基于多群智能算法的云计算任务调度策略

陈海燕

(华东政法大学计算机科学与技术系 上海 201620)

摘 要 为提高云计算任务调度的服务质量(QoS),提出一种多群智能算法的云计算任务调度策略。首先利用全局搜索能力强的遗传算法快速找到云计算任务调度问题的较优解,然后将较优解转换成蚁群优化算法的初始信息素,最后通过蚂蚁间的信息交流和反馈找到云计算任务调度的全局最优解。以CloudSim为仿真平台进行了模拟实验,结果表明,与同类算法相比,多群智能算法不仅大幅提高了云计算任务调度效率,而且减少了处理请求任务的平均完成时间。

关键词 云计算,遗传算法,蚁群优化算法,任务调度

中图法分类号 TP393 **文献标识码** A

Task Scheduling in Cloud Computing Based on Swarm Intelligence Algorithm

CHEN Hai-yan

(Department of Computer Science and Technology, East China University of Political Science and Law, Shanghai 201620, China)

Abstract In order to improve the QoS of task scheduling of cloud computing, this paper proposed a cloud computing task scheduling model based on warm intelligent algorithm. Firstly, genetic algorithm which has global search ability is used to quickly find he optimal solutions of cloud computing task ssheduling problem, and then the optimal solutions are converted to the initial pheromone of ant colony optimization algorithm, finally, the global optimal solution of cloud computing task scheduling is obtained by information communication and feedback among ants. The simulated experiment was carried out on cloudsim platform. The experimental results show that compared with other models, the proposed model significantly improves the efficiency of cloud computing tasks scheduling and reduces the completion time.

Keywords Cloud computing, Genetic algorithm, Ant colony opt optimization algorithm, Task scheduling

云计算是并行计算、分布式计算和虚拟技术相结合的产物,是当前计算机行业的技术热点。云系统首先将计算机、存储设备等组合在一起,组成资源池,然后用户可以根据需要选择相应的资源,为用户动态地提供可靠的、服务质量(QoS)保证的计算服务环境。任务调度是云计算的核心技术之一,对云计算系统整体性能影响很大[1]。

针对云计算任务调度问题,有学者提出了Hadoop的调度算法,该算法按照任务优先级和提交时间进行调度,简单、易实现,但忽略了任务间的差异性,导致任务响应时间长[2]。随后有学者提出效益最优的云计算任务调度算法、基于代价驱动的任务调度算法[3]、基于信任驱动的任务调度算法[4],这些算法灵活,较好地满足了用户需求,但是它们都是针对某个特定目标,适用范围比较窄。大量研究表明,云计算任务调度不仅为用户提供可靠的、服务质量(QoS)保证,同时要保证任务分配的公平性,云计算多任务调度问题是计算机界公认的NP难题[5]。如果采用穷举搜索法对其进行求解,计算复杂度过高,随着群智能算法不断成熟,近年来有学者将粒子群优化算法、遗传算法、蚁群算法引入到云汁算任务调度问题中,取得了不错的应用效果[6-9]。但是由于云计算资源的异构性、动态性能及用户需求差异性,云任务调度是一个复杂问题,单一遗传算法、蚁群算法均存在各自的不足,而且基于组合优化理论的组合算法可以利用算法之间优势进行互补,从而找到问题的最优解[10]。

为提高云计算任务调度的服务质量,提出一种多群智能算法的云计算任务调度策略(GA-ACO)。首先利用全局搜索能力强的遗传算法(GA)快速找到云计算任务调度问题的较优解,然后将较优解转换成蚁群优化算法(ACO)的初始信息素,最后通过蚂蚁间的信息交流和反馈找到云计算任务调度的全局最优解。以CloudSim为仿真平台进行了模拟实验,结果表明,GA-ACO算法的云计算任务调度的服务质量(QoS)能够很好地满足用户的要求。

1 云计算任务调度目标函数

云计算任务调度指的是在一个特定的云环境中,根据一定任务调度策略,最大限度地满足用户的需求,使资源利用率达到最大化,且缩短任务调度的时间。按照QoS标准可将任务按完成时间、带宽、可靠性、费用分为4类,因此本文通过不同的QoS参数对任务调度算法性能进行评价。

