第 36 卷 第 2 期 计 算 机 工 程 2010 年 1 月
Vol.36 No.2 Computer Engineering January 2010
·软件技术与数据库· 文章编号：1000—3428(2010)02—0045—03 文献标识码：A 中图分类号：TP311

基于广义随机 Petri 网的网格调度模型

袁志祥，王小平
(安徽工业大学计算机学院，马鞍山 243002)

摘 要：针对网格资源调度中负载不均衡问题，在基于 QoS 且具有容错性的任务调度算法基础上提出一种基于任务优先级的 QoS 约束参数的调度策略。采用广义随机 Petri 网建立网格调度模型，增加 Petri 网的抑制弧功能，实现优先调度策略。结果证明了该策略优先运行紧迫任务，并且其运行任务时间和费用的综合代价较小。

关键词：网格计算；任务调度；广义随机 Petri 网；QoS 约束

Grid Schedule Model Based on Generalized Stochastic Petri Net

YUAN Zhi-xiang, WANG Xiao-ping
(School of Computer Science, Anhui University of Technology, Maanshan 243002)

【Abstract】 This paper analyses the load imbalance problem and the QoS-based fault-tolerant schedule algorithm in grid resource schedule, and proposes a schedule algorithm based on the priority of a task-based parameters of QoS constrained schedule strategy. The method is based on using the generalized stochastic Petri net with inhibitor arc to establish the grid schedule 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.

【Key words】 grid computing; task schedule; generalized stochastic Petri net; QoS constraint

1 概述

资源调度是网格计算[1]中的主要研究内容之一，也是难点之一。在网格计算环境下，有效的资源调度对优化资源的使用起着非常重要的作用。目前，已提出不少网格计算资源调度模型与算法。Petri 网作为模拟与分析具有并发、异步、动态等特点的信息处理系统的有力工具，在描述工作流系统等模型和分析方面，显示出其强大的模型描述和性能分析能力[2]，而网格应用可分解其在网格资源上执行的任务的组合，从这点来说和工作流系统类似，因此，也常用 Petri 网来描述网格应用。

文献[3]定义了资源 QoS 约束和形式化描述，在任务完成期限和网络带宽的双重属性约束下结合预测机制，提出了网格资源调度算法 Senior，应用 GridSim 工具包实现了相关的调度算法，并对调度算法仿真结果中的数据进行了分析和比较，验证了 Senior 调度算法在解决类似问题方面的优势。本文借鉴了文献[4]的调度策略思想，针对该算法具有负载不平衡的特点对其中的算法进行了改进，增加了任务的优先级 QoS 约束，并利用广义随机 Petri 网[5]作为分析工具，为新算法建立了模型，并对模型进行了分析。

2 网格资源调度方案

2.1 问题分析

假设有 n 个计算资源$\{MA_i\}$，i=1,2,…,n，现在有 m 个任务映射到这 n 个同构的计算资源上。假设这 m 个任务是独立的。资源调度问题的实质就是在一个有 m 个需要调度的任务，n 个可用的任务执行单元(主机或集群)，把 m 个任务 T=$\{t_1, t_2, \cdots, t_m\}$以合理的方式调度到 n 个主机 H=$\{h_1, h_2, \cdots, h_n\}$上去，目的是得到尽可能小的最大完成时间(*makespan*)。引入了 QoS 约束之后，用户可以设定任务的优先级以及任务运行时间和费用的上限，也可以设定任务运行时间和费用之间的权重比(W_m, W_t)，以衡量时间和费用之间的重视程度。

根据上面的分析可以建立如下的 Petri 网模型。其中，L_{ij} 是指任务 i 在资源 j 上的完成时间；e_{ij} 表示任务 i 到资源 j 上的执行时间；S_{ij} 表示任务 i 到资源 j 的传输时间；r_j 表示资源 j 的就绪时间；P_{ij} 表示资源的单位计价；S_i 表示任务的数据传输量(预先估计)；$bw(j)$表示带宽；M_i 表示任务 i 的费用上限；T_i 表示任务 i 的时间上限。

$$L_{ij}=e_{ij}+r_j,(r_j>S_{ij}+S_i/bw(j)) \text{ or } \quad L_{ij}=e_{ij}+S_{ij}+S_i/bw(j)$$

