



基于价格时延 Petri 网的超级节点网格调度^{*}

潘善亮¹, 黄 希¹, 茅琴娇²

(1. 宁波大学计算机科学技术研究所 宁波 315211; 2. 西安交通大学电子与信息工程学院 西安 710049)

摘 要: 针对资源节点基于语义相似度聚类的超级节点模式网格, 提出了一种面向用户截止时间、费用上限及二者之间偏好参数等 QoS 参数约束的网格协作型任务调度算法, 其中考虑资源的动态性, 引入重调度机制。然后, 采用价格时延 Petri 网这一图形与数学工具, 对复杂的调度过程进行形式化、层次化建模, 增进对调度系统的理解。构建调度模型的可达任务图, 分析调度系统的诸如调度时间、费用、吞吐量、负载平衡等重要性能。实例验证调度算法的有效性, 分析了需要协调多种类型资源的网格协作型任务的调度过程及调度系统的若干重要特性。研究内容对于指导实际超级节点模式网格系统的开发、部署均具有重要意义。

关键词: 网格调度; 超级节点; QoS 参数; 价格时延 Petri 网

doi: 10.3969/j.issn.1000-0801.2013.11.009

Grid Scheduling of Super-Peer Model Based on Price Timed Petri Net

Pan Shanliang¹, Huang Xi¹, Mao Qinjiao²

(1. Institute of Computer Science & Technology, Ningbo University, Ningbo 315211, China;

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

Abstract: For grid of super-peer model where nodes were clustering together by means of semantic similarity, a compound task scheduling algorithm was proposed which taking users' time deadline, cost limit and the weight parameters between them as QoS requirements. Considering grid nodes were dynamic, the rescheduling mechanism was introduced. Price timed Petri net which was a powerful graphical and mathematical tool was used to formally and hierarchically model grid scheduling, in order to have better understanding about it. Reachable task graph of Petri net model was constructed to analyze scheduling performance, such as scheduling time, cost, throughout capacity and load balance, etc. Effectiveness of the scheduling algorithm was verified by an example, scheduling process of compound tasks which need several types of resources and the system's scheduling performances were analyzed. Related work had great significance on guiding the construction and implement of real grid systems of super-peer model.

Key words: grid scheduling, super-peer, QoS parameter, price timed Petri net

1 引言

近几年提出的超级节点模式网格资源管理^[1,2]很好地在高效的集中式搜索与扩展性强的分布式搜索之间实

现了平衡。超级节点模式网格按资源注册方式主要分为两种^[3]: 一种是资源节点按地域界限划分的超级节点模式网格; 另一种是资源节点基于语义相似度聚类的超级节点模式网格。其中, 在第二种超级节点模式网格中, 资源节点

^{*} 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)

基于语义相似度聚类,同一超级节点域内的资源类型相同。这种类型的网络至少具有以下优点:资源的描述包含本体语义特征,资源发现具有更高的查全率与查准率^[4];具有相同类型的资源预先分组聚类,能降低查找空间,资源发现具有更高的效率^[5,6]。因此,资源节点基于语义相似度聚类的超级节点模式网络受到了广泛的关注。

参考文献[7~9]提出了利用超级节点网络来管理文件共享系统,资源节点依据其共享文件的语义相似性进行分组。参考文献[10]将 Web 服务按照语义定义进行分组聚类,由超级节点进行管理。参考文献[11]构建了一个网格资源节点身份可靠性度量函数和行为表现信誉度评估策略,建立了一个网格任务调度的安全融合模型,以此为基础,提出一个时间—安全驱动的双目标优化网格依赖任务调度模型。参考文献[12]将节点的兴趣进行分组聚类,每个分组内的节点构成一个语义树,根节点之间互联形成一个覆盖网络。参考文献[13]提出语义覆盖簇的概念,讨论了资源节点如何加入对应的超级节点。参考文献[14]提出了基于超级节点模式网格的资源语义描述与注册框架。参考文献[15]在参考文献[14]的基础上提出了资源分类的处理过程以及超级节点的选择方法。以上学者虽然将本体以及资源分类聚集的思想引入网格中,但是他们的讨论局限于网格体系结构,资源的描述与注册框架以及资源分类的处理过程等方面,而没有具体讨论对于一个需要协同不同类型的资源以完成网格协作型任务的具体调度方法及其形式化建模和理论分析。参考文献[16]针对资源节点按地域界限划分的超级节点模式网格提出了一种两阶段网格任务调度方法,并用时延 Petri 网对调度过程进行了建模、分析。但是其讨论的对象是简单的元任务且只考虑了调度的时间特性。

不同于以上参考文献的研究,本文针对资源节点基于语义相似度聚类的超级节点模式网络,提出网格协作型任务的面向用户时间、费用及二者偏好参数等多 QoS 约束的调度算法且考虑了重调度机制。鉴于复杂的调度过程具有并行、分布式的特点,笔者利用价格时延 Petri 网对调度过程进行建模,图形化调度过程,构造并分析可达任务图,通过实例验证算法的有效性。