本文受国家社会科学基金项目(06BFX051),上海高校选拔培养优秀青年教师科研专项基金(hzf05046)资助。

陈海燕(1978—),男,硕士,讲师,主要研究领域为网络安全、计算机犯罪、智能算法。

(1)时间效用函数。设任务提交到被分配处理资源的等待时间为 T_{wait}，任务执行时间为 T_{exec}，任务传输时间为 T_{trans}，任务总完成时间 T_F 定义为：

$$T_F = T_{wait} + T_{exec} + T_{trans} \tag{1}$$

设任务的期待完成时间为 T_{expt}，那么第 i 个任务在资源节点 j 上的时间效用函数定义如下：

$$time\text{-}util(i,j)=\begin{cases}1, & T_F < T_{expt} \\ e^{\lambda}(T_F < T_{expt})/T_{expt}, & \text{other}\end{cases} \tag{2}$$

式中，λ 为一常数。

(2)安全效用函数。设 S_e 表示实际能获得的安全度，S_i 表示任务期待的安全性要求，则安全效用函数为：

$$\sec\text{-}util(i,j)=\begin{cases}1, & S_e > S_i \\ S_e/S_i, & \text{other}\end{cases} \tag{3}$$

(3)费用效用函数。虚拟机按照单位对资源计费，任务消费的全部费用 $cost_{i,j}$ 为：

$$cost_{vm} = P_1 C_{cpu/num} + P_2 C_{men/MB} + P_3 C_{stor/MB} + P_4 C_{bw/Mbps} \tag{4}$$

式中，P_1 表示资源数量，C 参数表示单位资源价格。

设 $cost_{user}$ 为用户期待的费用，则公平评判函数为：

$$cost\text{-}util(i,j)=\begin{cases}1, cost(i,j) < cost_{user} \\ e^{\lambda}(cost(i,j) - cost_{user})/cost_{user}, \text{other}\end{cases} \tag{5}$$

(4)可靠性效用函数。长时间运行的任务可能因为资源的失效而运行失败，任务的重调度将导致重复的资源消耗和系统性能降低，导致用户任务的服务质量降低。设单个任务在某个资源节点上获得的可靠性为 P，用户期待的可靠性为 P_{succ}，可靠性效用函数为：

$$reli\text{-}util(i,j)=\begin{cases}1, & P < P_{succ} \\ e^{\lambda}(P - P_{succ})/P_{succ}, & \text{other}\end{cases} \tag{6}$$

综上可知，单个任务在某个资源节点上执行的效用函数为：

$$J_{zh} = w_1 \times time\text{-}util(i,j) + w_2 \times \sec\text{-}util(i,j) + w_3 \times cost\text{-}util(i,j) + w_4 \times reli\text{-}util(i,j) \tag{7}$$

式中，w_i 表示权重，本文设置 $w_1=0.4, w_2=0.2, w_3=0.2, w_4=0.2$。

2 多群智能算法的任务调度

2.1 GA 与 ACO 融合思想

GA 具有较强的全局搜索能力，但在进化后期，由于不能利用系统反馈信息，收敛速度变慢，求解效率低，易获得局部最优解；ACO 具有反馈机制，但进化初期，信息匮乏，搜索速度慢，求解效率低，GA、ACO 求解曲线如图 1 所示。

