

# 基于 $a + bi$ 型联系数的不确定网格静态调度算法<sup>\*</sup>

黄德才<sup>1</sup> 张丽君<sup>1</sup> 赵克勤<sup>2</sup>

(浙江工业大学信息工程学院 杭州 310032)<sup>1</sup> (诸暨市联系数学研究所 诸暨 311811)<sup>2</sup>

**摘要** 任务调度算法是计算网格任务管理系统中的核心问题。由于网格环境中存在大量的不确定因素,导致传统网格调度算法和调度系统不能在开放、异构和动态的真实网格环境中有效运行。利用一种新的软计算方法——集对分析联系数研究和处理网格调度中的综合不确定性问题。在简单介绍集对分析概念和应用情况基础上,引入联系数概念、运算规律和全序关系,分别提出了基于联系数的不确定网格静态调度算法 CBU\_Min-min、CBU\_Max-min 和 CBU\_Surferage,并进行了数值仿真实验研究。理论和实验研究表明,这些算法能较好地描述网格任务预期执行时间的动态性和不确定性,并使传统网格调度方法成为其特例,在动态和不确定网格环境中有良好的理论和实际应用价值。

**关键词** 计算网格,不确定性,任务调度,调度算法

## Static Scheduling Algorithms Based on Connective-number of Type $a + bi$ for Uncertain Computing Grid

HUANG De-Cai<sup>1</sup> ZHANG Li-Jun<sup>1</sup> ZHAO Ke-Qin<sup>2</sup>

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310032)<sup>1</sup>

(Zhuji Institute of Connective Mathematics, Zhuji 311811)<sup>2</sup>

**Abstract** Job scheduling algorithms are kernel technique in task management system of computing grid. Because the dynamic and uncertainty exist in grid environment, the traditional job scheduling algorithms cannot be applied effectively in the real open, heterogeneous and dynamic grid environment, and let the job scheduling system do not work well. Using connective number of Set Pair Analysis( SPA ), a new soft computation method to express & process the synthetic uncertainty in task scheduling of the computing grid. After introducing SPA and its application briefly, present the definition of connective number, operational rules and total order relation suitable for computing grid scheduling, three static scheduling algorithms, CBU\_Min-min, CBU\_Max-min and CBU\_Surferage, are presented for the uncertain computing grid. Theory analysis and numerical experiment illustrate that these algorithms can express the dynamics and uncertainty of expected time to compute of tasks in the computing grid environment, they are the generalization of traditional grid scheduling algorithms, and there are high value in theory and application in the dynamic and uncertain grid environment.

**Keywords** Computing grid, Uncertainty, Task scheduling, Scheduling algorithm

### 1 引言

网格技术已成为下一代互联网应用的关键技术,也是当今计算机领域的前沿研究课题。虽然有计算网格、数据网格、存储网格、知识网格和服务网格等多种类型,但不论什么样的网格,任务调度系统都是其发挥潜在性能和优势所共有的核心系统,且任务管理、任务调度和资源管理是其必备的三种基本功能<sup>[1,2]</sup>,而任务调度模型及其优化算法则是网格调度必须解决的核心问题和关键技术。调度算法的好坏,将直接影响网格的信息传递效率、资源利用率、网格可靠性和应用程序的执行时间等<sup>[3]</sup>。虽然对网格任务调度算法的研究已取得不少成果<sup>[3-7]</sup>,但这些算法却都没有考虑网格环境的动态性和不确定性,导致当前的任务调度系统难以在开放、异构和动态的真实网格环境中有效运行。

网格系统与传统并行高性能计算的最大不同之处在于其动态性,且如何处理这种动态性被认为是网格的“十大”问题之一<sup>[8]</sup>。由于网格环境中存在大量的不确定因素,比如资源的共享性、资源的随时进入和退出、通信带宽和延迟差异性等,都导致计算资源的性能动态变化,正如文[9]所指出的那样,网络流量和资源的当前负载都是动态变化的,它们反映出的结果就直接影响任务的执行速度,即任务的预期执行时间是不确定的。因此,文[9]提出具有模糊处理时间的人工免疫网格任务调度算法,文[10]利用随机方法来预测任务的执行时间。然而,由于网格的广域、异构和动态性,网格任务调度中的不确定性已经不是单纯的模糊或随机不确定性,而是由随机、模糊、不确知、中介和突发等不确定性导致的综合不确定性问题,任何单一的不确定性方法都难于真实描述和表达这种综合不确定性问题。正是在这一点上,由我国学者赵克

<sup>\*</sup> 基金项目 浙江省自然科学基金资助项目(Y105118&Y105109)。黄德才 工学博士,教授,博士生导师,主要研究方向有网格调度、人工智能、图像处理、数据挖掘等;张丽君 硕士研究生,主要研究方向有网格调度和数据挖掘;赵克勤 高级工程师,理学学士,主要研究集对分析、联系数学及其应用等。

勤提出的集对分析及其联系数<sup>[11]</sup>在表示由随机、模糊、不确定、中介和突发等不确定性导致的综合不确定问题方面有独到之处,并在网络计划、气象预报、不完备信息系统和产品设计等许多领域得到应用<sup>[12-16]</sup>。

本文利用集对分析<sup>[11]</sup>(Set Pair Analysis, 简记 SPA, 又称为联系数学)这一新的数学工具来研究和处理网格调度的这种综合不确定性问题。在简单介绍集对分析概念和应用情况基础上,引入联系数概念、运算规律和全序关系,并借鉴传统网格任务静态调度算法 Min-min、Max-min、Sufferage 的思想,分别提出了基于联系数的不确定网格静态调度算法 CBU\_Min-min、CBU\_Max-min、CBU\_Sufferage(CBU 即为 Connected-number Based Uncertain 的首字母),并进行了数值仿真实验研究。理论和实验研究表明,这些算法能较好地描述网格任务预期执行时间的动态性和不确定性,符合客观实际,并使传统网格调度方法成为其特例,不仅在动态和不确定网格环境中有良好的理论和实际应用价值,还将成为网格任务调度理论建模的一个新的研究方法。

## 2 联系数运算及全序关系

### 2.1 联系数运算

集对分析是一种新的软计算方法<sup>[16]</sup>,而联系数则是集对分析中的重要概念之一<sup>[11]</sup>。引进联系数的最初目的是为了应用上的方便,其理论意义则在于拓宽了数的概念。一方面它把可确定数与所在范围的数与值联系起来,另一方面它把宏观层次上的确定量和微观层次上的不确定量联系起来。虽然联系数  $u = a + bi + cj$  的表达形式是统一的,但  $a$ 、 $b$ 、 $c$  和  $i$ 、 $j$  的语义可根据所研究问题的实际背景具体定义和解释。因此,本文根据计算网格中任务预期执行时间(ETC, Expected Time to Compute)的不确定性特点,重新定义网格调度问题中的联系数为当  $c = 0$  时的  $a + bi$  型联系数。

定义 1 称  $u = a + bi$  为网格调度问题中表示某个任务预期执行时间的联系数,其中  $a$  表示该任务在通常情况下的预期执行时间,  $b$  为网格环境不确定性引起的时间波动,  $i$  在区间  $[-1, 1]$  内不确定取值。特别当  $a = b = 0$  时,记联系数  $u = 0$ 。

比如,某项任务在一个节点上计算完成通常情况下需要 100s,但由于该节点上并发执行的其它任务退出或新任务的进入,则最多可能提前 31s 或延迟 31s 完成,因此可以用联系数  $u = 100 + 31i$  来表示该任务在这种不确定情况影响下的预期执行时间。