2 网格协作型任务调度算法

2.1 资源管理模型

图 1 显示了资源节点基于语义相似度聚类的超级节

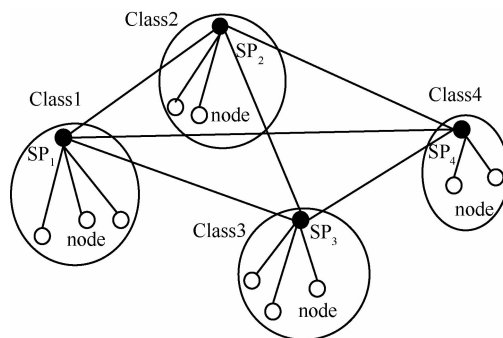


图 1 网格体系结构

点模式网格体系结构^[14,15]。

在图 1 中,网格资源节点基于语义相似度分组聚类,每个聚类对应的网格社区有一个超级节点负责管理。规定一个节点只提供一种类型的资源且资源的分类体系不考虑各类型之间的层次关系^[17]。

“Class”为一组语义相似的网格资源节点集合。每个 Class 包含两种类型的节点:资源节点(node)和超级节点(SP)。超级节点是性能评估参数较高的节点,为本聚类中的节点作为服务器运作,代表对应的资源分类。不同类型的 SP 之间采用 P2P 模式互连。每个 node 都在对应的 SP 处进行注册。

有关资源节点基于语义相似度的聚类技术、资源与任务的匹配方法等在参考文献[6]中有详细介绍,本文不再赘述。

2.2 调度算法

先定义如下的参数: t_{ij} 为任务 i 在资源节点 j 上的估计执行时间。 T 为任务的截止期限与到达时间之差。 c_{ij} 为任务 i 在其他用户资源节点 j 上的估计执行费用。 C 为任务的费用上限。 $\alpha + \beta = 1; \alpha \geq 0, \beta \geq 0$ 。其中, α, β 为用户定义的时间和费用偏好参数。

定义 1 在语义匹配且截止期限和费用上限均能满足的情况下,资源 j 对于任务 i 的评价函数为:

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

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 i 的调度目标。

在本文的调度模型中,超级节点具有一定的智能,用户向系统提交任务,如果任务能在该用户所在社区执行完成,就直接由该社区的超级节点调度完成任务的执行。如果任务不能在该社区内完成,就需要由该用户所在的社区的超级节点与其他社区的超级节点协调,找到合适的社区



完成任务的执行。调度算法简要描述如下。

步骤 1 用户将网格任务提交给本地超级节点(SP), SP 接收后,判断其是否可以根据资源类型分解为子任务。

步骤 2 对于任一任务(元任务或者子任务),SP 首先考虑其所需的资源类型是否对应于本地类型,若是,则转步骤 3;否则,转步骤 4。

步骤 3 SP 进行本地集中式调度。对于语义匹配的资源节点,以定义 1 中的评价函数作为衡量目标,择优调度,执行交易,返回执行结果给 SP。

步骤 4 SP 进行分布式远程调度。根据所需资源类型查找并发送任务至目标 SP,由其在对应分组内作集中式调度。对于匹配的资源节点,以定义 1 中的评价函数作为衡量目标,择优调度,执行交易,返回执行结果至源 SP。

步骤 5 不断重复步骤 2,直至协作型任务的所有子任务都执行完毕。

一个超级节点的网格任务调度算法用伪代码表示如下。

```

Super-peer scheduling algorithm
INPUT: grid task GT
OUTPUT: execution result of GT
Begin
ReceivedTask(Super-peer.s,GT);
While GT comes from local grid community do
If GT can decompose
Then according to resource classification, decompose it to
subtasks with the QoS parameters; establish a dispatch list;
Else GT is called meta-task;
For every grid task GTi (GTi is a sub-task or meta-task)
If the local resource type can satisfy GTi, Super-peer.s
search for the semantic matching node.k which also meets the
following two requirements:
ArrivalTime(GTi)+ExecutedTime(GTi, Super-peer.s, node.
k)DeadlineTime(GTi);//时间属性要求
ExecutedCost (GTi, Super-peer.s,node.k)CostLimit (GTi) ;
//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(GTi, Super-peer.s, node.k');
If an error occurs to node.k'
Then SendResult0(GTi, Super-peer.s);//错误报告,重调度

```

```

Else SendResult1(GTi, Super-peer.s);
Else Queuing(GTi,Peer-to-Peer Grid);
Search some Super-peer.t(t≠s) according to the resource
type //超级节点之间分布式资源类型查询
Send (GTi, Super-peer.t);
Endwhile
While GT comes from remote Super-peer.r
For every semantic matching node.l in Super-peer.s
which also meets the following two requirements:
ArrivalTime(GT)+ExecutedTime (GT, Super-peer.s, node.
l)DeadlineTime(GT);
ExecutedCost (GT, Super-peer.s, node.l)
CostLimit(GT);
Calculate the function I ;Seek for the smallest I and the
corresponding node.l';
Allocate(GT, Super-peer.s, node.l');
If an error occurs to node.l'
Then SendResult0 (GT, Super-peer.s, Super-peer.r); //重
调度
Else SendResult1(GT, Super-peer.s, Super-peer.r);
Endwhile
End

```

设任务 T 被分解为 n 个子任务 $t_1, t_2, \dots, t_n$ 执行,整个系统有 m 个网格社区。任务 t_i 的执行时间是由任务在网格社区里的某个节点的最大执行时间决定的,设为 $\max\{t_i\}$,则整个调度算法的时间复杂度为 $O(m \cdot \max\{t_i\})$ 。