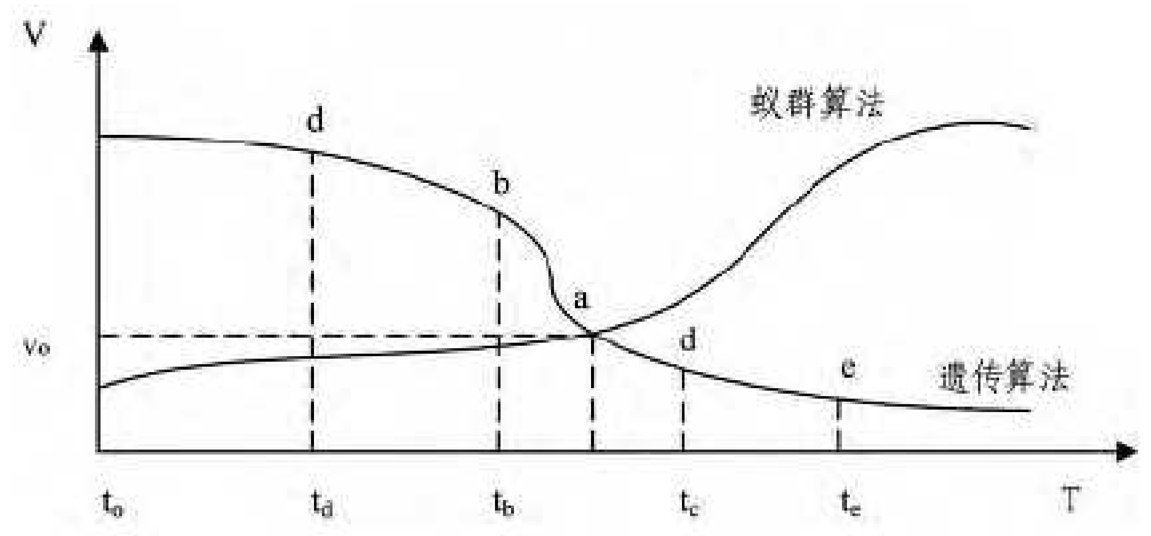

图 1 GA 与 ACO 的求解曲线

对图 1 进行分析可知，在 $t_0 \sim t_a$ 时间段，GA 搜索速度快，在 t_a 之后，解效率急剧下降；在 $t_0 \sim t_a$ 时间段，ACO 搜索速度相当慢，在 t_a 之后，随着信息素不断积累，速度大幅提高，为此，将 t_a 作为结合点，在 $t_0 \sim t_a$ 阶段，首先采用 GA 对云计算任务调度问题进行求解，得到问题较优的解，然后将问题较优的解转换成 ACO 的初始信息素，蚂蚁信息交流和反馈对较优的解进一步优化，最后找到云计算任务调度问题的最优解。

2.2 GA 设计

.染色体的编码

在采用 GA 对云计算任务调度问题进行求解的过程中，首先必须在个体与问题的解之间建立映射关系，即对个体编码。根据云计算资源调度的特点，采用间接编码方式，即对每一个子任务占用的资源进行编码，即 $d_1, d_2, \cdots, d_n$，其中 d_i 表示第 i 个资源，编码长度取决于任务的个数。一个染色体编码形式为：(4,6,2,9,8,3,1,10)，表示 4 个任务与资源的对应关系，见表 1。

表 1 个体的编码方式

子任务号	1	2	3	4	5	6	7	8
资源号	4	6	2	9	8	3	1	10

.种群初始化

初始化种群对云计算任务调度问题求解相当关键，传统 GA 采用随机方式产生初始种群，这样容易使个体在问题解的空间中分布极不均匀，产生聚集现象，导致后继进化过程种群单一化，陷入局部最优解的概率大，为此，本文采用均匀设计产生初始种群，使个体在解的空间种群分布比较合理，提高种群的多样性。

.适应度函数

GA 与云计算任务调度问题接口为适应度函数，因此，需要将适应度函数与云计算任务调度目标结合起来，因此第 i 个染色体的适应度函数定义如下：

$$f(i) = \sum_{j=1}^{n} J_{zh}(j) \tag{8}$$

式中，$J_{zh}(j)$ 表示第 j 个任务完成的效用函数。

.遗传操作

遗传操作主要包括选择、交叉和变异操作，其中选择操作采用部分最优个体进入一下代，其余个体采用轮盘赌方式进入下一代方式；交叉(P_c)和变异(P_m)采用自适应方式进行：

$$p_c=\begin{cases}k_1(f_0 - f')/(f_0 - {}^{*}f), & f' \geq {}^{*}f \\ k_3, & f' < {}^{*}f\end{cases} \tag{9}$$

$$p_m=\begin{cases}k_2(f_0 - f)/(f_0 - {}^{*}f), & f \geq {}^{*}f \\ k_4, & f < {}^{*}f\end{cases} \tag{10}$$

式中，$k_1=0.5, k_2=0.05, k_3=1, k_4=0.1$；$f_0$ 为种群的最大适应值，f' 为交叉的两个个体中较大个体的适应值，${}^{*}f$ 为种群的平均适应值，f 为变异个体的适应值。