下面引入联系数的四则运算规则。

定义 2<sup>[12]</sup> 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,则它们之和是一个联系数  $u = a + bi$ ,记作  $u_1 + u_2$ ,其中  $a = a_1 + a_2$ ,  $b = b_1 + b_2$ 。

从定义 2 可以看出,联系数的加法运算满足交换律和结合律。下面,我们引进联系数的减法运算。

定义 3<sup>[12]</sup> 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,则它们之差是一个联系数  $u = a + bi$ ,记作  $u_1 - u_2$ ,其中  $a = a_1 - a_2$ ,  $b = b_1 - b_2$ 。

由定义 1、定义 2 和定义 3 容易得出如下推论。

推论 1 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,则  $u_1 = u_2$  的充分必要条件是  $a_1 = a_2$ ,  $b_1 = b_2$ 。

定义 4 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,则它们之积是一个联系数  $u = a + bi$ ,记作  $u_1 \times u_2$ ,其中  $a = a_1 a_2$

$$+ b_1 b_2, b = a_1 b_2 + a_2 b_1。$$

定义 5 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,若存在一个联系数  $u = a + bi$ ,使得  $u_1 = u \times u_2$ ,则称  $u$  是  $u_1$  除以  $u_2$  的商,记作  $u = u_1 \div u_2$ 。这时称  $u_1$  能够除以  $u_2$ ,否则称  $u_1$  不能够除以  $u_2$ 。

此外,我们容易证明如下定理。

定理 1 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数,则  $u_1$  能够除以  $u_2$  的充分必要条件是以下矩阵  $M$  是非奇异的。

$$M = \begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}$$

且  $u_1$  除以  $u_2$  的商  $u = a + bi$  中的  $a$ 、 $b$  为以下线性方程组的解。

$$\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$$

此外,我们可以证明联系数乘法的一些性质,如交换律、结合律等。限于篇幅不予赘述。

### 2.2 调度问题与全序关系

粗粒度元任务的调度问题是网格任务调度中最基本的调度问题,许多其它的调度模型都是由它修正和推广而来。它假设有  $n$  项独立任务  $T = \{t_1, t_2, \dots, t_n\}$  和  $m$  个计算资源节点  $M = \{M_1, M_2, \dots, M_m\}$ ,任务  $t_i$  在节点  $M_j$  上的预期执行时间是  $E(i, j)$ ,调度目标是极小化 makespan。由于这个看似简单且无约束的单目标调度问题也是一个 NP-Hard 问题,因此,人们提出了许多启发式算法,如著名的静态调度算法 Max-min、Min-min 和 Sufferage 算法等。根据以元任务调度问题的定义,本文讨论基于联系数的不确定网格调度问题叙述如下:

设有  $n$  项独立任务  $T = \{t_1, t_2, \dots, t_n\}$  和  $m$  个计算机资源节点  $M = \{M_1, M_2, \dots, M_m\}$ ,任务  $t_p$  在节点  $M_q$  上的预期执行时间  $E(p, q)$  是不确定的,可用联系数表示为  $E(p, q) = a_{pq} + b_{pq}i$ ,并构成一个  $n \times m$  的 ETC 矩阵  $(E(p, q))_{n \times m}$ 。任务调度的目标是把  $T$  中任务安排到  $M$  中的节点上执行,使其总的执行时间(makespan)尽可能地“小”。

由于每个任务的预期执行时间是由联系数表达的不确定量,且根据联系数的运算法则,以上调度问题的总执行时间也是联系数,因此,如何比较联系数的“大”、“小”,即定义联系数的全序关系是建立基于联系数的不确定网格调度模型的关键。

因为联系数表达的是不确定量,一般可以根据实际应用问题的不同需要,定义不同的序关系。文[11]根据联系数中  $a/b$  的比值  $a/b$  是小于 1、大于 1 或是等于 1 的不同情况定义了联系数的同异反态势关系,但这种关系仅是偏序而非全序关系。文[12]根据不确定网络计划的需要,对联系数集合通过定义主关键联系数的概念给出了联系数的一种偏序关系,并在网络计划关键路径识别方面得到应用,但仍然不是全序关系,致使大多数联系数仍然无法进行比较。

下面我们结合网格调度的实际需要,给出联系数的两种全序关系。

定义 6 设  $u_1 = a_1 + b_1i$ ,  $u_2 = a_2 + b_2i$  是两个联系数。如果  $a_1 < a_2$  或  $a_1 = a_2$  且  $b_1 < b_2$ ,称  $u_1$  小于  $u_2$ ,记作  $u_1 <_p u_2$ 。

显然,若用  $R_c$  表示所有联系数的集合,则以上定义的小于关系  $<_p$  是  $R_c$  上的全序关系,它首先对确定性部分  $a_1$  和  $a_2$  进比较,因为确定部分执行时间是相对确定的,其优先级高于不确定部分;如果  $a_1 = a_2$ ,则对不确定部分  $b_1$  和  $b_2$  进行比较。这个全序关系是一种最坏情形的全序关系,也称为悲观

估计下的全序关系 因为它对不确定性部分进行比较时以  $b_1 < b_2$  确定  $u_1 <_p u_2$ 。

根据定义 6 我们容易证明如下定理。

定理 2 设  $R_c$  为所有联系数的集合,则对于任意  $u_1, u_2 \in R_c$  有,以下三个结论中有且恰有一个成立。

(1)  $u_1 = u_2$  (2)  $u_1 <_p u_2$  (3)  $u_1 >_p u_2$

下面定义一种乐观估计下的全序关系。

定义 7 设  $u_1 = a_1 + b_1 i$   $u_2 = a_2 + b_2 i$  是两个联系数。如果  $a_1 < a_2$  或  $a_1 = a_2$  且  $b_1 > b_2$  称  $u_1$  小于  $u_2$ , 记作  $u_1 <_o u_2$ 。

类似定理 2 我们可得如下定理 3。

定理 3 设  $R_c$  为所有联系数的集合,则对于任意  $u_1, u_2 \in R_c$ ,以下三个结论中有且恰有一个成立。

(1)  $u_1 = u_2$  (2)  $u_1 <_o u_2$  (3)  $u_1 >_o u_2$

在本文以下部分讨论中,当提到联系数或所表示的时间进行比较时的“最小”、“最大”、“小于”、“大于”或“最短”、“最长”等,都是指在全序关系  $<_p$  或  $<_o$  意义下的“最小”、“最大”等概念。而在一个算法中究竟选择哪一种全序关系,完全取决于对系统状态的宏观估计。如果根据最近网络系统运行的实际情况得知,实际运行的时间通常比静态调度的时间长,则宜选用悲观估计的序关系,反之就应该选用乐观估计的序关系。

### 3 静态调度算法

#### 3.1 参数定义

(1)  $M_q$ : 既表示编号为  $q$  的计算资源(节点),也表示该节点上所分配任务的集合,且按照任务分配的先后次序存放。

(2)  $S_q$ : 节点  $M_q$  在完成已分配给它的所有任务后,预期可执行另外一个新任务的开始时间;

(3)  $E(p, q)$ : 当任务  $t_p$  分配给可用节点  $M_q$  的预期执行时间,形成的矩阵称为 ETC 矩阵;

(4)  $\alpha(p, q)$ : 若现在将新任务  $t_p$  分配给可用节点  $M_q$  的预期完成时间  $\alpha(p, q) = S_q + E(p, q)$ ;

