

# Workload-aware Reliability Evaluation Model in Grid Computing

Peng Xiao

School of Computer and Communication, Hunnan Institute of Engineering, Xiangtan, China

Email: xpeng4623@yahoo.com.cn

Zhigang Hu

School of Information Science and Technology, Central South University, Changsha, China

Email: zghu@mail.csu.edu.cn

**Abstract**—To address the issue of reliability evaluation in dynamic grid environment, a workload-aware reliability evaluation model is proposed, in which queuing system is applied to describe the dynamic workload and working of grid resources. Meanwhile, a new type of resource fault, namely Deadline-Miss fault, is introduced to evaluate the reliability of real-time jobs. The validity of the proposed model and its approach to calculate real-time job's reliability are presented theoretically. Extensive experiments are conducted to verify its performance, and the results show that the proposed model can significantly improve the accuracy of reliability evaluation in presence of dynamic workload. Also, scheduling algorithm based on this model can reduce the jobs' mean response time and the deadline-miss rate.

**Index Terms**—grid computing, reliability evaluation, workload-aware, deadline-miss, queuing system

## I. INTRODUCTION

Grid computing [1] is an emerging technology to address issues such as large-scale resource sharing, wide-area communication, and multi-institutional collaboration. Due to its open architecture, grid resources from different virtual organizations are inherently heterogeneous and dynamic, which lead to resource failures occur frequently and hard to be detected [2, 3]. Furthermore, wide-area resource sharing over uncertain network increases the probability of resource failures since data transferring are often intensive in most of grid applications [4]. Therefore, reliability analysis and evaluation of grid systems becomes a challenging issue and attracts more and more attentions of researchers. Existing studies show that the difficulties of reliability evaluation in grid environments are as following:

- Heterogeneity of resources results in various types of resource fault, which can hardly be taken into consideration in a single model [3, 19, 20].
- Failure model is greatly influenced by resource's workload, which often fluctuates dramatically and unpredictable in runtime [2-3].
- Due to the large-scale and the complexity of grid systems, reliability are difficult to model, analyze, and evaluate [2-5].

Among the existing grid reliability evaluation models, the *Tree-structured Grid Reliability Model* [2] (TGRM) proposed by Y.S. Dai *et al.* is considered as the most effective and efficient. However, the shortcoming of TGRM is that it does not take into account the dynamic workload when calculating reliability. Motivated by this observation, in this paper we propose a workload-aware reliability evaluation model which is based on TGRM.

The rest of this paper is organized as follows: Section 2 presents the related work. In Section 3, we analyze the shortcomings of TGRM, and present the workload-aware TGRM. In Section 4, extensive simulations are conducted to verify the performance of the proposed model. Finally, Section 5 concludes the paper with a brief discussion of future work.

## II. RELATED WORK

In [6], *Distributed Program Reliability* (DPR) and *Distributed System Reliability* (DSR) are two metrics firstly introduced to the reliability evaluation of distributed systems by Raghavendra. Also, Raghavendra propose an algorithm based on graph-traversal to calculate DSR and DPR. However, the complexity of the algorithm is exponential, which means it is unsuitable to evaluation the reliability of large-scale systems. So, Chen *et al.* [7] improves Raghavendra's algorithm by using the graph-cutting approach. The common shortcoming of both two algorithms is that they all assume that the failure rates of resources and links are constant, which is unsuitable to the geographically distributed and dynamic grid environments.

So, in [5] Y.S. Dai *et al.* addresses the issue of reliability evaluation in grid environments. Dai's approach is based on *Minimal-Task-Spanning-Tree* (MTST) just like Raghavendra's, but he uses random failure model instead of assuming the failure rate being constant. In [2], Y.S. Dai *et al.* further their study and propose a *Tree-structured Grid Reliability Model* (TGRM) for analyzing the performance and the reliability of grid services in the presence of common failures caused by sharing communication links. TGRM is the first grid reliability model and has been proven to be effective and efficient.

