山西大学学报(自然科学版)47(5):943—953,2024
Journal of Shanxi University(Nat. Sci. Ed.)

DOI:10.13451/j.sxu.ns.2023155

基于Min(*c*,*V*)唤醒机制和活跃阈值的云虚拟机调度优化

李梦桃,徐秀丽*
(燕山大学 理学院,河北 秦皇岛 066004)

摘 要:为了满足云用户对系统响应性能的需求,同时进一步提高云系统的节能水平,论文提出一种新型虚拟机调度策略。当虚拟机处于休眠状态时,一旦缓冲区内的云任务数超过服务台数c或者休眠定时器到时,所有的虚拟机便立即停止休眠进入唤醒状态。在唤醒期结束时,如果缓冲区内等待的云任务数达到阈值N,虚拟机由唤醒状态转入活跃状态,否则转入空闲状态等待云任务到达。基于上述背景,构建具有Min(*c*,*V*)策略与活跃阈值N的可变到达速率的多服务台排队模型。利用矩阵几何解方法,推导出云系统稳态性能指标,并构建合理的收益函数讨论云系统收益问题,综合数值分析验证了所提策略的有效性。

关键词:云计算;唤醒机制;Min(*c*,*V*)策略;活跃阈值优化
中图分类号:TP393 **文献标志码**:A **文章编号**:0253-2395(2024)05-0943-11

Scheduling Optimization of Cloud Virtual Machines Based on Min(*c*,*V*) Wake-up Mechanism and Active Threshold

LI Mengtao, XU Xiuli*
(*School of Science, Yanshan University, Qinhuangdao* 066004*, China*)

Abstract: In order to meet the demand of cloud users for system response performance and further improve the energy saving level of cloud system, this paper proposes a new virtual machine (VM) scheduling policy. When VMs are in the sleep state, once the number of cloud tasks in the buffer exceeds the number of service stations c or the sleep timer ends, all VMs immediately stop sleeping and enter the wakeup state. Moreover, at the end of the wake-up period, if the number of waiting cloud tasks in the buffer reaches the threshold N, the VMs are transferred from the wake-up state to the active state. Otherwise, they are transferred to the idle state to wait for cloud tasks to arrive. Based on the above background, a multi-server queuing model with variable arrival rate and Min(c,V) mechanism and active threshold N is constructed. The steady-state performance indices of the cloud system are derived by using matrix-geometric solution method, and a reasonable benefit function is constructed to discuss the profit problem of the cloud system. Finally, the effectiveness of the proposed policy is verified by comprehensive numerical analysis.

Key words: cloud computing; wake-up mechanism; Min(c,V) policy; active threshold optimization

0 引言

云计算的概念是由谷歌于2006年首次提出[1],随着信息技术(Information Technology,IT)、计算机断层扫描(Computed Tomography,CT)行业的高速发展,云计算也被大众所熟知并逐渐成为信息技术产业的热点。随着5G时代的到来,云平台已经成为数字时代不可或缺的基础设施之一。

收稿日期:2023-04-21;**接受日期**:2023-10-12
基金项目:国家自然科学基金项目(62171143)
作者简介:李梦桃(1998-),女,河北邯郸人,硕士研究生,研究方向为排队论。E-mail:olimtao@163.com
* **通信作者**:徐秀丽(XU Xiuli),E-mail:xxl-ysu@163.com
引文格式:李梦桃,徐秀丽.基于Min(*c*,*V*)唤醒机制和活跃阈值的云虚拟机调度优化[J].山西大学学报(自然科学版),2024,47(5):943-953. DOI:10.13451/j.sxu.ns.2023155

随着其规模越来越大,云服务在为我们产生便利的同时也产生了巨大的能量消耗。有学者估计,2030年云数据中心能耗可达8 000 TWh[2],因而绿色云计算是时代发展的必然趋势。

Yang等[3]以博弈论为基础,将节点的可靠性作为优化目标,建立了云计算任务调度的合作博弈模型并提出一种平衡调度算法。Ding等[4]提出一种基于Q学习的高能效云计算任务调度模型,此模型分为两个阶段,第一阶段通过排队模型将到达的任务请求分配给系统中的服务器,第二阶段每台服务器会根据任务的松弛度对任务请求进行优先级排序,然后将任务分配给虚拟机,通过模拟实验证实了此模型的有效性。Duan等[5]提出了一种动态空间间隔预测方案,建立了可以预测中央处理器(central processing unit,CPU)空闲时间长度的统计模型,从而为即将到来的空闲时间选择耗能最少的休眠模式,提高CPU的利用率。Guo等[6]利用虚拟机的实时迁移技术,将空闲的虚拟机切换到睡眠模式或关机模式,并提出一种启发式算法,实验结果表明该模型和算法能有效降低系统能耗。Ashkan等[7]为了降低系统能耗,提出了一种基于负载平衡理论的节能模型,该模型空闲置和低负载状态的虚拟机于休眠状态,并提出了适用于该模型的扩展算法。Khorsand和Ramezanpour[8]为了满足不同用户的多种需求,提出一种基于最优最差算法和基于理想解相似度排序技术的任务调度算法,实验结果表明该算法可以有效降低能耗,提高虚拟机利用率。Panda和Jana[9]考虑到任务的完成时间和对资源的总利用率,提出一种节能的任务调度算法。实验结果显示此算法在节能水平和完工时间之间实现了良好的折中。Wang和Su[10]针对大数据环境下多云节点协作,提出了一种动态分层的资源分配算法,该算法根利用动态层次结构来减少资源分配过程中的通信流量。Li等[11]分别利用传统的时间序列模型和时间序列分割模型来降低系统能耗。

