

# 基于云计算的流媒体任务调度算法\*

杨戈<sup>1,2</sup>, 吴俊言<sup>1</sup>

(1. 北京师范大学珠海分校 智能多媒体技术重点实验室, 广东 珠海 519087;

2. 北京大学深圳研究生院 深圳物联网智能感知技术工程实验室, 广东 深圳 518055)

**摘要:** 针对当前流媒体的大量视频资源从而带来的云计算的负载均衡和任务分配问题, 在 Cloudsim 云环境下实现了任务调度的 GAAC 算法(Greedy And Ant Colony Algorithm, GAAC)。GAAC 算法具有迭代学习机制、局部最优和负载均衡的特点。并在 Cloudsim 的环境下, 完成了对 GAAC 算法、轮转算法(Round Roll Algorithm, RR)、贪心算法和蚁群算法的仿真比较。实验验证, GAAC 算法从总体上而言, 任务调度所用的时间明显低于贪心算法和传统的轮转算法和蚁群算法, 即其任务执行的时间更短, 效率更高。

**关键词:** 云计算; 任务调度; 贪心算法

中图分类号: TN949.2

文献标识码: A

DOI: 10.16157/j.issn.0258-7998.200770

中文引用格式: 杨戈, 吴俊言. 基于云计算的流媒体任务调度算法[J]. 电子技术应用, 2021, 47(8): 97-100, 105.

英文引用格式: Yang Ge, Wu Junyan. Task scheduling algorithm based on cloud computing for streaming media[J]. Application of Electronic Technique, 2021, 47(8): 97-100, 105.

## Task scheduling algorithm based on cloud computing for streaming media

Yang Ge<sup>1,2</sup>, Wu Junyan<sup>1</sup>

(1. Key Laboratory of Intelligent Multimedia Technology, Beijing Normal University(Zhuhai Campus), Zhuhai 519087, China;

2. Engineering Lab on Intelligent Perception for Internet of Things(ELIP), Shenzhen Graduate School, Peking University, Shenzhen 518055, China)

**Abstract:** Aiming at the problem of cloud computing load balancing and task allocation brought about by a large number of video resources in the current streaming media, the task scheduling GAAC algorithm(Greedy And Ant Colony Algorithm, GAAC) is implemented in the Cloudsim cloud environment. GAAC algorithm has the characteristics of iterative learning mechanism, local optimization and load balancing. In the context of cloudsim, simulations of GAAC algorithm, Round Roll Algorithm(RR), greedy algorithm and ant colony algorithm were completed. The experimental verification shows that GAAC algorithm is generally lower in the time spent on task scheduling than greedy algorithm, traditional rotation algorithm and ant colony algorithm.

**Key words:** Cloud computing; task scheduling; Greedy algorithm

### 0 引言

随着计算机时代的发展, 用户的基数正在不断扩大, 而对应的在线视频的量级也正逐步扩展, 为解决点对点的在线视频的服务器的速度和带宽问题, 以及大量的视频资源带来服务器计算负载问题, 增加其负载而带来了“云计算”<sup>[1]</sup>。

云计算分为 3 层, 分别是 IaaS(基础设施即服务)、Paas(平台即服务)和 SaaS(软件即服务)<sup>[2]</sup>。储存资源管理是计算机资源管理的一部分, 侧重于计算机的节点的高效性和节点的整体负载均衡。无论是一般的云计算, 还是快速发展的移动云计算, 云增效模式是最常见的云计

算模式<sup>[3]</sup>。因而在云计算方面, 最主要研究的是计算机资源、负载均衡的实现和任务调度的分配等方面。在任务调度方面, 文献[4]提出了一种面向多目标的两阶段任务调度算法, 具有让任务匹配最小时间资源的偏好, 重调度阶段, 实现负载均衡; 文献[5]提出了针对 P2P(对等网络, 即对等计算机网络)结构上的用数据副本来进行管理, 从而提高数据访问的效率和系统容错功能。文献[6]中提出了一种基于任务调度的模板策略, 通过任务集合求出任务量模版, 并依据模板对调度算法进行任务调度的 TTS(基于模板的任务调度策略)策略。该算法从全局的角度计算出调度模板, 有目标地实现了调度同

