

采用离散粒子群算法的网格任务安全级调度

朱海, 王宇平, 权义宁, 王晓丽

(西安电子科技大学计算机学院, 710071, 西安)

摘要: 针对异构网格环境中任务调度问题对所面临的安全性研究不足,在考虑了保密性、完整性和真实性等安全性因素的情况下,构造了相应的安全效益函数;依据网格节点的历史行为特点,提出了节点的信誉度动态评估策略;基于行为特点提出了一种离散粒子群算法,由此建立了任务安全级调度新模型. 算法基于安全调度的离散空间特征给出了粒子的位置表示方法,从而克服了连续空间解决离散问题所造成的解空间冗余问题. 采用分步计算和修改粒子位置的方式重新定义了粒子进化方程,避免了进化过程中速度之间的相互干扰问题. 为了防止算法陷入局部最优,引入了均匀扰动速度. 实验结果表明,与基于连续空间的粒子群算法和遗传算法相比,所提算法具有较快的收敛速度、较短的调度长度和较高的安全性能.

关键词: 网格计算;任务调度;安全模型;离散粒子群算法

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2010)06-0021-06

Security Scheduling of Grid Tasks Based on the Discrete Particle Swarm Algorithm

ZHU Hai, WANG Yuping, QUAN Yining, WANG Xiaoli

(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: For the lack of safety studies to the task scheduling under the heterogeneous grid environment, a security benefit function is constructed under the consideration of the performance requirements such as confidentiality, integrity, authenticity, and so on. A dynamic method to evaluate the credibility of nodes and a new discrete particle swarm optimization algorithm are proposed based on the characteristics of historical behavior of grid resource nodes. Then a new model to schedule task security levels is established. The algorithm gives representation for particle locations based on the discrete space features of security scheduling. Hence, solution space redundancy of scheduling based on continuous space is avoided. Particle evolution equation is redefined by means of stepwise calculation and modification of particle positions, which avoids the problem of mutual speed interference during the evolution. A uniform disturbance speed is introduced to prevent the algorithm into a local optimum. Experiments and comparisons with the particle swarm optimization algorithm based on continuous space and the genetic algorithm show that the proposed algorithm has a faster convergence speed, a shorter scheduling length, and a higher safety performance.

Keywords: grid computing; task scheduling; security model; discrete particle swarm algorithm

在网格环境中,任务调度系统是其重要的组成部分. 由于网格资源节点的开放性和异构性,在任务

调度执行时,既要保证结果正确也需要考虑敏感数据在执行或传输过程中不会被泄密、篡改或冒名顶

替等^[1],同时由于网络资源节点的不确定性和动态性,任务执行时可能由于节点行为属性失效而需要重新调度或延迟执行,因此任务在网格平台调度的可靠性或可信度也是一个必须考虑的因素^[2].随着可信计算的发展,网络信任模型的研究有了一定的发展,比较突出的工作如文献[3]在对等网络环境下提出了一种基于角色的信任模型,文献[4]针对网络资源节点的不确定性,参考社会人际关系信任模型建立了网格节点信任推荐机制,有效提高了任务在信任方面的服务质量需求.目前的这些研究工作基于不同研究侧面和数学工具提出了相应的信任模型,但是欠缺对信任的动态性和网格节点本身的固有安全性的考虑.本文结合网格任务调度问题的具体特点,首先考虑了网格节点的固有安全性,根据其采取的加密算法、Hash 函数及认证技术等类型构造了相应的安全效益函数,其次根据网格节点的行为表现,构建了网格节点的动态信誉度评估策略,从而提出一个网格任务调度的安全模型,这在一定程度上弥补了相关研究工作的不足.

