基于价格时间 Petri 网的网格计算应用模型及分析

刘卫东, 宋佳兴, 林 闯
(清华大学计算机科学与技术系, 北京 100084)

摘 要: 网格计算环境下资源管理和任务调度是当前研究的热点问题, OGSA 体系结构提出了网格服务的概念, 但缺乏应用对网格服务质量要求的形式化描述. 本文对 OGSA 体系结构进行了必要的扩展, 提出网格用户应用的价格时间 Petri 网模型, 利用该模型可对网格应用的 QoS 需求进行定义, 并可利用 Petri 网的工具分析用户应用模型的正确性和时间特性, 给出了具体的分析算法.

关键词: 网格服务分析; 用户应用模型; 网格服务质量; Petri 网
中图分类号: TP315 文献标识码: A 文章编号: 0372-2112 (2005) 08-1416-05

Modeling and Analysis of Grid Computing Application Based Price Timed Petri Net

LIU Wei-dong, SONG Jia-xing, LIN Chuang
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*)

Abstract: Resource management and task scheduling under Grid environment is a current hot research topic, OGSA (Open Grid Service Architecture) architecture of Grid defines all the resource as service, but it lacks the formal description to QoS of the Grid application. We extend the OGSA architecture and present price timed Petri net concept so the users of Grid can define their applications' QoS requirement using the price timed Petri net model. We also analyse the model's safety, reachability and gives the algorithm to compute the application's total time and total budget.

Key words: grid service analysis; user application modeling; grid service QoS; petri nets

1 引言

随着计算技术和互连网络技术的发展, 以网格为核心的新一代网络计算环境已经成为当前国内外研究的一个热点领域. 网格技术以实现对大规模、分布式自治资源的共享与协同为目标, 而网格资源具有分布性、动态性、异构性和协同性等特征. 因此, 如何组织、管理广域分布、异构、动态的网格资源, 使之协同完成用户的任务, 并满足用户应用的服务质量要求是网格计算的关键问题. 围绕这个问题, 人们在网格体系结构及网格资源的描述、发现、传播、交易等方面进行了研究, 取得了一定的成果[1~4]. 如网格的 OGSA (Open Grrid Service Architecture) 体系结构将所有网格资源封装为一种网格服务, 并定义了网格服务的接口标准和协议[2], 可实现网格的基本功能, 也提供了一定的 QoS 保证机制. 但是, 网格应用本身却缺乏一种形式化的工具来描述网格应用的协同性, 定义网格应用的服务质量要求.

Petri 网是信息处理系统描述和建模的有力工具之一, 它具有并行性、不确定性、异步和分布描述能力和分析能力[5]. 在描述如工作流系统等模型和分析方面, Petri 网显示出其强大的模型描述和性能分析能力[7,8]. 而网格应用可分解成在网格资源上执行的任务的组合, 从这点上说和工作流系统类似, 也常用 Petri 网来描述网格应用[9]. 本文试图在网格的 OGSA 体系结构上, 提出一种基于时间 Petri 网的网格计算应用模型, 并以此为基础, 分析网格计算应用模型的正确性和时间依赖关系的可行性, 使之成为定义网格应用各任务之间依赖关系和 QoS 要求的工具.

2 OGSA 体系结构扩展

为实现网格对分布、异构、动态资源共享和协同操作的目标, I Foster 首先提出了网格的五层沙漏结构, 定义了网格系统的一系列协议. 进而又提出了以服务为核心的 OGSA 体系结构, 将所有网格资源统一定义成网格服务(Grid Service). OGSA 定义了网格服务的接口, 解决了服务发现、动态服务创建、生命周期管理、通知等问题, 为网格服务提供者制定了统一的标准, 也为网格服务的消费者(网格应用)提供了使用服务功能的途径[2]. 但是, 对于用户如何定义其应用的服务质量(QoS)要求以及如何管理和调度服务, 以满足用户的 QoS 要求, OGSA 留在应用层解决.

用户应用对网格服务的质量要求, 一般可以从性能、价格两方面来要求, 分别用任务完成的时间和完成任务所需要服务的成本来衡量. 每一网格服务可以提出各自的服务响应时间以及对应于该服务质量的服务价格, 而用户也可定义他要求的网格应用完成时间以及愿意为之付出的成本. 网格系统应在网格服务的消费者和提供者之间进行撮合, 提出可行的

收稿日期: 2004-07-05; 修回日期: 2005-04-27
基金项目: 国家自然科学基金(No. 90412012, No. 90104002); 国家"973"重点基础研究发展规划项目(No. 2003CB314804)

调度策略. 为此, 可将 OGSA 结构扩展, 如图 1 所示.