3 网络调度的价格时延 Petri 网模型

有关 Petri 网、颜色 Petri 网的基本概念与性质可以参考文献[18,19]给出的价格时延 Petri 网的定义。

定义 2 资源节点基于语义相似度聚类的超级节点模式网格任务调度对应的全局 Petri 网模型是一个层次颜色 Petri 网: $HCPN=(P, S, T; F, C, W, M_0, I)$ 。具体解释如下。

- P 是子网的集合, $P=\{Super-peer_i(SP_i)/i=1, 2, \dots, n\}$, n 是超级节点的个数, $Super-peer_i(SP_i)$ 是第 i 个超级节点所对应的局部子网。
- S 是库所的集合, $S=\{S_{i1}, S_{i2}/i=1, 2, \dots, n\}$, n 是超级节点的个数, S_{i1} 是接收从远程超级节点发送来的信息的单元, S_{i2} 是向远程超级节点发送信息的单元。

- T 是变迁的集合, $T=\{T_{ij}/i,j=1,2,\dots,n,i\neq j\}$, n 是超级节点的个数, T_{ij} 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。其中, 每个变迁的时延、价格属性均为 0。
- F 是弧的集合, $F\subseteq(P\times S, S\times P, S\times T, T\times S)$ 。
- $C=\{(rt, dt, dc, \alpha, \beta)\} \cup \{\tau, et, ct\} \cup \{(gt, gs)\} \cup R$ 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及费用偏好参数。 τ 是变迁实施的时刻, et, ct 是任务(元任务或者子任务)在各资源节点上执行时所花费的时间和费用估计值。 gt, gs 分别是超级节点之间发送的任务信息及其执行结果信息。 R 为资源类型集合。
- W 为弧函数; M_0 是初始标识; I 是哨岗函数。

假设图 1 中有代表 4 种不同类型资源的超级节点互连, 则全局 Petri 网模型如图 2 所示。

定义 3 第 i 个超级节点 $Super-peer_i(SP_i)$ 所对应的局部

子网包括两个相互联系的颜色、价格时延 Petri 网: 集中式调度 Petri 网和分布式调度 Petri 网。

定义 4 集中式调度 Petri 网: $Super-peer_{ic}=\{S, T; F, C, W, M_0, I, G, D, E\}$ 。

- S 是库所的集合, $S=\{s/i=1,2,\dots,17\} \cup \{(CIN, COUNT)\}$, CIN 为集中式调度子网接收来自分布式调度子网信息的单元, $COUT$ 为集中式调度子网向分布式调度子网发送信息的单元。
- T 是变迁的集合; F 是弧的集合; C 是颜色的集合; W 为弧的加权函数集合; M_0 是库所的初始标识集合; I 是变迁的哨岗函数集合; T 是变迁实施时刻的集合。
- D 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 单位时间来完成。
- E 是定义在变迁集 T 上的价格函数, $E(t)=b$ 表示变迁的发生需要 b 单位费用来完成。

