Available online at www.sciencedirect.com
ScienceDirect
Reliability Engineering and System Safety 92 (2007) 40–46
RELIABILITY ENGINEERING & SYSTEM SAFETY
www.elsevier.com/locate/ress

Service reliability and performance in grid system with star topology

Gregory Levitin[a,*], Yuan-Shun Dai[b]

[a] The Israel Electric Corporation Ltd., Reliability Department, P.O. Box 10, Haifa 31000, Israel
[b] Department of Computer & Information Science, Purdue University School of Science, Indiana University, Purdue University, Indianapolis, 46202, USA

Received 23 March 2005; received in revised form 21 September 2005; accepted 3 November 2005
Available online 28 December 2005

Abstract

The paper considers grid computing systems in which the resource management systems (RMS) can divide service tasks into subtasks and send the subtasks to different resources for parallel execution. In order to provide desired level of service reliability the RMS can assign the same subtasks to several independent resources for parallel execution.

The service reliability and performance indices are introduced and a fast numerical algorithm for their evaluation for arbitrary subtask distribution in grid with star architecture is presented. This algorithm is based on the universal generating function technique.

Illustrative examples are presented.

Keywords: Grid system; Service time; Service reliability; Subtask distribution; Universal generating function

1. Introduction

Grid computing [1] is a newly developed technology for complex systems with large-scale resource sharing, wide-area communication, and multi-institutional collaboration etc., see e.g. [2–6]. Many experts believe that the grid technologies will offer a second chance to fulfill the promises of the Internet.

The real and specific problem that underlies the grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [4]. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources. This is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is controlled by the resource management system (RMS), see e.g. [7,8], with resource providers and consumers defining what is shared, who is allowed to share, and the conditions under which the sharing occurs.

Recently appeared Open Grid Services Architecture [5] enables the integration of services and resources across distributed, heterogeneous, dynamic virtual organizations, and also provides users a platform to easily request grid services. A grid service is desired to execute a certain task under the control of the RMS. When the RMS receives a service request from a user, the task can be divided into a set of subtasks that are executed in parallel. The RMS assigns those subtasks to available resources for execution. After the resources finish the assigned jobs, they return the results back to the RMS and then the RMS integrates the received results into entire task output which is requested by the user.

The above grid service process can be approximated by a structure with star topology, as depicted by Fig. 1, where the center is the RMS directly connected with the resources through respective communication channels.

The performance of grid computing is of great concern [9]. Usually the measure of grid performance is the task execution time (service time). This index can be significantly improved by using the RMS that divides a task into a set of subtasks which can be executed in parallel by multiple online resources. Many complicated and time-consuming tasks that could not be implemented before are currently working well under the grid computing environment, for instance, the

*Corresponding author.
E-mail address: levitin@iec.co.il (G. Levitin).

0951-8320/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2005.11.005

Nomenclature

RMS	resource management system
pmf	probability mass function
u-function	moment generating function
$1(x)$	unity function: $1(\text{TRUE}) = 1$, $1(\text{FALSE}) = 0$
$Pr(e)$	probability of event e
a_j	amount of data transmitted between the RMS and the resource processing subtask j
C	computational complexity of the entire task
c_j	computational complexity subtask j
m	number of subtasks
p_{kj}	probability that subtask j is correctly completed by resource k
q_{kj}	probability that communication between the RMS end resource k that processes subtask j is correctly completed
$R(\theta^*)$	probability that service time is less than θ^*
s_k	data transmission speed of communication channel k
T_{kj}	random time of subtask j processing by resource k
$u(z)$	u-functions representing distributions of discrete random variables
W	conditional expected system execution time
x_k	processing speed of resource k
λ_k	failure rate of resource k
π_k	failure rate of communication channel k
θ_{j,ω_j}	random time of subtask j completion by resources belonging to set ω_j
Θ	random time of task execution by the system (service time)
θ^*	maximum allowed service time
τ_{kj}	random communication time between RMS end resource k that processes subtask j
Ω	set of available resources
ω_j	set of resources processing subtask j

Human Proteome Folding Project (http://www.grid.org/projects/hpf/), earthquake simulation (http://it.nees.org/), and climate modeling (https://www.earthsystemgrid.org/), etc.