\* 基金项目: 广东高校省级重大科研项目(2018KTSCX288, 2019KZDXM015, 2020ZDZX3058); 广东省学科建设专项资金(2013WYXM0122); 北京师范大学珠海分校科研项目(201762005); 北京师范大学珠海分校 2019 校级“质量工程”项目(201932)

时充分考虑了通信开销。

而对于计算机资源管理方面,文献[7]研究了虚拟云桌面上的动态调度应用,文献[8]中提出了一种关于网络感知的计算资源下的对虚拟机的放置算法。

本文在针对于云计算任务调度的基础上提了一种贪心蚁群算法(Greedy And Ant Colony Algorithm, GAAC),它避免了贪心算法在任务数较大情况下的搜索能力不足及局部搜索能力低下,以及蚁群算法在初始阶段的搜索能力低下的缺点,同时充分考虑了在全局情况下的云计算的流媒体任务调度。

### 1 云计算资源调度模型

云计算中的资源调度可通过以下的模型表示:

$$\min \sum_i f_i(y_i) = \min \sum_{i,j} E(i,j) + \min \sum_{i,j} T(i,j) + \min \sum_{i,j} C(i,j) \quad (1)$$

式中,  $f_i(y_i)$  表示效率函数。为了能够达到云计算的资源调度的优化要求,要求云计算资源调度所消耗资源的为最小值即  $\min \sum_i f_i(y_i)$ , 其花费、时间和能量三方面达到最小值。 $C(i,j)$  表示任务  $i$  在资源  $j$  的花费,  $E(i,j)$  表示任务  $i$  在资源  $j$  的能量消耗,  $T(i,j)$  表示任务  $i$  在资源  $j$  的时间消耗。

$$\sum_j x_{ij} = y_i \quad (2)$$

$$x_{ij} \geq 0 \quad (3)$$

式中,  $y_i$  表示完成任务  $i$  的所需要的资源总和,  $\sum_j x_{ij} = y_i$ ,

其中  $\sum_j x_{ij}$  表示的是任务  $i$  和虚拟机  $j$  之间的关系。本文主要从时间消耗最小上解决云计算的效率优化问题。

## 2 GAAC 算法

GAAC 算法是在现今的任务调度贪心算法的后期融入了蚁群算法的特点。GAAC 算法同时具有贪心算法任务量较少时,负载均衡,能实现较好的局部优化和蚁群算法任务量较多时具有的学习机制和迭代优化的特点,即 GAAC 算法同时具备了贪心算法和蚁群算法的原理特点。

### 2.1 GAAC 算法实现

对于其中的贪心算法,学者们对任务调度及贪心算法进行了大量的研究<sup>[9-12]</sup>。而在文献[6]中,贪心法则为:按物品的价值大小排序,选择剩下的容量最大的背包,装最大件物品,直到所有的背包都装满物品为止。将该方法类比到云计算的任务调度问题下,提出贪心算法下的任务调度优化方法。

而对应于云计算调度,则是进行任务和虚拟机排序,按照虚拟机执行任务的能力和速度按大到小进行排序,然后再按照任务的大小同时进行排序,检索最大的

任务在哪个虚拟机运行得最快,然后完成分配,再进行下一个任务的分配,依次类推直到任务分发完毕。

而 GAAC 算法在任务量较大的后期则是具备仿生原理,它依据蚂蚁群寻找食物的原则而模拟而成的一种最短路径的智能算法<sup>[13]</sup>,而其具体步骤如下:

(1)初始化,首先设定相关参数;

(2)将  $m$  个蚂蚁随机放在各个食物上,每个食物至多分布一个蚂蚁,并将  $m$  修改禁忌表  $J_k$ ;

(3)所有蚂蚁根据概率转换公式和选择下一食物,并将该元素(食物)移动到该蚂蚁个体的禁忌表中;

