Availability Modeling of Grid Computing Environments Using SANs

Reza Entezari-Maleki
Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
E-mail: entezari@ce.sharif.edu

Ali Movaghar
Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
E-mail: movaghar@sharif.edu

Abstract: **In this paper, the availability of the Resource Management System (RMS) and computational resources distributed within grid computing environments is studied. Since the RMS acts as a heart of the grid environments, the unavailability of this system can render the entire environment to the inoperable phase. Furthermore, the unavailability of the grid resources may result in degradation of the performance of the grid. Therefore, considering the great importance of the availability issue in grid computing environments, the Stochastic Activity Networks (SANs) are exploited to model and evaluate the availability of grid environments. The proposed SAN models the failure of the resource management servers and grid resources, filling the queues of the RMS and resources and the arriving of the local and grid tasks to the resources, appropriately. Our proposed model not only can evaluate the availability of the RMS and grid resources, but also can consider the impact of the task scheduling algorithms on the grid availability. Simulations done on the hypothesis grid environments show the applicability of the model to the real grids.**

1. INTRODUCTION

Grid computing environments are composed of numerous different and heterogeneous resources distributed within multiple organizations and administrative domains [1, 2]. Computational grids have been found to be powerful environments to solve the computational and data intensive problems in science and industry. To use the tremendous capabilities of the grid environments, users should deliver their own requests to the Resource Management System (RMS). The RMS receives service requests from grid users and then converts the requests to the appropriate tasks to be dispatched within grid resources [3-5]. Availability of the RMS and grid resources is very important because failure of the RMS or all of the grid resources prevents users from using the grid capabilities. If the RMS fails or becomes unavailable, the grid environment will be unavailable for the grid users. In this situation, the availability of grid resources is not important because the unavailability of the RMS transforms the entire grid environment to the inoperable stage. On the other hand, if the RMS is available but the grid resources become unavailable, performance of the environment decreases. If all of the grid resources become unavailable, the whole grid environment will be unavailable.

Since the availability of the grid environment is one of the most important user satisfaction factors, this measure should be considered and suitable models for availability evaluation and analyzing should be presented. To fulfill this requirement, some research works have been done to evaluate the availability of the grid systems as well as the reliability of grid services. Dai et al. [5, 6] have studied the service reliability and availability of a wide-area distributed system which is one of the ancestors of the grid systems. Based on the model proposed in Ref. [6], the distributed service reliability which is defined as the probability of successfully providing the service in a distributed environment can be evaluated. The function of the control center in that model is similar to the RMS in computational grids. Reference [5] proposes an availability model for grid RMS using hierarchical Markov reward models. The model only studies the availability of the RMS and does not take any consideration to the availability of the grid resources. In addition, some research works have been carried out to model and evaluate the reliability of the grid environments. Levitin et al. [3, 7] have studied the grid services reliability, which is defined as a probability that the correct output is produced in time less than a predefined time. In Ref. [7], the dependency between the tasks has been considered, as well. Azgomi et al. [4] have presented a colored Petri net model to evaluate the reliability of the service execution in computational grids. The model uses CPN Tools to graphically model and analyze the reliability of a hypothesis grid environment. In addition to these works, some other research works have applied diverse modeling and analyzing approaches to model and evaluate the performance, availability and reliability of distributed systems [8-10].

To address the necessity for a suitable availability model for grid environments, this paper exploits Stochastic Activity Networks (SANs) [11] and proposes a model to evaluate the availability of the entire grid environment. A pervasive availability model for a grid computing environment should take into account the failure of the RMS and grid resources simultaneously. In the proposed model, the availability of both RMS and grid resources is considered. Furthermore, the impact of the task scheduling algorithms on the grid availability is taken into account. Applying appropriate scheduling algorithms to dispatch grid tasks in the resource management servers, the availability of the environment can be improved without using any redundant hardware or software. The simulation results obtained from different case studies show that the availability of the grid environment is dependent on various factors such as the task queue capacity of the RMS, the failure and repair rates of the resource management servers and grid resources, task scheduling algorithms applied by the servers to schedule the tasks on grid resources, execution priority between the local and grid tasks in each of the resources and so forth.

