Towards Resource Reliability Support for Grid Workflows

Jiong Yu[1,2], Guozhong Tian[2], Yuanda Cao[1], Xianhe Sun[3]

School of Computer, Beijing Institute of Technology[1]
School of Information Science and Engineering, Xinjiang University[2]
Department of Computer Science, Illinois Institute of Technology[3]
{yujiong@xju.edu.cn}

Abstract. Grid workflow can be defined as an organization of Grid Services into a well-defined flow of operations, and can be thought of as the composition of Grid Services over time on heterogeneous and distributed resources in a well-defined order to accomplish a specific goal. To the time-constrained workflow Scheduling in Grids, we present a scheduling algorithm in terms of the finite-state continuous-time Markov process by selecting a resource combination scheme which has the lowest expenditure under the certain credit level of the resource reliability on the critical path in the DAG-based workflow. The simulation shows the validity of theory analysis.

Key words: Scheduling Algorithm; Resource Reliability; Grid Workflow

1 Introduction

In order to support complex scientific experiments, the distributed Grid resources need to be orchestrated while managing the application workflow operations within Grid environments. Correspondingly, workflow systems for Grid Services are evoking a high degree of interest. Workflow scheduling is one of the key issues in the workflow management[1]. A workflow tasks scheduling is a process that maps and manages execution of inter-dependent tasks on distributed resources. It allocates suitable resources to workflow tasks to satisfy objective functions imposed by users. Proper scheduling can have significant impact on the performance of workflow systems. Due to aggregation of geographically distributed autonomous, volatile and heterogeneous resources in Grid environments, the workflow on Grids differs far from the traditional workflow and various workflow Scheduling Algorithms are discussed from different points of view, such as static vs. dynamic policies, objective functions, applications models, QoS constraints, strategies dealing with dynamic behavior of resources, and so on. In general, scheduling workflow applications in a distributed system is an NP-complete problem [2].

A Grid workflow can be represented as a Directed Acyclic Graph (DAG) or a non-DAG[3]. Nowadays, many scheduling algorithms about the DAG-based Grid workflow are developing and are divided into two schemes: OPTIMIST and PESSIMIST[4]. Time constraints are relaxed in the first scheme, and rigid in the latter. In the Grid workflow of PESSIMIST, it requires overall deadline, expenditure and reliability constraints for individual tasks, which are explicitly specified by end-users. There are also

other constraints, such as network and availability constraints, etc. Here we focus on the overall deadline, expenditure and reliability constraints for QoS requirements within workflows.

The remaining part of the paper is organized as follows: Section 2 introduces the related work in terms of the availability and scheduling scheme. Section 3 then describes the architecture and functionalities of the Grid workflow. Section 4 proposes the algorithm modeling, definition and calculation of the Grid resource reliability within the critical path and illustrates each step through a simple example. Section 5 provides performance evaluations through mathematical analysis and experiments. Section 6 presents the conclusions.

2 Related works

Prediction-based dynamic scheduling uses dynamic information in conjunction with some results based on prediction. Jin Hyun Son et al.[3] worked out the critical path first, and then used the M/M/c model of Markovian queuing systems to determine the minimum number of parallel Grid resources for tasks so as to cost as low as possible. Rajkumar Buyya[5] proposed a workflow scheduling algorithm that minimizes the cost of execution while meeting the deadline by using Markov Decision Process approach after finding out the critical path in the DAG. Analogously, X.-H.Sun's GHS[6] modeled the resource usage pattern with a M/G/1 queue system to evaluate the impact of resource availability on the performance of a remote task at a certain resource reservation rate, and its goal of scheduling is to minimize the failure-minimization while satisfying the deadline requirement of remote tasks. However, few of them really take account of the volatility of resources. The Grid resources are based on the assumption that the machines on the Grid never break down or never present abnormal performance when workflow tasks are running on them. In fact, the execution time of workflow is affected by 'normal state' and 'abnormal state' of Grid resources occurring in successive turns. As a result, unsuccessful execution of a workflow task at any point will result in the failure of meeting user's deadlines. GHS[6] considered this failure and used a rescheduling trigger system to migrate a task execution to another resource when its initial contract is broken or a better resource is found, but it is difficult to meet the overall deadline finally.