图 3 显示了 $Super-peer_i$ 子网包含的集中式调度 Petri

网 $Super-peer_{ic}$ 。

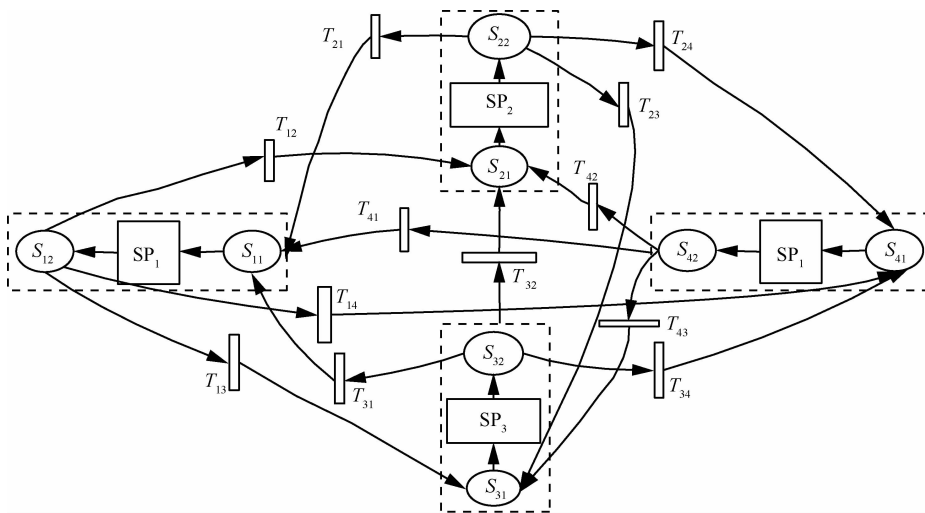


图 2 网络调度全局层次 Petri 网 HCPN 模型

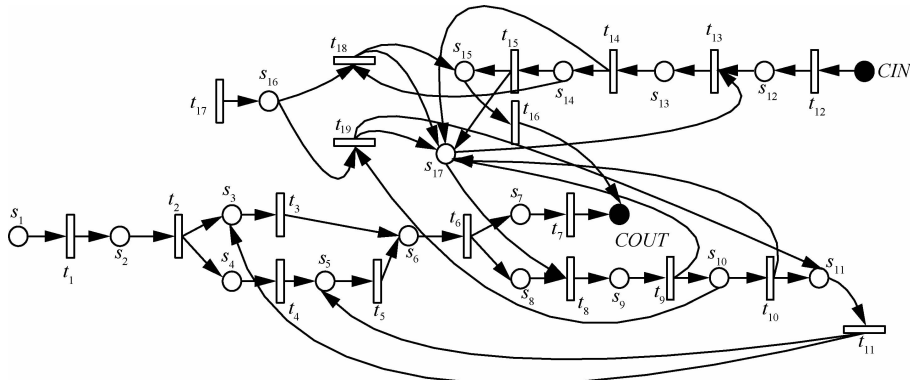


图 3 集中式调度 Petri 网 $Super-peer_{ic}$



为了简便,以下仅介绍图3中变迁的含义,见表1。

表 1 图 3 中变迁的含义

变迁	含义简介	时延, 价格
t_1	用户在资源节点向本地 Super-peer 提交任务	0,0
t_2	判断任务是否可以根据资源类型进行分解	0,0
t_3	从元任务集合里选择任务待调度	0,0
t_4	将协作型任务分解为子任务	0,0
t_5	选择将要调度的子任务	0,0
t_6	判断任务(元任务或子任务)所需资源类型为本地还是远程的	0,0
t_7	集中式调度子网向分布式调度子网发送任务	0,0
t_8	为任务查询本社区资源	0,0
t_9	将任务调度到本社区某一资源节点	0,0
t_{10}	本社区某一资源节点在执行任务	a, b
t_{11}	返回任务的执行结果	0,0
t_{12}	从分布式调度子网接收远程任务	0,0
t_{13}	为远程任务查询本社区资源	0,0
t_{14}	将远程任务调度到本社区某一资源节点	0,0
t_{15}	某一资源节点正在执行远程任务	a, b
t_{16}	将远程任务的执行结果返回给分布式调度子网	0,0
t_{17}	本社区内某一资源节点发生故障	0,0
t_{18}	为远程任务生成一个含有错误报告的执行结果	0,0
t_{19}	为本地任务生成一个含有错误报告的执行结果	0,0

定义 5 分布式调度 Petri 网: $Super\text{-}peer_D = (S, T; F, C, W, M_0, I, \Gamma, D, E)$ 。

其中,各元素的含义与定义4中类似。图4显示了 $Super\text{-}peer_i$ 子网包含的分布式调度 Petri 网 $Super\text{-}peer_D$ 。图中每个变迁的时延、费用属性均为0。为了简便,仅介绍图4中库所的含义,见表2。

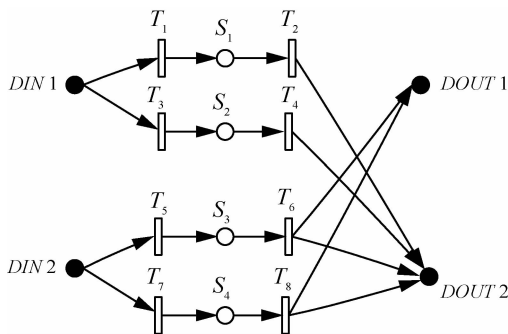


图 4 分布式调度 Petri 网 $Super\text{-}peer_D$

表 2 图 4 中库所的含义

库所	含义说明
S_1	分布式调度子网接收到集中式调度子网发送来的待分布式远程调度的任务(元任务或子任务)
S_2	远程任务在本社区的执行结果
S_3	中继转发的任务
S_4	中继转发的任务执行结果
DIN1	此单元即是集中式调度 Petri 网中的 COUT 单元
DOUT1	此单元即是集中式调度 Petri 网中的 CIN 单元
DIN2	接收从远程超级节点发送来的信息的单元,即是定义 2 中的 s_{i1}
DOUT2	向远程超级节点发送信息的单元,即是定义 2 中的 s_{i2}

记的标识组成的顶点集合, E 是由带标记的连接顶点的有向边所组成的边集合。有向图的构造算法类似于参考文献[16]中提出的算法。基于 RTG(HCPN), 可以进行时间和价格特性分析, 根据每个任务上到达时间、截止时间、费用上限等, 得出整个调度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进行调度。根据有向图 $RTG(HCPN) = (V, E)$ 的时间和价格信息, 分别可以得到对应的 AOE 网 N_t 和 N_p , N_t 和 N_p 是一个带权的有向无环图, 其中顶点表示事件, 弧表示任务, 权表示任务持续的时间或价格。以下算法给出了计算调度最长时间或最大费用的过程。

算法 根据 AOE 网计算调度的总时间和成本。

步骤 1 根据 AOE 网求得各个顶点的最大入度。

步骤 2 初始化堆栈 S 。

步骤 3 将入度为零顶点入堆栈 S , 置这些入度为零的顶点的最长时间 $ve(m)=0$; // 或最费用为零。

步骤 4 While 堆栈 S 不为空

从堆栈中弹出顶点 i

$$ve(i) = \max\{ve(k) + dut(<k, i>)\} \quad <k, i> \in T,$$

$1 \leq i \leq n-1$;

