J Supercomput
DOI 10.1007/s11227-014-1184-z

Combined performance and availability analysis of distributed resources in grid computing

Reza Entezari-Maleki · Ali Mohammadkhan · Heon Young Yeom · Ali Movaghar

Abstract This paper presents a mathematical model to evaluate the performance of grid resources when availability of the resources is taken into account. The proposed model uses continuous time Markov chains (CTMCs) to model the failure-repair behavior of a grid resource. In grid computing environment, a resource not only may fail during task execution, but also it can cancel its membership at any time. Hence, the proposed CTMC considers the availability of a grid resource to a grid user in both failure and membership refusal situations. After modeling the availability of the resource, the mean sojourn time of grid tasks in each of the availability states is estimated. Assigning the mean sojourn times of the tasks as performance levels to each of the CTMC's states, a Markov reward model (MRM) representing the combined performance and availability measures is obtained. Computing the cumulative state probability of the CTMC and multiplying reward rates of the MRM's states to each of the corresponding state probabilities, the expected accumulated sojourn time of grid tasks in each of the grid resources is achieved. An illustrative example is presented and the results obtained from the proposed model are reported in cases where various scheduling disciplines are considered inside the grid resource to simultaneously service grid and local tasks.

R. Entezari-Maleki (✉) · A. Mohammadkhan · A. Movaghar
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
e-mail: entezari@ce.sharif.edu

A. Mohammadkhan
e-mail: mohammadkhan@ce.sharif.edu

A. Movaghar
e-mail: movaghar@sharif.edu

H. Y. Yeom
School of Computer Science and Engineering, Seoul National University, Seoul, South Korea
e-mail: yeom@snu.ac.kr

Published online: 17 April 2014

Keywords Grid computing · Performance · Availability · Markov reward model · Mean sojourn time

Abbreviations

FTCS	Fault tolerant computer system
CTMC	Continuous time Markov chain
MRM	Markov reward model
RMS	Resource management system
GSPN	Generalized stochastic Petri net
SAN	Stochastic activity network

List of symbols

t	Time
i	Index of the system
α	Failure rate of the resource
β	Repair rate of the resource
λ_l	Local tasks arrival rate
μ_l	Resource service rate for local tasks
λ_g	Grid tasks arrival rate
μ_g	Resource service rate for grid tasks
μ	Total service rate of the resource
$Z(t)$	t $\geq$ 0, a random process representing the related CTMC
Ω	State space of CTMC
N	Number of states in CTMC
Q	Generator matrix
$P(t)$	Transient probability vector
p_0	Initial probability vector
π	Steady state probability vector
$L(t)$	Cumulative state probability vector during time period $[0, t)$
$X(t)$	Instantaneous reward rate of the related MRM
r	Reward rate vector over $Z(t)$
$\Phi(t)$	Accumulative reward over the period $[0, t)$
$E[X(t)]$	Expected instantaneous reward rate
$E[X]$	Expected steady state reward
$E[\Phi(t)]$	Expected accumulated reward rate

1 Introduction

Grid computing environments are composed of many diverse and heterogeneous resources distributed within multiple organizations and administrative domains [1,2]. Computational grids have been found to be very powerful environments to solve the computational and data intensive problems. To use the tremendous capabilities of the grid computing environment, grid users should deliver their own requests, named tasks, to the environment. After that, grid resources can service the tasks and return the results to the users [3–5]. To deliver the tasks to the grid environment, distributed resources should be available to interact with grid users. Since the grid resources are autonomous and self directed, each of the resources can join the environment or cancel its own membership during servicing the tasks [1,6]. Moreover, a resource can fail to execute the tasks at any time. Therefore, the availability of a grid resource can be influenced by various factors such as resource failure, communication link failure, and refusing or canceling the membership in the environment. On the other hand, grid resources should service local tasks submitted to the resources directly by local users. As mentioned earlier, grid resources belong to different virtual organizations and administrative domains which share their own resources to capture the grid environment. When a user existing in a resource's administrative domain submits a task to the resource, this task generally has higher execution priority over the tasks submitted by the grid users, called grid tasks [1,6–8]. This is a very normal behavior because a grid resource voluntarily joins the grid environment and executes grid tasks. This joining is done taking care not to disturb the local tasks' execution. Consequently, a grid task can only be executed within a grid resource when there is no local task in the waiting queue of the resource. In general, there are some other scheduling disciplines which allow grid tasks to be executed with local tasks, simultaneously. In this paper, three scheduling disciplines, first in first out (FIFO), non-preemptive and preemptive priorities [9–11], are considered.

In addition to the resource availability, performance of the grid is also considered as one of the most important user satisfaction factors. In grid computing environment, performance evaluation mainly focuses on completion time of the tasks submitted by grid and/or local users. In this respect, several related measures such as the expected waiting time, the mean service time and the expected sojourn time of the tasks can be considered as performance measures [3,4,12–15]. In traditional performance evaluation, performance measures are assessed without any consideration of resource availability and/or reliability. Nevertheless, in highly distributed systems (e.g., grid environment), each of the resources can be added to or removed from the system. Therefore, the performance of the system highly depends on the number and processing power of the existing resources. This kind of systems in which the performance of the system can differ in various times is considered as dependable fault tolerant computer systems (FTCSs), because when a resource fails, other resources in the environment can service the task assigned to the failed resource. In other words, grid environment can service the grid users even in the presence of several resource failures albeit in reduced performance level. Actually, this is one of the most popular characteristics of dependable FTCSs which provides continuity of service despite component failures. However, the performance delivered by the system may degrade in this situation [16,17].

