

基于多 QoS 约束条件的广域信息管理系统任务调度算法

李罡^{1,2}, 吴志军³

(1. 天津大学电气自动化与信息工程学院, 天津 300072; 2. 白城师范学院机械工程学院, 吉林 白城 137000;
3. 中国民航大学电子信息与自动化学院, 天津 300300)

摘要: 提出了面向广域信息管理系统 (SWIM) 的多 QoS 约束条件的蚁群优化任务调度算法 (QoS-ACO)。针对 SWIM 中用户对任务请求完成服务质量 (QoS) 的要求, 综合考虑任务完成时间、执行安全性和可靠性因素, 构造了新的用户综合满意度评价函数和系统任务调度模型, 使用 SWIM 中业务调度 QoS 总效用评价函数来更新蚁群算法中的信息素。仿真实验结果表明, 同等条件下 QoS-ACO 算法在任务完成时间、安全性、可靠性和 QoS 总效用值方面都优于传统 Min-Min 算法和粒子群优化算法, 满足用户的任务调度服务质量要求, 较好地完成了 SWIM 调度任务。

关键词: 广域信息管理系统; 服务质量; 任务调度模型; 蚁群优化任务调度算法; 粒子群优化算法

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019166

Task scheduling algorithm for system-wide information management based on multiple QoS constraints

LI Gang^{1,2}, WU Zhijun³

1. School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

2. School of Mechanical Engineering, Baicheng Normal University, Baicheng 137000, China

3. School of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Abstract: An ant colony optimization task scheduling algorithm based on multiple quality of service constraint (QoS-ACO) for SWIM was proposed. Focusing on the multiple quality of service (QoS) requirements for task requests completed in system-wide information management (SWIM), considering the task execution time, security and reliability factors, a new evaluate user satisfaction utility function and system task scheduling model were constructed. Using the QoS total utility evaluation function of SWIM service scheduling to update the pheromone of the ant colony algorithm. The simulation results show that under the same conditions, the QoS-ACO algorithm is better than the traditional Min-Min algorithm and particle swarm optimization (PSO) algorithm in terms of task completion time, security, reliability and quality of service total utility evaluation value, and it can ensure that the user's task scheduling quality of service requirements are met, and can better complete the scheduling tasks of the SWIM.

Key words: system-wide information management, quality of service, task scheduling model, ant colony optimization task scheduling algorithm, particle swarm optimization algorithm

收稿日期: 2018-09-28; 修回日期: 2019-03-27

通信作者: 吴志军, zjwu@cauc.edu.cn

基金项目: 天津市自然科学基金资助项目 (No.17JCZDJC30900); 国家自然科学基金资助项目 (No.61601467); 中央高校基本科研业务费专项资金资助项目 (No.3122018D007)

Foundation Items: The Natural Science Foundation of Tianjin (No.17JCZDJC30900), The National Natural Science Foundation of China (No.61601467), The Fundamental Research Funds for the Central Universities (No.3122018D007)

1 引言

1997年,欧洲航空安全组织向美国联邦航空局提出广域信息管理系统(SWIM, system-wide information management)概念^[1],国际民航组织(ICAO, International Civil Aviation Organization)于2002年接受了这个概念^[2]。SWIM是一个集成度很高的大规模网络系统,它可以对内部资源进行整合和管理,从而提高系统信息的利用率和资源的共享率。SWIM作为一个信息共享平台,为不同的民航业务子系统提供统一的数据传输和交换机制,并以提高子系统业务处理效率为目的,实现民航各相关单位之间的信息共享性和互通性^[3]。在空管、航空公司、机场等空中交通运输参与者之间实现一体化、标准化、灵活的通信、导航、监视、运行、气象、情报数据的发布与共享,满足未来民航高效协同运行的需求。

为了提高广域信息管理系统的资源利用与共享率,需要解决系统任务调度问题,对用户提出的需求找到合适节点进行任务处理,并且应在SWIM任务调度过程中综合考虑用户的服务质量(QoS, quality of service)需求。蚁群算法具有并行性和顽健性的特点,可以很快取得高质量的解,已经成功解决了很多不同类型的调度问题。本文在总结现有的网络任务调度技术并结合SWIM自身特点的基础上,以多QoS约束条件为立足点,利用蚁群优化(ACO, ant colony optimization)算法提出了一种改进传统蚁群算法的SWIM任务调度算法QoS-ACO,保证用户任务调度QoS要求尽可能地得到满足,提高SWIM整体性能。

2 相关工作

目前,国内外专家学者在大型网络系统的任务调度方面进行了很多研究,如针对现在比较流行的云计算、网格等网络系统的研究。Min-Min算法^[4]、Max-Min算法^[5]、Max-Int算法^[6]和遗传算法(GA, genetic algorithm)^[7]等任务调度算法都是经典的研究成果,在网络系统任务调度中得到应用并取得了比较好的效果。此外,Buyya等^[8]还提出一种基于经济模型的优化调度模型,为研究任务调度算法提供了新的思路。但这些算法也存在一些局限性和不足,它们都是基于最小化调度目标的思想提出的,而在任务调度算法的设计和改进中很少考虑到用

户对服务质量的需求。

现在也有一些针对服务质量方面的任务调度方法研究。Castillo等^[9]在解决资源调度过程中所产生的碎片问题和服务质量研究中,应用了数学几何方法建立资源预留调度机制。刘宴兵等^[10]提出的S-GTSA(grid task scheduling algorithm based on QoS similarity)在任务调度过程中应用了服务质量相似度,在一定程度上满足了用户的QoS需求,但在任务完成时间跨度方面的考虑不够全面。邓见光等^[11]针对云计算中的QoS任务调度问题进行了研究,对任务调度的多QoS目标约束条件进行形式化建模,提出了一种多QoS目标约束的云计算任务调度策略。文献[12]研究了一种基于离散粒子群优化的可重构系统任务调度方法,在任务调度过程中配置阈值,有效隐藏了任务的配置时间,提高了系统执行性能,但其目标单一,缺少对其他QoS参数的考虑。文献[13]设计的任务调度策略较合理,在任务调度过程中考虑的服务质量参数也比较全面,提高了系统任务调度效率,避免了系统负载不均衡现象的发生。

网络应用的广泛普及和网络技术的更新发展对任务调度的研究提出了新的要求,也对任务调度的服务质量提出了更高标准。本文分析、参考了大量分布和异构环境下的网络任务调度算法,在前人研究工作的基础上,结合广域信息管理系统自身的特点,针对SWIM中任务调度服务质量在任务完成时间、安全性和可靠性方面需求的特殊性进行了研究。通过采用并改进传统蚁群算法,提出了基于多QoS约束条件的广域信息管理系统任务调度算法QoS-ACO。仿真实验结果表明,所提出的QoS-ACO任务调度算法具有很好的综合QoS性能,在任务调度完成效率和满足客户服务质量需求方面的效果都比较好。

