New Generation Computing, 29(2011)345-364
Ohmsha, Ltd. and Springer

Modeling and Analysis of Grid Service Reliability Considering Fault Recovery

Suchang GUO, Hong-Zhong HUANG and Yu LIU
University of Electronic Science and Technology of China
Chengdu, P.R.CHINA, 611731
schguo@189.cn
{hzhuang,yuliu}@uestc.edu.cn

Received 27 May 2009
Revised manuscript received 29 March 2010

Abstract The extreme complexity of grid system makes it extremely difficult to achieve high service reliability, and this situation is aggravated by the fact that many grid services need to perform time-consuming tasks that may require several days or even months of computation. To improve grid service reliability, this paper studies a fault recovery technique in grid systems and conducts in-depth research on grid reliability modeling and analysis with fault recovery. Grid failures considered in this paper are classified into two categories: unrecoverable failures and recoverable failures. Software reliability is taken into account as well. To make fault recovery more practical, certain constraints on fault recovery are introduced and grid service reliability models under these practical constraints are developed. Numerical examples are presented, and based on the results obtained, the impact of fault recovery as well as that of practical constraints on grid service reliability is discussed.

Keywords: Grid, Service Reliability, Recoverability, Fault Tolerance, Fault Recovery.

§1 Introduction

Grid computing has emerged as the next-generation parallel and distributed computing methodology. Its goal is to provide a service-oriented infrastructure that leverages standardized protocols and services to enable pervasive access to and coordinated sharing of geographically distributed hardware, software and information resources for solving various kinds of large-scale parallel applications in the wide area network.[9, 10, 12] However, it is a big challenge to make service execution in grid systems in a reliable manner. Research has shown that the grid system, composed of thousands of heterogeneous resources located at disjoined domains, is very prone to failures due to its extreme complexity.[2,3] Moreover, the likelihood of failure occurrence is often increased by the fact that many grid services requested by grid users will perform time-consuming tasks that may require several days or even months of computation.[26] Therefore, it is very crucial to assure the quality and reliability of grid service so as to guarantee the correct outcomes of requested services to grid users.

As one of the important measures of quality of service (QoS), grid service reliability is considered to be one of the most critical and important issues in grid systems.[7] With any application requirement, a corresponding service combined with the desired operations is created.[11] Under the control of the resource management system (RMS), the service is supposed to execute certain task in the form of software programs. Grid service reliability is defined as the probability that all programs involved in the considered service are executed successfully.[7] Recently, grid service reliability has attracted substantial research and attention. Dai et al. presented a virtual approach to modeling grid services and obtained grid service reliability using the graph theory.[7] Levitin and Dai studied grid service reliability for grid systems with star topology.[19] Dai et al. studied grid service reliability and optimal task partition for grid systems with tree topology.[4,5] Levitin et al. studied grid service reliability taking into account the precedence constraints on programs execution.[20] Dai et al. presented a hierarchical model from the mapping of the physical architecture and the logical architecture in grid systems for grid service reliability analysis and evaluation.[6]

The extreme complexity of grid system makes it highly difficult to achieve high service reliability. One way to improve grid service reliability is to adopt fault tolerance in grid system. In Globus, a well-known grid application, the lack of support for fault tolerance is considered to be a noticeable flaw.[10,14] In Condor-G, a checkpoint fault tolerance mechanism is provided in batch queuing systems, but fault tolerance of grid resources is not used.[8,21] Recently, more and more research has been focusing on fault tolerance in grid system. Affaan and Ansari introduced a backup mechanism to achieve fault tolerance in grid system.[1] Jin et al. put forward a fault tolerance mechanism in grid system based on Java threads state capturing and Mobile Agent.[15] Kovacs and Kacsuk introduced the concept of job migration to achieve fault tolerance in grid system.[17] Townend and Xu developed an approach to fault tolerance in grid system based on job replication.[26]

In the grid, it is practically impossible for a grid node to run continuously without any interruptions.[23] During subtask execution, any failure occurring on a node or on a communication link will result in the termination of subtask execution. If a subtask requires a long execution time, the probability of failure occurrence is high, and it could often be the case that after a long time has been spent in executing the subtask, the subtask is terminated by a failure. This leads to a terrible waste of time and resources consumed. As one of the fault tolerance techniques, *fault recovery* can provide an opportunity for failed nodes to continue processing through recovery actions, which could be a good solution to the aforementioned problem.

Fault recovery, which is based on the checkpoint mechanism, is not new and some researches about fault recovery have been done in general distributed computing system (DCS).[24, 27] In grid system, fault recovery studied in most related research is achieved by migration mechanism,[1, 15, 17, 26] i.e., when a failure occurs on a grid node, the state information is migrated to another node on which the subtask execution is resumed from the interrupted point. However, another possible fault recovery mechanism is to resume the subtask on the failed node once the node is recovered, which is referred to as the *local fault recovery mechanism* in our research. The local fault recovery mechanism can save migration time, hence could be system-performance-beneficial compared with the fault recovery mechanism by migration. Heddaya analyzed the impact of local fault recovery on the service reliability of DCS.[13] However, to our knowledge, research on the impact of local fault recovery on the reliability of grid system is very scarce, especially grid service reliability modeling with local fault recovery.

In this paper, we conduct in-depth research on grid service reliability with local fault recovery and a model of grid service reliability for grid systems with star topology is presented. The proposed model is different from that in the Heddaya's research [13] where the service reliability of distributed system is obtained by analogy. In reliability modeling and analysis, we take into account software reliability, which is an important issue in system reliability analysis yet has not been addressed in related research. Moreover, to make fault recovery in grid system more practical, certain constraints on local fault recovery, i.e., constraints on the life times of subtasks and on the numbers of recoveries performed, are considered. We present two numerical examples by which the impact of fault recovery on grid service reliability and the impact of practical constraints on grid service reliability are clearly manifested.

The remainder of this paper is organized as follows. Section 2 reviews related research and discusses existing approaches to grid service reliability modeling and analysis. Section 3 studies local fault recovery mechanism in grid system, and a grid service reliability model considering fault recovery is developed. Section 4 discusses grid service reliability with certain practical constraints. Section 5 presents two numerical examples and gives detailed discussions on the results obtained. Section 6 concludes the paper.