However, TGRM dose not take into account the effects of workload on the reliability evaluation. In [3], Czajkowski has pointed out that “The reliability of grid system is greatly affected by many factors, and the dynamic changing workload on resources is one of the most significant one”. Many other studies also prove this conclusion. For example, in [8] Kermarrec *at el.* gives the formal expression to describe the relationship between performance and workload in large-scale distributed systems. In [9-10], Bucur and Epema conduct extensive experiments in grid testbed DAS-2 [11] to evaluate the performance of various co-allocation policies under different workload. Their experimental results show that workload-aware co-allocation policies are effective to reduce the probabilities of executing fault, since those policies are effective to avoid the negative effects brought by dynamic changing of workload on resources. In addition, the studies in [5, 7, 10, 22, 23] confirm that the conception of workload-aware is of significant importance not only for the performance of scheduler but also for the whole performance of grid systems.

In this paper, we apply queuing system [15] to describe the workload of grid resources. Recently, queuing system has been widely applied by many researchers to model the working of grid resources. For example, Sun Xian-He [12] has used queuing theory to predicate the availability of grid resources; Wu Ming [13] has applied M/G/1 queuing system to describe working model of single resource to analyze the effects of advance reservation on local job’s scheduling; Bertin [14] uses M/M/C queuing system to model the service of a single cluster to study the capability-based allocation policy in multi-cluster grid. Their studies indict that queuing system is capable of precisely describing the working and workload model of grid resources.

### III. RELIABILITY MODEL ANALYSIS AND DESIGN

#### A. TGRM Analysis

In TGRM, RMS (Resource Management System) is the root node of the tree model, which is responsible for task admission and scheduling. The resources are represented by leaf nodes, between which the edges represent the network links. TGRM assumes that the occurrence of resource and link faults follow exponential distribution. In order to simplify the evaluation of reliability, TGRM make three assumptions as following:

**A1.** The RMS is completely reliable, which means there will be no faults in RMS.

**A2.** The RMS receives tasks, assigns them to available resource, and integrates the received results for user’s tasks. All the intermediate data are assembled on RMS, and then being sent back to users.

**A3.** Each available resource can only provide service for a single user’s task.

Based on the above assumptions, TGRM gives the expressions to calculating the reliabilities of the resources and the links as following:

$$RF_j^k = e^{-\varpi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k})} \quad (1)$$

$$LF_j^k = e^{-\pi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k})} \quad (2)$$

where  $RF_j^k$  is the probability that sub-task  $j$  is successful completed on resource  $k$ ,  $LF_j^k$  is the probability that there is no occurrence of link fault during the execution of sub-task  $j$ ,  $\varpi_k$  and  $\pi_k$  are the mean fault rates of the resource  $k$  and the links,  $c_j$  is the size of sub-task  $j$ ,  $a_j$  is the amount of data transmitted between resource  $k$  and RMS when sub-task  $j$  is running,  $x_k$  is the computing speed of resource  $k$ ,  $s_k$  the minimal bandwidth of all the network links between resource  $k$  and RMS. Therefore, the reliability of a task is that

$$\text{Reliability} = \max_{1 \leq j \leq m} [\max_{k \in w_j} (RF_j^k \cdot LF_j^k)] \quad (3)$$

where  $m$  is the number of sub-tasks, and  $w_j$  is the set of selected resources for executing the task.

#### B. Workload-aware TGRM

As mentioned in Section IV, queuing system has been proven to be able to precisely describe the working and workload model of grid resources. So, we also use queuing system to construct the working and workload model of grid resources. By this way, we incorporate workload-aware mechanism into the TGRM.

Among the assumptions of TGRM, it is obviously that the third assumption is too conservative and unsuitable. For instance, multi-cluster systems are a typical class of grid system that generally used in scientific fields. In multi-cluster systems, a high-performance cluster exposes its service interface to the out-side users in form of *single-image*. So, a cluster should be considered as a single resource with paralleling computing capability. This is also true for those *Massive Parallel Processor* (MPP) systems. To be compatible with TGRM, we use M/M/1 queuing system to model those resources that can only provide service for a single task at a time, and M/M/C queuing system for those resources with parallel capability.

