Location of a Faulty Module in a Computing System

TEIN-HSIANG LIN, MEMBER, IEEE, AND KANG G. SHIN, SENIOR MEMBER, IEEE

Abstract—Considering the interplay between different phases of fault tolerance, a new problem of locating a faulty module in a computing system is formulated and solved in this paper. First, the probability of each module being faulty or faulty probability is calculated using the likelihood principle from the model parameters for fault detection, diagnostics, error propagation, and error detection. Then, based on the faulty probabilities and a given required diagnostic coverage, the order in which modules are to be diagnosed and the maximum time allotted to diagnose each module are determined by minimizing the average total diagnostic time. An example is presented and analyzed to answer the question of whether or not a system should delay the diagnosis upon detection of an error until more errors are detected.

Index Terms—Bayesian decision theory, detection latency, error propagation, fault (error) detection, fault (error) latency, fault location, periodic diagnostic, system level fault diagnosis.

I. INTRODUCTION

THE rapid progress in VLSI technology has made it possible to build systems with a massive number of components for high performance and high reliability. In such complex systems, it is extremely difficult to locate the faulty component(s) upon occurrence of a failure. A new probabilistic approach to the problem of locating a fault in a multicomponent system is thus proposed in this paper by taking into account the effects of such mechanisms as error detection, error propagation, and periodic diagnostics.

Due to the packaging and scope of each reconfigurable unit, it is sufficient in most systems to locate a “macro component,” called a module, which contains one or more faulty “micro” components. A module represents any well-defined subunit of the system; it could be a hardware unit, a software subroutine, or a combination of both. Each module is assumed to have its own independent detection mechanism(s).

Our approach is concerned with system level fault diagnosis for which various models have been proposed in the literature. The most notable are the model proposed in [22] and extensions thereof. Among them, only a couple of models are directly related to our work and thus will be reviewed below. (See Section VI for reviews on others.)

Bossen and Hsiao [4] proposed a probabilistic model for the detection of intermittent and transient faults. Since intermittent and transient faults that had occurred during normal operation may disappear during off-line testing, they used the information gathered from detection mechanisms to isolate faults. Their model was implemented as an automated diagnostic methodology for the IBM 3081 Processor Unit [30]. New hardware features were added in the IBM 3081 Processor Unit to record the system’s status at the time of occurrence of errors. Thus, in most cases the faulty module can be directly isolated from this information without additional diagnosis. If direct isolation of faults is not possible, then probable faulty modules are tested for stuck-at faults in combinational networks of each of the modules. If the fault is still not found, modules are sequentially replaced until the system is repaired. The sequence of replacement is determined based on the numbers of circuits in each module.

The above approach is deterministic in nature since abundant information on the fault is available, thus almost always reducing the number of probable faulty modules down to one. By contrast, our model will require only the knowledge of the times and places (module identity) of error detections and a probabilistic model of fault behavior. Based on this minimal information, our objective is to locate the faulty module as quickly as possible. For this, a human or another computer system is brought in upon detection of one or more errors to diagnose every module in the system sequentially until the faulty module is identified under the single faulty module assumption¹ or all the modules are exhausted. A Bayesian decision approach will be used to determine the order of modules to be diagnosed and the maximum diagnostic times for individual modules.

The paper is organized as follows. Section II defines the parameters associated with fault detection, periodic diagnostics, error propagation, and error detection. In Section III, the faulty probability of each module is derived, and the optimal order in which modules are to be diagnosed is determined by using a Bayes decision criterion. In Section IV, the optimal diagnostic times for individual modules are determined by minimizing the average total diagnostic time. An example is then presented in Section V. In Section VI, we compare our model to other system level diagnosis models and remark on

¹ Relaxation of this assumption will be addressed in Section VI.
II. SYSTEM MODEL

An N-module computing system is represented by a digraph \( D = (V, E) \), where \( V = \{v_1, \ldots, v_N\} \) denotes the set of nodes, \( E = \{e_{ij} ; 1 \leq i, j \leq N\} \) denotes the set of directed edges. Each node in \( V \) represents a module in the system, and a directed edge \( e_{ij} \) represents a communication link2 from \( v_i \) to \( v_j \). Typical methods of communication between software modules are message passing or shared memory. Hardware modules can communicate via control and data signals. If there is no communication link from \( v_i \) to \( v_j \), \( E \) will not contain \( e_{ij} \), or \( e_{ij} \) is a null edge.

The fault tolerance in a multimodule system is achieved by several sequential phases as shown in Fig. 1. The fault process determines the time, the location, and the type of a fault that occurs in the system. A fault may induce error(s), which is described by the error process. To distinguish an error from a fault, both are defined as follows.

* A fault is a damage, defect, or deviation from the normal state of the computing system on which tasks execute.

* An error is a deviation from the specification of a task.

For example, a broken wire, an electromagnetic interference, or a bug in a program is a fault, whereas an incorrect result or an untimely control signal is an error. A thorough discussion on the terminology about faults and errors can be found in [13].

Based on the definitions of fault and error, the detection phase can be divided into fault detection and error detection. The former is concerned with detecting the manifestation of a fault by some means other than program execution (e.g., a built-in tester), while the latter deals with detecting those errors in program execution induced by fault(s). Hence, the fault location phase will be called for only after an error detection. The problem of locating the faulty module becomes complicated when error detection mechanisms are not perfect (i.e., have less than 100 percent coverage), since an error may propagate from one module to others via intermodule communication before it is detected in some module(s).

In the remainder of this section, the above fault-tolerant phases will be characterized with probabilistic models. The density and cumulative distribution functions of a random variable \( V_i \) \((V_i)\) will be denoted as \( f_i^V \) \((f_i^V)\) and \( F_i^V \) \((F_i^V)\), respectively.

A. Faults and Errors

A module is said to be faulty if it contains one or more faults and is said to become contaminated if it contains an error. Let \( T_i^F \), the \( v_i \)'s faulty time, denote the time instant a fault occurs in \( v_i \). Let \( T_i^C \), the \( v_i \)'s contaminating time, denote the time instant the first error occurs in \( v_i \) as a result of either the manifestation of a fault within \( v_i \) or the propagation of error(s) from other module(s). We assume that the system contains at most one faulty module at a time, although the faulty module could contain multiple faults. An extension of our results to the case of multiple faulty modules will be addressed in Section VI.

If \( v_i \) is the faulty module, the fault latency of \( v_i \), denoted by \( L_i \), is defined as the time interval between \( v_i \)'s faulty time and contaminating time. A methodology for determining the distribution of the faulty latency on real systems was developed in [27].

The rate of fault occurrence in \( v_i \) is characterized by the fault cycle in \( v_i \), denoted by \( Y_i \), which is the time interval between two consecutive faulty times of \( v_i \). In Fig. 2, a time chart is used to illustrate the parameters defined in this subsection.