(5)  $CT$ : 任务与资源匹配形成的二元组,任务,资源集合,也称任务-资源对集合;

(6)  $Assigned_q$ : 在一次循环中,节点  $M_q$  是否已经分配一个新任务,取值 True 或 False;

#### 3.2 静态调度算法

##### (1) 算法 1 CBU\_Max-min 算法

输入: 任务集  $T$ , 节点集  $M$  和基于联系数的 ETC 矩阵;

输出: 任务与资源的映射方案  $M_1, M_2, \dots, M_m$ ;

Step0. for  $M$  中的每一个节点  $M_q$

$$M_q = \phi, S_q = 0$$

endfor

Step1. Repeat

$$CT = \phi$$

Step2. for  $T$  中每一个作业  $t_k$

Step3. for  $M$  中每一个节点  $M_q$

$$\alpha(p, q) = S_q + E(p, q);$$

Step4. endfor

找出使任务  $t_p$  完成时间“最短”的资源  $M_q, CT = CT \cup \{t_p, M_q\}$

Step5. endfor

Step6.  $C_{\max} = \max\{\alpha(p, q) \mid t_p, M_q \in CT\}$

$$M^{\max} = \{M_k \mid \alpha(p, k) = C_{\max} \text{ 且 } t_p, M_k \in CT\}$$

Step7. 在  $M^{\max}$  中任选节点  $M_k$  及其对应的  $t_r, M_k$ , 令

$$M_k = M_k \cup \{t_r\}, T = T - \{t_r\}$$

$$S_k = S_k + E(r, k)$$

Step8. Until(  $T = \phi$  )

##### (2) 算法 2 CBU\_Min-min 算法

输入: 任务集  $T$ , 节点集  $M$  和基于联系数的 ETC 矩阵;

输出: 任务与资源的映射方案  $M_1, M_2, \dots, M_m$ ;

Step0. for  $M$  中的每一个节点  $M_q$

$$M_q = \phi, S_q = 0$$

endfor

Step1. Repeat

$$CT = \phi$$

Step2. for  $T$  中每一个作业  $t_k$

Step3. for  $M$  中每一个节点  $M_q$

$$\alpha(p, q) = S_q + E(p, q);$$

Step4. endfor

找出使任务  $t_p$  完成时间“最短”的资源  $M_q, CT = CT \cup \{t_p, M_q\}$

Step5. endfor

Step6.  $C_{\min} = \min\{\alpha(p, q) \mid t_p, M_q \in CT\}$

$$M^{\min} = \{M_k \mid \alpha(p, k) = C_{\min} \text{ 且 } t_p, M_k \in CT\}$$

Step7.  $M^{\min}$  中任选一个节点  $M_k$  及其对应的  $t_r, M_k$

$$M_k = M_k \cup \{t_r\}, T = T - \{t_r\}$$

$$S_k = S_k + E(r, k)$$

Step8. Until(  $T = \phi$  )

##### (3) 算法 3 CBU\_Sufferage

输入: 任务集  $T$ , 节点集  $M$  和基于联系数的 ETC 矩阵;

输出: 任务与资源的映射方案  $M_1, M_2, \dots, M_m$ ;

Step0. for  $M$  中的每一个节点  $M_q$

$$S_q = 0, Assigned_q = False, M_q = \phi$$

endfor

Step1. Repeat

Step2. for  $T$  中的每一个作业  $t_p$

Step3. for  $M$  中每一个节点  $M_q$

$$\alpha(p, q) = S_q + E(p, q);$$

Step5. endfor

Step6. for  $M$  中每一个节点  $M_q$  找出使  $t_p$  完成时间“最短”的资源  $M_r$  和对应的完成时间  $\alpha(p, r)$ ;

Step7. 找出使  $t_p$  完成时间“次短”的资源  $M_s$  和对应的完成时间  $\alpha(p, s)$ ;

$$SufferV_p = \alpha(p, s) - \alpha(p, r)$$

Step8. endfor

Step9. if 任务  $t_p$  对应的节点  $M_s$  的  $Assigned_s = False$

$$M_q = M_q \cup \{t_p\}, T = T - \{t_p\};$$

$$Assigned_q = True;$$

Step11. else if  $t_k$  是本次循环已分配给节点  $M_q$  的任务且  $SufferV_p$

“小于”  $SufferV_k$ ,

$$M_q = M_q \cup \{t_p\} - \{t_k\}$$

$$T = T \cup \{t_k\} - \{t_p\};$$

endif.

Step12. for  $M$  中的每一个节点  $M_q$

$$S_q = \sum_{t_p \in M_q} E(p, q)$$

$$Assigned_q = False$$

endfor

Step13. Until(  $T = \phi$  )

显然,如果算法 1、算法 2 和算法 3 在 Step0 中令参数  $S_q \neq 0$  则它们就成为批模式下的在线调度算法。

### 4 算例与仿真分析

#### 4.1 计算实例

为说明算法的计算过程并对调度结果进行预测分析,下面给出一个简单的例子。

例 1 设计算资源数  $m = 3$ , 任务数  $n = 5$ , 每个任务在各个计算节点上的预期不确定执行时间矩阵 ETC 如下, 试在联系数的悲观序关系条件下用 CBU\_Min-min 算法完成其任务调度。

$$ETC = \begin{pmatrix} 101 + 32i & 100 + 22i & 102 + 17i \\ 100 + 3i & 99 + 9i & 99 + 7i \\ 101 + 12i & 99 + 6i & 98 + 3i \\ 100 + 5i & 102 + 8i & 101 + 6i \\ 97 + 16i & 101 + 20i & 97 + 19i \end{pmatrix}$$

解 根据 CBU\_Min-min 算法的计算过程,可知每次循环完成一个任务的指派,其具体过程如下:

1. 由于  $(S_1, S_2, S_3) = (0, 0, 0)$ , 而各任务的“最小”完成时间集合  $CT = \{\alpha(1, 2) = 100 + 22i, \alpha(2, 3) = 99 + 7i, \alpha(3, 3) = 98 + 3i, \alpha(4, 1) = 100 + 5i, \alpha(5, 1) = 97 + 16i\}$ , 由定义 6 和算法,将任务  $t_5$  安排到节点  $M1$   $(S_1, S_2, S_3) = (97 + 16i, 0, 0)$ 。

2. 对未安排的  $t_1, t_2, t_3, t_4$ , 计算任务的最小完成时间集合  $CT = \{\alpha(1, 2) = 100 + 22i, \alpha(2, 3) = 99 + 7i, \alpha(3, 3) = 98 + 3i, \alpha(4, 3) = 101 + 6i\}$ , 同理可知应将任务  $t_3$  安排到节点  $M_3$  执行, 得  $(S_1, S_2, S_3) = (97 + 16i, 98 + 3i)$ ;

3. 对未安排的  $t_1, t_2, t_4$ , 计算任务的最小完成时间集合  $CT = \{\alpha(1, 2) = 100 + 22i, \alpha(2, 2) = 99 + 9i, \alpha(4, 2) = 102 + 8i\}$ , 同理将任务  $t_2$  安排到节点  $M_2$  执行, 得  $(S_1, S_2, S_3) = (97 + 16i, 99 + 9i, 98 + 3i)$ ;