2.3 ACO 设计

(1)计算信息素初值

将 GA 获得的云计算任务调度问题较优解转换为 ACO

初始信息素，即：

$$\tau_{ij}(\alpha)=f(i) \tag{11}$$

(2)节点选择规则

在 t 时刻，根据信息素浓度选择下一个云计算资源节点，具体方式为：

$$p_j^k(t)=\frac{[\tau_j(t)]^\alpha[\eta_{k,j}(t)]^\beta}{\sum_{j=0}^{n-1}[\tau_j(t)]^\alpha[\eta_{i,j}(t)]^\beta} \tag{12}$$

(3)信息素更新规则

$$\tau_j^{new}=\rho\times\tau_j+\Delta\tau_j \tag{13}$$

式中，ρ 表示信息素残留因子，其取值为 $0\sim1$ 之间的随机数。ρ 的变化方式为：

$$\rho=0.1\times\frac{\log(t)}{e^{t_{max}}} \tag{14}$$

式中，t 为当前代数，t_{max} 为最大代数。

2.4 群智能算法的云计算任务调度步骤

(1)用户提交任务，根据任务设置 GA 的种群大小，选择交叉、变异概率，最大进化代数。

(2)根据云计算任务调度问题对种群进行初始化。

(3)对个体进行解码，计算其适应度值。

(4)根据适应度值对个体进行排序，选择部分较优解进入下一代，然后按轮盘赌方式选择其它个体进行一下代。

(5)根据 P_c、P_m 对个体进行交叉和变异操作，然后比较新个体与父个体之间的优劣，选择较优者进行下一代。

(6)若达到最大进化代数，则停止寻优，否则跳转步骤(3)继续进化。

(7)将 GA 找到的云计算任务调度方案转换成 ACO 的信息素初始值。

(8)将每只蚂蚁随机分布于云计算资源节点上。

(9)根据各资源节点的信息索浓度选择下一个任务执行的资源节点。

(10)当每一只蚂蚁均找到一个云计算任务调度方案后，对资源节点信息激素进行更新。

(11)当算法达到结束条件，输出云计算任务调度方案最优解，否则，跳转步骤(3)继续进行寻优。

具体流程如图 2 所示。

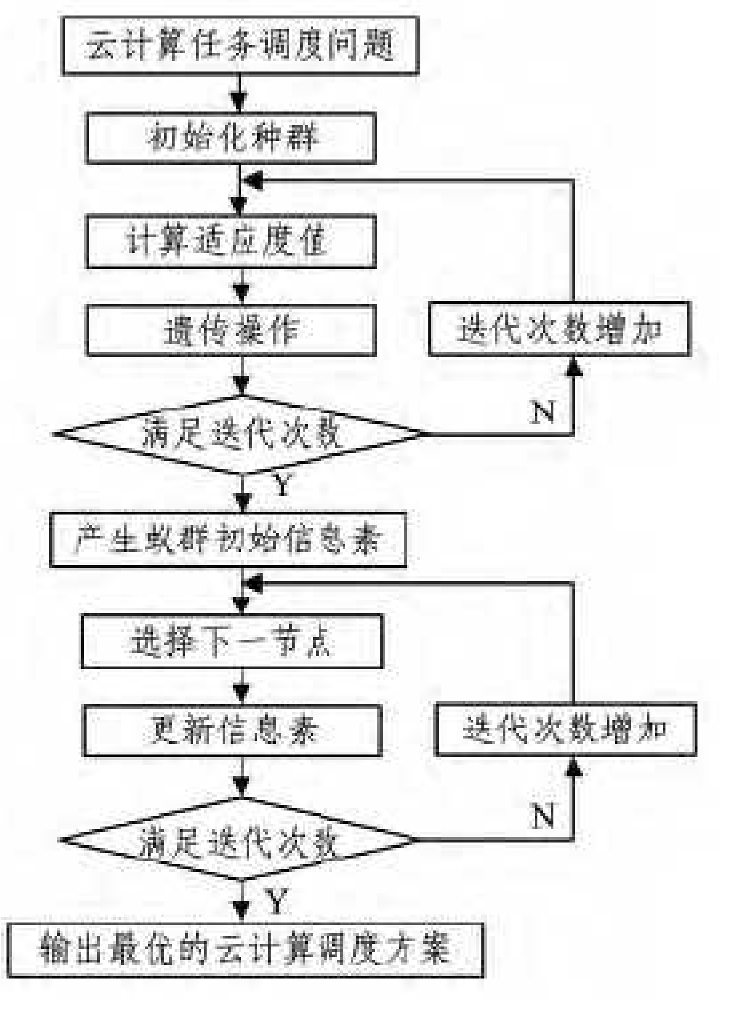

图 2　多群智能算法的任务调度求解流程

3 仿真实验

3.1 数据及参数设置

在云计算仿真平台 CloudSim 进行模拟，任务数量分别为：20、40、60、80、120、140、160、180、200，其参数见表 2；云计算系统资源节点为 10 数量，资源节点的参数见表 3。为了使 GA-ACO 仿真结果更具说服力，在相同实验条件下，采用 GA、ACO 算法进行对比实验。GA-ACO 的参数为：GA 个体数为 20；$P_c=0.9$，$P_m=0.1$；ACO 的蚁群为 10，$\alpha=1$，$\beta=2$，最大迭代次数为 1000。

表 2　任务参数列表

参数名称	参数范围
任务长度	1000～10000 百万条指令
用户期望时间	10～100 秒
单个任务的预算	0.01～0.09
用户对安全性的要求	1(低)、2(中)、3(高)、4(最高)

表 3　资源参数列表

资源编号	安全性	资源单价	可靠性	处理能力
0	1	0.03	0.04	200
1	2	0.08	0.03	400
2	2	0.04	0.05	100
3	3	0.05	0.02	50
4	1	0.07	0.01	100
5	2	0.09	0.07	200
6	1	0.06	0.06	400
7	2	0.02	0.08	50
8	4	0.01	0.09	100
9	2	0.02	0.05	200

3.2 结果与分析

随着任务数量的变化，GA、ACO、GA-ACO 的总效用变化曲线如图 3 所示。从图 3 可知，GA-ACO 的总效用要优于 GA、ACO 获得的云计算任务调度方案，对比结果表明，本文将 GA 和 ACO 进行融合，较好利用了两者的优点，克服了各自的不足，可以获得更优的云计算任务调度方案。