Meanwhile, as shown in (1), (2) and (3), TGRM ignores the waiting time of jobs when calculating the reliability. It may be feasible when resources are abundant or workload is in low-level, but not suitable to high-level workload which is common in grid environments. Another problem of TGRM is that it does not take into consideration the deadline constraint, which is often required by real-time tasks. For those real-time tasks, if it cannot be completed before the deadline the results of the tasks are of less or no value for users. So, deadline-miss should be considered a sort of resource failure.

To overcoming the above shortcomings of TGRM, we relax the third assumption of TGRM to provide supports for parallel serving resources. Also, we introduce a new type of fault, called *Deadline-Miss Fault* (DMF), into TGRM to support the reliability evaluation of real-time tasks. Therefore, equation (1), (2) and (3) listed previously is modified as following:

$$RF_j^k = e^{-\pi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k} + d_j)} \quad (4)$$

$$LF_j^k = e^{-\pi_k \cdot (\frac{c_j}{x_k} + \frac{a_j}{s_k} + d_j)} \quad (5)$$

$$\text{Reliability} = \max_{1 \leq j \leq m} [\max_{k \in w_j} (RF_j^k \cdot LF_j^k \cdot DMF_j^k)] \quad (6)$$

where  $d_j$  is the deadline requirement of sub-task  $j$ ,  $DMF_j^k$  is the probability that there is no deadline-miss when sub-task  $j$  is scheduled on resource  $k$ .

As shown in (6), the key issue of workload-aware TGRM is the approach to calculate the  $DMF_j^k$ , which is greatly affected by the workload on resources. So, in the next section, we will focus on the calculation of  $DMF_j^k$  by using queuing system.

### C. Calculation of Deadline Miss Failure

As mentioned above, workload-aware TGRM uses M/M/1 queuing system to model those resources that can only provide service for a single task at a time, and M/M/C queuing system for those resources with parallel capability (i.e. cluster or MPP). Therefore, the approaches to calculate the  $DMF_j^k$  for these two types of resources are different.

**Theorem 1.** If resource  $R_k$  is modeled as M/M/ $C_k$  queuing system, then the probability that there is no deadline-miss when sub-task  $j$  is scheduled on  $R_k$  is

$$DMF_j^k = \sum_{n=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^n}{n!} + \sum_{n=1}^{C_k \cdot \mu_k \cdot d_j - 1} \delta \cdot \frac{\rho_k^{n+C_k} \cdot C_k^{C_k}}{C_k!}$$

where  $\delta = \left[ \sum_{n=1}^{C_k} \frac{(\rho_k \cdot C_k)^n}{n!} + \frac{(\rho_k \cdot C_k)^{C_k}}{C_k} \frac{1}{1-\rho_k} \right]^{-1}$ ,  $\rho_k = \lambda_k / (C_k \cdot \mu_k)$ ,

$d_j$  is the deadline of sub-task  $j$ ,  $\lambda_k$  is the mean interval of sub-tasks on  $R_k$ ,  $\mu_k$  the mean service time of  $R_k$ , and  $C_k$  is the paralleling processing capability of  $R_k$ .

**Proof.** Let  $\psi$  be the random variable representing the number of waiting tasks in  $R_k$ . According to queuing theory [15], the probability that there are  $m$  waiting jobs in  $R_k$  is

$$\Pr\{\psi = m\} = \begin{cases} \delta \cdot \frac{\rho_k^{m+C_k} \cdot C_k^{C_k}}{C_k!}, & m > 0 \\ \sum_{n=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^n}{n!}, & m = 0 \end{cases} \quad (7)$$

$$\text{where } \delta = \left[ \sum_{n=1}^{c_i} \frac{(\rho_i \cdot c_i)^n}{n!} + \frac{(\rho_i \cdot c_i)^{c_i}}{c_i} \frac{1}{1-\rho_i} \right]^{-1} \quad (8)$$