4. 对未安排的  $t_1, t_4$ , 计算任务的最小完成时间集合  $CT = \{\alpha(1, 1) = 198 + 48i, \alpha(4, 1) = 197 + 21i\}$ , 同理将任务  $t_4$  安排到节点  $M_1$  执行, 得  $(S_1, S_2, S_3) = (197 + 21i, 99 + 9i, 98 + 3i)$ ;

5. 对未安排的  $t_1$ , 其最小完成时间为  $C(1, 2) = 199 + 31i$ , 因此将任务安排到节点  $M_2$  执行, 得  $(S_1, S_2, S_3) = (197 + 21i, 199 + 31i, 98 + 3i)$ 。

按照定义 6 可知, 该系统的 makespan =  $199 + 31i$ 。这个调度结果说明, 在正常情况下, 所有任务可以在 199 单位时间内完成。如果系统中多数节点上并发执行的其它任务退出或新任务的进入, 或者受其它不确定因素的影响, 则最多可能提前 31 或延迟 31 个单位时间完成, 但具体的完成时间应该根据不确定量  $i$  的取值来进行预测和分析。比如, 如果系统最近一段时间运行都比较慢, 则可以选择悲观估计的最坏情形, 即取  $i = 1$ , 这时系统的 makespan = 230; 也可以采用“面向式子”的取值法<sup>[11]</sup>, 取  $i = 31 / (199 + 31) \approx 0.1348$ , 因此, 在  $i$  取 0.1348 这种情形的悲观估计条件下, 系统的 makespan = 203.1788。关于  $i$  的取值还有动态取值法、概率取值法、专家取值法和综合取值法等<sup>[10]</sup>。在实际应用中一般要根据网格系统运行的实际状况来选择合适的取值方法, 并对系统执行任务的时间进行预测和分析, 也正是在这一点上, 基于联系数的网格任务调度算法更能体现网格的动态性和不确定性。

## 4.2 仿真分析

### 1. 数据产生方法

仿真实验考察 20 个计算资源组成的网格系统对 40 ~ 200 个独立任务构成的任务集调度情况。

任务在计算资源上的预期执行时间  $ETC$  根据文[8]中的方法来生成  $ETC$  矩阵中的  $a_{pq}$ , 然后根据  $a_{pq}$  值的大小, 利用均匀分布产生  $b_{pq}$ 。文[8]中的方法需要输入 4 个参数, 设控制数据均值变化的参数  $\mu_{task} = 100$ ,  $\mu_{mach} = 100$ , 另外的两个参数  $V_{task}$  和  $V_{mach}$  是用来控制任务和机器一致性的, 其值越高, 任务和机器的异构性越强, 将二者取值  $V_{task} = 0.1$  或  $V_{task} = 0.6$ ,  $V_{mach} = 0.1$  或  $V_{mach} = 0.6$ , 这样使得机器和任务分别达到较高或较底的异构性, 可用来测试异构性的高低对调度是否有大的影响。

### 2. 实验结果与性能分析

本文采用悲观估计的偏序关系情形下的任务的平均 makespan 值对改进后的新启发式算法进行综合评价, 每个绘图采用的是 50 次实验的平均值。

在实验结果中, 机器数取 20, 用字母  $m$  表示; 同时, 用符号 ' $V_t$ ' 表示 ' $V_{task}$ ', 用符号 ' $V_m$ ' 表示 ' $V_{mach}$ '。

因为仿真比较发现 CBU-Max-min 算法的性能比 CBU-Min-min 算法及 CBU-Sufferage 算法的性能要差很多, 所以我们没有将 CBU-Max-min 算法的结果放在图中进行比较。

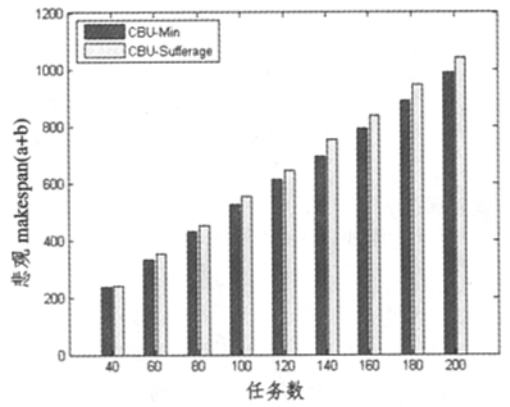


图 1  $m = 20, p_t = 0.1, p_m = 0.1$

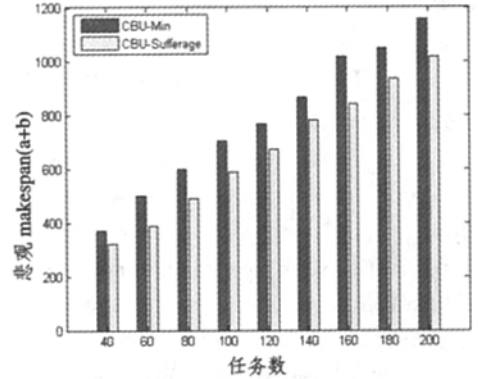


图 2  $m = 20, p_t = 0.6, p_m = 0.1$

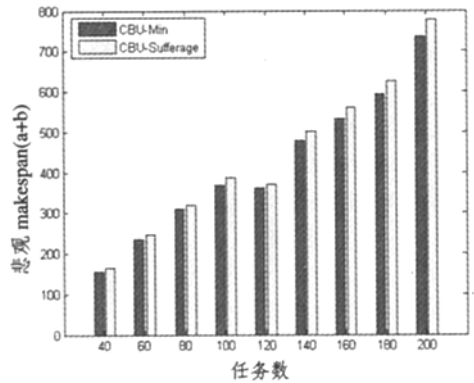


图 3  $m = 20, p_t = 0.1, p_m = 0.6$

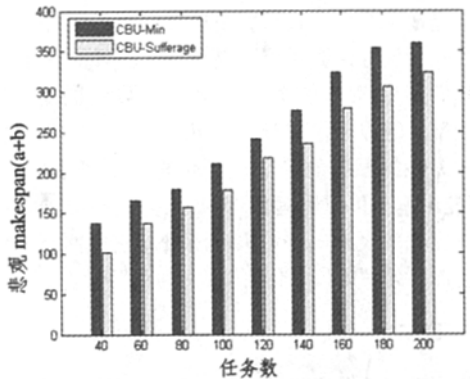


图 4  $m = 20, p_t = 0.6, p_m = 0.6$

从图 1、图 3 可以看出, 在非一致的  $ETC$  矩阵条件下, 当  
(下转第 179 页)

本实例是用来比较文本证据权 WET 和  $\chi^2$  统计量两种特征选择算法在特定环境下的效果。在分词接口实现正向匹配算法,在特征选择接口实现两种方案,在分类模块中选择 Naive Bayes 算法,利用宏平均精度来展示分类精度,预料库利用复旦大学提供的测试语料库,采用 10-fold cross validation 的方法。

表 1 分类测试方法比较

分词算法	特征选择算法	文本分类算法
正向最大匹配法	文本证据权 WET	Naive Bayes
	$\chi^2$ 统计量	

下表是比较结果。

表 2 分类测试结果比较

特征选择算法	score	point
$\chi^2$ 统计量 选取 40% 特征	79.87%	88.19
WET 选取 40% 特征	82.27%	100.90
WET 选取 10% 特征	75.15%	67.68

结果显示,在选择相同数目特征的情况下,文本证据权 WET 算法的特征选择算法相对于  $\chi^2$  统计量算法表现了较好的分类精度。同时,文本证据权 WET 算法在选取特征比较少的情况下,分类结果下降较明显。

