



基于 QoS 的超级节点模式网格调度研究 *

潘善亮¹, 黄 希¹, 茅琴娇²

(1. 宁波大学信息科学与工程学院 宁波 315211; 2. 西安交通大学电子与信息工程学院 西安 710049)

摘 要: 提出一种基于超级节点模式的网格管理模型, 引入市场经济机制, 允许网格用户提出任务的截止完成时间、费用上限以及时间、费用偏好参数, 并作为用户 QoS 参数给出对应的网格复合任务调度算法。根据此算法, 利用层次颜色 Petri 网、价格时延和颜色 Petri 网对调度过程进行形式化动态建模。最后, 构建模型的可达任务图, 通过实例分析系统的最佳调度方案及其吞吐量、负载平衡、时间和费用等性能参数。

关键词: 网格调度; QoS 参数; Petri 网

doi: 10.3969/j.issn.1000-0801.2014.02.003

Research on Grid Scheduling of Super-Peer Model Based on QoS

Pan Shanliang¹, Huang Xi¹, Mao Qinjiao²

(1. College of Information Science and Engineering, Ningbo University, Ningbo 315211, China;

2. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Firstly, a kind of grid resource management model——super-peer model was proposed. For this model, economic mechanism was introduced, allowing a grid user to put forward the task's expected execution deadline, cost limit and the weight parameters between them. Those are the QoS parameters of application-level. Then, a corresponding resource scheduling algorithm was shown. Next, hierarchical colored Petri net and price timed Petri net were used to model and the dynamic process of grid resource scheduling was simulated. Lastly, reachable task graph (RTG) of Petri net model was constructed and the system's scheduling performance was analyzed, such as throughout capacity, load-balancing, scheduling time and cost through an example.

Key words: grid scheduling, QoS parameter, Petri net

1 引言

网络计算^[1]环境下, 有效的资源管理方案及资源调度算法对优化资源的使用和资源调度效率的提高有着十分重要的作用。近年来提出的基于超级节点模式(super-peer model)的网格资源管理模式^[2]在集中式搜索与扩展性强的、具有负载平衡以及容错性的分布式搜索之间实现了平衡, 并为位于不同区域的网格物理组织之间进行互联提供了一个很好的框架。因此, 基于超级节点模式的网格计算

在提出后便成为研究的热点。超级节点网络与传统的 P2P 网络相似, 在传统 P2P 网络的基础上, 把每一个 P2P 节点变为一个超级节点, 每一个超级节点作为服务器与一系列的客户端机相连。这些超级节点之间则采用 P2P 方式在更高层次上相互连接。面向服务的开放网格服务体系结构(OGSA)^[3]和 Web 服务资源框架(WSRF)^[4]为网格与 P2P 的集成提供了一个框架, 使不同区域的服务器可以使用 Web 服务实现 P2P 互连。

目前, 国内外学者对超级节点模式网格计算的研究主

* 浙江省自然科学基金资助项目(No.LY12F02020), 宁波市自然科学基金资助项目(No.2012A610066)



要集中在如下 3 个方面:

- 资源的发现与定位;
- 超级节点之间的拓扑连接协议与体系结构;
- 超级节点的选择方法,如 Carlo M 等^[5]对超级节点模式的网格以及层次式网格、分布式纯 P2P 模式网格进行了性能仿真、分析和对比,得出超级节点模式网格的优点,还讲述了超级节点模式如何实现多机构网格资源管理中进行的服务发现^[2];Kwan S K 等^[6]采用一种 gossip 协议交换超级节点之间的资源信息,大大提高了资源的发现效率;Peter M 等人^[7]针对个人计算机 P2P 网络提出了一个能自行组织和管理的超级节点网络;Yin L 等人^[8]利用经常在线的节点作为超级节点来管理那些经常离开的不稳定节点,这些稳定的节点之间构成 DHT(distribute hash table)环;Pasquale C 等人^[9]提出了一个超级节点模式系统结构的网格,可以对庞大的数据量进行分析;Wu C C 等人^[10]提出了一种网格架构 G2G 以实现自治网格之间的互联;Sheng H Z 等人^[11]对超级节点的选择进行了研究,通过评估超级节点的计算能力,提出了一个超级节点的选择策略。

Petri 网^[12]具有并发、异步、动态等特点,已成为模拟和分析信息系统的有力工具,在资源调度中得到了广泛研究和应用,但还有一些问题值得探讨,如参考文献[13]中利用 Petri 网对超级节点模式的网格调度进行形式化建模与理论分析,将超级节点模式的网格任务、资源节点和时间限制映射到层次、颜色、时延 Petri 网,但所建立的模型只考虑了时间维度的服务属性,其调度对象是简单的不可分割的独立任务,没有考虑网格资源的动态性等机制。

目前 Petri 网调度建模机制在超级节点模式网格计算方面的应用研究还存在如下几个方面的不足。

- 资源分配和任务调度策略大部分以系统为中心,只考虑了服务时间维度的服务属性参数,较少考虑面向用户 QoS 的要求(如截止时间、费用上限以及二者之间的权衡参数)和用户需求的多样性,导致资源分配无效和不公平的情况发生。
- 超级节点模式的网格调度是针对相互独立的不可分割的任务进行的,而网格任务通常是可以分解的复合任务,因此有必要建立复合任务的调度机制,以优化调度策略。
- 没有考虑网格资源的动态性,当资源出现故障时

重调度机制考虑不足。

本文重点研究面向用户 QoS 参数(时间、价格以及二者之间的权重)要求的超级节点模式网格复合任务调度算法,分析复合任务的调度过程,给出系统的最佳调度方案和重调度机制,对任务的调度过程进行价格时延 Petri 网形式化建模与理论分析,以验证算法的有效性,为系统性能评价提供依据。

2 超级节点模式的网格资源管理模型

本文讨论的超级节点模型是如图 1 所示的分层体系结构。

依据节点的非功能参数,把性能评估参数较高的节点(以下称为超级节点)作为上层管理,构成系统中的骨干层。每个超级节点作为服务器与一系列由不同组织管理的客户机一起构成一个传统的网格模式,而这些上层超级节点采用 P2P 通信方式构成一个大规模网格。每个超级节点控制着一组本地计算机资源的访问权限,代表着一个超级管理域,称为网格社区。一般地,当网格社区内的用户节点需要访问网格时,超级节点在它的管理域内以传统网格集中式搜索方式查找匹配的资源,如果没有找到,则通过 P2P 方式在其他超级节点之间进行查询^[14,15]。由此,引入如下定义。

定义 1 $peer-to-peer\ grid = \{super-peer_1, super-peer_2, \dots, super-peer_n\}$, 其中, $super-peer_1, super-peer_2, \dots, super-peer_n$ 是 $grid\ community_1, grid\ community_2, \dots, grid\ community_n$ 所对应的网格虚拟社区。