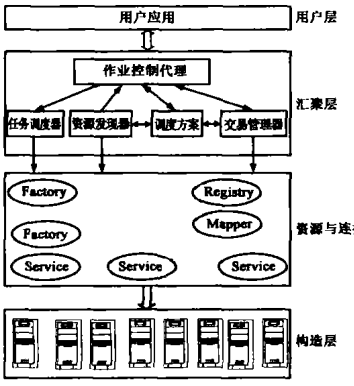

图 1　扩展的 OGSA 结构

从图中可看出, 对 OGSA 的扩展主要体现在称为汇聚层的网格资源管理上. 它的主要功能有两方面, 一是让用户定义其应用, 将应用分解成不同的任务 (Task) 和任务之间的调用关系, 并可定义应用的 QoS 需求; 二是根据用户定义, 通过与底层的网格服务进行谈判和协商, 选择合适的网格服务来完成各任务, 并在合适的时候完成. 因此, 该层包含有以下几个组成部分:

• 作业控制代理　作业控制代理是与用户层的接口. 它响应用户层的请求, 将网格中能提供的服务功能提供给用户, 并建立用户应用的任务模型和 QoS 需求.

• 资源发现器　接受网格服务的注册, 并将相同功能的网格服务合并到一个目录中, 供调度时选择.

• 交易管理器　与网格服务就服务质量进行协商和谈判, 以期能达到更优.

• 调度方案　根据用户应用模型和 QoS 需求, 选择对应的网格服务来完成应用, 提供调度方案.

• 任务调度器　根据生成的调度方案, 完成能满足 QoS 需求的调度, 最终完成用户应用.

采用本结构后, 用户层不再直接面对网格服务层, 且将应用分解成任务, 调度工作转变为为每个任务发现符合 QoS 需求的服务, 结构上十分清晰. 下一步的工作, 就是要建立一个模型, 来描述用户应用, 主要是各任务之间的时间依赖关系及以时间和价格为代表的服务质量要求.

3　网格服务模型

3.1　价格时间 Petri 网

对网格体系结构进行扩展后, 用户可利用汇聚层的作业控制代理模块, 将其应用定义为任务的集合, 这些任务之间, 存在着诸如顺序、分支、循环、并行等的依赖关系. 而且, 用户对其应用在网格环境下的完成存在 QoS 需求, 如完成应用的总时间以及完成应用的成本等. 以上这些, 都需要有一个形式化的工具来为用户应用建立模型, 以方便应用的描述, 并以此为基础, 调度网格中的服务, 来完成用户应用. 下面, 我们提出价格时间 Petri 网的定义, 并利用它为工具来描述用户应用的任务之间的依赖关系和 QoS 需求.

在文献[5]中, Petri 网的形式化定义为,

定义 **1**　Petri 网

Petri 网是一个三元组 $PN=(S, T, F)$, 其中

(1) $S\cap T=\Phi$　(二元性);

(2) $F\subseteq(S\times T)\cup(T\times S)$ (流关系仅存在于 S 和 T 的元素之间);

(3) $S\cup T\neq\Phi$　(网非空);

(4) $dom(F)\cup cod(F)=S\cup T$　(没有孤立元素).

为将时间因素引入到网中, 后来又提出了时间 Petri 网概念[6].

定义 **2**　时间 Petri 网

时间 Petri 网 $TPN=(PN, D, \Gamma)$, 其中

(1) PN 是 Petri 网;

(2) $D=\{d_j \mid j=1, \cdots, m\}$, $d_j\in R^+\cup\{0\}$, 表示完成每个变迁 t_j 所需要的时间;

(3) Γ 是一个集合, 其元素 (t_j, τ_k) 表示变迁的 t_j 实施时刻为 τ_k.

时间 Petri 网给出了每个变迁所需要的时间成本, 但并没有考虑其价格成本. 我们可在时间 Petri 网基础上, 定义价格时间 Petri 网 $CTPN$, 为每个变迁赋予一个价格(或预算).