**总结与展望** 本文介绍了一个中文文本自动分类评估体系,用于对中文文本自动分类过程中使用到的分词算法、特征选择算法以及分类算法进行综合评价,并实现了一个中文文本自动分类研究的开放平台,供人们实现和比较各种算法。

(上接第 129 页)

任务的异构性比较低时,CBU-Min-min 算法的性能比 CBU-Sufferage 算法的性能要好(这一点与传统方法认为 Sufferage 算法总是比 Min-min 算法好的结论不一样),但差距不是很大;从图 2 和图 4 可以看出,当任务的异构性较高时,CBU-Min-min 算法的性能没有 CBU-Sufferage 算法的性能好,且二者的性能差异较大。

**总结** 由于网格的广域、异构和动态特性,网格任务调度存在随机、模糊、不确定、中介和突发等多种不确定因素,任何单纯的不确定性方法都难于真实描述和表达这种综合不确定性问题。本文利用集对分析联系数来研究和处理网格调度的这种综合不确定性问题,并借鉴传统网格任务静态调度算法提出了相应的不确定网格静态调度新算法。研究结果表明,这些算法能较好地描述网格任务预期执行时间的动态性和不确定性,不仅在动态和不确定网格环境中有良好的理论和实际应用价值,还将成为网格任务调度理论建模的一个新的研究方法。

### 参考文献

- 1 Foster I, Kesselman C 著. 金海,袁平鹏,石柯译. 网格计算[M] (第 2 版). 北京:电子工业出版社,2004
- 2 Foster I, Kesselman C. The Anatomy of the Grid: Enabling Scalable Virtual Organizations[J]. International Journal of Supercomputer Applications, 2001, 15(3): 200~222
- 3 罗红,慕德俊,邓智群,王晓东. 网格计算中任务调度研究综述[J]. 计算机应用研究, 2005(5): 16~19

利用该平台,得出了若干已有中文文本自动分类算法的实验结果,评价了不同特征选择算法和不同参数对分类的影响。

目前中文方面并没有一个公开的、相对标准的语料库,所以本评估体系的下一步方向就是利用自动下载工具对权威网站进行定期下载,建立一个标准的语料库,并实现语料库的动态更新。

### 参考文献

- 1 Yang Yiming, Liu Xin. A Re-Examination of Text Categorization Methods. In: 22nd Annual International SIGIR. 1999. 42~49
- 2 刘延章,余义芳. 近五年来网络信息分类组织研究的现状及其展望. 情报学报, 2004, 23(2)
- 3 张春霞,郝天勇. 汉语自动分词的研究现状与困难. 系统方学学报, 2005(1)
- 4 孙茂松,邹嘉彦. 汉语自动分词研究评述. 当代语言学, 2001(1): 22~32
- 5 代六玲,黄河燕,陈肇雄. 中文文本分类中特征抽取方法的比较研究. 中文信息学报, 2004, 18(1): 26~32
- 6 Wang Yi, Wang Xiao-Jing. A new approach to feature selection in text classification, Machine Learning and Cybernetics, 2005. In: Proceedings of 2005 International Conference on, Aug. 2005 p 3814~3819
- 7 李凡,鲁明羽,陆玉昌. 关于文本特征抽取新方法的研究. 清华大学学报(自然科学版), 2001, 41(7): 98~101
- 8 Sebastiani F. Machine Learning in Automated Text Categorization. In: 18th International Conference on Computational Linguistics (COLING '00), Nancy, France, July 2000
- 9 Jain G, Ginwala A, Aslandogan Y A. An approach to text classification using dimensionality reduction and combination of classifiers, Information Reuse and Integration, 2004. IRI 2004. In: Proceedings of the 2004 IEEE International Conference on, Nov. 2004. 564~569
- 10 Wang Baoyi, Zhang Shaomin. A Novel Text Classification Algorithm Based on Naive Bayes and KL-Divergence, Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. In: Sixth International Conference on, Dec. 2005. 913~915

- 4 Brauny T D, Siegely H J, Becky N, et al. A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems[C]. In: Proceedings of the 8th Heterogeneous Computing Workshop (HCW '99), Apr. 1999. 15~29
- 5 陈志刚,刘安丰,熊策,张连明. 一种有效负载均衡的网格 Web 服务体系结构模型[J]. 计算机学报, 2005, 28(4): 458~466
- 6 刘安丰,陈志刚,陆静波,张连明. 网格环境中一种有效的 Web 服务资源组织机制[J]. 计算机研究与发展, 2004, 41(12): 2141~2147
- 7 张伟哲,刘欣然,云晓春,等. 信任驱动网格作业调度算法[J]. 通信学报, 2006, 27(2): 73~79
- 8 Schopf J M, Nitzberg B. Grids: Top Ten Questions. Scientific Programming, special issue on Grid Computing, 2002, 10(2): 103~111
- 9 李季,钟将,吴中福. 具有模糊处理时间的网格任务调度免疫算法[J]. 计算机科学, 2006, 33(2): 35~38
- 10 赵克勤. 集对分析及其初步应用[M]. 杭州:浙江科学技术出版社, 2000
- 11 黄德才,赵克勤. 用联系数描述和处理网络计划中的不确定性[J]. 系统工程学报, 1999, 14(2): 112~117
- 12 薛根元,王国强. 不确定性理论集对分析在预报模型建立中的应用研究[J]. 气象学报, 2003, 61(5): 592~599
- 13 黄兵,周献中. 不完备信息系统中基于联系度的粗集模型拓展[J]. 系统工程理论与实践, 2004, 24(1): 88~72
- 14 李志辉,夏少云,查建中. 基于案例推理的同异反产品设计及其应用[J]. 计算机辅助设计与图形学学报, 2003, 15(11): 1397~1403
- 15 蒋云良,徐从富. 集对分析理论及其应用研究进展[J]. 计算机科学, 2006, 33(1): 205~209
- 16 Ali S, Siegel H J, Maheswaran M, Hensgen D, Ali S. Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems[J]. Tamkang Journal of Science and Engr, 2000, 3(3): 195~207

# 基于a+bi型联系数的不确定网络静态调度算法

作者: [黄德才](#), [张丽君](#), [赵克勤](#), [HUANG De-Cai](#), [ZHANG Li-Jun](#), [ZHAO Ke-Qin](#)  
 作者单位: [黄德才,张丽君,HUANG De-Cai,ZHANG Li-Jun\(浙江工业大学信息工程学院,杭州310032\)](#), [赵克勤,ZHAO Ke-Qin\(诸暨市联系数学研究所,诸暨311811\)](#)  
 刊名: [计算机科学](#) **ISTIC** **PKU**  
 英文刊名: [COMPUTER SCIENCE](#)  
 年,卷(期): 2007, 34(8)  
 被引用次数: 1次

