

QIACO: 一种多 QoS 约束网格任务调度算法

孙伟峰, 覃振权, 李明楚, 胡 晶, 陈媛芳

(大连理工大学软件学院, 辽宁大连 116620)

摘 要: 网格环境下的任务调度问题属于 NP 难解, 难以得到精确的最优解, 适合使用蚁群算法等智能优化算法对最优解进行逼近; 同时, 服务质量 (QoS) 也是衡量网格性能的一个重要指标, 网格任务调度应该满足用户的 QoS 需求. 为解决具有 QoS 保证的网格任务调度问题, 本文以带有 QoS 约束的任务为研究对象, 结合改进的蚁群算法, 提出了一种基于蚁群算法的多 QoS 约束网格任务调度算法 (QIACO). QIACO 将蚁群算法用到网格任务调度问题中, 具体考虑了 5 种 QoS 约束, 并将 QoS 约束转换成效用, 提出了多约束 QoS 任务调度模型. 同时, 本文改进了蚁群算法的搜索策略、决策规则和信息素更新策略, 使总效用值即用户满意度达到最大. 理论分析和仿真实验表明 QIACO 无论是在 Makespan 方面, 还是在总效用方面都相比同类算法有较大的优势.

关键词: 多 QoS 约束; 网格任务调度; 蚁群算法; 伪随机比例

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2011) 05-1115-06

QIACO: An Algorithm for Grid Task Scheduling of Multiple QoS Dimensions

SUN Wei-feng, QIN Zhen-quan, LI Ming-chu, HU Jing, CHEN Yuan-fang

(School of Software of Dalian University of Technology, Dalian, Liaoning 116620, China)

Abstract: Task scheduling problem in grid is NP-hard, and it is difficult to attain an optimal solution, so we can use intelligent optimization algorithms to approximate the optimal solution (for example ant colony optimization). Moreover, Quality of Service (QoS) is also an important factor in determining the performance of grid. Task scheduling needs to satisfy user's QoS requirements. In this paper, we propose a novel Multiple QoS Dimensions (QIACO) algorithm for Grid Task Scheduling which is based on modified ant colony optimization algorithm and focuses on the task with QoS dimensions. The QIACO strategy use the ant colony algorithm to solve the task scheduling problem of grid, specifying 5 kinds of QoS dimensions and transforming the QoS to utility, and ultimately, representing the model of task scheduling with multiple QoS dimensions. At the same time, we improve many aspects of the ant colony algorithm, such as the search strategy, decision rule, pheromone update strategy of ant colony optimization, in order to maximize the user's satisfaction. QIACO shows a significant improvement in both makespan and total utility according to the theoretical analysis and simulation.

Key words: multiple QoS dimension; tasks scheduling; ant colony; pseudorandom proportional

1 引言

网格计算(云计算)是网格应用的一种, 将分布的计算机组织起来协同解决复杂的科学与工程计算问题; 而网格资源管理系统是网格计算的核心中间件与用户级中间件的重要组成部分, 是连接各类远程资源和进行任务协同调度的核心基础设施. 在网格资源管理系统的设计与实现中, 对任务调度策略的研究是核心内容. 高效的调度策略可以充分利用网格系统的处理能力, 从而提高应用程序的性能, 因此, 对于网格任务调度的研究显

得尤为重要. “提供非凡的服务质量(QoS)”是判断网格性能的准则之一, 是网格系统的一个重要的性能指标^[1], 因此网格的任务调度策略应该考虑用户的 QoS 需求. 如何提高网格调度算法性能, 并保证网格任务调度过程中的服务质量是网格需要解决的一个关键问题.