定义 **3**　价格时间 Petri 网

价格时间 Petri 网 $CTPN=(PN, D, \Gamma, C)$, 其中

(1) PN、D 和 Γ 同定义 2;

(2) $C=\{c_j \mid j=1, \cdots, m\}$, $c_j\in R^+\cup\{0\}$, 表示完成每个变迁 t_j 所需要的成本.

3.2　用户应用的价格时间 **Petri** 网模型

一个任务 (*Task*) 可以表示为一个基本的价格时间 Petri 网, 它由一个源位置 i, 一个变迁 t 和一个汇位置 o 组成. 变迁 t 的实施起始时刻为 τ, 需要时间 d, 成本为 c. 如图 2 所示.

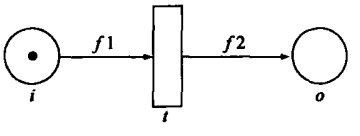

图 2　任务的基本价格时间 Petri 网表示

所以, *Task* 可以用价格时间 Petri 网形式化表示为:

$Task=(PN, D, \Gamma, C)$, 其中

$PN=(\{i, o\}, \{t\}, \{f1, f2\})$;

$D=\{d\}$;

$\Gamma=\{(t, \tau)\}$;

$C=\{c\}$.

用户可通过给定 d、c 和 τ, 来定义他对任务的 QoS 需求.

用户应用由基本的 *Task* 组成, *Task* 之间存在着诸如顺序、分支、循环、并行等关系. 我们可以分别构造出这些关系的组合后的价格时间 Petri 网模型, 通过这些组合, 构成完整的用户应用的价格时间 Petri 网模型.

用户应用 *UA* 可以用类 *BNF* 表示如下:

$UA ::= \varepsilon \mid X \mid T \mid T\diamond T \mid T\odot T \mid T\oplus T \mid \mu T$　其中,

ε 代表空任务;

X 代表一个任务常量, 需要固定的时间和成本来完成. 这两个任务也是为保证系统的完整性而引入的.

$T_1\diamond T_2$: 代表两个任务顺序执行; $T_1\odot T_2$: 代表 $T1$ 和 $T2$ 两个任务为分支执行, 一旦执行了其中的一个, 则不执行另一个; $T_1\oplus T_2$: 代表 $T1$ 和 $T2$ 两个任务为并行任务, $T1$ 和 $T2$ 应并行调度执行;

μT: 表示循环执行 T, 一共 μ 次.

下面, 我们逐一给出它们的价格时间 Petri 网模型. 我们定义参加组合的任务为 $T1$ 和 $T2$, $Ti=(PNi, Di, \Gamma i, Ci)$, 其中,

$PNi=(Pi, Ti, Fi)$; $D=\{di\}$; $\Gamma=\{(ti, \tau i)\}$;

$C=\{ci\}$, $i=1, 2$.

定义 **4** 空任务 ε

空任务 $\varepsilon=(PN, D, \Gamma, C)$, 其中

$PN=(\{i\}, \Phi, \Phi)$;

$D=\Phi$; $\Gamma=\Phi$; $C=\Phi$.

从图形上看, 空任务可以认为是一个只有一个位置的 Petri 网.

定义 **5** 常量任务 X

常量任务 $X=(PN, D, \Gamma, C)$, 其中

$PN=(\Phi, \{t\}, \Phi)$;

$D=\{d\}$; $\Gamma=\{(t, \tau)\}$; $C=\{c\}$.

从图形上看, 常量任务可以认为是一个只有一个变迁的 Petri 网.

定义 **6** 顺序执行任务 $T=T_1 \diamond T_2$

顺序执行任务 $T=(PN, D, \Gamma, C)$, 其中

$PN=(P, T, F)$, 而

$P=P1 \cup P2$,

$T=T1 \cup T2 \cup \{t\}$,

$F=F1 \cup F2 \cup \{o1, t\} \cup \{t, i2\}$.

$D=\{d1, d2, d\}$, 而完成变迁的总时间为 $d1+d2+d$;

$\Gamma=\Gamma1 \cup \Gamma2 \cup \{(t, \tau)\}$;

$C=\{c1, c2, c\}$, 而完成变迁的总成本为 $c1+c2+c$.

从图形上看, 顺序执行的两个任务可以用图 3 表示.