However, there is no model to analyze the grid service performance because the grid computing system by its nature is too large, complicated and dynamic to be tractable in modeling and analysis.

It is commonly accepted that the service time in the grid is a random variable [10,11]). Finding the distribution of this variable is important for evaluating the grid performance and improving the RMS functioning [12]. This paper presents a novel model and algorithm that can efficiently obtain the distribution of service time under the grid computing.

The service time is a random variable affected by many factors. First, there are many resources available online, that have different task processing speeds. Thus, the task execution time can vary depending on which resource is assigned to execute the task/subtasks. Second, some resources can fail when running the jobs, so the execution time is also affected by the resource reliability. Similarly, the communication links in grid service can fail during the data transmission. Thus, the communication reliability influences the service time as well as data transmission speed in the communication channels.

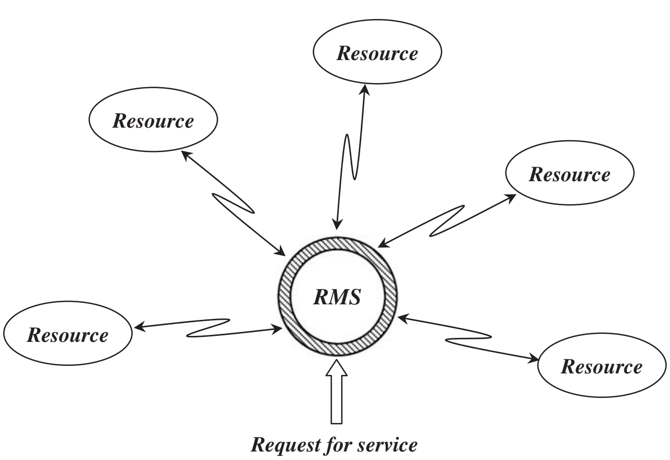

Fig. 1. Grid system with star architecture.

The above factors are considered in our grid performance model. Since some of these factors are based on the grid service reliability, it is necessary to study the reliability model as the foundation for the grid performance analysis. Some initial studies have been done in exploring the grid service reliability. Dai et al. [13] studied the service reliability for a wide-area distributed system that is one of the ancestors of the grid system. The function of control center in that model is similar to that of RMS for the grid computing. However, the reliability model of the sub-distributed systems inherits the approach used in the traditional models, see e.g. [14–17], etc. and has certain limitations. Those traditional models are based on a common assumption that the operational state probabilities of the nodes and links are constant. However, this assumption is unrealistic for the grid, so Dai et al. [18] relaxed this assumption by assuming that the failures of nodes and links follow Poisson processes so that their operational state probabilities are decreasing functions of their operation time. This paper adopts the same assumption to model the service reliability.

Most of the previous researchers separated performance and reliability into two different fields and studied them individually. However in fact, performance and reliability are closely related and affect each other, in particular when the grid computing is implemented. For example, when a task is fully parallelized into n subtasks executed by n

resources simultaneously, the performance is high but the reliability can be low because failure of any resource makes the entire task incomplete. This causes the task restart, which inversely increases its execution time (i.e. reduces its performance). Therefore, it is worth having some redundant resources to execute same subtask especially for those failure-prone resources. However, too many redundancies, even though improving the reliability, can decrease the performance by not fully parallelizing the task. Thus performance and reliability should be studied together in the grid service analysis.

Optimizing the division of service task into subtasks and distribution of these subtasks among available grid resources can considerably improve the service performance. Solving these optimization problems is impossible without developing a fast algorithm for evaluating the service performance and availability for arbitrary task division and subtask allocation solutions.

This paper presents the model for evaluating performance (service time) of grid with star topology taking the service reliability into account. According to the model, the paper obtains the random service time distribution and performance/reliability measures derived from this distribution (performability and expected execution time). Section 2 presents the grid service performance model as well as a reliability model. Section 3 describes an algorithm to efficiently obtain the pmf for service time by using universal generating function technique. Section 4 provides illustrative examples.