定义 2 $super-peer_i = \{node_1, node_2, \dots, node_m\}$, 其中 $node_1, node_2, \dots, node_m$ 表示具有自主能力的主机或其他资源。

定义 3 $node_i = (ID_i, ID_{super-peer_i})$ 表示节点的属性, ID_i 、 $ID_{super-peer_i}$ 分别表示节点标识符和节点所在的超级节点网格社区标识符。

3 基于市场经济体制的超级节点模式网格复合任务调度算法

3.1 复合任务调度模型

在网格环境中进行资源管理和调度是个非常复杂的问题。在网格系统中,大量地理上分布的各种资源为不同的组织所拥有,这些组织具有不同的使用规则、计费模型、负荷能力和使用模型,资源拥有者和资源使用者各自具有不同的目标、目的、策略和需求,一些传统的资源管理和调

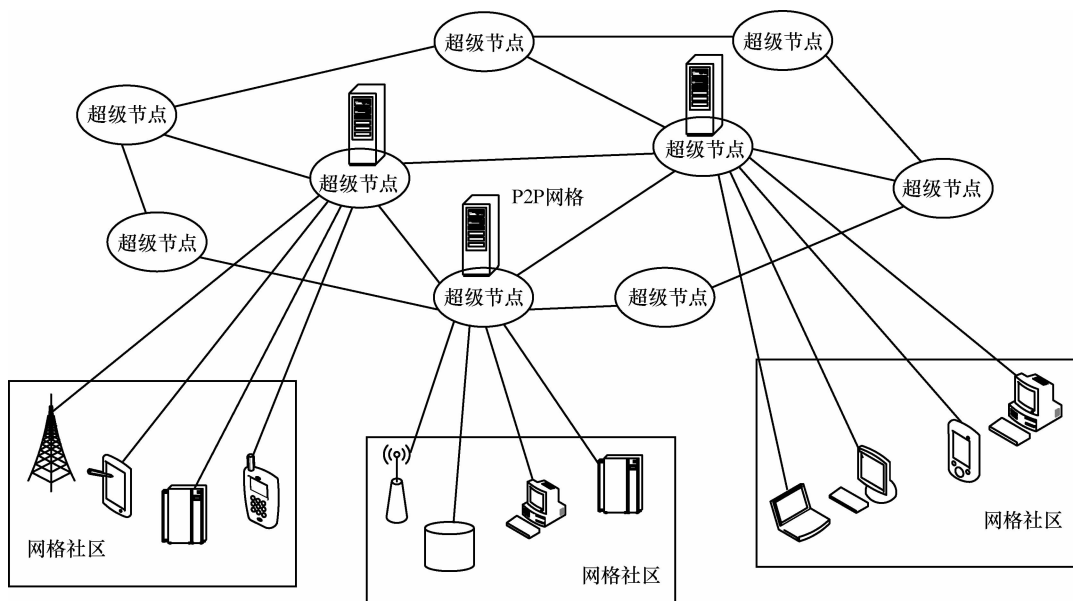


图1 网络计算超级节点模式的两层资源管理体系结构

度方式在网格系统中并不适用,而基于计算经济模型的网格任务调度方案是一个很好的解决方案。

基于市场经济的超级节点模式的网格复合任务调度模型如图2所示。在模型中,超级节点具有代理^[6]功能,用户直接在本地计算机(源节点)提交任务,并规定任务的截止完成时间和费用上限以及时间和费用的偏好参数,任务被传输到本地超级节点,称这个超级节点为源超级节点。超级节点判断该任务是否能分解成子任务,如果能,则将任务分解成具有时间依赖关系的子任务集合,同时将时间、费用总要求分解,形成每个子任务的具体要求,并建立调度列表,然后按照调度列表选取并判断任务(子任务)能否在超级节点社区内提交它的资源节点并在截止时间内完成,如果能,则直接调度到它上面执行。

这里做出如下规定:

- 每个资源节点在对应的超级节点处有一个用户账户,记录费用的使用状况,由超级节点负责管理;
- 超级节点根据它所管理的资源利用状况给出对于具体任务的估计执行时间和费用;
- 任务(子任务)在资源节点上执行时,不产生网格费用消耗。

相反地,当资源节点执行由其他节点(包括本网格社区内的其他资源节点和远程超级节点管理域内的资源节点)提交的任务(子任务)时,将收取一定的网格费用。因此,超级节点优先将任务(子任务)调度到提交它的资源节点上执行。若不行,则考虑将任务(子任务)调度到本地网格社区内满足用户要求的其他某一最优节点上执行。这一

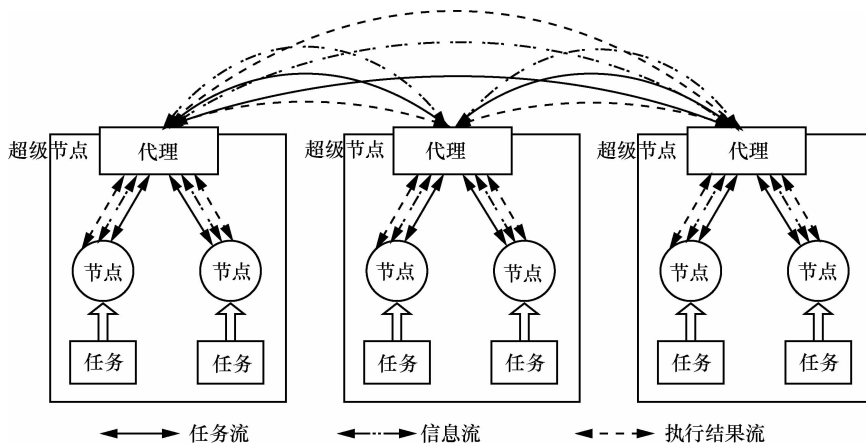


图2 超级节点模式的网格任务调度模型



阶段,超级节点对其管理域内的本地资源节点采用集中式调度并进行内部资源节点之间的转账交易,这种优先本地任务策略能够减少网络阻塞。

若任务(子任务)不能在超级节点社区内完成,超级节点将会与远程超级节点进行通信,以寻找符合要求的某一最优资源节点执行任务(子任务),并为之代付一定的费用。每个超级节点都有一张远程超级节点列表,并且可与它们交互,或存在一个中心目录来维护与每个远程超级节点相关的信息。这一阶段,超级节点之间采用分布式调度。

对于复合任务,以上过程不断地重复,直至调度列表中所有子任务都执行完毕。只有当子任务的所有前驱子任务都执行完毕后,该子任务才拥有被调度权。对于每一个任务(子任务),当执行它的资源节点发生故障时,携带错误报告的执行结果被传送到源超级节点,源超级节点进行必要的重调度,并撤销原来的相关交易。

