# A GSPN-based Grid Resource Schedule Algorithm

Zhi-xiang YUAN
School of Computer Science
Anhui University of Technology
Maanshan, 243002,China
zxyuan@ahut.edu.cn

Jun Zhou
School of Computer Science
Anhui University of Technology
Maanshan, 243002,China

Hong Ye
School of Computer Science
Anhui University of Technology
Maanshan, 243002,China

***Abstract*—The paper analyses the load imbalance problem and the QoS-based fault-tolerant scheduling algorithm in Grid Resource Scheduling, and proposes a new scheduling algorithm based on the priority of a task-based parameters of Qos constrained scheduling strategy. The method is based on using the generalized stochastic Petri nets with inhibitor arc to establish the grid scheduling model and improve the Min-Min algorithm. Experimental results show that the algorithm can decrease the overall cost of time and cost, when the Grid Resource Schedule runs an urgent task.**

***Keywords*—Grid computing, Resource scheduling, The Generalized stochastic Petri net, QoS constraint, load balance**

## I. INTRODUCTION

Resource Scheduling has played an important role in Resource Optimization of Service Computing. There are many Resource Scheduling algorithms and models in recent years[1]. As methods of modeling and analyzing physical systems, Petri nets have shown their abilities to deal with concurrence conflict and have been widely used to model, analyze, and verify workflow . Because the application of Service Computing is similar to workflow , which is composed of service task, Petri Nets have mainly been used for modeling and analysis of grid service[2,3].

In this paper, we discuss the modeling and analysis approach for this kind of Grid services based on Petri net. In previous modeling approaches for grid computing based on Petri nets, Dario present a methodology to analyze performance in gLite Grids through the use of Generalized Stochastic Petri Nets (GSPNs),and introduce a cluster-level model of a typical gLite site taking into account the coexistence between normal and MPI-based jobs[4]. The application which runs on the grid and provides continuous service has the dynamic reconfiguration capability to ensure the consistency of the system. Reliable services composition can be achieved through modeling and analysis of Web service composition. The method of the use of Petri net for modeling and analysis of service composition is researched in the paper[5,6]. The problem of quality of service (QoS) is the vital issue to the grid scheduling model. The scheduler in the Grid environment needs to consider application and QoS constraints to get a better match between applications and resources [7,8,9,12,13]. A QoS guided task Min-Min algorithm has been proposed in the papersto solve the QoS based scheduling problem, which based on the task requirement of QoS, classifies tasks into two categories: high QoS-requiring tasks and low QoS-requiring tasks, and schedules the two groups separately. However the Min-Min scheduling algorithm has load imbalance problem. We investigate the influence of several parameters (e.g., priorities, scheduling policies)on both the performance and benefits, study Qos-based fault-tolerant scheduling algorithm in Grid Resource Scheduling, then propose a new scheduling algorithm based on the priority of a task-based parameters of Qos constrained scheduling strategy. The Petri Net-based model grid services is presented for the grid service scheduling, which is the generalized stochastic Petri nets with inhibitor arc. the new scheduling algorithm merges the efficiency of Min-Min and the load balance of Max-Min for every QoS level group scheduling. The experimental results show that the algorithm can decrease the overall cost of time and cost, When the Grid Resource Schedule runs an urgent task.

## II. THE GRID SERVICE SCHEDULING MODEL

### A. Problem Description

In Grid service, task scheduling is to map tasks onto heterogeneous distributed computing resource and search the schedule with optimum resources objective. In this paper, task scheduling focuses on a class of independent tasks with QoS constraints.

Let $m$ independent tasks to be scheduled on $n$ service resources(machines). Let $\{t_1,t_2,\ldots,t_m\}$be the metatask with all the dependent tasks, and $\{R_1,R_2,\ldots,R_n\}$ be machines in Grid. The problem for resource scheduling tries to find the schedule with optimum objective, which maps $n$ tasks onto $m$ machines. The goal of problem is that how to find the minimum makespan. Due to the heterogeneity of Grid resource and OoS requirements, the user can set the task priority, time and cost constraints, and set the ratio of time and cost. This paper mainly focuses on how to map tasks to machines in order to optimize cost, priority and completion time for the metatask.

