Lifetime Reliability for Load-Sharing Redundant Systems With Arbitrary Failure Distributions

Lin Huang, *Student Member, IEEE*, and Qiang Xu, *Member, IEEE*

Abstract—In this work, a general closed-form expression is presented for the lifetime reliability of load-sharing k-out-of-n:G hybrid redundant systems. In such systems, m components are initially configured as active units. Depending on whether it is performing tasks, an active component can be in either a processing, or a wait state. Each state corresponds to an arbitrary failure distribution. The remaining $(n-m)$ are spares to provide fault tolerance. Each time an active component fails, a spare one converts into active mode, until there are no more spares in the system. Then, the system works in a gracefully degrading manner such that less than m components share the workload, until the number of good components is less than k. The task allocation, and service are modeled as queueing systems, wherein the utilization ratio essentially affects the aging effect of components. We integrate the various failure distributions for components in different operational states into an analytical model according to the statistical properties of the task allocation mechanisms, and the components' processing capacity, and analyse the lifetime reliability of the entire system. Finally, three special cases, and a series of numerical experiments are discussed in detail to show the practical applicability of the proposed approach.

Index Terms—Arbitrary distribution, hybrid k-out-of-n:G system, load-sharing, queueing model.

ACRONYM

CMOS	complementary metal-oxide semiconductor
FIFO	first-in-first-out
IC	integrated circuit
MTTF	mean time to failure

NOTATION

n	number of components in the system
m	number of active components in the standby phase
k	minimum number of components required for system operation
$|\mathcal{S}|$	number of elements in set $\mathcal{S}$
λ	task arrival rate of the entire system
μ	task service rate of an active component
ρ	utilization ratio of an active component
p	an active component's task branch-out probability
$\mathcal{R}(t)$	general reliability function
$R_p(t)$	reliability function of processing state
$R_w(t)$	reliability function of wait state
θ	general scale parameter
θ_p	scale parameter of processing state
θ_w	scale parameter of wait state
t^b	birth time of a component
$\mathbf{t}_\ell$	vector representing occurrence time of past ℓ component failures, $(t_1, t_2, \cdots, t_\ell)$
$\mathbf{t}_\ell^{(r)}$	r-dimensional subvector of $\mathbf{t}_\ell$, $(t_1, t_2, \cdots, t_r)$
$\psi_p(t, t^b, \mathbf{t}_\ell)$	cumulative time of a component with birth time t^b in the processing state having failures at $\mathbf{t}_\ell$
$\psi_w(t, t^b, \mathbf{t}_\ell)$	cumulative time of a component with birth time t^b in the wait state having failures at $\mathbf{t}_\ell$
$\psi(t, t^b, \mathbf{t}_\ell)$	unified cumulative usage time of a component with birth time t^b having failures at $\mathbf{t}_\ell$
$P^{sys}(t)$	system reliability at time t
$P_{j,i}^{sys}(t)$	the probability that the system contains j active components, and i good spare components at time t

$P_{n-\ell}^{sys}(t)$	the probability that the system contains $(n-\ell)$ good components at time t
$P_{n-\ell}^{sys}(t, \mathbf{t}_\ell; \mathbf{x}_\ell)$	the probability that the system contains $(n-\ell)$ good components at time t, and ℓ failures can be described by vectors $\mathbf{t}_\ell$, and $\mathbf{x}_\ell$
$P_{n-\ell}^{sys}(t\|\mathbf{t}_\ell; \mathbf{x}_\ell)$	the conditional probability that the system contains $(n-\ell)$ good components at time t given the ℓ failures can be described by vectors $\mathbf{t}_\ell$, and $\mathbf{x}_\ell$
$R(t, t^b\|\mathbf{t}_\ell)$	conditional reliability, the probability that a component with birth time t^b survives at time t given the system experiences ℓ failures at $\mathbf{t}_\ell$ respectively $(t > t_\ell)$
$f(t_r, t_{x_r}\|\mathbf{t}_\ell^{(r-1)})$	the probability that an indicated component with birth time t_{x_r} fails at time t_r given the past $(r-1)$ failures of the system occurs at $\mathbf{t}_\ell^{(r-1)}$
$\mathbf{x}_\ell$	ℓ-dimensional vector representing the birth time indices of past ℓ component failures, $(x_1, x_2, \cdots, x_\ell)$
$\mathbf{x}_\ell^{(r)}$	r-dimensional subvector of $\mathbf{x}_\ell$, $(x_1, x_2, \cdots, x_r)$
$\mathcal{X}_\ell$	set of all possible $\mathbf{x}_\ell$
$\pi_{i,j}$	the number of i in the first j elements of vector $\mathbf{x}_\ell$
$g_{n-r+1}^{sys}(t_r; x_r\|\mathbf{t}_\ell^{(r-1)}; \mathbf{x}_\ell^{(r-1)})$	the probability that a system containing $(n-r+1)$ good components experiences the rth failure at time t_r, and the failure component has birth time t_{x_r} given the past $(r-1)$ failures can be described by $\mathbf{t}_\ell^{(r-1)}$ and $\mathbf{x}_\ell^{(r-1)}$
$MTTF^{sys}$	system mean time to failure

ASSUMPTION

1) The system is a hybrid k-out-of-n:G system.
2) All components are s-independent.
3) A component is in either active mode, or spare mode (as cold standby). An active component alternates between the wait state (as warm standby), and the processing state (as operating component).
4) All active components in this system share the load equally.
5) The failure-time distributions for both warm standby, and operating components follow an arbitrary baseline reliability function. They differ in terms of their scale parameter. Components in cold standby have a zero failure rate.
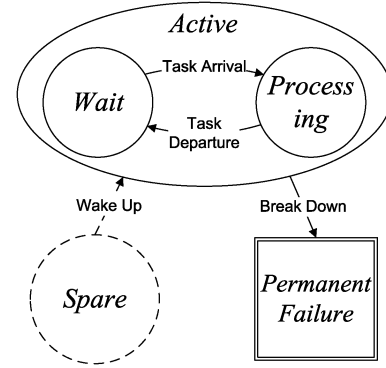

Fig. 1. Component behavior of hybrid redundant systems [3].

6) No repair or maintenance is considered.
7) Switching components is a perfect process.

I. INTRODUCTION

In many load-sharing systems, the load to be processed is specified as tasks, such as applications performed by processing elements in a multiprocessor computing system, bar codes read by laser scanners, or cars assembled by robots. The time scale for processing a single task is usually much smaller than that of a system's lifetime. Therefore, depending on whether or not a component is performing tasks, it may frequently alternate between the *processing state*, and the *wait state* in its lifetime, as depicted in Fig. 1. Generally speaking, components operate at higher temperature, higher pressure, and/or higher speed, and hence will wear out more quickly in the processing state than in the wait state. Consequently, it is more reasonable to regard components in the wait state as in warm standby, when compared to prior work that essentially assumes hot standby [1], [2]. In addition, to provide fault tolerance, a system may contain some *spare components* which convert into *active mode* (including processing, and wait states), one for one when an active one fails. Eventually, if no more spare components exist in the system, when an active component fails, the system will assign more tasks per unit time to the surviving components, which can increase their failure rate.

Without loss of generality, we consider hybrid redundant k-out-of-n:G systems [4], [5], in which m are initially set as active units, with the remaining $(n-m)$ components put aside $(k \le m \le n)$ as spare units. Upon detection of the failure of an active component, the system attempts to replace the faulty one with a spare one until there is no spare component in the system. This process is called the *standby phase*. **We assume that a dedicated switching component takes charge of system reconfiguration, and this process is perfect.** Because then the system works in a gracefully degrading manner, we refer to this phase as the *degrading phase*. That is, when a component failure is detected, the system attempts to reconfigure to a system with one fewer component but k active components, until no more than k good components are left in the system, all of which are active. Such a system functions correctly if at least k out of the n components do not fail. When $n = m$, the system discussed above becomes a load-sharing k-out-of-n:G gracefully degrading system; while

when $k = m$, it is essentially a standby redundant system. It is rather challenging to model the lifetime reliability of the above system. First, an active component can be in two states: processing, and wait. While the quantity of active components in the system is clear, that of components in each state at a particular moment depends on the current workload, and therefore is uncertain. Second, each state corresponds to its own failure distribution. To express the reliability of a component, and the entire system, we need an integrated failure distribution. Note that, in most cases, we cannot predict the exact arrival time, and service time of tasks (the only exception is deterministic arrival, and deterministic service). Therefore we do not know when a component transitions between processing and wait states. Moreover, the frequency of state transitions can be quite high, which also brings challenges to achieve an integrated failure distribution. This problem becomes even more complicated when these failure distributions are not exponential. Last but not least, because active components need to share the load, failures may result in higher workload on the surviving ones, and hence affect their failure distributions. The model should also be able to capture this fact.

To tackle the above problem, in this paper, we develop an analytical model that captures the complex relationship discussed above. We introduce the *cumulative time* concept to reflect the aging effect in each state with arbitrary failure distributions. Next, the cumulative time in all states are combined in a unified manner to express the reliability function of a single component. After that, we model the lifetime reliability of the entire system that involves the task allocation mechanism, and redundancy strategies. Then, we discuss several special cases in detail to show the practical applicability of the proposed technique.