其中, T 为所有以 i 为头的弧 $<k, i>$ 的集合, $dut(<k, i>)$ 表示与弧 $<k, i>$ 对应的活动的持续时间顶点 k 的入度减 1;

若顶点 k 的入度为零, 则将顶点 k 入堆栈;

步骤 5 最后返回各顶点的最长时间。

计算出的最长实施时间和最大成本, 可作为调度算法的依据之一。

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 。

4 可达任务图

定义 6 网格调度全局 Petri 网 HCPN 模型的可达任务图是一个有向图 $RTG(HCPN) = (V, E)$, 其中 V 是由带标

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 $\{se_k\}$ 的平均值与标准差, 则离散系数 $v = s/\mu$ 可以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 $\sum b_k$ 。

5 实例分析

假设在如图 5 所示的网格计算环境中, 节点 a_1 和 a_2 属于 SP_1 所代表的资源分组 A 类型网格社区; 节点 b_1, b_2, b_3 属于 SP_2 所代表的资源分组 B 类型网格社区; 节点 c_1 属于 SP_3 所代表的资源分组 C 类型网格社区; 节点 d_1, d_2 属于 SP_4 所代表的资源分组 D 类型网格社区。有两个相互之间独立的协作型任务 GT_1, GT_2 分别在节点 a_1, c_1 提交, 它们的子任务结构如图 6 所示。圆圈内的数字表示第几个子任务, 字母表示所需的资源类型。

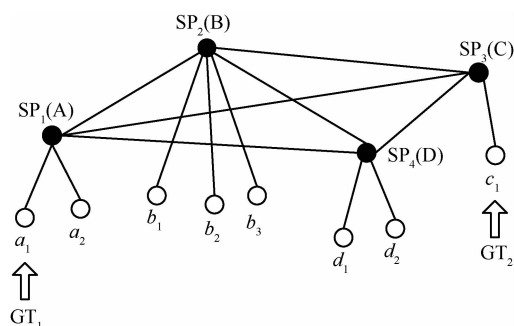
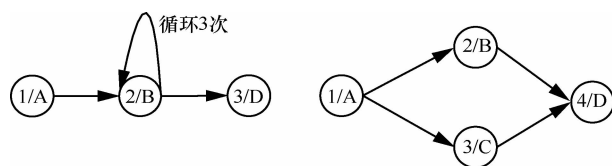


图 5 一个调度实例



GT_1 : 既有顺序, 又有循环

GT_2 : 既有顺序, 又有并行

图 6 协作型任务的子任务结构

表 3 给出了这些任务(子任务)的达到时间、截止期限、费用上限、时间/费用权重; 表 4 和表 5 分别给出了任务(子任务)在各资源节点上的估计执行时间和估计执行费用。为了讨论方便, 假设与每个子任务对应资源类型的网格节点均能满足其语义匹配要求。根据这些参数和调度算法、Petri 网模型、定义 6, 可以构造出该系统网格调度对应的 Petri 网可达任务图 RTG(HCPN), 如图 7 所示。

与参考文献[16]对比, 本文作了以下较大改进。

- 资源节点基于语义相似度预先分组聚类, 能提高资源发现准确率与效率。
- 将网格调度的目标由简单的元任务改为常见的协作型任务, 协作型任务调度机制复杂。
- 调度算法面向用户时间、费用及二者之间权重参数等多 QoS 约束, 而不只是考虑单一的时间参数; 引入市场机制, 调度算法更能够表征实际网格资源供需双方的自私性, 实现资源的优化分配。
- 建模工具价格时延 Petri 网相比于时延 Petri 网, 功能更强大, 能同时反映调度过程中产生的时延、价格问题。

总之, 本文的研究更加全面、系统和科学。

下面结合表 3~表 5 及可达任务图进行详细的调度分析。

(1) 网格任务 GT_1, GT_2 均为协作型任务, 都需要多种类型的资源协同配合才能完成。对于任务 GT_1 , 需要 3 种类型的资源: A、B、D。SP₁ 接收到任务后, 按照资源类型将其分解为 $GT_{1-1}, GT_{1-2}, GT_{1-3}$ 3 个子任务。对于子任务 GT_{1-1} , 其需求的资源类型为 A, 属于 SP₁ 所管理的网格社区。由于 a_1 是 GT_1 的提交节点, 故 a_1 对应于 GT_{1-1} 的执行费用为 0。SP₁ 进行集中式查询发现, a_1, a_2 均能满足 GT_{1-1} 的时间、费用等 QoS 参数的要求, 计算 a_1, a_2 相对于子任务 GT_{1-1} 的资源评价函数, 发现 a_1 优于 a_2 , 故 a_1 是 GT_{1-1} 的调度目标。其执行完成后所处的时刻为 25, 也是 GT_{1-2} 的开始调度时刻; 对于 GT_{1-2} , 需要资源类型为 B 的资源, SP₁ 将其发送给 SP₂ 进行分布式调度。由于其 3 次循环后的时刻不能超过 120, 故每次执行时间不能超过 $\frac{120-25}{3} \approx 31.7$ 。在网格社区 B 中, 同时满足其时间、费用 QoS 参数要求的只有 b_2 一个节点, SP₂ 调度 GT_{1-2} 到 b_2 上执行。 GT_{1-2} 执行完成后, 所处的时刻为 115, 累积费用消耗为 72; 对于 GT_{1-3} , 将被发送到 SP₄ 进行分布式调度。其执行时间不能超过 $160-115=45$, d_1, d_2 均能满足其 QoS 参数要求。SP₄ 计算它们的资源评价函数, $I_{d1} = \frac{40}{45} \times 0.6 + \frac{40}{45} \times 0.4, I_{d2} = \frac{38}{45} \times 0.6 + \frac{44}{45} \times 0.4, I_{d1} \leq I_{d2}$ 。 d_1 为 GT_{1-3} 的调度目标。当 GT_{1-3} 执行完成后, GT_1 便执行完毕。 GT_1 执行完毕后, 所处的时刻为 155, 消耗的总执行时间为 155, 总费用为 112。