为解决网格任务调度问题, 本文在将蚁群算法用到网络调度的基础上, 提出了多 QoS 约束网格任务调度模型. 在该模型上用改进的蚁群算法 QIACO 来解决多 QoS 约束的任务调度问题, 利用随机交换局部搜索结合蚁群系统来改进蚁群算法, 避免算法进入早熟阶段, 增强算

法的全局搜索能力,提高算法的运行速度。

2 相关工作

网格下的任务调度已有一些研究,Min-Min^[2],Sufferage^[3]等是比较简单有效的启发式算法^[4],但是它们都没有考虑用户的多 QoS 需求.在以 QoS 为指导的任务调度算法研究中,He^[5]等人以网络带宽作为 QoS 约束来指导任务调度,提出了 QoS-Min-Min 算法.QoS-Min-Min 只考虑了网络带宽约束,不适合多维 QoS 约束的情况.Weng^[6]等人以平均响应时间作为 QoS 约束提出了 QoS-Sufferage 算法,在计算 sufferage 值时将响应时间也作为影响任务调度的因素.该算法也只是考虑了平均响应时间情况,同样不适用多 QoS 约束.陈晶^[7]等提出了网格资源调度算法 Senior:根据网格任务的预测执行时间,调整任务的发送顺序和改变分配的目标资源.但该算法也只考虑了两个简单的 QoS 约束参数.蚁群算法^[8]具有很好的扩展性,即在一个规模 n 的问题上求出最优解后,再增加 m 个节点,可在原有基础上快速找到该问题的最优解,因此蚁群算法很适合解决网格计算中的调度问题.一些基于蚁群算法的网格调度算法已被提出,但是它们没有考虑 QoS 因素.

本文在提出具有多约束 QoS 的网格任务调度模型上,结合局部搜索改进原始蚁群算法 ACO,提出了 QIACO 算法.与改进的 Min-Min、QoS-Min-Min 和不包括局部搜索策略的 QACO 算法进行了仿真实验.仿真结果也验证了 QIACO 的高性能和高计算效率.

3 多 QoS 约束网格任务调度模型

网格环境下的批模式任务调度采用了周期性和任务数量限制相结合的方式,在调度周期到达或者本次调度任务量达到规定数量时进行调度算法的执行.批模式调度中的任务为元任务,元任务之间相互独立,且带有多个 QoS 需求.为衡量用户的 QoS 需求,在多 QoS 约束模型中使用效用函数将 QoS 约束转换成效用,用来表示用户的满意度.网格系统中的调度者应该最大化网格系统中所有用户的效用而不仅仅是满足某个用户的最大效应,因此要从整个系统角度考虑,设计算法最大化所有用户的总效用.

3.1 相关概念和定义

本文定义了网格下可度量的 QoS 模型,重点考虑以下 5 种 QoS 约束:

时间(Timeliness):每个任务可以接受的执行时间期限(deadline).

可靠性(Reliability):任务可以被成功执行的概率.

版本(Versions):考虑版本对任务执行时间的影响和用户偏好.

安全性(Security):在网格节点上可以获得的的安全等级.

优先级(Priority):任务的优先级,由用户和调度者协商设定.

为保证模型的可扩展性,本文的参数定义和解决方案可应用于更多 QoS 类型的模型.

在保证 QoS 需求的基础上,采用 Shoukat Ali 等人提出的预测方法^[9],可以得到任务在每个计算节点上的运行时间,并给出了如下的参数定义:

定义 1 计算资源集合 R . $R = \{r_1, r_2, \dots, r_m\}$ 表示网格环境下的 m 个异构的计算资源.

定义 2 任务集合 T . $T = \{t_1, t_2, \dots, t_n\}$ 表示一个任务集中的 n 个任务集合.

定义 3 预测执行时间矩阵 ET .

$ET = \begin{Bmatrix} et_{11} & \dots & et_{1n} \\ M & & M \\ et_{m1} & \dots & et_{mn} \end{Bmatrix}$ 表示任务预测执行时间.其中, et_{ij} 表示任务 t_i 在资源 r_j 上的预测执行时间.

定义 4 调度结果 $M(i)$, $1 \leq i \leq n$. 表示任务 t_i 被分配的资源.

定义 5 调度函数 S_j . $S_j = \{S_j(i) | 1 \leq i \leq n, 1 \leq j \leq m\}$, 表示 t_i 在 r_j 上的执行顺序.

定义 6 版本函数 $V(i)$, $1 \leq i \leq n$. 表示任务 t_i 执行时的版本.