The reminder of this paper is organized as follows. In Section II, we survey related work. Section III then details the proposed reliability model. Next, we verify the proposed models, and demonstrate the detailed analytical procedure with three special cases in Section IV. We then present a series of results obtained with the proposed modeling method, and Monte Carlo integration in Section V. Finally, Section VI concludes this paper.

II. Related Work

As highlighted in [1], the key feature of load sharing k-out-of-n:G systems is that workload has significant influence on every component's failure rate. While there are many studies on this topic capturing this feature, most of them assume an exponential lifetime distribution for every component [1], [2], [6]. By this assumption, the entire system can be represented by a Markov transition diagram, and hence the complexity analysis comes down to a relatively simple problem of applying mature techniques. For example, assuming all functioning components' failure rates are the same constant at any time, and depend on the number of functioning components in the system, [2] models a load-sharing k-out-of-n:G system by a discrete-state, continuous-time homogeneous Markov chain, and solves its differential equations using inverse Laplace transforms.

This assumption may be applicable for some special cases, such as modeling soft errors in IC products, but is obviously not always applicable. Consider a brand-new unit, and a 10-year old one. In reality, we usually expect their failure rates to be different. The exponential failure distribution assumption, however, implies that the failure rate stays the same after 10 years usage. The main reason for the popularity of the above assumption is its mathematical tractability rather than accuracy. To tackle this problem, [7] studied a 1-out-of-2:G system with time-varying failure rates, whose failure distribution can be expressed in a general polynomial format. For a deeper understanding, [1] proposed an analytical model for load-sharing k-out-of-n:G system with a general lifetime distribution. This work modeled the load on a component as a vector $\mathbf{z}$, and the effect of load as $\psi(\mathbf{z})$. They simply assumed that the effect of load is multiplicative in time without any justification. Also, in this work, the load is assumed to be z_1 initially, and it progressively changes to z_i after the $i - 1$ failure occurs. Each load corresponds to a unique failure distribution. Thus, [1] does not involve warm standby, and the corresponding state transitions. Moreover, although this paper claimed that the proposed approaches can be easily generalized to $n > 2$ systems, its main contribution is limited to load-sharing 1-out-of-2:G systems, and gracefully degrading k-out-of-n:G systems with equal shared loads. To reduce the computational complexity induced by multiple integrals for such a system, [8] proposed a novel method that transfers the complex calculation to two-dimensional integrals. This work is quite efficient for analysing load-sharing gracefully degrading systems, but it is difficult to be applied to analyse standby redundant systems. More related works were summarized in [9]–[11].

Another issue relevant to this work is how to model the idle components. Such a component can be regarded as a cold, hot, or warm standby unit, which has a zero failure rate, the same failure rate as active components, or a failure rate in between, respectively. Because of simplicity, hot, and cold standby are commonly assumed states in many related papers, which are summarized in [12]. Also, the models mentioned above (e.g., [1], [7]) assumed every component in the system conforms to a single failure distribution, and hence can only be applied to analyse systems with hot standby components. As discussed earlier, warm standby is clearly a more reasonable description for the reliability analysis in many cases, and hence it is chosen in this paper. Reference [13] provided an in-depth discussion of warm standby. Later, most of the work in this area considered two-unit warm standby systems. For instance, [14] analysed a two-unit standby redundant system in which a module can alternate between cold and warm standby states; [15] analysed the two-unit standby system with general lifetime distributions. Similar to many previous work on this topic, it is difficult to extend these models to be applicable to general k-out-of-n systems because of the calculation complexity. As for k-out-of-n warm-standby systems, [16] provided a closed-form expression for the k-out-of-$n : G$ systems with warm standby components, but its analysis is again based on the assumption of constant failure rates in both active, and standby states.

In [17], the authors examined a 1-out-of-3 system that includes a warm, and a cold standby unit. Recently, another mixture model is presented in [18]. This work aimed to handle k-out-of-$(M+N)$:G repairable warm standby systems that con-

sist of two different types of components, each having its own state sets: M type 1 units, and N type 2 units. The operative failure rates, and standby failure rates of type 1, and 2 are different yet assumed to be exponential.

III. SYSTEM MODEL

In this section, we build our analytical model of lifetime reliability for a hybrid redundant load-sharing system. We first examine the behavior of a single component in such a system, and construct the unified reliability function accordingly. And then, we investigate the lifetime reliability of the entire system. Note that, because of frequent mode transitions between wait and processing states, the quantity of components in each state at a particular time point depends on not only previous failure events, but also current workload. In this context, to capture the variation of system reliability with time becomes a nontrivial problem.

A. Reliability of a Surviving Component

1) Component Behavior: Consider a component in the hybrid k-out-of-n:G system. Its initial state can be *active*, or *spare*, as shown in Fig. 1. The *spare* mode corresponds to the lowest power consumption, and no interaction with other components or controller. To be specific, a component in the spare mode does not undertake any task. The *active* mode includes two states depending on whether a component has tasks to perform: *processing*, and *wait*. We assume tasks are assigned by a controller, and then performed by a component independently; we ignore the cases that a few components cooperate on a task. If a task is assigned to a busy component (which means this component is processing tasks), it will be stored in a first-in-first-out (FIFO) buffer with infinite capacity. Once a component finishes its task at hand, it will fetch a new one from the queue immediately, unless the queue is empty. In that case, this component will switch from processing to a wait state. Upon receiving a new task, a waiting component will enter a processing state again. Note that, although a component does not process any task in the wait mode, different from the spare mode, it grows older. In reality, consider Intel's StrongARM SA-1100 processor [19] as an example. Its power consumption in a processing state is 400 mW, and that in the wait state is 50 mW rather than ∼0 mW, because some parts are still powered. When an active component fails, if there are some spare components in the system, one of them will be configured to active mode. From then on, it will serve as an active unit, and share workload with other parts until it fails, or the entire system breaks down.

2) Load-Sharing Model: As mentioned before, workload has significant influence on a component's reliability. Thus, it is necessary to model for each component the load ρ to be used in the reliability function construction. Recall that the tasks are assigned to all active components with equal probability. Thus, given the set of active components in the system $\mathcal{S}_1$, an active component's task branch-out probability is given by $p = (1/|\mathcal{S}_1|)$. For the sake of completeness, we also define $\mathcal{S}_2$ as
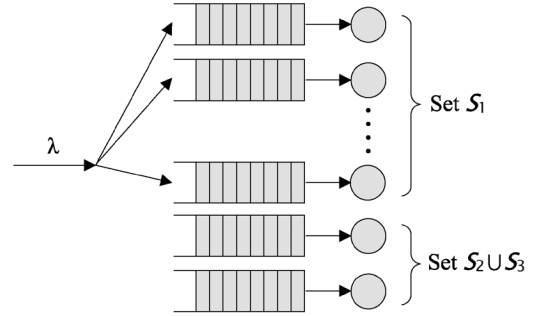

Fig. 2. Queueing model for task allocation in a load-sharing system.

the set of spare components, and $\mathcal{S}_3$ as the set of faulty components. The union of these three sets $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 = \mathcal{S}$ forms the entire system, where $|\mathcal{S}| = n$. Although our method could be easily extended to other queueing models (such as a $M^k/M/|\mathcal{S}_1|$ queue for central task assignment with bulk task arrival discussed in [3]), for ease of discussion, we focus on the distributed task allocation mechanism, and model each component as an $M/M/1$ queueing system (as shown in Fig. 2). To clarify, the task arrivals to the system are assumed to be Poisson with rate λ, and each component maintains a queue, where a FIFO buffer with infinite capacity is assumed. Because the probability of a task to be executed by an active component (i.e., p) is $1/|\mathcal{S}_1|$, the task inter-arrival time for an active component is exponentially distributed with mean $|\mathcal{S}_1|/\lambda$. Further, assuming the service time is exponentially distributed with mean $1/\mu$, the probability that an active component is occupied by a task, i.e., utilization ratio, is given by $\rho = (\lambda/\mu|\mathcal{S}_1|)$. For hybrid k-out-of-n:G systems, the utilization ratio of active components is constant $\lambda/m\mu$ in the standby phase, and then gradually increases to $\lambda/k\mu$ at the end of the degrading phase.

3) Unified Reliability Function: We now introduce our approach to combine the reliability functions in the processing state with those in the wait state, in a unified manner. The reliability functions in these two states can be regarded as having the same shape, but different scale parameters, which is defined as a value by which t is divided, because in many cases they can be distinguished by different aging rates.

First of all, we introduce the concept *cumulative time* in a certain state up to time t, which is defined as how long a component has spent in such a state from time 0 to t. Note that, as we ignore the aging effect in the spare state, we are only interested in the cumulative time in the processing, and wait states. Recall that some components initially serve as spare units. For the ease of discussion, we define another concept, a component's *birth time* t^b, as the time point when it begins to serve as an active component. Before birth time t^b, a component is in the cold standby mode, and has negligible failure rate. But after that, it alternates between wait and processing states until it breaks down. The components initially configured as active have birth time $t_0(= 0)$.