Therefore, analyzing the pure performance of the grid environment tends to be optimistic since it ignores the failure-repair behavior of grid resources. As mentioned earlier, the failure of a resource contains both actual system failures resulted from system and environmental faults and membership canceling which both lead to disturbance of task execution. On the other hand, pure availability analysis tends to be conservative since performance considerations are not taken into account. Consequently, combined performance and availability evaluation of the grid computing environment can present the most realistic view of the grid system behavior and help to appropriately estimate the completion time of the submitted tasks. To compute this composite measure, a Markov reward model (MRM) is presented in this paper to model the structural behavior of a grid resource, and then, analyze the performance of the resource in servicing grid tasks. In the proposed model, the structural behavior of the grid resource (failure-repair behavior) is modeled using continuous time Markov chain (CTMC), and then, appropriate reward rates are assigned to the states of the obtained CTMC to capture a MRM. The reward rates which show the performance levels of states of CTMC are computed using the queuing systems related to each of the structural states of the proposed CTMC. In the proposed model, the expected sojourn time of grid tasks is used as the performance level of each state in CTMC. Assigning the performance levels to the corresponding states and analyzing the obtained MRM in various time intervals, the expected accumulated sojourn time of grid tasks can be estimated in a grid resource. Considering different priority disciplines between grid and local tasks results in various rewards which can be assigned to states of the proposed CTMC to model and analyze more realistic situations of grid resources.

The remaining parts of the paper are organized as follows. Section 2 introduces the related research work done on composite performance and availability/reliability evaluation in various FTCSs. Section 3 presents some preliminaries on MRM and the combined performance and availability evaluation. In Sect. 4, the proposed model for composite performance and availability analysis of a grid resource is presented. A detailed example of how one can use the proposed model to simultaneously evaluate the performance and availability of a grid resource is given in Sect. 5. Finally, Sect. 6 concludes the paper and presents future work.

2 Related work

Many analytical models have been proposed to evaluate the performance and dependability measures of various FTCSs. Some of these models evaluate performance and dependability measures separately and some others simultaneously take both of them into account. Actually, the proposed models, evaluation methods and evaluated measures are very different in previously done research work, and they highly depend on the system under study. In the following, some of the related research papers in this area are introduced.

Dai et al. [18] have proposed a hierarchical MRM to evaluate the availability of resource management system (RMS) in grid computing environments. The proposed model in [18] appropriately considers the waiting queue of RMS and failure-repair behavior of servers existing inside RMS. After evaluating the RMS availability, a method to simultaneously analyze the availability and cost measures has been proposed. Entezari-Maleki et al. [6] have extended the model proposed in [18] to include the availability of grid resources to present the most realistic view of entire grid environment. The model proposed in [6] is based on the stochastic activity networks (SANs) which presents formal description and graphical representation of the problem.

Parsa et al. [14,19], Levitin et al. [3] and Dai et al. [4] have investigated the reliability and performance of grid environments. In [14], the performance of a grid environment in terms of the mean number of tasks waiting to be processed within the grid was investigated, and a queuing network solution together with a generalized stochastic Petri net (GSPN) was proposed to model the environment and evaluate the performance. The models proposed in [14] only evaluate the performance measure and are not able to simultaneously take the dependability concepts into account. A method to estimate the reliability of a service and mathematical expectation of service time was proposed in [19]. The proposed approach takes both permanent and transient failures into account. In addition, the passive replication scheme was considered in [19], and the reliability and service time in the presence of passive parallel replicated resources were analyzed. In both [3] and [4], universal generating function (u-function) technique was used to estimate the probability mass function (pmf) of grid service time. Using u-function, the pmf of random variable task completion time was evaluated, and then, the service reliability and performance of the grid services were analyzed by the obtained pmf. The network topologies of the grid systems considered in [3] and [4] are star and tree topologies, respectively. In star topology, all of the distributed resources are directly connected to the RMS, but in tree topology, the root of the tree is RMS, and the leaves and intermediate nodes represent distributed resources. Hence, in tree structure, when an intermediate node fails, all of the nodes existing in its sub-tree fail. Consequently, common cause failures or single point of failures should be considered in tree structures. This problem was appropriately considered in [4]. Azgomi et al. [5] have presented a task scheduling model and an evaluation framework based on colored Petri nets (CPNs) to evaluate the reliability of grid services. The model proposed in [5] exploits colored tokens to keep the path information within the Petri net, and then, uses the minimum and maximum functions to calculate service reliability. The reliability definition in [5] is based on service failure probabilities and service completion times.

Trivedi et al. [16, 20, 21] have presented some useful approaches to analyze the performance and dependability concepts. In [20] and [21], several practical issues together with solutions of dependability and performability models have been presented. In addition to CTMCs and MRMs, other approaches such as GSPNs and stochastic reward nets (SRNs) were introduced to evaluate the performance and dependability measures of FTCSs. In [16], MRMs and their extensions to semi-Markov reward models were studied to be used in composite performance and dependability analysis. After introducing the required concepts, the structural behavior of two FTCSs, a multiprocessor system and a multi-bus multiprocessor system, were modeled using CTMCs. After that, some rewards such as task rejection probability, bandwidth availability, expected number of processor hours and normalized computational availability which are common performance measures in multiprocessors were assigned to the states of the obtained CTMC. Meyer [22] coined the term *performability*, and he applied the performability models to computer system analysis. In [22], the performance and reliability of degradable computing systems were considered, and the combined evaluation of performance and reliability was named *system effectiveness evaluation*. To evaluate the performability of a system which shows the system effectiveness, a function called capability function is defined. The capability function relates low-level system behavior to user-oriented performance levels. In fact, this function computes reward rates assigned to each of the reliability states.

Ma et al. [23] have analyzed composite performance and availability in wireless communication networks. In [23], after reviewing the concepts of composite performance and availability analysis, three techniques were presented for this reason using queuing systems. In the model proposed in [23], some specific characteristics of wireless communication networks such as channel allocation models, and channel failures and repairs are taken into account and reward rates are calculated considering these factors. Reibman [17] has considered the effect of reliability on performance in degradable computing systems. In the proposed method in [17], the structural model of the system representing the failure-repair behavior of system is constructed in the first step. In the second step, reward rates are computed using corresponding queuing networks. Afterwards, the reward rates are assigned to the related states in previously constructed structural model. In [17], the probability of job completion has been considered as the reward rate in each of the states. Beaudry [24] has developed performance-related reliability measures which reflect the interaction between the reliability and performance features of computing systems. The proposed measures in [24] are as follows: the computation reliability, mean computation before failure, computation availability, and computation and capacity thresholds which show the time at which the computation reliability and availability reach specific values, respectively. These measures can be used in standby redundant systems, gracefully degrading systems and distributed systems. After introducing the measures, three examples were provided to show the applicability of the new measures to real systems in [24].