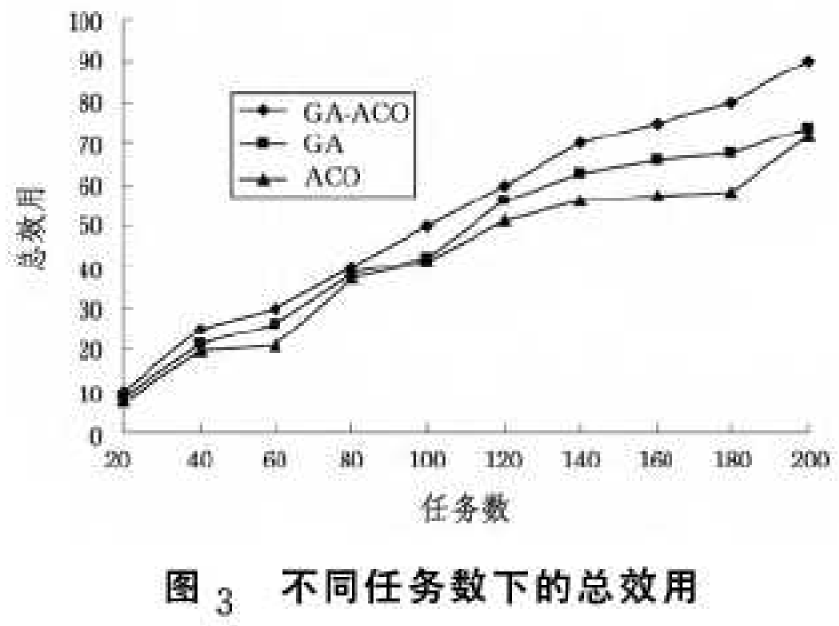

图 3　不同任务数下的总效用

为了全面对 GA、ACO、GA-ACO 的性能进行比较，采用费用、时间跨度、总延时作为评价指标，对它们的性能进行综合评价，具体结果如图 4—图 6 所示。

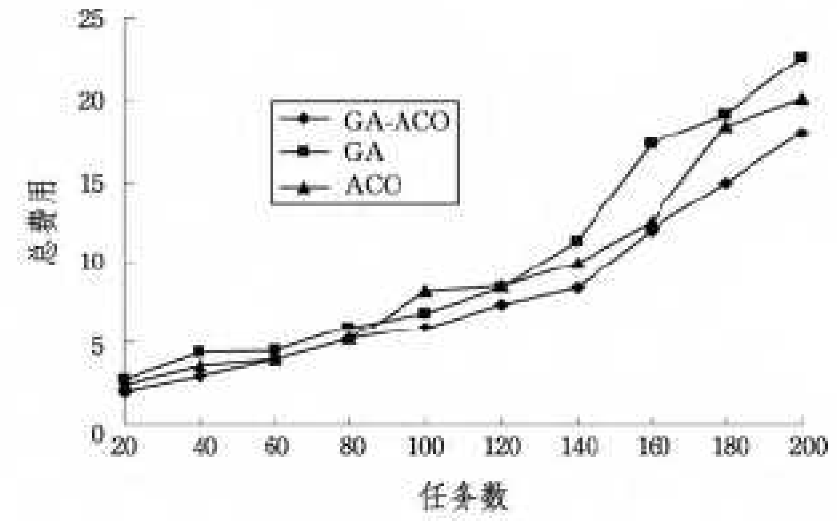

图 4　不同任务数量下的总费用

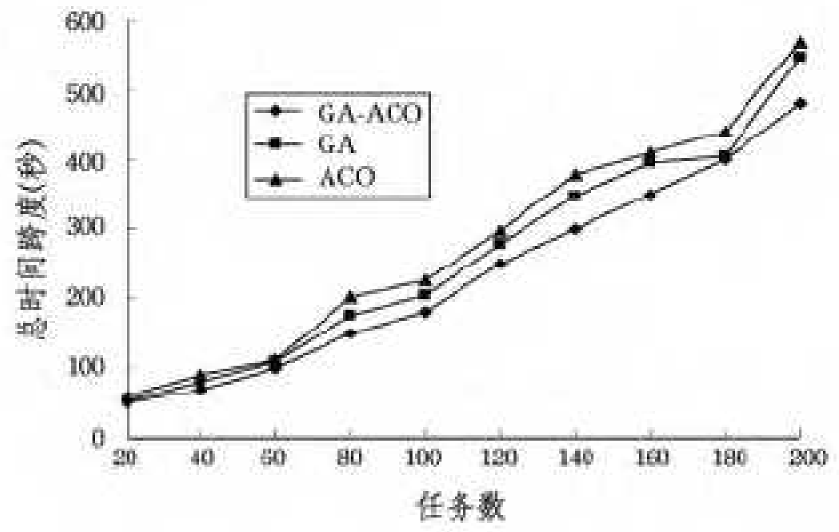

图 5　不同任务数量下的总时间跨度