定义 7 QoS 约束空间 Q_i . 每个任务 t_i 有 d_i 个 QoS 约束, Q_i^j 代表 t_i 的第 j 个 QoS 约束的有限集 ($1 \leq j \leq d_i$), q_i^j 为 Q_i^j 中 t_i 的 j 维 QoS 约束的具体值. $Q_i = \{Q_i^1, Q_i^2, \dots, Q_i^{d_i}\}$ 为任务 t_i 的 QoS 的 d_i 维空间, $q_i = \{q_i^1, q_i^2, \dots, q_i^{d_i}\}$ 为空间上的一点.

3.2 多 QoS 约束评价模型

在网格调度中网格节点性能不同,网络延迟也无法计算,因此网格中的调度算法将 Makespan 作为评价标准,即所有任务完成时间减去任务开始执行的时间. Makespan 越小,调度方法越好.由于本文的调度算法考虑了 QoS 约束,除 Makespan 外,还要考虑评价用户的 QoS 需求.定义效用函数 $U_i(q_i)$ 将 QoS 都转换成效用.任务 t_i 的效用函数定义如公式(1).

$$U_i(q_i) = \sum_{j=1}^{d_i} w_j^i U_i^j(q_i^j) \times p_i \quad (1)$$

其中, $U_i^j(q_i^j)$ 为任务 t_i 对应第 j 维 QoS 的 q_i^j 的效用值, w_j^i 为第 j 维 QoS 对应的权重, p_i 为 t_i 的优先级.网格任务调度的目标就是最大化所有用户的总效用 $\sum_{i=1}^n U_i(q_i)$.

4 QIACO 算法

论文前期工作将蚁群系统应用到网络任务调度中提出了 QACO 算法^[10]. QACO 使用蚁群来寻求任务与资源的最佳映射: 将调度者看成蚂蚁, 将调度的过程看成蚂蚁觅食的过程, 使用总体效用来更新信息素. 对于大规模的计算, 蚂蚁系统的性能会下降, 寻路时间长, 收敛慢, 时间复杂度高. 如果减少蚂蚁的数量又可能会陷入局部解, 所有个体发现的解几乎一致, 可能导致无法找到全局最优解. 为改进蚁群算法中的搜索问题, 设计了改进的算法 QIACO. QIACO 用伪随机比例规则改进了蚂蚁选路方向的控制方法; 把局部搜索算法和生成初始解的机制相结合, 设计了智能的局部搜索算法 (local search); 改进了信息素更新, 采用了局部和全局信息素结合的策略.

4.1 局部搜索算法和伪随机比例规则

局部搜索算法采用随机交换局部搜索策略, 每当生成可行解后, 对可行解进行扩大的局部搜索, 防止陷入局部最优解. QIACO 从可行解中随机选取 2 个任务, 将它们的顺序进行交换形成一个新的解, 重复多次将效用最好的解替换初始解. 经过实验证明局部搜索算法进行 $t/2$ 次交换可以得到新的 $t/2$ 个解 (t 为总任务数), 并能取得较好的结果.

QIACO 算法过程如图 1 所示.

参数定义: Tabu: 搜索表, U_i : 任务 t_i 的效用, P_{ij}^k : 状态转移概率, ANT: 蚂蚁集合, T : 任务集合, NC-max: 最大循环数, τ_{ij} : 任务 i 和任务 j 之间的信息素.

算法第 2-16 步是主体部分, 循环遍历 NC-max 遍. 首先初始化搜索表 (步骤 3), 每只蚂蚁从初始位置开始搜索直到所有任务都加入搜索表中形成完整路径. 在蚂蚁选择下一个任务时如步骤 7 所示, 蚂蚁 k 根据伪随机比例规则选择下一个任务, 即从任务 t_i 到任务 t_j , 选择方法如公式 (2).

$$t_j = \begin{cases} \arg \max_{l \in A} \{ \tau_{il} [\eta_l]^\beta \}, & \text{如果 } q \leq q_0 \\ P_{ij}^k(t), & \text{否则} \end{cases} \quad (2)$$

其中 q 是均匀分布在区间 $[0, 1]$ 的一个随机变量, q_0 ($0 \leq q_0 \leq 1$) 是一个参数, 随机变量 $P_{ij}^k(t)$ 根据公式 (3) 由概率分布产生. 使用效用 $\sum_{i=1}^n U_i(q_i)$ 来更新信息素 τ_{ij} .