网格任务调度问题被证明是一个 NP 完全问题,无需多项式的时间复杂性算法就可以找到全局最优解,不少学者将遗传算法、模拟退火算法和思维进化计算等^[5-6]应用到任务作业调度,并取得了很好的效果.粒子群算法作为一种新的智能算法^[7],一经提出就受到了广泛关注,目前已经在函数优化领域取得了成功的应用.对粒子群算法的研究目前主要集中在连续优化问题方面^[8-9],在离散组合优化方面也取得了一些进展,如文献[10]将网格任务调度问题映射到粒子连续运动空间后利用粒子群优化算法(PSO)求解,取得了不错的效果.但是,用基于连续空间的 PSO 求解任务调度离散问题时,算法生成的连续解与整数规划问题的目标函数评价价值之间存在多对一的映射,不能完全反映粒子群优化算法中连续解的质量,此外整数规划问题的连续化会导致大量冗余解空间和冗余搜索,影响算法的收敛速度.鉴于这些不足,本文结合任务安全调度问题的具体特点,基于实际问题的离散空间特征定义了粒子的位置表示方法,并对粒子的速度及其运算规则进行了重新定义,从而提出一种新的离散粒子群算法,并对算法的收敛性和复杂度进行了分析.本文还对算法进行了仿真实验,在同等条件下与基于连续空间的 PSO 和基于进化理论的遗传算法(GA)相比,本文算法具有更快的收敛速度及更好的调度长度和安全性.

1 任务安全调度模型

网格节点的固有安全性主要是由资源节点采用的算法或方法决定的,例如保密性一般主要依赖不同加密算法实现,完整性主要通过不同的 Hash 函数来确保,真实性使用不同的数字签名技术对通讯实体身份的真实性进行鉴别,以保证所处理的任务是由合法的用户提交的.下面以保密性为例,说明如何合理设计网络资源节点的保密性安全水平值.假定在网格环境下所有资源节点共采用了 8 个加密算法^[11],将性能最低的 IDEA 加密算法对应的保密性安全水平值假定为 1,则其他任一网络资源节点 p_j 上加密算法的安全水平值根据其性能由下式计算

$$S_j^e = 13.5/\mu_j^e(k), 1 \leq k \leq 8 \quad (1)$$

这样,根据加密算法的不同性能将其安全值设定为 0.08~1 之间的值.同理,网络资源节点 p_j 根据其上采用的 Hash 函数和数字签名技术类型,得到其完整性和真实性水平值分别为 S_j^g 和 S_j^a .

定义 1 任一网络资源节点 p_j 若只考虑保密性、完整性和真实性等安全性能,则其综合安全效益函数可定义为

$$\begin{aligned} S_j &= \omega_1 S_j^e + \omega_2 S_j^g + \omega_3 S_j^a \\ \text{s. t. } \sum_{i=1}^3 \omega_i &= 1, \omega_i \geq 0 \\ 0 \leq S^k &\leq 1, k \in \{e, g, a\} \end{aligned} \quad (2)$$

式中: ω_i 表示不同安全服务的权重,权重越大表示其相应的安全服务越重要.

定义 2 节点的信誉度为一个随时间变化的函数,除了和网络资源节点的累计使用率有关,还与节点在累计执行任务时获得的信任评价相关.节点 p_j 在网络系统中的信誉度评估策略可定义为

$$\begin{aligned} R_j &= \alpha \times e^{-B_j} + \beta \times D_j + \gamma \times U_j \\ \text{s. t. } 0 &< \alpha, \beta, \gamma < 1 \\ \alpha + \beta + \gamma &= 1 \end{aligned} \quad (3)$$

式中: α 为时间衰减权重系数; β 为节点 p_j 累计执行调度任务时获得的信任评价权重系数; γ 为资源累计使用率权重系数; B_j 为节点 p_j 最近一次成功执行任务到目前为止的时间间隔, $B_j > 0$; U_j 为节点 p_j 的累计使用率,可用节点 p_j 的累计执行任务时间与累计活跃时间的比值来表示; D_j 为节点 p_j 累计执行任务时获得的信任评价,包含 3 个因子即时间、安全值和信誉度,其计算方法可参见文献[12].

定义 3 安全模型由安全效益函数和信誉度评估策略组成,分别表征网格节点固有安全性(包括真实

性、保密性和完整性等)和行为安全性(如信誉度及可信性等).将任务 t_i 被分配到网格资源节点 p_j 上执行获得的安全性定义为

$$\begin{aligned} Q(i, j) &= \omega_1 S(i, j) + \omega_2 R(i, j) \\ \text{s. t. } 0 &\leq \omega_1, \omega_2 \leq 1, \\ \omega_1 + \omega_2 &= 1 \end{aligned} \quad (4)$$

式中: ω_1, ω_2 分别代表任务 t_i 在节点 p_j 上获得的固有安全性和行为安全性的权重; $S(i, j), R(i, j)$ 可分别由式(2)、(3)的定义转化获得.