(4)所有蚂蚁遍历完  $n$  个食物后在所经过的路径上根据信息更新公式更新所有信息素,并记录本次迭代过程重点最优路径和最优路径长度;

(5)清空禁忌列表  $J_k$ ,重复步骤(3)和(4),直到每一个蚂蚁都完成  $N_{\max}$  次遍历所有食物,最后输出的路径为最优路径。

而对应在任务调度中则是将食物替换成云计算中的各个节点的虚拟机,而其蚂蚁获得的路径则是对应任务分配给虚拟机的分法,而蚂蚁获得的路径的长度,则为分配好后所花费的总时间。算法通过其中的学习机制,不断迭代从而找到任务调度中的最短路径,进而实现路径的最优化。

### 2.2 GAAC 算法流程

由于蚁群算法的受信息素引导,需要较多次数的学习和迭代,才能形成较好的优化,对应在任务调度算法上则需要较多的任务量。而在起初阶段,由于任务数(对应蚁群中的食物点)较少,正反馈的尚不健全,无法形成良好的学习机制,从而无法得到较优解。相反,贪心算法由于不具备学习机制,在前期所耗时较少,但在后期任务数较大时,由于学习机制的缺失,优化效果较差,而 GAAC 算法正是将这两种算法融合形成的新算法。

设贪心算法所完成的时间为  $\text{tasktime}(t)_a$ , 蚁群算法所完成的时间为  $\text{tasktime}(t)_b$ 。通过比较可以得出两种算法在云任务的实现时间,则判断  $\text{tasktime}(t)_a = \text{tasktime}(t)_b = \gamma$  所对应的任务数值,求得大约为:

(1)当  $\text{tasktime}(t)_a < \text{tasktime}(t)_b$  时,即采用贪心算法:

①将任务按由大到小的顺序降序排列(MI),并将虚拟机按按小到大升序排列(MIPS),所消耗的时间为  $T$ ,它们之间满足公式:  $T = MI/MIPS$ 。

②将最后一列对应的虚拟机分配给矩阵中行号为 0 的任务。

(2)  $\text{tasktime}(t)_a > \text{tasktime}(t)_b$ , 此时即采用蚁群算法,其路径概率的选择和信息素更新选择如下:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & j \notin J^k \\ 0 & j \in J^k \end{cases} \quad (4)$$

式中,  $p_{ij}^k(t)$  表示在  $t$  时刻, 第  $k$  只蚂蚁从节点  $i$  走到节点  $j$  的状态转移,  $\alpha$  和  $\beta$  分别表示启发式因子。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

式中,  $\rho$  表示信息素挥发系数,  $\tau_{ij}^k(t)$  表示第  $k$  只蚂蚁在路径  $i \rightarrow j$  上的信息素增量。算法流程图如图 1 所示。