B. Error Propagation

If a module's detection mechanism is not perfect, faults in that module induce errors which may then propagate to other modules before some module detects one or more of these errors. To describe error propagation in the system, a (propagation) path from \( v_i \) to \( v_k \), written as \( (v_i, \ldots, v_k) \) is defined as a directed path in \( D \) where all nodes in the path are distinct. Only distinct nodes in an error propagation path are considered, because error propagation into an already contaminated module has negligible effects on the propagation behavior of the module. Error will propagate from \( v_i \) to \( v_k \) if there exists at least one propagation path from \( v_i \) to \( v_k \). Define the error propagation time from \( v_i \) to \( v_k \), denoted by \( X_{ik} \), as the time interval between the contaminating time of \( v_i \) and that of \( v_k \). Clearly, \( X_{ik} \) holds a physical meaning only when \( X_{ik} \geq 0 \). Thus, the definition of \( X_{ik} \) should be made under the assumption that \( v_i \) is the only faulty module or the first module to be contaminated in the system. Under the single faulty module assumption, the faulty module will be the first to be contaminated.

The error propagation times contain complete information...
on the behavior of error propagation and can be measured experimentally. However, due to the difficulty and cost of measuring the error propagation times (see [28] for a detailed account of this), we define the direct propagation time of every nonnull edge $e_{ij}$, denoted by $B_{ij}$, as the time for an error to propagate from $v_i$ to $v_j$ via $e_{ij}$. The distribution function of $B_{ij}$ is called the direct propagation function of $e_{ij}$.

The differences between an error propagation time and a direct propagation time are that 1) the latter is associated with a directed edge while the former is defined for every ordered pair of modules, and 2) the latter accounts for error propagation through a particular edge while the former is the minimum propagation time over all propagation paths between the given pair of modules. Since direct propagation times are defined for communication links in the system, without loss of generality, one can assume that they are independent of one another and their distributions do not change even if the underlying fault model is changed.

The relation between $X_{ij}$'s and $B_{ij}$'s can be obtained easily. First, identify all error propagation paths from $v_i$ to $v_j$ and calculate the error propagation time along each path by summing up the direct propagation times of all edges in the path. Then, $X_{ij}$ is the minimum among all these path propagation times. For example, in the system graph $D_1$ shown in Fig. 3,

$$X_{13} = \min \left( \begin{array}{l} B_{12} + B_{23}, \ B_{14} + B_{45} + B_{53}, \ B_{14} + B_{43} + B_{52} + B_{23}, \ B_{12} + B_{54} + B_{43} + B_{53} \end{array} \right).$$

(2.1)

Assuming that an error propagates via the push $(v_1, v_2, v_3)$ of graph $D_1$, the relation between $B_{ij}$ and $X_{ij}$ is shown in Fig. 2.

Deriving the distributions of $X_{ij}$'s from that of $B_{ij}$'s is not trivial. To perform this systematically and efficiently, two algorithms have been proposed in [28] where experimental measurements of the direct propagation times in the fault-tolerant multiprocessor (FTMP) were also reported.

C. Fault and Error Detection

All detection mechanisms detect only errors, while faults have to be diagnosed by coercing them into detectable errors. Based on the detection methodology employed, the scheme in [26] classifies detection mechanisms into three types: 1) fault detection, 2) error detection, and 3) periodic diagnostics. Examples of fault detection (or signal level detection in [26]) mechanisms are built-in self-checking circuits, error detection codes, and duplicate complementary circuits. Error detection (or function level detection in [26]) can be implemented in various ways, such as capability checking, acceptance test, invalid op-code checking, and time-out. Periodic diagnostics are off-line testing programs which are run periodically and can exercise the system's components with imitated inputs to activate and then detect faults.

The distinct feature of fault detection mechanisms is the immediate detection when a fault induces an error so that an error cannot propagate without being detected. Consequently, the module where a detection is made by fault detection mechanisms is the faulty module; the fault location phase is not needed after a fault detection. The faults that can be detected by fault detection mechanisms in a module are defined as FD-detectable faults. The fault detection coverage of $v_i$, denoted by $C_{v_i}^D$, is the probability of a fault in $v_i$ being FD-detectable. Ideally, every fault should be FD-detectable (i.e., $C_{v_i}^D = 1$ for all $i$), which is practically impossible. Thus, to complement (imperfect) fault detection mechanisms, the other two types of detection mechanisms must also be used.

A periodic diagnostic can usually detect more faults with less cost than a fault detection mechanism. The disadvantage is that errors may have been induced and propagated to other modules before the fault is diagnosed, since periodic diagnostics cannot detect faults between diagnostics. The fault location phase is not necessary following a detection with periodic diagnostics. The faults which are not FD-detectable but are detectable during periodic diagnostics are defined as PD-detectable faults. The coverage of periodic diagnostics in module $v_i$, denoted by $C_{v_i}^P$, is then the probability of a fault in $v_i$ being PD-detectable. $C_{v_i}^P$ is a monotonically increasing function of the diagnostic time. Because the system will not be available for useful work during periodic diagnostics, the time for a complete diagnostic (i.e., $C_{v_i}^P = 1 - C_{v_i}^F$) is usually too long to apply frequently. It is a common practice to perform a short periodic diagnostic during normal operation and perform a complete diagnostic as such a need arises or when the system is idle. Periodic diagnostics do not have to be scheduled at uniform intervals, but the time to perform each diagnostic is usually fixed. The scheduling of incomplete periodic diagnostics has been studied extensively by many researchers [12], [16], [20], [21], [32], [31]. Based on these results, we will assume that the scheduling of periodic diagnostics on each module has been determined a priori and each module records the time when the last periodic diagnostic was performed.

The faults that are neither FD-detectable nor PD-detectable are undetectable faults. Undetectable faults can be captured only during the fault location phase after the errors induced by these faults are detected by error detection mechanisms.
As discussed above, faults are classified according to the type of detection mechanisms that can detect them. This classification implies that there is no randomness in fault detection and periodic diagnostics. That is, if a fault is undetectable, it will never be detected by fault detection mechanisms or periodic diagnostics.

A module often employs several error detection mechanisms simultaneously to accelerate the error detection process. Errors induced by a fault proliferate until one of them is trapped by an error detection mechanism. The pattern of error proliferation depends heavily upon the system workload at the contaminating time, and thus, is random. Therefore, unlike the treatment of fault detection and periodic diagnostics, error detection will be modeled probabilistically.