In order to produce a good schedule, estimating the resource scheme reliability (i.e. a probability that all of working resources for workflow tasks are being in the 'normal state') is crucial, especially for constructing a preliminary workflow schedule. By using estimation techniques, it is possible for workflow schedulers to predict how tasks in a workflow or sub-workflow will behave on distributed heterogeneous resources and thus make decisions on how and where to run them.

Overall deadline time is a basic measure for the workflow performance. To model the entire workflow as an optimization problem will produce the larger scheduling overhead. In some literatures (e.g.[5][7][8]), Quite a few algorithms are mainly considered to meet the execution time request of the critical path tasks (activities), because the critical

path is a sequence of activities from the beginning to the end of a workflow that has the longest average execution time. The task within the critical path is called the critical task. Similarly, we focus on the resource combination reliability within the critical path. To the critical path task in a DAG, the reliability of Grid resources should be the first consideration which is prior to the expenditure limit, et al, but not to the non-critical path task. There is a simple reason that even if the lower reliability of resources result in unsuccessful execution of a non-critical path task which is relatively short, the task will be rescheduled to prevent the potential performance loss and there is very little probability for failure to recur because of the multiplication of probabilities principle. But we have allowed for another case at the same time: if difference in execution time makespan between tasks on the critical path and tasks on the non-critical path is less than some relative threshold, the question have to be explored further. Under the above-mentioned condition, if we just take into account the reliability of Grid resources within the critical path, the result is nonsense, because importance of the Grid resource reliability within the non-critical path is not less than within the critical path, from another point of view, which is logically equivalent to the reliability of the two-units system in series, and any execution failure of task within the non-critical path will affect the subsequence task within the critical path. Through above analysis, we proposed a stochastic algorithm based on finite-state continuous-time Markov process to solve the problem.

3 Overview of Grid workflow management system

Workflow is concerned with the automation of procedures whereby files and data are passed between participants according to a defined set of rules to achieve an overall goal[9]. Fig.1. shows the architecture and functionalities supported by various components of the Grid workflow system based on the workflow reference model proposed by Workflow Management Coalition (WfMC)[10]. At the highest level, functions of Grid workflow management systems could be characterized into build time functions and run time functions. The build-time functions are concerned with defining, and modeling workflow tasks and their dependencies; while the run-time functions are concerned with managing workflow executions and interactions with Grid resources for processing workflow applications. Users interact with workflow modeling tools to generate a workflow specification, which is submitted to a run-time service called the workflow enactment service for execution. Major functions provided by the workflow enactment service are scheduling, fault management and data movement. The workflow enactment service may be built on the top of low level Grid middleware (e.g. Globus toolkit), through which the workflow management system invokes services provided by Grid resources[1]. To ensure workflow management system adopt appropriate scheduling strategy and allocate the corresponding resource to execute task of workflow specification, the information about resources may need to be retrieved through GIS(Grid information services). This information include identifying the list of authorized machines, cost of resource access, keeping track of resource status information, parameters, the historical

data related to a particular user's application performance or experience which can also be used in predicting the share of available of resources for user while making scheduling decisions based on QoS constraints. As Grid resources are not dedicated to the owners of the workflow management systems, the Grid workflow management system also needs to identify dynamic information, such as resource accessibility, system workload, and network performance during execution time.

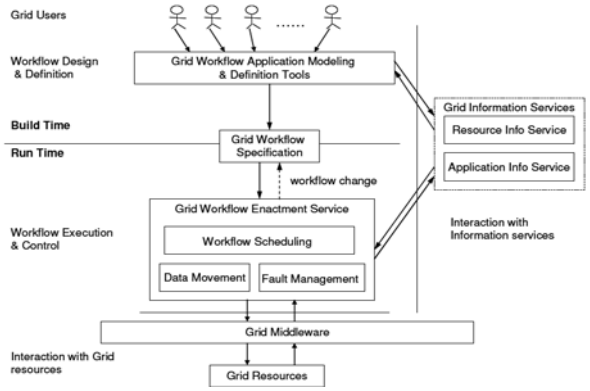

Fig.1. Grid workflow management systems

4 Description of Algorithm

This section consists of two parts to illustrate our approach. 1st part is the algorithm modeling, which include definition and calculation of reliability of Grid resources within the critical path. 2nd part describes main steps of algorithm.