For M/M/ $C_k$  queuing system, the service rate is  $C_k \cdot \mu_k$ , which means the system can complete  $C_k \cdot \mu_k$  tasks in a unit time. So, the amount of tasks that  $R_k$  can complete in period  $d_j$  is  $C_k \cdot \mu_k \cdot d_j$ . Therefore, the probability that  $R_k$  can guarantee a tasks's deadline  $d_j$  is equal to the probability that the waiting tasks in  $R_k$  is not more than  $C_k \cdot \mu_k \cdot d_j - 1$ . That is

$$DMF_j^k = \Pr\{\psi \leq c_i \cdot \mu_i \cdot d_j - 1\} \quad (9)$$

Combing formulas (7)(8)(9) and the above analysis, we can get that

$$\begin{aligned} DMF_j^k &= \Pr\{\psi \leq C_k \cdot \mu_k \cdot d_j - 1\} \\ &= \sum_{m=0}^{C_k \cdot \mu_k \cdot d_j - 1} \Pr\{\psi = m\} \\ &= \sum_{m=0}^{C_k} \delta \cdot \frac{(\rho_k \cdot C_k)^m}{m!} + \sum_{m=1}^{C_k \cdot \mu_k \cdot d_j - 1} \delta \cdot \frac{\rho_k^{m+C_k} \cdot C_k^{C_k}}{C_k!} \end{aligned} \quad (10)$$

**Theorem 2.** If resource  $R_k$  is modeled as M/M/1 queueing system, then the probability that there is no deadline-miss when task  $j$  is scheduled on  $R_k$  is

$$DMF_j^k = \sum_{m=0}^{\mu_k \cdot d_j - 1} (1 - \rho_k) \cdot \rho_k^m$$

where  $\rho_k = \lambda_k / \mu_k$ ,  $d_j$  is the deadline of task  $j$ ,  $\lambda_k$  is the mean interval of sub-tasks on  $R_k$ ,  $\mu_k$  the mean service time of  $R_k$ .

**Proof.** Let  $\psi$  be the random variable representing the number of waiting tasks in  $R_k$ . According to queuing theory [15], the probability that there are  $m$  waiting jobs in  $R_k$  is

$$\Pr\{\psi = m\} = (1 - \rho_k) \cdot \rho_k^m, \quad m \geq 0 \quad (11)$$

For M/M/ $C_k$  queuing system, the service rate is  $\mu_k$ , which means the system can complete  $\mu_k$  tasks in a unit time. Similar to the analysis in Theorem 1, we can obtain that

$$\begin{aligned} DMF_j^k &= \Pr\{\psi \leq \mu_i \cdot d_j - 1\} \\ &= \sum_{m=0}^{\mu_k \cdot d_j - 1} \Pr\{\psi = m\} \\ &= \sum_{m=0}^{\mu_k \cdot d_j - 1} (1 - \rho_k) \cdot \rho_k^m \end{aligned} \quad (12)$$

Based on Theorem 1 and Theorem 2, we can obtain the workload-aware TGRM by substitute the (10) or (12) into in (4), (5) and (6).

## IV. EXPERIMENTS AND PERFORMANCE COMPARISON

### A. Experimental Settings

In experiments, we use GridSim [16], a distributed resource management and scheduling simulator, to construct a multi-cluster grid model. The grid model follows the TGRM's tree-structured, and its topology and setting of individual resources are deprived from the grid test-bed DAS-2 [11]. As shown in Fig. 1, the grid model consists of twelve *Computational Elements* (CE 1 ~ CE 12), each representing a high-performance cluster. The clusters are grouped into five groups by their geographical positions. Within each group, the clusters are connected by LAN (Link 1 ~ Link 12). Then, they are connected by WAN (Link 13 ~ Link 17) between groups. The failure of links follows exponential distribution with various parameters. According to the statistics in [17, 19, 21], the failures of LAN link are different from the

WAN's. So, we set that the LAN's fault rate limited in [0.02,0.04] , and the WAN's fault rate limited in [0.04,0.12].