3 多QoS约束条件SWIM任务调度模型

3.1 SWIM任务调度模型

服务质量的约束条件有很多,在本文SWIM任务调度模型和QoS-ACO算法中,主要从以下3个方面进行考虑。

- 1) 时间。完成用户提交任务所需要的时间。
- 2) 安全性。在SWIM节点上可以获得的安全等级。
- 3) 可靠性。任务被成功执行的概率。

SWIM 分为以下三层。

1) SWIM 共享数据应用层 (SDA, shared data application)。实现空管、机场和航空公司等客户端发出服务请求。

2) SWIM 公共信息管理层 (CIM, common information management)。负责系统中资源配置、负载均衡监控和队列管理等内容。

3) SWIM 公共数据传输层 (CDT, common data transport)。实现对分布的共享航空飞行等资源数据的采集。

在 CIM 中, SWIM 任务调度中心能完成请求服务的任务调度、任务解析、任务分配以及建立服务虚拟机等功能。用户通过任务调度中心可以完全透明化地接入 SWIM 资源环境中。任务调度中心对系统资源和用户的需求进行统一分配和管理, 针对不同任务的各自特点和资源需求, 在任务调度模块中执行相应的任务调度算法。任务解析模块解析任务特点、获取特性, 把任务调度分配到合适的服务资源上进行执行, 保证 QoS 需求得到满足。由于服务节点在整个 SWIM 中的分布异构性, 分配调度模块结合采集到的共享资源数据, 按照任务解析模块对任务的分配要求来匹配资源, 把最终的资源节点信息发送到虚拟机模块上, 由其为用户提供所需的服务。多 QoS 约束条件 SWIM 任务调度模型架构如图 1 所示。

3.2 相关定义

为更好地描述 QoS-ACO 算法, 首先进行如下一般性定义。

1) SWIM 总资源集 $S = \{s_1, s_2, s_3, \dots, s_j, \dots, s_m\}$, 系统由 m 个资源节点构成, s_j 为第 j 个资源节点。

2) SWIM 总调度任务集 $T = \{t_1, t_2, t_3, \dots, t_i, \dots, t_n\}$, 包括 n 个调度任务, t_i 为系统中第 i 个调度任务。

3) 预期执行时间矩阵 (ETC, expected time to compute), $ETC = \begin{bmatrix} et_{11} & \dots & et_{1m} \\ \vdots & et_{ij} & \vdots \\ et_{n1} & \dots & et_{nm} \end{bmatrix}$, et_{ij} 为调度任务 t_i 在资源节点 s_j 上的预期完成时间。

4) 搜索表 Tabu 记录蚂蚁任务完成信息。

3.3 多 QoS 约束条件参数

为实现系统任务调度的 QoS 目标, 采用隶属度函数思想将任务完成时间、安全性和可靠性进行目标约束化处理, 具体过程如下。

1) 任务调度完成时间目标约束化处理

系统中参与任务调度的所有的调度任务完成时间为任务调度总完成时间。任务在不同节点上的执行时间会因系统的网络异构性和资源节点的性能差异性而不同, 所以任务的调度策略会直接影响系统任务的总完成时间。用资源节点上最后一个任务的完成时间来表示该节点的任务完成时间 T_i , 如式(1)所示。

$$T_i = t_{ig} \tag{1}$$

其中, t_{ig} 为资源节点 i 上最后一个任务的完成时间。NTG $\{T_i\}$ 为系统中所有的资源节点任务完成时间的集合, 以该集合中的最大值表示系统调度任务的总完成时间 TIM, 如式(2)所示。

$$TIM = \max(NTG) \tag{2}$$

所以, 实现 TIM 值的最小化就可以达到优化系

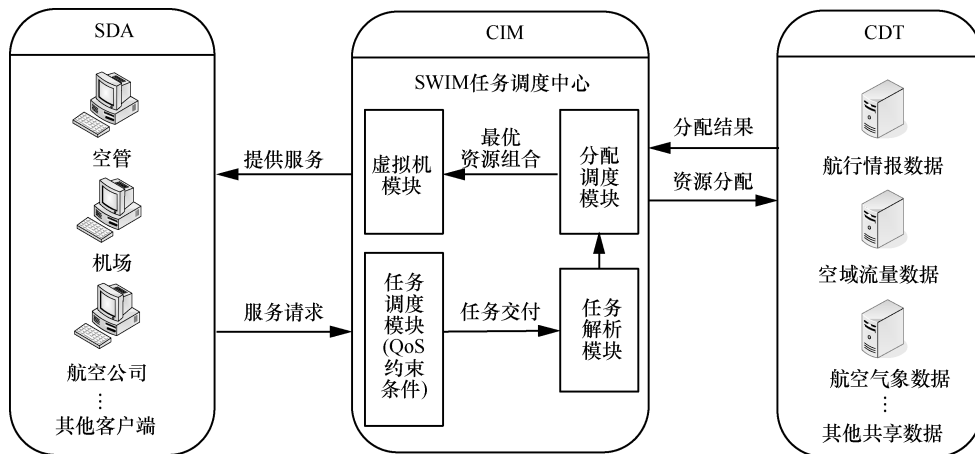


图 1 多 QoS 约束条件 SWIM 任务调度模型架构

统总完成时间的目的。

- 2) 任务调度安全性目标约束化处理
安全性定义为

$$SEC = \frac{\sum_{i=1}^n l_i \sec_i}{\sum_{i=1}^n l_i} \quad (3)$$

其中, n 为安全参数个数; l_i 为第 i 个安全数的权重; 当满足系统任务调度第 i 个安全参数时, $\sec_i=1$, 否则 $\sec_i=0$ 。

- 3) 任务调度可靠性目标约束化处理

任务调度的可靠性可以等效为在系统任务完成时间最小化的前提下, 完成所有任务调度的成功率。系统中节点 i 完成全部任务过程中的任务调度成功率用指数率来表示, 其表达式为

$$Q_{suc}^i = e^{-\delta_i T_i} \quad (4)$$

假设所有任务调度成功率互不影响, 则系统全部资源节点完成所有任务的成功率为

$$SUC = \prod_{i=1}^n Q_{suc}^i = e^{-\sum_{i=1}^n \delta_i T_i} \quad (5)$$

3.4 多 QoS 约束条件求解

SWIM 多 QoS 约束条件求解分为以下几种情况。

- 1) 把任务 t_i 调度到资源节点 s_j 上进行处理时获得的服务质量时间参数效用等效为时间评价函数, 如航空飞行计划任务在航行情报信息资源中进行分配时获得的 QoS 时间性效用。假设 $T_{\max t_i}$ 为任务 t_i 的最长完成时间, $T(t_i, s_j)$ 为任务 t_i 调度到资源节点 s_j 上需要的完成时间, λ 为常数, 则时间评价函数为

$$TIM_{\text{effort}}(t_i, s_j) = \begin{cases} 1, T(t_i, s_j) \leq T_{\max t_i} \\ e^{-\lambda(T(t_i, s_j) - T_{\max t_i})}, \text{其他} \end{cases} \quad (6)$$