其中，$bw(j)$表示带宽，目标函数 $\min Z_i=W_m\times e_{ij}\times P_{ij}+W_t\times L_{ij}$。

2.2 任务调度算法

引入任务的优先级 QoS 后，使得任务分为优先级高和优先级低 2 个任务子集，2 个子集里的任务都按照相同的任务影射策略进行调度，优先级高的子集里的任务优先调度完，而后再调度优先级低的子集里的任务。

具体的任务映射策略如下：

```
For 所有的任务 i
    If 任务 i 的优先级高 then
        任务 i∈高优先级任务集 PH
    Else
        任务 i∈低优先级任务集 PL
End for
If PH≠Φ then
```

基金项目：安徽省教育厅自然科学基金资助项目(KJ2008B105)
作者简介：袁志祥(1973－)，男，副教授、硕士，主研方向：Petri 网，协议验证，网格计算；王小平，硕士研究生
收稿日期：2009-08-07 **E-mail：**zxyuan@ahut.edu.cn

```
begin
  for 所有的高优先级任务集 PH 里的任务 i
    for 所有的计算资源 j
      if r_j>S_ij+S_i/bw(j) then
        L_ij=e_ij+ r_j
      else L_ij=e_ij+S_ij+S_i/bw(j)
      end if
      Z_i=W_m×e_ij×P_ij+W_t×L_ij
    end for
  end for
  do until 所有的任务都已被映射
    for 待影射的每个任务找到使得 Z 值最小的计算资源(e_ij×P_ij<=M_i and L_ij<=T_i)
      if 没有满足 QoS 要求的资源 then
        修改参数，重新进行调度
      end if
    end for
    在所有的任务中找到 Z 值最小的那个任务，分配到相应的计算资源删除该任务
    更新 r_j
    更新 L_ij for i
  end do
end begin
If PL≠Φ then
  begin
  for 所有的低优先级任务集 PL 里的任务 i
    for 所有的计算资源 j
      if r_j> S_ij+ S_i/bw(j) then
        L_ij= e_ij+r_j
      else L_ij= e_ij+S_ij+ S_i/bw(j)
      end if
      Z_i= W_m×e_ij×P_ij+ W_t×L_ij
    end for
  end for
  do until 所有的任务都已被映射
  for 待影射的每个任务找到使得 Z 值最小的计算资源(e_ij×P_ij<=M_iand L_ij<= T_i)
    if 没有满足 QoS 要求的资源 then
      修改参数，重新进行调度
    end if
  end for
  在所有的任务中找到 Z 值最小的那个任务，分配到相应的计算资源删除该任务
  更新 r_j
  更新 L_ij for i
  end do
end begin
```

相对于基于 QoS 且具有容错性的任务调度算法的任务调度模型，本文使用了广义 Petri 网来描述任务调度模型，其中(P_{01} ,T)是抑制弧，其作用在于使得模型中的低优先级任务要等到所有高优先级任务调度完了才可以调度。

任务调度模型 RGSPN 描述为：$RGSPN=(p,T,F,K,W,R,M_o,\lambda)$。RGSPN 如图 1 所示。

(1)$p=\{U,PL,PH, P_{01} , P_{02} , P_{03} \}\cup(Q_j,W_j,E_{pj},V_j| j=1,2,\cdots, m\}$。其中，$P_{01}$ 表示未映射的任务；P_{02} 表示按照算法选择的任务；P_{03} 表示需要再次映射的任务；U 表示任务集；PL 表示优先级低的任务；PH 表示优先级高的任务。$\{Q_j|j=1,2,\cdots,m\}$表示计算资源 i 接受该任务的缓冲队列；$\{W_j|j=1,2,\cdots,m\}$表示任务在计算资源 j 上的运行状态；$\{ E_{pj} |j=1,2,\cdots,m\}$表示计算资源 j 发生的过程；$\{ V_j |j=1,2,\cdots,m\}$表示计算资源 j。

(2)$T=\{C,Td,T, T_0 , R_1 ,Tr\}\cup\{ T_{j1} , S_{tj} , E_j , S_j , err_j |j=1,2,\cdots, m\}$。其中，$C$ 表示任务的到来；Td 表示判断任务优先级别；T 表示任务集传送；T_0 表示按照算法从未映射的任务中选择一个任务；R_1 表示任务需要重新调度；Tr 表示修改需要重新调度的任务数据。$\{ T_{j1} | j=1,2,\cdots,m\}$表示判断是否要把任务放在计算资源 j 上执行；$\{ S_{tj} \mid j=1,2,\cdots,m\}$表示计算资源 i 开始处理任务；$\{ S_j |j=1,2,\cdots,m\}$表示任务实施的时间变迁；$\{ E_j | j=1,2,\cdots,m\}$表示故障到来的时间变迁；$\{ err_j | j=1,2,\cdots,m\}$表示故障发生，并重新调度的瞬时变迁。