The rest of this paper is organized as follows. Section 2 proposes a SAN-based model for evaluating the availability of the grid computing environments. Section 3 presents the results obtained from simulating various task scheduling algorithms in a hypothesis grid environment using proposed SAN. Finally, section 4 concludes the paper and presents future work.

2. THE PROPOSED MODEL

In order to model the availability of grid computing environments, the required assumptions should be mentioned. Therefore, the assumptions of the availability model are firstly given, then the model is presented. To model the availability of the grid, two separate models are constructed: the first one for the RMS and the second one for the grid resources. Consequently, joining the models and applying task scheduling algorithms in resource management servers, the entire availability model of the grid environment can be constructed.

2.1 Model Assumptions

The assumptions needed for modeling the availability using SANs are given as follows.

1. There are *N* different resource management servers within the RMS running in parallel. These servers process the tasks existing in the RMS queue to find suitable resources for executing them.

2. The capacity of the task queue of RMS is finite and it cannot exceed *M*.

3. The arrival of the service requests to the RMS follows the Poisson process with the arrival rate λ_g.

4. The servers of the RMS may be down because of certain software or hardware failures. The failure occurrence on server *i* follows a Poisson process with the failure rate λ_{si}. The failures of different servers are independent of one another.

5. The repair time of the resource management servers follows the exponential distribution with the repair rate μ_{si} for each server *i*.

6. The service time of the resource management server *i* is exponentially distributed with the parameter μ_{seri}.

7. If all the *N* resource management servers are down, no grid task can be serviced by the RMS. Moreover, if the task queue exceeds its limitation of *M*, the new requests cannot be accepted by the RMS. Then, the availability of the RMS is the proportion of time in which none of aforementioned conditions occurs.

8. There are *R* different computational resources distributed within the grid environment. The failure occurrence in grid resource *j* follows a Poisson process with the failure rate λ_{rj}. The failures of the different resources are independent of one another.

9. The repair time of the grid resources follows the exponential distribution with the repair rate μ_{rj} for the grid resource *j*.

10. In each of the administrative domains, the local users submit their own tasks to the resources. The arrival of the local tasks to the resource *j* follows the Poisson process with the arrival rate λ_{lj}.

11. The service time of the resource *j* is exponentially distributed with the rate μ_{resj}.

12. Based on the scheduling policy in each of the administrative domains, the execution priority of the local and grid tasks can be varied. If the execution priority of the local and grid tasks is the same, the grid resource will be available to execute the grid tasks when it is up (not failed), and if the local tasks have higher priority than grid ones, the resource will be available for executing the grid tasks when it is up and there is no local task in its queue.

The aforementioned assumptions are very common in context of grid modeling and can be found in many research works [3-7, 9, 12, 13].

2.2 RMS Availability Model

The grid RMS availability is defined as the proportion of time that the system is available to the grid users. Based on this definition, two different conditions can be considered cause the RMS to be unavailable to the users [5, 6].

1. Failure of all of the resource management servers which cause the unavailability of the RMS in receiving and servicing the new tasks.

2. Fullness of the task queue of RMS in which prevents the new tasks to be submitted to the RMS.

In order to model the availability of the server *i*, the failure and repair rates of the server should be specified. Considering assumptions 4 and 5, the failure and repair of the server *i* can be modeled using two timed activities with exponential time distribution functions with the rates λ_{si} and μ_{si}, respectively. Figure 1 shows the failure and repair model of the server *i*. If there is a token in the place *Serveri_Up*, the server is up and it can service the submitted tasks. Existing a token in the place *Serveri_Down* shows that the server *i* has been failed and it should be repaired to be ready to service the tasks. In the beginning, there is one token in the place *Serveri_Up*. Firing the timed activity *Serveri_Failure* moves a token from place *Serveri_Up* to the place *Serveri_Down* and therefore, transforms the server *i* from up-state to the down-state and firing the timed activity *Serveri_Repair* performs the reverse action.