2. The model

2.1. Assumptions

1. Different resources are distributed in the grid system. The considered service uses a given set of resources.
2. The request for service (task execution) arrives to the RMS which can divide the task into subtasks and assign them to different resources for processing.
3. The total computational complexity of the entire task (number of operation) is equal to the sum of computational complexities of its subtasks.
4. Each resource is able to process any single subtask when it is available.
5. Each resource starts execution of the assigned subtask immediately after it gets the subtask data from the RMS through communication channels.
6. Each resource has a given constant processing speed when it is available.
7. Each resource has a given constant failure rate.
8. Each resource is directly connected to the RMS by single communication channel, which forms the star topology.
9. Each communication channel has constant failure rate.
10. Each communication channel has constant data transmission speed (bandwidth) when it is available.
11. The subtask processing time is proportional to its computational complexity.
12. The data transmission time is proportional to the amount of transmitted data.
13. If resource failure occurs during subtask processing or channel failure occurs during data transmission between the resource and the RMS, the subtask fails (cannot be completed).
14. The task is completed when all of the subtasks are completed and their results are returned to the RMS from the resources.
15. If subtask is processed by several resources, it is completed when first result is returned to the RMS.
16. The failures at different resources and communication channels are independent.
17. The RMS if fully reliable and the time of task processing by the RMS (division into subtasks, sending the subtasks to the resources, receiving the results and integrating them into entire task output) is negligible when compared with subtasks' processing time.

The above assumptions are general for the Grid service modeling. Most grid services can be represented by this model. For example, the Human Proteome Folding Project (http://www.grid.org/projects/hpf/) requires services that classify groups of proteomes into different categories. The RMS divides the task into small subtasks and each subtask is processed by a grid node/resource (PC) to classify some proteome based on a huge database by looking up the Gene Sequence dictionary. The classification service task succeeds when a set of resources/nodes covers the requested group of proteomes.

In this model, we consider the resource reliability as a factor that affects the service performance. Since the grid resources are highly heterogeneous, the order of magnitude for their reliability can vary in a wide range.

The assumption that the RMS does not fail during the grid service execution is reasonable because the RMS (based on multiple servers) usually has a very high reliability and service times are relatively short. In general, imperfect RMS can also be easily modeled as an element connected in series with the grid.

2.2. Service time distribution

According to the assumptions the entire task is divided into m subtasks such that

$$\sum_{j=1}^{m} c_j = C, \tag{1}$$

where C is the computational complexity of the entire task and c_j is the computational complexity of subtask j.

When subtask j is assigned to resource k the subtask processing time is a random variable that can take two possible values:

$$T_{kj} = \frac{c_j}{x_k} \tag{2}$$

if the resource does not fail during the subtask execution and $T_{kj}=\infty$ otherwise. For constant failure rate the probability that resource k does not fail during processing of subtask j can be obtained as

$$p_{kj}=e^{-\lambda_k(c_j/x_k)}. \tag{3}$$

These gives the distribution of the random subtask processing time: $Pr(T_{kj}=c_j/x_k)=p_{kj}$ and $Pr(T_{kj}=\infty)=1-p_{kj}$.

The amount of data a_j should be transmitted between the RMS and resource k that processes subtask j (input data from the RMS to the resource and output data from the resource to the RMS). Therefore the time of communication between RMS end resource k that processes subtask j can take two values:

$$\tau_{kj}=\frac{a_j}{s_k} \tag{4}$$

if communication channel k does not fail during the subtask execution and $\tau_{kj}=\infty$ otherwise. For constant failure rate the probability that communication channel k does not fail during processing of subtask j can be obtained as

$$q_{kj}=e^{-\pi_k(a_j/s_k)}. \tag{5}$$

This gives the distribution of the random subtask processing time: $Pr(\tau_{kj}=a_j/s_k)=q_{kj}$ and $Pr(\tau_{kj}=\infty)=1-q_{kj}$.