Let $L_{ij}$ be the completion time which task $t_i$ run on resource $R_j$. Let $e_{ij}$ be the expected execution time which task $t_i$ runs on resource $R_j$. Let $S_{ij}$ be the transmission time which task $t_i$ transfer to resource $R_j$. Let $r_j$ be the ready time for resource $M_j$. Let $P_{ij}$ be the price which task $t_i$ transfer to resource $R_j$. Let the require data throughput of the task $t_i$ be $S_i$. Let $bw$(j) be the bandwidth. Let $M_i$ be the maximum running cost of task $t_i$ .Let $T_i$ be the maximum running time of task $t_i$.

$$L_{ij} = \begin{cases} e_{ij} + r_j & r_j \geq S_{ij} + S_i / bw(j) \\ e_{ij} + S_{ij} + S_i / bw(j) & r_j < S_{ij} + S_i / bw(j) \end{cases}$$

978-0-7695-4313-0/10 $26.00 © 2010 IEEE
DOI 10.1109/GCC.2010.94

Let makespan be Zi, and the objective function be min$Z_i$, where $\min Z_\mathrm{i} = W_m \times e_{ij} \times P_{ij} + W_t \times L_{ij}$ .

### B. *Resource Scheduling alogrithm*

Resource scheduling in Grid is hard to solve for time-cost optimization under QoS restriction. Due to the heterogeneity of Grid resource and QoS requirement, independent task scheduling in Grid is actually a bi-criteria optimization problem with many constraints. The motivation of the proposed algorithm in this paper is to achieve less execution cost with minimum execution time according to task emergency (or priority).

When the task priority is joined in the QoS requirement, the task is dived into the high and lower priority. The different priority task runs the same scheduling policy, which the high priority tasks firstly run and the lower priority tasks run lastly. The Qos-based priority task scheduling algorithm is formally described as follows.

For all task $t_i$<br>
  If task $t_i$ priority is high then<br>
    $t_i \in$ PH(the high priority set)<br>
  Else<br>
    $t_i \in$ PL(the lower priority set)<br>
End for<br>
If PH≠Φ then<br>
begin<br>
for each task $t_i$ in PH<br>
for each resource $R_j$ in PH<br>
  if $r_j > S_{ij} + S_i$/bw(j) then<br>
    $L_{ij} = e_{ij} + r_j$<br>
  else<br>
    $L_{ij} = e_{ij} + S_{ij} + S_i$/bw($j$)<br>
  end if<br>
  $Z_i = W_m \times e_{ij} \times P_{ij} + W_t \times L_{ij}$<br>
end for<br>
end for<br>
do until all tasks are mapped<br>
  for each task find the minimum makespan Z($e_{ij} \times P_{ij} \le M_i$ and $L_{ij} \le T_\mathrm{i}$)<br>
  if not resource $r_j$ with QoS constraint then<br>
    exit<br>
  end if<br>
  end for<br>
  find the minimum Resource $r_j$, delete task $t_i$<br>
update $r_j$<br>
update $L_{ij}$ for i<br>
end do<br>
end begin<br>
If PL≠Φ then<br>
begin<br>
for each task $t_i$ in PH<br>
for each resource $R_j$ in PH<br>
  if $r_j > S_{ij} + S_i$/bw(j) then<br>
    $L_{ij} = e_{ij} + r_j$<br>
  else<br>
    $L_{ij} = e_{ij} + S_{ij} + S_i$/bw($j$)<br>
  end if<br>
  $Z_i = W_m \times e_{ij} \times P_{ij} + W_t \times L_{ij}$<br>
end for<br>
end for<br>
do until all tasks are mapped<br>
  for each task find the minimum makespan Z($e_{ij} \times P_{ij} \le M_i$ and $L_{ij} \le T_\mathrm{i}$)<br>
  if not resource $r_j$ with QoS constraint then<br>
    exit<br>
  end if<br>
  end for<br>
  find the minimum Resource $r_j$, delete task $t_i$<br>
update $r_j$<br>
update $L_{ij}$ for i<br>
end do<br>
end begin