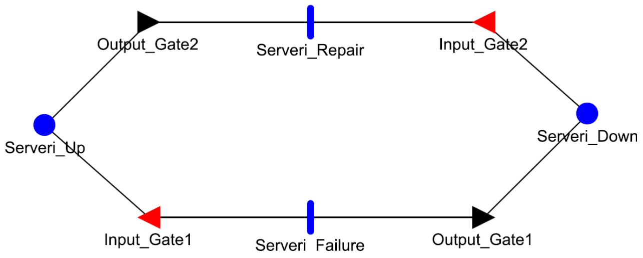

Fig. 1. Failure and repair model for the server *i*

In order to model the arrival of the service requests to RMS, the SAN model depicted in Fig. 2 is suggested. In this model, the arriving of the requests to the RMS is modeled by the timed activity *Requests_Arrival*. The time distribution function associated with this timed activity is an exponential function with the rate of λ_g. Furthermore, task queue of the RMS is modeled using a place named *RMS_Queue*. Since the maximum capacity of this queue is equal to *M*, the further tasks will be rejected when the number of the tasks existing in the queue reaches to *M*.

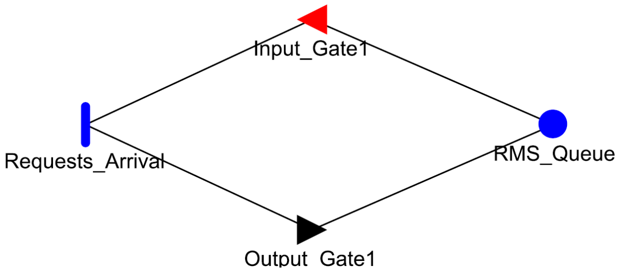

Fig. 2. Arriving of the requests to the RMS

Considering the failure and repair model of the resource management servers and the model of the requests arriving to the RMS, the task processing model of the servers can be depicted as Fig. 3. As shown in Fig. 3, the timed activity *Serveri* models processing of the tasks inside server *i* to find a suitable resource for executing the tasks. An exponential function with the rate $\mu_{\mathrm{ser}i}$ is assigned to this timed activity as its time distribution function.

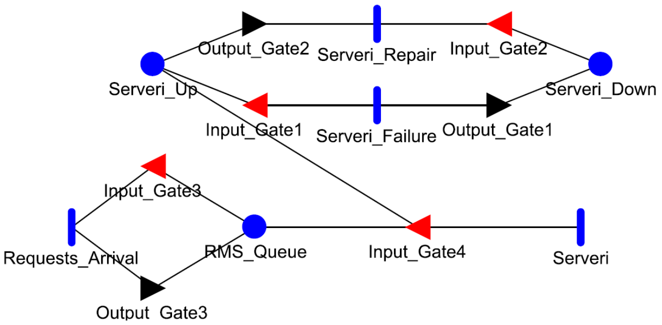

Fig. 3. Task processing model for the server *i*

2.3 Grid Resources Availability Model

The failure and repair model of the grid resources is the same as the model described for the resource management servers. Then, the model shown in Fig. 1 can be applied to the grid resource *j* with the failure and repair rates equal to $\lambda_{\mathrm{r}j}$ and $\mu_{\mathrm{r}j}$, respectively. Also, the arrival of the local tasks to the grid resources can be modeled as the same as the arrival of the service requests to the RMS. If the local tasks are prioritized to the grid ones, then two distinct queues for each of the grid resources can be considered. In this situation, the local and grid tasks are submitted to the local and grid queues of the resources, respectively. Figure 4 shows the availability model of the grid resource *j* in which two separate queues are considered for that resource.

In Fig. 4, the timed activity *Local_Tasks_Arrivalj* shows the arriving of the local tasks to the resource *j*. The time distribution function associated with this timed activity is an exponential function with the rate of $\lambda_{\mathrm{l}j}$. The local tasks queue of the resource *j* is modeled using a place named *Resourcej_Local_Queue*. In addition, the grid tasks queue of the resource *j* is modeled by the place *Resourcej_Grid_Queue*. If there is no priority between the local and grid tasks, then the place *Resourcej_Local_Queue* can be removed. In this situation, the output of the timed activity *Local_Tasks_Arrivalj* should be connected to the place *Resourcej_Grid_Queue*. The timed activity *Resourcej* models the execution of the local and grid tasks within the resource *j*. The time distribution function assigned to this activity is an exponential function with the rate of $\mu_{\mathrm{res}j}$. The enabling predicates and input and output functions of the model shown in Fig. 4 is very similar to the model shown in Fig. 3.