定义4 任务模型由含 n 个任务的集合 T 组成,其中任务 $t_i (i=1, 2, \dots, n)$ 为一个二元组,即 $t_i = (E_i, S_i)$,其中 E_i 和 S_i 分别为 t_i 的执行时间向量和安全需求向量. $E_i = (e_i^1, e_i^2, \dots, e_i^m)$ 表示任务 t_i 在异构系统不同处理器上的执行时间, $S_i = (s_i^1, s_i^2, \dots, s_i^q)$ 表示任务 t_i 的 q 个不同安全性能需求.

定义5 处理器模型可描述为一个异构处理器集 $P = \{p_1, p_1, \dots, p_m\}$,任一处理器 p_j 为一个二元组 $p_j = (\rho_j, S_j)$,其中 ρ_j 为处理器的执行能力, S_j 为处理器提供的安全服务向量,即 $S_j = (s_j^1, s_j^2, \dots, s_j^q)$.

定义6 任务调度长度的任务 t_i 分配到网格资源节点 p_j 上的调度长度,由节点可用时间 a_j 、通信时间 b_{ij} 、任务执行时间 e_{ij} 和获得安全效益值花费时间 c_{ij} 组成

$$\begin{aligned} F(i, j) &= a_j + b_{ij} + e_{ij} + c_{ij} = \\ &a_j + (b_{ij}^0 + g_i/b_{ij}^1) + e_{ij} + c_{ij} \end{aligned} \quad (5)$$

式中: b_{ij} 包含节点间建立连接时间 b_{ij}^0 和数据传输时间 g_i/b_{ij}^1 ; c_{ij} 可由一个 $n \times m$ 矩阵 ST 表示,其中任一元素 $H(i, j)$ 表示任务 t_i 在机器 p_j 上获得安全效益值的时间花费

$$H(i, j) = \sum_{k \in \{a, e, g\}} H^k(i, j) \quad (6)$$

式(6)的计算方法可参考文献[13].

定义7 任务安全调度模型由任务模型、处理器模型、安全模型及分配调度方案组成,可表示为一个四元组 (T, P, S, Z) ,其中 Z 为分配调度方案集合,即 $Z = \{z_1, z_2, \dots, z_n\}$. 分配调度方案为二元组 $z = (a, \varphi)$,其中 a 表示将任务集 T 中 n 个任务映射到处理器集 P 上的分配方案, φ 表示单个处理器 t_i 上的调度方案.异构网格系统下任务安全调度问题即为寻找一种最优分配调度方案,将任务集 T 中所有任务合理地分配到处理器集 P 上调度执行,使完成这 n 个任务的调度长度最短及获得的安全性最高,其数学模型表示如下

$$\begin{aligned} \max \quad & \sum_{i=1}^n \sum_{j=1}^m x_{ij} Q(i, j) \&\& \min \sum_{i=1}^n \sum_{j=1}^m x_{ij} F(i, j) \\ \text{s. t. } \quad & x_{ij} = \begin{cases} 1, & \text{表示任务 } t_i \text{ 被分配到} \\ & \text{机器 } p_j \text{ 上执行} \\ 0, & \text{其他} \end{cases} \end{aligned} \quad (7)$$

$$\sum_{j=1}^m x_{ij} = 1, i = 1, 2, \dots, n$$

式(7)中调度任务获得的安全性越高,表明任务执行越稳定,执行的结果愈加可信;任务调度长度越短,达到的目标越优,表明其性能越好.在对目标进行处理时可借鉴文献[14]中的隶属度函数方法,将其转化为单目标模型进行求解

$$\begin{aligned} \min(z') &= \sum_{i=1}^n \sum_{j=1}^m [\omega_1 F(i, j) + \omega_2 (1 - \\ &Q(i, j))] = \sum_{i=1}^n \sum_{j=1}^m \omega_k y_{ij}^k \end{aligned} \quad (8)$$

$$0 \leq \omega_1, \omega_2 \leq 1, \omega_1 + \omega_2 = 1$$

根据任务特性及要求,可通过调整权重 $\omega_k (k=1, 2)$ 比例对任意目标重新进行优化,以满足不同网格用户的倾向性需求.