4.1 The Model

The application can be represented by a directed acyclic graph G(V,E,T) as shown in Fig. 2., where :V(G) ={V1,V2,V3,V4,V5} is the set of v nodes, and each node vi(vi∈V) represents a task starting point or ending point. Not losing generality, v nodes can be classified as a disunited node (e.g. node v1 in Fig.2.) which have more than one subsequence node, simple node(e.g. node v2,v3,v5 in Fig.2.)which have not more than one previous node or subsequence node, united node(e.g. node v4 in Fig.2.) which have more than one previous node, and hybrid node which have both disunited node character and united node character(such node does not appear in our example to simply demonstrate our methodology).

E(G)={e1,e2,e3,e4,e5} is the set of workflow tasks, in which, e1=<V1,V2>, e2 =<V1,V3>, e3=<V2 ,V4>, e4=<V3,V4>, e5=<V4,V5>. The directed edge ei joins two nodes which denote task starting point and ending point respectively.

T(G)={t1,t2,t3,t4,t5}={7, 5, 8, 12, 4} is the set of computation costs in which each ti gives the estimated execution time of related task ei (e.g. t1=7 denotes that estimated execution time of ei is 7 time unit) .

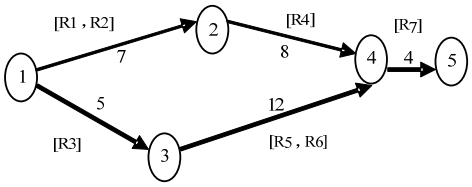

Fig. 2. Task and resource pool of DAG-Grid workflow application

In addition, we can reasonably assume in the Grid environment that the number of Grid resources is very likely more than one and that any of these resources can finish certain task ei within the same time limit. These resources are defined as a resource pool of ei which is denoted by Rei(e.g. Re1= [R1,R2]). Furthermore, the execution time of workflow is affected by 'normal state' and 'abnormal state' of Grid resources occurring in successive turns. As discussed in section2, we assume in this paper that both the 'normal state' and the 'abnormal state' of the Grid resource Ri are exponentially distributed with parameterλRi,μRi, and that resource owners specify their service price and charge users according to the amount of resources they consume. Service price per unit time required by Grid resource Ri is denoted by cRi. Thus, as showed in Fig.2, the number of resource pool is five, i.e. Re1=[R1(λR1,μR1, c R1), R2(λR2,μR2, cR2)]; Re2=[R3(λR3,μR3, cR3)]; Re3=[R4(λR4,μR4, c R4)] ; Re4=[R5(λR5,μR5, cR5), R6(λR6,μR6, cR6)]; Re5=[R7(λR7,μR7, cR7)]. By the way, according to section3, the above information related to resources can be obtained from GIS.

Definition 1. Critical united node is defined as a united node on the critical path in DAG, e.g. node v4 in Fig.2. (It is easy for us to write cp (critical path) out: cp=V1→V3→V4→V5, which is marked out with thick lines)

Definition 2. Critical disunited node is defined as a disunited node on the critical path in DAG, e.g. node v1 in Fig.2.

Definition 3. Critical region denoted by [Vi,Vj] is defined as a region of task between critical united node and critical disunited node in DAG, e.g. [V1,V4] is a critical region in Fig.2.

Definition 4. Set of related tasks within critical region denoted by $E_{[vi,vj]}$ is defined as all of tasks within critical region, which have to satisfy following condition(relative threshold): $t_v > t_r$ (t_v denotes estimated execution time of any task on the non-critical path within [Vi,Vj]; t_r is equal to the difference of total estimated execution time of all tasks between on the critical path and on the non-critical path within [Vi,Vj]). Further, total estimated execution time of $E_{[vi,vj]}$ denoted by $T_{[vi,vj]}$ is equal to the total estimated execution time of all tasks within [Vi,Vj]. For example, owing to t2+t4-t1-t3=t_r=T4=2, t1>t_r, t3>t_r, hence, $E_{[v1,v4]}$= {e1,e2,e3, e4}, $T_{[v1,v4]}$= t2+t4=17. For the sake of simplicity, our example only presented the single non-critical path and the single critical region.

Definition 5. The reliability of single resource denoted by P_{Ri} is defined as absolute probability by which single resource Ri is in 'normal state'.