Based on where errors are detected in a module, error detection mechanisms are divided into two groups: internal and boundary detection schemes. For a module, internal detection schemes refer to all the error detection mechanisms which check the module’s internal structure, while the boundary detection schemes detect errors by checking the module’s outputs. For example, the capability checking and module’s outputs. For example, the capability checking and module’s outputs. For example, the capability checking and module’s outputs. For example, the capability checking and module’s outputs.

Define the internal detection latency of \( v_i \), denoted by \( I_i \), as the time interval from the \( v_i \)'s contaminating time until an error is detected at the \( v_i \)'s side of \( e_{ij} \) by a boundary detection scheme. If there is no boundary error detection mechanism on \( e_{ij} \), then \( J_{ij} = \infty \). Note that the starting point of \( J_{ij} \) is the time when an error propagates from \( v_i \) to \( v_j \). Thus, \( \text{Prob}(J_{ij} = 0) \) can be interpreted as the probability of detecting an error just before it propagates to \( v_j \) via \( e_{ij} \).

For any module \( v_i \), we define the detection time as the time instant the first error detection is made in \( v_i \) by either an internal or a boundary detection scheme. The detection latency of \( v_i \), denoted by \( K_i \), is then defined as the time interval from \( v_i \)'s contaminating time to its detection time.

A timing chart is shown in Fig. 4 to illustrate all the latencies defined in this subsection.

Denote the density and cumulative distribution functions of \( K_i \) by \( c_i(t) \) and \( C_i(t) \), respectively. Traditionally, the error detection coverage, defined as the probability of detecting errors existing in a module or system, is used as a measure to evaluate the effectiveness of error detection mechanisms. However, this definition does not specify the detection latency which plays a key role in determining the actual coverage. It also implies that if an error is not detected within a certain time limit, the error is considered to be undetectable forever. Hence, it would be more appropriate to define the error detection coverage as a function of time, \( C_i(t) \).

The detection latency can be viewed as a result of the competition among internal detection and boundary detection schemes, i.e.,

\[ K_i = \min (I_i, B_{hi} + J_{hi}, \ldots, B_{hn} + J_{hn}) \]

with \( \{e_{a1}, \ldots, e_{an}\} \) being the set of all outgoing edges at \( v_i \). Assuming that \( I_i, B_{hi}, \) and \( J_{ij} \) are independent, \( C_i(t) \) can be obtained by

\[ [1 - C_i(t)] = [1 - F_i^n(t)][1 - F_{hi}^m(t)] \ldots [1 - F_{hn}^m(t)] \]

where \( \ast \) denotes convolution and

\[ f_{hi}^{bm} = f_{hi}^{m} \ast f_{hi}^{m} \ast \]

III. THE PROBLEM OF FAULT LOCATION

When an error is detected in a module, the following two decisions have to be made before the faulty module can be located:

1) the order of modules to be diagnosed (or the diagnostic order)
2) the maximum diagnostic time for each module, i.e., when to stop diagnosing a module if a fault is not yet found in the module.

The former is dealt with in this section and the latter will be treated in the next section.

Under the single faulty module assumption, the determination of the diagnostic order can be formulated as a Bayesian decision problem. The parameter space contains \( N \) hypotheses:

\[ H_i : v_i \text{ is the faulty module for } i = 1, \ldots, N. \]

If the decision time of a system, denoted by \( T \), is the time a system's normal operation is stopped due to the detection of error(s), the sample space is the set of fault syndromes defined below. The fault syndrome of a system, denoted by \( S \), is defined as a record of error detections made from 0 to \( T \), i.e.,

\[ S = [v_{01}, t_1; \ldots; v_{0n}, t_n] \]

where \( t_i \) is the time the first error detection in \( v_{0i} \) is made and \( n \) is the number of modules that have made error detection(s) up to \( T \). The prior and posterior fault probabilities, denoted respectively by \( \pi_i^0 \) and \( \pi_i^1 \) for \( i = 1, \ldots, N \), are then defined
as

\[ \pi_i' = \text{Prob}[H_i \text{ is true}] \]

before considering the fault syndrome \( S \),

\[ \pi_i = \text{Prob}[H_i \text{ is true}] \]

after considering the fault syndrome \( S \).

Usually, the decision time is immediately after the system's first error detection to minimize error propagation. However, it is possible for \( S \) to have multiple error-detecting modules, due mainly to the following two reasons. First, the decision time may be predetermined or at fixed intervals. Second, the decision time may be deliberately postponed to gather more information about error propagation.

An action \( A \) is defined as the actual diagnostic order and denoted by

\[ A = [a_1, a_2, \ldots, a_N] \]

where \( a_1 \) is the first module to be diagnosed, \( a_2 \) is the second, and so on. If the number of modules diagnosed before locating the faulty module is defined as the loss function

\[ L(H_i, A) = j \quad \text{if} \quad i = a_j, \]

the posterior Bayesian expected loss of \( A \) will be

\[ \beta(A) = \sum_{k=1}^{N} \pi_k L(H_k, A) \]

\[ = \sum_{j=1}^{N} j \pi_j. \]

The Bayes decision is then the action which has the minimum \( \beta(A) \).

**Theorem 1:** Given the faulty probabilities \( \pi_i, i = 1, \ldots, N \), \( \beta(A) \) is minimized if \( \pi_{a_1} \geq \pi_{a_2} \geq \cdots \geq \pi_{a_N} \).

**Proof:** Suppose there exist \( i \) and \( j \) such that \( i < j \) and \( \pi_{a_i} < \pi_{a_j} \) for an action \( A = [a_1, \ldots, a_N] \). Construct a new action \( A' \) by exchanging \( a_i \) and \( a_j \), i.e., \( A' = [a_1, \ldots, a_{i-1}, a_j, a_{i+1}, \ldots, a_{j-1}, a_i, a_{j+1}, \ldots, a_N] \). Then

\[ \beta(A) - \beta(A') = (j-i) (\pi_{a_j} - \pi_{a_i}) \]

\[ < 0. \]

Therefore, the Bayesian expected loss can be reduced by the above exchanges until the minimum is reached when \( \pi_{a_1} \geq \pi_{a_2} \geq \cdots \geq \pi_{a_N} \).

From Bayes' Theorem, the posterior faulty probability \( \pi_i \) can be calculated as

\[ \pi_i = \frac{\pi_i' \mathcal{L}_i(S)}{\sum_{k=1}^{N} \pi_k' \mathcal{L}_k(S)} \]

where \( \mathcal{L}_i(S) \) is the likelihood of the fault syndrome \( S \) under the hypothesis \( H_i \).