Theorem 1: Suppose the system has experienced exactly ℓ failures before time t, in the order of occurrence time at $t_1, t_2, \ldots, t_\ell$ (denoted as vector $\mathbf{t}_\ell$), for any surviving component with birth time t^b.

(a) Its cumulative time in the processing state up to t $(t > t_\ell)$ is

$$\psi_p(t, t^b, \mathbf{t}_\ell) \approx \begin{cases} \frac{\lambda}{m\mu}(t - t^b), & \ell \leq n - m \\ \frac{\lambda}{(n-\ell)\mu}t - \frac{\lambda}{m\mu}t^b \\ \quad - \sum_{j=n-m+1}^{\ell} \frac{\lambda}{(n-j+1)(n-j)\mu}t_j, & \ell > n - m \end{cases}$$

(1)

(b) Its cumulative time in the wait state up to t $(t > t_\ell)$ is

$$\psi_w(t, t^b, \mathbf{t}_\ell) \approx \begin{cases} \left(1 - \frac{\lambda}{m\mu}\right) \cdot (t - t^b), & \ell \leq n - m \\ \left(1 - \frac{\lambda}{(n-\ell)\mu}\right) t - \left(1 - \frac{\lambda}{m\mu}\right) t^b \\ \quad + \sum_{j=n-m+1}^{\ell} \frac{\lambda}{(n-j+1)(n-j)\mu}t_j, & \ell > n - m \end{cases}$$

(2)

The proof of the above theorem is illustrated in the Appendix.

With the cumulative time in two active states, we perform the integration as follows. The general reliability function which provides us the function shape is defined as $R(t, \theta)$, where θ is the general scale parameter. We drop the notation θ because of its generality, and refer to the general reliability function as $\mathcal{R}(t)$ in the rest of this paper. The functions in processing, and wait states can be therefore expressed as $R(t, \theta_p)$, and $R(t, \theta_w)$ respectively, where the scale parameters θ_p, and θ_w represent the wear-out rates under the two conditions. Typically, $\theta_w \geq \theta_p$. For the sake of simplification, they are abbreviated as $R_p(t)$, and $R_w(t)$. By unification, we achieve two simple relationships between the functions $R_p(t) = \mathcal{R}((\theta/\theta_p) \cdot t)$, and $R_w(t) = \mathcal{R}((\theta/\theta_w) \cdot t)$, which enable us to perform reliability function integration for any surviving component with the help of the cumulative time obtained by Theorem 1.

Theorem 2: Given a system has experienced ℓ failures which occur at $t_1, t_2, \ldots, t_\ell$ (i.e., vector $\mathbf{t}_\ell$) respectively, the probability that a certain component with birth time t^b survives at time t $(t > t_\ell)$ can be computed as

$$R(t, t^b | \mathbf{t}_\ell) = \mathcal{R}\left(\psi(t, t^b, \mathbf{t}_\ell)\right)$$

(3)

where,

$$\psi(t, t^b, \mathbf{t}_\ell) = \frac{\theta}{\theta_p}\psi_p(t, t^b, \mathbf{t}_\ell) + \frac{\theta}{\theta_w}\psi_w(t, t^b, \mathbf{t}_\ell).$$

(4)

Again, the proof is given in the Appendix.

B. Reliability of a Hybrid k-Out-of-n:G System

After resolving the lifetime reliability of a single component, we then move to study the lifetime reliability of the entire system, calculating its mean time to failure $MTTF^{sys}$. Let $P_{j,i}^{sys}(t)$ be the probability that the hybrid redundant system has j active components, and i good spare components at time t. As a functioning hybrid k-out-of-n:G system may have m active components with no more than $(n - m)$ good spare ones, or no less than k active components without good spare ones, the system reliability $P^{sys}(t)$ can be expressed by two summations:

$$P^{sys}(t) = \sum_{i=0}^{n-m} P_{m,i}^{sys}(t) + \sum_{j=k}^{m-1} P_{j,0}^{sys}(t).$$

(5)

Hence, the mean time to failure of the entire system

$$MTTF^{sys} = \int_0^\infty P^{sys}(t)\mathrm{d}t$$

(6)

can be written as

$$MTTF^{sys} = \int_0^\infty \left[\sum_{i=0}^{n-m} P_{m,i}^{sys}(t) + \sum_{j=k}^{m-1} P_{j,0}^{sys}(t)\right] \mathrm{d}t$$

$$= \int_0^\infty \sum_{i=0}^{n-m} P_{m,i}^{sys}(t)\mathrm{d}t + \int_0^\infty \sum_{j=k}^{m-1} P_{j,0}^{sys}(t)\mathrm{d}t.$$

(7)

For the sake of simplicity, let $P_{n-\ell}^{sys}(t)$ be the probability of a system containing $(n - \ell)$ good components, including both active components, and good spares. Equation (7) can therefore be rewritten as

$$MTTF^{sys} = \int_0^\infty \sum_{\ell=0}^{n-k} P_{n-\ell}^{sys}(t)\mathrm{d}t$$

(8)

$P_n^{sys}(t)$ is simply the probability that the system has no failures up to time t. As the $(n - m)$ spare components have zero failure rate, $P_n^{sys}(t)$ is the probability that all m active components do not fail from t_0 to t, i.e.,

$$P_n^{sys}(t) = R^m(t, t_0 | t_0).$$

(9)

If a failure occurs at time t_1, a spare component converts into active mode with a very short reconfiguration time (when compared to the system's lifetime). Before the next failure, this system consists of $(m-1)$ active components with birth time t_0, and one with birth time t_1. Thus, the conditional probability that the system contains $(n-1)$ good components at time t $(t > t_1)$, given a component with birth time t_0 fails at t_1, is given by

$$P_{n-1}^{sys}(t | t_1; (0)) = R^{m-1}(t, t_0 | t_1) \cdot R(t, t_1 | t_1).$$

(10)

Here, vector (0) in the notation $P_{n-1}^{sys}(t | t_1; (0))$ represents that the failure component has birth time t_0. Related notations will be formally introduced later.

Because the probability that an indicated component with birth time t_0 fails at time t_1 is $(\mathrm{d}/\mathrm{d}t)(1 - R(t, t_0 | t_0))|_{t=t_1}$, and that there are m such type of components, the probability that any one of them fails at t_1 is

$$g_n^{sys}(t_1) = m\frac{\mathrm{d}}{\mathrm{d}t}(1 - R(t, t_0 | t_0))|_{t=t_1}.$$

(11)

The event that the system experiences exactly one failure up to t is a union of a set of continuous elementary events in which a failure occurred in an infinitesimal interval $\mathrm{d}t_1$ at time t_1, and the probability for this is $P_{n-1}^{sys}(t | t_1; (0))g_n^{sys}(t_1)\mathrm{d}t_1$. By the theorem of total probability, the unconditional probability can be obtained by integration over t_1, i.e.,

$$P_{n-1}^{sys}(t) = \int_0^t R^{m-1}(t, t_0 | t_1) \cdot R(t, t_1 | t_1)$$

$$\cdot m\frac{\mathrm{d}}{\mathrm{d}t}(1 - R(t, t_0 | t_0))|_{t=t_1}\mathrm{d}t_1$$

(12)

When $\ell \geq 2$, because there might be more than one possible case of failures, the expression of $P_{n-\ell}^{sys}(t)$ becomes more complex. To estimate it, we use two $1 \times \ell$ row vectors $\mathbf{t}_\ell = (t_1, t_2, \cdots, t_\ell)$, and $\mathbf{x}_\ell = (x_1, x_2, \cdots, x_\ell)$ to capture the dominant characteristic of ℓ failures. t_i represents the occurrence time of the ith failure. x_i indicates the birth time index of the ith failure component, meaning that, if the ith failure component has birth time t_r, the corresponding birth time index x_i is r. In general, there are 2ℓ elements

$$\left\{ \begin{array}{cccc} t_1 & t_2 & \cdots & t_\ell \\ x_1 & x_2 & \cdots & x_\ell \end{array} \right\} \tag{13}$$

each column corresponding to a failure. The occurrence time satisfies the constraints that

$$t_1 < t_2 < \cdots < t_\ell \tag{14}$$

Because the ith failure component must convert into active mode before t_i, its birth time $t^b \in \{t_0, t_1, \cdots, t_{i-1}\}$. On the other hand, all components start to serve as active ones on or before t_{n-m}. Therefore, the birth time indices satisfy the constraints that

$$x_i = 0, 1, \cdots, \min(i - 1, n - m) \tag{15}$$

The quantity of i in the first j elements of vector $\mathbf{x}_\ell$ is denoted as $\pi_{i,j}$. Because only one component turns into the active mode after each failure, there is essentially only one component with birth time t_i in the entire system if $i \neq 0$. In addition, the system contains no more than m components with birth time t_0. Hence, the birth time indices should also satisfy

$$\forall x_i, x_j \neq 0 : x_i \neq x_j, \quad \text{and} \quad \pi_{0,\ell} \leq m \tag{16}$$