- 2) 把某个任务 t_i 调度到资源节点 s_j 上进行处理时获得的服务质量安全性参数效用等效为安全性评价函数, 如航空飞行计划任务在航行情报信息资源中进行分配时获得的 QoS 安全性效用。假设 $S_{\min t_i}$ 为任务 t_i 应具有安全等级的最小值, S_{s_j} 为资源节点 s_j 具有的安全等级, S_{\max} 是系统资源节点具有的最高安全等级, 则安全性评价函数为

$$SEC_{\text{effort}}(t_i, s_j) = \begin{cases} 1, S_{\min t_i} \leq S_{s_j} \\ 1 - \frac{S_{\min t_i} - S_{s_j}}{S_{\max} - S_{s_j}}, \text{其他} \end{cases} \quad (7)$$

- 3) 把任务 t_i 调度到资源节点 s_j 上进行处理时获得的服务质量可靠性参数效用等效为可靠性评价函数, 如航空飞行计划任务在航行情报信息资源中进行分配时获得的 QoS 可靠性效用。本文用调度任务完成的成功率来表征其可靠性, 假设 $R_{\min t_i}$ 为任务 t_i 完成成功率的最小概论值, $R(t_i, s_j)$ 为把任务 t_i 调度到资源节点 s_j 上进行处理所获得的可靠性, 则可靠性评价函数为

$$REL_{\text{effort}}(t_i, s_j) = \begin{cases} 1, R(t_i, s_j) \leq R_{\min t_i} \\ \frac{1 - e^{-R(t_i, s_j)}}{1 - e^{-1}}, \text{其他} \end{cases} \quad (8)$$

假设系统资源节点 s_j 的固有非成功率为 F_{s_j} , 任务 t_i 在资源节点 s_j 上进行处理所需要的完成时间为 $T(t_i, s_j)$, 则

$$REL(t_i, s_j) = e^{(-T(t_i, s_j)F_{s_j})} \quad (9)$$

用户满意度评价标准的计算式为

$$E(t_i, s_j) = \varphi_1 TIM(t_i, s_j) + \varphi_2 SEC(t_i, s_j) + \varphi_3 REL(t_i, s_j) \quad (10)$$

其中, $E(t_i, s_j)$ 为调度任务 t_i 的服务质量总体效用评价函数, 作为用户满意度的评价标准, 用此函数值对蚁群优化算法中的信息素进行更新; 时间、安全性和可靠性评价函数对应的权值分别用常数 φ_1 、 φ_2 和 φ_3 来表示, 且 $\varphi_1 + \varphi_2 + \varphi_3 = 1$ 。

对总体效用评价函数 $E(t_i, s_j)$ 进行线性加权处理, 最终得到用户对任务完成服务质量的总效用评价函数, 其计算式为

$$E = \sum_{i=1}^n E(t_i, s_j) \quad (11)$$

系统资源调度的目标就是确保总效用评价函数 E 值最大, 以提高任务完成的服务质量, 也用此函数来评价任务调度算法的性能。

4 基于多 QoS 约束条件的蚁群优化任务调度算法

本文根据 SWIM 核心服务的特点对蚁群算法进行改进, 使用 SWIM 中业务调度 QoS 总效用评价函数来更新蚁群算法中的信息素, 提出了面向

SWIM 的多 QoS 约束条件蚁群优化任务调度算法 QoS-ACO，在保障完成 SWIM 业务调度的同时，能够最大化用户满意度，提高 SWIM 核心服务的质量。

4.1 QoS-ACO 算法优化规则

Dorigo 等^[14]通过观察自然界中的蚂蚁觅食行为产生了任务调度灵感，并于 1991 年首次提出了蚁群算法，模拟蚂蚁觅食行为来达到全局寻优目的。假设 t 时刻蚂蚁 k 从资源节点 i 转移到资源节点 j 的概率值为 $P_{ij}^k(t)$ ，利用该值来选择要调度的下一个任务。 $P_{ij}^k(t)$ 的计算式为

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{i=1}^n [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, j \in A \\ 0, \text{其他} \end{cases} \quad (12)$$

通过式(12)可知，影响资源节点选择的 2 个关键因子是 $\tau_{ij}(t)$ 和 $\eta_{ij}(t)$ ，对这 2 个因子的改进是算法优化研究的重点。本文提出的面向 SWIM 任务调度的多 QoS 约束条件蚁群优化任务调度算法，就是在传统蚁群算法基础上对 $\tau_{ij}(t)$ 做出相应改进，使用 SWIM 中业务调度 QoS 总效用评价函数 E 来更新蚁群算法中的信息素。

蚁群算法中更新信息素 τ_{ij} 的计算式为

$$\tau_{ij}(t+1) = (1-\mu)\tau_{ij}(t) + \mu\Delta\tau_{ij}(t) \quad (13)$$

本文使用 SWIM 中业务调度 QoS 总效用评价函数 E 来更新信息素 τ_{ij} ，其计算式为

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (14)$$

$$\Delta\tau_{ij}^k = \gamma E \quad (15)$$

蚁群算法中启发函数 $\eta_{ij}(t)$ 的表达式为

$$\eta_{ij}(t) = \frac{1}{et_j} \quad (16)$$

为保障完成所有任务的时间最小，利用预期完成时间获得启发函数。从式(16)可知，预期完成时间的值与 $\eta_{ij}(t)$ 成反比关系。 $\eta_{ij}(t)$ 和 $P_{ij}^k(t)$ 成正比关系，因此通过降低预期完成时间可以提高转移概率^[15]。

把改进后的信息素浓度函数式(13)和启发函数式(16)代入任务转移概率式(12)中，得到优化蚁群算

法 QoS-ACO 后的任务 t_i 被调度到资源节点 s_j 的概率值 $P_{ij}^k(t)$ 。本文提出的 QoS-ACO 算法计算复杂度为 $O(NC_{\max}n^3)$ ，其中， NC_{\max} 为循环次数， n 为任务总数。本节涉及的函数及参数描述如表 1 所示。