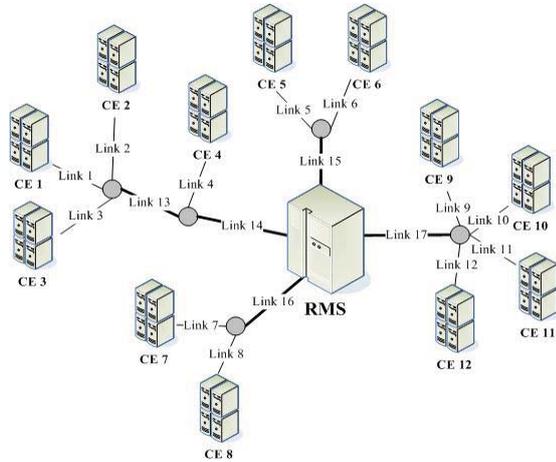


Figure 1. Grid Model in Simulations

The detailed configurations of each CE are presented in Table 1, which is also deprived from grid test-bed DAS-2.

TABLE 1  
Setting of Computational Grid Model

| ID    | Processor number | MIPS / processor |
|-------|------------------|------------------|
| CE 1  | 64               | 377              |
| CE 2  | 128              | 410              |
| CE 3  | 256              | 380              |
| CE 4  | 128              | 285              |
| CE 5  | 128              | 285              |
| CE_6  | 64               | 515              |
| CE 7  | 128              | 215              |
| CE 8  | 64               | 285              |
| CE 9  | 256              | 380              |
| CE 10 | 512              | 215              |
| CE 11 | 256              | 333              |
| CE 12 | 128              | 410              |

In simulations, the basic workload (tasks stream) is generated by using Lublin-Feitelson model [18], which is derived from the logs of real supercomputers. It consists of 10000 tasks, each is characterized by its arrival time  $T_a$ , resource demands  $R$ , execution time  $T_e$ . However, this basic workload can not meet the requirements of our simulation, because it lacks of deadline and time of data transmission. So, we modify the basic workload by append each task with deadline  $d$  and data transmission time  $T_{data}$ , which is obtained as following.

$$d = T_a + g \cdot T_e \tag{13}$$

$$T_{data} = h \cdot T_e \tag{14}$$

where  $g$  is a random variable that uniformly distributed in [5.5,10.5], and  $h$  is a uniformly distributed in [0.5,1.5]. Therefore, each task in the modified workload is characterized by five-tuple:  $\langle T_a, T_e, T_{data}, R, d \rangle$ .

*B. Comparison of Accuracy*

In simulations, we mainly focus on the accuracy of *Workload-aware TGRM* (WA\_TGRM) comparing with TGRM. Also, we investigate WA\_TGRM's performance in term of *Mean Response Time* (MRT). As mentioned in

Section 3.2, the differences between WA\_TGRM and TGRM are the reliability calculating formulas as shown in (4), (5) and (6). So, the algorithms of reliability calculation and task scheduling in WA\_TGRM are as same as that in TGRM, which are specified in the appendix of [2].

In this experiment, we first calculate the reliabilities of tasks by using TGRM and WA\_TGRM respectively, then schedule the tasks onto resources. As to each task, there are three type of executing results: success, abort (resource fault or link fault), deadline-miss fault. The experimental results are shown in Fig. 2, in which two curves represent the reliabilities calculated by TGRM and WA\_TGRM. The third curve is the real reliability that calculated as  $\frac{\text{amount of successful tasks}}{\text{amount of total tasks}}$ .