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_j(t)]^\beta}{\sum_{l \in A} [\tau_{il}(t)]^\alpha [\eta_l(t)]^\beta}, & \text{如果 } j \in A \\ 0, & \text{否则} \end{cases} \quad (3)$$

公式 (3) 中, 启发函数 $\eta_j = \frac{1}{et_j}$. 其中, et_j 为任务 t_j 的

```

1: Initialize  $\alpha, \beta, et, \rho, Q, q_0, NC\text{-max}$ 
2: while  $NC < NC\text{-max}$  do
3:   initialize  $Tabu$ 
4:   for all tasks in meta-task  $T, t_i \in T$ 
5:     for all ants in  $ANT, ant_k \in ANT$ 
6:       Compute  $U_i$ 
7:       select the next task  $t_j$  through pseudorandom proportional
8:       assign the machine  $M(j)$  of the task maximum utility to the task  $t_j$ 
9:       update  $Tabu$ , add the task  $t_j$  into  $Tabu$ 
10:      update the local pheromone  $\tau_{ij}$ 
11:    end for
12:  end for
13:  local search
14:  set the best utility  $U\_best$  of the cycle as the jumping-off of the next cycle
15:  update the global pheromone only for the best route
16: end while
17: compute  $U, T$ 
18: end

```

图 1 QIACO 算法

执行时间. α 和 β 分别用来表示信息素和启发函数的重要性. A 代表还没被调度过的任务集合.

在伪随机比例规则中, 蚂蚁选择当前可能的最优移动方式的概率是 q_0 , 这种最优的移动方式是根据信息素的积累量和启发式信息求出的 (在这种情况下, 蚂蚁继续开发已知的知识). 同时蚂蚁以 $1 - q_0$ 的概率有偏向性地探索各条边. 通过调整参数 q_0 , 我们可以调节算法对新路径的探索度, 从而决定算法是应该集中搜索至今最优路径附近的区域还是应该探索其他区域.

4.2 信息素更新

在信息素更新算法上采用了局部信息素和全局信息素更新结合的策略. 局部信息素更新指在路径构造过程中, 蚂蚁每经过一条路径 (i, j) , 都将使用公式 (4) 来更新该路径 (i, j) 的信息素.

$$\tau_{ij} = (1 - \xi) \tau_{ij} + \xi \tau_0 \quad (4)$$

公式 (4) 中 τ_0 是信息素的初始值, τ_{ij} 是该路径之前信息素的值, ξ 为调节因子, 满足 $0 < \xi < 1$. ξ 的作用是降低初始信息素对其他蚂蚁寻路的影响, 防止其他蚂蚁过早停滞, 扩大搜索范围, 不至于找不到解或陷于局部解中.

全局信息素则只更新迄今为止最好路径上的信息素, 更新公式如式 (5). 其中, ρ 是信息素挥发系数, $(1 - \rho) \tau_{ij}(t)$ 是原来路径上的遗留信息素. $\Delta \tau_{ij}(t)$ 是对于前 m 次累计的总效用 U_k 的变化, Q 是调整蚂蚁信息素与总效用关系的调节因子 (在本文中取一简单的常数), 再考虑到挥发系数, $\rho \Delta \tau_{ij}(t)$ 为蚂蚁在本次循环中留在路径上的信息素.

$$\begin{cases} \tau_{ij}(t + n) = (1 - \rho) \tau_{ij}(t) + \rho \Delta \tau_{ij}(t) \\ \Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k, \Delta \tau_{ij}^k = QU_k \end{cases} \quad (5)$$

5 实验仿真及结果分析

仿真模拟实验在 Windows XP 下实现,分别考察网格节点数目 m 、任务数 t 的变化和固定的两种情况,具体为:

(1) $m = 10$ 个网格节点的网格系统对 t 个(20 ~ 100)独立任务组成的任务组的调度情况;

(2) $t = 100$ 个独立任务在 m 个(5 ~ 30)网格节点组成的网格系统的任务调度情况.