### C. *Petri Net Model for Resource Scheduling*

Resource scheduling in Grid is hard to solve for time-cost optimization under QoS restriction. Due to the heterogeneity of Grid resource and QoS requirement, independent task scheduling in Grid is actually a bi-criteria optimization problem with many constraints. The motivation of the proposed algorithm in this paper is to achieve less execution cost with minimum execution time according to task emergency (or priority). The system we want to analyze is grid system made up of computational resources which are composed of a set of worker nodes, and of one or more computing elements. In this section, The service scheduling model based on GSPN with inhibitor arc is illustrated. let the ($P_{01}$,T) arc be the inhibitor arc which the high priority task should be scheduled firstly, then the lower priority task be scheduled. Here we discuss only the generalized stochastic Petri nets.

*Definition* 1: the generalized stochastic Petri nets mode for task scheduling (TGSPN) is a eitght-tuple TGSPN=$(P,T,F,K,W,R,M_0,\lambda)$, where ,

- P=$\{U,PL,PH,P_{01},P_{02},P_{03}\}$U$(Q_j,W_j,E_{pj},V_j|j$=1,2,3,4,..., $m\}$ is finite set of places, where, the descriptions of places are given in table 1,
- T=$\{C,Td,T,T_0,R_1,T_r\}$U$\{T_{j1},S_{tj},E_j,S_j,err_j|j$=1,2,3,4,...,$m$ } is finite set of transitions, where, the descriptions of transitions are given in table 1,
- F=$\{(C,U),(U,Td),(Td,PH),(Td,PL),(PL,T),(T,P_{01}),(PH,T),(P_{01},T),(P_{01},R_1),(R_1,P_{03}),(P_{03},Tr),(Tr,P_{01})(P_{01},T_0),(T_0,P_{02})\} \cup \{(P_{02},T_{j1}),(T_{j1},Q_j),(Q_j,S_{tj}),(S_{tj},W_j),(W_j,S_j),(S_j,V_j),(V_j,S_{tj}),(E_j,E_{pj}),(E_{pj},err_j),(W_j,err_j),(err_j,V_j)(err_j,P_{01})|$ $j$=1,2,3,4,...,$m$ },
- K($Q_j$)=$d_j$,$j$=1,2,...,$m$(constant $d$>1),K($U$)=K($PH$)=K($PL$)=K($P_{01}$)=h (constant h >> d ), K($P_{03}$)=K($P_{02}$)=1, K($E_{pj}$)=K($W_j$)=K($V_j$)=1, where $1 \le j \le m$,
- W($C$,$U$) =$X$,where X≥1，
- $M_0$(U)=u,$M_0(P_{01})=M_0(PH)=M_0(PL)=M_0(P_{02})=M_0(P_{03})$ =0,$M_0$ ($Q_j$)=$M_0$ ($E_{pj}$)=$M_0$ ($W_j$)=0, $M_0$ ($V_j$|)=1,where 1≤j≤m。
- λ= ( $\lambda_1,\lambda_2,\ldots,\lambda_m$ ) is the firing rate of timed

transitions, the enabling predicate of $T_0$ is(for $\forall j$, $1 \le j \le M(P_{01})$ find min $Z_i$,the enabling predicate of $T_{j1}$ is（$M(Q_j) < d_j$）$\wedge$(for $\forall k$, $1<k \ne j < d_j$, $Z_j \le Z_k$)$\vee$(for $\forall k$ , $1<k \ne j < d_j$, $M(Q_j)= d_j$), ,the enabling predicate of $T_{j1}$ is（for $\forall j$ ,$1 \le j \le m, M_i \le e_{ij} \times P_{ij}$）$\vee$(for $\forall j$ ,$1 \le j \le m$, $T_j \le L_{ij}$).

