ of the $N$ modules. The number of subsets of type $R_i$, except for type $R_1$, is equal to $\binom{N}{i}$ the number of different ways of selecting $i$ modules from the available $N$ modules. There is just one subset of type $R_1$ and the module outputs are independent for inputs from this subset. The table below illustrates the pattern for the conditional probabilities $Pr[\Theta_N(X) = j/N|X \in R_i]$ when $N = 5$. 

3
The probability entries in the table constitute a $5 \times 6$ matrix $P$ which can be expressed as the sum $A + B$ of the two triangular matrices

$$A = \begin{bmatrix}
(1-p) & 0 & 0 & 0 & 0 & 0 \\
(1-p)^2 & p(1-p) & 0 & 0 & 0 & 0 \\
(1-p)^3 & 2p(1-p)^2 & p^2(1-p) & 0 & 0 & 0 \\
(1-p)^4 & 3p(1-p)^3 & 3p^2(1-p)^2 & p^3(1-p) & 0 & 0 \\
(1-p)^5 & 5p(1-p)^4 & 10p^2(1-p)^3 & 10p^3(1-p)^2 & 5p^4(1-p) & p^5
\end{bmatrix}$$

and

$$B = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & p \\
0 & 0 & 0 & 0 & p(1-p) & p^2 \\
0 & 0 & 0 & p(1-p)^2 & 2p^2(1-p) & p^3 \\
0 & 0 & p(1-p)^3 & 3p^2(1-p)^2 & 3p^3(1-p) & p^4 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

The above pattern persists for all $N$ and the two $N \times N + 1$ matrices, in the general case, have the form $A = (a_{ij})$ and $B = (b_{ij})$ where
The entries in the matrix $P = (a_{ij} + b_{ij})$ are the conditional probabilities $Pr[\Theta_N = (j-1)/N | X \in R_i], i = 1, \ldots, N$ and $j = 1, \ldots, N + 1$. The unconditional probabilities or the intensity distribution is obtained by multiplying the matrix $P$ on the left by the 1-row matrix $Q = [q_1, q_2, \ldots, q_N]$ where

$$q_i = \binom{N}{i} K^{N-i-1}(1-K)^i \quad \text{for} \quad i = 1, 2, \ldots, N-1$$

$$q_N = 1 - \sum_i q_i$$

Note that $Q$ is just the vector of probabilities for an input to be in each of the subset types $R_N, R_{N-1}, \ldots, R_1$.

The algorithm for computing the intensity pmf of $\Theta_N(X)$ can be described by the following 3-step process. For specified values of the parameters $N$ and $K$

1. Determine the 1-row matrix $Q$ of input probabilities.

2. Evaluate the matrix $P = A + B$, the matrix of conditional probabilities, $Pr[\Theta_N = (j-1)/N | X \in R_i]$.

3. Compute the matrix product $Q \times P$ which is a $1 \times N + 1$ matrix to obtain the intensity pmf of $\Theta_N(X)$. 
APPENDIX

Three MAPLE procedures \( r(N, K) \), \( Q(N, K) \) and \( P(N, K) \) to generate \( q_N \) the matrix \( Q \) in (6) and (7) and the matrix \( P = A + B \) in (3) are shown below. A printout of the MAPLE output creating these procedures and the computational results for \( N = 5 \) and \( K = .1 \) is also included.

\[
\begin{align*}
\text{r: } &= (N, K) \rightarrow 1 - \sum ((\text{binomial}(n, i)) \cdot (K^{N-1-i}) \cdot (1 - K)^i), \; i = 0..N-2) \\
\text{Q: } &= (N, K) \rightarrow \text{array} \left( \left[ \left( \text{seq}(\text{binomial}(N, i) \cdot K^{N-1-i} \cdot (1 - K)^i, \; i = 0..n-2) \right), \; r(n,k) \right] \right) \\
\text{P: } &= \text{proc}(N, K) \text{ local } A, B, C, s, t; \\
&\text{A: } = \text{array}(1..N, 1..N+1): \\
&\text{for } s \text{ to } N \text{ do} \\
&\text{for } t \text{ to } N + 1 \text{ do} \\
&\text{if } s < N \text{ then} \\
&\text{if } s > t - 1 \text{ then} \\
&\text{A}[s,t]: = \left( \text{binomial}(s - 1, t - 1) \cdot (K^{t-1}) \cdot (1 - K)^{(s - 1 - t)} \right) \\
&\text{else } A[s,t]: = 0 \text{ fi;} \\
&\text{else } A[s,t]: = \left( \text{binomial}(n, t-1) \cdot (K^{t-1}) \cdot (1 - K)^{(N-t-1)} \right); \; \text{fi; od;} \\
&\text{fi; od;} \\
&\text{B: } = \text{array}(1..N, 1..N + 1): \\
&\text{for } s \text{ to } N \text{ do} \\
&\text{for } t \text{ to } N + 1 \text{ do} \\
&\text{if } s < N \text{ then} \\
&\text{if } t > N - s + 1 \text{ then} \\
&\text{B}[s,t]: = \left( \text{binomial}(s - 1, N+1 - t) \cdot K^{s+t-N-1} \cdot ((1 - K)^{(N+1-t)}) \right) \\
&\text{else } B[s,t]: = 0 \text{ fi;} \\
&\text{else } B[s,t]: = 0 \text{ fi; od;} \\
&\text{od;} \\
&\text{C: } = \text{evalm} (A + B); \\
&\text{end;} \\
\end{align*}
\]