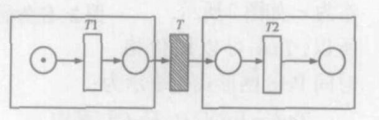

图 3 顺序执行任务模型

定义 **7** 分支执行任务 $T=T_1 \otimes T_2$

分支执行任务 $T=(PN, D, \Gamma, C)$, 其中

$PN=(P, T, F)$, 而

$P=P1 \cup P2 \cup \{i, o\}$,

$T=T1 \cup T2 \cup \{ti1, to1, ti2, to2\}$,

$F=F1 \cup F2 \cup \{(i, ti1), (i, ti2), (ti1, i1), (ti2, i2), (o1, to1), (o2, to2), (to1, o), (to2, o)\}$.

$D=\{d1, d2, d_{ti1}, d_{ti2}, d_{to1}, d_{to2}\}$, 而完成变迁的总时间为 $d_{ti1}+d1+d_{to1}$ ($T1$ 实施); 或者是 $d_{ti2}+d2+d_{to2}$ ($T2$ 实施);

$\Gamma=\Gamma1 \cup \Gamma2 \cup \{(ti1, \tau i1), (ti2, \tau i2), (to1, \tau o1), (to2, \tau o2)\}$;

$C=\{c1, c2, c_{ti1}, c_{ti2}, c_{to1}, c_{to2}\}$, 而完成变迁的总成本为 $c_{ti1}+c1+c_{to1}$ ($T1$ 实施); 或者是 $c_{ti2}+c2+c_{to2}$ ($T2$ 实施).

从图形上看, 分支执行的两个任务可以用图 4 表示.

定义 **8** 并行执行任务 $T=T_1 \oplus T_2$

并行执行任务 $T=(PN, D, \Gamma, C)$, 其中

$PN=(P, T, F)$, 而

$P=P1 \cup P2 \cup \{i, o\}$,

$T=T1 \cup T2 \cup \{ti, to\}$,

$F=F1 \cup F2 \cup \{(i, ti), (ti, i1), (ti, i2), (o1, to), (o2, to), (to, o)\}$.

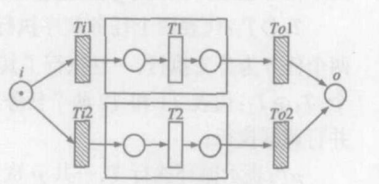

图 4 分支执行任务模型

$D=\{d1, d2, d_{ti}, d_{to}\}$, 而完成变迁的总时间为 $d_{ti}+\max(d1, d2)+d_{to}$;

$\Gamma=\Gamma1 \cup \Gamma2 \cup \{(ti, \tau i), (to, \tau o)\}$;

$C=\{c1, c2, c_{ti}, c_{to}\}$, 而完成变迁的总成本为 $c_{ti}+c1+c2+c_{to}$.

从图形上看, 并行执行的两个任务可以用图 5 表示.

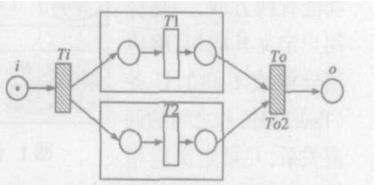

图 5 并行执行任务模型

定义 **9** 循环执行任务 $T=\mu T$

循环执行任务 $T=(PN, D, \Gamma, C)$, 其中

$PN=(P, T, F)$, 而

$P=P \cup \{i, o\}$, $T=T \cup \{ti, to, tu\}$,

$F=F \cup \{(i, ti), (ti, i1), (o1, to), (to, o), (o1, tu), (tu, i1)\}$.

$D=\{d1, d_u, d_{ti}, d_{to}\}$, 而完成变迁的总时间为 $d_{ti}+\mu(d1+d_u)+d_{to}$;

$\Gamma=\Gamma \cup \{(ti, \tau i), (to, \tau o), (tu, \tau u)\}$;

$C=\{c1, c_u, c_{ti}, c_{to}\}$, 而完成变迁的总成本为 $c_{ti}+\mu(c1+c_u)+c_{to}$.

从图形上看, 循环执行的任务关系可以用图 6 表示.