For a possible $\mathbf{x}_\ell$, let $\mathbf{x}_\ell^{(r)} = (x_1, x_2, \cdots, x_r)$ be the r-dimensional subvector of $\mathbf{x}_\ell$. Similarly, let $\mathbf{t}_\ell^{(r)} = (t_1, t_2, \cdots, t_r)$ be the corresponding r-dimensional subvector of $\mathbf{t}_\ell$. That is, for any possible case of failures, its first r failures can always be described by $\mathbf{x}_\ell^{(r)}$, and $\mathbf{t}_\ell^{(r)}$. Further, we define

$$f\left(t_r, t^b | \mathbf{t}_\ell^{(r-1)}\right) \equiv \frac{\mathrm{d}}{\mathrm{d}t}\left(1 - R\left(t, t^b | \mathbf{t}_\ell^{(r-1)}\right)\right)\Big|_{t=t_r}. \tag{17}$$

We therefore express the probability that the system contains $(n-\ell)$ good components having ℓ failures whose characteristics can be described by $\mathbf{t}_\ell$, and $\mathbf{x}_\ell$ as

$$\begin{aligned} P_{n-\ell}^{sys}(t, \mathbf{t}_\ell; \mathbf{x}_\ell) &= P_{n-\ell}^{sys}(t|\mathbf{t}_\ell; \mathbf{x}_\ell)g_n^{sys}(t_1; x_1) \\ &\quad \cdot g_{n-1}^{sys}\left(t_2; x_2|\mathbf{t}_\ell^{(1)}; \mathbf{x}_\ell^{(1)}\right) \\ &\quad \cdots g_{n-\ell+1}^{sys}\left(t_\ell; x_\ell|\mathbf{t}_\ell^{(\ell-1)}; \mathbf{x}_\ell^{(\ell-1)}\right) \end{aligned} \tag{18}$$

where $P_{n-\ell}^{sys}(t|\mathbf{t}_\ell; \mathbf{x}_\ell)$ is the conditional probability that the system contains $(n - \ell)$ good components at time t, given the past ℓ failures described by $\mathbf{t}_\ell$, and $\mathbf{x}_\ell$. $g_{n-r+1}^{sys}(t_r; x_r|\mathbf{t}_\ell^{(r-1)}; \mathbf{x}_\ell^{(r-1)})$ denotes the probability that, in the system containing $(n - r + 1)$ good components, a

component with birth time t_{x_r} fails at time t_r given the past $(r - 1)$ failures can be described by $\mathbf{t}_\ell^{(r-1)}$ and $\mathbf{x}_\ell^{(r-1)}$.

After ℓ failures, the number of good components with birth time t_0 is $(m - \pi_{0,\ell})$; and that with birth time $t_i (i \neq 0)$ is $(1 - \pi_{i,\ell})$. Consequently, the conditional probability $P_{n-\ell}^{sys}(t|\mathbf{t}_\ell; \mathbf{x}_\ell)$ can be computed as

$$P_{n-\ell}^{sys}(t|\mathbf{t}_\ell \mathbf{x}_\ell) = R^{m-\pi_{0,\ell}}(t, t_0|\mathbf{t}_\ell) \cdot \prod_{i=1}^{\min(n-m,\ell)} R^{1-\pi_{i,\ell}}(t, t_i|\mathbf{t}_\ell). \tag{19}$$

As for the computation of $g_{n-r+1}^{sys}(t_r; x_r|\mathbf{t}_\ell^{(r-1)}; \mathbf{x}_\ell^{(r-1)})$, we consider two cases: (i) the rth failure component has birth time t_0, or (ii) it has birth time t_i $(i \neq 0)$. For the first case, the probability that an indicated component with birth time t_0 fails at time t_r is $f(t_r, t_0|\mathbf{t}_\ell^{(r-1)})$, and there are $(m - \pi_{0,r-1})$ surviving components with birth time t_0 in such a system. For the second case, there is only one good component with birth time t_{x_r}, and the probability for its failure at t_r is $f(t_r, t_{x_r}|\mathbf{t}_\ell^{(r-1)})$. Therefore, we have

$$g_{n-r+1}^{sys}\left(t_r; x_r|\mathbf{t}_\ell^{(r-1)}; \mathbf{x}_\ell^{(r-1)}\right)$$
$$= \begin{cases} (m - \pi_{0,r-1}) \cdot f\left(t_r, t_0|\mathbf{t}_\ell^{(r-1)}\right), & x_r = 0 \\ f\left(t_r, t_{x_r}|\mathbf{t}_\ell^{(r-1)}\right), & \text{otherwise} \end{cases} \tag{20}$$

After analysing a single failure case, we can now compute $P_{n-\ell}^{sys}(t)$ for all possible cases. Denote the set of all possible $\mathbf{x}_\ell$ as $\mathcal{X}_\ell$. Because $t_0 < t_1 < \cdots < t_\ell < t$, we have

$$\begin{aligned} P_{n-\ell}^{sys}(t) &= \int_0^t \mathrm{d}t_1 \int_{t_1}^t \mathrm{d}t_2 \int_{t_2}^t \mathrm{d}t_3 \\ &\quad \cdots \int_{t_{\ell-1}}^t \mathrm{d}t_\ell \sum_{\mathbf{x}_\ell \in \mathcal{X}_\ell} P_{n-\ell}^{sys}(t, \mathbf{t}_\ell; \mathbf{x}_\ell) \end{aligned} \tag{21}$$

Interchanging the integration limits yields

$$\begin{aligned} P_{n-\ell}^{sys}(t) &= \int_0^t \mathrm{d}t_\ell \int_0^{t_\ell} \mathrm{d}t_{\ell-1} \int_0^{t_{\ell-1}} \mathrm{d}t_{\ell-2} \\ &\quad \cdots \int_0^{t_2} \mathrm{d}t_1 \sum_{\mathbf{x}_\ell \in \mathcal{X}_\ell} P_{n-\ell}^{sys}(t, \mathbf{t}_\ell; \mathbf{x}_\ell) \end{aligned} \tag{22}$$

IV. SPECIAL CASES

In this section, we simplify the proposed models under the assumption that the system is a gracefully degrading one (Case I), or a standby redundant one (Case II) respectively, for verification purposes. Further, we assume a constant failure rate in both cases, resulting in exactly the same results as previous work. We also present a special case in Case III to demonstrate the detailed analytical procedure by using the proposed approach.

A. Case I: Gracefully Degrading System

When $n = m$, the system discussed above becomes a load-sharing k-out-of-n:G gracefully degrading system. In this case, there is only one possible failure case for any ℓ, i.e.,

$$\mathcal{X}_\ell = \{\underbrace{(0,\cdots,0)}_{\ell}\}.$$

Thus, dropping $\mathbf{x}_\ell$ from notations, and rewriting (22), (18)–(20), yields

$$P^{sys}_{n-\ell}(t) = \int_0^t dt_\ell \int_0^{t_\ell} dt_{\ell-1} \int_0^{t_{\ell-1}} dt_{\ell-2}\cdots \int_0^{t_3} dt_2 \int_0^{t_2} dt_1 P^{sys}_{n-\ell}(t, \mathbf{t}_\ell)$$

where,

$$P^{sys}_{n-\ell}(t, \mathbf{t}_\ell) = P^{sys}_{n-\ell}(t|\mathbf{t}_\ell) g^{sys}_n(t_1) g^{sys}_{n-1}\left(t_2|\mathbf{t}^{(1)}_\ell\right)$$
$$\cdots g^{sys}_{n-\ell+1}\left(t_\ell|\mathbf{t}^{(\ell-1)}_\ell\right)$$
$$P^{sys}_{n-\ell}(t|\mathbf{t}_\ell) = R^{n-\ell}(t, t_0|\mathbf{t}_\ell)$$
$$g^{sys}_{n-r+1}\left(t_r|\mathbf{t}^{(r-1)}_\ell\right) = (n-r+1)f\left(t_r, t_0|\mathbf{t}^{(r-1)}_\ell\right) \quad (23)$$

If we further assume that all components have the same constant failure rate in both processing and wait states, then

$$\mathcal{R}(t) = e^{-\frac{t}{\theta}}, \quad R_p(t) = R_w(t) = e^{-\frac{t}{\theta_p}},$$

and the computation is greatly simplified. For the ease of expression, let $\eta = (1/\theta)$. By Theorems 1, and 2, we have

$$R(t, t_0|\mathbf{t}_\ell) = \mathcal{R}\left(\psi(t, t_0, \mathbf{t}_\ell)\right)$$
$$= \mathcal{R}\left(\frac{\theta}{\theta_p}\psi_p(t, t_0, \mathbf{t}_\ell) + \frac{\theta}{\theta_w}\psi_w(t, t_0, \mathbf{t}_\ell)\right)$$
$$= \mathcal{R}\left(\frac{\theta}{\theta_p}\cdot(\psi_p(t, t_0, \mathbf{t}_\ell) + \psi_w(t, t_0, \mathbf{t}_\ell))\right)$$
$$= \mathcal{R}\left(\frac{\theta}{\theta_p}\cdot t\right) = e^{-\eta t}$$

Thus, by (17),

$$f\left(t_r, t_0|\mathbf{t}^{(r-1)}_\ell\right) = \frac{d}{dt}(1 - e^{-\eta t})|_{t=t_r} = \eta e^{-\eta t_r}$$

Substituting them into (23), we obtain

$$P^{sys}_{n-\ell}(t) = e^{-(n-\ell)\eta t}\binom{n}{\ell}(1 - e^{-\eta t})^\ell.$$

With (8), we obtain

$$MTTF^{sys} = \int_0^\infty \sum_{\ell=0}^{n-k} P^{sys}_{n-\ell}(t)dt = \frac{1}{\eta}\sum_{\ell=k}^n \frac{1}{\ell}.$$

This is the same results shown in previous works [20].