Since the subtask is completed when its output reaches the RMS, the random total completion time $\theta_{j,\{k\}}$ for subtask j assigned to resource k is equal to $T_{kj}+\tau_{kj}$. It can be easily seen that the distribution of this time is

$$Pr\left(\theta_{j,\{k\}}=\left(\frac{c_j}{x_k}+\frac{a_j}{s_k}\right)\right)=p_{kj}q_{kj}$$

and

$$Pr(\theta_{j,\{k\}}=\infty)=1-p_{kj}q_{kj}. \tag{6}$$

Assume that each subtask j is assigned to resources composing set ω_j such that $\cup_{j=1}^{m}\omega_j=\Omega$, $\omega_i\cap\omega_j=\varnothing$ for any $i\neq j$. In this case the random time of subtask j completion is

$$\theta_{j,\omega_j}=\min_{k\in\omega_j}(\theta_{j,\{k\}}). \tag{7}$$

The entire task is completed when all of the subtasks (including the slowest one) are completed. Therefore the random task execution time takes the form:

$$\Theta=\max_{1\leqslant j\leqslant m}\theta_{j,\omega_j}=\max_{1\leqslant j\leqslant m}[\min_{k\in\omega_j}(\theta_{j,\{k\}})]. \tag{8}$$

Having the distributions of random times T_{kj} and τ_{kj} one can obtain the distribution of the entire task execution time Θ (probability mass function of discrete variable Θ) in the form of pairs (Θ_i, Q_i), $0\leqslant i\leqslant I$ where Θ_i is ith realization of Θ, $Q_i=Pr(\Theta=\Theta_i)$ and I is total number of different realizations of Θ.

Since the task execution time (service time) Θ can take different values, the service should be considered as a multi-state system [19] with performance depending on combination of states of its elements (different combinations of available and failed resources and communicational channels correspond to different values of service time).

2.3. Service reliability and expected performance

In order to estimate both the service reliability and its performance, different measures can be used depending on the application. In applications where the execution time of each task (service time) is of critical importance, the system reliability $R(\theta^*)$ is defined (according to performability concept in [20,21]) as a probability that the correct output is produced in time less than θ^*. This index can be obtained as

$$R(\theta^*)=\sum_{i=1}^{I}Q_i\cdot 1(\Theta_i<\theta^*). \tag{9}$$

In applications where the average service performance (the number of executed tasks over a fixed time) is of interest [22], the service reliability is defined as the probability that it produces correct outputs without respect to the service time. This index can be referred to as $R(\infty)$. The conditional expected service time W is considered to be a measure of its performance.

This index determines the expected service time given that the service does not fail. It can be obtained as

$$W=\sum_{i=1}^{I}\Theta_iQ_i/R(\infty). \tag{10}$$

The following section presents an algorithm for determining the pmf of the service time Θ.

3. Algorithms for determining the pmf of the service time

The procedure used in this paper for the evaluation of service time distribution is based on the universal generating function (u-function) technique, which was introduced in [23] and which proved to be very effective for the reliability evaluation of different types of multi-state systems [24]. The main advantage of this technique is its high computational efficiency that allows it to be used in optimization procedures where large number of different solutions should be estimated [25].

The u-function representing the pmf of a discrete random variable Y is defined as a polynomial

$$u(z)=\sum_{k=1}^{K}\alpha_k z^{y_k}, \tag{11}$$

where the variable Y has K possible values and α_k is the probability that Y is equal to y_k.

To obtain the u-function representing the pmf of a function of two independent random variables $\varphi(Y_i, Y_j)$, composition operators are introduced. These operators determine the u-function for $\varphi(Y_i, Y_j)$ using simple algebraic operations on the individual u-functions of the variables. All of the composition operators take the form

$$U(z) = u_i(z) \underset{\varphi}{\otimes} u_j(z) = \sum_{k=1}^{K_i} \alpha_{ik} z^{y_{ik}} \underset{\varphi}{\otimes} \sum_{h=1}^{K_j} \alpha_{jh} z^{y_{jh}} = \sum_{k=1}^{K_i} \sum_{h=1}^{K_j} \alpha_{ik}\alpha_{jh} z^{\varphi(y_{ik},\, y_{jh})}. \quad (12)$$

The u-function $U(z)$ represents all of the possible mutually exclusive combinations of realizations of the variables by relating the probabilities of each combination to the value of function $\varphi(Y_i, Y_j)$ for this combination. For example for functions $\min(Y_i, Y_j)$ and $\max(Y_i, Y_j)$ operator (12) takes the form

$$u_i(z) \underset{\min}{\otimes} u_j(z) = \sum_{k=1}^{K_i} \sum_{h=1}^{K_j} \alpha_{ik}\alpha_{jh} z^{\min(y_{ik},\, y_{jh})} \quad (13)$$

and

$$u_i(z) \underset{\max}{\otimes} u_j(z) = \sum_{k=1}^{K_i} \sum_{h=1}^{K_j} \alpha_{ik}\alpha_{jh} z^{\max(y_{ik},\, y_{jh})}, \quad (14)$$

respectively.