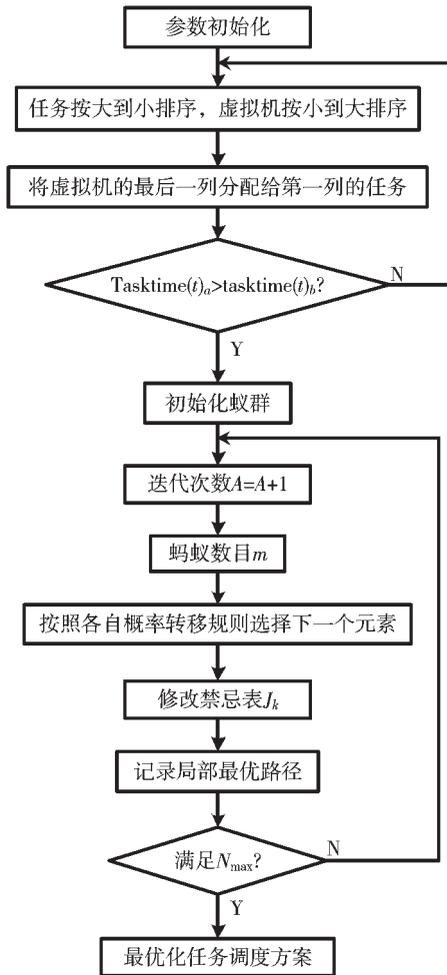


图 1 GAAC 算法流程图

### 3 仿真实验

本文运用了云计算的模拟平台 Cloudsim, 通过 Cloudsim 实现云环境的模拟配置, 达到异构的云环境。

#### 3.1 软硬件

实验使用的物理机和软硬件配置如表 1 所示。

表 1 硬件及软件配置

参数	参数值
处理器	2.50 GHz Intel® Core i7-6500M CPU
内存/GB	8.00
操作系统	Windows 10
开发平台	Eclipse
仿真平台	Cloudsim-3.0.3

#### 3.2 仿真结果

通过修改 Datacenterbroker 类加入了自定义的多任务状态的贪心算法和 GAAC 算法, 并在里面写入了传统的轮转算法进行对比, 进行仿真对比分析, 仿真实验参数如表 2 所示。

表 2 仿真实验参数

参数	参数值
虚拟机个数	5
虚拟机平台属性	异构
任务个数	500, 1 000, 1 500, 2 000, 2 500
GAAC( $\alpha, \beta$ )	1.0, 1.0
GAAC(ant)个数	5
$N_{max}$	50

图 2 显示了异构平台下, 虚拟机数为 5 个时, 任务个数由 0 个增加到 2 500 个时, GAAC 算法与贪心算法、轮转算法和蚁群算法的任务集合完成时间的对比。由于轮转算法的优化效果较差, 因此其耗时明显多于贪心算法和 GAAC 算法。从图中可看出在任务个数大于 500 时 4 种算法的差异, 贪心算法的用时和 GAAC 算法重合, 原因是在  $\gamma$  小于 1 000 时, GAAC 算法采用的是贪心算法的原理, 但是 GAAC 算法的用时明显少于蚁群算法和轮转算法。但随着任务个数的增加, 贪心算法和 GAAC 算法的用时明显开始减少, 而至任务数约为 1 000 时, GAAC 算法的用时开始明显比贪心算法少。其原因是  $\gamma$  大于 1 000 时, GAAC 算法采用的是蚁群算法的原理, 在迭代学习次数较大, 训练集较大, 训练时间较长的情况下进行, 由于迭代学习的原因, 此时的任务调度实现了更好的优化, 因此相对应的用时更少。而蚁群算法和 GAAC 算法在约为 1 000 个任务之后时间几乎一致, 但由于任务具有随机性设置的原因, 示例图中的时间上仍有些许的不吻合。但 GAAC 算法总体的任务调度时间少于贪心算法和轮转算法, 且在前期也优于蚁群算法。即 GAAC 从总体的任务优化效果而言, 相对优于轮转算法、贪心算法和蚁群算法。

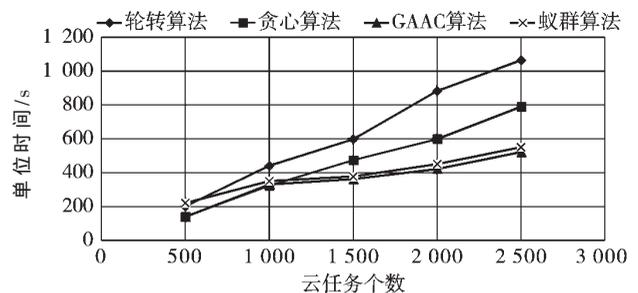


图 2 任务调度算法对比 1

针对表 3 仿真实验参数, 图 3 显示了异构平台下, 虚拟机数为 50 个, 任务个数由 0 个增加到 2 500 个时, GAAC 算法与贪心算法、轮转算法和蚁群算法的任务集

表 3 仿真实验参数

参数	参数值
虚拟机个数	50
虚拟机平台属性	异构
任务个数	500, 1 000, 1 500, 2 000, 2 500
GAAC( $\alpha, \beta$ )	1.0, 1.0
GAAC(ant)个数	5
$N_{max}$	50