B. Case II: Standby Redundant System

When $k = m$, this system becomes a load-sharing m-out-of-n:G standby redundant system. In this case, the number of failure components ℓ in a functioning system cannot exceed $(n - m)$; otherwise the system fails. Therefore, a surviving component's cumulative time in either the processing or wait state only depends on its birth time t^b, i.e.,

$$\psi_p(t, t^b, \mathbf{t}_\ell) \approx \frac{\lambda}{m\mu}(t - t^b),$$

and

$$\psi_w(t, t^b, \mathbf{t}_\ell) \approx \left(1 - \frac{\lambda}{m\mu}\right)(t - t^b).$$

Again, if we further assume the failure rate to be constant, and $\theta_p = \theta_w = (1/\eta)$, we obtain

$$R(t, t^b|\mathbf{t}_\ell) = e^{-\eta(t-t^b)}$$
$$f\left(t_r, t^b|\mathbf{t}^{(r-1)}_\ell\right) = \eta e^{-\eta(t_r-t^b)}.$$

Thus, the system reliability at time t becomes

$$P^{sys}(t) = e^{-m\eta t}\sum_{\ell=0}^{n-m} \frac{(m\eta t)^\ell}{\ell!},$$

and mean time to failure of this system is

$$MTTF^{sys} = \frac{n - m + 1}{m\eta}$$

which is the same result as that in [10].

C. Case III: 1-Out-of-3:G System With $m = 2$

In this system, $k = 1$, $m = 2$, and $n = 3$. By (8), it is necessary to determine $P^{sys}_{3-\ell}(t)$ ($0 \le \ell \le 2$) to achieve $MTTF^{sys}$, where $P^{sys}_{3-\ell}(t)$ can be defined in two equivalent ways: first, the probability that the system contains $(3 - \ell)$ good components at time t; and second, the probability that exactly ℓ failures have happened in the system up to time t.

Initially, two components are in the active mode, with the remaining one in the spare mode. The active components share the workload of the entire system. The traffic intensity of each component is $\lambda/2\mu$. If no failures occur in the system up to time t, the cumulative time of an active component remaining in the processing, and wait state are $(\lambda/2\mu)t$, and $(1 - (\lambda/2\mu))t$ respectively. By Theorem 2, the probability that an active component survives at time t is given by

$$R(t, t_0|\mathbf{t}_0) = \mathcal{R}\left(\psi(t, t_0, \mathbf{t}_0)\right)$$
$$= \mathcal{R}\left(\frac{\theta}{\theta_p}\cdot\frac{\lambda}{2\mu}t + \frac{\theta}{\theta_w}\cdot\left(1 - \frac{\lambda}{2\mu}\right)t\right).$$

Substituting this reliability function into (9) yields

$$P^{sys}_3(t) = R^2(t, t_0|\mathbf{t}_0) = \mathcal{R}^2\left(\frac{\theta}{\theta_p}\cdot\frac{\lambda}{2\mu}t + \frac{\theta}{\theta_w}\cdot\left(1 - \frac{\lambda}{2\mu}\right)t\right). \quad (24)$$

After the first failure, which may occur on either active component with the same failure rate, the spare component is activated; and its traffic intensity is $\lambda/2\mu$. The traffic intensity of the surviving active component remains at the past level. There is only one possible case of failure, that is, $\mathbf{t}_1 = (t_1)$, and $\mathbf{x}_1 = (0)$. Suppose exactly one failure occurs up to time t, and the occurrence time is denoted as t_1. Then, by Theorem 1, the

cumulative time of the component with birth time t_0 in the processing, and wait state are $(\lambda/2\mu)t$, and $(1 - (\lambda/2\mu))t$ respectively; while those time intervals of the component with birth time t_1 are $(\lambda/2\mu)(t - t_1)$, and $(1 - (\lambda/2\mu))(t - t_1)$ respectively. Using Theorem 2, and substituting into (12) yields

$$P_2^{sys}(t) = \int_0^t R(t, t_0|\mathbf{t}_1) \cdot R(t, t_1|\mathbf{t}_1)$$
$$\cdot 2 \cdot \frac{\mathrm{d}}{\mathrm{d}t}(1 - R(t, t_0|\mathbf{t}_0))|_{t=t_1} \mathrm{d}t_1 \quad (25)$$

where,

$$R(t, t_0|\mathbf{t}_1) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot \frac{\lambda}{2\mu}t + \frac{\theta}{\theta_w} \cdot \left(1 - \frac{\lambda}{2\mu}\right)t\right),$$

and

$$R(t, t_1|\mathbf{t}_1) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot \frac{\lambda}{2\mu}(t - t_1) + \frac{\theta}{\theta_w} \cdot \left(1 - \frac{\lambda}{2\mu}\right)(t - t_1)\right).$$

After the first failure, two components in the system have different ages, and hence different failure rates. When the system has experienced two failures up to t, there is only one good component left in the system at time t. This component's traffic intensity from t_2 to t is λ/μ. By using vector $\mathbf{t}_\ell$, and vector $\mathbf{x}_\ell$ to describe all possible cases of two failures, we obtain two cases:

$$\begin{Bmatrix} t_1 & t_2 \\ 0 & 0 \end{Bmatrix}, \quad \text{and} \quad \begin{Bmatrix} t_1 & t_2 \\ 0 & 1 \end{Bmatrix}.$$

For the first case, the number of surviving components with birth time t_0 is $(m - \pi_{0,2}) = 0$, and that with birth time t_1 is $(1 - \pi_{1,2}) = 1$, meaning that the only good component left after two failures has birth time t_1. This component's cumulative time in processing, and wait states are $\psi_p(t, t_1, \mathbf{t}_2) = (\lambda/\mu)t - (\lambda/2\mu)t_1 - (\lambda/2\mu)t_2$, and $\psi_w(t, t_1, \mathbf{t}_2) = (1 - (\lambda/\mu))t - (1 - (\lambda/2\mu))t_1 + (\lambda/2\mu)t_2$ respectively. Subvector $\mathbf{x}_2^{(1)} = (0)$. By (19), and (20), we obtain

$$P_1^{sys}(t|\mathbf{t}_2; \mathbf{x}_2) = R(t, t_1|\mathbf{t}_2)$$
$$g_3^{sys}(t_1; x_1) = 2f\left(t_1, t_0|\mathbf{t}_2^{(0)}\right)$$
$$g_2^{sys}\left(t_2; x_2|\mathbf{t}_2^{(1)}; \mathbf{x}_2^{(1)}\right) = f\left(t_2, t_0|\mathbf{t}_2^{(1)}\right).$$

For the second case, the remaining component has birth time t_0, and subvector $\mathbf{x}_2^{(1)}$ is also (0). Similar to the analysis of the first case, we obtain $\psi_p(t, t_0, \mathbf{t}_2) = (\lambda/\mu)t - (\lambda/2\mu)t_2$, $\psi_w(t, t_0, \mathbf{t}_2) = (1 - (\lambda/\mu))t + (\lambda/2\mu)t_2$, and

$$P_1^{sys}(t|\mathbf{t}_2; \mathbf{x}_2) = R(t, t_0|\mathbf{t}_2)$$
$$g_3^{sys}(t_1; x_1) = 2f\left(t_1, t_0|\mathbf{t}_2^{(0)}\right)$$
$$g_2^{sys}\left(t_2; x_2|\mathbf{t}_2^{(1)}; \mathbf{x}_2^{(1)}\right) = f\left(t_2, t_1|\mathbf{t}_2^{(1)}\right)$$

With (22), we have

$$P_1^{sys}(t) = \int_0^t \mathrm{d}t_2 \int_0^{t_2} \mathrm{d}t_1 \sum_{\mathbf{x}_\ell \in \mathcal{X}_\ell} P_1^{sys}(t, \mathbf{t}_2; \mathbf{x}_2)$$
$$= \int_0^t \mathrm{d}t_2 \int_0^{t_2} \mathrm{d}t_1 \left[R(t, t_1|\mathbf{t}_2) \cdot 2f\left(t_1, t_0|\mathbf{t}_2^{(0)}\right) \right.$$
$$\cdot f\left(t_2, t_0|\mathbf{t}_2^{(1)}\right) + R(t, t_0|\mathbf{t}_2)$$
$$\left. \cdot 2f\left(t_1, t_0|\mathbf{t}_2^{(0)}\right) \cdot f\left(t_2, t_1|\mathbf{t}_2^{(1)}\right) \right].$$
$$(26)$$