2.4 Joining the Models Using Scheduling Algorithms

After constructing the availability model of the RMS and grid resources, the models should be joined to form the entire availability model of the grid environment. To do this, the output of the timed activities *Serveri* for each server *i* should be connected to the places *Resourcej_Grid_Queue* representing the grid tasks queue of the resource *j*. To achieve this, task scheduling algorithms can be defined in output function of the output gates related to the timed activities *Serveri*. Using the scheduling algorithms, the assignment of the tasks to the grid resources can be handled. Various task scheduling algorithms have been presented to appropriately dispatch the tasks to the resources [9, 13, 14]. Based on the resource management strategies and the administrative policies existing inside each of the administrative domains, different task scheduling algorithms can be applied to increase the availability of the grid environments. One of the most successful strategies which can be used to schedule the tasks to achieve the good amount of availability is assigning the tasks to the resource that has

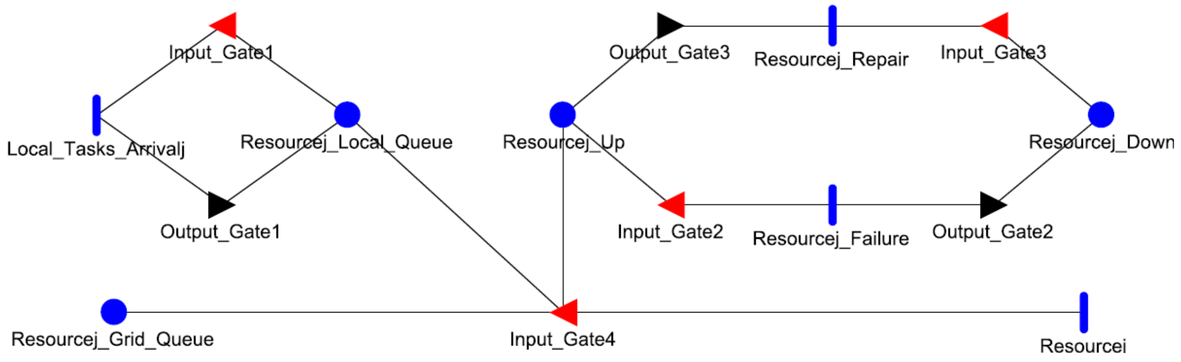

Fig. 4. The availability model of the grid resource *j* with two distinct queues for grid and local tasks

the least number of waiting tasks. However, other scheduling strategies can be applied and then compared to each other in terms of their influence on the overall availability of the grid environment. In the next section, four different task scheduling algorithms are simulated and the availability of the grid environment is evaluated in presence of them.

3. SIMULATION RESULTS

We use Möbius tool [15] to model and analyze the proposed SAN. To be able to model and evaluate the availability of the grid environments using Möbius, a sample grid environment with the predefined number of resource management servers and grid resources can be considered. Note that, the model presented in this paper is general and can be applied to any grid environment with different number of servers and resources.

Consider a grid computing environment with three different grid resources R_1, R_2 and R_3. The RMS of the environment is also composed of two heterogeneous servers with the names S_1 and S_2. The availability model of such grid environment can be constructed using SAN models shown in Fig. 1 to Fig. 4, but since the entire availability model is very huge, it is not depicted here. In order to compare the impact of the task scheduling algorithms on the availability of the grid environment, four different scheduling algorithms in the form of four different case studies named *case 1*, *case 2*, *case 3* and *case 4* are simulated. For the sake of brevity, the functions representing the code of the scheduling algorithms are not presented here, but the description about the algorithms is provided in the below.