(3)$F=\{(C, U), (U, Td), (Td, PH), (Td, PL), (PL, T), (T, P_{01}), (PH, T), (T, P_{01}), (P_{01} , R_1), (R_1 , P_{03}), (P_{03} ,Tr), (Tr, P_{01}), (P_{01} , T_0), (T_0 , P_{02})\}\cup\{(P_{02} , T_{j1}),(T_{j1} , Q_j),(Q_j , S_{tj}),(S_{tj} , W_j), (W_j , S_j),(S_j , V_j),(V_j , S_{tj}),(E_j , E_{pj}),(E_{pj} , err_j),(W_j , err_j), (err_j , V_j)(err_j , P_{01})| j=1,2,\cdots,m\}$。

(4)$K(Q_j)= d_j , j=1,2,\cdots,m$(常数 d>1), $K(U)=K(PH)=K(PL)=K(P_{01})=h$(常数 h 远大于 d), $K(P_{03})=K(P_{02})=1$, $K(E_{pj})=K(W_j)=K(V_j)=1$，其中，$1\le j\le m$。

(5)$W(C,U) =X$，其中，$1\le X$。

(6) $M_o (U)=u$, $M_o (P_{01})= M_o (PH)= M_o (PL)= M_o (P_{02})= M_o (P_{03})=0$, $M_o (Q_j)= M_o (E_{pj})= M_o (W_j)=0$, $M_o (V_j)=1$，其中，$1\le j\le m$。

(7)$\lambda=(\lambda_1 , \lambda_2 ,\cdots, \lambda_m)$是平均变迁实施速率集合。

T_0 的实施谓词是：(for $1\le j\le M(P_{01})$ find min Z_i , and for $1\le j\le M(P_{01})$, find min Z,M,T)。

T_{j1} 的实施谓词是：$(M(Q_i)< d_j)\wedge$(for $1<k< d_j$ and $k\ne j$, $Z_j \le Z_k$)$\vee$(for $1<k< d_j$ and $k\ne j$, $M(Q_i)= d_j$)。

R_1 的实施谓词为：(for $1\le j\le m, M_i \le e_{ij}\times P_{ij}$)$\vee$(for $1\le j\le m, T_j \le L_{ij}$)。

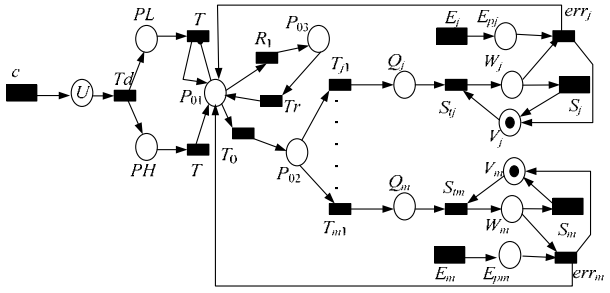


图 1　网格调度的 Petri 网模型 RGSPN

3　可达树的构造

因为故障是不可预测的，所以在不考虑故障的前提下构造可达图。

定义 1 可达图构造算法 $RSG=(V, E1)$步骤如下：

(1)V={M_o}, E1={Φ}, f=0

(2)把 M_o 放在第一层，并把 M_o 标为 new

(3)如果 V 中不存在新的标识，则算法终止，否则继续

(4)选择一个新的标识 M

for 每一个在 M 下如果存在可实施变迁 t_k

获得 t_k 变迁实施后的新标识 M'

if M' ∉ V then

V=V+{M' }

if $t_k=T_0$ then

f=f+1，把 M'标记为 new 并放在 f 层

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

把 M'标记为 new，并和 L_{ij}, min Z_i 一起放在 f 层

else

把 M'标记为 new，并放在 f 层

E1=E1+{M, M'}，用 t_k 标记{M, M'}

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

用“T_{si}/M_{aj}”来标识{M, M'}，T_{si},M_{ai} 分别表示任务和计算资源

if 不存在变迁使得 M[t>，标记 M 为死节点

去除 M 的 new 标记

Go to(3)