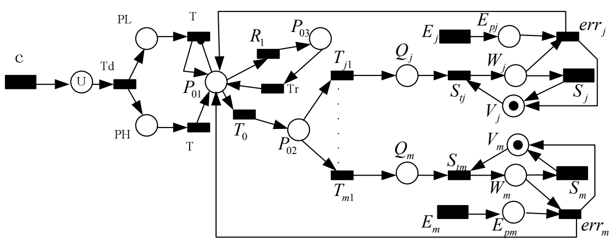


Fig. 1. A GSPN model of the task scheduling of grid

**Table 1** DESCRIPTION OF FIG.1

| Place | Description |
|---|---|
| $P_{01}$ | Tasks unmapped by user |
| $P_{02}$ | Tasks selected by algorithm |
| $P_{03}$ | Tasks remapped |
| U | Task set |
| PL | The lower priority tasks |
| PH | The higher priority tasks |
| $Q_j$ | The task queue allocated to Machine j |
| $W_j$ | The running state on Machine j |
| $E_{pj}$ | The used Machine j |
| $V_j$ | Machine j |
| **Transition** | **Description** |
| C | Submitting task to machine |
| Td | Judging the priority level |
| T | Sending task set |
| $T_0$ | selecting task from unmapped tasks |
| $R_1$ | Rescheduling task |
| $T_r$ | Setting the data of rescheduled task |
| $T_{j1}$ | Judging the task to submit Machine j |
| $S_{tj}$ | Executing task |
| $S_j$ | Committing task |
| $E_j$ | Submitting error |
| $err_j$ | Rescheduling after error happened |

## D. *Reachable Scheduling Graph*

Because the fault is unpredicted, the reachable scheduling graph(RSG) of the GSPN is constructed by following algorithm without fault.

Step 1) V={$M_0$},E={Φ},f=0. Place $M_0$ in level 0 , and tag it "new".

Step 2) while "new" marking exist, do the following:

Step 2.1) Select a new marking M.

Step 2.2) while there exist enabled predicate transition of $t_k$ at M, do the following for each enabled predication transition at *M*:

Step 2.2.1) obtain the marking *M'*, that results from firing $t_k$ at M.

Step 2.2.2) if $M' \notin V$ then $V=V+\{ M' \}$

Step 2.2.3) if $T_k=T_0$ then

$f=f+1$，Label the marking *M'* as the *f* level, and tag it "new"

else if $t_k \in \{ T_{j1} | j=1,2,\ldots,m\}$ then

Label the marking *M'*, $L_{ij}$, min$Z_i$ as the *f* level, and tag it "new"

else

Label the marking *M'* as the *f* level, and tag it "new"

Step 2.2.4) E=E+{M, M' }，label $t_k$ with {M, M' }

Step 2.2.5) if $t_k \in \{ T_{j1} | j=1,2,\ldots,m \}$ then

Label $T_{si}/M_{aj}$ with {M, M' },where $T_{si}$ *is task* and $M_{aj}$ is resource.

Step 2.3) if no transition are enabled at M, tag M "dead-node"

Step 2.4) Remove the "new" tag.

Step 3) Go to step 2).

*Definition* 2: Let RSG (V,E) be the reachable task graph of TGSPN, E is the state transition sequence ,and the task schedule sequence is composed of $T_{si}/M_{aj}$ on machine $M_{aj}$. In last level of section *j*, $l_{ij}$ is the makespan for the complete schedule time, and $max\{ l_{ij}\}$ is one of the task completion time in Grid.