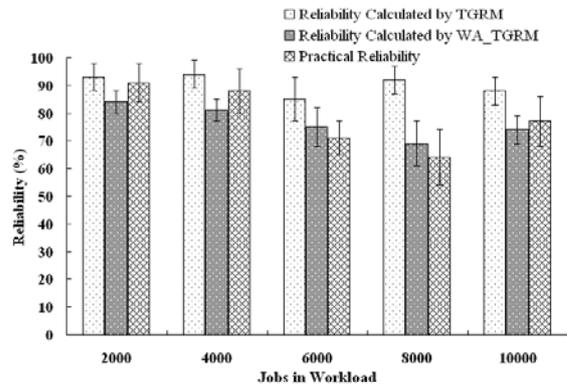


Fig. 2 Accuracy of TGRM and WA\_TGRM

We define the ratio of theoretic reliability to real reliability as the accuracy of TGRM or WA\_TGRM. As shown in Fig. 2, the accuracy of TGRM is about 97% for the first 2000 tasks, and 93% for the second 2000 tasks. As to WA\_TGRM, they are about 92% and 91% correspondingly. However, the high accuracy of TGRM does not be maintained with the continuing of the simulation. In fact, TGRM's accuracy decreases dramatically for the third and the fourth 2000 tasks, which are only about 84% and 69%. On the contrary, WA\_TGRM's accuracy is always kept above 90%. In order to examine such results, we record all the causes of unsuccessful execution and their percentages, which are shown in Fig. 3. As we can see, the percentages of resource fault and link fault are very high for the first 4000 tasks. However, percentage of deadline-miss fault keeps increasing with the continuing of the experiment, which becomes the highest (about 34%) when executing the 6000<sup>th</sup> - 8000<sup>th</sup> tasks.

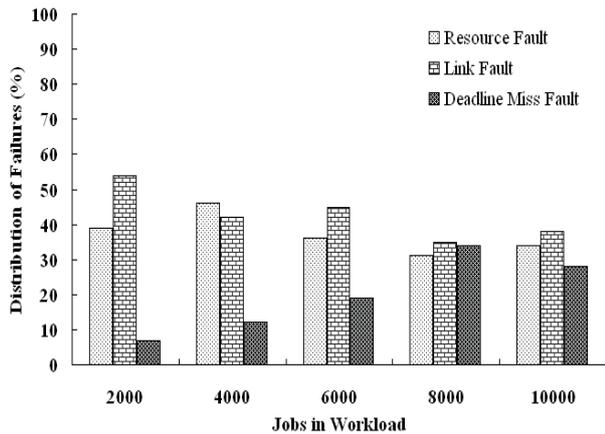


Fig. 3 Distribution of Execution Failure Causes

The detailed observation of the simulation shows that most of the tasks are scheduled onto low-load resources at the beginning. With the increasing of new arriving tasks, the workload of resources gradually becomes heavier and heavier. As a result, the waiting time of tasks increased consequently, which causes higher deadline-miss fault. As TGRM does not take the waiting time into account, nor dose it consider deadline-miss fault as a type of fault, its accuracy inevitably decreases in presence of high deadline-miss fault just as shown in Fig. 2. On the contrary, WA\_TGRM overcomes those TGRM’s shortcomings. As a result, TGRM’s accuracy can always be kept in high-level in various cases, which indicates that WA\_TGRM is more adaptive than TGRM in practice grid systems.

C. Comparison of Scheduling Performance

We investigate the scheduling performance of TGRM and WA\_TGRM in term of Mean Response Time (MRT). Although the scheduling algorithm of both TGRM and WA\_TGRM are the same, their scheduling schemes (Minimal-Task-Spanning-Tree, MTST) for individual tasks are different, because their reliability evaluation formulas are different. The experimental results are show in Fig. 4.