2 离散粒子群算法

在求解任务安全调度(组合优化)问题时,由于当前粒子与最优粒子位置难以用定量的方式表示,导致粒子速度和位置不能参照标准粒子群进化方程更新,此时 PSO 基本失效.

2.1 粒子进化方程的重新定义

定义8 粒子的位置用 n 维向量 $\mathbf{X} = (x_1, \dots, x_i, \dots, x_n)$ 定义,其表示一个任务分配调度方案.向量中 $1 \leq x_i \leq m$ 表示第 i 个任务分配到了节点 x_i 上,不同任务分配到同一节点上的调度次序采用短任务优先或安全级别高的任务优先的调度准则.

定义9 两个位置相减的结果为速度 V ,使用符号“ $\ominus$ ”表示新的减法运算.个体极值 P_{best}^i 和全局极值 G_{best} 对当前粒子位置的影响程度用速度表示,其定义为

$$V_1 = P_{\text{best}}^i \ominus X_i = c_1 h(P_{\text{best}}^i, X_i) \quad (9)$$

$$V_2 = G_{\text{best}} \ominus X_i = c_2 h(G_{\text{best}}, X_i) \quad (10)$$

式中: c_1, c_2 为影响因子(也称学习因子);函数 $h(X_i, P_{\text{best}}^i)$ 为 X_i 对 P_{best}^i 的学习操作,其 X_i 向 P_{best}^i 第 j 维分量学习的过程为

$$h(x_{i,j}, P_{\text{best}}^{i,j}) = \begin{cases} P_{\text{best}}^{i,j}, & p_1 \leq \text{rand}[0, 1] \\ x_{i,j}, & \text{其他} \end{cases} \quad (11)$$

式中: p_1 为个体极值 P_{best}^i 对当前粒子影响程度的概率参数; $\text{rand}[0,1]$ 为 $[0,1]$ 间的随机数. X_i 对全局极值 G_{best} 的学习操作类似.

定义 10 位置与速度的加法运算实现了粒子位置的移动,使粒子进入了一个新的位置,使用符号 $\oplus$ 表示新的加法运算,其定义为

$$X' = X_i \oplus V_j = (x_{i1} \oplus v_{j1}, \dots, x_{ij} \oplus v_{jj}, \dots, x_{in} \oplus v_{jn}) \quad (12)$$

新位置的任一维 x'_{ij} 的值由下式获得

$$x'_{ij} = x_{ij} \oplus v_{jj} = w(x_{ij}, v_{jj}) \quad (13)$$

由于离散量运算的特殊性,惯性的主要作用是产生扰动,旨在增加粒子种群的多样性,避免算法早熟及局部收敛. 本文引入了一个均匀扰动速度,即依次以等概率从 $[1, m]$ 中随机产生一个 $v'_i \in [1, m]$, 连续执行 n 次,得到一个新的速度 $V_{\text{even}} = (v'_1, \dots, v'_n)$ 作为扰动速度. 该扰动速度有利于保持种群的多样性,可以有效防止算法陷入局部最优,而且该扰动速度具有简单、易执行的特点.

定义 11 为使速度之间不会相互干扰,采用分步计算速度和修改粒子位置的方式,更新后的粒子进化方程定义为

$$\left. \begin{aligned} X &= X \oplus V_{\text{even}} \\ X &= X \oplus V_1 = X \oplus c_1 h(P_{\text{best}}^i, X) \\ X &= X \oplus V_2 = X \oplus c_2 h(G_{\text{best}}, X) \end{aligned} \right\} \quad (14)$$

2.2 离散粒子群算法设计

基于前文重新定义的粒子进化方程,网格任务安全调度问题的离散粒子群算法 DPSO 流程如下.

步骤 1 定义粒子群的种群规模 pop 及最大进化代数 maxgen , 随机初始化粒子群中粒子的位置与速度.

步骤 2 将粒子的 P_{best}^i 设置为当前位置, G_{best} 设置为初始群体中最佳粒子的位置.