Definition 6. The reliability of parallel resource group denoted by $P_{<Ri\text{-}Ti,Rj\text{-}Ti>}$ is defined as following: Clearly, $T_{[vi,vj]}$ can be separated into several phases T1,T2,…Tn (we call them the parallel period of time), and if ei and ej (or a group of tasks)in $E_{[vi,vj]}$ are always being executed at Ri and Rj (or a group of resources) during the phase Ti respectively, we define Ri and Rj as parallel resource group in the phase Ti, denoted by <Ri-Ti, Rj-Ti>; define ei and ej(possibly partial or entire) as the parallel task segment on the phase Ti denoted by <ei-Ti, ej-Ti>; while probability that Ri and Rj are always being in 'normal state' during the phase Ti is called reliability of parallel resource group denoted by $P_{<Ri\text{-}Ti,Rj\text{-}Ti>}$. Fig.3. can help to comprehend the definition: T[v1,v4] can be separated into 4 phases, T1=5; T2=2; T3=8; T4=2; parallel task segment is, respectively, <e1-T1, e2-T1>, <e1-T2, e4-T2>, <e3-T3, e4-T3>, <0, e4-T4>; if we choose R2 in R_{e1}={R1,R2} as execution resource of e1 and R6 in R_{e4}={R5,R6} as execution resource of e4, parallel resource groups related to above 4 parallel resource group will be <R2-T1,R3-T1>, <R2-T2,R6-T2>, <R4-T3, R6-T3>, <0,R6-T4>, while related 4 reliability of parallel resource groups will be $P_{<R2\text{-}T1,R3\text{-}T1>}$, $P_{<R2\text{-}T2,R6\text{-}T2>}$, $P_{<R4\text{-}T3,R6\text{-}T3>}$, $P_{<0,R6\text{-}T4>}$, respectively.

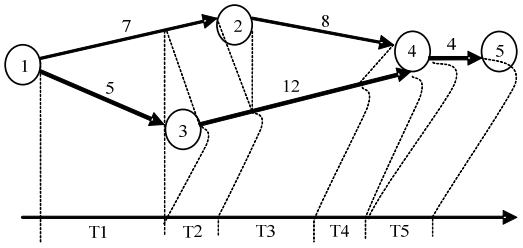

Fig.3. Parallel execution time of DAG-Grid workflow application

Definition 7. The reliability of critical region denoted by $P_{[Vi,Vj]}$ is defined as a probability that all parallel resource groups are being in the 'normal state' during their respective parallel phase. By multiplication of probabilities principle, We have the following relation: $P_{[Vi,Vj]}=\prod_{T(i=1)}^{n} P_{<Ri\text{-}Ti,Rj\text{-}Ti>}$; Assume that, in [Vi,Vj], the number of resource combination scheme is m, then the scheme i reliability of critical region is denoted by $P_{[Vi,Vj]\text{-}i}$. For example, in Fig.3., there are 4 schemes, i.e. ①{R1,R3,R4,R5},②{R1,R3,R4,R6},③{R2,R3,R4, R5},④{R2,R3,R4,R6}, then the 4th scheme reliability of critical region is denoted by $P_{[V1,V4]\text{-}4}$, and $P_{[V1,V4]\text{-}4}=P_{<R2\text{-}T1,R3\text{-}T1>}*P_{<R2\text{-}T2,R6\text{-}T2>}*P_{<R4\text{-}T3, R6\text{-}T3>}*P_{<0,R6\text{-}T4>}$.

Definition 8. The reliability of resource combination on the critical path denoted by P_{cp} is defined as a probability which is equal to the certain scheme reliability of the critical region multiplied by the certain scheme reliability of the non-critical region.

Definition 9. Credit level of reliability of resource denoted by α is defined as a

reliability of resource combination on a critical path required by user.

In the following, according to above definition, we derive a mathematic model to ascertain P_{Ri}, $P_{<Ri\text{-}Ti,Rj\text{-}Ti>}$, $P_{[Vi,Vj]}$ and P_{cp} by computation.