Some other related models can be also found in the literature. A good survey on resource allocation and task assignment in distributed systems has been done in [25]. In general, each of the methods presented in this area has its own pros and cons. There are some problems with the previously proposed methods in grid context. One of the problems is that only a few of them consider both local and grid tasks, and simultaneous execution of them inside a grid resource. Some papers that consider both types of tasks only compute pure performance of the grid environment paying no attention to the failure-repair behavior of the resources. Moreover, previous methods mostly focus on simulating a hypothesis grid using a simulator ignoring the mathematical model which can be used to model and analyze a grid resource precisely. We have tried to address the aforementioned difficulties existing in some previous research papers in this paper.

3 Background information

In this section, formal definition of MRMs is presented and some useful metrics for performability analysis are introduced. For more information on this subject and related proofs, please see [16,17,20–22,26].

Let $\{Z(t), t \geq 0\}$ represent a homogenous finite state CTMC with state space Ω. Let N be the number of states existing in $Z(t)$. Then, the generator matrix of $Z(t)$ can be written as an $N \times N$ matrix, $Q = \left[q_{ij}\right]$, in which each element q_{ij} represents the transition rate from state i to state j. Based on the definition of generator matrix in CTMCs, the diagonal elements of Q, q_{ii}, can be written as $-q_i$ which is equal to $-\sum_{i \neq j} q_{ij}$. Let $P_i(t)$ denote the probability of being in state i at time t, then the state probability vector of $Z(t)$ can be represented as $P(t)$. Now, the transient behavior of $Z(t)$ can be described by Eq. 1.

$$\frac{\mathrm{d}}{\mathrm{d}t}(P(t)) = P(t)Q, \quad P(0) = p_0 \tag{1}$$

where p_0 is the initial probability vector of $Z(t)$. Moreover, the steady state probability vector of $Z(t)$, represented by π, can be obtained by substituting $\frac{\mathrm{d}}{\mathrm{d}t}(P(t)) = 0$ in Eq. 1. Therefore, steady state probability vector π can be computed using Eq. 2.

$$\pi Q = 0, \quad \sum_{i \in \Omega} \pi_i = 1 \tag{2}$$

where π_i is the steady state probability of being in state i of the CTMC $Z(t)$.

As described above, the transient and steady state probabilities of being in state i of $Z(t)$ can be computed using Eqs. 1 and 2, respectively. Nevertheless, in some cases, it is necessary to compute the cumulative state probability of $Z(t)$. Let $L_i(t)$ denote the expected total time spent by CTMC $Z(t)$ in state i during the time interval $[0, t)$, then the cumulative state probability vector of $Z(t)$ can be computed by Eq. 3.

$$L(t) = \int_0^t P(\tau)\ \mathrm{d}\tau \tag{3}$$

Assigning reward rate to each of the states of $Z(t)$, a MRM can be constructed. Let r denote the reward rate vector over the states of $Z(t)$ such that the reward rate r_i is associated with the state i. Let $X(t) = r_{Z(t)}$ be the instantaneous reward rate of the obtained MRM. Then, the accumulative reward over a period $[0, t)$ is given by:

$$\Phi(t) = \int_0^t X(\tau)\ \mathrm{d}\tau = \int_0^t r_{Z(\tau)}\ \mathrm{d}\tau \tag{4}$$

The expected instantaneous reward rate and expected steady state reward can be calculated using Eqs. 5 and 6, respectively.

$$E[X(t)] = \sum_{i \in \Omega} r_i P_i(t) \tag{5}$$

$$E[X] = \sum_{i \in \Omega} r_i \pi_i \tag{6}$$

Moreover, one can compute the expected accumulated reward using Eq. 7.

$$E[\Phi(t)] = \sum_{i \in \Omega} r_i L_i(t) \tag{7}$$

Modeling a system using an appropriate CTMC and assigning suitable reward rates to the states of the obtained CTMC, one can compute the expected instantaneous and accumulated reward rates using Eqs. 5 and 7, respectively.

4 The proposed model

In this section, the basic structure of the combined performance and availability model is discussed. For this reason, in the first step, the availability model of a grid resource in the form of a CTMC is presented, and then, the performance evaluation is done to calculate the appropriate reward rates to be assigned to the related states of the obtained CTMC and capture the MRM. After that, performance and availability of the resource can be simultaneously analyzed.

To model the availability of a grid resource, various situations for the resource should be considered. When a resource becomes a member of a grid environment, it should execute grid tasks assigned by RMS. On the other hand, each of the grid resources belongs to its own virtual organization, and thereby it should execute local tasks assigned by the local manager (LM) [1,2,27,28]. Therefore, a grid resource can execute grid and local tasks simultaneously. In this case, the processing speed of the resource is shared between grid and local tasks. In addition to this case, a resource can fail or cancel its own membership at any time. In this situation, the resource is unavailable for grid users and it cannot execute any grid task. Considering aforementioned statements, three different cases can be distinguished in terms of grid resource availability from grid users' perspective.

1. In the first case, it is assumed that the resource is a member of grid environment, and there is no local load on the resource. In this case, all the entire processing power of the resource belongs to the grid environment, and RMS can schedule grid tasks to the resource to use the entire capability of the resource.
2. In the second case, local tasks were submitted to the corresponding virtual organization and LM has scheduled the tasks to the resource. In this case, the resource should answer local users in addition to the grid users. Therefore, the processing power of the resource is shared between the grid and local tasks. When a resource becomes a member of the grid environment, it is mainly concerned with grid tasks, but in some time intervals, several local tasks may be submitted to the resource. Although the numbers of local tasks submitted to the resource and arrival rate of them are typically less than grid tasks, a precise model should consider these tasks to present a more realistic image of the system.
3. In the third case, the resource is unavailable for grid users, and therefore, it cannot respond to the grid tasks. This can be caused by two different conditions. First, the resource fails to execute the tasks. Resource failure can occur in hardware, software or the virtual organization in which the resource exists. Second, the resource cancels its own membership and goes out of the grid environment. In both conditions, the resource cannot execute grid tasks, and therefore it is considered as an unavailable resource for grid users.