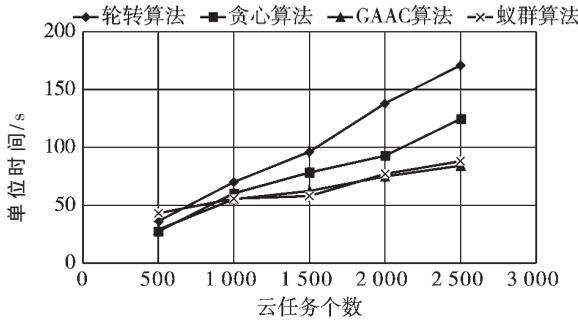


图 3 任务调度算法对比 2

合完成时间的对比。此时可以看出,  $\gamma$  所对应的值约为 800, 在  $\gamma$  小于 800 时, GAAC 算法采用的是贪心算法的原理, 相对于蚁群算法, 其优化效果大约为 40%, 而相对于轮转算法其优化效果大约为 22%, 这是由于 GAAC 算法前期采用的是现今的贪心算法, 其在任务数较少时, 能有较好的局部优化效果, 而蚁群由于迭代和学习机制尚未健全, 因此其时间优化相对较差于轮转算法和贪心算法; 而在  $\gamma$  大于 800 时, 由于具有蚁群算法的学习和迭代机制的原因, GAAC 算法和蚁群算法接近吻合, 但由于每次任务的随机性, 并没有重合在一起。GAAC 算法由于在现今的贪心算法中融入了采用了蚁群算法的原理, 在  $\gamma$  大于 800 时, 相对于现今的贪心算法优化效果大概为 33%。可以看出 GAAC 算法在总体上相对于贪心算法和轮转算法的优化效果更好。

针对表 4 仿真实验参数, 图 4 所示为虚拟机个数为 100 个时任务调度算法的对比, 此时的  $\gamma$  值约为 900, 在  $\gamma$  小于 900 时, GAAC 算法相对于轮转算法其优化效果大约为 20%, 对于蚁群算法其优化效果大约为 34%; 而在  $\gamma$  大于 900 时, GAAC 算法相对于贪心算法, 其优化效果大约为 25%。

随着任务数的增大优化的时间在不断扩大, 这是因

表 4 仿真实验参数

参数	参数值
虚拟机个数	100
虚拟机平台属性	异构
任务个数	500, 1 000, 1 500, 2 000, 2 500
GAAC( $\alpha, \beta$ )	1.0, 1.0
GAAC(ant)个数	5
$N_{max}$	50

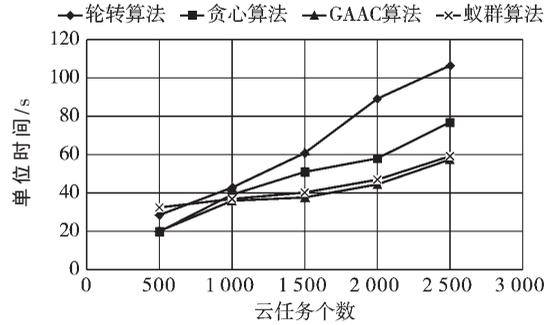


图 4 任务调度算法对比 3

为随着算法学习与迭代次数的增大, 其优化结果越来越倾向于局部最优。即是用 GAAC 算法进行云计算任务调度时, 考虑了以任务执行时间为优化目标, 同时也考虑以负载均衡为优化目标, 从而通过局部最优实现全局最优而实现了任务调度的时间和负载的均衡, 达到了云计算中的任务调度优化的效果。