基于以上调度模型,网格计算超级节点模式的超级节点代理框架如图3所示,其中GKD是global knowledge database的缩写,LKD是local knowledge database的缩写。

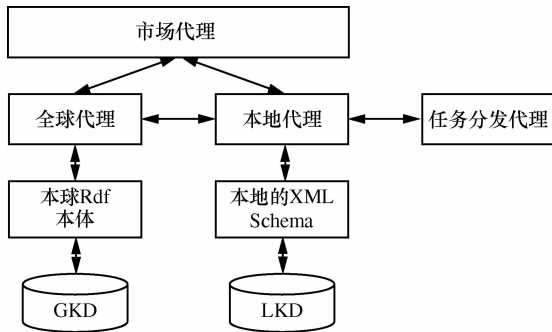


图3 超级节点代理的架构

超级节点代理架构包括4个分代理:任务分发代理(task-dividing agent)、本地代理(local agent)、全球代理(global agent)、市场代理(market agent),各分代理的功能职责介绍如下。

- 任务分发代理负责将本地资源节点上提交的网格应用分解成具有相互依赖关系的子任务集合(如果网格应用为元任务(meta task),则不用分解),建立调度列表,并将准备调度的任务交给本地代理。
- 本地代理收到调度请求后,优先将任务(子任务)调度到源节点。如果不行,则在本地网格资源社区内查找相关的资源信息,若找到符合要求的资源,则联系市场代理为任务(子任务)选择最优的资源进

行调度,并由市场代理进行交易管理。否则,把调度请求发送给全球代理。

- 全球代理负责两方面的职责:一方面是将本地的XML Schema转换为全球的Rdf本体,存储在GKD中;另一方面是接收本地代理发送过来的调度请求,与其他超级节点代理中的全球代理进行交互,寻找合适的资源并联系市场代理,为任务(子任务)选择最优的资源进行调度,并由市场代理完成超级节点间的交易管理。全球代理准确地知道其他超级节点域的所有本体,并且含有相关本体的概要信息。

3.2 复合任务调度算法

3.2.1 基于时间和价格的多目标线性规划评价模型

定义如下参数: t_{ij} 为任务(子任务) i 在资源节点 j 上的估计执行时间, T 为任务(子任务)的截止期限与到达时间之差, c_{ij} 为任务(子任务) i 在资源节点 j 上的估计执行费用, C 为任务(子任务)的费用上限, $\alpha+\beta=1, \alpha \geq 0, \beta \geq 0$ 。 α, β 为用户定义的时间和费用权重。

定义4 在截止期限和费用上限均能满足的情况下,资源 j 对于任务(子任务) i 的评价函数为:

$$\min I = \alpha \frac{t_{ij}}{T} + \beta \frac{c_{ij}}{C} \quad (1)$$

即超级节点将从符合要求的多个资源中选择使目标函数 I 最小的资源 j 作为任务(子任务) i 的调度目标。

为了简化模型,假设超级节点模式的网络带宽足够宽,资源节点与超级节点以及超级节点与超级节点之间的通信、任务(子任务)以及结果的传输时延很小,相对于任务的执行时间可以忽略不计,所有任务或子任务均能在系统内找到合适的资源。

3.2.2 超级节点模式网格复合任务调度算法

算法具体如下。

Input: grid task GT

Output: mapping and assignment of grid tasks on nodes

Begin

ReceivedTask(super-peer_s, GT); ($1 \leq s \leq n$)

While GT comes from local grid community do

If GT can decompose

Then decompose it to subtasks with the QoS parameters; establish a dispatch list;

Else GT is called indivisible task;

For every grid task GT_i (GT_i is an sub-task or indivisible task)

If ExecutedTime (GT_i , $super-peer_s$, $node_j$) $\leq$ DeadlineTime (GT_i) // $node_j$ 是提交 GT_i 的源资源节点

Then Allocate(GT_i , $super-peer_s$, $node_j$);

If an error occurs to $node_j$

Then SendResult0(GT_i , $super-peer_s$);

// 返回一个含有错误报告的执行结果, 超级节点将会对 GT_i 进行重调度

Else SendResult1 (GT_i , $super-peer_s$); // 返回正常的执行结果

Else if in $super-peer_s$ exist $node_k$ ($k \neq j$) which meets the following two requirements:

ArrivalTime(GT_i) +

ExecutedTime (GT_i , $super-peer_s$, $node_k$) $\leq$ DeadlineTime(GT_i); // 时间属性要求

ExecutedCost (GT_i , $super-peer_s$, $node_k$) $\leq$ CostLimit (GT_i); // 费用属性要求

Then for all $node_k$ calculate the function I ;

Seek for the smallest I and the corresponding $node_k$;

Allocate(GT_i , $super-peer_s$, $node_k$);

If an error occurs to $node_k$

Then SendResult0(GT_i , $super-peer_s$);

Else SendResult1(GT_i , $super-peer_s$);

Else Queuing (GT_i , Peer-to-Peer Grid); // 先将 GT_i 插入队列中

Interact with other super-peers;

For every $node_l$ in every ($super-peer_s$) which meets the following two requirements:

ArrivalTime(GT_i) +

ExecutedTime (GT_i , $super-peer_s$, $node_l$) $\leq$ DeadlineTime(GT_i);

ExecutedCost (GT_i , $super-peer_s$, $node_l$) $\leq$ CostLimit (GT_i);

Calculate the function I ;

Seek for the smallest I and the corresponding $super-peer_s$, $node_l$;

Send (GT_i , $super-peer_s$); // 将 GT_i 发送到目标资源节点所属的超级节点

End While

While GT comes from remote $super-peer_s$,

Allocate (GT , $super-peer_s$, $node_m$); // $node_m$ 指上面提到的 $node_r$

If an error occurs to

Then SendResult0 (GT , $super-peer_s$, $super-peer_r$);

Else SendResult1 (GT , $super-peer_s$, $super-peer_r$);

End While

End

4 网络调度的层次颜色、价格时延 Petri 网模型

Petri 网是一个描述异步并发的图形工具, 具有可达树、可达图、关联矩阵等多种分析方法, 并且可以通过数学方法证明其正确性, 本文把它同网络调度结合起来作为研究的工具。

4.1 价格时延 Petri 网以及用户应用定义

有关 Petri 网的基本概念、详细内容见参考文献[20]。本文给出价格时延 Petri 网的定义, 具体如下。

定义 5 价格时延 Petri 网^[21] CTPN = (PN, Γ, D, C), 其中:

- PN 是 Petri 网, $PN = (S, T, F)$;
- Γ 是一个集合, 其元素 (t_j, τ_k) 表示变迁 t_j 实施的时刻为 τ_k ;
- $D = \{d_j | j = 1, \dots, m\}$, $d_j \in R^+ \cup \{0\}$, 表示完成每个变迁 t_j 所需要的时间, m 为变迁的个数;
- $C = \{c_j | j = 1, \dots, m\}$, $c_j \in R^+ \cup \{0\}$, 表示完成每个变迁 t_j 所需要的费用, m 为变迁的个数。

用户可通过给定的 d, c, τ 定义对任务的 QoS 需求。用户复合任务进行分解后形成相互之间具有依赖关系的子任务集合, 子任务之间存在诸如顺序、分支、循环、并行等关系。

定义 6 用户应用 UA 可以用类 BNF 表示为 $UA ::= \varepsilon / X / T / T \diamond T / T \otimes T / T \oplus T / \mu T$, 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完成, 这两个任务是为保证系统的完整性而引入的;
- $T \diamond T$ 代表两个任务顺序执行, 总的执行时间为两者执行时间之和, 费用为两者执行费用之和;
- $T \otimes T$ 代表两个任务分支执行, 一旦执行了其中的一个, 则不执行另一个, 相应的执行时间、费用等于



被选中任务的执行时间、费用；

- $T \oplus T$ 代表两个任务并行执行, $T1$ 和 $T2$ 应并行调度执行, 总的执行时间取两者中执行时间最大的, 而费用是两者之和;
- μT 表示循环执行 T 共 μ 次, 总的执行时间和费用为一次执行任务 T 对应值的 μ 倍。

4.2 网络任务调度的层次颜色、价格时延 Petri 网模型

定义 7 超级节点模式网络任务调度所对应的 Petri 网模型是一个层次颜色 Petri 网, $HCPN=(\Sigma, P, S, T, F, C, G, E, I)$, 其中:

- $\Sigma=\{(rt, dt, dc, \alpha, \beta)\} \cup \{(et, ct)\} \cup \{(ms, gt, gs)\}$ 是颜色的集合, rt, dt, dc 分别是任务(子任务)的到达时间、截止期限、费用上限; α, β 分别是用户规定的时间费用权重; et, ct 是任务(子任务)在各资源节点上执行时所花费的时间和费用估计值; ms, gt, gs 分别是资源的状态信息、任务(子任务)及其执行结果;
- P 是子网的集合, $P=\{super-peer_i | i=1, 2, \dots, n\}$, n 是全局超级节点的个数, $super-peer_i$ 是第 i 个超级节点所对应的子网;
- S 是库所的集合, $S=\{s_{i1}, s_{i2} | i=1, 2, \dots, n\}$, n 是全局超级节点的个数, s_{i1} 是超级节点负责接收从远程超级节点发送过来的任务(子任务)与资源信息的单元, s_{i2} 是超级节点向远程超级节点发送任务(子任务)与资源信息的发送单元;
- T 是变迁的集合, $T=\{t_{ijk} | i, j=1, 2, \dots, n, i \neq j, k=1, 2, 3\}$,

n 是全局超级节点的个数, t_{ijk} 表示第 i 个超级节点的发送单元向第 j 个超级节点的接收单元发送消息, 此处每个变迁的时延、价格属性均为 0, $k=1, 2, 3$ 分别表示发送本地资源节点的状态信息、任务(子任务)及其执行结果;

- F 是弧的集合, $F \subseteq (P \times S, S \times P, S \times T, T \times S)$;
- C 是颜色函数, $C: S \rightarrow \Sigma$;
- G 是哨岗函数, $G: T \rightarrow BoolExpression$ 并且满足 $\forall t \in T: Type(G(t)) = Boolean \wedge Type(var(G(t))) \subseteq \Sigma$, 其中, $var(G(t))$ 表示函数 $G(t)$ 所含变量的集合;
- E 为弧函数, $E: F \rightarrow BoolExpression$, 并且满足 $\forall f \in F, Type(E(f)) = C(s)_m \wedge type(var(E(f))) \subseteq \Sigma$ 其中, s 为 f 连接的库所;
- I 为初始化函数, $I: S \rightarrow \Sigma$ 为每一个库所赋颜色值生成初始标识 M_s , 即 $\forall s \in S: Type(I(s)) = C(s)_m$;

针对图 1, 为了简化模型, 假定超级节点模式的网格由 3 个互连的超级节点组成, 则对应的层次颜色 Petri 网 (HCPN) 模型如图 4 所示。

定义 8 第 i 个超级节点 $super-peer_i$ 所对应的子网为一个价格时延、颜色、增广 Petri 网。 $super-peer_i=(\Sigma, S, T, F, W, M_0, D, C)$, 各变量的定义介绍如下:

- $\Sigma=\{(rt, dt, dc, \alpha, \beta)\} \cup \{(et, ct)\} \cup \{(ms, gt, gs)\}$ 是颜色的集合, rt, dt, dc 分别是任务(子任务)的到达时间、截止期限、费用上限; α, β 分别是用户规定的时间费用权重; et, ct 是任务(子任务)在各资源节点上

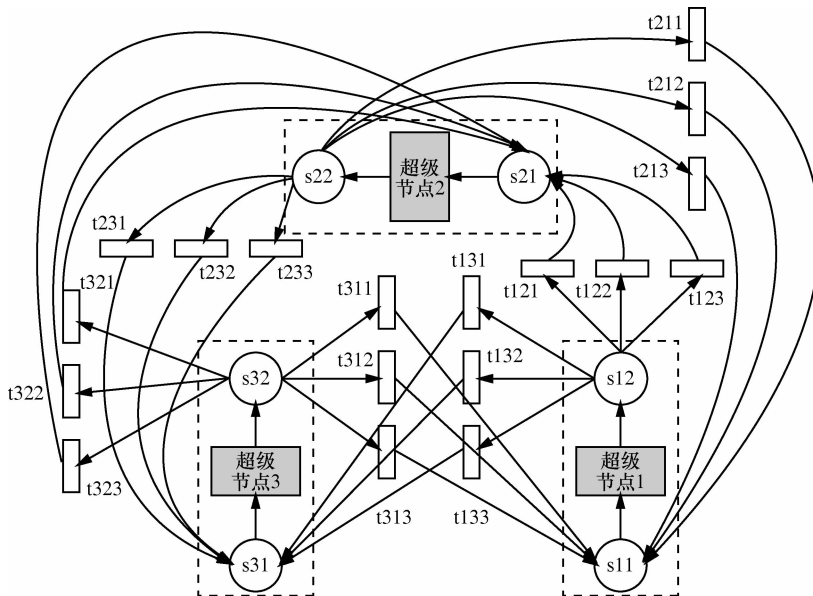


图 4 超级节点模式网络调度全局层次颜色 Petri 网模型

- 执行时所花费的时间和费用估计值; ms 、 gt 、 gs 分别是资源的状态信息、任务(子任务)及其执行结果;
- S 是库所的集合, $S=\{s_i|i=1,2, \cdots,13\} \cup \{(in, out)\}$, in 对应超级节点的接收单元, out 对应超级节点的发送单元;
 - T 是变迁的集合, $T=\{t_j|j=1,2,\cdots,17\}$;
 - F 是弧的集合, W 是弧的加权函数的集合, M_0 是初始标识;
 - $D:T \rightarrow R_0$ 是定义在变迁集 T 上的时延函数, $D(t)=a$ 表示变迁的发生需要 a 个单位时间来完成;
 - $C:T \rightarrow R_0$ 是定义在变迁集 T 上的价格函数, $C(t)=b$ 表示变迁的发生需要 b 个单位费用来完成。