## 参考文献(16条)

1. [Foster I, Kesselman C, 金海, 袁平鹏, 石柯](#) [网络计算](#) 2004
2. [Foster I, Kesselman C](#) [The Anatomy of the Grid:Enabling Scalable Virtual Organizations](#) 2001(03)
3. [罗红, 慕德俊, 邓智群, 王晓东](#) [网络计算中任务调度研究综述](#)[期刊论文]-[计算机应用研究](#) 2005(05)
4. [Braun T D, Siegely H J, Becky N](#) [A Comparison Study of Static Mapping Heuristics for a Class of Meta-tasks on Heterogeneous Computing Systems](#) 1999
5. [陈志刚, 刘安丰, 熊策, 张连明](#) [一种有效负载均衡的网格Web服务体系结构模型](#)[期刊论文]-[计算机学报](#) 2005(04)
6. [刘安丰, 陈志刚, 陆静波, 张连明](#) [网格环境中一种有效的Web服务资源组织机制](#)[期刊论文]-[计算机研究与发展](#) 2004(12)
7. [张伟哲, 刘欣然, 云晓春](#) [信任驱动网格作业调度算法](#)[期刊论文]-[通信学报](#) 2006(02)
8. [Schopf J M, Nitzberg B](#) [Grids:Top Ten Questions](#) 2002(02)
9. [李季, 钟将, 吴中福](#) [具有模糊处理时间的网格任务调度免疫算法](#)[期刊论文]-[计算机科学](#) 2006(02)
10. [赵克勤](#) [集对分析及其初步应用](#) 2000
11. [黄德才, 赵克勤](#) [用联系数描述和处理网络计划中的不确定性](#)[期刊论文]-[系统工程学报](#) 1999(02)
12. [薛根元, 王国强](#) [不确定性理论集对分析在预报模型建立中的应用研究](#)[期刊论文]-[气象学报](#) 2003(05)
13. [黄兵, 周献中](#) [不完备信息系统中基于联系度的粗集模型拓展](#)[期刊论文]-[系统工程理论与实践](#) 2004(01)
14. [李志辉, 夏少云, 查建中](#) [基于案例推理的同异反产品设计及其应用](#)[期刊论文]-[计算机辅助设计与图形学学报](#) 2003(11)
15. [蒋云良, 徐从富](#) [集对分析理论及其应用研究进展](#)[期刊论文]-[计算机科学](#) 2006(01)
16. [Ali S, Siegel H J, Maheswaran M, Hensgen D, Ali S](#) [Representing Task and Machine Heterogeneities for Heterogeneous Computing Systems](#) 2000(03)

## 相似文献(10条)

### 1. 学位论文 [黄昌勤](#) [计算网格中任务管理的若干问题研究](#) 2005

计算网格是一种正在兴起的计算基础设施,它能协调地理上分布的各类资源为用户提供有效的一致访问,以满足虚拟组织VOs(Virtual Organizations)的需要.计算网格的巨大潜能正吸引者众多的研究者和工程人员研究其技术以支持跨域的协同工作.然而,网络系统呈现的性能因为资源竞争和任务不确定性等因素而高度动态变化,因此,网络资源调度和任务管理是计算网格中一个关键性的研究课题.在网格环境中,任务从提交给网格系统至任务结果处理过程完成,都一直处于网格任务管理系统的操控之下.由于网格的大规模、异构、动态、分布和自治等特性,使得任务管理变得更加复杂和难以有效实现.本研究以大规模科学和工程计算为背景,以计算网格为基础环境,对网格任务管理中的诸多问题进行了研究.文中所述研究皆得到了国家杰出青年基金项目(60225009)和国家863/CIMS主题资助项目(2002AA414070)的资助.本文所提出的任务管理系统是MASSIVE(Multidisciplinary ApplicationS-oriented SIMulation and Visualization Environment)工程的一部分,后者旨在构建面向多学科应用的模拟与可视化网络环境.本研究主要是从任务调度、网络不确定性处理及QoS操纵、任务授权三方面展开的.在任务调度的研究中,一个双组件双队列分布式调度模型D'3>SM(a Dual-Component and Dual-Queue Distributed Schedule Model)被提出,基于此模型,研究了两种调度策略以用来提高网格计算性能.为了更好的应对网格环境的不确定性,对D'3>SM调度模型进行了扩展以增加专门不确定性处理辅助部件.以扩展模型中各个模块的协作为基础,通过对全局调度器调度策略的修订来解决网络的不确定性;任务授权是网络基础设施中的重要挑战.在考虑社区策略的前提下,提出了并行化子任务级授权服务PSAS(a Parallelized Subtask-level Authorization Service)体系结构以满足权限的最小化原则,同时给出了柔性任务管理和上下文感知的动态授权方法.

### 2. 学位论文 [单世民](#) [基于网格和密度的数据流聚类方法研究](#) 2006

随着硬件技术的不断发展,人们遇到了大量无法利用数据库进行存储的海量数据.这些数据数量非常巨大,并且产生速度很快.为了对这些数据进行有效处理,人们提出了数据流数据模型.作为数据挖掘在新环境的延伸,面向数据流的数据挖掘(简称数据流挖掘)问题已成为当前国内外研究的焦点,而数据流聚类方法则是数据流挖掘的一个重要研究方向.

本文的研究目标是以研究传统聚类方法为基础,通过对传统方法的改进,实现对数据流的聚类处理.通过研究,发现基于网格和密度的聚类方法具有很多适用于处理数据流的特征,有利于实现对数据流的聚类处理.因此,本文在对基于网格和密度的传统聚类方法进行研究改进的基础上,从聚类过程所处理数据集合的动态性角度出发,将基于网格和密度的数据流聚类方法分类为静态方法和动态方法,对其进行了一系列研究.



围绕着数据流聚类问题, 论文主要做了以下四方面的理论研究及应用工作:

1. 对基于网格和密度的数据流静态聚类方法进行了讨论, 通过对传统的基于网格和密度的聚类方法进行分析与改进, 提出了一种新的网格单元密度计算方法。研究发现, 已有方法基本使用数据点计数方式计算网格单元的密度。这种方法会造成数据点对其周围空间影响信息(innuence)的部分丢失, 容易导致同属一类的相邻数据点被分配到不同的数据类中。针对此问题, 提出了“贡献度”概念。“贡献度”即是指在网格化的特征空间中, 数据点对相邻网格单元(即数据点的周围空间)的影响程度。在此基础上, 提出了一种新的网格单元密度计算方法。实验结果证明, 与利用数据对象个数计算网格单元密度的方法相比, 这种方法能够有效减少数据点对周围空间影响信息的丢失。
2. 针对已有的基于网格和密度的聚类方法在稠密单元判定方式以及聚类生成过程方面的不足进行了改进, 最终提出了一种新的基于网格和密度的微粒群混合聚类方法。现有方法基本使用单一阈值来判定稠密网格单元。这种方法限制了对聚类数据与噪音数据进行区分的能力。同时, 已有方法都将首个遇到的稠密单元作为聚类生成过程的起始点。这种操作对生成的数据类缺乏选择性, 使数据类的生成顺序具有不确定性, 进而影响对结果的可控性。针对这些问题, 使用了新的参数—“核心单元密度下限”。此参数限定了聚类中密度极大值的最小允许取值, 对能够生成的聚类进行了限制, 增强了区分聚类数据和噪音数据的能力。同时, 通过将微粒群算法引入聚类过程, 使得数据类能够根据本身的密度极大值有序生成, 解决了聚类生成顺序的不确定性问题。最终, 提出了基于网格和密度的微粒群混合聚类方法(CGDP)。
3. 讨论了动态的网格空间环境下对聚类进行追踪的方法, 提出了动态环境下改进的自适应微粒群算法。通过将网格单元密度转化评估函数取值的方式, 把对聚类密度极值的搜索问题转化为优化问题, 然后利用微粒群算法解决此优化问题。针对数据流的动态性特征, 分析了已有的动态环境下微粒群算法, 发现已有方法的环境变化检测能力仍然存在不足, 具体表现为微粒种群容易停滞在动态环境中静止的局部极点, 进而失去对其他区域中全局极点的追踪能力。针对此问题, 使用“活性因数”概念及分布式处理模式, 解决了微粒种群在动态环境中的停滞问题。进一步的, 提出了动态环境下改进的自适应微粒群算法(IAPSO)。实验证明, 该方法能够适应更多类型的复杂动态环境, 具有更为普遍的实际意义, 为论文对数据流聚类方法的研究提供了技术支持。
4. 以入侵检测为代表的具数据流环境为研究对象, 对基于网格和密度的数据流动态聚类方法进行了研究, 提出了数据流中孤立点识别方法。在此类数据流环境中, 可以认为只存在在一个主要聚类, 并且与其他数据相比, 属于主要聚类的数据具有相对更高的密集程度。这种环境的基本需求就是判断新出现的数据对象是否属于主要聚类。针对这种情况, 首先扩展了“孤立点”的含义, 将不属于主要聚类的数据对象定义为相对于主要聚类而言的“孤立点”。然后, 将本文的前三项工作成果进行综合, 提出了一种面向数据流的动态聚类方法—数据流中孤立点识别方法(ODDS)。设计并实现了数据流分析原型系统, 并在此基础上, 以入侵检测的标准测试数据为基准进行实验, 以满足具数据流环境的应用需求为目标, 通过与以往基于聚类的入侵检测方法进行结果对比, 说明了ODDS方法的有效性。