§2 Traditional Reliability Modeling and Analysis of Grid System

2.1 Star Topology of Grid System

The Open Grid Services Architecture (OGSA),[12] which is expected to be adopted widely in industry and research, enables a grid to develop from a computing grid, a data grid, or other dedicated grids to a "service grid." That is, a grid has a widely distributed server on which all applications are packaged as services. The interaction between grid users and the grid system is nothing but service request and response. When a user's request arrives, the grid initiates a particular service to respond, which is to execute a certain task under the control of the RMS. Generally, the RMS divides the task into a set of subtasks so as to improve the efficiency of task execution. Once the RMS determines which set of resources to use, the subtasks are assigned to the corresponding resources held on certain nodes and are executed in parallel. When the nodes finish the assigned subtasks, they return the results to the RMS. The RMS then integrates the received results into an entire task output and presents it to the user.

In grid system, resource discovery and system selection play important roles in the process of resource determination.[23)] Different from traditional distributed computing environments, the RMS does not have complete control over all the resources in grid system. Although all online nodes, or resources, are linked through communication links with one another, only a small portion of nodes, or resources, available for a specific grid service is discovered by the RMS. At the same time, through system selection, the RMS normally selects more than one resource from the discovered resources to which assign a subtask so that the grid service reliability can be improved. Therefore, in the case of one RMS in grid system, the RMS and the selected resources can be regarded as a star topology.[19, 20)]

An example of star topology is given in Fig. 1. In Fig. 1, three subtasks are assigned to six nodes connected with the RMS through respective communication links. Each of the three subtasks is assigned to two nodes for parallel execution, e.g., subtask 1 is assigned to node 1 and node 2.

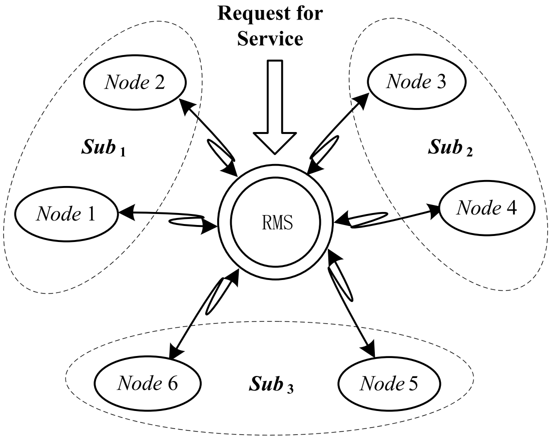


Fig. 1 An Example of Grid System with Star Topology

2.2 Grid Service Reliability Modeling and Analysis

During the execution of subtasks, failures may occur on grid nodes and/or communication links. If a failure occurs when the node is executing a subtask, the output of the subtask will be incorrect, or no output will be send to the RMS at all. Similarly, if a failure occurs on the link between a node and the RMS when it is transferring data, the received information may be unexpected. Current approaches to grid service reliability modeling and analysis mainly consider the above two situations, and the basic assumptions made are as follows:[7,19]

(a) The RMS is perfect during the processing of the grid service, i.e., the RMS never fails, and the time of task processing by the RMS is negligible when compared with subtasks' processing times.

(b) When a service request arrives at the RMS, the RMS responds to it immediately; when a subtask is assigned to a node, the node executes the subtask immediately.

(c) There is no precedence constraint on the order of subtask execution.

(d) Each node can execute only one subtask at any time.

(e) The failure processes of nodes and those of communication links can be modeled by Poisson processes.[7,19,20]

(f) The failures in different elements (nodes or communication links) are independent.[7,19,20]

After the RMS receives a service S, it divides S into m subtasks and assigns them to w nodes. The required processing time of subtask $i (i = 1, 2, \cdots, m)$ on node k is [7,19]

$$\tau_{ik} = c_i/s_k, \tag{1}$$

where c_i is the computational complexity of subtask i and s_k is the processing capability of node $k (k = 1, 2, \cdots, w)$.

According to assumption (e), denoted by λ_k the failure intensity of node k, the probability for node k to be functioning without any failure during the processing time of subtask i is

$$p_{ik} = \exp(-\lambda_k \tau_{ik}), \tag{2}$$

where τ_{ik} is given by (1).

Denote by a_{ik} the amount of data exchanged between the RMS and node k when executing subtask i, and denote by y_k the mean speed of communication link between the RMS and node k. The required communication time between the RMS and node k when executing subtask i is

$$l_{ik} = \exp(-a_k/y_k). \tag{3}$$

Denote by ε_k the failure intensity of communication link between the RMS and node k. According to assumption (e), the probability for this communication link to be functioning without any failure during l_{ik} is

$$q_{ik} = \exp(-\varepsilon_k l_{ik}), \tag{4}$$

where l_{ik} is given by (3).

According to assumption (f), the reliability of subtask i executed on node k, i.e., the probability that subtask i can be successfully completed by node k, is

$$R_{ik} = p_{ik} q_{ik} = \exp(-\lambda_k \tau_{ik} - \varepsilon_k l_{ik}). \tag{5}$$

To improve grid service reliability, a subtask is normally assigned to several nodes for parallel execution. Whenever a node successfully completes a subtask and returns the output to the RMS, the subtask is considered to be completed. Denote by $D(i)$ the node set to which subtask i is assigned. The reliability of subtask i, which is often referred to as *grid program reliability*,[7] is

$$R(Sub_i) = 1 - \prod_{k \in D(i)} (1 - R_{ik}), \tag{6}$$

where R_{ik} is given by (5).

When the RMS receives all the outcomes of subtasks, the grid service is considered to be completed successfully. Therefore, the *grid service reliability* is

$$R(S) = \prod_{i=1}^{m} R(Sub_i) = \prod_{i=1}^{m} \left[1 - \prod_{k \in D(i)} (1 - R_{ik}) \right]. \tag{7}$$

The processing times, τ_{ik}'s, and failure intensities of grid nodes and communication links, λ_k's and ε_k's , can be estimated by grid monitoring systems.[25] Thus, the grid service reliability can be obtained by (7).

§3 Reliability Modeling and Analysis of Grid System with Fault Recovery

In the above grid service reliability modeling, fault recovery is not considered. In reality, fault recovery as well as other fault tolerance techniques may be adopted in grid system to achieve higher service reliability. In this section, we study grid service reliability considering fault recovery.

3.1 Fault Recovery in Grid System

It should be noted that not all failures that occur in grid system can be recovered. According to the recoverability, the failures in grid system can be classified into two categories: unrecoverable failures and recoverable failures. Failures occurring on communication links are unrecoverable failures since generally there aren't any recovering modules on the communication links by which the interrupted data transfer can be resumed with/after recovery actions. Moreover, software failures, which are caused by embedded faults in programs,[22, 28] are unrecoverable failures since no fault removal activities are performed (as is the case in software testing process), and the source codes of the programs are not changed. However, failures occurring on grid nodes can be unrecoverable or recoverable failures. For unrecoverable failures, the subtask is terminated. For recoverable failures, such as those caused by human operation errors or performance overload, once the failed node becomes operational, some particular recovery procedures in grid nodes can resume the interrupted execution of a subtask by recovering as much state information as needed. It should be noted that the recovery mechanism considered in the paper is different from that performed in Levitin and Dai's research[19] where assumes that each failed subtask is repeated from the beginning after the fault recovery. Figure 2 illustrates the classification of failures in grid system according to recoverability.