$super-peer_i$ 所对应的 Petri 网子网如图 5 所示,它是一个价格时延、颜色、增广 Petri 网。

对图 5 中对应的库所、变迁和抑制弧的说明见表 1~表 3。

5 系统性能

为了得到超级节点模式的网格任务最优调度方案,观察各任务(子任务)在执行过程中产生的时延和费用情况,对系统进行性能评价,构造和分析对应的 Petri 网可达任务图。

定义 9 层次颜色 Petri 网是超级节点模式网格调度对应的全局 Petri 网模型, HCPN 的可达任务图(RTG)是一

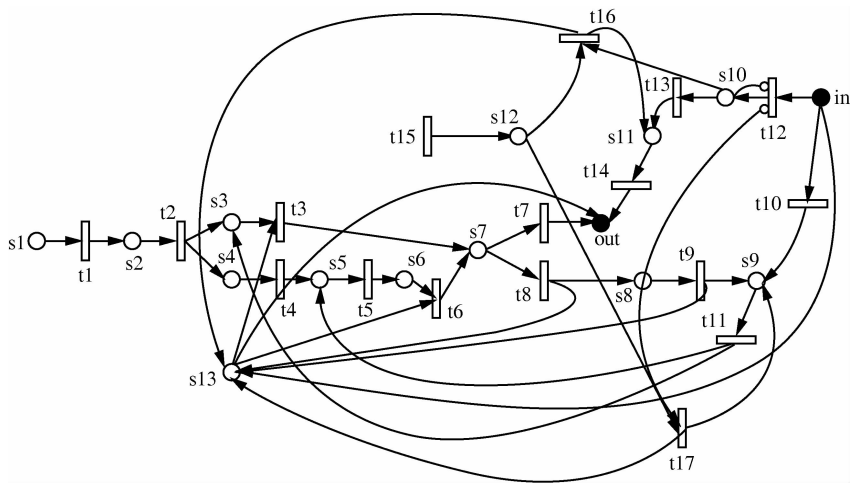


图 5 $super-peer_i$ 所对应的价格时延、颜色、增广 Petri 网

表 1 图 5 中的库所含义说明

库所	含义说明
s1	超级节点管理域内某一资源节点上用户准备提交的任务
s2	超级节点接收到来自资源节点提交的任务
s3	接收的任务是不可分解的
s4	接收的任务是复合任务,可以进行分解
s5	复合任务分解后具有时间先后依赖关系的子任务集合
s6	从等待调度的子任务集合里挑选出来准备调度的子任务
s7	已确认调度去向及目标的任务(子任务)
s8	在本地网格社区某一资源节点上准备执行的任务(子任务)
s9	任务(子任务)的执行结果
s10	待在本地网格社区某一资源节点上执行的远程任务(子任务)
s11	远程任务(子任务)的执行结果
s12	本网格社区内某一资源节点处于故障状态
s13	资源的状态信息
in	本地超级节点接收远程超级节点发送的待调度任务(子任务)、本地任务(子任务)的执行结果或远程超级节点管理域内资源节点状态信息的单元
out	本地超级节点向远程超级节点发送任务(子任务)、远程任务(子任务)执行结果或本地资源节点状态信息的单元



表 2 图 5 中的变迁含义说明

变迁	含义说明	变迁发生产生的时延、价格
t1	用户在资源节点向本地超级节点提交任务	0,0
t2	判断任务是否可以分解	0,0
t3	为不可分解的任务查询资源的状态信息	0,0
t4	将复合任务分解为具有时间先后依赖关系的子任务集合	0,0
t5	根据执行时间的先后顺序选择将要调度的子任务(并行任务可以同时调度),具有前驱的子任务必须等所有前驱执行完成后才能进行调度	0,0
t6	为挑选出来的子任务查询资源的状态信息	0,0
t7	将任务(子任务)传送到超级节点的发送单元,进行远程任务调度	0,0
t8	将任务(子任务)传送到本超级节点管理域内的某一资源节点	0,0
t9	本网格社区内的某一资源节点在执行任务(子任务)	若资源节点为任务(子任务)的提交节点,则价格为0,时延为一个正数,否则时延、价格均为正数
t10	接收从远程超级节点发送的本地任务(子任务)的执行结果	0,0
t11	判断任务(子任务)是否顺利执行。如果是,则子任务的后继可能获得调度的机会;否则准备进行重调度	0,0
t12	将接收到的远程任务(子任务)分配到本网格社区内的某一资源节点	0,0
t13	本地网格社区某一资源节点正在执行远程任务(子任务)	时延、价格均为正数
t14	将远程任务(子任务)的执行结果从某一资源节点传送到超级节点的发送单元	0,0
t15	本地网格社区某一资源节点发生故障	0,0
t16	为远程任务(子任务)生成一个含有错误报告的执行结果	0,0
t17	为本地任务(子任务)生成一个含有错误报告的执行结果	0,0

表 3 图 5 中的抑制弧含义说明

抑制弧	含义说明
s8 与 t12 之间	资源节点优先执行本网格社区内的任务(子任务)
s10 与 t12 之间	当资源节点执行远程任务(子任务)时,避免新的远程任务(子任务)分配到此资源节点,有利于资源节点间的负载均衡

个有向图,表示为 $RTG(HCPN)=(V,E)$,其中, V 是由带标记的标识所组成的顶点集合, E 是由带标记的连接标识的有向边所组成的边集合。借鉴参考文献[20], $RTG(HCPN)=(V,E)$ 的构造算法如下。