表 1 函数及参数描述

函数及参数	描述
$P_{ij}^k(t)$	任务转移概率
$\tau_{ij}(t)$	资源节点 i 到 j 在 t 时刻的信息素浓度函数
$\eta_{ij}(t)$	t 时刻资源节点 i 转移到资源节点 j 的启发函数
α	信息启发式因子
β	期望启发式因子
A	下一步可以选择的任务
μ	路径信息的残留因子， $0 < \mu < 1$
$1-\mu$	信息素挥发系数
$\Delta\tau_{ij}$	节点 i 到节点 j 路径上的信息素增量，初始时刻 $\Delta\tau_0=0$
et_j	任务 t_j 的预期完成时间

4.2 QoS-ACO 算法优化流程

QoS-ACO 算法主要利用蚁群算法优化任务调度安排，依据任务在资源节点上的总效用值来选择节点处理调度任务。本文设计的 QoS-ACO 算法需要在完成调度任务的同时最大化用户的满意度，尽量获得比较高的总体评价。因此使用 QoS 总效用评价函数来更新蚁群算法中信息素，并且期望完成所有任务的时间最小，使用预期完成时间来计算蚁群算法的启发函数。最终依据任务转移概率 $P_{ij}^k(t)$ 的值，选择一个信息素浓度函数 $\tau_{ij}(t)$ 值高的资源节点来处理分配的任务。

SWIM 任务调度的关键是保证总体效用评价函数 $E(t, s_j)$ 值最大，并将所有调度任务合理分配到相应的资源节点上进行处理。假设给定一个由 n 个任务组成的 SWIM 任务调度，其算法流程如图 2 所示，具体步骤如下。

1) 参数初始化。任务数量为 n ，资源数量为 m ，蚂蚁数量为 q ，最大的循环次数为 t_{\max} ，初始时间 t 设置为 0，在 SWIM 中假设有 n 个航空飞行计划任务， m 个航行情报信息资源。

2) 初始化搜索表 Tabu。

3) 利用 $P_{ij}^k(t)$ 的值来确定接下来的调度任务，并选择具有最大效用的资源节点进行任务处理。

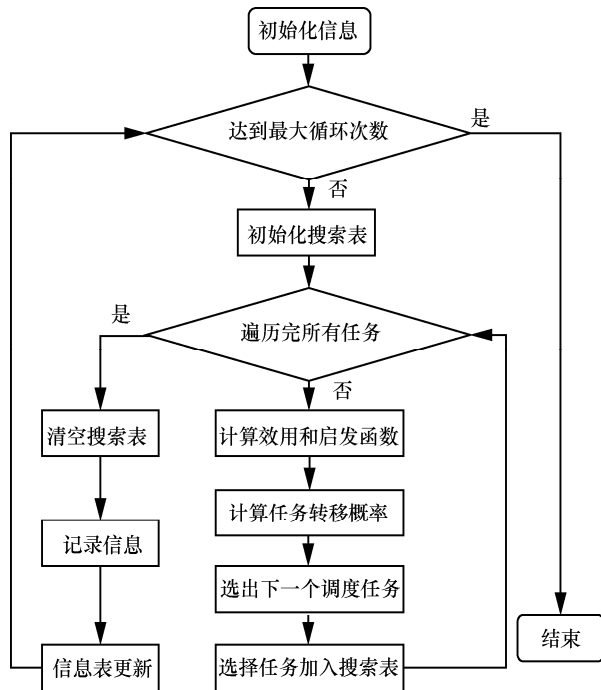


图2 QoS-ACO 算法流程

- 4) 把新的调度任务添加到搜索表中进行更新。
- 5) 根据式(13), 使用总效用评价函数 E 来进行

蚁群算法的信息素更新, 并把筛选出具有最佳调度结果的蚂蚁信息进行保存。

6) 重复以上步骤直至 $t > t_{max}$, 算法结束^[16]。

5 仿真实验及结果分析

为验证本文所提出算法的可行性和性能, 在参考了民航某空中交通管理局已经部署的广域信息管理系统结构基础上, 利用网络仿真软件 NS-2 搭建仿真实验环境。SWIM 局部任务调度结构如图 3 所示。

5.1 实验环境及参数设置

在 NS-2 网络仿真平台搭建的实验环境主要由调度器模块、服务器模块、客户端模块和路由器模块构成。调度器模块扮演调度者的角色, 路由器模块负责路由转发, 客户端模块和调度器模块通过路由器模块完成交互, 调度器的功能是发现客户可以访问的资源, 把任务映射到资源上后执行任务, 最后收集结果。服务器模块为具有不同处理能力的服务节点, 接收调度器模块分配的任务并执行, 同时定期向调度器模块汇报自己的运行状态。网络测试拓扑如图 4 所示。