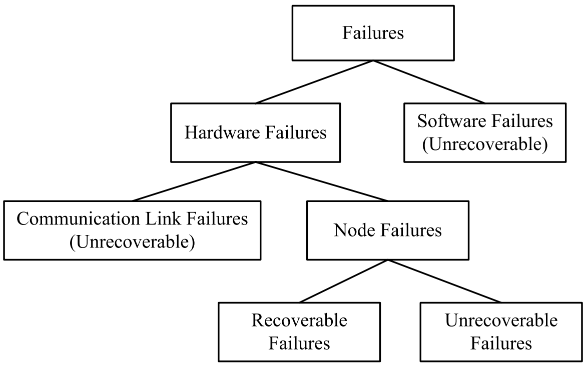


Fig. 2 Classification of Failures in Grid System According to Recoverability

To describe recoverability of hardware failures, a random variable $X_k^{(j)}$ is defined as follows:

$$X_k^{(j)} = \begin{cases} 0 & \text{if the } j\text{:th failure on node } k \text{ is recoverable} \\ 1 & \text{if the } j\text{:th failure on node } k \text{ is unrecoverable} \end{cases} \tag{8}$$

Denoted by x_k the probability that a hardware failure on node k is recoverable, then $\Pr\{X_k^{(j)} = 0\} = x_k$, $\Pr\{X_k^{(j)} = 1\} = 1 - x_k$. Figure 3 gives an example of a subtask execution process with fault recovery. In this example, subtask i is executed on node k. The first failure occurs at t_1 and is recovered at t_2; the second failure occurs at t_3 and is recovered at t_4; and so on. The subtask execution process goes on until the subtask is successfully completed or is terminated by an unrecoverable failure. In the former case, the total execution time of subtask i on node k is τ_{ik}, which is given by (1); in the latter case, the subtask fails. In this example, a total n failures have occurred, all of which are recoverable failures; thus the subtask is completed successfully.

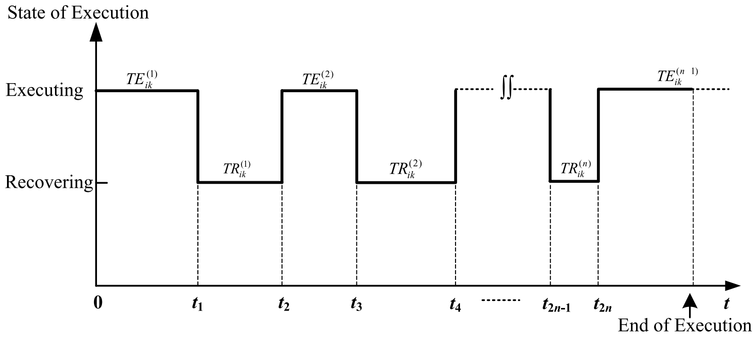


Fig. 3 An Example of the Execution Process of Subtask i on Node k with Fault Recovery

Denote by N_{ik} the number of recoverable failures that occur when subtask i is executed on node k, and N_{ik} is a random variable. If $N_{ik} = 0(n \geq 1)$, then denote by $TE_{ik}^{(j)}(j = 1, 2, \cdots, n, n+1)$ and $TR_{ik}^{(j)}(j = 1, 2, \cdots, n)$ the execution times and recovering times in the execution process of subtask i on node k, respectively. Denote by TE_{ik} and TR_{ik} the total execution time and total recovery time before the occurrence of an unrecoverable failure or the successful completion of the subtask. Then if $N_{ik} = n(n \geq 1)$, we have

$$TE_{ik} = \sum_{j=1}^{n+1} TE_{ik}^{(j)}, TR_{ik} = \sum_{j=1}^{n} TR_{ik}^{(j)}. \tag{9}$$

The *life time* of subtask i executed on node k, T_{ik}, is

$$T_{ik} = TE_{ik} + TR_{ik}. \tag{10}$$

If subtask i is successfully completed on node k, then its life time is

$$T_{ik} = \tau_{ik} + TR_{ik}. \tag{11}$$

3.2 Grid Service Reliability Considering Fault Recovery

Grid service reliability is determined by the hardware reliability of grid nodes, the software reliability of grid nodes, and the reliability of communication links. Due to the complexity of hardware failures in recoverability, We first model the hardware reliability of grid nodes. Denote by λ_k^h the hardware failure intensity of node k. According to assumptions (e) and (f), $TE_{ik}^{(j)}$'s are independent and identically distributed (i.i.d.) random variables, each following exponential distribution with parameter λ_k^h. For the recovering process, it is reasonable to further make the following assumptions:

(g) $TR_{ik}^{(j)}$'s $(j = 1, 2, \cdots, n)$ are i.i.d. random variables, each following exponential distribution with parameter μ_k (μ_k is often referred to as recovery rate in the literature);

(h) $TR_{ik}^{(j)}$'s are independent with $TE_{ik}^{(j)}$'s.

The reliability of node k executing subtask i, considering only hardware failures, is

$$p_{ik}^h = P_1 + P_2. \tag{12}$$

In the above, $P_1 = Pr\{E_1\}$, where E_1 is the event that during the execution of subtask i on node k, no hardware failure occurs; $P_2 = Pr\{E_2\}$, where E_2 is the event that during the execution of subtask i on node k there is at lease one hardware failure; and all the failures are recoverable failures. P_1 can be easily obtained as

$$P_1 = \exp(-\lambda_k^h \tau_{ik}), \tag{13}$$

where τ_{ik} is given by (1). P_2 can be obtained as

$$P_2 = \sum_{n=1}^{\infty} \Pr\{E^{(n)}\}, \tag{14}$$

where $E^{(n)}(n \geq 1)$ is the event that during the execution of subtask i on node k only n recoverable failures occur (thus subtask i is successfully completed).