In the case of grid system, the u-function $u_{j,\{k\}}(z)$ can define pmf of total completion time $\theta_{j,\{k\}}$ for subtask j assigned to resource k. According to (6) this u-function takes the form

$$u_{k,\{j\}}(z) = p_{kj}q_{kj} z^{(c_j/x_k + a_j/s_k)} + (1 - p_{kj}q_{kj}) z^{\infty}, \quad (15)$$

where p_{kj} and q_{kj} are determined according to Eqs. (3) and (5) respectively.

The total completion time of subtask j assigned to a pair of resources k and h is equal to the minimum of completion times for different resources according to (7). To obtain the u-function representing the pmf of this time, composition operator with $\varphi(Y_i, Y_j) = \min(Y_i, Y_j)$ should be used:

$$\begin{aligned} u_{j,\{k,h\}}(z) &= u_{j,\{k\}}(z) \underset{\min}{\otimes} u_{j,\{h\}}(z) \\ &= p_{kj}q_{kj}p_{hj}q_{hj} z^{\min\{((c_j/x_k)+(a_j/s_k)),((c_j/x_h)+(a_j/s_h))\}} \\ &\quad + (1 - p_{kj}q_{kj})p_{hj}q_{hj} z^{((c_j/x_h)+(a_j/s_h))} \\ &\quad + (1 - p_{hj}q_{hj})p_{kj}q_{kj} z^{((c_j/x_k+(a_j/s_k))} \\ &\quad + (1 - p_{kj}q_{kj})(1 - p_{hj}q_{hj}) z^{\infty}. \end{aligned} \quad (16)$$

The u-function representing the pmf of completion time θ_{j,ω_j} of subtask j assigned to all of the resources from set $\omega_j = \{k_1,\ldots,k_n\}$ can be obtained recursively:

$$\begin{aligned} &u_{j,\{k_1,k_2\}}(z) = u_{j,\{k_1\}}(z) \underset{\min}{\otimes} u_{j,\{k_2\}}(z), \\ &u_{j,\{k_1,k_2,k_3\}}(z) = u_{j,\{k_1,k_2\}}(z) \underset{\min}{\otimes} u_{j,\{k_3\}}(z), \\ &\ldots, \\ &u_{j,\omega_j}(z) = u_{j,\{k_1,\ldots,k_n\}}(z) = u_{j,\{k_1,\ldots,k_{n-1}\}}(z) \underset{\min}{\otimes} u_{j,\{k_n\}}(z). \end{aligned} \quad (17)$$

Having the u-functions $u_{j,\omega_j}(z)$ for each subtask j ($1 \leqslant j \leqslant m$) one can obtain the u-function representing the pmf of the entire task completion time Θ using composition operator $\otimes_{\max}$ according to Eq. (8).

$$\begin{aligned} &U_1(z) = u_{1,\omega_1}, \\ &U_2(z) = U_1(z) \underset{\max}{\otimes} u_{2,\omega_2}(z), \\ &\ldots, \\ &U_m(z) = U_{m-1}(z) \underset{\max}{\otimes} u_{m,\omega_m}(z). \end{aligned} \quad (18)$$

The final u-function $U_m(z)$ represents the pmf of random task completion time Θ in the form

$$U_m(z) = \sum_{i=1}^{I} Q_i z^{\Theta_i}. \quad (19)$$

The service reliability and performance can be obtained from this pmf using Eq. (9) and (10).

4. Illustrative examples

4.1. Analytical example

Consider a grid service that uses four resources. Assume that the RMS divides the service task into two subtasks. The first subtask is assigned to resources 1 and 2, the second subtask is assigned to resources 3 and 4. The reliability of the resources (including communication channels) and assigned subtask completion times for available resources are presented in Tables 1 and 2.