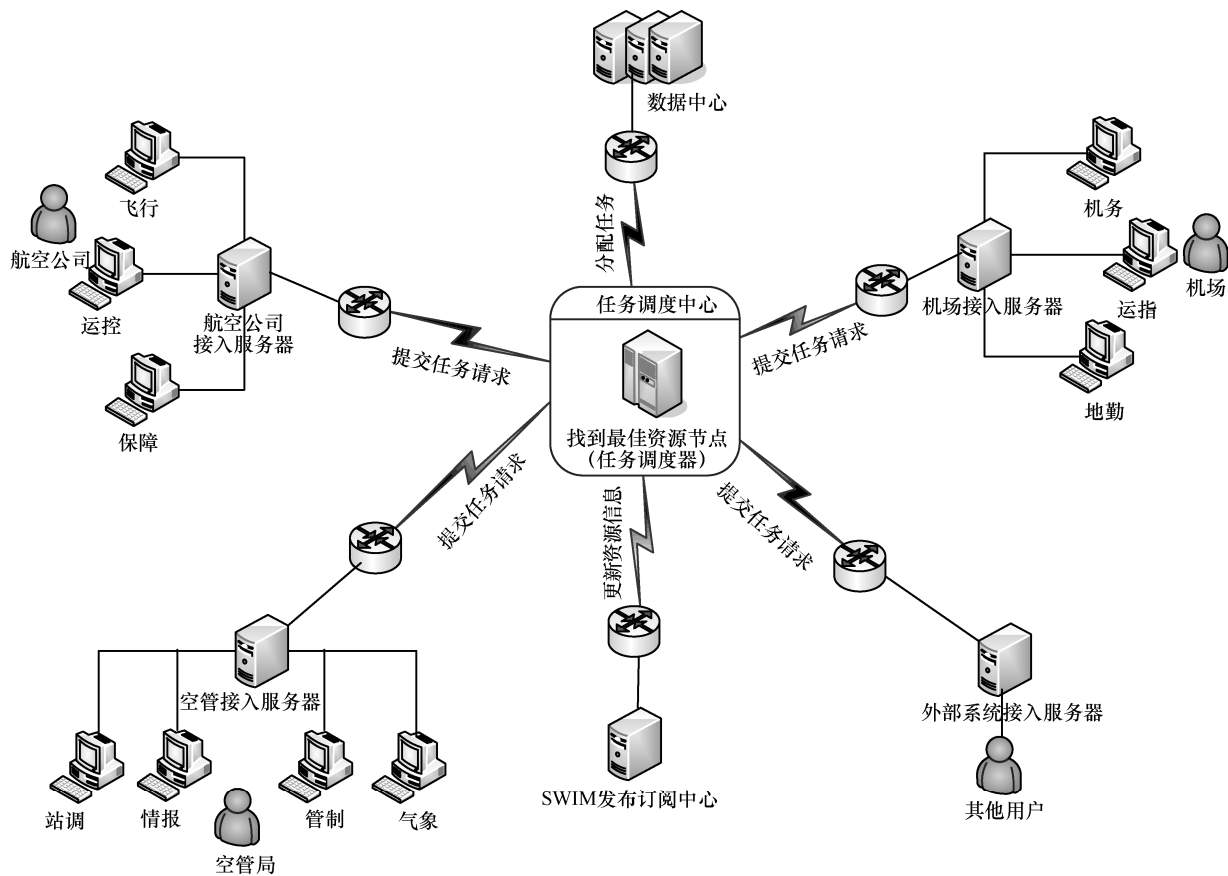


图3 SWIM 局部任务调度结构

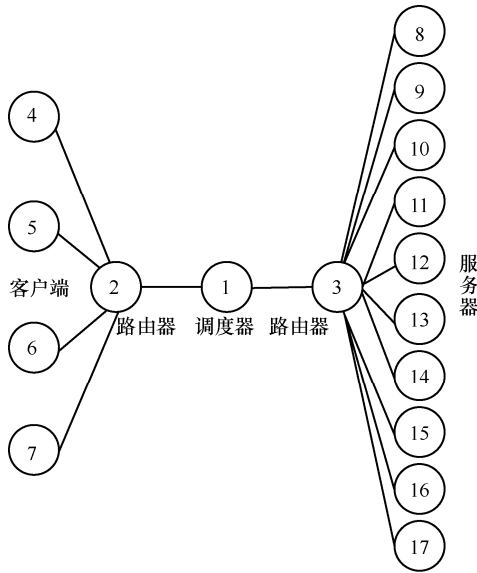


图4 仿真实验网络拓扑结构

仿真实验设计了一个具有一个调度器模块、2个路由器模块、10个服务器模块和4个客户端模块的拓扑结构模型。在模型中，节点1为系统调度器；节点2和节点3为路由器；节点4~节点7为客户端；节点8~节点17为服务器。

本实验选取了任务完成时间、安全性和可靠性作为三维 QoS 约束，在 NS-2 环境的仿真平台上对改进的蚁群优化算法进行仿真实验，并进行如下参数设置。

1) 任务完成时间。在任务调度过程中，任务的完成时间一定要小于用户定义的最大时间，任务的执行期限计算式为

$$T_{\max t_i} = \gamma_i e t_i + k t_i \quad (17)$$

其中， γ_i 为 1~6 的随机值， $e t_i$ 是任务 t_i 的平均完成时间， $k t_i$ 是任务的到达时间。任务完成时间评价函数如式(6)所示。

2) 安全性。安全性级别设置为 {poor, low, medium, high} 4 个等级，安全性评价函数如式(7)所示。

3) 可靠性。每个服务节点可以成功完成任务的概率。可靠性评价函数如式(8)所示。

4) QoS 总效用评价函数的权值参数设置为： $\varphi_1=0.4$ ， $\varphi_2=0.3$ ， $\varphi_3=0.3$ 。

5) 参考文献[17]中的 ACO 算法设置： $\alpha=1.0$ ， $\beta=2.0$ ， $\mu=0.5$ 。

6) 设置节点之间的链路带宽为 200 Mbit/s，单向时延为 20 ms。

7) 通过 NS-2 编译器创建调度服务器、调度路

由器、用户和资源节点服务器对象。

8) 运用 NS-2 的解释器连接调度服务器，将调度路由器加载到其相邻节点。

仿真实验测试参数如表 2 所示。

表2 仿真实验测试参数

参数名称	参数值
α	1.0
β	2.0
μ	0.5
带宽/(Mbit.s ⁻¹)	200
时延/ms	20
φ_1	0.4
φ_2	0.3
φ_3	0.3
种群规模/个	50
最大迭代次数	200

5.2 实验结果及分析

仿真实验通过任务完成时间、安全性、可靠性和 QoS 总效用值 4 个评价指标对本文提出的 QoS-ACO 算法进行综合评价，并在相同条件下将 QoS-ACO 算法与经典的 Min-Min 算法^[4]、启发式粒子群优化 (PSO, particle swarm optimization) 算法^[18] 进行比较分析，验证 QoS-ACO 算法的有效性。采用在 SWIM 资源节点数目 m 和任务数 n 不同情况下进行 2 种实验。

1) 资源节点数量 m 固定，任务数量 n 变化。在构建的 SWIM 仿真系统中，资源节点个数 $m=10$ ，任务数量为 n ， $1 \leq n \leq 400$ ，比较 3 种算法在任务完成时间、安全性和可靠性值方面的性能，并通过总效用值来验证 QoS-ACO 算法的优越性。

2) 任务数量 n 固定，资源节点数量 m 变化。在构建的 SWIM 仿真系统中，任务数量 $n=200$ ，资源节点个数为 m ， $1 \leq m \leq 40$ ，比较 3 种算法在任务完成时间、安全性和可靠性值方面的性能，并通过总效用值来验证 QoS-ACO 算法的优越性。

为使验证结果更加准确，每种实验进行 1 000 次，取实验结果的平均值，并利用 Matlab 进行处理对比分析。

5.2.1 任务完成时间

任务完成时间可以反映出算法的高效性。在 NS-2 上运行 QoS-ACO 算法、PSO 算法和 Min-Min 算法，经过路由选择操作后确定最终路径完成任务

调度。通过 Wireshark 统计分析工具对得到的 trace 文件进行统计和分析, 并利用 Matlab 对结果进行处理, 得到 3 种算法的任务完成时间, 如图 5 和图 6 所示。