Define $W_{ik}^{(n)} = TE_{ik}^{(1)} + TE_{ik}^{(2)} + \cdots + TE_{ik}^{(n)}$, the probability being summed in (14) can be calculated by

$$\Pr\{E^{(n)}\} = \Pr\{\tau_{ik} - TE_{ik}^{(n+1)} < W_{ik}^{(n)} < \tau_{ik}, \sum_{j=1}^{n} X_k^{(j)} = 0\}. (15)$$

It is known that $W_{ik}^{(n)}$ is an Erlang random variable with parameters (n, λ_k^h).[16] Moreover, according to assumption (f), we have that $W_{ik}^{(n)}$ is independent of $TE_{ik}^{(n+1)}$. Therefore, the joint density of $(W_{ik}^{(n)}, TE_{ik}^{(n+1)})$ is

$$f_{ik}^{(n)}(x,y) = \begin{cases} \dfrac{(\lambda_k^h)^{n+1} x^{n-1} \exp[-\lambda_k^h (x+y)]}{(n-1)!}, & x, y \geq 0 \\ 0, & \text{else} \end{cases} \quad (16)$$

From (16), (15) can be calculated by

$$\Pr\{E^{(n)}\} = \frac{(x_k \lambda_k^h \tau_{ik})^n}{n!} \exp(-\lambda_k^h \tau_{ik}). \quad (17)$$

It can be noted that the result obtained in (13) is the special case of (17) for which $n = 0$. Therefore, substituting (17) into (14), and then substituting the result obtained into (12), the reliability of node k executing subtask i considering only hardware failures, is obtained as

$$p_{ik}^h = exp[-(1 - x_k)\lambda_k^h \tau_{ik}]. \quad (18)$$

Besides hardware failures, software failures may also occur on grid nodes, which are unrecoverable failures. Denote by λ_i^s the software failure intensity of the program performing subtask i; as programs of the same version are executed on all the nodes in $D(i)$, λ_i^s is a constant for all these nodes.[22, 29] The reliability of node k executing subtask i, considering only software failures, is

$$p_{ik}^s = \exp(-\lambda_i^s \tau_{ik}). \quad (19)$$

Besides hardware and software failures on nodes, failures may also occur on communication links. The reliability of communication links can be calculated by (4).

Based on the above analysis, the reliability of subtask i executed on node k, i.e., the probability that subtask i can be successfully completed by node k, is

$$R_{ik} = p_{ik}^h p_{ik}^s q_{ik} = \exp[-(1 - x_k)\lambda_k^h \tau_{ik} - \lambda_i^s \tau_{ik} - \varepsilon_k l_{ik}]. \quad (20)$$

Substituting (20) into (6) and then substituting the result obtained into (7), the grid service reliability considering fault recovery is obtained as

$$R(S) = \prod_{i=1}^{m} \left[1 - \prod_{k \in D(i)} (1 - p_{ik}^h p_{ik}^s q_{ik}) \right]. \quad (21)$$

From (21), we can have some insight into the relationship between grid service reliability and fault recovery. The current approach to grid service reliability modeling does not consider fault recovery, i.e., $x_k \equiv 0$. In this case, the grid service reliability obtained from (21) is the same as that obtained from (7), except that in (7) both hardware reliability and software reliability are considered. In the other extreme case, where grid node k has a perfectly recoverable system, i.e., $x_k \equiv 0$, from (21) it can be seen that only software failures and communication link failures influence grid service reliability, while hardware failures of grid nodes have no effect on grid service reliability. This is expected since all hardware failures of grid nodes are recoverable failures. Therefore, (21) is an extension of the current grid service reliability model (7) and is a more generic model which caters to software failures as well as local fault recovery on grid nodes.

For a grid service reliability model (21), some practical considerations have to be taken and incorporated into the model. For instance, the model allows a grid node to recover any large number of failures; however, this may not be reasonable in practice. In the following section, we will study grid service reliability under some practical considerations.

§4 Grid Service Reliability Modeling Under Practical Considerations

Although a fault recovery mechanism provides an efficient way to improve the reliability of grid system, some disadvantages may also be brought forward. With the introduction of fault recovery, the life time of subtasks in grid nodes is extended, especially when the mean recovery time is rather long on some nodes. In grid, the service time is very critical to users, since it influences the amount of money that users have to pay when grid goes commercial. On the other hand, the resource providers in grid may not be willing to spend a long time in performing one subtask. Moreover, fault recovery requires a large amount of state information so as to enable the node to execute the subtask continuously as if there were no failures occurring. In some particular situations, failures may occur frequently and then be recovered again and again, which imposes a great burden on grid nodes and has a strong influence on the availability of the nodes. Therefore, it is advisable to take some measures to limit the life time of any subtask as well as the number of recoveries performed.

4.1 Constraints on the Life Times of Subtasks

To prevent the life time of a subtask from exceeding an allowed time limit, a deadline can be set for the subtask execution. Once the life time of subtask i executed on node k, T_{ik}, exceeds this deadline, denoted by T^*_{ik}, the node will claim failure of the subtask to the RMS.

The life time T_{ik}, if the subtask is successfully completed, is given by (11). Since the required execution time τ_{ik} is a constant, T_{ik} mainly depends on the total recovery time, TR_{ik}. From assumption (g), if $N_{ik} = n (n \geq 1)$, then TR_{ik} follows the Erlang distribution with parameters (n, μ_k), whose cumulative distribution function (c.d.f.) is given by

$$F_{TR_{ik}} \equiv Pr\{TR_{ik} \leq t\} = exp(-\mu_k t) \sum_{j=n}^{+\infty} \frac{(\mu_k t)^j}{j!}; t \geq 0. \quad (22)$$

Under the constraint of a time deadline, T_{ik}^*, the reliability of node k executing subtask i considering only hardware failures, is

$$p_{ik}^{(1)} = P_1 + P_3, \quad (23)$$

where P_1 is the probability of no hardware failure occurring in the subtask's life time, which can be calculated by (13). P_3 is the probability that there is at least one hardware failure, all of the failures are recoverable failures, and the subtask's life time does not exceed the allowed deadline, i.e.,

$$P_3 \equiv \sum_{n=1}^{\infty} \Pr\{E^{(n)}, T_{ik} \leq T_{ik}^*\}. \quad (24)$$

The probability being summed in (24) can be calculated by

$$\Pr\{E^{(n)}, T_{ik} \leq T_{ik}^*\} = \Pr\{TR_{ik} \leq T_{ik}^* - \tau_{ik} | E^{(n)}\} \Pr\{E^{(n)}\}. \quad (25)$$

From (22), we have

$$\begin{aligned} &\Pr\{TR_{ik} \leq T_{ik}^* - \tau_{ik} | E^{(n)}\} \\ &= 1 - \exp[-\mu_k(T_{ik}^* - \tau_{ik})] \sum_{j=0}^{n-1} \frac{[\mu_k(T_{ik}^* - \tau_{ik})]^j}{j!}. \end{aligned} \quad (26)$$