In *case 1*, a very simple and straightforward scheduling algorithm is defined. In this case, the tasks are dispatched among the grid resources R_1, R_2 and R_3 without any consideration about their processing speeds and task queues. In the first step, the capacity of the task queue of the resource R_1 is checked. If there is an empty space in the queue, the new task is assigned to R_1; otherwise the waiting queue of the resource R_2 is checked. This procedure is continued until the task is scheduled or all of the grid resources have been found to be filled. If all of the grid resources are completely loaded, then the new task cannot be scheduled and the grid resources will be unavailable to the new tasks until at least one of the resources becomes available.

In *case 2*, the number of the tasks waiting to be processed within the grid resources is taken into account and the new task is assigned to the resource with more empty spaces in its task queue. Actually, this algorithm tries to balance the number of the waiting tasks between the grid resources and thereby uses the time slots in which the resources are left free in the algorithm illustrated in the *case 1*. Therefore, it is expected that the availability of the environment is increased when this algorithm is applied. In *case 3*, the algorithm explained in the *case 2* is extended to consider the availability status of the grid resources in scheduling phase. In this algorithm, both the number of the empty spaces in task queue of the resource and the availability of that resource are taken into account and a task is assigned to the resource when both conditions are satisfied.

The scheduling algorithm defined in *case 4* schedules the tasks considering the processing speeds of the grid resources. In this algorithm, tasks are firstly scheduled on the grid resource with high processing capability. After filing the most powerful resource, the second computationally powerful resource is selected and the tasks are scheduled on it. This process is continued until the tasks are scheduled or all of the grid resources have been found to be filled. If all of the grid resources are satiated, the new task cannot be scheduled until at least one of the resources becomes free. The availability of the model in all of the case studies is defined using a reward function capability in Möbius. In each experiment, this function returns one if there is a free space in the request queue of the RMS, else it returns zero. Hence, if the RMS can accept a new request from a grid user, then we say that the grid environment is available for the grid users, otherwise the grid will be unavailable.

In all of the experiments, the request arrival rate to the RMS (λ_g) is equal to 10 tasks per second. In the first experiment, the RMS queue capacity is varied from 100 to 1000 tasks and the steady state availability of the grid environment is evaluated. Figure 5 shows the results obtained from the first experiment. As shown in Fig. 5, task scheduling algorithm defined in *case 1* shows low availability in comparison with other algorithms. It should be mentioned the algorithms correspond to the *case 2*, *case 3* and *case 4* show an acceptable level of availability. Also, the availability of the *case 2* and *case 3* becomes better when the queue capacity of the RMS is higher. Figure 6 shows the results of the second experiment in which the instantaneous availability of the assumed grid environment is calculated. In this experiment, the RMS queue capacity is 500 tasks and the time is varied from 0 to 200 seconds with the increment step equal to 20. As shown in Fig. 6, task scheduling algorithm defined in the *case 3* shows higher availability than other algorithms. Considering Fig. 5 and Fig. 6 it can be concluded that the scheduling algorithms which consider the capacity of the grid queue of the resources, act better than other similar algorithms and provide relatively high availability in comparison with the others.

In another experiment, the SAN model is changed to eliminate the priority of the local tasks to the grid ones, and therefore remove the local queue of the resources. In this situation, a single queue is considered for each of the grid resources and the capacity of this queue is equal to the sum of the grid and local queues capacities used in the previous experiments. Applying this modification to the aforementioned experiments, the results shown in Fig. 7 and Fig. 8 can be obtained. As shown in Fig. 7, task scheduling algorithm defined in *case 4* shows higher availability in comparison with other algorithms. Figure 8 further shows this claim where the availability of the algorithm correspond to the *case 4* overcomes other algorithms' availabilities. It should be mentioned that this result was predictable, because in two last experiments, there is no priority between local and grid tasks, and therefore it is very reasonable to consider the processing capabilities of the resources as a scheduling metric instead of the number of the free spaces in waiting queue of the resources.