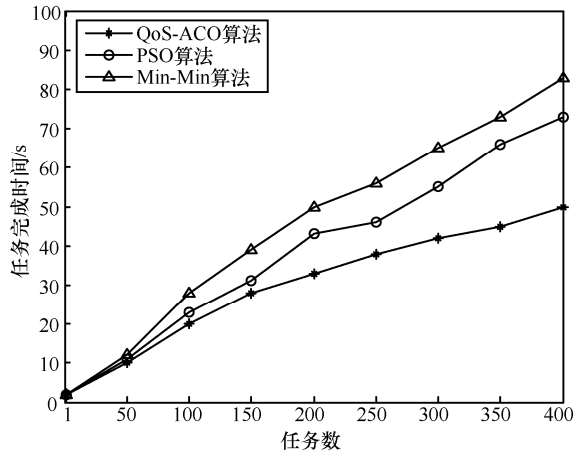


图 5 任务完成时间随任务数变化

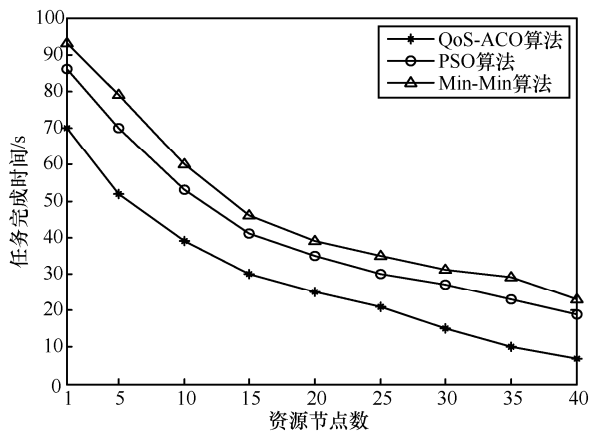


图 6 任务完成时间随资源节点数变化

从图 5 可以看出, 在蚁群寻径的初期, QoS-ACO 算法、PSO 算法和 Min-Min 算法在任务数量不多的情况下, 任务完成时间相差不多。但随着任务数量的增多, QoS-ACO 算法的任务完成时间明显小于 PSO 算法和 Min-Min 算法。从图 6 可以看出, 在资源节点数量增加的情况下, QoS-ACO 算法的执行调度任务时间始终小于 PSO 算法和 Min-Min 算法。这是由于在 QoS-ACO 算法中采用了预期完成时间来计算启发函数, 调度路径空间得到扩展并分布均匀。虽然扩展路径空间在初期会带来总时延问题, 但在之后的路径寻找过程中, 蚂蚁会倾向于信息素浓度高并且完成时间最小的路径。由此可以得出结论, QoS-ACO 算法可以提高任务的执行效率, 使蚂蚁

具有更强的任务调度能力。

5.2.2 安全性

安全性是任务调度的前提和保障。通过在 NS-2 上运行测量 QoS-ACO 算法、PSO 算法和 Min-Min 算法的任务安全性数据得到 trace 文件, 通过 Wireshark 统计分析工具对得到的 trace 文件进行统计和分析, 并利用 Matlab 对结果进行处理, 结果如图 7 和图 8 所示。

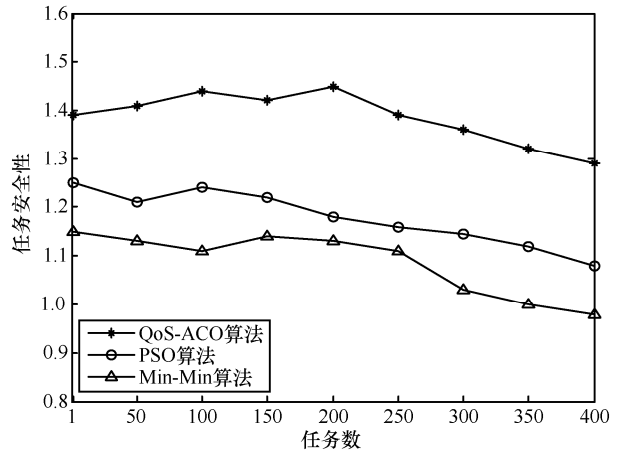


图 7 任务安全性随任务数变化

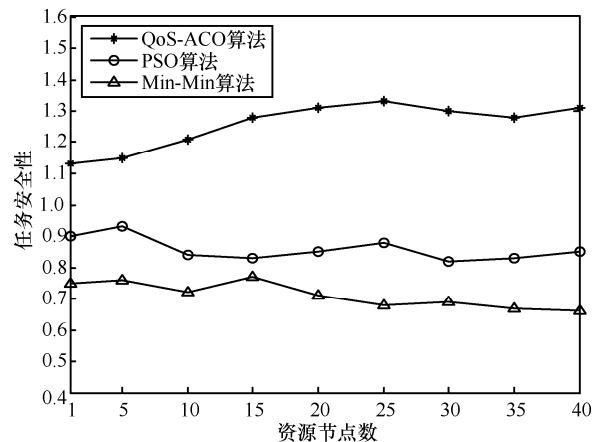


图 8 任务安全性随资源节点数变化

从图 7 和图 8 可以看出, 随着任务数量和资源节点数量的增加, QoS-ACO 算法的任务安全性值始终高于 PSO 算法和 Min-Min 算法。这是因为 QoS-ACO 算法中的蚁群信息素更新是通过 QoS 总效用评价函数, 而安全性评价函数为总效用评价函数的一部分, 为保证总效用评价函数最大, 就需要在任务调度过程中对其安全性进行考虑。由此可以得出结论, QoS-ACO 算法可以保障蚁群在任务调度过程中资源的安全性。

5.2.3 可靠性

可靠性反映整个系统完成所有任务调度的成功性。在 NS-2 上运行得到 QoS-ACO 算法、PSO 算法和 Min-Min 算法任务可靠性数据的 trace 文件。运用 Wireshark 统计分析工具对得到的 trace 文件进行统计和分析，并利用 Matlab 工具对结果进行处理，结果如图 9 和图 10 所示。