在专注于云系统节能的同时还应考虑到云用户对响应性能的需求,Dong等[12]为了最小化云任务执行时间,提出一种基于深度强化学习架构(reinforcement learning task scheduling,RLTS)的任务调度机制,该机制将具有优先关系的任务动态调度到云服务器中,实验结果表明,该算法在计算复杂度、求解质量和灵活度方面具有更高的性能。Singh等[13]为了降低执行成本和时间,提出了一种使用蚁群优化算法的资源调度技术,仿真结果显示,该算法能有效提高云系统效率。Zhang等[14]为了最小化数据集的访问延迟,提出了一种重点为优化虚拟机放置的模型,并利用分支定界算法求解出模型的最优解。Jin等[15]引入备用虚拟机模块,基于系统负载状况,动态关闭部分空闲虚拟机,同时调整剩余虚拟机的服务速率,构建了具有可变服务率和部分服务台同步休假的排队模型。李吉良等[16]将唤醒阈值和工作休假结合在一起,建立了N策略多重异步工作休假M/M/c排队模型,构建云系统成本函数,并利用蚁群智能寻优算法,给出了虚拟机调度策略的优化方案。王晓琛等[17]引入了Min(N, V)策略,将唤醒阈值N和长度为V的休眠定时器结合在一起,建立了多重同步休假排队模型,通过改进飞蛾扑火优化算法,给出了系统参数的最优组合。王秀双和金顺福[18]引入N-策略和休假延迟机制,建立了多重同步休假排队模型,构建系统成本函数,通过改进遗传算法,给出了系统参数的最优组合。Tong等[19]将Q学习与异构最早完成时间算法结合起来,提出一种新型任务调度算法,实验结果表明,该算法可以有效降低系统的响应时间。Qi[20]为了优化服务质量和任务执行时间,提出一种基于改进粒子群优化的资源调度方法。Wei和Zeng[21]为了提高云计算的效率,提出了一种基于混合差分并行调度的资源分配算法,该算法将资源分配问题转换成最小二乘问题,然后采用混合微分并行计算方法寻找最优解。Samriya和Kumar[22]为了在提高用户的服务质量的同时不违反服务水平协议,提出了一种模糊蚁群算法。Kumar等[23]为了在尽可能短的时间内将最佳资源分配给合适的虚拟机,提出了基于混合梯度下降金鹰优化算法高效异构资源调度过程,通过模仿金鹰的智能,来解决大数据流处理过程中用户需求的波动。Shen等[24]为了平衡系统能耗和响应性能,提出了一种基于排队论的随机模型并且开发了一种启发式算法来寻找最优系统参数。Shabeera等[25]为了优化虚拟机之间的数据传输延迟,提出一种用于数据中心虚拟机分配的近似算法。但是上述学者都忽略了物理机频繁切换状态带来的能量消耗。

本文综合考虑云系统的节能效果和响应性能,以及由于虚拟机频繁切换状态造成的能量浪费,引入一种新型的虚拟机调度策略,建立一个具有Min(c, V)策略与活跃阈值N的可变到达速率的多服务台排队模型。结合系统内云请求数量和虚拟机所处状态,构建二维连续时间马尔可夫链,通过矩阵几何解的方法求出稳态分布。进一步给出云系统的性能指标和社会收益函数,利用Matlab软件进行数值分析,分别讨论模型参数对于平均逗留时间、节能水平和系统收益的影响,并给出社会最优策略。

1 虚拟机调度策略及云系统模型构建

1.1 云虚拟机调度策略

在传统的云数据中心,即使系统中没有云任务,虚拟机仍保持活跃状态,这会产生大量的能量浪费,所以当系统中没有云任务时让虚拟机进入深度休眠状态是非常有必要的。考虑虚拟机休眠时对系统响应性能产生的负面影响,引入Min(c, V)唤醒模式,由阈值为c的计数器和长度为V的计时器双重控制虚拟机何时由休眠状态转入唤醒状态。为了避免虚拟机频繁切换状态,设置一个活跃阈值N($N>c$)决定虚拟机何时进入活跃状态。在快节奏的社会背景下,顾客的耐心是有限的,在面对网络卡顿、堵塞等情况下,云任务并不一定进入系统,存在顾客流失现象,故设定可变到达速率。虚拟机调度策略具体描述如下:

1)假设所有的虚拟机都托管在同一个物理机上,当所有的云任务(包括缓冲区内的)被处理完,虚拟机进入休眠状态,同时启动一个阈值为c的计数器和长度为V的计时器。此状态下虚拟机不处理云任务,云系统耗能较低,在此期间到达的云任务均在缓冲区内等待。若计时器或计数器任何一个失效,虚拟机结束休眠进入唤醒状态。

2)在唤醒状态下,系统处于高耗能状态。虚拟机仍不提供服务,唤醒期间到达的云任务在缓冲区内等待。唤醒期结束后,如果系统中云任务数量达到阈值N个,虚拟机由唤醒状态进入活跃状态,开始处理云任务。否则虚拟机进入空闲状态等待云任务到达。

3)当虚拟机处于活跃状态时,云任务以概率1进入系统。当虚拟机处于休眠状态、唤醒状态和空闲状态这三种情况时,云任务以概率p进入系统,以概率1−p消失。

在所提出的调度策略中,云系统中的虚拟机全部受控于一台物理机,物理机上运行着休眠定时器,唤醒定时器以及计数器,根据计时器和计数器的结果,虚拟机在四种状态之间进行切换。虚拟机的状态转移如图1所示。

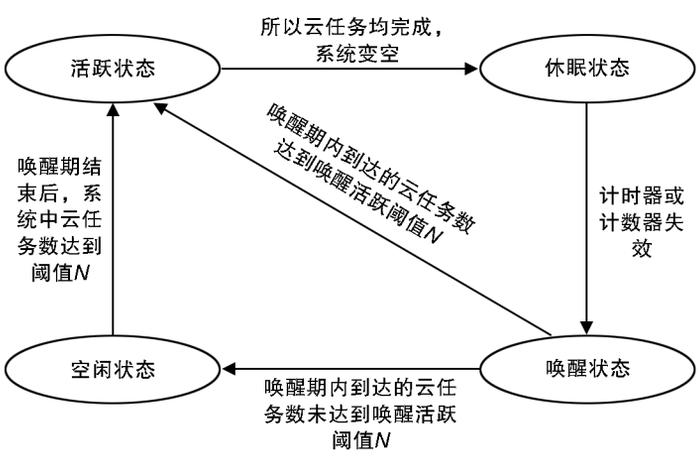

图1 虚拟机状态转移图

Fig. 1 State transition diagram of VM

1.2 系统模型

基于上述背景,将活跃状态抽象为忙期,休眠状态抽象为休假,唤醒状态抽象为启动过程,活跃阈值视为N-策略,本文构建一个具有N-策略与Min(c,V)策略的可变到达速率单重休假M/M/c排队模型。

虚拟机工作时间服从参数为$\mu(\mu>0)$的指数分布,当虚拟机服务完所有的云任务立即开始一段随机长度的休假,云系统会启动一个休假定时器,定时器的长度V服从参数为$\theta(\theta>0)$的指数分布。当定时器结束或系统中云任务数超过c个,虚拟机立即结束休假开始启动,启动时间服从参数为$\alpha(\alpha>0)$的指数分布。启动期结束后,由阈值N控制虚拟机转入活跃状态或者空闲状态。云任务到达时间间隔服从参数为$\lambda(\lambda>0)$的指数分布。云系统按照先到先服务(FCFS)的服务规则。假设到达间隔、服务时间、休假时间、启动时间相互独立。