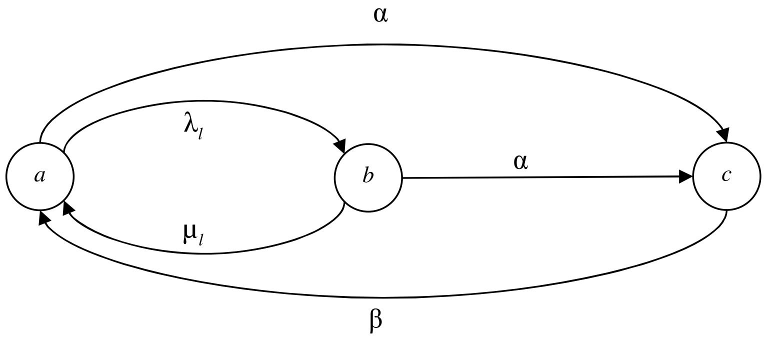

Fig. 1 CTMC showing the structure-state process of a grid resource

Considering the aforementioned three cases, the structure-state process of a grid resource can be depicted as Fig. 1. As can be seen in the CTMC shown in Fig. 1, states *a* and *b* are operational states of the resource in which the resource can execute grid and/or local tasks. Moreover, state *c* shows the non-operational state of the resource.

As shown in Fig. 1, the resource can fail during executing grid or local tasks. Therefore, it can transit from operational states *a* and *b* to the non-operational state *c* with the rate α, where α is resource failure rate which actually takes into consideration both actual resource failure and membership cancelation. Furthermore, there is a transition, with rate β, from non-operational state *c* to the operational state *a* to show the repair (membership) possibility of the failed (non-member) resource in the grid environment. Since the proposed model wants to study the availability of the resource from grid users' standpoint, there is no transition from state *c* to state *b*. In other words, it is assumed that when a failed resource is repaired (or a non-member resource joins the environment), it can receive grid tasks and service them without having any pre-submitted local task. However, local tasks can be submitted to the resource when it is available for grid users. This situation is shown in Fig. 1 using a transition from state *a* to state *b* labeled with λ_l. The label λ_l represents local tasks arrival rate to the resource and its value is lower than the value assigned to λ_g representing grid tasks arrival rate ($\lambda_l << \lambda_g$). After submitting local tasks, the resource services them with the rate μ_l which causes to move from state *b* to state *a* in the related CTMC. It should be mentioned that all the times assigned to transitions in the proposed CTMC follow exponential distribution function.

To compute the transient and steady state probability vectors of the CTMC shown in Fig. 1, generator matrix of the CTMC should be constructed. Equation 8 represents the generator matrix of the CTMC depicted in Fig. 1.

$$Q = \begin{bmatrix} -(\lambda_l + \alpha) & \lambda_l & \alpha \\ \mu_l & -(\mu_l + \alpha) & \alpha \\ \beta & 0 & -\beta \end{bmatrix} \tag{8}$$

Substituting Eq. 8 in 1 and setting the initial probability vector to $p_0 = (1, 0, 0)$, Eq. 9 can be obtained. Solving Eq. 9, the state probability vector of the CTMC, $P(t)$, can be computed.

$$\begin{bmatrix} \frac{\mathrm{d}P_1(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}P_2(t)}{\mathrm{d}t} \\ \frac{\mathrm{d}P_3(t)}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} \begin{bmatrix} -(\lambda_1+\alpha) & \lambda_1 & \alpha \\ \mu_1 & -(\mu_1+\alpha) & \alpha \\ \beta & 0 & -\beta \end{bmatrix}, \; P(0) = (1, 0, 0) \quad (9)$$

The steady state probability vector, π, can be also computed by replacing $\frac{\mathrm{d}}{\mathrm{d}t}(P(t)) = 0$ in Eq. 9, and taking into account the equation $\pi_1 + \pi_2 + \pi_3 = 1$.

Based on the traditional availability analysis, the instantaneous and steady state availability of the resource for grid users can be computed by Eqs. 10 and 11, respectively.

$$A(t) = P_1(t) + P_2(t) \quad (10)$$

$$A = \pi_1 + \pi_2 \quad (11)$$

Nevertheless, as mentioned earlier, our purpose is to evaluate combined performance and availability of a grid resource instead of computing only the availability measure. Therefore, performance levels of each of the states shown in Fig. 1 should be calculated and assigned to the corresponding states. To do this, the mean response times of the grid tasks (mean sojourn times of grid tasks in the resource) are computed as performance levels of the states. Other performance measures such as makespan, expected execution and/or waiting times of tasks, number of waiting tasks, overall mean response time and so forth can be also considered as performance levels in grid environments [5,7,10,12–15]. To evaluate the mean sojourn time of a grid task in a resource which is actually the summation of mean waiting time and mean service time of the task in that resource, queuing systems related to all three states of the obtained CTMC should be studied. To achieve this, the following situations for each of the states are considered.

1. In *state a* shown in Fig. 1, there is no local load on the resource, so the resource can service grid tasks with its entire processing capability. Let λ_g and μ denote the grid tasks arrival rate and total service rate of the resource, respectively. If both inter-arrival time of the grid tasks and service time of the resource follow the exponential distribution, the resource can be considered as M/M/1 queuing system. It should be mentioned that considering infinite queue size for grid resources to relax the problem into one that can be solved by conventional queuing systems, and considering exponential distribution for tasks' inter-arrival and service times are acceptable assumptions in grid environments which can be found in many research papers such as [3–8,14,15,18] and their references.
   Assuming $\lambda_g < \mu$ for the obtained M/M/1 queuing system to be stable and applying steady state analysis to compute the mean sojourn time of grid tasks in the resource, Eq. 12 can be written.

$$W_{g_1} = W_{q_1} + W_s \quad (12)$$

   where W_{q_1} and W_s denote the mean steady state waiting and service times of grid task t_g assigned to the resource R, respectively. Based on the basic formulas of queuing systems, the value of W_{g_1} for M/M/1 queues can be calculated as Eq. 13.

$$W_{g_1} = \frac{1}{\mu - \lambda_g} \tag{13}$$

For the sake of brevity, the related proofs are not presented here, but for more information about them, one can see [29,30]. After finding the mean steady state sojourn time of grid tasks in this situation, it can be considered as performance level or reward rate of state *a*.