## E. *Example*

Suppose that there are 3 independent tasks and there are two machines . Table 1 gives the expected execution time, transmitting time, budget and time-cost.

**Table 2.** THE EXPECTED EXECUTION TIME, TRANSMITTING TIME, BUDGET AND TIME-COST

| Task | expected execution time | | transmitting time | | budget | | time-cost | | Wm:Wt |
|---|---|---|---|---|---|---|---|---|---|
| | M1 | M2 | M1 | M2 | M1 | M2 | cost | time | |
| T1 | 5 | 6 | 3 | 7 | 10 | 9 | 18 | 13 | 1:2 |
| T2 | 2 | 3 | 2 | 2 | 4 | 4.5 | 5 | 6 | 1:3 |
| T3 | 4 | 10 | 3 | 10 | 8 | 15 | 10 | 14 | 1:4 |

Based on the model of the task scheduling of grid and the RSG algorithm, we know that the scheduling sequence of the Min-min algorithm with QoS is $T_2/M_1 \rightarrow T_1/M_1 \rightarrow T_3/M_1$, that of Min-min algorithm is $T_2/M_1 \rightarrow T_3/M_1 \rightarrow T_1/M_1$.

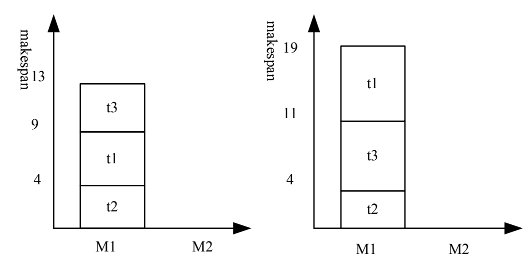


Fig.2. (a) Min-min algorithm with QoS (b)Min-Min algorithm

In Fig.2, The makespan of the Min-min with QoS algorithm is less than one of the Min-min algorithm, and there are the load imbalance problem in them. Now suppose that there are 4 independent tasks and there are two machines . Table 2 gives the user demand, expected execution time, transmitting time, budget and time-cost.

**Table 2.** THE USER DEMAND, EXPECTED EXECUTION TIME, TRANSMITTING TIME, BUDGET AND TIME-COST

| Task | user demand | expected execution time | | transmitting time | | budget | | time-cost | | Wm:Wt |
|---|---|---|---|---|---|---|---|---|---|---|
| | | M1 | M2 | M1 | M2 | M1 | M2 | cost | time | |
| T1 | Low priority | 5 | 6 | 3 | 7 | 10 | 9 | 18 | 13 | 1:2 |
| T2 | Low priority | 2 | 3 | 2 | 2 | 4 | 4.5 | 5 | 6 | 1:3 |
| T3 | High priority | 4 | 10 | 3 | 9 | 8 | 15 | 10 | 14 | 1:4 |
| T4 | High priority | 6 | 9 | 4 | 8 | 13 | 12 | 13 | 15 | 1:3 |

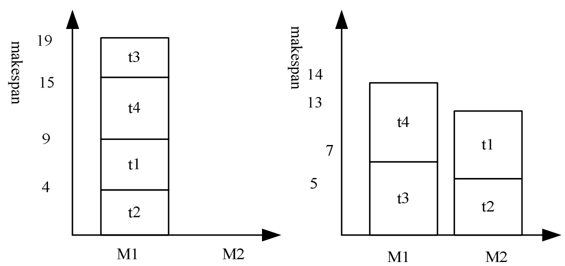


Fig.3.(a) Min-min algorithm without priority Qos (b) Min-min algorithm with priority Qos

In Fig.3, The makespan without priority QoS is 19,and the average value of result function is 47. The makespan with priority QoS is 19,and the average value of result function is 47.The new scheduling algorithm is better than the old one, and which has the load balance function and meet with the user priority requirements.