Finally, the MAPLE expression

\[
\text{evalm(} Q(N, K) \&* P(N, K)) \]

will display the desired intensity pmf.
\[ r := (N, K) \rightarrow 1 - \sum_{i=0}^{N-2} \text{binomial}(N, i) K^{(N-1-i)} (1-K)^i \]

\[
Q := (N, K) \rightarrow \text{array}([\text{seq}(\text{binomial}(N, i) K^{(N-1-i)} (1-K)^i, i=0..N-2), r(N, K))]\]

\[ P := \text{proc}(N, K) \text{ local } A, B, C, s, t; \]
\[ A := \text{array}(1..N, 1..N+1); \]
\[ \text{for } s \text{ to } N \text{ do } \]
\[ \text{if } s < N \text{ then } \]
\[ \text{A}[s, t] := (\text{binomial}(s-1, t-1)) K^{(t-1)} (1-K)^{s-t+1}) \]
\[ \text{else } A[s, t] := 0 \text{ end if;} \]
\[ \text{else A}[s, t] := (\text{binomial}(N, t-1)) K^{(t-1)} (1-K)^{N-t+1}); \text{ fi; od}; \]
\[ \text{od;} \]
\[ B := \text{array}(1..N, 1..N+1); \]
\[ \text{for } s \text{ to } N \text{ do } \]
\[ \text{if } s < N \text{ then } \]
\[ \text{if } t < s \text{ then } \]
\[ A[s, t] := \text{binomial}(s-1, t-1) K^{(t-1)} (1-K)^{(s-t+1)} \]
\[ \text{else A}[s, t] := 0 \text{ end if;} \]
\[ \text{else A}[s, t] := \text{binomial}(N, t-1) K^{(t-1)} (1-K)^{N-t+1}) \]
\[ \text{fi} \]
\[ \text{fi} \]
\[ \text{od;} \]
\[ C := \text{evalm}(A+B); \]
\[ \text{end;} \]

\[ P := \text{proc}(N, K) \text{ local } A, B, C, s, t; \]
\[ A := \text{array}(1..N, 1..N+1); \]
\[ \text{for } s \text{ to } N \text{ do } \]
\[ \text{for } t \text{ to } N+1 \text{ do } \]
\[ \text{if } s < N \text{ then } \]
\[ \text{if } t-1 < s \text{ then } \]
\[ A[s, t] := \text{binomial}(s-1, t-1) K^{(t-1)} (1-K)^{(s-t+1)} \]
\[ \text{else A}[s, t] := 0 \text{ end if;} \]
\[ \text{else A}[s, t] := \text{binomial}(N, t-1) K^{(t-1)} (1-K)^{N-t+1}) \]
\[ \text{fi} \]
\[ \text{fi} \]
\[ \text{od;} \]
\[ B := \text{array}(1..N, 1..N+1); \]
\[ \text{for } s \text{ to } N \text{ do } \]
\[ \text{for } t \text{ to } N+1 \text{ do } \]
\[ \text{if } s < N \text{ then } \]
\[ \text{if } N-s+1 < t \text{ then } B[s, t] := \]
\[ \text{binomial}(s-1, N-t+1) K^{(s+t-N-1)} (1-K)^{N-t+1}) \]
\[ \text{else B}[s, t] := 0 \text{ end if;} \]
\[ \text{fi} \]
\[ \text{fi} \]
else \( B[s, t] := 0 \)
fi

do
  od;
C := evalm(A + B)
end

> r(5,.1);

\[ .1854000000 \]

> Q(5,.1);

\[ [.0001, .0045, .0810, .7290, .1854000000] \]

> P(5,.1);

\[
\begin{bmatrix}
.9 & 0 & 0 & 0 & 0 & .1 \\
.81 & .09 & 0 & 0 & .09 & .01 \\
.729 & .162 & .009 & .081 & .018 & .001 \\
.6561 & .2187 & .0972 & .0252 & .0027 & .0001 \\
.59049 & .32805 & .07290 & .00810 & .00045 & .00001 \\
\end{bmatrix}
\]

> evalm(Q(5,.1) &* P(5,.1));

\[
\begin{bmatrix}
.6505577460, .2337797700, .0851034600, .02643354000, .003914730000, .0002107540000 \\
\end{bmatrix}
\]
REFERENCES


<table>
<thead>
<tr>
<th>Distribution List</th>
<th>No. of copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Director</td>
<td>2</td>
</tr>
<tr>
<td>Defense Technology Information Center</td>
<td></td>
</tr>
<tr>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td>Alexandria, VA 22314</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Research Office</td>
<td></td>
</tr>
<tr>
<td>Code 81</td>
<td></td>
</tr>
<tr>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td>Monterey, CA 93943</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Library</td>
<td></td>
</tr>
<tr>
<td>Code 52</td>
<td></td>
</tr>
<tr>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td>Monterey, CA 93943</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Professor Toke Jayachandran</td>
<td></td>
</tr>
<tr>
<td>Department of Mathematics</td>
<td></td>
</tr>
<tr>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td>Monterey, CA 93943</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Chairman, W. Max Woods</td>
<td></td>
</tr>
<tr>
<td>Department of Mathematics</td>
<td></td>
</tr>
<tr>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td>Monterey, CA 93943</td>
<td></td>
</tr>
</tbody>
</table>