Substituting (26) and (15) into (25), we get

$$\begin{aligned} \Pr\{E^{(n)}, T_{ik} \leq T_{ik}^*\} &= \frac{(x_k \lambda_k^h \tau_{ik})^n}{n!} \exp(-\lambda_k^h \tau_{ik}) * \\ &\left\{ 1 - \exp[-\mu_k(T_{ik}^* - \tau_{ik})] \sum_{j=0}^{n-1} \frac{[\mu_k(T_{ik}^* - \tau_{ik})]^j}{j!} \right\}. \end{aligned} \quad (27)$$

Substituting (27) into (24), and then submitting the result obtained into (23), we get

$$p_{ik}^{(1)} = p_{ik}^h - \exp(-\lambda_k^h \tau_{ik}) \sum_{n=1}^{\infty} \frac{(x_k \lambda_k^h \tau_{ik})^n \Gamma[n, \mu_k(T_{ik}^* - \tau_{ik})]}{\Gamma[n]\Gamma[n+1]}, \quad (28)$$

where the Gamma function $\Gamma[n, y] \equiv \int_y^{+\infty} \exp(-x) x^{n-1} dx$ and $\Gamma[n] \equiv (n-1)!$ for any $n \geq 1$, and p_{ik}^h is given by (18).

Taking into consideration the software reliability and the reliability of communication links, the reliability of subtask i executed on node k, with a deadline T_{ik}^*, is

$$R_{ik}^{(1)} = p_{ik}^s q_{ik}(p_{ik}^h - C_1), \quad (29)$$

where p_{ik}^{s} is given by (19), q_{ik} is given by (4), and p_{ik}^{h} is given by (18). For any given value of $T_{ik}^{*} > \tau_{ik}$, C_1 is a positive constant, which is

$$C_1 \equiv \exp(-\lambda_k^h \tau_{ki}) \sum_{n=1}^{\infty} \frac{(x_k \lambda_k^h \tau_{ki})^n \Gamma[n, \mu_k T_{ki}^{*} - \mu_k \tau_{ki}]}{\Gamma[n]\Gamma[n+1]}. \tag{30}$$

It is interesting to compare the result obtained, (29), with the result obtained without time deadline constraint, (20). It can be easily seen that the value of $R_{ik}^{(1)}$, given by (29), is smaller than that of R_{ik}, given by (20), which is expected. Moreover, if $T_{ik}^{*} \to \infty$, i.e., there is no constraint on the subtask's life time, then $C_1 \to 0$, and thus $R_{ik}^{(1)} \to R_{ik}$.

Substituting (29) into (6) and then substituting the result obtained into (7), the grid service reliability with constraints on the life times of subtasks is

$$R^{(1)}(S) = \prod_{i=1}^{m} \left[1 - \prod_{k \in D(i)} \left(1 - R_{ik}^{(1)} \right) \right], \tag{31}$$

where $R_{ik}^{(1)}$ is given by (29).

4.2 Constraints on the Numbers of Recoveries Performed

Denote by $L_k (L_k \geq 1)$ the allowed number of recoveries that may be performed during the execution of a subtask on node k. When the $(L_k + 1)^{st}$ recoverable failure occurs before the completion of the subtask, the node will claim failure of the subtask to the RMS. Under the constraint of L_k, the reliability of node k executing subtask i, considering only hardware failures, is

$$p_{ik}^{(2)} = P_1 + P_4, \tag{32}$$

where P_1 is given by (13) and P_4 is

$$P_4 \equiv \sum_{n=1}^{L_k} \Pr\{E^{(n)}\}. \tag{33}$$

Substituting (17) into (33), and then substituting the result obtained into (32), we get

$$p_{ik}^{(2)} = \sum_{n=0}^{L_k} \frac{(x_k \lambda_k^h \tau_{ik})^n}{n!} \exp(-\lambda_k^h \tau_{ik}). \tag{34}$$

According to the incomplete Gamma function, (34) can be rewritten as

$$p_{ik}^{(2)} = \frac{p_{ik}^{h} \Gamma[1 + L_k, x_k \lambda_k^h \tau_{ik}]}{\Gamma[1 + L_k]}, \tag{35}$$

where p_{ik}^{h} is given by (18).

Taking into consideration the software reliability and the reliability of communication links, the reliability of subtask i executed on node k, with constraint on the number of recoveries performed, is

$$R_{ik}^{(2)} = p_{ik}^h p_{ik}^s q_{ik} \frac{\Gamma[1+L_k, x_k \lambda_k^h \tau_{ik}]}{\Gamma[1+L_k]}. \tag{36}$$

where p_{ik}^s is given by (19) and q_{ik} is given by (4).

Since $x_k \lambda_k^h \tau_{ik}$ is a positive constant, using the property of incomplete Gamma function, it is obtained that

$$\Gamma[1+L_k, x_k \lambda_k^h \tau_{ik}] < \Gamma[1+L_k]. \tag{37}$$

Therefore, from (37) and (20) it can be seen that $R_{ik}^{(2)} < R_{ik}$ for any value of $L_k \geq 1$, which is expected. Furthermore, if $L_k \to \infty$, i.e., there is no constraint on the number of recoveries performed, and then $R_{ik}^{(2)} \to R_{ik}$.

L_k is an important and useful parameter of grid node k. From (22), we can obtain that the mean recovery time is $E(TR_{ik}) = n/\mu_k$, which increases when the number of recoveries performed, n, increases. To prevent the mean recover time from being unreasonably long, a constraint can be placed on the number of recoveries performed, L_k. At the same time, by placing different constraints of L_k on different nodes, we can dynamically manage the grid according to the situations of the system.

Substituting (36) into (6) and then substituting the result obtained into (7), the grid service reliability with constraints on the numbers of recoveries performed is

$$R^{(2)}(S) = \prod_{i=1}^{m} \left[1 - \prod_{k \in D(i)} \left(1 - R_{ik}^{(2)} \right) \right], \tag{38}$$

where $R_{ik}^{(2)}$ is given by (36).

4.3 Constraints on Both the Life Times of Subtasks and the Numbers of Recoveries Performed

It is sometimes reasonable to have constraints on both the life times of subtasks and the numbers of recoveries performed in grid. In this case, the reliability of subtask i executed on node k is

$$R_{ik}^{(3)} = p_{ik}^s q_{ik} \left(p_{ik}^{(2)} - C_2 \right), \tag{39}$$

where p_{ik}^s, q_{ik}, and $p_{ik}^{(2)}$ are given by (19), (4), and (35), respectively. The positive constant C_2 is

$$C_2 \equiv \exp(-\lambda_k^h \tau_{ik}) \sum_{n=1}^{L_k} \frac{(x_k \lambda_k^h \tau_{ik})^n \Gamma[n, \mu_k(T_{ik}^* - \tau_{ki})]}{\Gamma[n]\Gamma[n+1]}. \tag{40}$$

Substituting (39) into (6) and then substituting the result obtained into (7), the grid service reliability with the constraints on both the life times of subtasks and the numbers of recoveries performed is

$$R^{(3)}(S) = \prod_{i=1}^{m} \left[1 - \prod_{k \in D(i)} \left(1 - R_{ik}^{(3)} \right) \right]. \tag{41}$$

§5 Numerical Examples

In this section, we give two numerical examples to illustrate the modeling and analysis procedures of grid service reliability. The first example is used to exemplify the importance and usefulness of fault recovery for grid in terms of reliability improvement. In the second example, we will discuss the impact of practical constraints on grid service reliability.