算法 2 计算可达任务图

输入:Petri 网系统 $HCPN=\langle N, M_0 \rangle$ 。

输出:对于有界网系统,可达任务图 $RTG(HCPN)=(V,E)$ 。

步骤 1 RTG 垂直分成 $m_1+m_2+\dots+m_k+n$ 个区域,其中 n 是超级节点的个数, m_k 是第 k 个超级节点中的客户机的个数。

步骤 2 初始化任务可达图 $RTG(HCPN)=({M_0}, \Phi)$, M_0 标记为“新”。

步骤 3 While 在集合 V 中还存在标记为“新”的节点 do

(1)从集合 V 中任意选择一个标记为“新”的节点 $M \in V$

(2)For 每一个在标识 M 下可以发生的变迁 t_j do

计算 M' ,使得 $M \xrightarrow{t_j} M'$;

If $M' \notin V$

Then i $V=V \cup \{M'\}$;

ii If $t_j \in \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_{11}, t_{14}, t_{16}, t_{17}\}$, 则将 M' 放在超级节点域内;

iii If $t_j \in \{t_8, t_9, t_{10}, t_{12}, t_{13}, t_{15}\}$ 则将 M' 放在超级节点内某一指定的资源节点域内。

$E=E+\{M, M'\}$,并将 $\{M, M'\}$ 标记为 t_j/GT_i ,若 $t_j \in \{t_9, t_{13}\}$,则将 t_j 附加标记 $[et, ec]$, et, ec 分别为 GT_i 任务(子任务)在指定的资源节点上执行产生的时延和费用。

将 M' 标记为 $[a, b]$, a 表示当前的时刻, b 表示目前任务执行到此累计消耗的费用。

(3)如果在标识 M' 下,原不可分解的任务执行完毕或者复合任务的所有子任务均已执行完毕,则将 M' 标记为“端点”;否则将 M' 标记为“新”。

(4)删除 M 的“新”,然后转到步骤 3 继续执行。

步骤 4 输出可达任务图 $RTG(HCPN)$ 。

命题 1 设 n_e 是 $RTG(HCPN)$ 中“端点”的个数, 则系统的吞吐量为 n_e 。

证明 由算法 2 知, 在 $RTG(HCPN)$ 中, 若为 M' 标记的“端点”, 则表明有一个任务执行完毕, 因此整个系统当前已完成的任务数, 即系统的吞吐量为 $RTG(HCPN)$ 中“端点”的个数。

命题 2 设 et_j 为 $RTG(HCPN)$ 的第 k 个资源节点区域中标注在执行变迁 t_j 边上的时延分量, $se_k = \sum et_j, \mu$ 与 s 分别是序列 $\{se_k / 1, 2, \dots, m_1 + m_2 + \dots + m_k\}$ 的平均值与标准差, 则离散系数 $v = s/\mu$ 可用来判断系统的负载平衡状态。

证明 由算法 2 知, $RTG(HCPN)$ 的第 k 个资源节点区域中标注在执行变迁 t_j 边上的 et_j 为某一任务 (子任务) 在第 k 个机器上的执行时间, 而 $se_k = \sum et_j$ 即第 k 个机器执行时间之和。又因为离散系数 $v = s/\mu$ 用来度量序列 $\{se_k\}$ 中数据的离散程度, 如果离散系数较小, 说明各机器执行时间之和比较接近, 因而系统就处于负载平衡状态, 反之, 系统就处于负载不平衡状态。因此, 离散系数 $v = s/\mu$ 可用来判断系统的负载平衡状态。

命题 3 设 M_k 是 $RTG(HCPN)$ 中的“端点”, a_k, b_k 分别

为 M_k 的时间、费用标识, 则整个系统目前完成所有任务后所处的时刻为 $\max\{a_k\}$, 所消耗的总费用为 $\sum b_k$ 。

证明 由算法 2 知, 若 M_k 为 $RTG(HCPN)$ 的“端点”, 则表明某一个任务执行完毕, 标注在 M_k 上的实数 a_k 为当前时刻, 即任务执行完毕的时刻, 而系统完成所有任务后所处的时刻就是最后一个任务执行完毕的时刻, 因此整个系统完成所有任务后所处的时刻为 $\max\{a_k\}$; 标注在 M_k 上的实数 b_k 为当前任务执行完毕所耗费的累积费用, 各复合任务之间相互独立, 所以, 系统目前完成所有任务后所消耗的总费用为 $\sum b_k$ 。

6 实例分析

假设在如图 6 所示的一个超级节点模式的网络计算环境中, 有两个相互独立的复合任务分别在节点 1、节点 3 提交, 它们的子任务结构如图 7 所示, 其中节点 1 和节点 2 属于超级节点 1 所在的网络社区; 节点 3 属于超级节点 2 所在的网络社区; 节点 4 属于超级节点 3 所在的网络社区。

用户在提交任务时, 可以定义其 QoS 参数要求, 如完

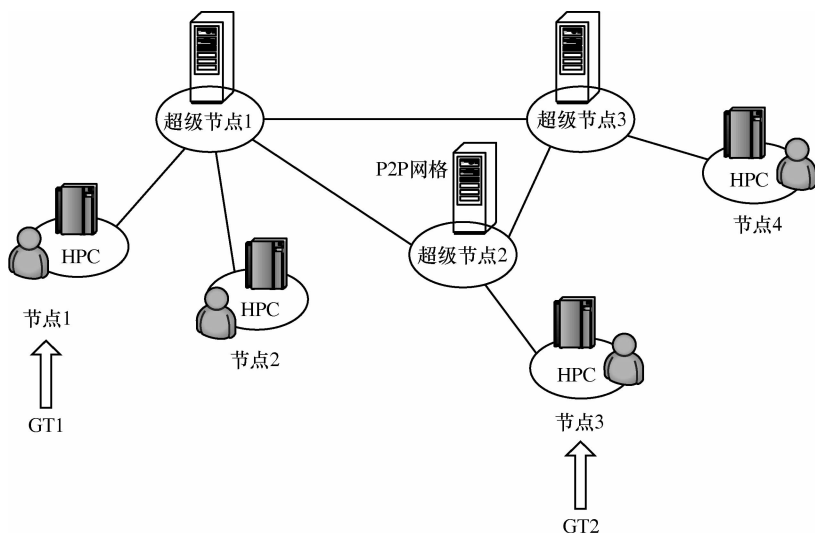


图 6 一个超级节点模式网络任务调度实例

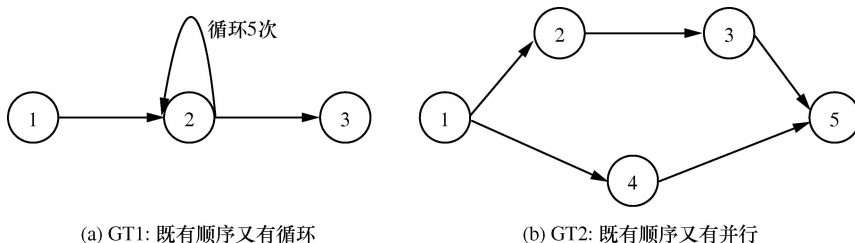


图 7 复合任务的子任务执行依赖关系结构



成截止时间、能承受的费用上限、时间与费用的权重。超级节点在接收到复合任务后,为了使调度方案的形成相对简单^[2],将复合任务的时间和价格 QoS 要求分解到每一个子任务,方便了调度方案的形成,大大减小了调度的复杂性。