步骤 3 对离散粒子群中的所有粒子执行如下操作:

(1) 产生一个均匀扰动速度 V_{even} , 将其作用于当前粒子;

(2) 运用式(8)计算粒子群中所有粒子的适应度值,运用式(9)、(10)及(14)更新粒子速度和位置.

步骤 4 更新各粒子的个体极值 P_{best}^i , 以及根据各粒子的个体极值更新全局极值 G_{best} .

步骤 5 判断算法终止条件是否满足,如果满足转向步骤 6,否则转向步骤 3.

步骤 6 输出全局极值 G_{best} .

2.3 算法收敛性及复杂度分析

定理 1 (全局收敛性) DPSO 算法以概率 1 收敛到

问题的全局最优解.

证明 (1) 在 DPSO 算法中,对于任意两个粒子 a 与 b , b 是由 a 通过粒子进化可达的.

设粒子 $c = (c_1, c_2, \dots, c_n)$ 是由 $a = (a_1, a_2, \dots, a_n)$ 通过均匀扰动速度 V_{even} 作用后产生的任一后代,由均匀扰动的过程可知,从 a_i 产生 c_i 的概率为 $(m-1)^{-1}$, 则

$$P\{E(a) = c\} = (m-1)^{-1}(m-1)^{-1} \dots$$

$$(m-1)^{-1} = (m-1)^{-n} > 0 \quad (15)$$

由于粒子参与进化的概率 p_g 为 1, 则粒子 a 通过粒子位置更新产生粒子 b 的概率为

$$P\{E(a) = b\} \geq P\{E(a) = c\} * P\{G(c) = b\} = p_g(m-1)^{-n} > 0 \quad (16)$$

即对于可行域 S 中的任意两粒子 a 与 b , b 是由 a 通过粒子进化可达的.

(2) 由 DPSO 算法的粒子更新过程知, $X(0)$, $X(1), \dots, X(t), \dots$ 是单调的, 即满足 $\forall t$, 有

$$\min\{f(x) \mid x \in X(t)\} \leq \min\{f(x) \mid x \in X(t+1)\} \quad (17)$$

综合证明(1)、(2)可知, DPSO 算法以概率 1 收敛到任务调度问题的全局最优解.

下面讨论 DPSO 的时间复杂度. 由 DPSO 算法输入信息知每个粒子包含 n 个任务, 种群的规模 sizeofGroup 假定为 m , 则种群初始化时的算法时间复杂度为 $O(mn)$. 初始化 P_{best}^i 与初始化种群的复杂度均为 $O(mn)$, 初始化 P_{best}^i 和在 m 个数中找最大数这一问题的复杂度均为 $O(m)$, 因此可得算法初始化阶段的时间复杂度为 $O(m(2n+1))$. 在粒子种群迭代阶段, 每个粒子生成 V_{even} 的复杂度为 $O(n)$, 获得速度 V_1 和 V_2 的复杂度与粒子规模相关, 包括 2 次乘法和 1 次赋值, 复杂度为 $O(3n)$, 之后是运用粒子进化方程更新粒子的操作, 其时间复杂度为 $O(3n)$. P_{best}^i 和 G_{best} 的更新嵌入到了迭代环节中, 每步都要进行比较运行, 其更新是概率事件, 时间复杂度可分别表示为 $O\left(\frac{mn}{2}\right)$ 和 $O(n)$, 则整个粒子种群

单步迭代过程的时间复杂度可表示为 $O\left(m(3n+3n+n)+\frac{mn}{2}+m\right)$. 由于算法最坏情况下将执行到最大迭代次数, 即第 t 代结束, 故算法整体复杂度可表示为 $O(m(2n+1))+O\left(t\left(m(3n+3n+n)+\frac{mn}{2}+m\right)\right)$, 即为 $O(mnt)$.