定义 2 $E1$ 表示 $RSG(V,E1)$里状态的转换顺序，而计算资源 j 上的任务调度顺序由 T_{si}/M_{aj} 组成。每部分最后一层，状态标记 L_{ij} 表示计算资源上所有任务的完成时间，max{ L_{ij} } 则表示网格系统中一组任务的完成时间。

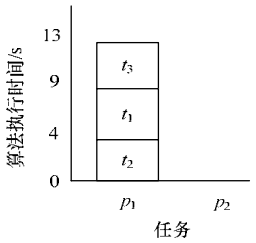
4 应用实例

假设网格中某时有 4 个独立的任务和 3 个计算资源，其运行时间及费用的比值如表 1 所示。

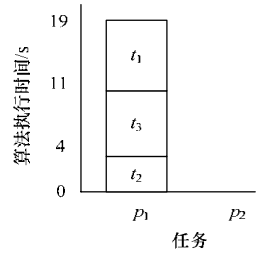
表 1 计算任务运行时间及费用比值

计算任务	执行时间		传输时间		预计费用		要求上限		$W_m:W_t$
	p_1	p_2	p_1	p_2	p_1	p_2	费用	时间	
t_1	5	6	3	7	10	9	18	13	1:2
t_2	2	3	2	2	4	4.5	5	6	1:3
t_3	4	10	3	10	8	15	10	14	1:4

从图 2 中可以知道，基于 QoS 且具有容错性的任务调度算法与 Min-Min 算法相比所花费的时间较少，且能满足 QoS 要求，但该算法与 Min-Min 算法同样具有负载不平衡的特点。



(a)基于QoS具有容错性的任务调度算法



(b)Min-Min算法

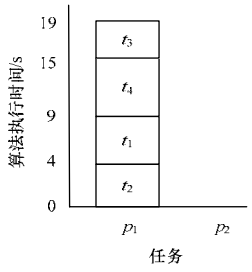
图 2 2 种算法的对比

现假设网格中某时有 4 个独立的任务和 2 个计算资源，其运行时间及费用的比值如表 2 所示。

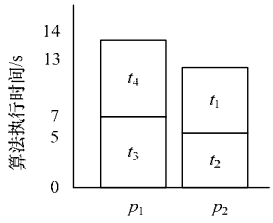
表 2 用户要求任务运行时间及费用比值

计算任务	用户要求	执行时间		传输时间		预计费用		要求上限		$W_m:W_t$
		p_1	p_2	p_1	p_2	p_1	p_2	费用	时间	
t_1	低优先级	5	6	3	7	10	9	18	13	1:2
t_2	低优先级	2	3	2	2	4	4.5	5	6	1:3
t_3	高优先级	4	10	3	9	8	15	10	14	1:4
t_4	高优先级	6	9	4	8	13	12	13	15	1:3

由图 3 可看出，在没有优先级 QoS 的情况下映射策略的完成时间是 19，目标函数平均值约为 47，引入优先级后新映射策略的完成时间为 14，目标函数平均值约为 44。新映射策略要优于增加 QoS 参数之前的影射策略，能较好地实现负载平衡，而且还能完成任务的优先级 QoS 要求。



(a)未引入任务优先级QoS的调度算法



(b)引入任务优先级QoS的调度算法

图 3 引入任务优先级 QoS 前后调度算法的比较

5 结束语

本文改进了网格的资源调度方案，利用广义随机 Petri 网建立了网格资源调度模型，并利用该模型对网格的资源调度过程进行分析。分析结果表明，新策略能使得比较紧迫的任务先行运行，并且使得运行任务的时间和费用的综合花费更少和较好地实现了负载平衡。下一步工作将主要研究一组具有拓扑关系的任务在多 QoS 约束条件下的调度。

参考文献

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

[2] 刘卫东, 宋佳兴, 林 闯. 基于价格时间 Petri 网的网格计算计算模型及分析[J]. 电子学报, 2005, 33(8): 1416-1420.

[3] 陈 晶, 孔令富, 潘 勋. 结合预测机制和 QoS 约束的网格资源调度算法的研究[J]. 计算机研究与发展, 2008, 45(z1): 13-15.

[4] 曹盛勇, 赵瑞芳, 胡志刚. 基于 Petri 网的网格调度模型的研究[J]. 计算机技术与自动化, 2005, 24(4): 123-125.

[5] 林 闯. 随机 Petri 网和系统性能评价[M]. 北京: 清华大学出版社, 2000.

编辑 顾逸斐