Example 5.1

Consider a grid service that uses five grid nodes. The service is divided into two subtasks, i.e., $m = 2$. Subtask 1 is assigned to nodes 1 and 2; while subtask 2 is assigned to nodes 3, 4, and 5. The parameters of the grid nodes and communication links are given in Table 1.

Table 1 Parameters of Grid Nodes and Communication Links in Example 5.1

	Node1	Node2	Node3	Node4	Node5
λ_k^h(/s)	0.0010	0.0004	0.0010	0.0006	0.0014
λ_k^s(/s)	0.0001	0.0001	0.0002	0.0002	0.0002
ε_{ik}(/s)	0.0010	0.0010	0.0030	0.0040	0.0010
μ_k(/s)	0.2000	0.3000	0.5000	0.7000	0.2000

Without loss of generality, in this example we assume that both subtasks have the same total execution time, i.e., $\tau_{ik} = \tau$ for $1 \le i \le 2, 1 \le k \le 5$. Similarly, we assume that the communication times with the RMS are the same, i.e., $l_{ik} = 0.02\tau$ for $1 \le i \le 2, 1 \le k \le 5$. Furthermore, we assume that the five nodes have the same recoverability, i.e., $x_k = x$ for $1 \le k \le 5$. The grid service reliability without any constraint with respect to subtask execution time is shown in Fig. 4.

In Fig. 4, it can be seen that the grid service reliability decreases with the increase of subtask's execution time. If the subtask execution time is quite long, which implies that the subtask has high computational complexity, then if no fault recovery mechanism is adopted, the grid service reliability will be rather low. For instance, when $\tau = 2000s$ and $x = 0$, $R(S) = 0.0832$. However, if the recoverability of grid nodes increases, then the grid service reliability can be improved significantly, as shown in Fig. 4. If all the nodes have a perfect recovery system, i.e., $x = 1$, then for $\tau = 2000s$, the service reliability rises to 0.8892. Therefore, fault recovery is very important in grid system and can be of great benefit to the improvement of grid service reliability.

It can be noted that even if all the nodes have a perfect recovery system, i.e., $x = 1$, the grid service reliability is still not 1. This is because in this case, hardware failures on nodes have no impact on grid service reliability; however, software failures and failures of communication links, which are unrecoverable failures, affect grid service reliability.

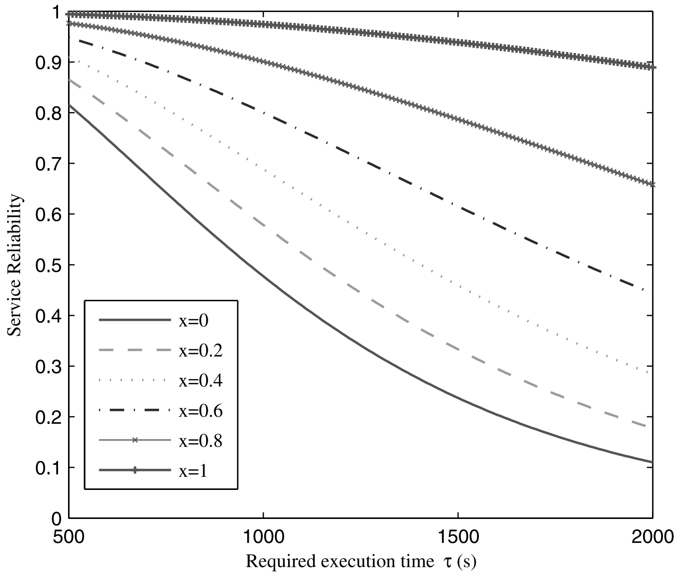


Fig. 4 Reliability Analysis with Respect to Subtask's Execution Time in Example 5.1, without Constraints

Example 5.2

For the second example, the structure of the grid and the parameters are assumed to be the same as in Example 5.1. The processing capability of each node and the mean communication speed of each link are shown in Table 2. Moreover, assume that both subtasks have the same computational complexity, $c_1 = c_2 = c = 10000$ (Mega Operations), and the amount of data exchanged between the RMS and each node is the same, $a_{ik} = a = 0.002c$ for $1 \le i \le 2, 1 \le k \le 5$.

Table 2 Parameters of Grid Nodes and Communication Links in Example 5.2

	Node1	Node2	Node3	Node4	Node5
s_k(MO/s)	10	20	18	15	20
y_k(MB/s)	2	3	5	6	3

Now we study the grid service reliability when practical constraints are imposed. Firstly, the constraints on the numbers of recoveries performed are imposed on grid nodes. Assume that all the five nodes have the same constraint on the number of recoveries performed, $L_k = L$ for $1 \le k \le 5$. The grid service reliability curves without any practical constraint, i.e., $L = +\infty$, and the curves under $L = 1$, with respect to node recoverability x, are shown in Fig. 5.

In Fig. 5, it can be seen that with the increase of node recoverability, the grid service reliability increases. However, the increase with recovery number constraints is slower than that without any constraint. Especially, at the point at $x = 1$, $R(S) = 0.9920$ when $L = +\infty$ while $R(S) = 0.9579$ when $L = 1$. We can also obtain the grid service reliability for $L = 2$ and $x = 1$, which is $R(S) = 0.9861$. Therefore, grid service reliability decreases with the decrease of allowed number of recoveries performed, L_k, which is expected. Furthermore, compared with Fig. 5, the constraint on the number of recoveries performed has a greater influence on the reliability of nodes 1, 3, and 5, whose failure intensities are larger than those of nodes 2 and 4. Therefore, when users decide to impose some practical constraints on local nodes, it seems more meaningful to place the constraint on the number of recoveries performed on nodes whose failure intensities are large, which can accomplish the users' purpose to decease the life time of grid service in local nodes.