Learning from knowledge of stochastic process, we are able to conclude that the process, in which the 'normal state' and the 'abnormal state' of the Grid resource occur in successive turns and both the 'normal state' and the 'abnormal state' of the Grid resource Ri are exponentially distributed with parameterλ_{Ri},μ_{Ri} , is a finite-state continuous-time homogeneity Markov process. If 'abnormal state' happened to any resource of <Ri-Ti,Rj-Ti> within Ti, the successor task of [Vi,Vj] would not start at appointed time, it is logically convenient to visualize $P_{<Ri\text{-}Ti,Rj\text{-}Ti>}$ as reliability of the two-units system in series, as analyzed in section 2. Thus, we can get the two-units system in series models with state space S={0,1,2}, where 0 implies that any resource of <Ri-Ti, Rj-Ti> is in the 'normal state'; 1 implies that the Ri resource of <Ri-Ti,Rj-Ti> is in the 'abnormal state'; 2 implies that the Rj resource of <Ri-Ti,Rj-Ti> is in the 'abnormal state'. Setting N_t, $N_t \in S$, implies the state of the two-units system in series on time t, and the infinitesimal behavior of N_t is governed by the following q-matrix (the infinitesimal generator matrix) [11].

$$Q = \begin{bmatrix} -(\lambda_{Ri} + \lambda_{Rj}) & \lambda_{Ri} & \lambda_{Rj} \\ \mu_{Ri} & -\mu_{Ri} & 0 \\ \mu_{Rj} & 0 & -\mu_{Rj} \end{bmatrix} \tag{1}$$

The q-matrix as in (1) is obviously communicating and is possessed of symmetric properties with symmetric distribution $\alpha = (\alpha_0, \alpha_1, \alpha_2)$, and satisfies $\alpha_0 \lambda_{Ri} = \alpha_1 \mu_{Ri}$, $\alpha_0 \lambda_{Rj} = \alpha_2 \mu_{Rj}$, namely, $\alpha_1 = \frac{\lambda_{Ri}}{\mu_{Ri}} \alpha_0$, $\alpha_2 = \frac{\lambda_{Rj}}{\mu_{Rj}} \alpha_0$. Setting $\pi = (\pi_0, \pi_1, \pi_2) = (1, \frac{\lambda_{Ri}}{\mu_{Ri}}, \frac{\lambda_{Rj}}{\mu_{Rj}})(1 + \frac{\lambda_{Ri}}{\mu_{Ri}} + \frac{\lambda_{Rj}}{\mu_{Rj}})^{-1}$, then it is the reversible distribution of N_t and the reliability of the two-units system in series

$$P_{<Ri\text{-}Ti,Rj\text{-}Ti>} = \pi_0 = (1 + \frac{\lambda_{Ri}}{\mu_{Ri}} + \frac{\lambda_{Rj}}{\mu_{Rj}})^{-1} \tag{2}$$

By virtue of (2), it can be deduced that reliability of Single resource

$$PRi = \mu Ri / (\lambda Ri + \mu Ri) \tag{3}$$

And it can also extend to reliability of the multi-units system in series[11]. For the simplicity, we just only describe our algorithm through example of the two-unit system in series.

As showed in Fig.3, if we select the 4th resource combination scheme, by virtue of (2), can get results as following:

$P_{<R2\text{-}T1,R3\text{-}T1>} = (1 + \frac{\lambda_{R2}}{\mu_{R2}} + \frac{\lambda_{R3}}{\mu_{R3}})^{-1}$; $P_{<R2\text{-}T2,R6\text{-}T2>} = (1 + \frac{\lambda_{R2}}{\mu_{R2}} + \frac{\lambda_{R6}}{\mu_{R6}})^{-1}$;

$P_{<R4\text{-}T3,R6\text{-}T3>}=(1+\frac{\lambda_{R4}}{\mu_{R4}}+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}$; $P_{<0,R6\text{-}T4>}=(1+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}$;

$P_{[V1,V4]\text{-}4}=P_{<R2\text{-}T1,R3\text{-}T1>}*P_{<R2\text{-}T2,R6\text{-}T2>}*P_{<R4\text{-}T3,R6\text{-}T3>}*P_{<0,R6\text{-}T4>}$

$=(1+\frac{\lambda_{R2}}{\mu_{R2}}+\frac{\lambda_{R3}}{\mu_{R3}})^{-1}*(1+\frac{\lambda_{R2}}{\mu_{R2}}+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}*(1+\frac{\lambda_{R4}}{\mu_{R4}}+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}*(1+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}$。