### 3. 期刊论文 [黄德才, 龚卫华, 张丽君, 赵克勤, HUANG De-cai, GONG Wei-hua, ZHANG Li-jun, ZHAO Ke-qin 基于联系数的网格任务动态调度算法 -计算机工程2009, 35\(8\)](#)

针对传统网格任务调度模型将每个任务的预期执行时间设定为一个确定常量, 无法表达其不确定性的问题, 利用一种新的软计算方法—集对分析联系数来表示并处理网格任务预期执行时间的综合不确定性。提出基于联系数的不确定性网格动态调度模型, 在线动态调度算法OUD\_MCT和批模式动态调度算法BUD\_Surferage。数值仿真结果表明, 该算法能较好地描述网格任务预期执行时间的动态性和不确定性, 使传统网格调度方法成为其特例, 在动态和不确定网格环境中具有良好的理论意义和实用价值。

### 4. 学位论文 [任姣娜 基于Gridsim的网格资源调度算法及其应用的研究 2004](#)

网格计算是近年来得到快速发展的广域网网络计算技术。研究人员试图将很大范围上地理分布的异构计算机系统集合在一起形成一个大规模的计算平台, 我们称之为网格(Grid)。通过这种平台, 用户能够方便、快捷地使用自己需要的资源而不必了解其中的细节。网格系统提供给用户的除了廉价、高效的高性能计算以外, 还包括合作存取各种数据信息, 广域多媒体应用等等。

然而, 由于网格是个新的研究领域, 网格环境相对于一般网络环境有着更为复杂的特征, 如存在多管理域和站点自治, 系统的动态性、异构性和通信延迟的不确定性更高, 硬件和软件两个层次上都存在异构性等等。因此, 实现有效的网格计算还有很多需要解决的问题, 具体包括资源调度和管理、系统安全、编程模式、性能评测和数据存取等等。其中, 在网格环境里如何有效地管理资源和调度计算是影响网格计算是否成功的最重要因素之一。由于资源在广域网上的分布性、异构性以及存在着不同的存取和花费模式, 使得网格环境下的资源管理变得十分复杂和具有挑战性。

本文试图从网格资源的特点出发, 寻求一种更加合理的网格资源管理模式和更加有效的调度策略。作者参阅了国内外的大量文献, 在研究目前的网格资源管理模型的基础上, 提出了基于经济的网格市场模型并且实现了受任务完成时间和费用约束的贪婪算法, 取得了一定的研究成果。

本文中作者的贡献是: (1) 针对目前国际上对网格资源管理的研究所采取的普遍的方法, 深入分析现有网格资源管理模型及计算网格系统的特点, 提出基于经济的网格市场模型的资源调度管理方法。

- (2) 借用市场、拍卖等经济学模型描述了基于经济的网格市场模型的资源调度管理的内部机制和行为。
- (3) 研究网格计算环境下一组相互独立的计算任务(Meta-Task)的资源映射策略, 分析了常用的资源调度算法, 并且比较了它们的优缺点和适用情况。
- (4) 在基于经济模型的网格模拟器Gridsim环境下, 实现了受任务完成时间与费用约束的贪婪算法, 在追求较小的任务完成时间的同时兼顾任务的服务质量(QoS)需求, 使之能够更好地支持大规模用户使用网格资源。

### 5. 会议论文 [黄德才, 刘端阳, 张丽君, 赵克勤 不确定网格调度问题的联系数学建模方法初探法 2007](#)

任务调度算法是计算网格任务管理系统中的核心技术问题。由于网格环境中存在大量的不确定因素, 导致传统网格调度算法和调度系统不能在开放、异构和动态的真实网格环境中有效运行。利用一种新的软计算方法—集对分析联系数研究和处理网格调度中的综合不确定性问题。在简单介绍集对分析概念基础上, 引入联系数全序关系, 提出了一个基于联系数的不确定网格静态调度算法CBU\_Min-min, 并进行了数值实例计算。理论和实例计算表明, 这种模型能较好地描述网格任务预期执行时间的动态性和不确定性, 并使传统网格调度方法成为其特例, 在动态和不确定网格环境中有良好的理论和实际应用价值。

### 6. 学位论文 [吴娟 基于策略域的分布式访问控制模型 2009](#)

随着越来越多的分布式系统的出现, 如P2P系统、跨多个域的计算机网络、Web服务组合, 软件系统的形态发生了根本性地变化。作为软件系统安全保障的访问控制系统也需要重新的设计和实现。本文针对现有访问控制模型在分布式环境中表现出的不足, 以对各种分布式访问控制系统的定性研究为基础, 设计并实现了一个分布式访问控制模型, 取得了以下重要成果: <br>

1. 对典型的分布式访问控制系统做了定性研究。首先从权限委派方式、身份鉴别机制、状态信息的维护、决策模型、信任度评估、实现与方案的一致性这6个方面进行了性质分析。然后以这几个指标为依据, 对具体的访问控制系统做了量化。基于这些研究对这一方向的研究现状做出了总结, 并定下了本文研究工作的目标和基调。<br>
  2. 对分布式系统做了抽象, 提出了策略域的概念, 并且基于策略域, 提出了一个分布式访问控制模型PDAC。对于访问控制策略, 文章首次将能力和契约结合在一起使用, 能力从客观上反映了节点提供的操作, 而对这些操作的约束则由策略域实时地给出。这样, PDAC模型就具备了动态授权的特征。对于决策模型, 策略域既可以根据本地信息做出独立决策, 也可以联合其他策略域根据全局信息做出合意决策。因此, PDAC模型具备分布式决策的特征。另外, PDAC模型支持权限的委派, 并且当委派权限的策略域对权限进行修改后, 策略域能够记录这种改变, 使得我们能够以更加细粒度的方式来管理授权。<br>
  3. 以联想广播为基础, 实现了策略域的协同工作模型。使用该模型, 策略域可以通过指定状态来选择希望与之协作的其他策略域, 也可以通过设置自己的本地状态来获得与其他策略域协作的机会。基于联想广播的协同工作模型能够在—组策略域中动态地定义多个协同子集, 满足分布式环境的动态协作要求。以这个协同模型为基础, 为PDAC模型制定了服务发现机制和邻域查询机制。<br>
  4. 基于主观逻辑方法, 在信任评估中引入不确定性因素, 确立了动态信任评估机制。实验结果表明, 基于主观逻辑的信任评估能够更好地反映分布式环境的不确定性, 对于促进节点间的协作有着积极的作用。考虑到现实世界的信任具有随时间衰减的特征, 在信任评估模型中引入了老化机制。信任评估的老化能够客观地反映节点间的信任关系, 对节点的行为起到了很好地约束作用。<br>
  5. 结合BAN逻辑和时序逻辑, 提出了策略描述语言PDPL, 给出了形式化的语法规则和语义解释。进而, 以PDAC模型作为形式系统的一个语义模型, 定义了形式系统的公理和推理规则。以这些工作为基础, 对PDAC模型的握手、委派、协作、决策等性质进行了可靠性论证。<br>
- 总的来说, 本文提出的PDAC模型是对访问控制模型的一种新的解决思路, 本文的研究成果推动了分布式访问控制模型研究的发展, 对构建更加安全可靠的分布式系统具有重要的实践指导意义。<br>