Min-Min 与其他同类网格调度算法相比可以获得整体较好的效果^[11],本文的仿真试验将与具有 QoS 保证的 Min-Min 算法比较,来说明 QIACO 的调度性能.

改进的 Min-Min 算法是将 Min-Min 调度算法应用到多 QoS 约束网格任务调度模型中,如图 2 所示.改进的 Min-Min 算法针对前面提出的 QoS 模型稍加改进,依据从最小完成时间换成效用 $U_i^k(q_i^k)$ 来计算.算法中的 U_{ij} 即为总效用.本文同时与 He^[5]等人提出的 QoS-Min-Min 以及不带局部搜索的 QACO 进行比较.除了比较总效用外,也用 Makespan 作为衡量标准进行了仿真.

```

1: for each task  $t_i \in T$  (in an arbitrary order)
2:   for each machines  $m_j \in M$  (in a fixed arbitrary order)
3:     compute each utility
           
$$U_{ij} = \sum_{k=1}^{d_i} w_i^k U_{ij}^k(q_i^k) \times p_i$$

4:   endfor
5: endfor
6: repeat
7:   for each task in  $T$  find the maximum utility and the machine that obtains it
8:   find the task  $t_k$  with the maximum utility
9:   assign task  $t_k$  to the machine  $m_l$  that gives the maximum utility
10:  remove task  $t_k$  from  $T$ 
11:  update  $U_{ij}$  for all tasks
12: until all tasks in  $T$  are mapped
    
```

图2 改进的 Min-Min 算法

5.1 算法实现及参数设置

同第 3 部分多 QoS 约束网格任务调度模型对应,本实验选取了 5 维 QoS 约束.

时间:通过 $d_i = (\delta_i \times etc_i) + a_i$ 计算, δ_i 为 $\{1, 2, 3, 4\}$ 中的随机数, a_i 为任务到达时间,效用函数为 $d_i = \begin{cases} 1 \\ 0 \end{cases}$,如果在执行期限之前完成, $d_i = 1$,否则 $d_i = 0$.

可靠性:每台网格节点失效的概率为每单位时间 0.0005 ~ 0.0015,随机选取.

安全:分为 4 个等级, poor(1), low(2), medium(3), high(4).

版本:仿真环境中 $M = 10$ 台机器以及 $V = 3$ 个版本,不同的版本机器的性能不相同,比如执行时间和用户对该版本的偏好 (preference) 不同, $ET(V_{k+1}) = random$

$\ast ET(V_k)$, $random$ 为 50% ~ 90% 之间一个随机数.用 r_{ik} 来表示用户对任务 t_i 对版本 v_k 的偏好, $r_{i0} = 0$ (用户最偏好), $r_{i1} = r_{i0} \times UR(0, 1)$, $UR(0, 1)$ 是一个在区间 $[0, 1]$ 内的统一分布随机函数.

优先级:将优先级简单的划分为三个等级(1, 2, 3), 值越大优先级越高.

执行时间 ETC 预测,使用 Shoukat Ali 等人提出的预测模型^[11],设定 $\mu_{task} = \mu_{mach} = 100$, $V_{task} = V_{mach} = 0.5$.

经过参考、对比相关研究^[12],及反复多次实验得出 QACO 和 QIACO 的参数组合如表 1.此参数表是针对上一小节的两种参数设置得来的,不同的网格环境应该有较优的参数设定,其他的网络环境也可用相同的方法进行训练,设置合适的参数.

表 1 QACO 和 QIACO 参数设置

参数名称	参数值	参数名称	参数值
Ant(QIACO)	2	ρ	0.1
Ant(QACO)	t (任务数)	Q	10
NC-max	100	q_0	0.95
α	1	ξ	0.1
β	2		

5.2 仿真结果及分析

任务的总效用公式(1)是调度算法性能的系统级评价标准,获得总效用最大的调度算法能够为系统提供较好的服务;本文同时也以 Makespan 作为衡量标准来评估几种算法的性能.实验结果对比如图 3 ~ 图 6 所示.