Analogously, $P_{[V1,V4]\text{-}1}$, $P_{[V1,V4]\text{-}2}$, $P_{[V1,V4]\text{-}3}$ can be deduced. In addition, there is only one task e5 in non-critical region and R_{e5}=[R7] means that there is only one available resource, $P_{R7}=(1+\frac{\lambda_{R7}}{\mu_{R7}})^{-1}$, given by (3). Thus, $P_{cp}=P_{<V1,V4>\text{-}4}*P_{R7}=(1+\frac{\lambda_{R2}}{\mu_{R2}}+\frac{\lambda_{R3}}{\mu_{R3}})^{-1}*\ (1+\frac{\lambda_{R2}}{\mu_{R2}}+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}*(1+\frac{\lambda_{R4}}{\mu_{R4}}+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}\ *(1+\frac{\lambda_{R6}}{\mu_{R6}})^{-1}*\ (1+\frac{\lambda_{R7}}{\mu_{R7}})^{-1}$, namely it is one of reliability of resource combination on critical path in DAG(the 4th scheme on the critical region and R7 on the non-critical path). Other P_{cp} can be analogously calculated through the above-mention methodology.

4.2 Main steps of algorithm

Step 1. Find out the critical path e in DAG employing the traditional classic algorithm. (We omit them in this paper, to allow for being less related to the topic).
Step 2. Find out the entire critical region [Vi,Vj] on the critical path.
Step 3. While (each critical region)
{ If ($E_{[vi,vj]}$ exist)
//Discriminance is based on the //relativethreshold condition
{ Compute each $P_{[Vi,Vj]\text{-}i}$;
// (i∈{1,2,…m=the number of //resource combination scheme //in [Vi,Vj]})
E(G)=E(G)- E[vi,vj]
}
}
Step 4. If (E(G)≠NULL)
{WHILE (each task ei)
//ei∈E(G)
Compute each reliability of single resource Ri;
//Ri∈R_{ei} and ei∈E(G)
}
Step 5. Compute each P_{cp} and each cost of resource combination on the critical path in DAG.
Step 6. Choose a scheme of resource combination on the critical path in DAG, which should satisfy the following condition: $P_{cp}>\alpha$ and minimize the total expenditure.

5 Performance evaluation

We have performed simulations of the example discussed in Fig.2 and Fig.3, in which QoS requirements is under the α=75% credit level of reliability of resource combination on the critical path in DAG and used related simulation parameters is illustrated as following Table 1.

According to our algorithm and above parameters, we get each value of P_{cp} (the reliability of resource combination on the critical path in DAG) and related total expenditure, as showed in Table 2.

Table 1. Simulation Parameters

Resource Ri	λ_{Ri}	μ_{Ri}	$P_{Ri}=\mu_{Ri}/(\lambda_{Ri}+\mu_{Ri})$	c_{Ri}
R1	1/5	5	0．9615	12
R2	1/6	4	0．9600	9
R3	1/7	3	0．9545	5
R4	1/8	2	0．9412	4
R5	1/12	3	0．8780	3
R6	1/11	6	0．9851	13
R7	1/13	7	0．9891	14

Table 2. Value of P_{cp} and Total expenditure

Scheme of resource	P_{cp}	Total expenditure
①{R1,R3,R4,R5,R7}	0.5638	233
②{R1,R3,R4,R6,R7}	0.7879	353
③{R2,R3,R4,R5,R7}	0.5622	212
④{R2,R3,R4,R6,R7}	0.7854	332

Clearly, scheme ④ of resource combination on the critical path in DAG, i.e. {R2, R3, R4, R6, R7} should be the best preliminary workflow schedule, because it be able to satisfy following condition: Pcp>75% and with lower total expenditure, From Table 2.

We present the most important part of our simulation and analysis below. Fig.4. illustrates a case of two states of all resources, in which, higher intermittent line attaching to Ri denotes ‘abnormal state’, while lower intermittent line attaching to Ri denotes the

'normal state'. To scheme ④, in T3, the 'abnormal state' happen to R2, but R2 have already finished e1 task in the first two phase T1 and T2 ; it is R4 and R6 being in the 'normal state' that is required to carry out task e3 and e4, according to Fig.3., R6 keeps on executing e4 and its state keeps on being 'normal state' throughout T4; subsequently, the 'abnormal state' happen to R6 in T5, but R6 have already finished workflow task in T4 and these 'abnormal state' do not affect the entire workflow. Therefore, in all the phase T1-T5, corresponding resources always are being the 'normal state' during the time of executing respective task. Based on these observations, workflow has been carried out successfully. To scheme ③, parallel resource group in the phase T3 is <R4-T3,R5-T3>. Although R4 can complete task e3, the 'abnormal state' happened to R5 in T3, which will make R7 to fail to start e5 on time eventually. Hence, if choose scheme ③, it would not meet the time limit in experiments this time. We have conducted a large number of above experiments under the same conditions, successful workflow execution rate of each scheme is as in Fig.5.