**A. Prior Distribution of \( H_i \)**

The prior distribution of \( H_i \) is our subjective belief about the location of the faulty module before the results from error detection are considered. If we know very little *a priori* the whereabouts of the fault module, a noninformative prior can be employed, which assigns an equal probability to all hypotheses. An informative prior can be determined if we know the previous faulty times of all modules, denoted by \( T_j^f \), and the distributions of fault cycles of all modules, denoted by \( F_j^f \). In case \( v_i \) has never been faulty, \( T_j^f \) may represent the time when \( v_i \) was first put into operation. This informative prior will be calculated on the basis of our suspicion of an individual module being faulty, which is just the conditional probability that one module is faulty and others are not at time \( T \). For example, our suspicion of \( v_i \) being faulty is

\[ F_j^f(T - T_j^f) \prod_{j=1, j \neq i}^{N} (1 - F_j^f(T - T_j^f)). \]

The prior probability of \( H_i \) is then the relative suspicion of \( v_i \) being faulty, i.e.,

\[ \pi_i = \frac{F_j^f(T - T_j^f) \prod_{j=1, j \neq i}^{N} (1 - F_j^f(T - T_j^f))}{\sum_{j=1}^{N} \left( F_j^f(T - T_j^f) \prod_{j=1, j \neq i}^{N} (1 - F_j^f(T - T_j^f)) \right)}. \]

**B. Likelihood of Fault Syndrome**

The likelihood \( \mathcal{L}_i(S) \) is the conditional probability of \( S \) given that \( v_i \) is the faulty module. To determine \( \mathcal{L}_i(S) \), we first derive the density function of \( v_i \)'s contaminating time given \( S \), denoted by \( f_i^c(t) \).

Let \( T_D^p \) denote the time when the previous complete diagnosis on \( v_i \) is done and let \( T_F^p \) denote the time when the previous periodic diagnostic on \( v_i \) is done. A complete diagnosis is assumed to have 100 percent coverage; so we are certain that \( v_i \) is fault-free immediately after \( T_F^p \) if it passed the complete diagnosis. \( T_D^p \) is assumed to equal \( T_F^p \) if no complete diagnosis has been applied since \( T_D^p \). \( T_F^p \) is assumed to equal \( T_D^p \) if no periodic diagnostic has been applied since \( T_D^p \). In \( v_i \), only undetectable faults could have occurred during the interval between \( T_D^p \) and \( T_F^p \), and only PD-detectable and undetectable faults could have occurred during the interval between \( T_F^p \) and \( t_i \), the time of the first error detection. The likelihood of \( T_F^p = t_i, \) denoted by \( f_i^c(t_i) \), is then

\[ f_i^c(t_i) = \begin{cases} (1 - C_i^f - C_i^p)f_i^f(t_i - T_F^p)(1 - F_i^f(t_i - t_i)) & \text{if } T_F^p \leq t_i < T_D^p \\
 & (1 - C_i^p)f_i^f(t_i - T_F^p)(1 - F_i^f(t_i - t_i)) & \text{if } T_D^p \leq t_i < t_i 
\end{cases} \]

Thus, the density function of \( T_F^p \) is expressed as

\[ f_i^T(t) = \frac{\int f_i^c(t_i)}{f_i^c(t_i) \, dt}. \]

By definition, \( T_F^C = T_F^p + L_i \) so that the density function of \( T_F^C \) can be calculated as

\[ f_i^T(t) = f_i^T(t) * f_i^C(t). \]
where "*" denotes the convolution. However, if S is given, $T^*_{ij}$ can be no later than $t_i$ and no earlier than $T'_{ij}$. Hence, $f'_{ij}(t)$ should be expressed as

$$f'_{ij}(t) = \frac{f_{ij}(t)}{\int_0^{t_i} f_{ij}(\tau) \, d\tau}.$$  

Now, the likelihood of S can be calculated as

$$\mathcal{L}_i(S) = \int_{T_f}^{t_i} \mathcal{L}_i(S, \tau) f'_{ij}(\tau) \, d\tau$$

where $\mathcal{L}_i(S, \tau)$ is the likelihood of S given that $U_i$ is the faulty module and $T_f = \tau$.

Let the error latency from $U_i$ to $U_j$, denoted by $E_{ij}$, be the time interval from the $u_i$'s contaminating time to the $u_j$'s detection time. Then, $E_{ij}$ is the sum of the error propagation time $X_{ij}$ and the detection latency in $u_j$, i.e.,

$$E_{ij} = X_{ij} + K_j.$$

Note that if $i = j$, $E_{ij} = K_j$. The spread of errors from a faulty $U_i$ can be characterized by the joint distribution of $E_{i1}, E_{i2}, \cdots, E_{iN}$, called the $U_i$'s joint distribution of error latencies. Having the joint distribution of error latencies, $\mathcal{L}_i(S, \tau) = \text{Prob}[E_{i1} = t_1 - \tau, E_{i2} = t_2 - \tau, \cdots, E_{iN} = t_N - \tau, E_{w_1} > T - \tau, \cdots, E_{w_{N-m}} > T - \tau]$ where $w_1, \cdots, w_{N-m}$ are the modules which had not detected any error. It is very complex and time consuming to analytically derive the joint distribution of error latencies, because $E_{ij}$'s are not independent. A more practical approach to get $\mathcal{L}_i(S, \tau)$ will be numerical simulation using random number generators.

IV. OPTIMAL DIAGNOSIS PROCEDURE

Given a system diagnostic coverage and the optimal diagnostic order determined earlier, we now derive the optimal diagnostic times for individual modules by minimizing the average total diagnostic time.

Periodic diagnostics are usually self-diagnostic routines, but diagnostics at this stage have to be run by an independent system to ensure consistency and high coverage. Assuming that all modules are diagnosed sequentially by an independent fault-free system, the average total diagnostic time, denoted by $TD$, is defined as the average time from the start of the first diagnostic until the faulty module is identified or all the modules are exhausted. Given that there is a faulty module in the system, the system diagnostic coverage, denoted by $M$, is the conditional probability of locating the faulty module. Let $\phi_i(t)$ be the percentage of faults that the diagnostic can discover in $u_i$ for the diagnostic time $t$, and let $\phi_i(t) = d\Phi_i(t)/dt$. All $\phi_i(t)$'s are assumed to be continuous functions on $[0, \infty]$. Let $D_i$, $1 \leq i \leq N$, denote the optimal diagnostic time for $u_i$. Our problem can be formally stated as follows.

The Optimal Diagnostic Problem: Given $M$ and $\pi_i, 1 \leq i \leq N$, determine $D_i, 1 \leq i \leq N$, which minimize $TD$.

Without loss of generality, we can assume that $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_N$ and the diagnostic order is $A = [1, 2, \cdots, N]$. The following theorem provides a necessary condition for $D_i$.