Finally, we study the grid service reliability when constraints are placed on both the life times of subtasks and the number of recoveries performed. The grid service reliability with respect to node recoverability, with $T_{ik}^* = T = 10s$ and $L_k = L = 1$ for $1 \leq i \leq 2, 1 \leq k \leq 5$, is shown in Fig. 6. Compared with the results shown in Fig. 5, the grid service reliability under the same constraint of L is lower. For example, when $x = 1$, $R(S) = 0.9363$, which is lower than the reliability obtained previously, $R(S) = 0.9579$. We can also obtain the grid service reliability when $x = 1$, with constraints $T = 5s$ and $L = 1$, which is $R(S) = 0.8860$. The reliability is lower than that under $T = 10s$ and $L = 1$. Therefore, the grid service reliability decreases with the decrease of the deadline set, T_{ik}^*, which is expected. Furthermore, compared with Fig. 5, the constraint on the life times of subtasks has a greater influence on the reliability of nodes 1 and 5, whose recovery rates are smaller than those of nodes 2, 3, and 4. Therefore,

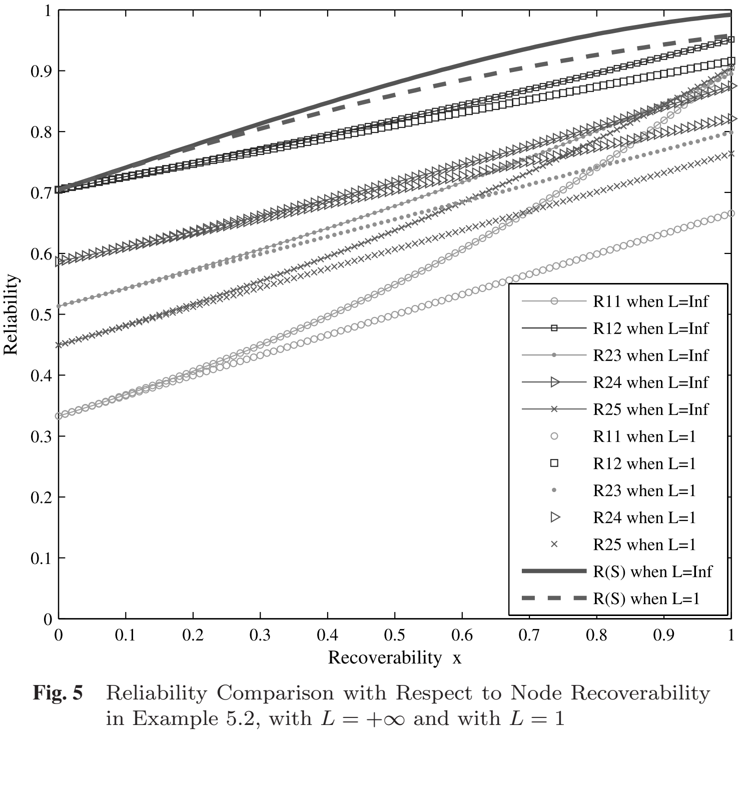


Fig. 5 Reliability Comparison with Respect to Node Recoverability in Example 5.2, with $L = +\infty$ and with $L = 1$

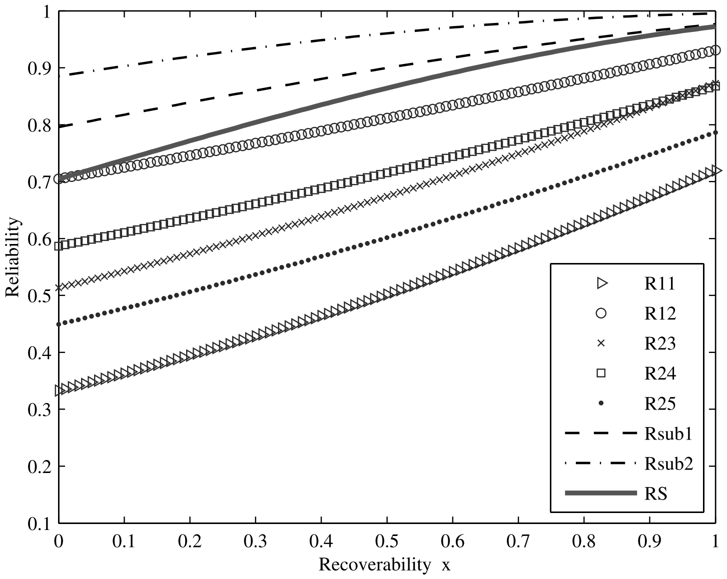


Fig. 6 Reliability analysis with respect to node recoverability in Example 5.2, with L=1 and T=10s

it seems more meaningful to place the constraint on the life times of subtasks executed on nodes which have low recovery rates.

§6 Conclusions

This paper analyzes fault recovery mechanism in grid system and presents the modeling of grid service reliability considering fault recovery. In order to take practical considerations into account, two efficient methods, i.e., to set a deadline for the life time of each subtask and to place a constraint on the number of recoveries performed by each node, are introduced. Under these constraints, grid service reliability is modeled and analyzed. Although the modeling and analysis of grid service reliability in this paper are based on some simplified assumptions, this paper addresses the important issue of adopting fault recovery mechanism in grid system, and the models developed could be of practical use. As for the implementation of fault recovery in grid resources, it can be achieved by embedding fault recovery module in grid clients located at grid nodes. In the module, there are some options, such as the allowed life times of grid subtasks and the allowed numbers of recoveries performed. By those options, resources providers can be free to choose appropriate fault recovery strategies according to the local situations.

Yet more in-depth research on grid service reliability modeling and analysis is needed. For example, in realistic grid system, some precedence constraints on the order of subtask execution may be imposed and the usage amount of grid resources may be dynamic during the execution of grid subtask. Moreover, software fault tolerance techniques could be considered, which may help to further improve grid service reliability. The performance of grid service, which is of great concern to management, could also be addressed besides the optimization of grid service reliability. These are the issues that we shall address in our future research.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China under the contract number 50775026 and the Provincial Key Technologies R&D Program of Sichuan under the contract number 07GG012-002. The authors would like to acknowledge valuable suggestions and comments by Dr. Yuanshun Dai in the Dept of Electrical Engineering & Computer Science, University of Tennessee, Knoxville, USA.

References

1) Affaan, M. and Ansari, M. A., "Distributed Fault Management for Computational Grids," in *Proc. of the Fifth International Conference on Grid and Cooperative Computing 2006*, IEEE Computer Society Press, pp. 363–368, 2006.

2) Bolosky, W. J., Douceur, J. R., Ely, D. and Theimer, M., "Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs," in *Proc. of the ACM International Conference on Measurement and Modeling of Computer Systems 2000*, ACM Press, pp. 34–43, 2000.