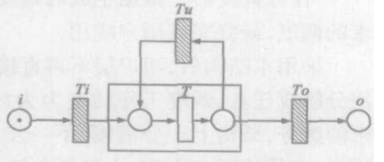

图 6 循环执行任务模型

上面, 我们分别给出了任务之间不同依赖关系的价格时间 Petri 网模型构建方法. 对于一般的用户应用, 可以采用下面的算法构建整个应用的价格时间 Petri 网模型.

算法 **1** 构建用户应用的价格时间 Petri 网模型

Step1: 为用户应用的每个任务建立基本的价格时间 Petri 网模型.

Step2: 对任务之间的依赖关系, 根据上述定义 4 至定义 9, 分别构建新的价格时间 Petri 网模型.

Step3: 任务间所有关系处理完毕后, 得到的价格时间 Petri 网模型即为用户应用的模型, 可计算出整个用户应用的整体时间要求和成本要求.

4 模型分析

建立起用户应用的价格时间 Petri 网模型后, 我们可以通过对模型进行分析, 来检验用户应用模型的正确性. 如通过检查模型的一致性和可达性, 确定用户应用的逻辑正确性; 也可通过检查各任务之间的时间关系, 确定应用的时间限定条件是否满足.

4.1 一致性分析

由于用户应用的模型是用户自行定义的, 难免由于各种原因, 造成任务之间调用关系的不一致现象, 最终使用户应

用模型的结果不能达到要求. 其中, 最主要的可能就是任务的死锁, 也就是说, 模型中的某些任务可能根本就无法得到实施. 我们可以通过 Petri 网的分析工具对模型进行分析, 来消除死锁的现象.

检查用户应用中是否存在死锁, 可通过检查其对应的 Petri 网模型中是否存在 Siphon 来进行, 而这可通过解下述方程来完成[5].

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}=0 \qquad (1)$$

其中, $\boldsymbol{A}$ 为模型的关联矩阵, $\boldsymbol{X}$ 为一个非负整数的集合, 称为模型的 S- 不变量. 如果方程(1)存在非负解, 且解〈x〉确为一 Siphon(〈x〉中不存在标识), 则该用户应用存在死锁.

证明过程可参考文[5].

4.2　安全性分析

安全性是用户应用分析的一个重要特征. 所谓安全性, 一般是指用户应用不会处于非正常状态. 对网格应用, 安全性是指其在完成其规定的任务后可在定义的终止状态结束. 因此, 构造完网格应用的价格时间 Petri 网模型后, 安全性实际上是指其对应的 Petri 网从其唯一的源位置到汇位置的可达性. 文[5]把可达性定义为:

定义 **10**　PN 的可达性

若存在一个实施序列, 可将 PN 网的标识从 M_0 转换为 M, 则标识 M 称为从标识 M_0 可达.

定理 **1**　可达性必要条件

若表示 M_d 从 M_0 可达, 则存在一个非负向量 $\boldsymbol{X}$, 使

$$M_d=M_0+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X} \qquad (2)$$

根据定理 1, 我们可以设计算法 2, 通过求 $\boldsymbol{X}$ 来判断从初始状态 M_0 到最终状态 M_d 是否不可达. 从基本的 *Task* 模型和用户应用模型的构建方法可知, 初始状态 M_0 和终止状态 M_d 均为一个 n 维向量, M_0 的 i 位置的向量值为 1, 其他位置值均为 0; 而 M_d 的 o 位置的向量值为 1, 其他位置值均为 0. 若不存在从 M_0 到 M_d 的实施序列, 则用户应用模型中存在错误.

算法 **2**　不可达判别

Step1: 给出用户应用的价格时间 Petri 网模型;

Step2: 给初始状态 M_0 和终止状态 M_d 赋值;

Step3: 解方程 $M_d=M_0+\boldsymbol{A}^{\mathrm{T}}\boldsymbol{X}$;

Step4: 若 $\boldsymbol{X}$ 不为非负整数向量, 则从 M_0 到 M_d 不存在可达的实施序列.

如果不存在非负整数解, 则从 M_0 到 M_d 无可实施序列, 需要检查用户应用模型的正确性.

如果存在非负整数解, 则可能存在从 M_0 到 M_d 可实施序列, 这样, 可以在调度时再求解具体的实施序列.

4.3　时间和价格特性分析

时间和价格特性分析, 是指在用户定义好用户应用模型, 并给出他在每个任务上可承受的价格和时间成本后, 可通过对模型的计算, 得出整个用户应用的总时间和总价格成本. 这样, 可在总体预算和时间的限度下, 来对用户应用进行调度.