表 4 给出了这些任务(子任务)的到达时间、截止时间、费用上限、时间费用权重、任务(子任务)在各资源节点上的估计执行时间和估计执行费用。

根据这些参数和算法 1、Petri 网模型以及算法 2,可以构造出该系统网格调度对应的 Petri 网的 RTG(HCPN),如图 8 所示。

由表 4 及图 8 可得到如下分析结果。

网格任务 GT1、GT2 的子任务均不能完全在各自的提交节点上完成。对于任务 GT1,其子任务 GT1-1 可以在截止时间内在提交它的节点 1 上执行完成,并不产生费用消耗。GT1-2、GT1-3 虽然不能在提交它的资源节点 1 上按时完成,但是在截止时间和预算允许的情况下,它们都能在本地超级节点管理的资源节点 2 上执行。其中,GT1-2 循环执行 5 次。对于任务 GT2,其子任务 GT2-1、GT2-2 可以在截止时间内在提交它的节点 3 上执行完成,并不产生费用消耗。对于 GT2-3,本地网格社区不能满足其 QoS 要求,超级节点 2 与超级节点 1、3 进行交互,发现满足其截止时间和费用上限要求的节点有节点 1、节点 2、节点 4,此时超级节点 2 中的市场代理将计算每个符合要求的资源节点

的评价函数:

$$\text{节点 1: } I = \frac{105}{110} \times 0.3 + \frac{45}{60} \times 0.7 = 0.811$$

$$\text{节点 2: } I = \frac{100}{110} \times 0.3 + \frac{50}{60} \times 0.7 = 0.856$$

$$\text{节点 4: } I = \frac{110}{110} \times 0.3 + \frac{60}{60} \times 0.7 = 1$$

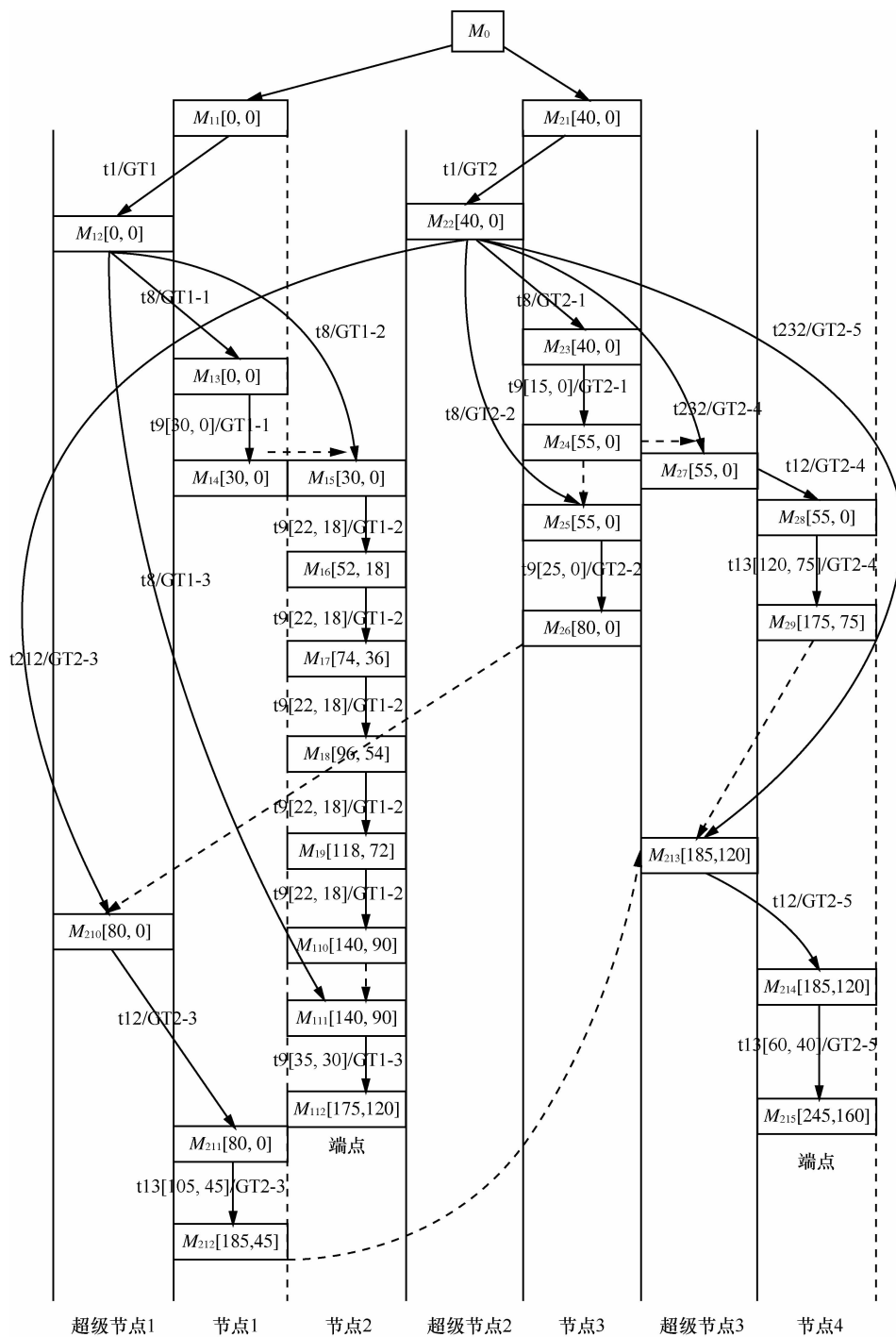
通过对比可以看出,节点 1 的评价函数值最小,将 GT2-3 调度到超级节点 1 管理的节点 1 节点上执行。对于 GT2-4,本地网格社区也不能满足其 QoS 要求,同时符合其截止时间和费用上限要求的资源节点只有超级节点 3 管理的节点 4,故节点 4 就是其调度目标。同理,对于 GT2-5,其调度目标也是节点 4。