Combining (24)–(26), we finally obtain the lifetime reliability, and mean time to failure of this system:

$$P^{sys}(t) = P_3^{sys}(t) + P_2^{sys}(t) + P_1^{sys}(t)$$
$$MTTF^{sys} = \int_0^\infty P^{sys}(t)\mathrm{d}t$$

V. NUMERICAL RESULTS

The relentless scaling of CMOS technology has enabled the integration of a great amount of embedded processor cores on a single silicon die. Because of their advantages in power-efficiency, and short time-to-market, such large-scale manycore systems have received lots of attention from both industry [21]–[23], and academia [24], [25]. At the same time, the ever-increasing on-chip power density accelerates the aging effects caused by various failure mechanisms, making their lifetime reliability a serious concern [26], [27]. As a consequence, designers typically introduce on-chip redundant cores to make the product fault-tolerant (e.g., [21], [28]). In this section, we present numerical results obtained with Monte Carlo integration based on our closed-form expressions for the lifetime reliability analysis of manycore systems with different redundancy schemes, and various workloads. In particular, we first analyse the system with the proposed modeling method to achieve the closed-form expressions, and then resort to the Monte Carlo method to numerically approximate the values of multiple integrals in the resulting expressions.

A. Experimental Setup

Two widely-used non-exponential lifetime distributions are assumed in the experiments: Weibull, and linear failure rate. These reliability functions can be written as $\mathcal{R}(t) = e^{-(t/\theta)^\beta}$, and $\mathcal{R}(t) = e^{-a \cdot (t/\theta) - b \cdot (t/\theta)^2}$, respectively. The scale parameters are different in the processing state (θ_p), and wait state (θ_w). Typically, they are in units of years or hours. Clearly, θ_p is no more than θ_w. The property of the Weibull distribution, whose failure rate function $h(t) = (\beta/\theta) \cdot (t/\theta)^{\beta-1}$, highly depends on its shape parameter β. We set $\beta = 4$ in our experiment, implying an increasing failure rate with respect to time. The linear failure rate distribution has the hazard function $h(t) = (a/\theta) + (2b/\theta^2) \cdot t$, where $a, b \geq 0$. When $b = 0$, it reduces to an exponential distribution; when $a = 0$, it becomes a Rayleigh distribution. Different from the Weibull distribution,

TABLE I
LIFETIME RELIABILITY OF MANYCORE SYSTEM WITH CONSTANT FAILURE RATE

| $u+v$	Sojourn Time (years)					$MTTF^{sys}$
	0-Failure State	1-Failure State	2-Failure State	3-Failure State	4-Failure State	
0 + 0	0.2188	—	—	—	—	0.2188
1 + 0	0.2121	0.2188	—	—	—	0.4309
0 + 1	0.2188	0.2188	—	—	—	0.4376
2 + 0	0.2059	0.2121	0.2188	—	—	0.6368
1 + 1	0.2121	0.2121	0.2188	—	—	0.6430
0 + 2	0.2188	0.2188	0.2188	—	—	0.6564
3 + 0	0.2000	0.2059	0.2121	0.2188	—	0.8368
2 + 1	0.2059	0.2059	0.2121	0.2188	—	0.8427
1 + 2	0.2121	0.2121	0.2121	0.2188	—	0.8551
0 + 3	0.2188	0.2188	0.2188	0.2188	—	0.8752
4 + 0	0.1944	0.2000	0.2059	0.2121	0.2188	1.0312
3 + 1	0.2000	0.2000	0.2059	0.2121	0.2188	1.0368
2 + 2	0.2059	0.2059	0.2059	0.2121	0.2188	1.0486
1 + 3	0.2121	0.2121	0.2121	0.2121	0.2188	1.0672
0 + 4	0.2188	0.2188	0.2188	0.2188	0.2188	1.0940
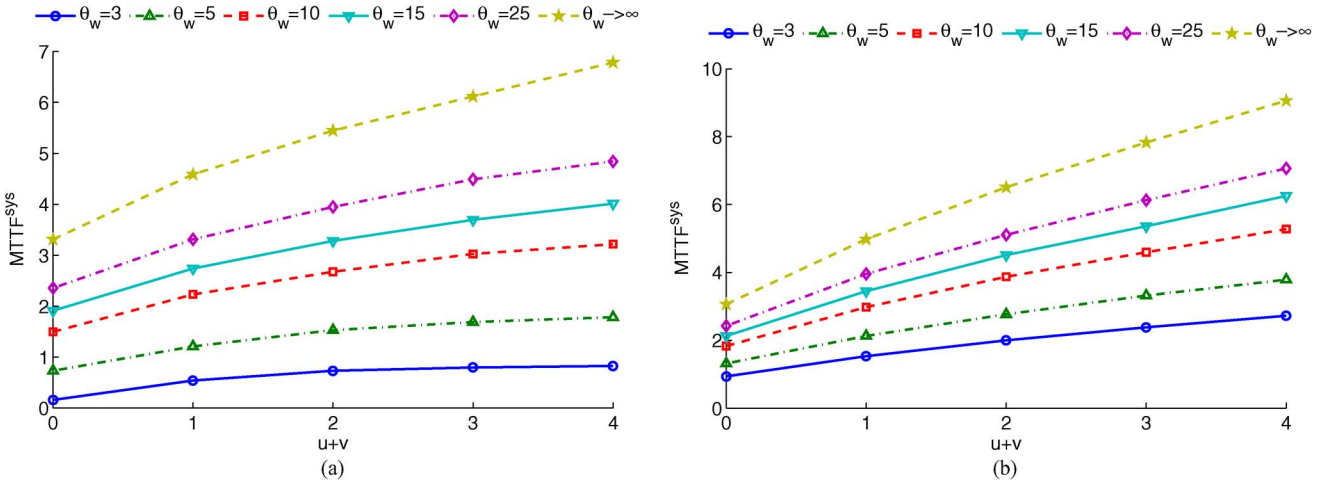
Fig. 3. Lifetime enhancement of manycore system. (a) Weibull distribution. (b) Linear failure rate distribution.

the linear failure rate distribution may have non-zero failure rate at $t = 0$. We set $a = 0.03$, and $b = 0.15$ in our experiments.

The number of embedded cores in the manycore system is set to be $(32 + u + v)$, meaning that the system has $(32 + u)$ active cores, while the remaining v cores are put aside at time zero. If an active core is detected to be faulty, the system replaces it with a spare one until there are no spares in the system. Then the system enters its degrading phase until the number of good cores is less than 32. In other words, this system has parameters $n = 32 + u + v$, $m = 32 + u$, and $k = 32$. It becomes a load-sharing k-out-of-n:G gracefully degrading system when $v = 0$, and a standby redundant system when $u = 0$.

B. Experimental Results and Discussion

We discuss an issue that attracts attention: how much benefit can be expected from adding redundant cores into a manycore system? As shown in Table I, if we assume an exponential lifetime distribution, the sojourn time only depends on the number of active cores in the system, s-independent of the aging effect. With this assumption, we expect significant lifetime enhancement using redundant cores, as shown in the last column of Table I, where we set $\theta_w = \theta_p = 7$. For instance, the system lifetime increases from 0.2188 to around 0.64 by employing two

redundant cores. Clearly, this result does not conform to our common sense. In practice, IC products experience increasing failure rates over their life cycles. In this sense, a Weibull or linear failure rate distribution could be a better approximation of such wearout effect, and bring us more reasonable results.

Fig. 3 shows the lifetime enhancement achieved by redundant cores with Weibull and linear failure rate distributions for $\theta_p = 3$. The same quantity of redundant cores could have different redundant schemes, and hence result in $MTTF^{sys}$ deviations. In these figures, we simply plot the maximum $MTTF^{sys}$ achieved with the given $(u + v)$ because the area overhead depends on the core quantity only. Needless to say, the lifetime reliability of manycore systems is enhanced with redundant cores at the cost of area overhead. At the same time, the lifetime improvement gradually slows down with the increase of $(u+v)$. For example, see the curve for the case with $\theta_w = 25$ in Fig. 3(a). The addition of first redundant core results in 26.66% lifetime extension; those of the second, third, and fourth one lead to 17.85%, 14.46%, and 9.34% extension, respectively. Consequently, designers need to set $(u+v)$ with an appropriate value to tradeoff area overhead with lifetime extension, rather than set $(u + v)$ as large as possible under the area overhead constraints.