定义 **11**　实施序列

令 $PN=(S, T, F)$ 是一个 Petri 网, $\sigma=M_0t_1M_1t_2\cdots t_nM_n$ 是 PN 的一个有限实施序列 iff $\forall i, 1\leqslant i\leqslant n; M_{i-1}[t_i>M_i; t_1t_2\cdots t_n]$; 叫做变迁实施序列.

变迁实施序列可通过可达树等算法求解. 得到实施序列后, 我们可以用算法 3 得到整个用户应用在不同实施序列的最短实施时间和实施成本.

算法 **3**　计算用户应用的总时间和成本

Step1: 根据可达树算法, 求 M_0 到 M_d 可实施序列{σ};

Step2: /* 求每个可实施序列的实施时间和实施成本* /

```
Count = 0;
While ∀σ∈{σ} Do
  {Totaltime[count] = 0;
    Totalbudget[count] = 0;
    n = |σ|;
    While i < = n Do
    if(t[i]与 t[i-1]不是并行关系)
    then{Totaltime[count] + = t[i].d;
      Totalbudget[count] + = t[i].c};
    else{MaxParellTime = 0;
  while i < = n and t[i]与 t[i-1]不是并行关系 Do
        {Totalbudget[count] + = t[i].c;
          i++ ;
          if t[i].d > MaxParellTime then MaxParellTime =
t[i].d;
        }/* End of Parell transaction * /
    Totaltime[count] + = MaxParellTime;
      }/* End of else * /
    i++ ;}
  count++ ;}
```

Step3: 返回所有实施序列的总时间 Totaltime 和总成本 Totalbudget.

算法 3 的核心是将并行执行的任务的时间取其中的最大值, 而其价格则照常计算. 计算出的每个实施序列最短实施时间和成本, 可作为调度算法的依据之一.

4.4　QoS 满足性分析

根据用户应用的价格时间 Petri 网模型, 网格中的交易管理器将就模型中的每个任务, 通过资源发现器提供的资源信息, 与服务提供者进行协商, 以发现其中最能满足任务功能要求和 QoS 需求的服务, 形成调度方案, 供任务调度器进行调度时使用. 我们可以在用户应用和任务两个层次上定义服务的 QoS 满足性.

4.4.1　应用级 **QoS** 满足性　应用级 QoS 满足性, 主要是从整个应用的层面上, 来考察网格提供的服务是否满足用户定义的 QoS 要求. 4.3 节中, 我们已经计算出用户应用的总时间要求 Totaltime 和总成本要求 Totalbudget.

与此类似, 交易管理器在形成调度方案后, 也就是说, 为应用的每个任务找到匹配的网格服务后, 根据每个网格服务承诺的服务完成时间和价格, 也可用算法 3, 计算出实现用户应用的总服务时间 $\mathrm{Totaltime}_s$ 和总服务价格 $\mathrm{Totalprice}_s$.

定义 **12** 应用级 QoS 满足性令用户应用的价格时间 Petri 网模型为 UA, 其定义的应用的总时间为 $\mathrm{Totaltime}_t$, 总成本为 $\mathrm{Totalbudget}_t$. 用户应用的一个调度方案为 S, 其总服务时间 $\mathrm{Totaltime}_s$ 和总服务价格 $\mathrm{Totalprice}_s$. 若

$$\mathrm{Totaltime}_s \leqslant \mathrm{Totaltime}_t \tag{3}$$

且

$$\mathrm{Totalprice}_s \leqslant \mathrm{Totalbudget}_t \tag{4}$$

则我们说调度方案 S 在应用级满足 UA 的 QoS 要求.

在满足 QoS 要求的情况下, 采用不同的调度策略, 可能得到不同的调度方案. 如在不超出时间要求的情况下, 使总价格最低; 或者是在不超过总成本预算的情况下, 使应用的完成时间最短等. 当然, 也可以将时间和价格组合成一个整体的评价函数, 使得调度方案在评价函数下的结果最优.

4.4.2 任务级 **QoS** 满足性 应用级 QoS 满足性定义了对整个应用在价格和时间两方面的 QoS 满足要求. 但是, 由于网格服务的分布性, 单纯考察应用级 QoS 满足性对交易管理器形成调度方案增加了很大的复杂性. 因此, 有必要考虑任务级 QoS 满足性, 以简化调度方案的形成过程.