In order to determine the completion time distribution for both subtasks, define the u-functions $u_{j,\{k\}}(z)$:

$$u_{1,\{1\}}(z) = 0.7z^5 + 0.3z^{\infty}, \quad u_{1,\{2\}}(z) = 0.8z8 + 0.2z^{\infty}$$

for subtask 1 and

$$u_{2,\{3\}}(z) = 0.6z^{10} + 0.4z^{\infty}, \quad u_{2,\{4\}}(z) = 0.9z^6 + 0.1z^{\infty}.$$

The u-functions representing the pmf of completion times θ_{j,ω_j} are obtained as follows:

$$\begin{aligned} u_{1,\omega_1}(z) &= u_{1,\{1,2\}}(z) = u_{1,\{1\}}(z) \underset{\min}{\otimes} u_{1,\{2\}}(z) \\ &= (0.7z^5 + 0.3z^{\infty}) \underset{\min}{\otimes} (0.8z^8 + 0.2z^{\infty}) \\ &= 0.7z^5 + 0.24z^8 + 0.06z^{\infty}. \end{aligned}$$

Table 1
Reliability $p_{kj}\, q_{kj}$ of grid system resources

No of resource k	No of subtask j	
	1	2
1	0.7	—
2	0.8	—
3	—	0.6
4	—	0.9

Table 2
Subtask completion times $\theta_{j,\{k\}}$

No of resource k	No of subtask j	
	1	2
1	5	—
2	8	—
3	—	10
4	—	6

$$u_{2,\omega_2}(z) = u_{2,\{3,4\}}(z) = u_{2,\{3\}}(z) \underset{\min}{\otimes} u_{2,\{4\}}(z)$$
$$= (0.6z^{10} + 0.4z^{\infty}) \underset{\min}{\otimes} (0.9z^6 + 0.1z^{\infty})$$
$$= 0.9z^6 + 0.06z^{10} + 0.04z^{\infty}.$$

The u-function representing the pmf of entire task completion time Θ is obtained as follows:

$$U(z) = u_{1,\omega_1}(z) \underset{\max}{\otimes} u_{2,\omega_2}(z) = (0.7z^5 + 0.24z^8 + 0.06z^{\infty})$$
$$\underset{\max}{\otimes} (0.9z^6 + 0.06z^{10} + 0.04z^{\infty})$$
$$= 0.63z^6 + 0.216z^8 + 0.0564z^{10} + 0.0976z^{\infty}.$$

This u-function represents the pmf of Θ:

$$Pr(\Theta = 6) = 0.63, \quad Pr(\Theta = 8) = 0.216,$$
$$Pr(\Theta = 10) = 0.0564, \quad Pr(\Theta = \infty) = 0.0976.$$

From the obtained pmf we can calculate the service reliability using Eq. (9):

$$R(\theta^*) = 0.63 \text{ for } 6 < \theta^* \leqslant 8,$$
$$R(\theta^*) = 0.846 \text{ for } 8 < \theta^* \leqslant 10, \quad R(\infty) = 0.9024$$

and the conditional expected service time according to Eq. (10):

$$W = (0.63 * 6 + 0.216 * 8 + 0.0564 * 10)/0.9024 = 6.729.$$

4.2. Numerical example

Consider a grid service that uses six resources distributed in the grid system. The entire service task has complexity $C = 6000$ (mega operations). The parameters of the resources and communication channels are presented in Table 3. The transmitted data amount corresponding to each subtask is proportional to the computational complexity of this subtask: $a_j = 0.05c_j$.

Five different cases of subtask distributions have been analyzed.

In case A the task is not divided into subtasks and entire task is sent for parallel processing to each one of six resources. In this case $m = 1$ and $\omega_1 = \{1,2,3,4,5,6\}$.

In cases B and C the task is divided into two subtasks ($m = 2$) with equal complexity $c_1 = c_2 = 3000$. The subtask distribution is $\omega_1 = \{1,3,5\}$, $\omega_2 = \{2,4,6\}$ in case B and is $\omega_1 = \{1,2,3\}$, $\omega_2 = \{4,5,6\}$ in case C.