对于 GT_2 , 因其费用权重为 0.7, 用户更看重执行费用。SP₃ 接收 GT_2 后, 按照其对资源类型的要求将其分解为 4 个具有相互依赖关系的子任务: $GT_{2-1}, GT_{2-2}, GT_{2-3}, GT_{2-4}$ 。对于

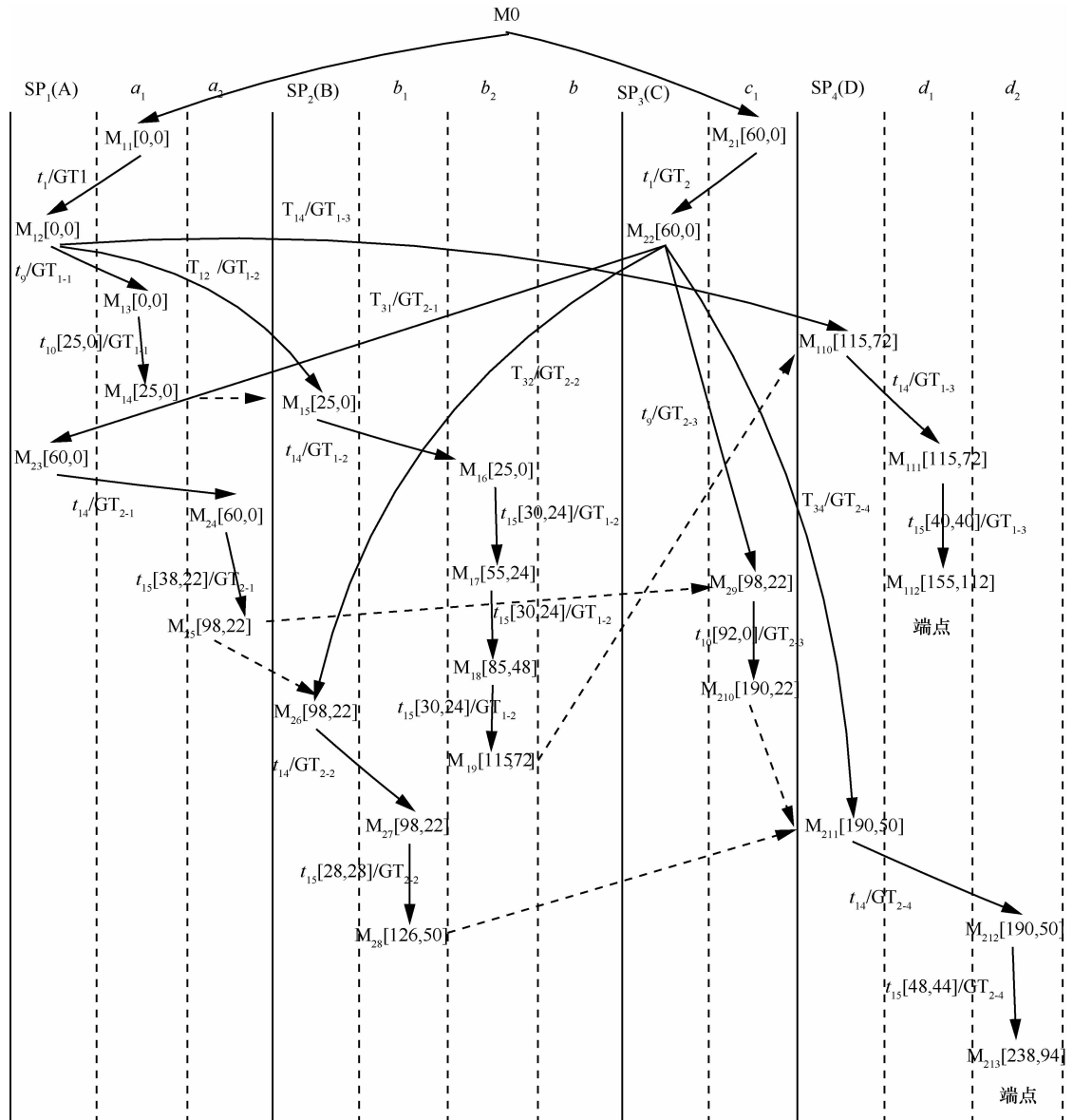


图7 网络调度 Petri 网模型的可达任务图

表3 网络任务(子任务)的 QoS 约束

任务/子任务	到达时间	截止期限	费用上限	α	β
GT ₁	0	160	140	0.6	0.4
GT ₁₋₁	0	30	20		
GT _{1-2/3} 次		120/3 次循环后	25/次		
GT ₁₋₃		160	45		
GT ₂	60	240	190	0.3	0.7
GT ₂₋₁	60	100	25		
GT ₂₋₂		195	30		
GT ₂₋₃		195	90		
GT ₂₋₄		240	45		