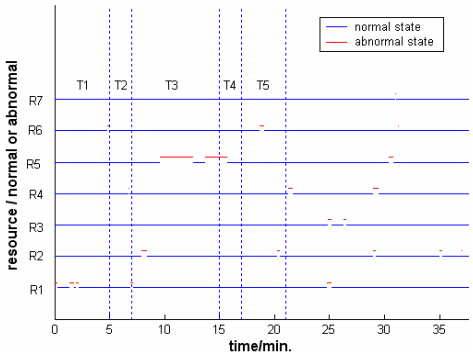

Fig.4. Two State Alternation of all resource

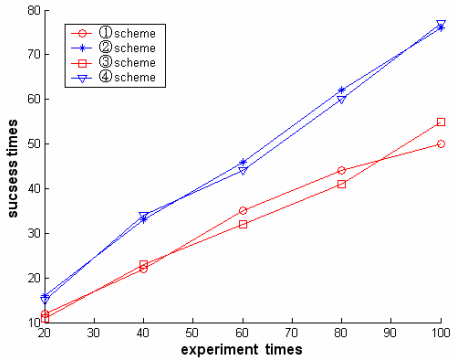

Fig.5. Successful execution rate of each scheme

Above simulation experiments demonstrate the efficacy of workflow scheduling algorithm. In the Grid computing environment, when there are a few resources which can finish a certain task within the same time limit and whose reliability are different, our algorithm is able to efficiently choose more reliable Grid resource to carry out time-restricted workflow with the lower total expenditure.

6 Conclusion

In this paper, we present an approximation method for computing the reliability of resource combination. This method is constructed based on stochastic processes, finite-state continuous-time Markov process. Simulation experiments show that our approach produces both good execution performance and scheduling results. Assuming that both the 'normal state' and the 'abnormal state' of the Grid resource are exponentially distributed is approximate approach. In our future work, we are going to substitute more appropriate probability density functions for the exponential distribution function to improve Grid workflow scheduling performance.

Acknowledgements

This research is supported by National Natural Science Foundation of China under Grant No. 60563002 and Scientific Research Program of the Higher Education Institution of XinJiang under Grant No. XJEDU2004I03.

References

1. Jia, Y., Rajkumar, B.: A Taxonomy of Workflow Management Systems for Grid Computing. J. Grid Computing. 3(3-4), 171-200 (2005)
2 Ullman, J.D.: NP-complete Scheduling Problems. J. Computer and System Sciences. l(10), 384-393 (1975)
3 Jin, H.S., Myoung H.K.: Improving the performance of time-constrained workflow processing. J. System and Software. 58(3), 211-219 (2001)
4 Eunjoung, B., Sungjin, C., Maengsoon, B., et al.: MJSA Markov job scheduler based on availability in desktop grid. Future Generation Computer Systems. 23(4), 616–622 (2007)
5 Jia, Y., Rajkumar, B., Chen, K.T.: QoS-based Scheduling of workflow Applications on Service Grids. In: 1st IEEE International Conference one-Science and Grid Computing, IEEE CS Press, Melbourne (2005)
6 Xianhe, S., Lingou, G., Edward, F.W.: Performance modeling and prediction of non-dedicated network computing. IEEE Trans. Comput. 51(9), 1041-1055 (2002)
7 Jia, Yu , Rajkumar Buyya and Chen Khong Tham. Cost-based Scheduling of Scientific Workflow Applications on Utility Grids. J. System and Software. 233, 236-242 (2005)
8 Hagras, T., Janec, J.: A high performance and lower complexity algorithm for

compile-time task scheduling in heterogeneous systems. Parallel Computing. 31, 653-670 (2005)
9 Hollinsworth, D.: The Workflow Reference Model. Technical report, Workflow Management Coalition, TC00-1003 (1994)
10 W3C. XML Pipeline Definition Language Version 1.0, http://www.w3.org/TR/2002/NOTE-xml-pipeline-20020228
11 Guang-lu, G.: Applied Stochastic process tutorial. Beijing, China. Tsinghua University publishing company, 149-155 (2004)