假设$L(t)$表示t时刻系统中顾客的数量,$J(t)$表示t时刻虚拟机的状态,$J(t)=0,1,2,3$分别表示云系统处于休假、启动、空闲和忙期,则$\{(L(t),J(t)),t\geqslant 0\}$构成一个二维连续时间Markov链,其状态空间为:

$$\boldsymbol{\Omega}=\{(0,j):j=0,1,2\}\cup\{(i,j):1\leqslant i\leqslant c-1,j=0,1,2,3\}\cup$$
$$\{(i,j):c\leqslant i\leqslant N-1,j=1,2,3\}\cup\{(i,j):i\geqslant N,j=1,3\}。$$

模型的状态转移过程如图2所示,可以看出系统的状态转移只发生在相邻水平之间,二维Markov链$\{(L(t),J(t)),t\geqslant 0\}$是一个拟生灭过程。

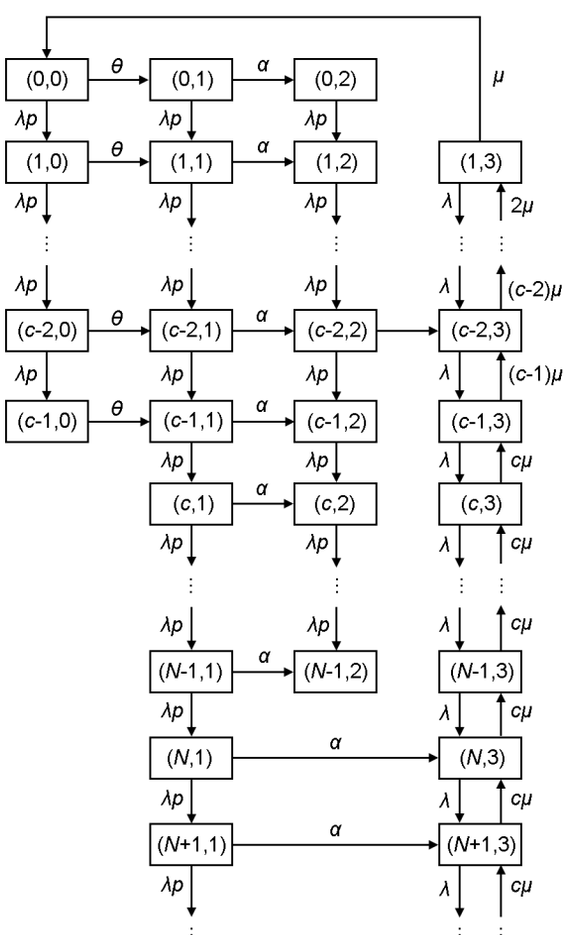

图2 Markov链的状态转移

Fig. 2 The state transition of Markov chain

使用词典序列,二维连续时间Markov链的无穷小生成元$\boldsymbol{Q}$可以表示为如下分块三对角形式:

$$\boldsymbol{Q}=\begin{pmatrix}
\boldsymbol{A}_0 & \boldsymbol{C}_0 & & & & & & & \\
\boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{C}_1 & & & & & & \\
 & \ddots & \ddots & \ddots & & & & & \\
 & & \boldsymbol{B}_c & \boldsymbol{A}_c & \boldsymbol{C}_c & & & & \\
 & & & \ddots & \ddots & \ddots & & & \\
 & & & & \boldsymbol{B}_{N-1} & \boldsymbol{A}_{N-1} & \boldsymbol{C}_{N-1} & & \\
 & & & & & \boldsymbol{B}_N & \boldsymbol{A} & \boldsymbol{C} & \\
 & & & & & & \boldsymbol{B} & \boldsymbol{A} & \boldsymbol{C} \\
 & & & & & & & \ddots & \ddots & \ddots
\end{pmatrix},$$

其中$\boldsymbol{B}_k$,$\boldsymbol{C}_k$和$\boldsymbol{A}_k$分别表示经过一步转移后系统水平下降子阵、系统水平上升子阵和系统水平不变子阵,各子块矩阵具体表达式如下:

(1) $\boldsymbol{B}_1=\begin{pmatrix}0&0&0\\0&0&0\\0&0&0\\\mu&0&0\end{pmatrix}$, $\boldsymbol{B}_k=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&k\mu\end{pmatrix}$, $2\leqslant k<c$, $\boldsymbol{B}_c=\begin{pmatrix}0&0&0&0\\0&0&0&0\\0&0&0&c\mu\end{pmatrix}$, $\boldsymbol{B}_k=\begin{pmatrix}0&0&0\\0&0&0\\0&0&c\mu\end{pmatrix}$, $c+1\leqslant k\leqslant N-1$, $\boldsymbol{B}_N=\begin{pmatrix}0&0&0\\0&0&c\mu\end{pmatrix}$, $\boldsymbol{B}=\begin{pmatrix}0&0\\0&c\mu\end{pmatrix}$, $k\geqslant N+1$。

(2) $\boldsymbol{C}_0=\begin{pmatrix}\lambda p&0&0&0\\0&\lambda p&0&0\\0&0&\lambda p&0\end{pmatrix}$, $\boldsymbol{C}_k=\begin{pmatrix}\lambda p&0&0&0\\0&\lambda p&0&0\\0&0&\lambda p&0\\0&0&0&\lambda\end{pmatrix}$, $1\leqslant k<c-1$, $\boldsymbol{C}_{c-1}=\begin{pmatrix}\lambda p&0&0\\\lambda p&0&0\\0&\lambda p&0\\0&0&\lambda\end{pmatrix}$, $\boldsymbol{C}_k=\begin{pmatrix}\lambda p&0&0\\0&\lambda p&0\\0&0&\lambda\end{pmatrix}$, $c\leqslant k<N-1$, $\boldsymbol{C}_{N-1}=\begin{pmatrix}\lambda p&0\\0&\lambda p\\0&\lambda\end{pmatrix}$, $\boldsymbol{C}=\begin{pmatrix}\lambda p&0\\0&\lambda\end{pmatrix}$, $k\geqslant N$。