TABLE II
LIFETIME RELIABILITY OF MANYCORE SYSTEM WITH NON-EXPONENTIAL LIFETIME DISTRIBUTION (WEIBULL)

| $u+v$	Sojourn Time (years)					$MTTF^{sys}$
	0-Failure State	1-Failure State	2-Failure State	3-Failure State	4-Failure State	
0+0	2.2039	—	—	—	—	2.2039
1+0	2.2153	0.5574	—	—	—	2.7727
0+1	2.2039	0.5617	—	—	—	2.7656
2+0	2.2260	0.5601	0.3593	—	—	3.1453
1+1	2.2153	0.5643	0.3581	—	—	3.1377
0+2	2.2039	0.5617	0.3558	—	—	3.1213
3+0	2.2359	0.5626	0.3642	0.2398	—	3.4025
2+1	2.2260	0.5667	0.3864	0.2578	—	3.4368
1+2	2.2153	0.5643	0.3613	0.2446	—	3.3855
0+3	2.2039	0.5617	0.3558	0.2375	—	3.3588
4+0	2.2452	0.5649	0.3633	0.2672	0.1555	3.5961
3+1	2.2359	0.5689	0.3765	0.2754	0.1652	3.6219
2+2	2.2260	0.5667	0.3663	0.2657	0.1750	3.5995
1+3	2.2153	0.5643	0.3613	0.2560	0.1480	3.5426
0+4	2.2039	0.5617	0.3558	0.2375	0.1069	3.4658
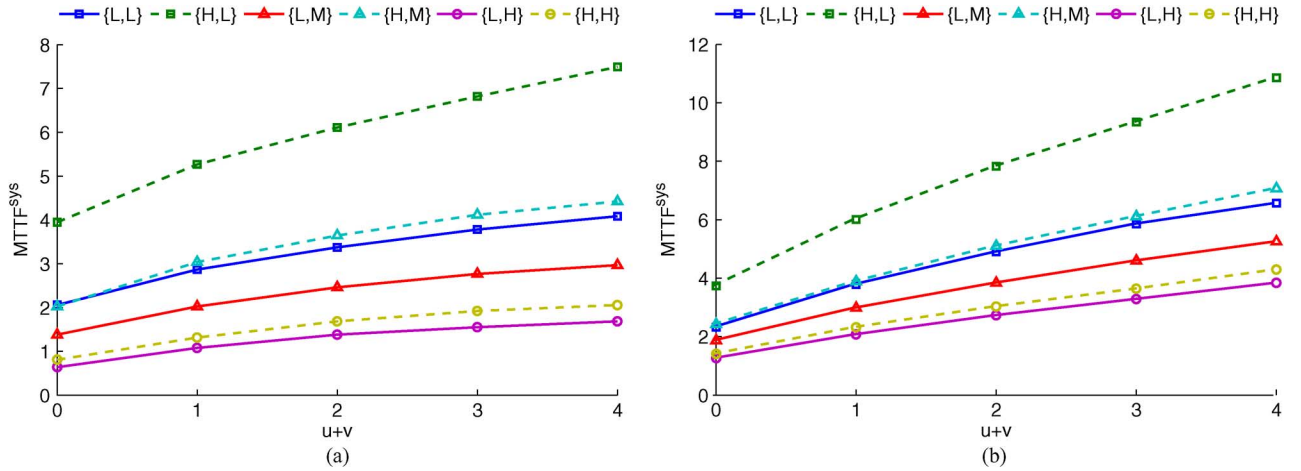
Fig. 4.   Variation in lifetime reliability with workload. (a) Weibull distribution. (b) Linear failure rate distribution.

We also need to place emphasis on the phenomenon that lifetime reliability highly depends on the scale parameters (i.e., θ_p, and θ_w) in the reliability functions. There are two extreme cases. The first is $\theta_p = \theta_w$, meaning that there is no difference between wait state and processing state in terms of the reliability function. It is essentially the so-called hot standby scheme. The second is $\theta_w \to \infty$, implying that an embedded core in the wait state is a cold standby component, and cannot fail. Taking the Linear failure rate distribution as an example, $MTTF^{sys}$ values in these two cases are 6.4324, and 1.6690 with four redundant cores, as shown in Fig. 3(a). Due to this huge gap, we advocate to reexamine the conclusions made under the hot or cold standby assumption, and deal carefully with the components in the wait state.

A closer observation for various redundant schemes is shown in Table II, setting $\theta_p = 3$, $\theta_w = 10$, and $(\lambda/\mu) = 10$. Due to the increasing failure rate, the manycore system contains no faulty cores in most of its lifetime, especially for systems suffering from severe wearout effects. For example, consider the $(32 + 2 + 1)$ manycore system. Its sojourn time in the 0-failure state is 2.2260 years, while the expected value of its whole lifetime is 3.4368 years. From this perspective, one core's failure may imply the entire system is old, and we cannot expect much

residual useful lifetime. Another interesting observation is, given the number of redundant cores, the maximum $MTTF^{sys}$ could occur with a hybrid redundant scheme. For example, the manycore system with four redundant cores achieves its maximum lifetime when $u = 3$, and $v = 1$.

Finally, we study the influence of workloads on the lifetime reliability, and plot the results for $\theta_p = 3$ in Fig. 4, where the values of θ_w, and λ/μ are labeled as $\{A, B\}$ in the legend; and $A = \{L(10), H(25)\}$, and $B = \{L(5), M(10), H(20)\}$. As can be observed, the workload has significant influence on the lifetime reliability of manycore systems, and should be paid much attention by designers. That is, with the increase of workload λ/μ, the system lifetime is significantly shortened, yet the scale of decrease in $MTTF^{sys}$ is much smaller than that of the increase in workload. Consider $\theta_w = 10$, and $u + v = 4$ in Fig. 3(b) as an example. The $MTTF^{sys}$ values are 4.5808, 3.6219, and 2.4750 for $(\lambda/\mu) = 5$, 10, and 20, respectively. We attribute this phenomenon to the wearout effects of warm standby.

VI. CONCLUSION

In this work, we present a general closed-form expression for the lifetime reliability of load-sharing k-out-of-n:G hybrid re-

dundant systems. The load assigned to the system is modeled using queueing theory. We integrate the various failure distributions for components in different operational states into our analytical model with their corresponding aging effects, which are then used to estimate the lifetime reliability of the entire system. Finally, the practical applicability of the proposed model is verified with several special cases, and numerical experiments.

APPENDIX

Proof of Theorem 1

Case 1) *Case I.$\ell \leq n - m$.* In this case, a surviving component's birth time $t^b \in \{t_0, t_1, \cdots, t_\ell\}$. Before t^b, the component serves as a cold standby. After that, it alternates between the processing and wait states. Because the number of faulty components is no more than $(n - m)$, there are exactly m active components in the system from t^b to t. According to queueing theory, the utilization ratio of an $M/M/1$ queue is λ_q/μ, where λ_q is the arrival rate of this queue. In an equally load-sharing system with m active components, $\lambda_q = (\lambda/m)$. Because the time scale of the system lifetime is usually much larger than that of task processing, the cumulative time in the processing state can be approximated as $(\lambda/m\mu)(t - t^b)$. In addition, because the component is in either a processing or wait state from t^b to t, the cumulative time in the wait state $\psi_w(t, t^b, \mathbf{t}_\ell) = (t - t^b) - \psi_p(t, t^b, \mathbf{t}_\ell) = (1 - (\lambda/m\mu))(t - t^b)$.

Case 2) *Case II.$\ell > n - m$.* It is important to note that the birth time of any component must be no later than t_{n-m}, because any surviving component at time t $(t > t_\ell)$ has been configured as active at or before t_{n-m}. Therefore, the birth time $t^b \in \{t_0, t_1, \cdots, t_{n-m}\}$. From t^b to t_{n-m+1}, there are m active components in the system. Thus, the utilization ratio in this period is $\lambda/m\mu$. Hence, a component's cumulative time in the processing state from t^b to t_{n-m+1} is $(\lambda/m\mu)(t_{n-m+1} - t^b)$. From t_j to t_{j+1}, $(n - m + 1 \leq j \leq \ell - 1)$, the system contains $(n - j)$ active components. By the same argument, the component's cumulative time in the processing state from t_j to t_{j+1} is $(\lambda/(n - j)\mu)(t_{j+1} - t_j)$. Similarly, from t_ℓ to t, it is $(\lambda/(n - \ell)\mu)(t - t_\ell)$. Summing all these $(\ell - n + m + 1)$ terms up results in (1). As for $\psi_w(t, t^b, \mathbf{t}_\ell)$, we compute it by $(t - t^b) - \psi_p(t, t^b, \mathbf{t}_\ell)$, similar to the computation for Case I.

$\square$

Proof of Theorem 2

As $[t^b, t]$ is a closed interval, we can partition it into d subintervals according to state transitions: $t^b = T_0 < T_1 < T_2 <$

$\cdots < T_d = t$. To be specific, at any T_i $(1 \leq i \leq d - 1)$, the component converts from processing mode to wait mode, or opposite.