Table 3
Parameters of grid resources and communication channels

No of resource (channel) j	λ_j (s^{-1})	s_j (megaoperations/s)	π_j (s^{-1})	s_j (megabytes/s)
1	0.0006	10	0.002	3
2	0.0010	12	0.003	5
3	0.0010	15	0.002	3
4	0.0015	15	0.001	7
5	0.0010	20	0.003	4
6	0.0015	20	0.002	7

Table 4
Service performance indices for different subtask distributions

Subtask distribution	$\Theta_{\min}$	$\Theta_{\max}$	$R(\infty)$	W
A	342.86	700.00	0.9923	374.46
B	187.50	350.00	0.9670	214.03
C	250.00	350.00	0.9671	261.39
D	186.67	233.33	0.9072	194.80
E	147.62	233.33	0.9079	168.02

In cases D and E the task is divided into three subtasks ($m = 3$) with equal complexity $c_1 = c_2 = c_2 = c_3 = 2000$. The subtask distribution is $\omega_1 = \{1,2\}$, $\omega_2 = \{3,4\}$, $\omega_3 = \{5,6\}$ in case D and is $\omega_1 = \{1,4\}$, $\omega_2 = \{2,5\}$, $\omega_3 = \{3,6\}$ in case E.

Table 4 contains service performance indices obtained for different cases by numerical procedure based on the algorithm described in previous sections. For each case the minimal and maximal realizations of random service time Θ, the reliability that service succeeds (without respect to service time) $R(\infty)$ and the conditional expected service time W are presented. The service reliabilities as functions of maximal allowable service time $R(\theta^*)$ are presented in Fig. 2.

It can be observed that the division of the major task into subtasks executed in parallel by different resources reduces the service time by the price of reducing the service reliability. For the same number of subtasks the distribution of subtasks among different resources affects both the service reliability and its expected execution time.

5. Conclusions and further research

Grid technology is a newly developed method for large-scale distributed system. This technology allows effective distribution of computational tasks among different resources presented in the grid. The resource management system (RMS) can divide service task into subtasks and send the subtasks to different resources for parallel execution. In order to provide desired level of service reliability the RMS can assign the same subtasks to several independent resources.

In order to evaluate the quality of service its reliability and performance indices should be defined. This paper

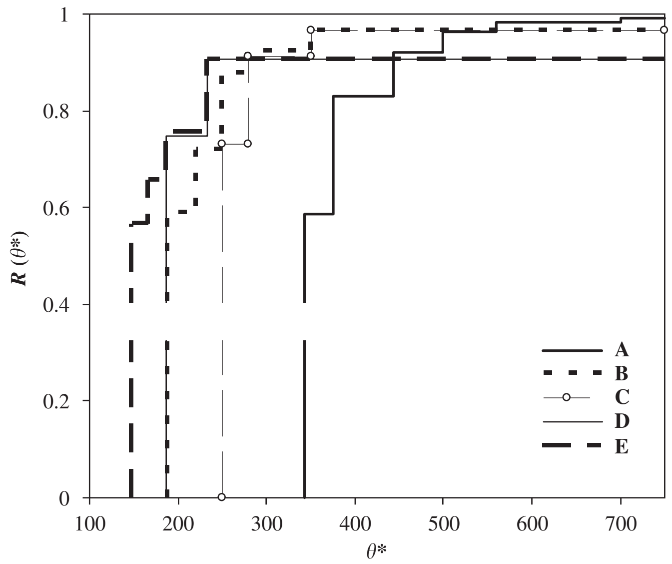

Fig. 2. System reliability functions $R(\theta^{*})$ for cases A–E.

introduces the indices: service reliability (probability that the service task is accomplished within a specified time) and conditional expected system time and presents the numerical algorithm for their evaluation for arbitrary subtask distribution in a given grid with star architecture.

It is shown that a trade-off exists between service reliability and its expected time. It is also shown that for any given number of subtasks the reliability and performance indices depend on subtask distribution among resources.

The algorithm can be used by the RMS for comparison of different resource management alternatives and making decisions aimed at service performance improvement. The high computational efficiency of the proposed algorithm allows it to be used in optimization procedures where large number of different solutions should be estimated.

The following directions of the further research based on the suggested generic model can be outlined:

- incorporating transient failures and precedence constraints on the subtask execution into the model;
- incorporating the variable processing and communication speeds and variable failure rates of the grid elements into the model;
- finding the service task division into subtasks and subtask allocation among different resources that provide the greatest service performance subject to reliability and budget constraints.