(3) $\boldsymbol{A}_0=\begin{pmatrix}-(\lambda p+\theta)&\theta&0\\0&-(\lambda p+\alpha)&\alpha\\0&0&-\lambda p\end{pmatrix}$, $\boldsymbol{A}_k=\begin{pmatrix}-(\lambda p+\theta)&\theta&0&0\\0&-(\lambda p+\alpha)&\alpha&0\\0&0&-\lambda p&0\\0&0&0&-(\lambda+x\mu)\end{pmatrix}$, $1\leqslant k\leqslant c-1$, $\boldsymbol{A}_k=\begin{pmatrix}-(\lambda p+\alpha)&\alpha&0\\0&-\lambda p&0\\0&0&-(\lambda+c\mu)\end{pmatrix}$, $c\leqslant k\leqslant N-1$, $\boldsymbol{A}=\begin{pmatrix}-(\lambda p+\alpha)&\alpha\\0&-(\lambda+c\mu)\end{pmatrix}$, $k\geqslant N$。

2 系统稳态解析

拟生灭过程正常返当且仅当矩阵二次方程$\boldsymbol{R}^2\boldsymbol{B}+\boldsymbol{R}\boldsymbol{A}+\boldsymbol{C}=0$最小非负解$\boldsymbol{R}$的谱半径$SP(\boldsymbol{R})<1$,$\boldsymbol{R}$称为率阵。由于$\boldsymbol{B}$,$\boldsymbol{C}$,$\boldsymbol{A}$都是2阶上三角阵,因此,率阵$\boldsymbol{R}$也必然有相同的结构。设

$$\boldsymbol{R}=\begin{pmatrix}r_{11}&r_{12}\\0&r_{22}\end{pmatrix},$$

将$\boldsymbol{B}$,$\boldsymbol{C}$,$\boldsymbol{A}$,$\boldsymbol{R}$代入矩阵方程二次方程$\boldsymbol{R}^2\boldsymbol{B}+\boldsymbol{R}\boldsymbol{A}+\boldsymbol{C}=0$中,建立二次代数方程组

$$\begin{cases}\lambda p-(\lambda p+\alpha)r_{11}=0,\\ c\mu(r_{11}r_{12}+r_{12}r_{22})+\alpha r_{11}-(\lambda+c\mu)r_{12}=0,\\ c\mu(r_{22})^2-(\lambda+c\mu)r_{22}+\lambda=0,\end{cases}$$

求解上述方程组可以得到转移率矩阵如下

$$\boldsymbol{R}=\begin{pmatrix}\dfrac{\lambda p}{\lambda p+\alpha} & \rho p\\ 0 & \rho\end{pmatrix},$$

可以看出$SP(\boldsymbol{R})=\max\left\{\dfrac{\lambda p}{\lambda p+\alpha},\rho\right\}<1$,当且仅当$\rho<1$时。由文献[26]的定理2.4可知$\rho<1$时该拟生灭过程正常返,存在平稳概率分布。

当$\rho<1$时,记$\pi_{i,j}$表示系统稳态概率分布,即

$$\pi_{i,j}=\lim_{t\to\infty}\{L(t)=i,J(t)=j\},(i,j)\in\boldsymbol{\Omega},$$

令$\boldsymbol{\pi}_i(i\geqslant 0)$表示稳态下系统水平为$i$的概率向量,则

$$\boldsymbol{\pi}_i=\begin{cases}(\pi_{i,0},\pi_{i,1},\pi_{i,2}),i=0,\\(\pi_{i,0},\pi_{i,1},\pi_{i,2},\pi_{i,3}),1\leqslant i\leqslant c-1,\\(\pi_{i,1},\pi_{i,2},\pi_{i,3}),c\leqslant i\leqslant N-1,\\(\pi_{i,1},\pi_{i,3}),i\geqslant N。\end{cases}$$

构造方阵$B[\boldsymbol{R}]$如下

$$B[\boldsymbol{R}]=\begin{pmatrix}\boldsymbol{A}_0 & \boldsymbol{C}_0 & & & & & & & \\ \boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{C}_1 & & & & & & \\ & \ddots & \ddots & \ddots & & & & & \\ & & \boldsymbol{B}_c & \boldsymbol{A}_c & \boldsymbol{C}_c & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & \boldsymbol{B}_{N-1} & \boldsymbol{A}_{N-1} & \boldsymbol{C}_{N-1} & & \\ & & & & & \boldsymbol{B}_N & \boldsymbol{A} & \boldsymbol{C} \\ & & & & & & \boldsymbol{B} & \boldsymbol{A}+\boldsymbol{RB}\end{pmatrix}。$$

记$\boldsymbol{e}$和$\boldsymbol{e}_i$,$i=1,2,3$均为适当维数的全1列向量,由拟生灭过程与矩阵几何解方法知,稳态概率满足如下方程组:

$$\begin{cases}(\boldsymbol{\pi}_0,\boldsymbol{\pi}_1,\boldsymbol{\pi}_2,\cdots,\boldsymbol{\pi}_{N+1})B[\boldsymbol{R}]=0, & (1)\\ \boldsymbol{\pi}_i=\boldsymbol{\pi}_{N+1}\boldsymbol{R}^{i-N-1},i>N+1, & (2)\\ \sum\limits_{i=0}^{\infty}\boldsymbol{\pi}_i\boldsymbol{e}=1。 & (3)\end{cases}$$

展开(1)式可得

$$\boldsymbol{\pi}_0\boldsymbol{A}_0+\boldsymbol{\pi}_1\boldsymbol{B}_1=0, \tag{4}$$

$$\boldsymbol{\pi}_{n-1}\boldsymbol{C}_{n-1}+\boldsymbol{\pi}_n\boldsymbol{A}_n+\boldsymbol{\pi}_{n+1}\boldsymbol{B}_{n+1}=0,\ n=1,2,\cdots,N, \tag{5}$$

$$\boldsymbol{\pi}_N\boldsymbol{C}+\boldsymbol{\pi}_{N+1}(\boldsymbol{A}+\boldsymbol{RB})=0。 \tag{6}$$

整理(4)式可得

$$\boldsymbol{\pi}_0=\boldsymbol{\pi}_1\boldsymbol{B}_1(-\boldsymbol{A}_0)^{-1}=\boldsymbol{\pi}_1\boldsymbol{\omega}_1。 \tag{7}$$

利用(7)式和(5)式依次递推可得

$$\boldsymbol{\pi}_n=\boldsymbol{\pi}_{n+1}\boldsymbol{B}_{n+1}[-(\boldsymbol{\omega}_n\boldsymbol{C}_{n-1}+\boldsymbol{A}_n)]^{-1}=\boldsymbol{\pi}_{n+1}\boldsymbol{\omega}_{n+1},\ n=1,2,\cdots,N。 \tag{8}$$