The initial reliability of a component is given by

$$R(T_0) = R(t^b) = \mathcal{R}(0).$$

Then, for the first sub-interval $[T_0, T_1)$, suppose the component does not have tasks to process in this time interval. Then the reliability at time τ $(T_0 \leq \tau < T_1)$ is given by

$$R(\tau) = \mathcal{R}\left(\frac{\theta}{\theta_w} \cdot (\tau - T_0)\right).$$

By this equation, at the end of this sub-interval, we have

$$R\left(T_1^-\right) = \mathcal{R}\left(\frac{\theta}{\theta_w} \cdot (T_1 - T_0)\right).$$

Next, we analyse the second sub-interval. Using c to represent the accumulated aging effect in $[T_0, T_1)$, the reliability at τ $(T_1 \leq \tau < T_2)$ can be written as

$$R(\tau) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot (\tau - T_1 + c)\right).$$

Therefore, at the beginning of this sub-interval,

$$R\left(T_1^+\right) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot c\right).$$

We then compute c by the continuity of the reliability function; that is, the reliability function must satisfy the constraints

$$\forall \ell = 1, 2, \cdots, d - 1 : R\left(T_\ell^-\right) = R\left(T_\ell^+\right).$$

Thus, because of $R(T_1^-) = R(T_1^+)$, we obtain

$$c = \frac{(T_1 - T_0)\theta_p}{\theta_w} \tag{27}$$

; and hence

$$R(\tau) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot \left(\tau - T_1 + \frac{(T_1 - T_0)\theta_p}{\theta_w}\right)\right), \quad T_1 \leq \tau < T_2.$$

This equation implies that, if a component stays in the processing state for

$$\frac{(T_1 - T_0)\theta_p}{\theta_w} \tag{28}$$

, then its age is the same as if it had stayed in the wait state for $(T_1 - T_0)$. After a simple derivation, this equation can be further rewritten as

$$R(\tau) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot (\tau - T_1) + \frac{\theta}{\theta_w} \cdot (T_1 - T_0)\right).$$

So, at time T_2, we have

$$R(T_2) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot (T_2 - T_1) + \frac{\theta}{\theta_w} \cdot (T_1 - T_0)\right).$$

By generalizing the above calculation steps, the lifetime reliability of a component at time t can be written as

$$R(t) = \mathcal{R}\left(\frac{\theta}{\theta_p} \cdot \sum_{i=1}^{\lfloor d/2 \rfloor}(T_{2i} - T_{2i-1}) + \frac{\theta}{\theta_w} \cdot \sum_{i=1}^{\lceil d/2 \rceil}(T_{2i-1} - T_{2i-2})\right).$$

By Theorem 1, $\sum_{i=1}^{\lfloor d/2 \rfloor}(T_{2i} - T_{2i-1})$, and $\sum_{i=1}^{\lceil d/2 \rceil}(T_{2i-1} - T_{2i-2})$ have been approximated as $\psi_p(t, t^b, \mathbf{t}_\ell)$, and $\psi_w(t, t^b, \mathbf{t}_\ell)$ respectively. Additionally, this conclusion is obviously independent of the component's starting state. Therefore, (3) holds. $\qquad\square$

ACKNOWLEDGMENT

The authors wish to thank the associate editor, and the anonymous reviewers for their constructive comments that have helped to improve the article.

REFERENCES

[1] H. Liu, "Reliability of a load-sharing k-out-of- n:g system: Non-iid components with arbitrary distributions," *IEEE Trans. Reliability*, vol. 47, no. 3, pp. 279–184, September 1998.

[2] J. Shao and L. R. Lamberson, "Modeling a shared-load k-out-of- n: G system," *IEEE Trans. Reliability*, vol. 40, no. 2, pp. 205–209, June 1991.

[3] L. Huang and Q. Xu, "On modeling the lifetime reliability of homogeneous manycore systems," in *Proceedings Pacific Rim International Symposium on Dependable Computing*, 2008, pp. 87–94.

[4] F. P. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid redundant digital system: generalized triple modular redundancy with self-repair," in *Proceedings AFIPS Conference, Spring Joint Computer Conference*, 1970, pp. 375–383.

[5] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, 2nd ed. : John Wiley & Sons, 2002.

[6] H.-H. Lin, K.-H. Chen, and R.-T. Wang, "A multivariate exponential shared-load model," *IEEE Trans. Reliability*, vol. 42, no. 1, pp. 165–171, March 1993.

[7] T. F. Hassett, D. L. Dietrich, and F. Szidarovszky, "Time-varying failure rates in the availability and reliability analysis of repairable systems," *IEEE Trans. Reliability*, vol. 44, pp. 155–160, March 1995.

[8] S. V. Amari and R. Bergman, "Reliability analysis of k-out-of- n load-sharing systems," in *Proceedings Annual Reliability and Maintainability Symposium*, 2008, pp. 440–445.

[9] S. V. Amari, K. B. Misra, and H. Pham, "Tampered failure rate load-sharing systems: status and perspectives," in *Handbook of Performability Engineering*, K. B. Misra, Ed. : Springer London, 2008, pp. 291–308.

[10] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling: Principles and Applications*. : John Wiley & Sons, 2003.

[11] M. Xie, Y.-S. Dai, and K.-L. Poh, *Computing Systems Reliability: Models and Analysis*. : Kluwer Academic Publishers, 2004.

[12] D. J. Sherwin and A. Bossche, *The Reliability, Availability and Productiveness of Systems*. : Chapman & Hall, 1993.

[13] I.-R. Chen and F. B. Bastani, "Warm standby in hierarchically structured process-control programs," *IEEE Trans. Software Engineering*, vol. 20, no. 8, p. 1994, August 1994.

[14] R. Subramanian and V. Anantharaman, "Reliability analysis of a complex standby redundant system," *Reliability Engineering and System Safety*, vol. 48, pp. 57–70, 1995.

[15] E. J. Vanderperre, "Reliability analysis of a warm standby system with general distributions," *Microelectron Reliability*, vol. 30, no. 3, pp. 487–490, 1990.

[16] J. She and M. G. Pecht, "Reliability of a k-out-of- n warm-standby system," *IEEE Trans. Reliability*, vol. 41, no. 1, pp. 72–75, March 1992.

[17] L. R. Goel, R. Gupta, and R. K. Agnihotri, "Analysis of a three-unit redundant system with two types of repair and inspection," *Microelectron Reliability*, vol. 29, no. 5, pp. 769–773, 1989.

[18] T. Zhang, M. Xie, and M. Horigome, "Availability and reliability of k-out-of- $(m + n)$:g warm standby systems," *Reliability Engineering and System Safety*, vol. 91, pp. 381–387, 2006.

[19] Intel, "SA-1100 microprocessor technical reference manual," 1998.

[20] M. D. Beaudry, "Performance-related reliability measures for computing systems," *IEEE Trans. Computers*, vol. 27, no. 6, pp. 540–547, June 1978.

[21] Nvidia, "Geforce 8800 graphics processors," [Online]. Available: http://www.nvidia.com\/page/geforce8800.html

[22] Tilera, "Tile64 processor family," [Online]. Available: http://www.tilera.com/products\/processors.php

[23] S. Vangal *et al.*, "An 80-tile 1.28TFLOPS network-on-chip in 65nm CMOS}," in *Proceedings International Solid State Circuits Conference (ISSCC)*, 2007, pp. 98–99.

[24] A. Agarwal and M. Levy, "The KILL rule for multicore," in *Proceedings ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 750–753.

[25] S. Borkar, "Thousand core chips—A technology perspective," in *Proceedings ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 746–749.

[26] S. Borkar, *Designing reliable systems from unreliable components: The challenges of transistor variability and degradation*, vol. 25, no. 6, pp. 10–16, Nov./Dec. 2005.

[27] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *Proceedings IEEE/ACM International Symposium on Computer Architecture (ISCA)*, 2004, pp. 276–287.

[28] Cisco, "Cisco and IBM collaborate to design and build world's most sophisticated, high-performance 40Gbps custom chip," [Online]. Available: http://newsroom.cisco.com/dlls/partners/news/2004/pr_prod_06-09.html

Lin Huang (S'08) received the B.S. degree in electronic engineering from Shanghai Jiaotong University, Shanghai, China, in 2007. She is currently a Ph.D. student in the CUHK Reliable Computer Laboratory (CURE Lab.), the Department of Computer Science and Engineering at the Chinese University of Hong Kong, Hong Kong.

Her research interests include reliability analysis of multicore systems, fault-tolerant computing, and network-on-chip testing.

Qiang Xu (S'02–M'05) received the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2005.

Since 2005, he has been an Assistant Professor with the Department of Computer Science and Engineering, at The Chinese University of Hong Kong, Hong Kong. He leads the CUHK Reliable Computing Laboratory (CURE Lab.). His research interests range from testing and debugging of system-on-a-chip integrated circuits to fault tolerance and reliable computing. He has published more than 50 technical papers in these areas.

Dr. Xu was a recipient of the Best Paper Award in 2004 IEEE/ACM Design, Automation and Test in Europe Conference (DATE). He is a member of the ACM SIGDA, and the IEEE Computer Society. He has served as a technical program committee member for a number of conferences on VLSI design and testing.