2. In state *b*, both grid and local tasks have been submitted to the resource and the resource should service all of the submitted tasks. Therefore, the mean sojourn time of the grid and local tasks should be computed separately. To do this, queuing systems with different classes [30] should be considered. Since the tasks submitted to the resource belong to only two types, a queuing system with two different classes can be used to model this situation. On the other hand, the scheduling discipline of the tasks in each of the classes is a very important factor in estimating the mean sojourn time of tasks. Therefore, in this paper, three different scheduling disciplines, FIFO, non-preemptive and preemptive priorities, are considered. In FIFO discipline, there is no priority between grid and local tasks, and the resource executes tasks considering their arrival times. Hence, a task with early arrival time is processed earlier than a task with late arrival time.
Let W_{g_2} denote the mean steady state sojourn time of grid tasks while local tasks have been submitted to the resource and the scheduling discipline is FIFO. Eq. 14 shows how W_{g_2} can be computed.

$$W_{g_2} = W_{q_2} + E\left[s_g\right], \tag{14}$$

where W_{q_2} and $E\left[s_g\right]$ denote the mean steady state waiting time and the mean service time of grid task t_g assigned to the resource R, respectively. Considering the service rate of the resource to the grid tasks, $E\left[s_g\right]$ can be replaced by $1/\mu_g$. Moreover, W_{q_2} can be computed as Eq. 15.

$$W_{q_2} = \frac{(\lambda_g + \lambda_l)}{(\mu_g + \mu_l)((\mu_g + \mu_l) - (\lambda_g + \lambda_l))} \tag{15}$$

To simulate more realistic situations, priority discipline is considered to be applied to simultaneously schedule local and grid tasks. For this reason, an acceptable approach for priority assignment in grids is used in which the higher priority is assigned to the local tasks against the grid tasks [6,8,10,11,28]. Therefore, a grid task is executed on a resource only if there is no local load on the resource. Furthermore, two different disciplines can be considered for assigning higher priority to the local tasks against the grid ones: *non-preemptive* and *preemptive* disciplines. In non-preemptive priority discipline, an arriving local task can be executed only when the running grid task finishes its execution and leaves the resource. Based on queuing systems with non-preemptive scheduling discipline to capture aforementioned characteristics for local and grid tasks, the mean steady state sojourn time of grid tasks, W_{g_3}, can be calculated using Eq. 16.

$$W_{g_3} = W_{q_3} + E\left[s_g\right] = \frac{\left(\frac{\lambda_l}{\mu_l^2} + \frac{\lambda_g}{\mu_g^2}\right)}{\left(1 - \frac{\lambda_l}{\mu_l}\right)\left(1 - \frac{\lambda_l}{\mu_l} - \frac{\lambda_g}{\mu_g}\right)} + \frac{1}{\mu_g} \tag{16}$$

Against the non-preemptive discipline, in preemptive priority discipline, when a local task arrives to the resource, the running grid task is preempted (or blocks itself), and the resource is given to the local task. Whenever the local task finished running, the preempted grid task is scheduled to continue from the point where it stopped. Let W_{g_4} denote the mean steady state sojourn time of grid tasks while preemptive priority discipline is applied. Equation 17 shows how W_{g_4} can be computed.

$$W_{g_4} = \frac{1}{\left(1 - \frac{\lambda_l}{\mu_l}\right)}\left(\frac{1}{\mu_g} + \frac{\left(\frac{\lambda_l}{\mu_l^2} + \frac{\lambda_g}{\mu_g^2}\right)}{\left(1 - \frac{\lambda_l}{\mu_l} - \frac{\lambda_g}{\mu_g}\right)}\right) \tag{17}$$

Considering aforementioned explanations, values of W_{g_2}, W_{g_3} and W_{g_4} can be considered as the performance levels or reward rates of state b, where the FIFO, non-preemptive priority and preemptive priority disciplines are applied to service grid and local tasks, respectively.

It should be emphasized that related proofs and justifications of the formulas corresponding to the queuing systems with three different scheduling disciplines mentioned above can be found in the related reference such as [29] and [30].

3. As mentioned before, state c represents a situation in which the resource cannot respond to the grid tasks. This situation can be caused by failing the resource to service the tasks or by canceling membership from the grid environment. Anyway, the reward rate assigned to state c, W_{g_5}, is zero.

   It should be mentioned that the tasks considered above are independent and have no connections to each other. Therefore, executing a grid (local) task can be started inside a resource just after receiving its required data. Assigning the obtained performance levels (reward rates) to corresponding states of the CTMC shown in Fig. 1, the related MRM can be achieved. Although the mean steady state sojourn time of grid tasks can be computed using Eqs. 6 and 13–17, generally in performability analysis, we need to compute the cumulative reward by summing the rewards over interval $[0, t)$. To do this, the cumulative state probability vector, $L(t)$, should be computed using Eq. 3 (note that the state probability vector of the CTMC, $P(t)$, has been calculated by solving Eq. 9). Knowing the reward rate vector r, and the cumulative state probability vector $L(t)$, the expected accumulated reward can be computed using Eqs. 18–20.

$$E\left[\Phi_1(t)\right] = W_{g_1} L_1(t) + W_{g_2} L_2(t) \tag{18}$$

$$E\left[\Phi_2(t)\right] = W_{g_1} L_1(t) + W_{g_3} L_2(t) \tag{19}$$

$$E\left[\Phi_3(t)\right] = W_{g_1} L_1(t) + W_{g_4} L_2(t) \tag{20}$$

where $E[\Phi_1(t)]$, $E[\Phi_2(t)]$ and $E[\Phi_3(t)]$ denote the expected accumulative sojourn time of grid tasks when the scheduling disciplines are FIFO, non-preemptive priority and preemptive priority, respectively.

5 Detailed example

To facilitate the explanation of the proposed model, an example of system performance and availability analysis is presented in this section. Since the proposed model is a mathematical model, the accuracy of the model can be easily checked by following the formulas mentioned in previous sections. The example provided in this section is only to show the applicability of the proposed method to a sample grid resource. All of the numbers (e.g., failure, repair, task arrival and resource processing rates) are selected randomly, and they can be replaced with any value satisfying the mentioned constraints. Characteristics of the sample grid resource are given in Table 1. The units of all the rates are tasks per second.