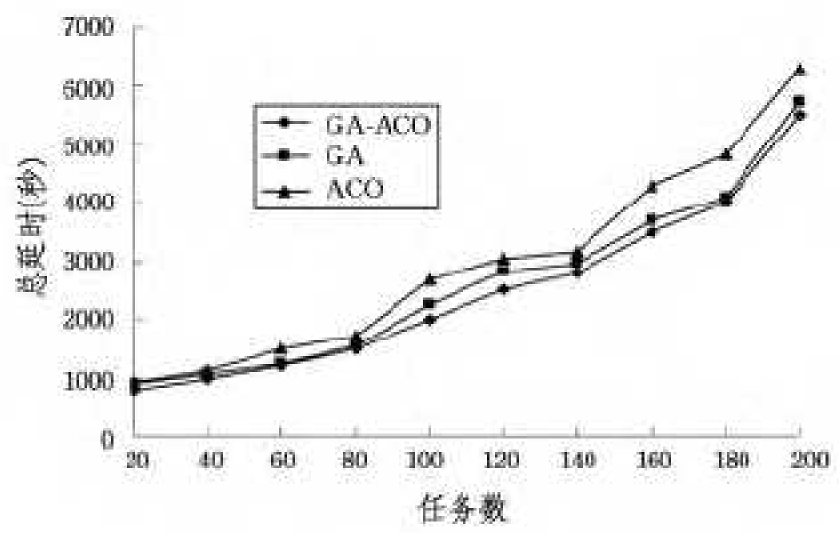

图 6　不同任务数下的总延时

从图 4—图 6 可以看出,单独使用 GA 或 ACO 进行云计算系统的任务调度问题求解,得到调度方案的任务执行总时间跨度大,用户等待的时间长,而且总费用相对较高,算法收敛效果不好。而 GA-ACO 集成了 GA 和 ACO 的优点,在寻优前期采用 GA 对云计算任务调度问题进行求解,可以较快地找到较优解,然后采用 ACO 对 GA 获得的解进行精细寻优,加快了收敛速度,获得了更加令人满意的云计算任务调度方案。