References

[1] Foster I, Kesselman C. The Grid 2: Blueprint for a New Computing Infrastructure. Los Alios: Morgan-Kaufmann; 2003.

[2] Kumar A. An efficient SuperGrid protocol for high availability and load balancing. IEEE Trans Comput 2000;49(10):1126–33.

[3] Das SK, Harvey DJ, Biswas R. Parallel processing of adaptive meshes with load balancing. IEEE Trans Parallel Distribut Syst 2001;12(12):1269–80.

[4] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations. Int J High Perform Comput Appl 2001;15:200–22.

[5] Foster I, Kesselman C, Nick JM, Tuecke S. Grid services for distributed system integration. Computer 2002;35(6):37–46.

[6] Berman F, Wolski R, Casanova H, Cirne W, Dail H, Faerman M, et al. Adaptive computing on the Grid using AppLeS. IEEE Trans Parallel Distribut Syst 2003;14(14):369–82.

[7] Krauter K, Buyya R, Maheswaran M. A taxonomy and survey of grid resource management systems for distributed computing. Software—Practice and Experience 2002;32(2):135–64.

[8] Nabrzyski J, Schopf JM, Weglarz J. Grid resource management. Dordrecht: Kluwer Publishing; 2003.

[9] Abramson D, Giddy J, Kotler L. High performance parametric modeling with Nimrod/G, In: 14th International parallel and distributed processing symposium, 2000. p. 520–8.

[10] Shan H, Oliker L, Biswas R. Job superscheduler architecture and performance in computational grid environments. In: Proceedings of the ACM/IEEE Supercomputing'03 Conference (SC'03), 2003. p. 44–59.

[11] England D, Weissman JB. A stochastic control model for the deployment of dynamic grid services. The 5th IEEE/ACM International Workshop on Grid Computing, 2004. p. 192–9.

[12] Weissman JB, Kim S, England D. Supporting the dynamic grid service lifecycle, IEEE/ACM CCGrid International Symposium on Cluster Computing and the Grid, 2005.

[13] Dai YS, Xie M, Poh KL, Liu GQ. A study of service reliability and availability for distributed systems. Reliab Eng Syst Safety 2003; 79(1):103–12.

[14] Kumar VKP, Hariri S, Raghavendra CS. Distributed program reliability analysis. IEEE Trans Software Eng 1986;SE-12:42–50.

[15] Chen DJ, Huang TH. Reliability analysis of distributed systems based on a fast reliability algorithm. IEEE Trans Parallel Distribut Syst 1992;3(2):139–54.

[16] Chen DJ, Chen RS, Huang TH. A heuristic approach to generating file spanning trees for reliability analysis of distributed computing systems. Comput Math Appl 1997;34:115–31.

[17] Lin MS, Chang MS, Chen DJ, Ku KL. The distributed program reliability analysis on ring-type topologies. Comput Oper Res 2001;28:625–35.

[18] Dai YS, Xie M, Poh KL. Reliability analysis of grid computing systems, IEEE Pacific rim international symposium on dependable computing (PRDC2002). IEEE Computer Press; 2002. p. 97–104.

[19] Lisnianski A, Levitin G. Multi-state system reliability. Singapore: World Scientific; 2003.

[20] Tai A, Meyer J, Avizienis A. Performability enhancement of fault-tolerant software. IEEE Trans Reliabil 1993;42(2):227–37.

[21] Meyer J. On evaluating the performability of degradable computing systems. IEEE Trans Comput 1980;29:720–31.

[22] Grassi V, Donatiello L, Iazeolla G. Performability evaluation of multicomponent fault tolerant systems. IEEE Trans Reliabil 1988;37(2):216–22.

[23] Ushakov I. Optimal standby problems and a universal generating function. Soviet J Comput Syst Sci 1987;25:79–82.

[24] Levitin G, Lisnianski A, Beh-Haim H, Elmakis D. Redundancy optimization for series-parallel multi-state systems. IEEE Trans Reliabil 1998;47:165–72.

[25] Levitin G. Universal generating function in reliability analysis and optimization. Berlin: Springer; 2005.