将(2)式、(7)式和(8)式代入(3)式可得

$$\boldsymbol{\pi}_{N+1}\prod_{i=1}^{N+1}\boldsymbol{\omega}_i\boldsymbol{e}_2+\boldsymbol{\pi}_{N+1}\sum_{k=1}^{c-1}\prod_{i=k}^{N+1}\boldsymbol{\omega}_i\boldsymbol{e}_1+\boldsymbol{\pi}_{N+1}\sum_{k=c}^{N-1}\prod_{i=k}^{N+1}\boldsymbol{\omega}_i\boldsymbol{e}_2+\boldsymbol{\pi}_{N+1}\boldsymbol{\omega}_{N+1}\boldsymbol{e}_3+\boldsymbol{\pi}_{N+1}(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_3=1。 \tag{9}$$

整理(9)式得

$$\boldsymbol{\pi}_{N+1}\left[\sum_{k=1}^{c-1}\prod_{i=k}^{N+1}\boldsymbol{\omega}_i\boldsymbol{e}_1+\left(\prod_{i=1}^{N+1}\boldsymbol{\omega}_i+\sum_{k=c}^{N+1}\prod_{i=k}^{N+1}\boldsymbol{\omega}_i\right)\boldsymbol{e}_2\right]+\boldsymbol{\pi}_{N+1}\left[\boldsymbol{\omega}_{N+1}\boldsymbol{e}_3+(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_3\right]=1。 \tag{10}$$

将(8)式代入(6)式整理可得

$$\boldsymbol{\pi}_{N+1}\boldsymbol{\omega}_{N+1}\boldsymbol{C}+\boldsymbol{\pi}_{N+1}\boldsymbol{A}+\boldsymbol{\pi}_{N+1}\boldsymbol{RB}=0。\tag{11}$$

联合(10)式和(11)式可以求出$\boldsymbol{\pi}_{N+1}$,进而可得稳态概率分布。

基于系统模型的稳态分布,给出系统稳态性能指标:

(1)云系统处于休假期、启动期、空闲期、忙期的概率分别为:

$$P_1=\sum_{i=0}^{c-1}\pi_{i,0},P_2=\sum_{i=0}^{\infty}\pi_{i,1},P_3=\sum_{i=0}^{N-1}\pi_{i,2},P_4=\sum_{i=1}^{\infty}\pi_{i,3}。$$

(2)云任务损失概率P_l为:

$$P_l=\sum_{i=0}^{c-1}(1-p)(\pi_{i,0}+\pi_{i,1}+\pi_{i,2})+\sum_{i=c}^{N-1}(1-p)(\pi_{i,1}+\pi_{i,2})+\sum_{i=N}^{\infty}(1-p)\pi_{i,1}=$$

$$\sum_{i=0}^{c-1}(1-p)(\pi_{i,0}+\pi_{i,1}+\pi_{i,2})+\sum_{i=c}^{N-1}(1-p)(\pi_{i,1}+\pi_{i,2})+(1-p)\left(\pi_{N,1}+\boldsymbol{\pi}_{N+1}\left(\boldsymbol{I}+\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-1}\right)\boldsymbol{e}^*\right),$$

其中$\boldsymbol{e}^*=(1\quad 0)^{\mathrm{T}}$。

(3)云任务平均队长$E(T)$为:

$$E(T)=\sum_{i=0}^{c-1}i(\pi_{i,0}+\pi_{i,1}+\pi_{i,2}+\pi_{i,3})+\sum_{i=c}^{N-1}i(\pi_{i,1}+\pi_{i,2}+\pi_{i,3})+\sum_{i=N}^{\infty}i(\pi_{i,1}+\pi_{i,3})=\sum_{i=0}^{c-1}i\boldsymbol{\pi}_i\boldsymbol{e}_1+\sum_{i=c}^{N-1}i\boldsymbol{\pi}_i\boldsymbol{e}_2+$$

$$N\boldsymbol{\pi}_N\boldsymbol{e}_3+(N+1)\boldsymbol{\pi}_{N+1}\boldsymbol{e}_3+\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-2}\boldsymbol{e}_3+(N+1)\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_3。$$

(4)云任务平均逗留时间$E(S)$为:

$$E(S)=\frac{1}{\lambda p}\left(\sum_{i=0}^{c-1}i(\pi_{i,0}+\pi_{i,1}+\pi_{i,2})\right)+\frac{1}{\lambda p}\left(\sum_{i=c}^{N-1}i(\pi_{i,1}+\pi_{i,2})\right)+\frac{1}{\lambda p}\sum_{i=N}^{\infty}i\pi_{i,1}+\frac{1}{\lambda}\sum_{i=0}^{\infty}i\pi_{i,3}=$$

$$\frac{1}{\lambda p}\left(\sum_{i=0}^{c-1}i\boldsymbol{\pi}_i\boldsymbol{e}_1^*+\sum_{i=c}^{N-1}i\boldsymbol{\pi}_i\boldsymbol{e}_2^*+N\boldsymbol{\pi}_N\boldsymbol{e}^*\right)+\frac{1}{\lambda p}\left((N+1)\boldsymbol{\pi}_{N+1}\boldsymbol{e}^*+\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-2}\boldsymbol{e}^*+\right.$$

$$\left.\frac{1}{\lambda p}((N+1)\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-2}\boldsymbol{e}^*+\frac{1}{\lambda}\left(\sum_{i=0}^{N+1}i\pi_{i,3}+\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-2}\boldsymbol{e}_3^*+(N+1)\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}_3^*\right),\right.$$

其中$\boldsymbol{e}_1^*=(1,1,1,0)^{\mathrm{T}},\boldsymbol{e}_2^*=(1,1,0)^{\mathrm{T}},\boldsymbol{e}_3^*=(0,1)^{\mathrm{T}}$。