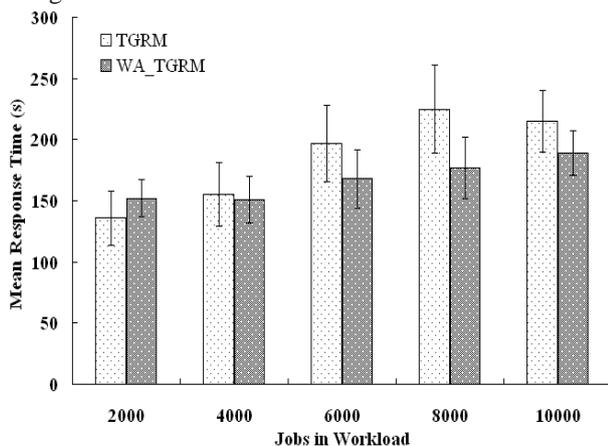


Fig. 4 Mean Response Time of Jobs

The results show that the mean response time of WA\_TGRM is about 20% lower than TGRM since after executing the first 4000 tasks. The detailed observation of

the simulation shows that the MTSTs generated by WA\_TGRM often obtains many low-load resources. As shown in (6),  $DMF_j^k$  is an important factor when calculating reliability in WA\_TGRM. The effects of  $DMF_j^k$  on the results of reliability evaluation are influenced by the resource’s workload, which can be clearly seen from the conclusions in Theorem 1 and Theorem 2. More specifically, when workload is low the effects of  $DMF_j^k$  is also lower, and vice versa. So, when the system is in face of the heaviest workload (during the 6000<sup>th</sup> – 8000<sup>th</sup> tasks) WA\_TGRM is more effective to be aware such heavy workload and generate better scheduling scheme. The experimental results further confirm the Bucur’s conclusion that “workload-aware scheduling algorithm is more effective to reduce mean response time and load-balance” in [9].

V. CONCLUSION

In this paper, we propose a novel Workload-aware Reliability evaluation model, which is deprived from TGRM. In the proposed model, queuing system is applied to describe the workload grid resources. In order to evaluate the reliability of jobs with constrain to deadline, a new type of resource fault (Deadline-Miss fault) is introduced to the proposed model. In the simulations, we compare the performance of WA\_TGRM with TGRM in terms of accuracy. The experimental results show that WA\_TGRM can provide more accurate reliability evaluation when grid system in presence of high-level workload. This indicates that WA\_TGRM is more adaptive than TGRM in practice grid systems. Also, simulations are conducted to examine the performance of reliability-based job scheduling. Experimental results show that WA\_TGRM is able to reduce the jobs’ mean response time about 15% because its ability to generate workload-aware scheduling schemes.

At present, our WA\_TGRM concentrates on the reliability evaluation when allocating resources. In practice grid systems, advance reservation is an effective technique that generally used to improve the reliability of co-allocation from multi-institutions. So, our future work is to take advance reservation into account when calculating resources’ reliability. More specific, we plan to introduce another type of fault, namely Reservation-miss Fault, into current WA\_TGRM design.

ACKNOWLEDGMENT

This work was supported by a grant from the National Natural Science Foundation of China (No. 60673165 and No. 60970038).

REFERENCES

[1] I. Foster, C. Kesselman. The Grid2: Blueprint for a New Computing Infrastructure. San Francisco: Morgan Kaufmann, 2004.