**Theorem 2**: Given $M$ and $\pi_1 \geq \pi_2 \geq \cdots \geq \pi_N$, $D_i$'s must satisfy the following equations:

$$\overline{\Phi}_N(D_N) = \pi_N \Phi_N(D_N)$$

$$\overline{\Phi}_{N-1}(D_{N-1}) = \frac{\pi_N \Phi_N(D_N) + \pi_{N-1} \Phi_{N-1}(D_{N-1})}{1 + \Phi_{N-1}(D_{N-1})}$$

$$\vdots$$

$$\overline{\Phi}_1(D_1) = \frac{\pi_N \Phi_N(D_N) + \pi_{N-1} \Phi_{N-1}(D_{N-1}) + \cdots + \pi_2 \Phi_2(D_2)}{1 + \Phi_{N-1}(D_{N-1}) + \cdots + D_2}$$

**Proof**: The diagnostic coverage of the system $M$ is a weighted sum of all modules' individual diagnostic coverages, i.e.,

$$M = \sum_{i=1}^{N} \pi_i \overline{\Phi}_i(D_i).$$

According to the given diagnostic order, $TD$ is computed as

$$TD = \pi_1 \left[ \int_0^{D_1} x_1 \, d\Phi_1(x_1) + (1 - \Phi_1(D_1)) \sum_{i=1}^{N} D_i \right]$$

$$\vdots$$

$$\pi_N \left[ \sum_{i=1}^{N} D_i + \int_0^{D_N} x_N \, d\Phi_N(x_N) + (1 - \Phi_N(D_N)) \sum_{i=1}^{N} D_i \right]$$

$$= \sum_{i=1}^{N} \pi_i \left[ \int_0^{D_i} x_i \, d\Phi_i(x_i) + \sum_{i=1}^{N} D_i - \sum_{i=1}^{N} \pi_i \overline{\Phi}_i(D_i) \sum_{i=1}^{N} D_i \right].$$

Let $\theta$ be a Lagrangian multiplier. The solution for $D_i, 1 \leq i \leq N$, must satisfy the differential equation

$$\nabla \left[ TD + \theta \left( M - \sum_{i=1}^{N} \pi_i \overline{\Phi}_i(D_i) \right) \right] = 0.$$
which is equivalent to

\[
\frac{\partial \text{TD}}{\partial D_k} = \pi_k \phi_k(D_k) = \frac{\partial \text{TD}}{\partial D_1} - \pi_2 \phi_2(D_2) = \cdots = \frac{\partial \text{TD}}{\partial D_N} - \pi_N \phi_N(D_N) = 0. 
\]

(4.3)

For \(1 \leq k \leq N\),

\[
\frac{\partial \text{TD}}{\partial D_k} = \pi_k \phi_k(D_k) + 1 - \sum_{i=1}^{N} \pi_i \Phi_i(D_i) - \pi_k \phi_k(D_k) \sum_{i=k+1}^{N} D_i
\]

\[= 1 - \sum_{i=1}^{k} \pi_i \Phi_i(D_i) - \pi_k \phi_k(D_k) \sum_{i=k+1}^{N} D_i. \]

(4.4)

Equation (4.4) is derived by the fundamental theorem of calculus that gives

\[
\frac{\partial}{\partial D_k} \left[ \pi_k \int_{0}^{C} x_k \Phi_k(x_k) dx_k \right] = \pi_k \frac{\partial}{\partial D_k} \left[ \int_{0}^{C} x_k \phi_k(x_k) dx_k \right] = \pi_k \left( \Phi_k(D_k) - \phi_k(D_k) \right). 
\]

and the fact

\[
\sum_{i=1}^{N} \pi_i \Phi_i(D_i) \sum_{j=1}^{N} D_j = \sum_{j=1}^{N} D_j \sum_{i=1}^{N} \pi_i \Phi_i(D_i),
\]

since \(\pi_i, \Phi_i(D_i), \) and \(D_i\) are positive. From (4.3) and (4.4), we can derive for \(1 \leq k \leq N\),

\[
0 = \left( \frac{\partial \text{TD}}{\partial D_k} - \pi_k \phi_k(D_k) \right) - \left( \frac{\partial \text{TD}}{\partial D_N} - \pi_N \phi_N(D_N) \right)
\]

\[= \sum_{i=k+1}^{N} \pi_i \Phi_i(D_i) - \pi_k \phi_k(D_k) \sum_{i=k+1}^{N} D_i
\]

\[= \pi_N \phi_N(D_N) - \sum_{i=k+1}^{N} \pi_i \Phi_i(D_i)
\]

\[= \pi_k \phi_k(D_k) \left( 1 + \sum_{i=k+1}^{N} D_i \right)
\]

which is another form of (4.1).

Using the following numerical procedure, the exact value of \(D_N\) can be determined by (4.1) and the constraint \(M = \sum_{i=1}^{N} \pi_i \Phi_i(D_i)\).

Step 1: Assign an initial value to \(D_N\).
Step 2: Let \(n := N\).
Step 3: Let \(\Delta t := D_N\).
Step 4: Calculate \(D_i, 1 \leq i \leq n - 1\), according to (4.1).
Step 5: Calculate \(m = \pi_1 \Phi_1(D_1) + \cdots + \pi_N \Phi_N(D_N)\).
Step 6: If \(|m - M| < \epsilon\) where \(\epsilon\) is a prespecified accuracy, then stop. Otherwise, do the following in sequence:

1. If \((m - M) \cdot \Delta t < 0\) then let \(\Delta t := -\Delta t / 4\).
2. If \(D_n + \Delta t \leq 0\), let \(D_n := 0\), decrement \(n\) by 1, and then go to Step 3.
3. Let \(D_n := D_n + \Delta t\) and then go to Step 4.

The unidirectional search employed in the above procedure is feasible because \(m\) is a monotonically increasing function of \(D_i\) for \(1 \leq i \leq N\). If \(D_i = 0\) in the final result, it indicates that \(\pi_i\) is so small that the required diagnostic coverage can be achieved without even diagnosing \(v_i\).

The value of \(D_N\) should satisfy

\[1 - \frac{1 - M}{\pi_N} < \Phi_N(D_N) < 1\]

since

\[M = \sum_{i=1}^{N-1} \pi_i \Phi_i(D_i) + \pi_N \Phi_N(D_N)\]

\[< \sum_{i=1}^{N} \pi_i + \pi_N \Phi_N(D_N)\]

\[= (1 - \pi_N) + \pi_N \Phi_N(D_N).\]

The above range is useful in selecting an initial value of \(D_N\).