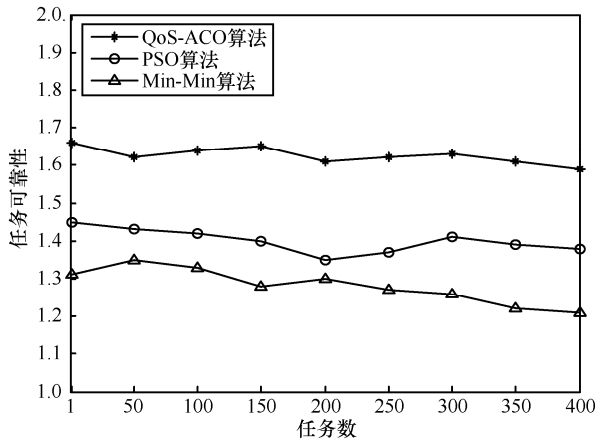


图 9 任务可靠性随任务数变化

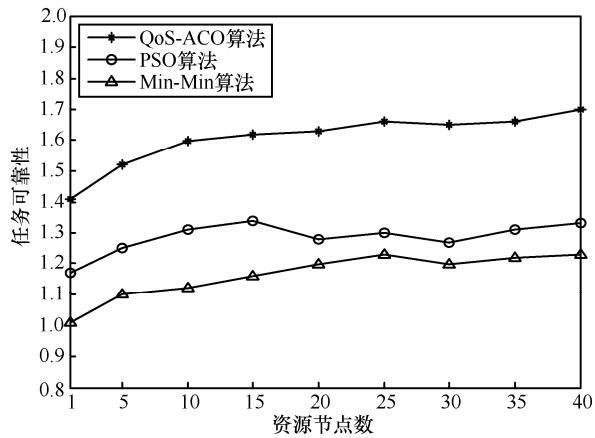


图 10 任务可靠性随资源节点数变化

从图 9 和图 10 可以看出，随着任务数量和资源节点数量的增加，QoS-ACO 算法的任务可靠性值始终高于 PSO 算法和 Min-Min 算法。这是因为 QoS-ACO 算法中的蚁群信息素更新是用 QoS 总效用评价函数，而可靠性评价函数为总效用评价函数的一部分，为保证总效用评价函数最大，就需要在任务调度过程中对其可靠性进行考虑。由此可以得出结论，QoS-ACO 算法可以提高任务调度完成的成功率，提升系统的任务调度性能。

5.2.4 总效用值

为提高用户对任务处理的满意度，需要保证服

务质量总效用评价价值最大，本文通过式(11)对任务调度算法的 QoS 总效用进行评价。在 NS-2 仿真平台中进行对比实验，并把每个任务依次提交至仿真系统中进行调度处理，实验结果如图 11 和图 12 所示。

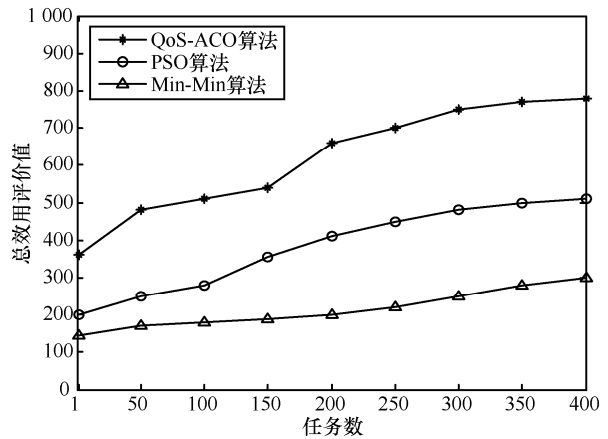


图 11 不同任务数下总效用值

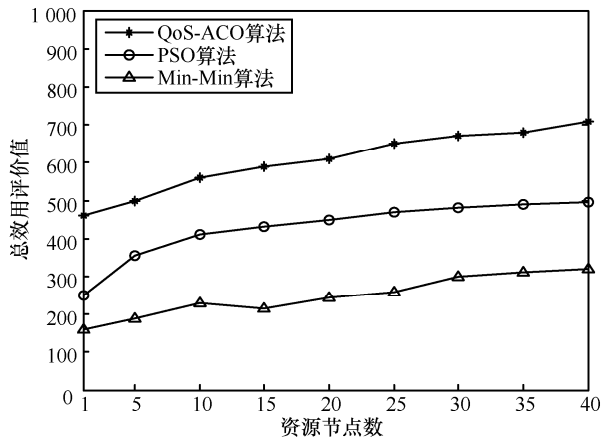


图 12 不同资源节点数下总效用值

从图 11 和图 12 可以看出，在任务数量增加和资源节点数量增加这 2 种情况下，QoS-ACO 算法所获得的总效用值都明显优于 PSO 算法和 Min-Min 算法。这主要是因为 QoS-ACO 算法考虑了用户对任务完成时间、安全性和可靠性 3 个服务质量方面的要求，并从这 3 个方面建立各自的约束条件，根据式(10)得到任务调度的 QoS 总效用评价函数，并使用该值对 QoS-ACO 算法中的蚁群信息素进行全局更新。由此可以得出结论，QoS-ACO 算法在保障系统调度任务完成的前提下，可以提高系统对用户的服务质量。

5.3 比较分析

目前，在 SWIM 任务调度方面的研究及成果不

表 3 部分任务调度算法任务调度性能比较

算法名称	任务完成时间/s			安全性 (安全等级)	可靠性(成功率)			总效用值 (满意度)
	最优值	平均值	最优值-平均值		最优值	平均值	最优值-平均值	
Min-Min 算法	0.35	0.52	0.17	low	0.89	0.76	0.13	68.75%
PSO 算法	0.21	0.29	0.08	medium	0.92	0.81	0.11	83.78%
本文 QoS-ACO 算法	0.15	0.19	0.04	high	0.93	0.85	0.08	89.85%

是很多, 针对 SWIM 任务调度算法的研究也很少。而在其他大型网络系统应用领域关于任务调度算法的研究已经取得了一定的成果, 例如 Min-Min 算法^[4]和启发式任务调度 PSO 算法^[18]等。因此, 将本文提出的 QoS-ACO 算法与上述算法进行性能比较, 结果如表 3 所示。

表 3 中性能比较结果数据中 3 组数值分别表示各自检测项目的最优值、平均值及两者差的绝对值, 一般用最优值和平均值之差的绝对值来总体衡量检测项目的性能, 其值越小则性能越好。从表 3 可以看出, 本文提出的 QoS-ACO 算法在完成时间、安全性和可靠性方面的性能指标都优于 Min-Min 算法和启发式任务调度 PSO 算法, 且算法总效用值最高。

6 结束语

服务质量是衡量 SWIM 任务调度策略好坏的重要指标, 本文以多 QoS 约束条件为立足点, 实现 SWIM 任务最优调度目标; 提出的基于多 QoS 约束条件 SWIM 任务调度蚁群优化算法, 综合考虑了 SWIM 中用户的 QoS 需求; 通过对蚁群算法的改进, 使用 SWIM 中业务调度 QoS 总效用评价函数来更新蚁群算法中的信息素, 保证在有效完成调度任务的同时, 最大化满足用户的 QoS 需求。仿真实验表明, 在多约束条件 QoS 任务调度模型中, QoS-ACO 算法在任务完成时间、安全性、可靠性和总效用值方面比 PSO 算法和 Min-Min 算法有一定的优势, 可以获得较好的任务调度结果, 能够解决 SWIM 资源共享问题, 满足用户对服务质量需求, 提高了航空运营质量和效率。但 QoS-ACO 算法在以下方面需要进一步研究和完善。