4. CONCLUSIONS AND FUTURE WORK

Availability of the grid environment to accept service requests from the grid users is one of the most important QoS factors in grid context. To evaluate the availability of the grids and study the impact of the task scheduling algorithms on the entire system availability, a pervasive model is required. In this paper, a model based on SANs has been presented to graphically model and evaluate the availability of the grid RMS and grid resources when the task scheduling algorithms are applied. The model takes into account the failure of the resource management servers and grid resources influencing the availability of the system. Furthermore, the filling of the RMS queue and the local and grid tasks queuing policies has been considered in the proposed model.

There are numbers of research issues remaining open for future work. One can model and evaluate other QoS measures such as the reliability and performance of the grid environments using SANs. Modeling the reliability and availability of the grid RMS and grid resources and then computing the performance and throughput of the grid environment is one of the most interesting issues in this field. Therefore, we can model and compute the performability of grid computing environments using SANs. Moreover, the impact of the global task scheduling algorithms inside the resource management servers for dispatching the grid tasks among grid resources and the local scheduling policies for scheduling the local and grid tasks within each of the administrative domains can be taken into account in the new performability model. In addition, modeling using queuing networks and Markov reward models may result in new and valuable research works.

REFERENCES

[1] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, 2nd ed., Morgan Kaufmann, 2004.

[2] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing," *Software: Practice and Experience*, Vol. 32, No. 2, pp. 135–164, 2002.

[3] G. Levitin and Y. S. Dai, "Service Reliability and Performance in Grid System with Star Topology," *Reliability Engineering and System Safety*, Vol. 92, No. 1, pp. 40–46, 2007.

[4] M. A. Azgomi and R. Entezari-Maleki, "Task Scheduling Modelling and Reliability Evaluation of Grid Services Using Coloured Petri Nets," *Future Generation Computer Systems*, Vol. 26, No. 8, pp. 1141–1150, 2010.

[5] Y. S. Dai, M. Xie, and K. L. Poh, "Availability Modeling and Cost Optimization for the Grid Resource Management System," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, Vol. 38, No. 1, pp. 170–179, 2008.

[6] Y. S. Dai, M. Xie, K. L. Poh, and G. Q. Liu, "A Study of Service Reliability and Availability for Distributed Systems," *Reliability Engineering and System Safety*, Vol. 79, No. 1, pp. 103–112, 2003.

[7] G. Levitin, Y. S. Dai, and H. Ben-Haim, "Reliability and Performance of Star Topology Grid Service with Precedence Constraints on Subtask Execution," *IEEE Transactions on Reliability*, Vol. 55, No. 3, pp. 507– 515, 2006.

[8] S. Jian, W. Shaoping, and S. Yaoxing, "Petri-Nets Based Availability Model of Fault-Tolerant Server System," in *2008 IEEE Conference on Robotics, Automation and Mechatronics*, September 2008, pp. 444–449.

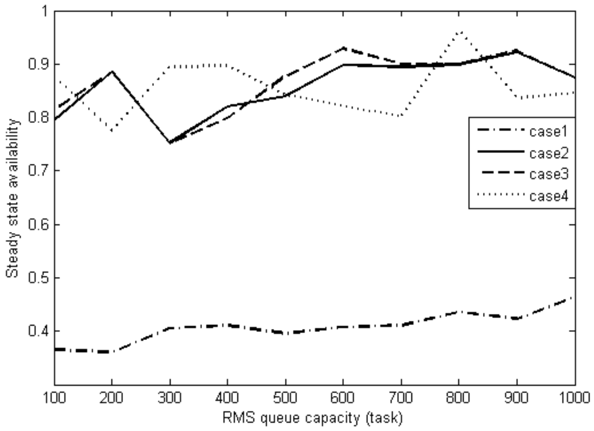

Fig. 5. Steady state availability for the various RMS queue capacities in the *Experiment 1*

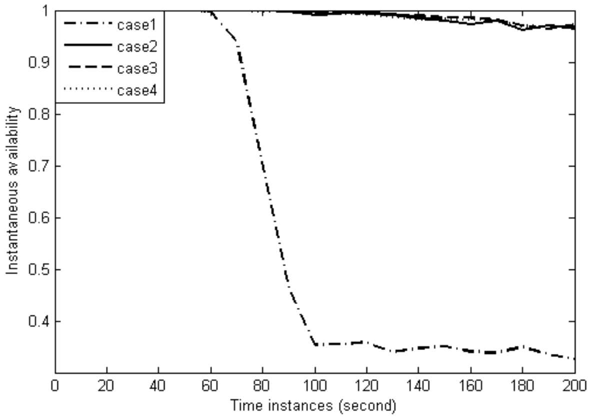

Fig. 6. Instantaneous availability in different time instances in the *Experiment 2*