If \(M\) is so close to 1 that for \(1 \leq i \leq N, \Phi_i(D_i) = 1, \phi_i(D_i) \ll 1\), and the tail part of \(\Phi(t)\) is approximated by the exponential function \((1/\lambda t)e^{-t/\lambda t}\), then the optimal diagnostic time can be approximated by

\[
\frac{D_N}{\lambda_{N-1}} = \ln \left( \frac{\pi_{N-1}}{\pi_N} \left( \frac{1 + D_N}{\lambda_{N-1}} \right) \right)
\]

\[
\frac{D_{N-2}}{\lambda_{N-2}} = \ln \left( \frac{\pi_{N-2}}{\pi_{N-1} + \pi_{N-2}} \left( \frac{1 + D_N + D_{N-1}}{\lambda_{N-2}} \right) \right)
\]

\[
:\vdots
\]

\[
\frac{D_1}{\lambda_1} = \ln \left( \frac{\pi_1}{\pi_1 + \cdots + \pi_N} \left( \frac{1 + D_N + D_{N-1} + \cdots + D_2}{\lambda_1} \right) \right).
\]

From this approximation, it is easy to see that when all modules are equally likely to be faulty (i.e., \(\pi_1 = \cdots = \pi_N = 1/N\)), the optimal diagnostic time of each module increases exponentially with its position in the diagnostic order.

To see the effect of the faulty probabilities on the optimal diagnostic times, we consider three cases of the faulty probabilities for a five-module system which are tabulated in Table I. Diagnostic time for individual modules are assumed to be exponentially distributed with \(\lambda_i = \cdots = \lambda_5 = 100\). The choice for the units of all time variables and \(\lambda_i\)’s are arbitrary but identical for the purpose of comparison. In Case 1, all five modules have the same faulty probability. In Case 2 (Case 3),

<table>
<thead>
<tr>
<th>Case</th>
<th>(\pi_1)</th>
<th>(\pi_2)</th>
<th>(\pi_3)</th>
<th>(\pi_4)</th>
<th>(\pi_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
<td>20.0</td>
</tr>
<tr>
<td>Case 2</td>
<td>51.6</td>
<td>25.8</td>
<td>12.9</td>
<td>6.5</td>
<td>3.2</td>
</tr>
<tr>
<td>Case 3</td>
<td>67.0</td>
<td>22.3</td>
<td>7.4</td>
<td>2.5</td>
<td>0.8</td>
</tr>
</tbody>
</table>
the faulty probability of a module is twice (three times) that of the next module in the diagnostic order. Using the numerical procedure described earlier, the optimal diagnostic times for all three cases under 90 percent system diagnostic coverage are computed and plotted in Fig. 5. The last category in Fig. 5 is the average diagnostic time derived from (4.2) and the exponential distribution of diagnostic times as follows:

$$TD = \sum_{i=1}^{N} \pi_i (\lambda_i - (D_i + \lambda_i)) e^{-D_i/\lambda_i}$$

Several conclusions can be drawn from Fig. 5 as follows.

- If two modules have the same faulty probability and the same distribution of diagnostic times, the one diagnosed earlier requires less diagnostic time, as evidenced in Case 1.
- Generally, the diagnostic time for any module increases (and decreases) with its faulty probability, but the amount of increase (decrease) is larger for those modules diagnosed later.
- The average total diagnostic time is shorter when there are larger disparities in the faulty probabilities. For example, Case 1 (Case 3) has the least (most) disparity in the faulty probabilities and, thus, the longest (shortest) average total diagnostic time.

In Fig. 6, the average total diagnostic times are computed for all three cases with the system diagnostic coverage ranging from 81 to 98 percent. The average total diagnostic time is observed to increase substantially when the system diagnostic coverage approaches 1.

V. NUMERICAL EXAMPLE

An example is presented here to demonstrate the use of our model for locating a faulty module. For a given distribution of each basic random parameter introduced thus far, the following will be computed for the example system:

1) the prior faulty probabilities,
2) the likelihoods of two types of fault syndromes,
3) the posterior faulty probabilities,
4) the optimal diagnostic times for individual modules.

All random parameters, except for $J\gamma^0$'s and $B\gamma^0$'s, are assumed to be exponentially distributed. $J\gamma^0$ is assumed to have the following density function:

$$f_{J\gamma^0}(t) = \begin{cases} 
\xi_{J\gamma^0}(1/\kappa_J)e^{-t/\kappa_J} & \text{if } t > 0 \\
\xi_{J\gamma^0}(1/\kappa_J) & \text{if } t = 0
\end{cases}$$

The discrete probability at $J\gamma^0 = 0$ is to describe the instantaneous detection coverage of the boundary detection mechanisms.

From our experiments on FTMP [28], $B\gamma^0$'s are found to have multimodal distributions. Hence, the $B\gamma^0$'s will be assumed to have a bimodal distribution, i.e.,

$$B\gamma^0 = \begin{cases} 
\mathcal{N}(\mu_{B\gamma^0}, \sigma_{B\gamma^0}^2) & \text{with probability } \eta_{B\gamma^0} \\
\mathcal{N}(\mu_{B\gamma^0}', \sigma_{B\gamma^0}'^2) & \text{with probability } 1 - \eta_{B\gamma^0}
\end{cases}$$

where $\mathcal{N}(\mu, \sigma)$ denotes a normally distributed random variable with mean $\mu$ and standard deviation $\sigma$.

The likelihoods of fault syndromes are obtained via simulation because of the analytic difficulty discussed in Section II. To increase accuracy, every independent random parameter in our model is generated by an independent random generator in the simulation. One million sets of data are generated for the example.

Two types of fault syndromes are considered: 1) the single-detection type where the decision time is set at the detection time of the first error-detecting module of the system, and 2)
TABLE I

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>Exponential</td>
<td>Mean = 100000</td>
</tr>
<tr>
<td>$I_1$</td>
<td>Exponential</td>
<td>Mean = 40</td>
</tr>
<tr>
<td>$I_j$</td>
<td>Exponential</td>
<td>Mean = 140</td>
</tr>
<tr>
<td>$J_{ij}$</td>
<td>Hybrid</td>
<td>$\zeta_{ij} = 0.2$, $\kappa_{ij} = 100$</td>
</tr>
<tr>
<td>$B_{ij}$</td>
<td>Bimodal Normal</td>
<td>$\eta_{ij} = 0.6$, $\mu_{ij}^1 = 40$, $\sigma_{ij}^1 = 25$, $\mu_{ij}^2 = 80$, $\sigma_{ij}^2 = 60$</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>Constant Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

The double-detection type where the decision time is set at the detection time of the second error-detecting module of the system. The single-detection and double-detection types of fault syndromes will be written as $(9, T)$, $(9, T - dt; 3, T)$, respectively, where $T$ is the decision time and $9$ and $3$ are the first and second error-detecting modules. To obtain the double-detection type of fault syndrome, the system has to delay diagnosis for time $dt$ from the first error detection. Will the average total diagnostic time for the double-detection type of fault syndrome be shorter than that for the single-detection type? Our example will answer this question. Since there is a small probability that the delay for the second error-detecting module is unrealistically long, the system will stop waiting after some predetermined time. The delay is set to be no longer than 100 units of time because there are very few double-detection syndromes with a delay longer than 100 in our simulation.