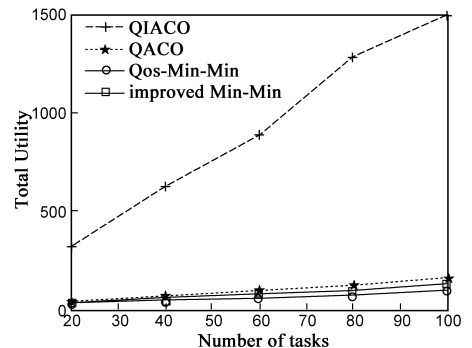


图3 不同任务数下总效用比较

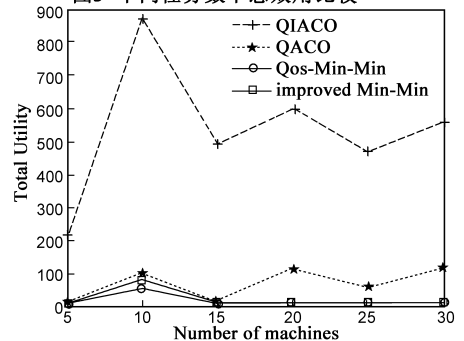


图4 不同节点数下的总效用比较

图 3、图 4 分别是四个算法在不同任务数和不同节点数情况下总效用的对比图, QIACO 所获得的总效用要明显占优. 其中 QoS-Min-Min 的效用最差, 因为该算法只考虑了与时间相关的 QoS 因素作为调度的启发信息而没有考虑到综合效用. 改进后的 QIACO 相比 QACO 增大了搜索范围, 并且降低了前期结果对后期的影响, 性能得到明显的提高.

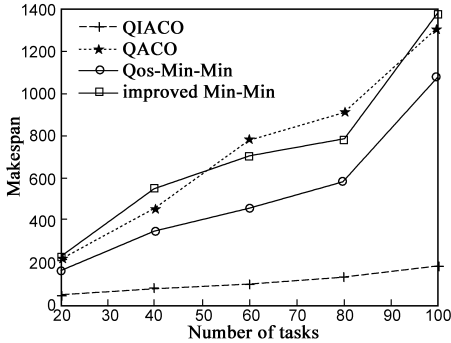


图5 不同任务数下的Makespan比较

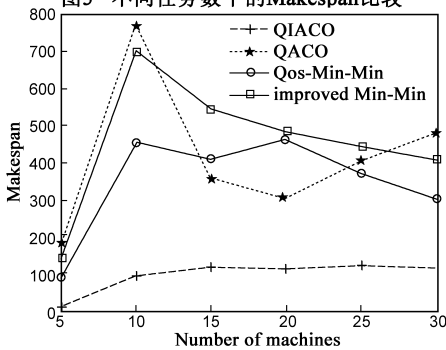


图6 不同节点数下的Makespan比较

图 5 和图 6 是 Makespan 的对比图. QIACO 的 Makespan 最小, QoS-Min-Min 的 Makespan 也较小, 因为它专门针对完成时间进行了优化. QACO 在计算转移概率的时候也考虑了时间因素, 在保证效用最大的前提下寻找比较小的完成时间, 有时候比 QoS-Min-Min 的 Makespan 更好. 而经过改进的 QIACO 在时间方面考虑得更周全, 蚂蚁选择和信息素更新策略更优, 能够获得最好的 Makespan.

综上所述, 基于蚁群算法的任务调度算法 QACO 和 QIACO 在多 QoS 约束模型上比基于 Min-Min 的任务调度算法性能高. 无论是 Makespan 还是总效用, QIACO 都表现了最好的性能. 并且 QIACO 的执行时间相比 QACO 少很多, 执行的更快.

6 总结和展望

本文提出了基于蚁群算法的多 QoS 约束网络任务调度算法 QIACO. 通过改进局部搜索来改进蚁群系统, 使用 QoS 效用作为算法的启发信息, 最大化任务的总效用, 能在有效的完成任务调度的同时最大化用户的

QoS 效用. 分析和仿真实验表明, 在多约束 QoS 任务调度模型中, QIACO 在 Makespan 和总效用方面都比 ACO、改进的 Min-Min、QoS-Min-Min 等算法有很大的优势, 可以获得较好的调度结果.

下一步的工作将考虑更全面的 QoS 参数, 并动态调整算法中的相应参数; 验证蚁群算法在一般非 QoS 模型下的通用性; 研究蚁群的参数和局部搜索方法的对整体性能影响的理论模型.