3) Bosilca, G., Bouteiller, A., Cappello, F., Djilali, S., Fedak, G., Germain, C., Herault, T., Lemarinier, P., Lodygensky, O., Magniette, F., Neri, V. and Selikhov, A., "MPICH-V: Toward a Scalable Fault Tolerant MPI for Volatile Nodes," in *Proc. of the ACM/IEEE conference on Supercomputing 2002*, IEEE Computer Society Press, pp. 1–18, 2002.

4) Dai, Y. S. and Levitin, G., "Reliability and Performance of Tree-structured Grid Service," *IEEE Transactions on Reliability, 55, 2*, pp. 337–349, 2006.

5) Dai, Y. S., Levitin, G. and Wang, X. L., "Optimal Task Partition and Distribution in Grid Service System with Common Cause Failures," *Future Generation Computer Systems, 23, 2*, pp. 209–218, 2007.

6) Dai, Y. S., Pan, Y. and Zou, X. K., "A Hierarchical Modeling and Analysis for Grid Service Reliability," *IEEE Transactions on Computers, 56, 5*, pp. 681–691, 2007.

7) Dai, Y. S., Xie, M. and Poh, K. L., "Reliability of Grid Service Systems," *Computers and Industrial Engineering, 50, 1*, pp. 130–147, 2006.

8) Epema, D. H. J., Livnyb, M., Dantzigc, R. Va., Eversa, X. and Pruyneb J., "A Worldwide Flock of Condors: Load Sharing among Workstation Cluster," *Future Generations Computer Systems, 12, 1*, pp. 53–65, 1996.

9) Foster, I., "The Grid: a New Infrastructure for 21st Century Science," *Physics Today, 55, 2*, pp. 42–47, 2002.

10) Foster, I. and Kesselman, C., *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, 2003.

11) Foster, I., Kesselman, C. and Nick, J. M., "Grid Services for Distributed System Integration," *Computer, 35, 6*, pp. 37–46, 2002.

12) Foster, I., Kesselman, C. and Tuecke, S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of High Performance Computing Applications, 15, 3*, pp. 200–222, 2001.

13) Heddaya, A. and Helal, A., "Reliability, Availability, Dependability and Performability: a User-centered View," Technical Report 1997-011, 1997.

14) Hwang, S. and Kesselman, C., "A Flexible Framework for Fault Tolerance in the Grid," *Journal of Grid Computing, 1, 3*, pp. 251–272, 2003.

15) Jin, L., Tong, W. Q., Tang, J. Q. and Wang, B., "A Fault-tolerance Mechanism in grid," in *Proc. of IEEE International Conference on Industrial Informatics 2003*, IEEE Computer Society Press, pp. 351–357, 2003.

16) Kao, E. P. C., *An Introduction to Stochastic Processes*, Wadsworth Publishing Company, 1997.

17) Kovacs, J. and Kacsuk, P., "A Migration Framework for Executing Parallel Programs in the Grid," in *European across Grids Conference 2004*, Springer, pp. 80–89, 2004.

18) Levitin, G. and Dai, Y. S., "Performance and Reliability of Star Topology Grid Service with Data Dependency and Two Types of Failures," *IIE Transactions, 39, 8*, pp. 783–794, 2007.

19) Levitin, G. and Dai, Y. S., "Service Reliability and Performance in Grid System with Star Topology," *Reliability Engineering and System Safety, 92, 1*, pp. 40–46, 2007.

20) Levitin, G., Dai, Y. S. and Hanoch, B. H., "Reliability and Performance of Star Topology Grid Service with Precedence Constraints on Subtask Execution," *IEEE Transactions on Reliability, 55, 2*, pp. 507–515, 2006.

21) Litzkow, M., Tannenbaum, T., Basney, J. and Livny, J., "Checkpoint and Migration of UNIX Processes in the Condor Distributed Processing System," *Technical Report UW-CS-TR-1346*, 1997.

22) Musa, J. D., Iannino, A. and Okumoto, K., *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, 1987.

23) Nabrzyski, J., Schopf, J. M. and Weglarz, J., *Grid Resource Management*, Kluwer Publishing Company, 2003.

24) Pradhan, D. K. and Vaidya, N. H., "Roll-forward Checkpointing Scheme: a Novel Fault-tolerant Architecture," *IEEE Transactions on Computers, 43, 10*, pp. 1163–1174, 1994.

25) Tierney, B., Aydt, R., Gunter, D., Smith, W., Taylor, V., Wolski, R. and Swany, M., "White Paper: A Grid Monitoring Service Architecture," Grid Performance Working Group, 2001.

26) Townend, P. and Xu, J., "Fault Tolerance within a Grid Environment," in *Proc. of the UK e-Science All Hands Meeting 2003*, Nottingham Conference Center, pp. 272–275, 2003.

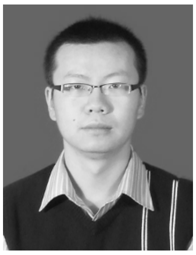
27) Treaster, M., "A Survey of Fault-tolerance and Fault-recovery Techniques in Parallel Systems," *ACM Computing Research Repository (CoRR)*, pp. 1–11, 2005

28) Xie, M., *Software Reliability Modeling*, World Scientific Publishing Company, 1991.

29) Yang, B. and Xie, M., "A Study of Operational and Testing Reliability in Software Reliability Analysis," *Reliability Engineering and System Safety, 70, 3*, pp. 323–329, 2000.

Suchang Guo, Ph.D.: He is currently an Engineer in No. 30 Institute of China Electronics Technology Group Corporation, Chengdu, Sichuan, 610041, China. He received a Ph.D. degree in Mechatronics Engineering from University of Electronic Science and Technology of China, China. His research areas include software reliability, network reliability, and trusted networks.

Hong-Zhong Huang, Ph.D.: He is a full professor, and the Dean of the School of Mechanical, Electronic, and Industrial Engineering at the University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China. He has held visiting appointments at several universities in Canada, USA, and elsewhere in Asia. He received a Ph.D. degree in Reliability Engineering from Shanghai Jiaotong University, China. He has published over 150 journal articles, and 5 books in the fields of reliability engineering, optimization design, fuzzy sets theory, and product development. His current research interests include MDO, system reliability analysis, warranty, maintenance planning and optimization, and computational intelligence in product design.



Yu Liu, Ph.D.: He is currently an Associate Professor in the School of Mechanical, Electronic, and Industrial Engineering, at the University of Electronic Science and Technology of China. He received his Ph.D. degree in mechatronics engineering from the University of Electronic Science and Technology of China. He was a visiting pre-doctoral fellow in the Department of Mechanical Engineering at Northwestern University, Evanston, U.S.A. from 2008 to 2010. His research interests include design under uncertainty, reliability of multi-state systems, maintenance decisions, and optimization.