The example system is represented by the graph $D_1$ as shown in Fig. 3. For all nodes and edges of $D_1$, the associated random parameters are assumed to have the distributions in Table II. The assumed values for the constant parameters of $D_1$ are tabulated in Table III. Among the simulation results for all single-detection and double-detection types of fault syndrome, we present only a subset of them that are deemed interesting. The prior faulty probabilities for individual modules are calculated to be $\pi^*_1 = 0.348$, $\pi^*_2 = 0.267$, $\pi^*_3 = 0.193$, $\pi^*_4 = 0.126$, and $\pi^*_5 = 0.066$. The posterior faulty probabilities for syndromes $(4, T - dt; 1, T)$, $(4, T - dt; 2, T)$, $(4, T - dt; 3, T)$, and $(4, T - dt; 5, T)$ are plotted in Figs. 7-10 where the point for $dt = 0$ is actually the result of syndrome $(4, T)$.

As shown in Fig. 7 where $v_4$ is the first error-detecting module of the system, its faulty probability (about 0.5) is the highest among all the modules in $D_1$. However, if $v_1$ subsequently detects an error, $v_1$'s faulty probability becomes the highest at about 0.9, because it is much easier for an error to propagate from $v_1$ to $v_4$ than from $v_4$ to $v_1$. In Fig. 8, where $v_2$ will detect an error after $v_1$, if the delay $dt < 60$, $v_2$ has the highest faulty probability; but, if the delay $dt \geq 60$, $v_4$ again
has the highest faulty probability. This is because errors propagating from \(v_4\) to \(v_2\) via \(v_5\) take about twice the time than propagating from \(v_2\) to \(v_4\) and the detection coverages of \(v_2\) and \(v_4\) are the same. Note that \(v_1\) also has a moderate faulty probability since errors are also likely to elude the detection mechanisms in \(v_1\) and propagate to \(v_2\) and \(v_4\). In Fig. 9, if \(dt\) is small, all modules except \(v_2\) would have some nonzero faulty probabilities, but when \(dt\) becomes larger, \(v_4\) will dominate others in the faulty probability. Fig. 10 is typical for those syndromes where the first error-detecting module has an output link to the second error-detecting module. The longer the delay, the higher the faulty probability of the first error-detecting module will result.

Fig. 11 compares the average total diagnostic times for all the syndromes shown in Figs. 7–10. It shows that not all double-detection fault syndromes lead to shorter average diagnostic times than single-detection fault syndromes do. There is the time delay \((dt)\) penalty for the double-detection type. Then, how can we justify the use of the double-detection type? Given that \(v_4\) has made the first error detection, the likelihoods of obtaining those different double-detection fault syndromes are derived and listed in Table IV. It is apparent that a double-detection fault syndrome may increase the average total diagnostic time over the single-detection type only with a probability of 0.091. To evaluate these two types of faulty syndromes further, we calculate and compare the time overheads each of which is defined as the mean time from the first error detection until the end of diagnosis. The results are listed in Table V. It can be concluded that if the first error detection is made in \(v_1\), \(v_2\), or \(v_3\), it is better to diagnose the system immediately; otherwise, it would be better to wait for a second error detection.

VI. REMARKS

System level fault diagnosis has been dealt with in numerous papers [1], [5], [10], [11], [18], [19], [22], [29], [24], [23]. The models used/proposed in these papers will collectively be referred to as the PMC model. The basic framework of the PMC model is a system consisting of modules which are
TIME OVERHEADS FOR SINGLE- AND DOUBLE-DETECTION FAULT

The faulty module(s) using the test results from all modules. The object of the
capable of diagnosing other modules via available test links between modules. The object of the
PMC model is to identify the faulty module(s) using the test results from all modules. Interesting problems in the PMC model are 1) determination of the number of faulty modules which can be identified for a
given set of test links, 2) assignment of test links to be able to
diagnose r faulty modules, and 3) development of algorithms to identify faulty modules from test results.

There are three major differences between our model and the PMC model as discussed below. The first difference lies in
our assumption that an independent fault-free system is available to diagnose all modules. In our model, a module can be defined at a lower level since it is not required to diagnose other modules and there must be test links from the independent system to all modules. In the PMC model, an independent fault-free system is also required to collect and analyze the test results. The second difference is that we consider fault diagnosis as the next step to error detection while in the PMC model fault diagnosis does not consider any interplay between different phases of fault tolerance. We can therefore use the fault syndrome to determine the posterior faulty probabilities and the diagnostic order. In an extension of the PMC model [15], the failure rates of individual modules are considered. The third difference is in the assumption on the diagnosis itself. The PMC model assumes that every diagnosis is complete, i.e., a nonfaulty module can always diagnose a faulty module. This unrealistic assumption is relaxed somewhat in [3] and [2] where incomplete diagnosis is considered. In our model, not only that all diagnoses are incomplete but the diagnostic coverage is assumed to be an increasing function of the diagnostic time.

Although our model and the PMC model have different assumptions and approaches, they could complement each other. One interesting problem is how to minimize the
diagnosis time for a PMC model, using partial test results and sequential decision theory.

Another approach to system level fault diagnosis is based on comparison [6], [8], [14]. The idea behind this approach is to detect and identify the faulty units with user tasks. Assuming that a set of identical processing units are available, a task (or job) will be executed on a selected pair of processing units and the results from both units are compared. Disagreement between the results indicates that one of the processing units is faulty. The faulty processing unit may be identified when a sufficient number of comparisons are made. Although the comparison can be carried out by hardware matchers [17] or any processing unit other than the units that had executed the task [14], an independent system is required to collect and analyze the comparison results. The advantages of this approach are that 1) it covers transient and intermittent faults, 2) it does not have the hardware and software overheads of detection mechanisms and diagnostic programs. It can even utilize the idle time of the processing units to execute duplicated tasks [7]. However, usefulness of the comparison approach is limited because a) it only covers the faults in the processing units, b) it may take a very long time to diagnose the faulty units executing user tasks, and c) it does not consider intertask communications and hence error propagation among tasks.

A fault can be diagnosed only if it is active during diagnosis. When a fault syndrome is obtained, we do not know whether it is the result of a transient, intermittent, or permanent fault. To derive the optimal diagnostic time for each module, the system diagnostic coverage is determined first under the assumption that the fault is permanent. The determination of this diagnostic coverage is an interesting problem in its own right, which is related to system reconfiguration and error recovery. If no faults are found after diagnosing all the modules, the fault could be transient or intermittent or permanent but nondiagnosable. At that time, the posterior faulty probabilities can still be used, in deciding what to do next.