- [2] Y.S. Dai, G. Levitin. Reliability and Performance of Tree-Structured Grid Services. *IEEE Transactions on Reliability*, 2006, 55(2):337-349.
- [3] K. Czajkowski, I. Foster, C. Kesselman. Resource Co-Allocation in Computational Grids. *Proceedings of Int'l Symp. on High Performance Distributed Computing*, 1999:219-228.
- [4] Y.W. Li, Z.L. Lan. Exploit Failure Prediction for Adaptive Fault-Tolerance in Cluster. *Proceedings of IEEE Int'l Symp. on Cluster Computing and the Grid*, 2006:531-538.
- [5] Y.S. Dai, M. Xie, K.L. Poh. Reliability Analysis of Grid Computing Systems. *Proceedings of Pacific Rim Int'l Symp. on Dependable Computing*, 2002:97-104.
- [6] C.S. Raghavendra, V.K.P. Kumar, S. Hariri. Reliability Analysis in Distributed Systems. *IEEE Transactions on Computers*, 1988, 37(3):352-358.
- [7] D.J. Chen, T.H. Huang. Reliability Analysis of Distributed Systems Based on a Fast Reliability Algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 1992, 3(2):139-154.
- [8] A.M. Kermarrec, L. Massoulie, A.J. Ganesh. Probabilistic Reliable Dissemination in Large-Scale Systems. *IEEE Transactions on Parallel and Distributed System*, 2003, 14(3):248-258.
- [9] A.I.D. Bucur, D.H.J. Epema. Scheduling Policies for Processor Co-allocation in Multicluster System. *IEEE Transactions on Parallel and Distributed Systems*, 2007, 18(7):958-962.
- [10] A.I.D. Bucur, D.H.J. Epema. The Performance of Processor Co-Allocation in Multicluster Systems. *Proceedings of IEEE Int'l Symp. on Cluster Computing and the Grid*, 2003:302-309.
- [11] H. Bal, R. R. Bhoedjang, R. Hofman, et al. The Distributed ASCI Supercomputer Project. *ACM Operating Systems Review*, 2000, 34(4):76-96.
- [12] X.H. Sun, M. Wu. Grid Harvest Service: A System for Long-Term, Application-Level Task Scheduling. *Proceedings of Int'l Symp. on Parallel and Distributed Processing (IPDPS'03)*, 2003.
- [13] M. Wu, X.H. Sun, Y. Chen. QoS Oriented Resource Reservation in Shared Environments. *Proceedings of Int'l Symp. on Cluster Computing and the Grid*, 2006:601-608.
- [14] V. Bertin, J. Goossens, E. Jeannot. On the Distribution of Sequential Jobs in Random Brokering for Heterogeneous Computational Grids. *IEEE Transactions on Parallel and Distributed Systems*, 2006, 17(2):113-124.
- [15] D. Gross, C.M. Harris. *Fundamentals of Queuing Theory*. USA: John Wiley and Sons, 1998.
- [16] R. Buyya, M. Murshed. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. *Journal of Concurrency and Computation: Practice & Experience*, 2002, 14:1175-1220.
- [17] D.M. Nicol, W.H. Sanders, K.S. Trivedi. Model-Based Evaluation: From Dependability to Security. *IEEE Transactions on Dependable and Secure Computing*, 2004, 1(1):48-65.
- [18] U. Lublin, D.G. Feitelson. The Workload on Parallel Supercomputers: Modeling the Characteristics of Rigid Jobs. *Journal of Parallel and Distributed Computing*, 2003, 63(11):1105-1122.
- [19] D.K. Kim, Y. Jiao, E. Tilevich. Flexible and Efficient In-Vivo Enhancement for Grid Applications. *Proceedings of International Symp. on Cluster Computing and the Grid*, 2009:444-451.
- [20] H. Zhao, X.L. Li. Efficient Grid Task-Bundle Allocation Using Bargaining Based Self-Adaptive Auction. *Proceedings of International Symp. on Cluster Computing and the Grid*, 2009:4-11.
- [21] M. Waldburger, M. Göhner, H. Reiser, G.D. Rodosek, B. Stiller. Evaluation of an Accounting Model for Dynamic Virtual Organizations. *Journal of Grid Computing*, 2009, 7(2):181-204.
- [22] D. Lingrand, J. Montagnat, T. Glatard. Modeling the Latency on Production Grids with Respect to the Execution Context. *Proceedings of Int'l Symp. on Cluster Computing and the Grid*, 2008:753-758.
- [23] K Verstoep, J Maassen, H E Bal, J W Romein. Experiences with Fine-grained Distributed Supercomputing on a 10G Testbed. *Proceedings of Int'l Symp. on Cluster Computing and the Grid*, 2008::376-383.

**Peng Xiao** was born in 1979. He received his master degree in Xiamen University in 2004. Now, he is a Ph.D candidate in Central South University. His research interests include grid computing, parallel and distributed systems, network computing, distributed intelligence.

**Zhigang Hu** was born 1963. He is a Professor and Ph.D supervisor in Central South University. Currently, he is the Chief Secretary of Computer Science in Hunan Province. His research interests including grid computing, embedded systems, high-performance platform.