## III. CONCLUSIONS

In this paper, the resource scheduling algorithm with priority were presented. As a powerful graphical and analytical tool, A GSPN with inhibitor arc was presented and a simple model for the resource scheduling in grid computing environment was constructed using GSPN. The reachable scheduling graph was given and used to analyze the timing property and the sequence of the resource scheduling efficiently and intuitively. The example and experimental results show that the Min-min with priority QoS algorithm has better performance than the Min-min without QoS, which runs the emergency task firstly, and achieves the load balance function with less time-cost. We will consider the resource scheduling algorithm with multi-QoS in our future work.

## ACKNOWLEDGMENT

This work is supported by in part by the projects of Anhui Provincial Education Department National Research Foundation of China under Grants No. KJ2008B105.

## REFERENCES

[1] I Foster, C Kesselman, *et al*. The Grid: Blueprint for New Computing Infrastructure. San Francisco, CA: Morgan Kaufmann Publishers,1999.

[2] Murata, T.,"Petri Nets: Properties, Analysis And Applications," Proceeding of IEEE. Vol. 77,4 ,1989,pp541-580.

[3] Falko Bause "Stochastic Petri Nets- An Introduction to the Theory," unpublished.

[4] Dario Bruneo, Marco Scarpa, Antonio Puliafito, "Performance Evaluation of gLite Grids through GSPNs," IEEE Transactions on Parallel and Distributed Systems, 10 Feb. 2010. IEEE computer Society Digital Library. IEEE Computer Society, <http://doi.ieeecomputersociety.org/10.1109/TPDS.2010.35>

[5] Lin C, Wang YZ, Yang Y, Qu Y, "Research on network dependability analysis methods basedon stochastic Petri net". Acta Electronica Sinica ,vol.34,pp.322–332, February 2006 (in Chinese)

[6] Yu Xue-Li, Jiang, Jing, Xia, Bai-Qiang, Pan Zhen-Kuan,"Petri-net-based analysis method for grid services composition model", in Proceeding of the 2010 International Conference on Innovative Computing and Communication, 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, Macao, China, Jan 2010, pp180-184

[7] He Xiaoshan, "QoS Guided Min-Min Heuristic for Grid Task Scheduling," Journal of Computer Science and Technology, vol.18,pp442-451,April 2003.

[8] Guping Zheng,Yang Liu ,"A Selective Algorithm Based on Multiple QoS Constraints for Grid Task Scheduling", in Proceeding of the 2008 first International Conference on Intelligent Networks and Intelligent Systems (ICINIS), Wuhan, China , Jul 2008,pp239-342

[9] Kobra Etminani, M. Naghibzadeh. A Min-Min Max-Min Selective Algorihtm for Grid Task Scheduling. Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in CentralAsia on, Sept. 2007.

[10] LIU Wei-dong; SONG Jia-xing; LIN Chuang, "Modeling and Analysis of Grid Computing Application Based Price Timed Petri Net," Acta Electronica Sinica, vol. 33, pp. 1416–1420, August 2005

[11] M.A. Azgomi, R. Entezari-Maleki, "Task scheduling modelling and reliability evaluation of grid services using coloured Petri nets," Future Generation Computer Systems (2010), doi:10.1016/j.future.2010.05.015

[12] Han Y.-J., Jiang C.-J., "Extended Colored Time Petri Net -Based Resource Scheduling in Grid Computing," Proc. of the Int. Workshop on Grid and Cooperative Computing(GCC2002), Sanya,, China,2002,pp7345-7353.

[13] Han Y.-J., Jiang C.-J., "Resource Scheduling Algorithms for Grid Computing and Its Modeling and Analysis Using Petri Net," Proc. of the Int. Workshop on Grid and Cooperative Computing(GCC2003), LNCS 3033, pp. 73–80, 2004

[14] X-He. Sun, M. Wu, Quality of service of grid computing: resource sharing, in:Proceedings of the Sixth International Conference on Grid and CooperativeComputing, GCC'07, China, August, 2007, pp. 395-402.