In this paper, we assumed that there is at most one faulty module in the system, i.e., the single faulty module assumption. (There could be multiple faults in the faulty module, though.) To extend our model to the case of multiple faulty modules, the number of hypotheses in the parameter space of the Bayesian decision problem has to be increased as follows:

$$H_{ij} : u_i \text{ and } u_j \text{ are the faulty modules for } 1 \leq i < j \leq N,$$

$$H_{ik} : u_i, u_j, \text{ and } u_k \text{ are the faulty modules for } 1 \leq i < j < k \leq N,$$

Then, the same approach for the case of single faulty module can be used to obtain the posterior likelihood of each hypothesis being true. However, the complexity in calculating these likelihoods will increase dramatically.

The main issue in using our model for real systems is how to collect the necessary statistics and estimate the distributions of the parameters introduced in our model. This issue is discussed below from two different viewpoints. First, the difficulty in obtaining the information on input parameters is

<table>
<thead>
<tr>
<th>Fault Syndrome</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>(4, T - dt; 1, T)</td>
<td>19.8%</td>
</tr>
<tr>
<td>(4, T - dt; 2, T)</td>
<td>26.2%</td>
</tr>
<tr>
<td>(4, T - dt; 3, T)</td>
<td>9.1%</td>
</tr>
<tr>
<td>(4, T - dt; 5, T)</td>
<td>44.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First error-detecting module</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>
inevitable and is the price to pay for any realistic model like ours. As mentioned in Section II, we have already developed methodologies to measure direct propagation times and the distribution of fault latencies on real systems [27], [28]. The error latency and coverage have also been measured by some other researchers [9], [25] to evaluate different detection mechanisms. Similar methodologies are expected to be developed in the future to monitor the behavior of faults and errors. It is also important to note that the parameters introduced in our model are also needed to model other phases of fault tolerance.

Second, distributions of the random parameters in our model do not have to be exact to be useful. The faulty module is actually diagnosed by the diagnostic programs applied to each module. All these distributions are used to decide the diagnostic order and the diagnostic time. It is meaningless to verify that a parameter follows a certain distribution, because the distribution is just a summary of our past information about the parameter. If we know little about a parameter, a uniform distribution is just a summary of our past information about the parameter. This applies to the determination of the prior faulty probabilities as well as other parameters. The key idea is to make the best use of available information for fault diagnosis, instead of assuming the perfect (or very accurate) information about faults and errors.

VII. CONCLUSION

In this paper, we have formulated and solved a new problem of locating a faulty module. The distinguishing characteristics of our model as compared to other models on system level fault diagnosis are the consideration of 1) the information from other phases of fault tolerance, 2) the time overheads of fault diagnosis, and 3) the imperfectness of diagnostic coverage.

Based on the likelihood principle, the faulty probabilities for individual modules have been calculated from parameters of fault detection, periodic diagnostic, error propagation, and error detection. The optimal diagnostic order of modules is then found to be the decreasing order of the faulty probabilities. Given a system diagnostic coverage and the optimal diagnostic order, the optimal diagnostic time for each module is then determined to minimize the average total diagnostic time. A numerical example is presented and discussed, and the following conclusions have been drawn. First, in most cases the faulty probabilities show large disparities so that it is worthwhile to compute the faulty probabilities for individual modules before actual diagnosis. Third, delaying diagnosis until the second error detection can sometimes improve the total diagnostic time overhead. The determination of the optimal delay period for diagnosis purpose is an interesting future problem.

APPENDIX

LIST OF VARIABLES

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{ij}$</td>
<td>Density function of the random variable $V_{ij}$</td>
</tr>
<tr>
<td>$F_{ij}$</td>
<td>Distribution function of the random variable $V_{ij}$</td>
</tr>
<tr>
<td>$Y_i$</td>
<td>Fault cycle of $u_i$</td>
</tr>
<tr>
<td>$L_i$</td>
<td>Fault latency of $u_i$</td>
</tr>
<tr>
<td>$C_i$</td>
<td>Coverage of fault detection in $u_i$</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Coverage of the periodic diagnostic in $u_i$</td>
</tr>
<tr>
<td>$B_{ij}$</td>
<td>Direct propagation time from $u_i$ to $u_j$</td>
</tr>
<tr>
<td>$X_{ij}$</td>
<td>Error propagation time from $u_i$ to $u_j$</td>
</tr>
<tr>
<td>$I_i$</td>
<td>Internal detection latency in $u_i$</td>
</tr>
<tr>
<td>$J_i$</td>
<td>Boundary detection latency on $e_i$</td>
</tr>
<tr>
<td>$K_i$</td>
<td>Detection latency in $u_i$</td>
</tr>
<tr>
<td>$H_i$</td>
<td>Hypothesis of $u_i$ being the faulty module</td>
</tr>
<tr>
<td>$\pi_i$</td>
<td>Prior Prob[$H_i$]</td>
</tr>
<tr>
<td>$\pi$</td>
<td>Posterior Prob[$H_i$]</td>
</tr>
<tr>
<td>$S$</td>
<td>Fault syndrome</td>
</tr>
<tr>
<td>$L(S)$</td>
<td>Fault syndrome's likelihood function if $H_i$ is true</td>
</tr>
<tr>
<td>$E_0$</td>
<td>Error latency for the error originated from $u_i$ and detected in $u_j$</td>
</tr>
<tr>
<td>$T^Y$</td>
<td>Last time a repair was done on $u_i$</td>
</tr>
<tr>
<td>$T^D$</td>
<td>Last time a thorough diagnostic was done on $u_i$</td>
</tr>
<tr>
<td>$T^P$</td>
<td>Last time a periodic diagnostic was done on $u_i$</td>
</tr>
<tr>
<td>$T^C$</td>
<td>The contaminating time of $u_i$</td>
</tr>
<tr>
<td>$T^F$</td>
<td>The faulty time of $u_i$</td>
</tr>
<tr>
<td>$T^C$</td>
<td>The decision time of the system</td>
</tr>
<tr>
<td>$\Phi_i$</td>
<td>Percentage of faults that the diagnostic can uncover in $u_i$ and $\phi_i = d\Phi_i/dt$</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Optimal diagnostic time for $u_i$</td>
</tr>
</tbody>
</table>

REFERENCES


Tein-Hsiang Lin (S'83-M'88) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Republic of China, in 1980, the M.S. degree in electrical engineering from Iowa State University, Ames, in 1984, and the Ph.D. degree in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1988.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, The State University of New York at Buffalo.

His research interests include fault-tolerant computing, real-time distributed systems, and performance evaluation.

Kang G. Shin (S'75-M'78-SM'83), for a photograph and biography, see p. 18 of the January 1990 issue of this TRANSACTIONS.