As mentioned in Table 1, resource service rates for grid and local tasks are $\mu_g = 2.1$ and $\mu_l = 2.5$, respectively. Nevertheless, when there is no local load on the resource (state a), all the processing power of resource is gathered to service grid tasks; therefore, the service rate of the resource becomes $\mu = \mu_g + \mu_l = 4.6$. Replacing the parameters existing in Eq. 9 with the values mentioned in Table 1, the following formula is obtained.

$$\begin{bmatrix} \frac{dP_1(t)}{dt} \\ \frac{dP_2(t)}{dt} \\ \frac{dP_3(t)}{dt} \end{bmatrix} = \begin{bmatrix} P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix} \begin{bmatrix} -0.34 & 0.3 & 0.04 \\ 2.5 & -2.54 & 0.04 \\ 0.09 & 0 & -0.09 \end{bmatrix}, \quad P(0) = (1, 0, 0) \tag{21}$$

Solving Eq. 21, the transient state probability vector of the CTMC is computed as Eq. 22.

$$\begin{cases} P_1(t) = 0.62 + 0.27e^{-0.13t} + 0.11e^{-2.84t} \\ P_2(t) = 0.08 + 0.03e^{-0.13t} - 0.11e^{-2.84t} \\ P_3(t) = 0.31 - 0.31e^{-0.13t} \end{cases} \tag{22}$$

After finding the transient state probability vector of the CTMC, one can find the steady state probability vector, π, as follows.

Table 1 A sample resource characteristics

Parameter	Value
Resource failure rate (α)	0.04
Resource repair rate (β)	0.09
Grid tasks arrival rate (λ_g)	1.6
Local tasks arrival rate (λ_l)	0.3
Resource service rate for grid tasks (μ_g)	2.1
Resource service rate for local tasks (μ_l)	2.5

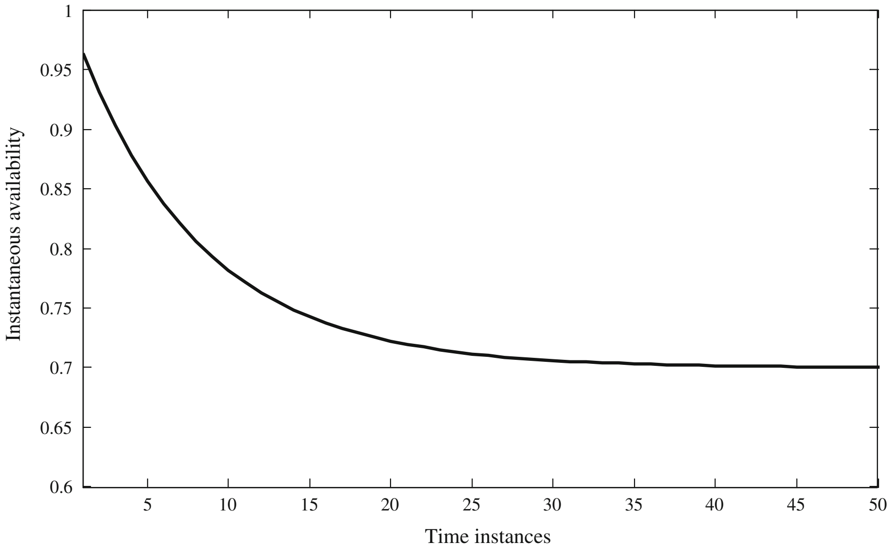

Fig. 2 Instantaneous availability of the sample grid resource

$$\begin{cases} \pi_1 = 0.62 \\ \pi_2 = 0.08 \\ \pi_3 = 0.31 \end{cases} \tag{23}$$

Considering Eqs. 10 and 11, the instantaneous and steady state availability of the sample grid resource can be computed as Eqs. 24 and 25, respectively.

$$A(t) = 0.7 + 0.3\mathrm{e}^{-0.13t} \tag{24}$$

$$A = 0.7 \tag{25}$$

Figure 2 shows the instantaneous availability of the resource in various time instances starting from 1 to 50. As displayed in Fig. 2, the availability of the resource decreases when the time increases. In addition, it can be seen that the instantaneous availability converges to the steady state availability, $A = 0.7$, when the time instances increase.

To compute performability of the sample resource, cumulative state probability vector, $L(t)$, should be calculated as Eq. 26.

$$\begin{cases} L_1(t) = 2.14 + 0.62t - 2.1\mathrm{e}^{-0.13t} - 0.04\mathrm{e}^{-2.84t} \\ L_2(t) = 0.22 + 0.08t - 0.26\mathrm{e}^{-0.13t} + 0.04\mathrm{e}^{-2.84t} \\ L_3(t) = 2.37 + 0.31t + 2.37\mathrm{e}^{-0.13t} \end{cases} \tag{26}$$

Using Eqs. 13–17, the reward rates, W_{g_1}, W_{g_2}, W_{g_3} and W_{g_4}, can be computed as Eq. 27.

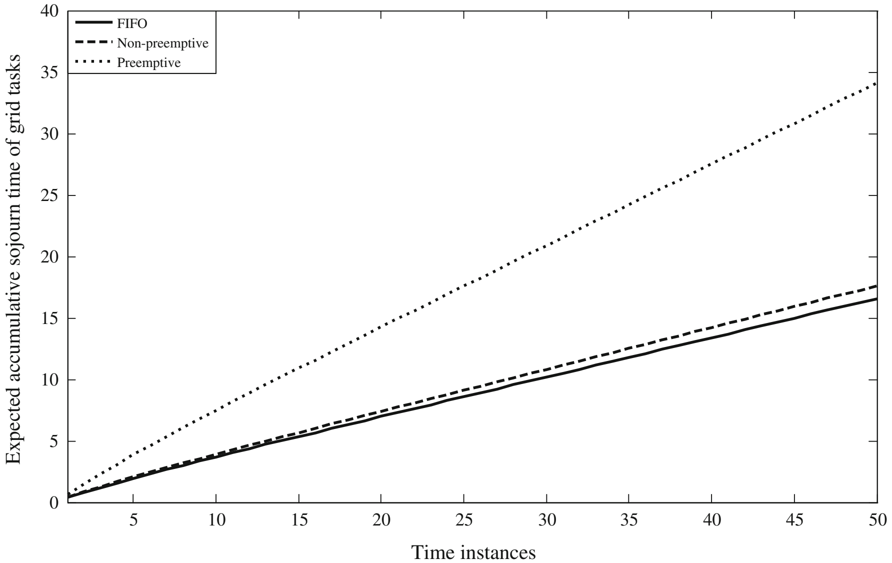

Fig. 3 The expected accumulated sojourn time of grid tasks in the sample resource for various time instances

$$\begin{cases} W_{g_1} = 0.33 \\ W_{g_2} = 0.63 \\ W_{g_3} = 0.90 \\ W_{g_4} = 4.49 \end{cases} \tag{27}$$