表4 网络任务(子任务)的执行时间估计值

	a_1	a_2	b_1	b_2	b_3	c_1	d_1	d_2
GT ₁								
GT ₁₋₁	25	23	-	-	-	-	-	-
GT _{1-2/3} 次	-	-	26/次	30/次	34/次	-	-	-
GT ₁₋₃	-	-	-	-	-	-	40	38
GT ₂								
GT ₂₋₁	36	38	-	-	-	-	-	-
GT ₂₋₂	-	-	28	31	35	-	-	-
GT ₂₋₃	-	-	-	-	-	92	-	-
GT ₂₋₄	-	-	-	-	-	-	52	48

表 5 网格任务(子任务)的执行费用估计值

	a_1	a_2	b_1	b_2	b_3	c_1	d_1	d_2
GT ₁								
GT ₁₋₁	0	18	-	-	-	-	-	-
GT _{1-2/3} 次	-	-	30/次	24/次	28/次	-	-	-
GT ₁₋₃	-	-	-	-	-	-	40	44
GT ₂								
GT ₂₋₁	24	22	-	-	-	-	-	-
GT ₂₋₂	-	-	28	29	32	-	-	-
GT ₂₋₃	-	-	-	-	-	0	-	-
GT ₂₋₄	-	-	-	-	-	-	43	44

GT₂₋₁, 需要 A 类型的资源, SP₃ 将其发送给 SP₁, 其开始时刻为 60, 截止期限为 100, 故执行时间不能超过 40。SP₁ 进行集中式查询发现, a_1 、 a_2 均能满足 GT₂₋₁ 的时间、费用等 QoS 参数要求, 计算 a_1 、 a_2 相对于子任务 GT₂₋₁ 的资源评价函数, 发现 a_2 优于 a_1 , 故 a_2 是 GT₂₋₁ 的调度目标。GT₂₋₁ 执行完成后所处的时刻为 60+38=98; 此后, 因 GT₂₋₂、GT₂₋₃ 是并行任务且分别需要 B 类型和 C 类型的资源, 故可以在 SP₂、SP₃ 处进行并行调度。对于 GT₂₋₂, SP₂ 进行集中式查询发现 b_1 、 b_2 能满足其 QoS 参数要求且 b_1 的资源评价函数优于 b_2 , GT₂₋₂ 被调度到 b_1 上执行; 对于 GT₂₋₃, SP₃ 管理的社区中只有 c_1 一个节点且它是 GT₂ 的提交节点, 对应于 GT₂₋₃ 的执行费用为 0。它能在截止期限内完成子任务 GT₂₋₃、GT₂₋₃ 被调度到 c_1 上执行; 只有当 GT₂₋₂、GT₂₋₃ 都顺利执行完成后, GT₂₋₄ 才能获得调度机会, 此时刻为 60+38+92=190。由于其截止期限为 240, 故其执行时间限制为 240-190=50, 它需要 D 类型的资源, SP₄ 管理的社区中只有 d_2 能满足其 QoS 参数要求, GT₂₋₄ 被调度到 d_2 节点上执行。当 GT₂₋₄ 执行完成后, GT₂ 便执行完毕。GT₂ 执行完毕后, 所处的时刻为 238, 消耗的总执行时间为 178, 总费用为 94。

(2) 资源节点 a_1 、 a_2 、 b_1 、 b_2 、 b_3 、 c_1 、 d_1 、 d_1 执行所有任务(子任务)所用的时间分别为: $se_1=25$, $se_2=38$, $se_3=28$, $se_4=90$, $se_5=0$, $se_6=92$, $se_7=40$, $se_8=48$ 。它们的平均数为 $\mu=45.1$, 标准差 $S=29.6$, 离散系数 $v=\frac{S}{\mu}=0.66$ 。由命题 5 知, 系统目前的负载平衡状态较差, 主要是因为目前统计的时间段短, 子任务数量比较少, 以致资源节点忙闲程度不同。相信在长时间的观察统计中, 系统的负载平衡特性会处于良好状态。

(3) 可达任务图有 2 个端点: M_{112} 、 M_{213} 。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 $\max\{155, 238\}=238$, 所消耗的总费用为: $112+94=206$ 。

6 结束语

本文针对资源节点基于语义相似度聚类的超级节点模式网格, 提出网格协作型任务的面向用户时间、费用及二者偏好参数等多 QoS 约束的调度算法, 并运用价格时延 Petri 网对调度过程进行建模, 构造并分析可达任务图, 实例验证算法的有效性。将子任务的传输过程、资源分配状况以及每一步产生的时延、费用等信息通过标识的形式反映在可达任务图中, 便于观察、统计及分析任务的详细调度过程。下一步的工作重点是进一步优化调度算法。