#### 4 结论

优化的任务调度算法能高效地提升云计算的性能, 从而大大提升云计算的算力、用户响应时间等<sup>[14]</sup>。本文是基于模拟的 Cloudsim 云计算环境通过对现有的两种较常用的云计算算法进行不同的任务时间数的比较, 并将由于不同任务数这两种算法所具有的不同优点结合而形成了新的 GAAC 算法, 可以看出 GAAC 算法实现了基本的时间优化和初步的负载平衡, 并解决了蚁群算法任务数较低、学习次数较低而导致的优化效果较差和 Greedy 算法后期由于没有迭代机制而优化效率较差的问题。但是 GAAC 后期, 由于学习机制和大量的训练的原因会导致计算开销大, 并容易陷入局部优化、过早收敛而非全局最优化的结果。而现今对于云计算的任务调度算法有诸如 ACO、遗传、退火算法和两种或多种算法之间的融合实现优化, 其能更好地优化其云计算中局部最优和过早收敛的问题。而在云计算中, 诸如云平台中数据迁移的开销、引入安全性约束条件、服务间资源、资源共享等问题的提出和解决, 都将再进一步地优化现今的云计算算法, 从而实现更好的算力, 降低用户响应时间等问题, 这些都需对云计算的任务调度算法进一步完善和改进。

#### 参考文献

- [1] 周悦芝, 张迪. 近端云计算: 后云计算时代的机遇与挑战[J]. 计算机学报, 2019, 42(4): 677-700.
- [2] 谢志强, 邵侠. 一种考虑资源状态动态反馈的云计算调度算法[J]. 北京理工大学学报, 2019, 39(6): 624-637.
- [3] 高洪皓, 缪准扣, 刘浩宇, 等. 基于概率模型检验的云渲染任务调度定量验证[J]. 软件学报, 2020, 31(6): 1839-1859.
- [4] 张骥先, 谢宁, 张学杰, 等. 基于监督学习的可信云计算

(下转第 105 页)

## 3 结论

本文对差分进化算法进行了适当改进,较好地平衡了算法前期的全局探索和后期的局部开发能力,不仅能够避免早熟收敛现象,同时加快了收敛速度。算法以GDOP为数据支撑,设计合理的适应度函数,结合多点定位系统的实际工程建设情况,建立仿真模型,通过不断地迭代产生最优布站的站点坐标。仿真结果验证了最优布站方法相比于传统布站,对目标的高度和角度变化表现出了较好的鲁棒性,布站区域内的定位精度得到了显著的提高,为系统实际工程建设提供可靠的理论依据。

## 参考文献

- [1] 李家蓬,韩明,苑克剑.多点定位系统的布站设计[J].电光与控制,2016,23(1):71-74.
- [2] 宫峰勋,雷艳萍,许跃,等.机场场面多点定位系统定位精度研究[J].计算机工程,2011,37(15):276-278.
- [3] 崔自峰,闫修林,钱葵东.广域多点定位系统设计与验证[J].指挥信息系统与技术,2016,7(38):55-60.
- [4] 吴宏刚,刘昌忠,黄忠涛.机场场面多点定位系统远端站优选方法[J].电讯技术,2009,49(12):5-8.
- [5] 李露.基于标记牌识别的机场场面车辆定位技术研究[D].南京:南京航空航天大学,2015.
- [6] 易文周.粒子群与差分进化混合算法的研究[D].桂林:广西师范大学,2009.
- [7] 方力智,张翠芳,易芳.基于改进差分进化算法的RBF神经网络优化方法[J].成都大学学报(自然科学版),2009,

28(3):231-233.

- [8] 胡畔.基于差分进化算法的预测控制及其应用研究[D].兰州:兰州交通大学,2011.
- [9] 杨艳.基于微分进化算法的电力系统最优潮流[D].成都:西南交通大学,2009.
- [10] 苗晓锋,刘志伟.一种基于混合策略的差分进化算法研究[J].计算机应用与软件,2019,36(3):266-270.
- [11] 文武,乔龙辉,何鹏.基于自适应差分进化极限学习机的车牌识别算法[J].电子技术应用,2017,43(1):133-136,140.
- [12] 刘红平,黎福海.面向多目标优化问题的自适应差分进化算法[J].计算机应用与软件,2015,32(12):249-252.
- [13] 刘波,王凌,金以慧.差分进化算法研究进展[J].控制与决策,2007(7):3-11.
- [14] 吴亮红.差分进化算法及应用研究[D].长沙:湖南大学,2007.
- [15] 邵良琪,李署坚,邵定蓉.某导航系统中基于DSP的几何精度因子(GDOP)计算[J].测控技术,2004(z1):174-176.
- [16] 赵琨,何青益.基于GDOP的三站时差定位精度分析[J].无线电工程,2012,45(5):19-21.