(5)云系统平均节能水平$E(L)$如下:

$$E(L)=\sum_{i=0}^{c-1}(W_b-W_v)\pi_{i,0}+\sum_{i=0}^{N-1}(W_b-W_l)\pi_{i,2}-\sum_{i=0}^{\infty}(W_s-W_b)\pi_{i,1}-$$

$$W_t\left(\sum_{i=0}^{c-1}\theta\pi_{i,0}+\lambda p\pi_{c-1,0}\right)-W_t\left(\lambda p\pi_{N-1,2}+\sum_{i=0}^{\infty}\alpha\pi_{i,1}\right)=$$

$$\sum_{i=0}^{c-1}(W_b-W_v)\pi_{i,0}+\sum_{i=0}^{N-1}(W_b-W_l)\pi_{i,2}-(W_s-W_b)\left(\sum_{i=0}^{N+1}\pi_{i,1}+\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}^*\right)-$$

$$W_t\left(\sum_{i=0}^{c-1}\theta\pi_{i,0}+\lambda p\pi_{c-1,0}+\lambda p\pi_{N-1,2}\right)-W_t\alpha\left(\sum_{i=0}^{N+1}\pi_{i,1}+\boldsymbol{\pi}_{N+1}\boldsymbol{R}(\boldsymbol{I}-\boldsymbol{R})^{-1}\boldsymbol{e}^*\right),$$

其中W_v、W_s、W_l、W_b分别表示虚拟机处于休假、启动、空闲和忙期时单位时间内产生的能量消耗,W_t表示虚拟机进行一次状态切换产生的能量消耗。

3 数值分析

设置如表1所示的实验参数,进一步验证云虚拟机调度策略的有效性 。

通过图3可以发现,当阈值N和休假参数θ固定时,$E(S)$随着启动参数α的增加而减少。当启动参数α变大,启动阶段持续的时间变短,因此云任务平均逗留时间减少。此外由图3可知,当虚拟机启动参数α和休假参数θ固定时,$E(S)$随着阈值N的增加而增加。当阈值N变大,云任务处于空闲阶段的时间变长。而虚拟机处于空闲阶段时不工作,所有的云任务均在缓冲区等待。因此

云任务平均逗留时间增加。当虚拟机启动参数α和阈值N固定时,$E(S)$随着休假参数θ的增加而减少。当休假参数θ变大,虚拟机休假时间变短,因此云任务平均逗留时间减少。

表1 数值实验中系统参数值及单位的设置情况

Table 1 The settings of system parameter values and units in numerical experiments

参数	值
虚拟机总数	10台
虚拟机服务率μ	16 ms^{-1}
云任务到达率λ	90 ms^{-1}
云任务进入系统的概率p	0.6
空闲过程耗能水平W_I	5 mW
忙期过程耗能水平W_b	20 mW
休假过程耗能水平W_v	2 mW
启动过程耗能水平W_s	25 mW
状态切换时的耗能水平W_t	0.2 mJ

通过图4可以发现,当阈值N和休假参数θ固定时,$E(L)$随着启动参数α的增加而增加。启动阶段虚拟机处于高耗能状态,当启动参数α变大,启动持续的时间变短,因此云系统节能水平增加。此外,由图4可知当虚拟机启动参数α和休假参数θ固定时,$E(L)$随着阈值N的增加而增加。当阈值N变大,云任务处于空闲阶段的时间变长。空闲阶段虚拟机处于节能状态,因此云系统节能水平增加。当虚拟机启动参数α和阈值N固定时,$E(L)$随着休假参数θ的增加而减少。休假阶段虚拟机处于节能状态,当休假参数θ变大,休假持续的时间变短,因此云系统节能水平减少。

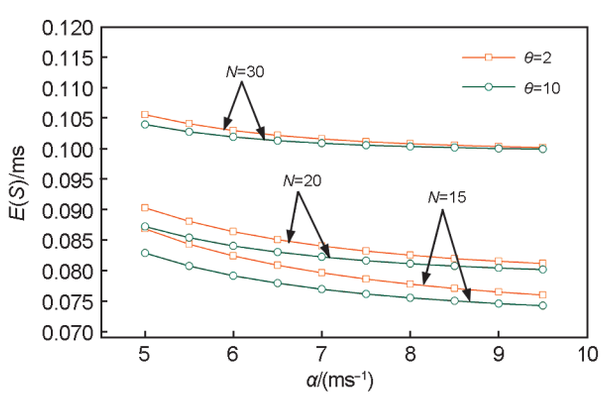

图3 在不同的阈值N和休假参数θ下云任务平均逗留时间$E(S)$随唤醒参数α的变化趋势

Fig. 3 The variation trend of average sojourn time $E(S)$ with the wake-up parameter α under different threshold N and vacation parameter θ

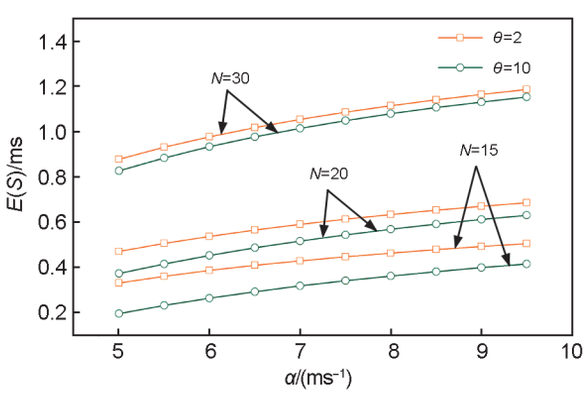

图4 在不同阈值N和休假参数θ下云系统节能水平$E(L)$随唤醒参数α的变化趋势

Fig. 4 The variation trend of cloud system energy-saving level $E(L)$ with the wake-up parameter α under different threshold N and vacation parameter θ

图3和图4的实验结果表明,启动参数α、休假参数θ和阈值N对云系统的性能都有着不容忽视的影响。从云任务的响应需求来看,启动参数α和休假参数θ越大越好,而阈值N越小越好。从云系统的节能需求来看,阈值N和启动参数α越大越好,休假参数θ越小越好。因此,面对不同的需求,需要联合优化阈值N、休假参数θ和启动参数α。

4 收益分析

下面讨论云系统收益问题,假设R表示云任务被服务完后获得的回报,C表示云任务停留在系统中需要付出的成本,U表示云系统通过节约单位能耗而增加的收入,B表示一个云任务请求消

失后带来的潜在损失,$\bar{\lambda}$为云任务的有效到达率。云系统收益函数表示为:

$$G=\bar{\lambda}(R-CE(S))+UE(L)-BP_l,$$

其中$\bar{\lambda}=\lambda\sum_{i=1}^{N+1}\pi_{i,3}+\lambda p(1-\sum_{i=1}^{N+1}\pi_{i,3})$。