3 仿真实验

为了验证本文提出的 DPSO 算法的高效性,分别从任务调度的性能指标(调度长度和安全性)和算法执行效率(收敛性)方面进行了仿真实验,并与求解同类问题的基于连续空间的 PSO^[10] 和基于进化思想的 GA^[15] 进行了对比分析。

实验 1 为比较算法的收敛特性,给出了本文算法、PSO 及 GA 在 500 次迭代过程中的调度长度变化曲线,如图 1 所示,本文算法和 PSO 比 GA 具有更好的收敛性,且本文的收敛性要优于 PSO。

实验 2 性能指标比较。

(1)对 $m=20$ 个网格节点,在任务数 n 变化(为 100~500 个子任务)的情形下,本文算法、PSO 及 GA 各运行 20 次的平均性能指标如图 2 所示。

在网格环境不变的情况下,由图 2a 可知,随着任务数的增加,任务调度长度都有所增加,其中本文算法和 PSO 明显优于 GA,特别是随着问题规模的变大其优势更明显。由图 2b 可知,随着任务数增大,任务对在网格节点上获得较优安全值的竞争更激

烈,致使任务平均安全值都有所降低,其中在同等条件下本文算法明显占优。

(2)固定任务数 $n=200$,当网格环境节点数变化时不同性能指标的比较结果如图 3 所示。

由图 3a 可知,随着节点数的扩充,任务调度长度都有所减小,这主要是因为任务在调度时可以选择性能更好的网格节点执行,从而减小了任务调度长度。由图 3b 可知,网格节点数扩充时任务获得的安全值有所提升,其中本文算法在同等条件下获得的值最大。

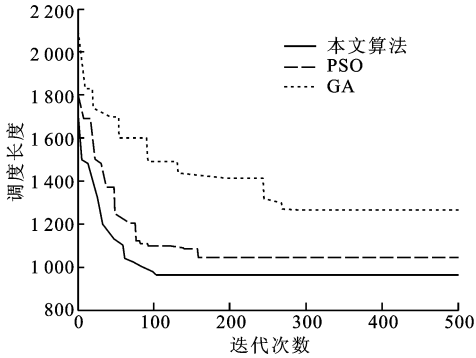
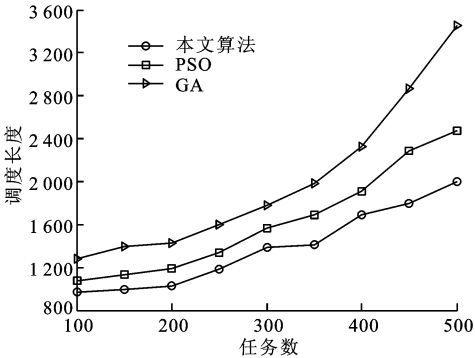
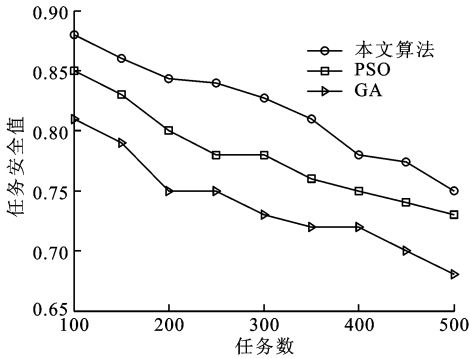


图 1 本文算法与 GA 的收敛性比较

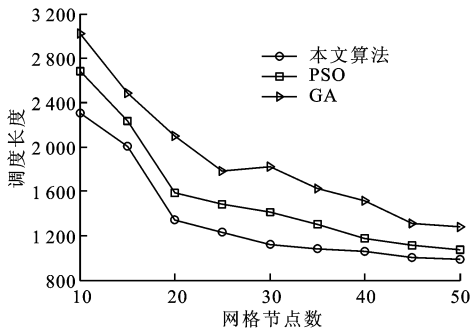


(a)任务调度长度比较

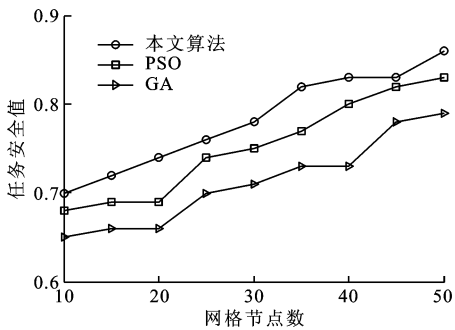


(b)任务安全值比较

图 2 节点数固定任务数变化时算法的性能指标比较



(a)任务调度长度比较



(b)任务安全值比较

图 3 任务数固定网格节点数变化时算法的性能指标比较

4 结 论

本文通过对异构网格任务调度问题进行综合分析,建立了一个网格任务安全级调度新模型,并提出了一种新的离散粒子群算法对模型进行求解,下一步将着重对依赖网格任务调度和动态多目标的网格任务调度问题进行研究探讨。