[9] S. Parsa and R. Entezari-Maleki, "Modeling and Throughput Analysis of Grid Task Scheduling Using Stochastic Petri Nets," in *International Conference on Parallel and Distributed Processing Techniques and Applications*, July 2009, pp. 458–464.

[10] C. D. Lai, M. Xie, K. L. Poh, Y. S. Dai, and P. Yang, "A Model for Availability Analysis of Distributed Software/Hardware Systems," *Information and Software Technology*, Vol. 44, No. 6, pp. 343–350, 2002.

[11] A. Movaghar and J. F. Meyer, "Performability Modeling with Stochastic Activity Networks," in *The 1984 Real-Time Systems Symposium*, December 1984, pp. 215–224.

[12] S. Parsa and R. Entezari-Maleki, "Task Dispatching Approach to Reduce the Number of Waiting Tasks in Grid Environments," *to appear in* The Journal of Supercomputing.

[13] Y. Qu, C. Lin, Y. Li, and Z. Shan, "Performability evaluation of resource scheduling algorithms for computational grids," in *5th International Conference on Grid and Cooperative Computing*, October 2006, pp. 319–326.

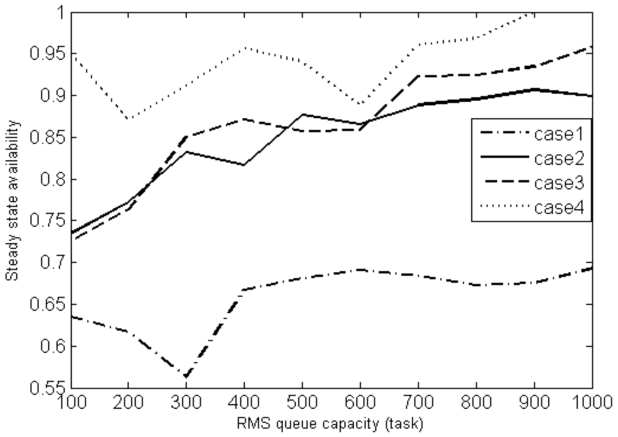

Fig. 7. Steady state availability for the various RMS queue capacities in the *Experiment 3*

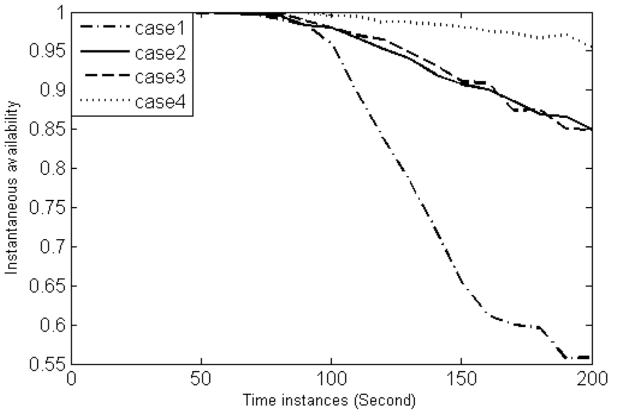

Fig. 8. Instantaneous availability in different time instances in the *Experiment 4*

[14] R. Entezari-Maleki and A. Movaghar, "A Genetic-Based Scheduling Algorithm to Minimize the Makespan of the Grid Applications," in *Grid and Distributed Computing, Control and Automation*, ser. Communications in Computer and Information Science, T. Kim et al., Eds., Vol. 121. Springer, December 2010, pp. 22–31.

[15] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders, "Möbius: An Extensible Tool for Performance and Dependability Modeling," in *11th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools*, ser. Lecture Notes in Computer Science, B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith, Eds., Vol. 1786. Springer, March 2000, pp. 332–336.