为了保证云系统盈利,需要满足$\bar{\lambda}(R-CE(S))+UE(L)-BC_l>0$,即:

$$\bar{\lambda}>\frac{BC_l-UE(L)}{R-CE(S)}。$$

沿用表1的实验参数,并设置$N=15$,$R=1$,$C=0.7$,$U=0.4$,$B=0.5$,当$\lambda=90$,$\alpha=2.5$,由图5可知,不管θ取何值,G随着μ的增加均呈现先增加后减少的趋势,这是因为随着μ增加,虚拟机的服务时间变短,云任务停留在系统中需要付出的成本减少,系统收益增加,但是随着μ不断增加系统节能水平开始下降,使得系统收益减少。当μ固定时,θ增加即休假时间变短,一方面使得云任务停留在系统中需要付出的成本减少,另一方面云任务平均逗留时间变短所以云任务更倾向于进入系统,故系统的潜在损失变小,G便会越来越大。

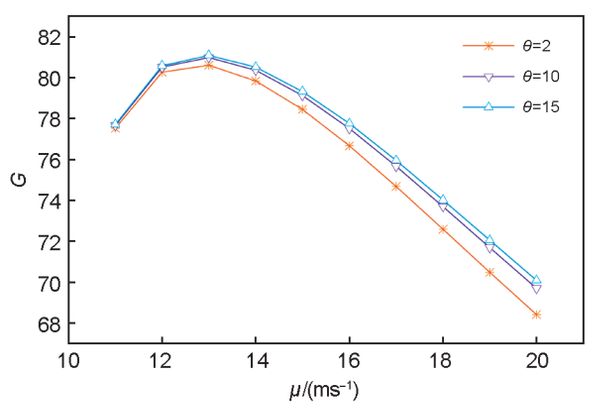

图5 在不同的休假参数θ下系统收益G随着服务率μ的变化趋势

Fig. 5 The variation trend of system benefit G with the service rate μ under vacation parameter θ

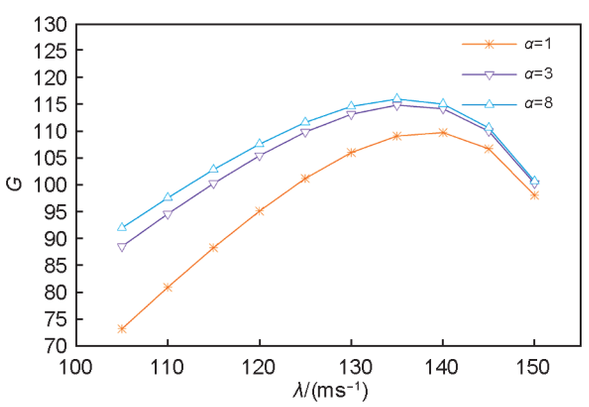

图6 在不同的唤醒参数α下系统收益G随着到达率λ的变化趋势

Fig. 6 The variation trend of system benefit G with the arrival rate λ under wake-up parameter α

当$\mu=16$,$p=0.5$,由图6可知,G随着λ的增加均呈现先增加后减少的趋势,并且有唯一的到达率λ^*使得G达到最大值,λ^*称为社会最优到达率,之所以呈现这种趋势是因为当λ变大时进入系统的顾客变多,系统收益变大,但随着λ越来越大,平均逗留时间增加,云任务停留在系统中需要付出的成本增加,系统收益减少。当λ固定时,随着α增加,系统通过节约单位能耗而产生的收入增加,系统收益增加。

所以为了提高系统收益,一方面需要降低云系统启动时间,另一方面要促使到达率λ接近社会最优到达率λ^*,当λ较大时,可以通过对选择进入系统的云任务征收入场费等方式降低云任务进入系统的欲望,当λ较小时,可以降低休假时间来吸引云任务进入系统。

5 结论

本文从云任务的排队场景出发,综合考虑云系统的节能效果和响应性能,提出一种融合Min(c,V)唤醒机制与活跃阈值N的云计算中心节能策略。利用Matlab软件进行数值分析,通过数值例子评估了系统参数对性能指标和收益函数的影响。实验结果表明,从系统的响应性能出发,需要提高启动参数和休假参数,降低阈值N。从系统的节能需求出发,需要增加阈值N和启动参数,

降低休假参数,故而面对不同的需求,须联合优化多个系统参数。云系统还应采取一定的措施促使到达率接近社会最优到达率,当到达率较大时,可以通过对选择进入系统的云任务征收入场费等方式降低云任务进入系统的欲望,当到达率较小时,可以降低休假时间来吸引云任务进入系统,以此来提高云系统收益。

参考文献:

[1] 任鹏燕, 田康叶, 张敏. 云计算:新一轮黄金发展期开启[J]. 中国电信业, 2022, **4**: 15−18. DOI: 10.3969/j.issn.1671-3060.2022.04.004.
REN P Y, TIAN K Y, ZHANG M. Cloud Computing: A New Golden Period of Development[J]. *China Telecommun Trade*, 2022, **4**: 15−18. DOI: 10.3969/j. issn. 1671-3060.2022.04.004.

[2] JAYANETTI A, BUYYA R. J-OPT: a Joint Host and Network Optimization Algorithm for Energy-efficient Workflow Scheduling in Cloud Data Centers[C]//Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing. New York: ACM, 2019: 199−208. DOI: 10.1145/3344341.3368822.

[3] YANG J C, JIANG B, LV Z H, *et al*. A Task Scheduling Algorithm Considering Game Theory Designed for Energy Management in Cloud Computing[J]. *Future Gener Comput Syst*, 2020, **105**: 985−992. DOI: 10.1016/j.future.2017.03.024.

[4] DING D, FAN X C, ZHAO Y H, *et al*. Q-learning Based Dynamic Task Scheduling for Energy-efficient Cloud Computing[J]. *Future Gener Comput Syst*, 2020, **108**: 361−371. DOI: 10.1016/j.future.2020.02.018.

[5] DUAN L D, ZHAN D Y, HOHNERLEIN J. Optimizing Cloud Data Center Energy Efficiency *via* Dynamic Prediction of CPU Idle Intervals[C]//2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015: 985−988. DOI: 10.1109/CLOUD.2015.133.

[6] GUO L Z, ZHANG Y F, ZHAO S G. Heuristic Algorithms for Energy and Performance Dynamic Optimization in Cloud Computing[J]. *Int Compu Inform*, 2017, **36**(6): 1335−1360. DOI: 10.4149/cai_2017_6_1335.

[7] PAYA A, MARINESCU D C. Energy-aware Load Balancing and Application Scaling for the Cloud Ecosystem [J]. *IEEE Trans Cloud Comput*, 2017, **5**(1): 15−27. DOI: 10.1109/TCC.2015.2396059.