结束语　云计算具有规模大、可动态伸缩等特点,任务调度十分复杂,针对当前云计算任务调度算法存在的不足,基于组合优化理论,提出一种多群智能算法的云计算任务调度策略,仿真实验验证了 GA-ACO 在云计算任务调度求解中的有效性和正确性。

参 考 文 献

[1] 张建勋,古志民,郑超.云计算研究进展综述[J].计算机应用研究,2010,27(2):429-433

[2] 陈全,邓倩妮.云计算及其关键技术[J].计算机应用,2009,29(9):562-2568

[3] Iosup A,Ostermann S,Yigitbasi M N. Performance Analysis of cloud computing services for many-tasks scientific computing [J]. IEEE Trans. on Parallel and Distributed System,2011,22(6):931-945

[4] Jeffrey D,Sanjay G. Map/Reduce:simplified data processing on large clusters[J]. Communications of the ACM,2008,51(1):107-113

[5] 左利云,左利锋.云计算中基于预先分类的调度优化算法[J].计算机工程与设计,2012,33 (4):1357-1361

[6] Rochwerger B,Breitgand D,Levy E,et al. The reservoir model and architecture for open federated cloud computing [J]. IBM Journal of Research and Development,2009,53(4):1-17

[7] 张春艳,刘清林,孟珂.基于蚁群优化算法的云计算任务分配[J].计算机应用,2012,32(5):1418-1420

[8] 李建锋,彭舰.云计算环境下基于改进遗传算法任务调度算法[J].计算机应用,2011,31(1):184-186

[9] 刘万军,张孟华,郭文越.基于 MPSO 算法的云计算资源调度策略[J].计算机工程,2011,37(11):43-48

[10] 申丽君,刘丽,陆锐.基于改进免疫进化算法的云计算任务调度[J].计算机工程,2012,38(9):208-210

(上接第 66 页)

[4] Zhang Li-chen. MDA Approach for Non-functional Characteristics of Cyber Physical Systems Based on Aspect-Oriented Method[C]// Communications in Computer and Information Science. Springer,2011,215:323-328

[5] 黎作鹏,张天驰,张菁.信息物理融合系统(CPS)研究综述[J].计算机科学,2011(9):25-31

[6] Yun H,Liang Wu-po,Rahmaniheris M,et al. A Reduced Complexity Design Pattern for Distributed Hierarchical Command and Control System[C]// Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. 2010:42-49

[7] Parolini L,Tolia N,Sinopoli B,et al. A Cyber-Physical Systems Approach to Energy Management in Data Centers[C]// ICCPS'10. Stockholm,Sweden,2010:168-177

[8] Tan Ying, Vuran M C, Goddard S, et al. A Concept Lattice-based Event Model for Cyber-Physical Systems[C]// Proceedings of the 1 st ACM/IEEE International Conference of Cyber Physical Systems. 2010:50-60

[9] Chun I,Park J,Kim W,et al. Autonomic Computing Technologies for Cyber-Physical Systems[C]// Proceedings of the International Conference on Advanced Communication Technology. 2010:1009-1014

[10] Woo H, Yi Jiang-liang, Browne J C. Design and Development Methodology for Resilient Cyber-Physical Systems[C]// Proceedings of the 28 th International Conference on Distributed Computing Systems Workshops. 2008:525-528

[11] Vicaire P A, Hoque E, Xie Zhi-heng, et al. Bundle: A Group Based Programming Abstraction for Cyber Physical[C]// Proceedings of the 1st ACM/IEEE International Conference of Cyber-Physical Systems. 2010:32-41

[12] Ahmadi H,Abdelzaher T F,Gupta I. Congestion control for spatiotemporal data in cyber-physical systems[C]// Proceedings of the 1st International Conference on Cyber-physical Systeam(ICCPS'10). New York,NY,USA:ACM,2010:89-98

[13] Tan Ying, Vuran M C, Goddard S, et al. A Concept Lattice-based Event Model for Cyber-Physical Systems[C]// Proceedings of the 1st ACM/IEEE International Conference on Cyber Physical Systems. 2010:50-60