参考文献

- [1] Ian Foster. The grid: A new infrastructure for 21st century science [J]. Physics Today, 2002, 55 (2): 42 - 47.
- [2] Muthucumar M, Shoukat A, Howard JS, et al. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems [A]. Proc of the 8th Heterogeneous Computing Workshop [C]. Puerto Rico: IEEE Computer Society Press, 1999.
- [3] Henri C, Arnaud L, Dmitrii Z, et al. Heuristics for scheduling parameter sweep applications in grid environments [A]. Proc of the 9th Heterogeneous Computing Workshop [C]. Mexico: IEEE Computer Society Press, 2000.
- [4] 王新军, 洪晓光, 王海洋, 孟祥旭. 网格计算中一种启发式数据分配算法的讨论 [J]. 电子学报, 2004, 32 (4): 648 - 650.
Wang Xinjun, Hong Xiaoguang, Wang Haiyang, Meng Xiangxu. Discussion on a heuristic algorithm of data distribution in grid computing [J]. Acta Electronica Sinica, 2004, 32 (4): 648 - 650. (in Chinese)
- [5] He XS, Sun XH, von Laszewski G. QoS guided min-min heuristic for grid task scheduling [J]. Journal of Computer Science and Technology, 2003, 18 (4): 442 - 451.
- [6] C Weng, X Lu. Heuristic scheduling for bag-of-tasks applications in combination with QoS in the computational grid [J]. Future Generation Computer Systems, 2005, 21 (2): 271 - 280.
- [7] 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 (增刊): 11 - 16.
Chen J, Kong L, Pan X. Research on grid resource scheduling algorithm integrating forecast mechanism with QoS constraint [J]. Journal of Computer Research and Development. 2008, 45 (Suppl.): 11 - 16. (in Chinese)
- [8] M Dorigo, V Maniezzo, A Colomi. Positive feedback as a search strategy [R]. Milano, Italy: Technical Report 91 - 016, Politecnico di Milano, 1991: 91 - 106.
- [9] Zhihong X, Xiangdan H, Jizhou S. Ant algorithm-based task scheduling in grid computing [A]. Proc of Electrical and Computer Engineering [C]. Canadian: IEEE Computer Society Press, 2003: 1107 - 1110.

- [10] Jing H, Mingchu L, Weifeng S, Yuanfang C. An ant colony optimization for grid task scheduling with multiple QoS dimensions [A]. Proc of Eighth International Conference on Grid and Cooperative Computing [C]. Lanzhou. 2009. 415 – 419.
- [11] S Ali, H J Siegel, M Maheswaran, et al. Task execution time modeling for heterogeneous computing [A]. Proc of the 9th Heterogeneous Computing Workshop [C]. Mexico: IEEE

Computer Society Press, 2000. 185 – 199.

- [12] 吴春明, 陈治, 姜明. 蚁群算法中系统初始化及系统参数的研究[J]. 电子学报, 2006, 34(8): 1530 – 1533.
- Wu C, Chen Z, Jiang M. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm [J]. Acta Electronica Sinica, 2006, 34(8): 1530 – 1533. (in Chinese)

作者简介



孙伟峰 男, 1978 年 11 月出生于黑龙江牡丹江. 2002、2007 年在中国科学技术大学获计算机科学技术学士、博士学位. 现为大连理工大学大学讲师、硕士生导师, 主要研究方向为无线网络、QoS、分布式网络.

E-mail: wfsun@dlut.edu.cn



覃振权 男, 1978 年 3 月出生于广东罗定. 2002、2007 年在中国科学技术大学获通信与信息科学系统学士、博士学位. 现为大连理工大学大学讲师、硕士生导师, 主要研究方向为感知网络, P2P 网络.

李明楚(通讯作者) 男, 1963 年 10 月出生于江西. 1998 年在加拿大多伦多大学 (University of Toronto) 获博士学位. 现为大连理工大学教授、博士生导师, 主要研究方向为网格技术、信息安全、QoS、信任模型、生物特征识别软件技术.

E-mail: li_mingchu@yahoo.com