Therefore, using Eqs. 18–20, 26 and 27, the expected accumulated rewards can be calculated as Eq. 28 in which $E\,[\Phi_1(\mathrm{t})]$, $E\,[\Phi_2(\mathrm{t})]$ and $E\,[\Phi_3(\mathrm{t})]$ denote the expected accumulative sojourn time of grid tasks where the scheduling disciplines are FIFO, non-preemptive priority and preemptive priority, respectively.

$$\begin{cases} E\,[\Phi_1(t)] = 0.85 + 0.26t - 0.85\mathrm{e}^{-0.13t} + 0.02\mathrm{e}^{-2.84t} \\ E\,[\Phi_2(t)] = 0.91 + 0.28t - 0.92\mathrm{e}^{-0.13t} + 0.03\mathrm{e}^{-2.84t} \\ E\,[\Phi_3(t)] = 1.70 + 0.57t - 1.86\mathrm{e}^{-0.13t} + 0.17\mathrm{e}^{-2.84t} \end{cases} \tag{28}$$

Figure 3 shows the expected accumulated sojourn time of grid tasks in the sample resource for various time instances from 1 to 50 seconds. As shown in Fig. 3, FIFO scheduling discipline shows the minimum expected accumulated sojourn time for grid tasks compared to the non-preemptive priority and preemptive priority disciplines. This shows that the best scheduling discipline in view of grid users and among three scheduling disciplines studied in this paper is FIFO, because when this discipline is applied, grid tasks can receive faster service compared to the local tasks.

In another experiment, we consider the reverse of mean steady state sojourn times of grid tasks as reward rates of the states. In this case, the expected accumulated completion rate of grid tasks can be computed instead of expected accumulated sojourn times. Equation 29 shows the expected accumulated completion rate of the grid tasks

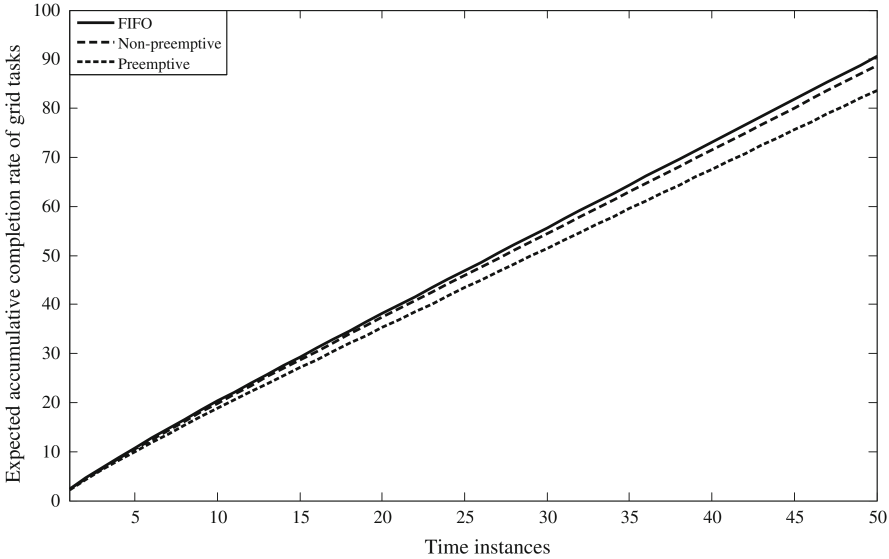

Fig. 4 The expected accumulated completion rate of grid tasks in the sample resource for various time instances

in which $E\left[\Phi_1'(\mathrm{t})\right]$, $E\left[\Phi_2'(\mathrm{t})\right]$ and $E\left[\Phi_3'(\mathrm{t})\right]$ denote this metric when the scheduling disciplines are FIFO, non-preemptive priority and preemptive priority, respectively.

$$\begin{cases} E\left[\Phi_1'(t)\right] = \frac{L_1(t)}{W_{g1}} + \frac{L_2(t)}{W_{g2}} = 6.77 + 1.99t - 6.71\mathrm{e}^{-0.13t} - 0.06\mathrm{e}^{-2.84t} \\ E\left[\Phi_2'(t)\right] = \frac{L_1(t)}{W_{g1}} + \frac{L_2(t)}{W_{g3}} = 6.66 + 1.95t - 6.59\mathrm{e}^{-0.13t} - 0.08\mathrm{e}^{-2.84t} \\ E\left[\Phi_3'(t)\right] = \frac{L_1(t)}{W_{g1}} + \frac{L_2(t)}{W_{g4}} = 6.47 + 1.88t - 6.36\mathrm{e}^{-0.13t} - 0.11\mathrm{e}^{-2.84t} \end{cases} \quad (29)$$

Figure 4 shows the expected accumulated completion rate of grid tasks in the sample resource when FIFO, non-preemptive priority and preemptive priority disciplines are applied. Actually, Fig. 4 emphasizes the results shown in Fig. 3.

6 Conclusions and future work

Availability of a grid resource to the grid users is one of the measures which can influence user perception of grid effectiveness. This measure can be affected by some system and environmental faults causing failure of a resource. In addition to the system and environmental faults, considering the specific characteristics of grid environments, the resource availability to grid users can also be influenced by arriving local tasks to the resource and membership refusal or cancelation by the resource. On the other hand, a grid resource shows various service rates and performance levels to execute the grid tasks based on its availability situation and local scheduling discipline. Consequently, isolated performance evaluation of a grid resource may cause undependable results; therefore, the performance and availability of a grid resource should be evaluated simultaneously. To do this, a MRM is presented in this paper to appropriately model and analyze the composite performance and availability. The proposed model uses CTMCs and queuing systems to evaluate the expected accumulative sojourn times of grid tasks in a grid resource. Moreover, the impact of the different local scheduling disciplines on the performability of a grid resource can be studied using the proposed model.