定义 **13** 任务级 QoS 满足性

令用户应用中的任何任务 t, 其在调度方案 S 中对应的网格服务为 s. s 的时间 d_s, 价格 c_s. 若

$$d_s \leqslant d_t \tag{5}$$

且

$$c_s \leqslant c_t \tag{6}$$

则我们说调度方案 S 在任务级满足 UA 的 QoS 要求.

任务级 QoS 满足性将应用的价格和时间 QoS 要求, 分解到每一个任务, 这样, 调度时的方案形成将相对简单, 但不同方案之间的比较, 还需要在一个统一的评价函数下, 通过考察整个应用的性能来进行.

5 结论和进一步工作

本文提出了一种价格时间 Petri 网模型, 来描述网格环境下用户应用的 QoS 需求, 给出了模型正确性和可达性的分析方法, 并给出了整个用户应用的总价格和总时间的计算算法, 提出了应用级 QoS 满足性和任务级 QoS 满足性的定义, 为进一步研究网格资源管理和任务调度提供了基础. 下一步研究工作将围绕网格服务的调度展开, 根据用户应用的价格时间 Petri 网模型, 提出对 QoS 的评价函数, 并根据评价函数对网格服务的调度算法进行评价. 同时, 对单个的网格节点来说, 由于可能存在多个用户应用调用其提供的网格服务, 如何进行服务调度, 以保证提供承诺的服务质量, 也是进一步研究的一个方向.

参考文献:

[1] I Foster, C Kesselman, S Tuecke. The anatomy of the grid: Enabling scalable virtual organizations[J]. Int' l J Highr Performance computing Applications, 2001, 15(3): 200– 222.

[2] Ian Foster, Carl Kesselman, Jeffrey M Nick, Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed System Integration [R/OL]. http://www.globus.org/research/papers/ogsa.pdf, 2002– 06.

[3] K Czajkowski, S Fitzgerald, I Foster, C kesselman. Grid information services for distributed resource sharing[A]. High Performance Distributed Computing, 2001 Proceeding, 10th IEEE International Symposium on[C]. Washington, DC, USA: IEEE Computer society, 2001, (7 – 9): 181– 194.

[4] Ian Foster, Carl Kesselman, Jeffrey M Nick, Steven Tuecke. Grids services for distributed system integration[J]. Computer, 2002, 35(6): 37 – 46.

[5] Tadao Murata. Petri nets: Properties, analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4): 541– 580.

[6] Lopez-Mellado E. Simulation of timed petri net models[A]. Systems, Man and Cybernetics, IEEE International Conference on[C]. Vancouver, BC, Canada, Oct. 1995. 2270– 2273.

[7] N R Adam, V Atluri, W Huang. Modeling and analysis of workflow using Petri nets[J]. Journal of Intelligent Information Systems, 1998, 10: 131– 158.

[8] 刘婷, 林闯, 刘卫东. 基于时间 Petri 网的工作流系统模型的线性推理[J]. 电子学报, 2002, (30)2: . 245– 248.
Liu Ting, Lin Chuang, Liu Weidong. Linear temporal inference of workflow management system based on timed Petri net model[J]. Acta Electronica Sinica, 2002, 30(2): 245– 248(In Chinese).

[9] J Cao, S A Jarvis, S Saini, G R Nudd. GridFlow: Workflow management for grid computing[A]. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003)[C]. Tokyo, 2003.

作者简介:

刘卫东 男, 1968 年生于江西省分宜县, 博士研究生, 副教授, 主要研究方向为计算机网络及应用, 网络计算、网络服务质量、工作流等. E-mail: liuwd@tsinghua.edu.cn.

宋佳兴 男, 1974 年生于吉林省农安县, 博士研究生, 助理研究员, 主要研究方向为计算机网络及应用, 网络计算、P2P 网络等.

林 闯 男, 1948 年生于辽宁省, 现任清华大学计算机科学与技术系主任、教授、博士生导师, 主要研究领域为计算机网络, 系统性能评价, 随机 Petri 网, 逻辑推演和推理系统, 主持多项 973、863、国家自然科学基金和国际合作科研项目, 在国内外一级期刊和会议上发表论文 180 多篇, 出版专著 3 本.