1) 由于蚁群算法普遍存在过早收敛的现象, 尤其是当任务数达到一定数量时, 总效用值会过早收敛。因此本文提出的 QoS-ACO 算法也存在不足, 在后续算法改进中, 会借鉴其他启发式算法优点, 如模拟退火算法和遗传算法等。根据这些算法

的不同特点进行综合应用, 优劣互补, 提高算法整体任务调度性能。

2) 本文提出的 QoS-ACO 算法, 只考虑了服务质量中的任务完成时间、安全性、可靠性和总效用值这几个方面, 还需要更全面地考虑 SWIM 中的其他 QoS 需求, 以及系统负载均衡等因素, 不断地改进和完善任务调度策略。

3) 蚁群算法中参数设置很重要, 如果设置不当会导致求解速度变慢, 影响结果质量。在下一步的 SWIM 任务调度研究工作中, 还将考虑任务的动态性, 动态调整算法中的相应参数。

4) 本文提出的 QoS-ACO 算法只是在模拟的 SWIM 环境下, 利用网络仿真工具进行仿真实验, 而没有在实际 SWIM 应用环境下测试。后续工作应在真实的 SWIM 环境下进行实验测试, 进一步验证所提 QoS-ACO 算法的有效性。

参考文献:

- [1] STEPHENS B. System-wide information management (SWIM) demonstration security architecture[C]//2006 IEEE/AIAA Digital Avionics Systems Conference. IEEE, 2006:1-12.
- [2] CRESCENZO D D, STRANO A, TRAUSMUTH G. SWIM: a next generation ATM information bus-the SWIM-SUIT prototype[C]//IEEE International Enterprise Distributed Object Computing Conference Workshops. IEEE Computer Society, 2010:41-46.
- [3] 赵婷. 基于 Diameter 协议的民航 SWIM 架构认证的研究[D]. 天津: 中国民航大学, 2014.
ZHAO T. Research on the authentication of SWIM architecture in civil aviation based on diameter protocol[D]. Tianjin: Civil Aviation University of China, 2014.
- [4] MAHESWARAN M, ALI S, SIEGAL H J, et al. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems[C]// Heterogeneous Computing Workshop. IEEE, 1999:30-44.
- [5] CHAUHAN S S, JOSHI R C. A weighted mean time Min-Min Max-Min selective scheduling strategy for independent tasks on grid[C]// Advance Computing Conference. IEEE, 2010:4-9.
- [6] WU M Y, SHU W, ZHANG H. Segmented Min-Min: a static mapping algorithm for meta-tasks on heterogeneous computing systems[C]// Heterogeneous Computing Workshop. DBLP, 2000:375-385.
- [7] 胡艳华, 唐新来. 基于改进遗传算法的云计算任务调度算法[J]. 计

- 计算机技术与发展, 2016, 26(10):137-141.
- HU Y H, TANG X L. A task scheduling algorithm based on improved genetic algorithm in cloud computing environment [J]. Computer Technology and Development, 2016, 26(10):137-141.
- [8] BUYYA R, ABRAMSON D, VENUGOPAL S. The grid economy[J]. Proceedings of the IEEE, 2005, 93(3):698-714.
- [9] CASTILLO C, ROUSKAS G N, HARFOUSH K. Online algorithms for advance resource reservations[J]. Journal of Parallel and Distributed Computing, 2011, 71(7):963-973.
- [10] 刘宴兵, 陈杰, 熊仕勇. 基于 QoS 相似度的网格任务调度算法[J]. 重庆邮电大学学报(自然科学版), 2009, 21(3):416-420.
- LIU Y B, CHEN J, XIONG S Y. Grid task scheduling algorithm based on QoS similarity[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2009, 21(3):416-420.
- [11] 邓见光, 赵跃龙, 袁华强. 一种多 QoS 目标约束的云计算任务调度策略[J]. 计算机应用研究, 2016, 33(8):2479-2482.
- DENG J G, ZHAO Y L, YUAN H Q. Multi-QoS objective constrained task scheduling strategy of cloud computing[J]. Application Research of Computers, 2016, 33(8):2479-2482.
- [12] 祁晓峰, 张兴明, 高彦钊. 基于离散粒子群优化的可重构系统任务调度算法[J]. 小型微型计算机系统, 2018, 39(3):556-561.
- QI X F, ZHANG X M, GAO Y Z. Discrete particle swarm optimization-based task scheduling algorithm in reconfigurable system[J]. Journal of Chinese Computer Systems, 2018, 39(3):556-561.
- [13] RANGLES M, LAMB D, TALEB-BENDIAB A. A comparative study into distributed load balancing algorithms for cloud computing[C]// International Conference on Advanced Information NETWORKING and Applications Workshops. IEEE Computer Society, 2010:551-556.
- [14] DORIGO M, GAMBARDILLA L M. Ant colony system: a cooperative learning approach to the traveling salesman problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1):53-66.
- [15] 张海燕, 吴凡, 王建新. 基于蚁群算法的 Hadoop 资源感知调度器研究[J]. 计算机工程与应用, 2014, 50(15):65-71.
- ZHANG H Y, WU F, WANG J X. Research of Hadoop resource-aware scheduler based on ant colony algorithm[J]. Computer Engineering and Applications, 2014, 50(15):65-71.
- [16] 华丹. 基于 QoS 的蚁群优化在网格资源调度中的研究[D]. 南京: 南京信息工程大学, 2011.
- HUA D. Research on grid resources scheduling based on QoS-Ant colony optimization algorithm[D]. Nanjing: Nanjing University of Information Science & Technology, 2011.
- [17] LU X, GU Z. A load-adaptive cloud resource scheduling model based on ant colony algorithm[C]// IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE, 2011:296-300.
- [18] 童钊, 陈洪剑, 陈明, 等. 一种云环境下基于混合型 BBO 的任务调度算法[J]. 计算机工程与科学, 2018, 40(5):765-772.
- TONG Z, CHEN H J, CHEN M, et al. A hybrid biogeography-based optimization algorithm for task scheduling in cloud computing[J]. Journal of Computer Engineering & Science, 2018, 40(5):765-772.

[作者简介]



李罡(1981-), 男, 满族, 吉林长春人, 天津大学博士生, 主要研究方向为网络与信息安全。



吴志军(1965-), 男, 河南固始人, 博士, 中国民航大学教授、博士生导师, 主要研究方向为网络空间安全、大数据信息安全和云计算安全等。