There are numbers of research issues remaining open for future work. The following are some ideas which can be used for further research in this area:

- Applying some other scheduling disciplines such as EDF discipline to select critical and impatient tasks from the list of the submitted tasks to be rapidly executed on the grid resource can be considered as one of the open research issues in grid performability analysis.
- Using distribution function of the rewards as performance levels of the CTMC states instead of mean steady state values may lead to invaluable results. One can find distribution function of sojourn time of grid tasks and use this function as a reward rate of the states to calculate the time-dependent measures.
- Modeling the proposed CTMC and MRM using GSPNs and SANs which helps to collect grid resources together and estimate the overall performability of the grid environment is an interesting future work in this research field.
- Computing other rewards or performance levels such as mean waiting time of grid tasks and mean number of waiting tasks can be considered as another future work in this area.

Acknowledgments The authors would like to thank Iran Telecommunication Research Center (ITRC) for their support. This research was also supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the CPRC (Communications Policy Research Center) support program supervised by the KCA (Korea Communications Agency)(KCA-1194100004).

References

1. Foster I, Kesselman C (2004) The Grid 2: blueprint for a new computing infrastructure, 2nd edn. Morgan Kaufmann, San Francisco
2. Krauter K, Buyya R, Maheswaran M (2002) A taxonomy and survey of grid resource management systems for distributed computing. Softw Pract experience 32(2):135–164
3. Levitin G, Dai Y-S (2007) Service reliability and performance in grid system with star topology. Reliab Eng System Saf 92(1):40–46
4. Dai Y-S, Levitin G (2006) Reliability and performance of tree-structured grid services. IEEE Trans Reliab 55(2):337–349
5. Azgomi MA, Entezari-Maleki R (2010) Task scheduling modelling and reliability evaluation of grid services using coloured Petri nets. Futur Gener Comput Systems 26(8):1141–1150
6. Entezari-Maleki R, Movaghar A (2011) Availability modeling of grid computing environments using SANs. In: The 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2011), Split, Croatia, September 15–17, 2011, pp 1–6
7. Caron E, Garonne V, Tsaregorodtsev A (2007) Definition, modeling and simulation of a grid computing scheduling system for high throughput computing. Futur Gener Comput Systems 23(8):968–976
8. Parsa S, Entezari-Maleki R (2009) Modeling and throughput analysis of grid task scheduling using stochastic Petri nets. In: The 2009 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'09), Las Vegas, July 13–16, 2009, pp 458–464

9. Berten V, Devillers R, Louchard G (2007) FIFO queuing of constant length fully synchronous jobs. In: The 4th International Conference on Grid Service Engineering and Management (GSEM '07), Leipzig, September 25–26, 2007, pp 81–95
10. Jayasinghe M, Tari Z, Zeephongsekul P, Zomaya AY (2011) Task assignment in multiple server farms using preemptive migration and flow control. J Parallel Distributed Comput 71(12):1608–1621
11. Zhang L, Zhang B, Zhang Y, Zeng L (2011) Efficient scheduling strategies in high performance computing service platform for Shanghai Colleges. In: The 3rd International Conference on Computer Research and Development (ICCRD'11), Shanghai, March 11–13, 2011, pp 176–180
12. Entezari-Maleki R, Movaghar A (2012) A probabilistic task scheduling method for grid environments. Futur Gener Comput Systems 28(3):513–524
13. Adabi S, Movaghar A, Rahmani AM, Beigy H (2013) Market_based grid resource allocation using new negotiation model. J Netw Comput Appl 36(1):543–565
14. Parsa S, Entezari-Maleki R (2012) Task dispatching approach to reduce the number of waiting tasks in grid environments. J Supercomput 59(1):469–485
15. Parsa S, Entezari-Maleki R (2012) A queuing network model for minimizing the total makespan of computational grids. Comput Electr Eng 38(4):827–839
16. Trivedi KS, Muppala JK, Woolet SP, Haverkort BR (1992) Composite performance and dependability analysis. Perform Eval 14(2–3):197–215
17. Reibman AL (1990) Modeling the effect of reliability on performance. IEEE Trans Reliab 39(3):314–320
18. Dai Y-S, Xie M, Poh K-L (2008) Availability modeling and cost optimization for the grid resource management system. IEEE Trans Systems Man Cybern Part A Systems Hum 38(1):170–179
19. Parsa S, Parand F-A (2012) Estimation of service reliability and performance in grid environment. J King Saud Univ Eng Sci 24(2):151–157
20. Trivedi KS, Ciardo G, Malhotra M, Sahner RA (1993) Dependability and performability analysis. NASA Contractor Report 191565, Report no. 93–85, November 1993
21. Trivedi KS, Malhotra M (1993) Reliability and performability techniques and tools: a survey. In: The 7th ITG/GI Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB), Germany, September 1993, pp 27–48
22. Meyer JF (1980) On evaluating the performability of degradable computing systems. IEEE Trans Comput c–29(8):720–731
23. Ma Y, Han JJ, Trivedi KS (2001) Composite performance and availability analysis of wireless communication networks. IEEE Trans Vehicular Technol 50(5):1216–1223
24. Beaudry MD (1978) Performance-related reliability measures for computing systems. IEEE Transact Comput C–27(6):540–547
25. Hussain H et al (2013) A survey on resource allocation in high performance distributed computing systems. Parallel Comput 39(11):709–736
26. Meyer JF (1982) Closed-form solutions of performability. IEEE Transact Comput C–31(7):648–657
27. Balasangameshwaraa J, Rajub N (2012) A hybrid policy for fault tolerant load balancing in grid computing environments. J Netw Comput Appl 35(1):412–422
28. Sun X-H, Wu M (2007) Quality of service of grid computing: resource sharing. In: The 6th International Conference on Grid and Cooperative Computing (GCC'07), Los Alamitos, August 16–18, 2007, pp 395–402
29. Bolch G, Greiner S, Meer HD, Trivedi KS (2006) Queueing networks and Markov chains: modeling and performance evaluation with computer science applications, 2nd edn., Wiley, New York
30. Allen AO (1990) Probability, statistics, and queueing theory with computer science applications, 2nd edn., Harcourt Brace Jovanovich, New York