参考文献:

- [1] CHAKRABARTI A, DAMODARAN A, SENGUP-TA S. Grid computing security: a taxonomy [J]. IEEE Security & Privacy, 2008, 6(1): 44-51.
- [2] SONG S S, HWANG Kai, KWOK Yu Kwong. Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling [J]. IEEE Trans on Computers, 2006, 55(6): 703-719.
- [3] PARK J S, AN G, CHANDRA D. Trusted P2P computing environments with role-based access control [J]. Information Security, 2007, 1(3): 27-35.
- [4] 袁禄来,曾国荪,姜黎立,等. 网格环境下基于信任模型的动态级调度[J]. 计算机学报, 2006, 29(7): 1217-1224.
YUAN Lulai, ZENG Guosun, JIANG Lili, et al. Dynamic level scheduling based on trust model in grid computing [J]. Chinese Journal of Computers, 2006, 29(7): 1217-1224.
- [5] BRAUN T D, SLEGEL H J, BECJ N. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems [J]. IEEE Trans on Parallel and Distributed Computing, 2001, 61(6): 810-837.
- [6] 薛正华,刘伟哲,董小社,等. 基于思维进化的集群作业调度方法研究 [J]. 西安交通大学学报, 2008, 42(6): 651-654.
XUE Zhenghua, LIU Weizhe, DONG Xiaoshe, et al. Research on mind evolutionary computation based job scheduling for server clusters [J]. Journal of Xi'an Jiaotong University, 2008, 42(2): 651-654.
- [7] KENNEDY I, EBERHART R. Particle swarm optimization [C]//Proceedings of the Fourth IEEE International Conference on Neural Networks. Piscataway, NJ, USA: IEEE Service Center, 1995: 1942-1948.
- [8] 倪庆剑,张志政,邢汉承. 一种基于可变多簇结构的动态概率粒子群优化算法[J]. 软件学报, 2009, 20(2): 339-349.
NI Qingjian, ZHANG Zhizheng, XING Hancheng. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure [J]. Journal of Software, 2009, 20(2): 339-349.
- [9] 王峰,邢科义,徐小平. 系统辨识的粒子群优化方法[J]. 西安交通大学学报, 2009, 43(2): 116-120.
WANG Feng, XING Keyi, XU Xiaoping. A system identification method using particle swarm optimization [J]. Journal of Xi'an Jiaotong University, 2009, 43(2): 116-120.
- [10] 季一木,王汝传. 基于粒子群的网格任务调度算法研究[J]. 通信学报, 2007, 28(10): 60-66.
JI Yimu, WANG Ruchuan. Study on PSO algorithm in solving grid task scheduling [J]. Journal on Communications, 2007, 28(10): 60-66.
- [11] XIE T, QIN X. Security-aware resource allocation for real-time parallel jobs on homogenous and heterogeneous clusters [J]. IEEE Trans on Parallel and Distributed Systems, 2008, 19(5): 682-697.
- [12] ZHU H, WANG Y P. Evolutionary algorithm for solving constrained multi-objective grid tasks scheduling problem [C]//Proceeding of the 2009 International Conference on Computer Network and Multimedia Technology. Washington, DC, USA: IEEE Computer Society, 2009: 10-14.
- [13] ZHU H, WANG Y P. Security-driven task scheduling based on evolutionary algorithm [C]//Proceedings of the 2008 Computational Intelligence and Security. Washington, DC, USA: IEEE Computer Society, 2008: 451-456.
- [14] 孟宪福,张晓燕. 对等网络环境下多目标约束的并行任务调度策略研究 [J]. 计算机集成制造系, 2008, 14(4): 761-766.
MENG Xianfu, ZHANG Xiaoyan. Parallel task scheduling strategy with multi-objective constraints in P2P [J]. Computer Integrated Manufacturing Systems, 2008, 14(4): 761-766.
- [15] WU A S, YU H, JIN S, et al. An incremental genetic algorithm approach to multiprocessor scheduling [J]. IEEE Trans on Parallel and Distributed Systems, 2004, 15(9): 824-834.

(编辑 武红江 苗凌)