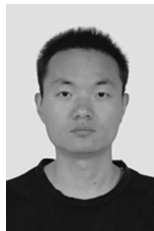
参考文献

- 1 Kurve A, Griffin C, Miller D J, *et al.* Optimizing cluster formation in super-peer networks via local incentive design. Peer-to-Peer Networking and Applications, 2013(4)
- 2 Liu M R, Koskela T, Ou Z H, *et al.* Super-peer-based coordinated service provision. Journal of Network and Computer Applications, 2011, 34(4):1210~1224
- 3 Hassan M I, Abdullah A. Semantic-based grid resource discovery systems a literature review and taxonomy. Proceedings of International Symposium in Information Technology (ITSim), Kuala Lumpur, June 2010
- 4 吴健, 吴朝晖, 李莹等. 基于本体论和词汇语义相似度的 Web 服务发现. 计算机学报, 2005, 28(4):595~602
- 5 曹洁, 曾国荪, 钮俊等. 云环境下可用性感知的并行任务调度方法. 计算机研究与发展, 2013, 50(7):1563~1572
- 6 倪晚成, 刘连臣, 吴澄等. 基于概念关联程度的网格服务组合方法. 清华大学学报, 2007, 7(40):1581~1585
- 7 Tan Y H, Lü K, Lin Y P. Organisation and management of shared documents in super-peer networks based semantic hierarchical cluster trees. Peer-to-Peer Networking and Applications, 2012, 5(3):292~308
- 8 Garbacki P, Epema D H J, Steen M. The design and evaluation of a self-organizing superpeer network. IEEE Transactions on Computers, 2010, 59(3):317~331
- 9 Doukeridis C, Vlachou A, Norvag K, *et al.* Efficient search based on content similarity over self-organizing P2P networks. Peer-to-Peer Networking and Applications, 2009, 3(1):67~79
- 10 Ayorak E, Bener A B. Super peer web service discovery architecture. Proceedings of IEEE the 23rd International Conference on Data Engineering, Istanbul, Turkey, 2007: 1360~1364
- 11 朱海, 王宇平. 融合安全的网格依赖任务调度双目标优化模型及算法. 软件学报, 2011, 22(11)

- 2 Commission F C. Notice of proposed rule making and order: facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. ET Docket, 2005(3/108)
- 3 DARPA X Working Group. The XG Vision Request for Comments (Version 2.0), 2003
- 4 Wang B, Liu K. Advances in cognitive radio networks: a survey. IEEE Journal of Selected Topics in Signal Processing, 2011, 5(1): 5~23
- 5 郭永明, 李宁, 李军芳. 未来智能无线电将对频谱管理带来深刻影响. 电信科学, 2012, 28(7): 94~100
- 6 Li H, Han Z. Socially optimal queuing control in cognitive radio networks subject to service interruptions: to queue or not to queue. IEEE Transactions on Wireless Communications, 2011, 10(5): 1656~1666
- 7 郭彩丽, 曾志民, 冯春燕等. 机会频谱接入系统的切换请求排队机制及性能分析. 电子与信息学报, 2009, 31(6): 1505~1508
- 8 Hu J, Yang L L, Hanzo L. Maximum average service rate and optimal queue scheduling of delay-constrained hybrid cognitive radio in nakagami fading channels. IEEE Transactions on Vehicular Technology, 2013, 62(5): 2220~2229
- 9 Boon M, Van Der Mei R, Winands E. Applications of polling systems. Surveys in Operations Research and Management Science, 2011, 16(2): 67~82
- 10 杨志军, 赵东风, 丁洪伟等. 两级优先级控制轮询系统研究. 电子学报, 2009(7): 1452~1456
- 11 Guan Z, Zhao D, Zhao Y. A discrete time two-level mixed service parallel polling model. Journal of Electronics (China),

2012, 29(1): 103~110

[作者简介]



鲁进, 男, 云南大学博士研究生, 主要研究方向为认知无线电、轮询系统和多址接入。



黄铭, 男, 博士, 云南大学教授、博士生导师, 主要研究方向为频谱传感与无线电监测。



杨志军, 男, 博士, 云南省教育厅教育科学研究院副院长、高级工程师, 主要研究方向为计算机通信与网络、无线通信和轮询系统。

(收稿日期: 2013-05-02)

(上接第 59 页)

- 12 Li J, Vuong S. A scalable semantic routing architecture for grid resource discovery. Proceedings of the 11th International Conference on Parallel and Distributed Systems, Fukuoka, Japan, 2005: 29~35
- 13 Lser A, Naumann F, Siberski W, *et al.* Semantic overlay clusters within super-peer networks. Databases, Information Systems and Peer-to-Peer Computing, 2004(2944): 35~38
- 14 Hassan M I, Abdullah A. A semantic description and registration framework for large grid resource discovery systems. Computer Science Letters, 2009, 1(1)
- 15 Hassan M I, Abdullah A. A new grid resource discovery framework. The International Arab Journal of Information Technology, 2011, 8(1): 99~107
- 16 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
- 17 孙瑞志, 杨璐, 欧阳娅. 基于改进遗传算法的网格任务调度. 解放军理工大学学报(自然科学版), 2012, 13(4)
- 18 Jensen K. Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Berlin: Springer, 1996

- 19 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, 宁波大学计算机科学技术研究所副教授, 主要研究方向为服务计算、信息检索。

黄希, 男, 宁波大学计算机科学技术研究所硕士研究生, 主要研究方向为网格计算, Petri 网理论与应用。

茅琴娇, 女, 西安交通大学电子与信息工程学院博士研究生, 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: 2013-07-31)