[8] KHORSAND R, RAMEZANPOUR M. An Energy-efficient Task-scheduling Algorithm Based on a Multi-criteria Decision-making Method in Cloud Computing[J]. *Int J Commun Syst*, 2020, **33**(9): e4379. DOI: 10.1002/dac.4379.

[9] PANDA S K, JANA P K. An Energy-efficient Task Scheduling Algorithm for Heterogeneous Cloud Computing Systems[J]. *Cluster Comput*, 2019, **22**(2): 509−527. DOI: 10.1007/s10586-018-2858-8.

[10] WANG Z J, SU X X. Dynamically Hierarchical Resource-allocation Algorithm in Cloud Computing Environment[J]. *J Supercomput*, 2015, **71**(7): 2748−2766. DOI: 10.1007/s11227-015-1416-x.

[11] LI J, LIU X, ZHAO Z, *et al*. Energy Consumption Prediction Based on Time-series Models for CPU-intensive Activities in the Cloud[M]//Algorithms and Architectures for Parallel Processing. Cham: Springer International Publishing, 2015: 756−769. DOI: 10.1007/978-3-319-27140-8_52.

[12] DONG T T, XUE F, XIAO C B, *et al*. Task Scheduling Based on Deep Reinforcement Learning in a Cloud Manufacturing Environment[J]. *Concurr Comp-Pract E*, 2020, **32**(11): e5654. DOI: 10.1002/cpe.5654.

[13] SINGH H, BHASIN A, KAVERI P. SECURE: Efficient Resource Scheduling by Swarm in Cloud Computing[J]. *J Discrete Math Sci Cryptogr*, 2019, **22**(2): 127−137. DOI: 10.1080/09720529.2019.1576334.

[14] ZHANG X Y, LI K Q, ZHANG Y. Optimising Data Access Latencies of Virtual Machine Placement Based on Greedy Algorithm in Datacentre[J]. *Int J Comput Sci Eng*, 2019, **18**(2): 186−194. DOI: 10.1504/ijcse.2019.097945.

[15] HAO S S, QIE X C, JIN S F. Virtual Machine Allocation Strategy in Energy-efficient Cloud Data Centres[J]. *Int J Commun Netw Distributed Syst*, 2019, **22**(2): 181−195. DOI: 10.1504/ijcnds.2019.10018156.

[16] 李吉良, 秦兵, 李文江, 等. 融合唤醒阈值与半休眠模式的云虚拟机调度策略[J]. 燕山大学学报, 2020, **44**(4):370−378. DOI: 10.3969/j.issn.1007-791X.2020.04.006.
LI J L, QIN B, LI W J, *et al*. Cloud Virtual Machine Scheduling Strategy with Awakening Threshold and Semi-sleep Mode[J]. *J Yanshan Univ*, 2020, **44**(4): 370−378. DOI: 10.3969/j.issn.1007-791X.2020.04.006

[17] 王晓琛, 王宇廷, 张丽媛, 等. 基于(N, T)休眠机制的云计算中心节能策略及优化[J]. 高技术通讯, 2020, **30**(8)805−813. DOI: 10.3772/j.issn.1002-0470.2020.08.006.
WANG X C, WANG Y T, ZHANG L Y, *et al*. Energy Saving Strategy and Optimization of Cloud Computing Centers Based on (N, T) Sleep Mechanism[J]. *Chin High*

Technol Lett, 2020, **30**(8): 805−813. DOI: 10.3772/j.issn.1002-0470.2020.08.006.

[18] 王秀双, 金顺福. 基于新型休眠机制的云任务调度策略的研究[J]. 高技术通讯, 2018, **28**(11): 907−914. DOI: 10.3772/j.issn.1002-0470.2018.11-12.003.
WANG X S, JIN S F. Research on a Cloud Task Scheduling Strategy Based on a Novel Sleep Mechanism[J]. *Chin High Technol Lett*, 2018, **28**(11): 907−914. DOI: 10.3772/j.issn.1002-0470.2018.11-12.003.

[19] TONG Z, DENG X M, CHEN H J, *et al*. QL-HEFT: A Novel Machine Learning Scheduling Scheme Base on Cloud Computing Environment[J]. *Neural Comput & Applic*, 2020, **32**(10): 5553−5570. DOI: 10.1007/s00521-019-04118-8.

[20] QI W Q. Optimization of Cloud Computing Task Execution Time and User QoS Utility by Improved Particle Swarm Optimization[J]. *Microprocess Microsyst*, 2021, **80**: 103529. DOI: 10.1016/j.micpro.2020.103529.

[21] WEI J, ZENG X F. Optimal Computing Resource Allocation Algorithm in Cloud Computing Based on Hybrid Differential Parallel Scheduling[J]. *Cluster Comput*, 2019, **22**(3): 7577−7583. DOI: 10.1007/s10586-018-2138-7.

[22] SAMRIYA J K, KUMAR N. Fuzzy Ant Bee Colony for Security and Resource Optimization in Cloud Computing [C]//2020 5th International Conference on Computing, Communication and Security (ICCCS). New York: IEEE, 2020: 1−5. DOI: 10.1109/ICCCS49678.2020.9276898.

[23] KUMAR J N, BALASUBRAMANIAN C. Hybrid Gradient Descent Golden Eagle Optimization (HGDGEO) Algorithm-based Efficient Heterogeneous Resource Scheduling for Big Data Processing on Clouds[J]. *Wireless Pers Commun*, 2023, **129**(2): 1175−1195. DOI: 10.1007/s11277-023-10182-0.

[24] SHEN D, LUO J Z, DONG F, *et al*. Stochastic Modeling of Dynamic Right-sizing for Energy-efficiency in Cloud Data Centers[J]. *Future Gener Comput Syst*, 2015, **48**: 82−95. DOI: 10.1016/j.future.2014.09.012.

[25] SHABEERA T P, MADHU KUMAR S D, CHANDRAN P. Curtailing Job Completion Time in MapReduce Clouds through Improved Virtual Machine Allocation[J]. *Comput Electr Eng*, 2017, **58**: 190−202. DOI: 10.1016/j.compeleceng.2016.10.009.

[26] 田乃硕, 岳德权. 拟生灭过程与矩阵几何解[M]. 北京: 科学出版社, 2002.
TIAN N S, YUE D Q. Quasi-birth and Death Process and Matrix Geometric Solution[M]. Beijing: Science Press, 2002.