节点 1 执行所有任务所用时间之和 $se_1 = 30 + 105 = 135$, 节点 2 执行所有任务所用时间之和 $se_2 = 110 + 35 = 145$, 节点 3 执行所有任务所用时间之和 $se_3 = 15 + 25 = 40$, 节点 4 执行所有任务所用时间之和 $se_4 = 120 + 60 = 180$ 。 se_1 、 se_2 、 se_3 、 se_4 的平均值为 $\mu = 125$, 标准差为 $s = 51.84$, 离散系数 $v = \frac{s}{\mu} = 0.41$, 由命题 2 知,系统目前的负载平衡状态一般,从执行时间分布上看,节点 3 的负载比较轻。

可达任务图有 2 个端点: M_{112} 、 M_{215} , 系统的吞吐量为 2。目前整个系统执行完所有任务后所处的时刻为 $\max\{175, 245\} = 245$, 所消耗的总费用为 $120 + 160 = 280$ 。

表 4 网格任务(子任务)的时间、价格等 QoS 要求

任务 (子任务)	到达 时间	截止期限	费用上限	估计执行时间				估计执行费用				α		β	
				节点 1	节点 2	节点 3	节点 4	节点 1	节点 2	节点 3	节点 4	时间权重	费用权重		
GT1	0	180	160	210	180	170	220	0	170	150	175	0.6	0.4		
GT1-1	0	40	35	30	35	40	50	0	50	30	45				
GT1-2(5 次 循环)		150(5 次 循环后)	19 / 单次	26 / 单次	22 / 单次	12 / 单次	14 / 单次	0 / 单次	18 / 单次	8 / 单次	8 / 单次				
GT1-3		180	30	50	35	70	100	0	30	70	90				
GT2	40	270	210	235	320	260	220	210	208	0	202	0.3	0.7		
GT2-1		60	10	10	60	15	20	10	8	0	15				
GT2-2		90	15	30	90	25	30	20	10	0	12				
GT2-3		190	60	105	100	120	110	45	50	0	60				
GT2-4		190	80	135	130	140	120	90	85	0	75				
GT2-5		270	45	90	70	100	60	45	55	0	40				

图 8 网格调度 Petri 网模型的 $RTG(HCPN)$

7 结束语

本文首先给出了一种超级节点模式的网格资源管理模型。针对此模型,引入市场经济机制,允许网格用户提出

任务的完成截止期限、费用上限以及时间费用权重,将它们作为 QoS 参数。然后,给出对应的网格任务(复合任务)调度算法,并利用一种新的 Petri 网——价格时延 Petri 网对超级节点模式的网格任务调度进行建模。考虑到网格环



境中资源是动态变化的,在调度算法及 Petri 网模型中增加相应的容错机制,当资源出现故障时,对任务进行重调度。最后,构建 Petri 网模型的可达任务图,通过一个例子,分析了具有顺序、并行或循环结构的复合任务的最佳调度方案、调度过程以及系统的吞吐量、负载平衡、调度时间、调度费用等重要特性。下一步的工作是进一步优化此模型,完善网格任务的调度算法,使各资源节点的时间负载特性尽量保持平衡。

参考文献

- 1 Foster I, Kesselman C. The Grid: Blueprint for New Computing Infrastructure. Morgan Kaufmann Publishers, San Francisco, CA, 1999
- 2 Mastroianni C, Talia D, Verta O. A super-peer model for resource discovery services in large-scale grids. Future Generation Computer Systems, 2005, 21(10): 1235~1248
- 3 Foster I, Kesselman C, Jeffrey M, *et al.* Grid services for distributed system integration. IEEE Computer, 2002,35(6):37~46
- 4 The web services resource framework. <http://www.globus.org/wsrf/>
- 5 Mastroianni C, Talia D, Verta O. Designing an information system for grids: comparing hierarchical, decentralized P2P and super-peer models. Parallel Computing, 2008(34):593~611
- 6 Kwan S K, Muppala J K. Resource discovery and scheduling in unstructured peer-to-peer desktop grids. Proceedings of the International Conference on Parallel Processing Workshops, San Diego, CA, 2010:303~312
- 7 Merz P, Wolf S, Schwerdel D, *et al.* A self-organizing super-peer overlay with a Chord core for desktop grids. Proceedings of IWSOS 2008, Vienna, Austria, 2008:23~34
- 8 Li Y, Huang X L, Ma F Y, *et al.* Building efficient super-peer overlay network for DHT systems. Proceedings of GCC 2005, Beijing, China, 2005:787~798
- 9 Cozza P, Talia D. A Super-Peer Model for Multiple Job Submission on a Grid. Core GRID Technical Report Number TR-0067, 2007
- 10 Wu C C, Chin J H, Lin Y S, *et al.* G2G:a meta-grid framework for the convergence of P2P and grids. Proceedings of GPC 2009, Geneva, Switzerland, 2009
- 11 Zhao S H, Chen G L, Wu G X, *et al.* A strategy for selecting super-peer in P2P and grid based hybrid system. Proceedings of Edutainment 2008, Nanjing, China, 2008:192~199
- 12 Colored J K. Petri Nets-Basic Concepts, Analysis Methods and Practical Use: Basic Concepts (2nd Edition). Springer-Verlag, Heidelberg, Berlin, 1996
- 13 熊曾刚, 杨扬, 曾明. 基于 Petri 网的两阶段网格任务调度模型与分析. 通信学报, 2009,30(8):69~77
- 14 Andrade, Brasileiro N, Cirne F, *et al.* Discouraging free riding in a peer-to-peer CPU-sharing grid. High Performance Distributed Computing, 2004, 12(3):129~137
- 15 熊曾刚, 杨扬, 刘丽等. 网络资源管理的 Grid 和 P2P 集成方案及其关键技术分析. 控制与决策, 2008,23(1):1~7
- 16 Foster I, Jennings N R, Kesselman C. Brain meets brawn: why grid and agents need each other. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, USA, July 2004
- 17 Xiong Z G, Yang Y, Zhang X M. Integrated agent and semantic P2P grid resource discovery model. Proceedings of Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE Computer Society Press, Qingdao, China, 2007
- 18 袁崇义. Petri 网原理与应用. 北京:电子工业出版社,2005
- 19 吴哲辉. Petri 网导论. 北京:机械工业出版社,2006
- 20 吉罗, 瓦尔克. 系统工程 Petri 网建模、验证与应用指南. 北京:电子工业出版社,2005
- 21 刘卫东, 宋佳兴, 林闯. 基于价格时间 Petri 网的网格计算应用模型及分析. 电子学报, 2005,33(8):1416~1420

作者简介



潘善亮,男,博士,宁波大学副教授,主要研究方向为服务计算、信息检索。



黄希,男,宁波大学硕士研究生,主要研究方向为网格计算。



茅琴娇,女,西安交通大学博士研究生,主要研究方向为个性化推荐、用户兴趣建模、统计学习。

(收稿日期:2014-01-12)