### 7. 期刊论文 [殷锋, 何先波, 刘韬, YIN Feng, HE Xian-bo, LIU Tao 基于自适应模糊-PID反馈模型的网格调度技术 -计算机科学2009, 36\(7\)](#)

服务调度技术是计算网格任务管理系统中的核心问题.为适应网格服务部署的需要,提出了一种基于自适应“模糊-PID”反馈控制模型的Agent技术,该技术融合了模糊理论与PID(Proportionment-Integral-Differential coefficient)技术的优点.实验证明本技术能解决网格服务部署中的动态性和不确定性,并充分发挥网格服务的虚拟执行功能,可在网络带宽效率、延时和可靠性等方面做出更好的权衡.

## 8. 学位论文 [王铁强](#) [实际地形下风场CFD模拟与参数分析](#) 2009

利用计算流体力学(Computational Fluid Dynamics, CFD)模型模拟近地面风场是风能利用的技术发展趋势.由于风场CFD模拟的复杂性,现阶段在面向风能利用的风场模拟中应用还比较少.本文的主要目的是通过风场模拟实验,研究风场CFD模拟的可行性和模拟精度.

本文选择了具有较完整现场测风数据的苏格兰Askervein山地作为模拟计算场址,解决了真实地形的建模问题,并对各种近地面风场计算网格的生成方法进行了研究.在此基础上,本文通过实验从首层网格高度和网格高度增长率方面考察了网格精度对模拟结果的影响;比较了多种湍流模型在实际地形下的近地面风场模拟中的表现,并总结出具有较高模拟精度、适用于近地面风场模拟计算的湍流模型;研究了不同离散格式对模拟结果的影响,并总结了各种离散格式的适用范围;分析了不同的边界条件如地表粗糙度条件对模拟结果的影响,并提出了适用于近地面风场模拟计算的边界条件设置方案.

本文建立了利用CFD模型进行实际地形下近地面风场模拟计算的总体方案及流程,提出了模拟计算的自动化方案,并得到了较好的应用效果,一方面证明了利用CFD模型进行风场模拟的可行性,另一方面也表明利用CFD模型进行风场模拟具有较高的准确性,能降低风场模拟的不确定性.

## 9. 期刊论文 [张辉亚](#).[张煜盛](#).[肖合林](#).[徐波](#) [提高喷雾模拟精度的算法](#) -[科学通报](#)2007, 52(5)

燃油喷雾是直喷式发动机燃烧的关键过程.喷雾模拟的精度决定了燃烧计算的可靠性.然而,在KIVA和商业CFD代码中,传统喷雾模拟技术对网格精度非常敏感.因而,预测的发动机性能和排放依赖于计算网格.导致这个问题的两个主要原因是液滴碰撞算法和气流相间耦合.为了提高喷雾模拟精度,采用交错网络液滴碰撞(CMC)算法和气相速度插值算法对原始KIVA代码进行了修正.在定容燃烧室和直喷式柴油机条件下,分别从喷雾结构、预测平均液滴尺寸和喷雾贯穿距三个方面检验了改进KIVA代码对喷雾模拟精度的改善.结果表明,网格依赖性显著降低.通过这些改进,喷雾结构的失真现象消失.在定容燃烧室计算中,预测的平均液滴尺寸的不确定性从30  $\mu\text{m}$ 减小到5  $\mu\text{m}$ ;在发动机模拟中,这种不确定性进一步减小到2  $\mu\text{m}$ .在中等和精细网格条件下,发动机模拟预测的喷雾贯穿距也获得了较好的一致性.

## 10. 学位论文 [马晨明](#) [基于多QoS的网格资源管理与任务调度算法研究](#) 2008

网格是高性能计算和信息服务的战略性基础设施,而网络技术已成为下一代互联网应用的关键技术.高速网络的发展将分散的、异构的计算资源有机地整合到一起形成计算网格,为解决科学和工程领域一些大规模计算问题提供了理想的平台.由于网格所具有的广域性、动态性、异构性等特点,如何对资源进行管理及任务调度以满足用户的需求是一个极具挑战性的问题.

随着网格的飞速发展,服务质量(QoS)成为网格资源管理与任务调度过程中必须要考虑的一个重要因素.在这种情况下,本文通过对网格QoS框架的研究,将多QoS的模糊属性融合到资源的管理与任务调度算法中,使网格任务在不确定环境下按时执行,本文的主要工作包括以下几部分:

1. 首先介绍了网格计算的概念和目的、网格QoS的体系结构并同时指出了基于QoS任务调度的重要性;然后在此基础上对网格技术与调度算法的研究现状进行了详细的阐述和总结.

2. 针对用户QoS需求的模糊性和不确定性,提出基于D-S理论的区间值模糊多QoS测量方法,将用户提交的区间QoS需求有效地进行融合,形成优先队列,仿真结果表明,该方法可以以区间数的形式满足用户的多QoS的任务调度需求.

3. 为了能在任务调度中更快地发现资源,对网络资源的合理管理显得尤为重要.通过对资源的QoS属性进行划分,利用数据挖掘中改进的DBSCAN聚类算法将网络环境中的各异构资源进行资源绑定是一种有效的资源管理方法.

4. 基于网络环境中的多QoS因素,改进了现有的网络调度算法,提出了以makespan值作为优先目标的网络调度新算法,仿真实验证实算法在不同异构环境下都有很好的效果.

5. 不确定网络环境中通常将服务质量的高低作为任务的优先调度目标,借鉴人类社会学中的人与人之间的信任模型,考虑网络节点间的服务质量信任需求,提出了基于QoS的网格关联任务调度算法.算法考虑任务之间的通信与执行的先后性,更符合实际的网络环境,该算法以较小的时间跨度为代价,取得了较好的服务质量性能.

最后,对本文的研究工作进行了总结,对目前研究内容存在的问题进行了分析,并提出了进一步的展望.

## 引证文献(1条)

### 1. [黄德才](#).[龚卫华](#).[张丽君](#).[赵克勤](#) [基于联系数的网格任务动态调度算法](#)[期刊论文]-[计算机工程](#) 2009(8)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_jsjcx200708034.aspx](http://d.g.wanfangdata.com.cn/Periodical_jsjcx200708034.aspx)

授权使用: 浙江工业大学图书馆(wfzjgydx), 授权号: df75fd71-de79-405b-8bbe-9e0000ab353e

下载时间: 2010年9月29日