(收稿日期:2020-06-06)

## 作者简介:

胥松寿(1987-),男,硕士,工程师,主要研究方向:多点定位、雷达监视、无线通信。

郭歆莹(1989-),女,博士,讲师,主要研究方向:无线通信。



扫码下载电子文档

(上接第100页)

- 资源拍卖机制研究[J].电子与信息学报,2019,41(5):1243-1250.
- [5] 王意洁,孙伟东.云计算环境下的分布式存储关键技术[J].软件学报,2012,23(4):962-986.
- [6] 周平,殷波,邱雪松,等.面向服务可靠性的云资源调度方法[J].电子学报,2019,47(5):1036-1043.
- [7] 尚志会,张建伟,蔡增玉,等.云桌面环境下基于蚁群算法的作业调度方法[J].计算机工程与设计,2017,38(6):1668-1672,1680.
- [8] 王万良,臧泽林,陈国棋,等.大规模云计算服务器优化调度问题的最优二元交换算法研究[J].通信学报,2019,40(5):180-191.
- [9] Wei Yi, Pan Li, Liu Shijun, et al. DRL-scheduling: an intelligent QoS-aware job scheduling framework for applications in clouds[J]. IEEE Access, 2018, 6: 55112-55125.
- [10] BHATTI S S, Fan Jiahao, Wang Kangrui, et al. An approximation algorithm for bounded task assignment problem in spatial crowdsourcing[J]. IEEE Transactions on Mobile Computing, 2020(99): 1.

- [11] ALHUSSIAN H, ZAKARIA N, PATEL A, et al. Investigating the schedulability of periodic real-time tasks in virtualized Cloud environment[J]. IEEE Access, 2019(7): 29533-29542.
- [12] HSIEH S Y, CHEN C T, CHEN C H, et al. Novel scheduling algorithms for efficient deployment of MapReduce applications in heterogeneous computing environments[J]. IEEE Transactions on Cloud Computing, 2018, 6(4): 1080-1095.
- [13] 胡涛,马晨辉,申立群,等.基于蚁群算法的测试任务调度优化方法[J].兵工学报,2019,40(6):1310-1316.
- [14] 胡志刚,常健,周舟.面向云环境中任务负载的粒子群优化调度策略[J].湖南大学学报(自然科学版),2019,46(8):117-123.

(收稿日期:2020-07-21)

## 作者简介:

杨戈(1974-),通信作者,男,博士,副教授,主要研究方向:计算机视觉、视觉跟踪技术, E-mail: yangge@pkusz.edu.cn。

吴俊言(1992-),男,硕士研究生,主要研究方向:云计算、下一代网络。



扫码下载电子文档

## 版权声明

经作者授权，本论文版权和信息网络传播权归属于《电子技术应用》杂志，凡未经本刊书面同意任何机构、组织和个人不得擅自复印、汇编、翻译和进行信息网络传播。未经本刊书面同意，禁止一切互联网论文资源平台非法上传、收录本论文。

截至目前，本论文已经授权被中国期刊全文数据库（CNKI）、万方数据知识服务平台、中文科技期刊数据库（维普网）、DOAJ、美国《乌利希期刊指南》、JST 日本科技技术振兴机构数据库等数据库全文收录。

对于违反上述禁止行为并违法使用本论文的机构、组织和个人，本刊将采取一切必要法律行动来维护正当权益。

特此声明！

《电子技术应用》编辑部

中国电子信息产业集团有限公司第六研究所