

# 基于( $N, T$ )休眠机制的云计算中心节能策略及优化<sup>①</sup>

王晓琛<sup>②\*\*\*</sup> 王宇廷<sup>\* \*\*</sup> 张丽媛<sup>\*\*\*</sup> 金顺福<sup>③\* \*\*</sup>

(<sup>\*</sup> 燕山大学信息科学与工程学院 秦皇岛 066004)

(<sup>\*\*</sup> 河北省计算机虚拟技术与系统集成重点实验室 秦皇岛 066004)

(<sup>\*\*\*</sup> 廊坊师范学院理学院 廊坊 065000)

**摘要** 随着云应用种类的不断增多及数量的持续增长,云用户对响应性能的需求呈现出多样化趋势,同时云计算中心对其节能水平的要求越来越高。为了在保证云计算中心节能效果的前提下,提高云用户响应性能,面向云数据中心的虚拟机引入( $N, T$ )休眠机制,提出一种虚拟机节能策略。结合唤醒阈值  $N$  及长度为  $T$  的休眠计时器,建立多重同步休假随机模型,刻画节能策略的随机行为。运用拟生灭过程及矩阵几何解方法进行理论分析,基于蒙特卡罗方法进行系统仿真,在不同云服务参数下定量分析休眠参数与唤醒阈值对云系统节能水平及云用户请求平均响应时间的影响。基于不同性能指标之间的折衷关系,建立系统成本函数,改进飞蛾扑火优化(MFO)算法,给出节能策略的联合优化方案。

**关键词** 云计算; 休眠机制; 唤醒阈值; 休眠计时器; 多重同步休眠; 飞蛾扑火优化(MFO)算法

## 0 引言

随着云计算应用的普及与云服务规模的扩大,云数据中心在面向各界提供高性能服务的同时,产生了巨大的能量消耗<sup>[1,2]</sup>。如何在保证云系统节能效果的前提下,提高云用户请求响应性能成为相关领域的研究热点。

针对云计算数据中心的高能耗问题,部分学者使用虚拟机整合技术对节能策略进行了相关研究。Fayyaz 等人<sup>[3]</sup>为了降低云数据中心的总体能耗,提出基于虚拟机/任务整合的云基础架构,通过虚拟机监视器将任务从未充分利用的服务器迁移到合适的服务器,使用动态阈值电压调节算法关闭空闲服务器。Zhou 等人<sup>[4]</sup>提出一种虚拟机整合算法,针对不同类型的主机,进行基于主机负载预测的虚拟机迁

移,研究云数据中心的空闲能耗问题。Tang 等人<sup>[5]</sup>为了降低云数据中心服务器、通讯网络及冷却系统的总体能耗,提出基于工作负载预测的节能作业调度算法,使用虚拟机迁移及整合技术,关闭相应模块及机架。

还有部分学者基于休眠模式研究了云数据中心的节能问题。Fan 等人<sup>[6]</sup>为了降低云数据中心的总体能耗,提出基于多种休眠模式的服务器调度方案。使用绿色休眠管理器制定服务器活跃模式及多睡眠模式的时间表,确保有足够的服务器以最低的能耗为云请求提供服务。Jin 等人<sup>[7]</sup>为了提高云数据中心的能源效率,提出具有速度切换及多重休眠模式的节能策略。基于当前的工作负载,确定主虚拟机的运行速度及备用虚拟机的状态。为了实现系统节能,Hou 等人<sup>[8]</sup>提出基于多种休眠模式的嵌入式系统的节能优化算法。在估计空闲持续时间长度之

① 国家自然科学基金(61872311,61973261)和河北省自然科学基金(F2017203141)资助项目。

② 女,1993 年生,硕士生;研究方向:计算机通信网络的系统建模与性能分析;E-mail: 18713501880@163.com

③ 通信作者,E-mail: jsf@ysu.edu.cn

(收稿日期:2019-10-12)

后,制定最优控制算法选择合适的休眠模式。在上述文献的休眠模式中,只有休眠定时器超时,系统才能从休眠期切换至唤醒期,而不考虑系统中的云用户请求数量,在一定程度上影响了云用户请求的响应性能。

本文兼顾云数据中心的能量效率与云用户请求的响应性能,面向多虚拟机引入( $N, T$ )休眠机制,研究了云系统的节能策略与优化问题。

## 1 能量节省策略及系统模型

### 1.1 能量节省策略

在传统的云数据中心中,除了处理云用户请求需产生的必要能耗外,云系统在空闲时也会产生额外的能耗,造成能源浪费。为了在保证云用户服务质量的前提下,最大限度地提高云数据中心的能量效率,本文面向多虚拟机提出一种带有( $N, T$ )休眠机制的节能策略。该策略的工作原理如图1所示。

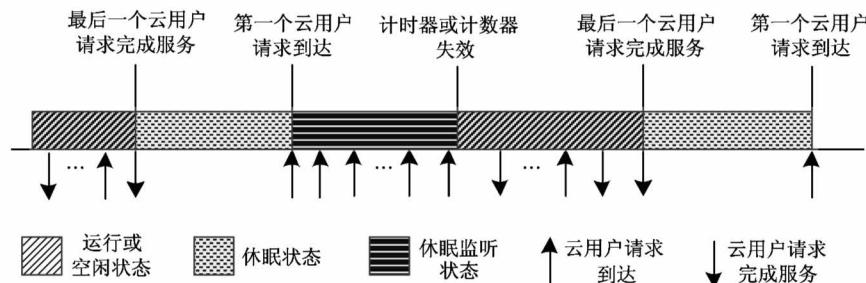


图1 ( $N, T$ ) 休眠机制的工作原理

### 1.2 系统模型

将云用户请求抽象为顾客,将虚拟机抽象为服务台,将休眠状态抽象为休假,基于( $N, T$ )休眠机制可建立多重同步休假随机模型。

假设所有云用户请求是同质的,所有虚拟机是同构的。云用户请求到达间隔服从参数为 $\lambda$ ( $\lambda > 0$ )的指数分布,服务一个云用户请求的时间服从参数为 $\mu$ ( $\mu > 0$ )的指数分布,休眠定时器的长度 $T$ 服从参数为 $\theta$ ( $\theta > 0$ )的指数分布。

令随机变量 $L(t)$ 表示 $t$ 时刻系统中云用户请求的数量,称为系统水平。令随机变量 $J(t)$ 表示云系统所处的状态。 $J(t) = 0$ 表示云系统处于节能状态,虚拟机或者休眠或者休眠监听。 $J(t) = 1$ 表示云系统处于唤醒状态,虚拟机或者空闲或者运行。 $\{(L(t), J(t)), t \geq 0\}$ 构成一个2维连续时间马尔可夫链,其状态空间 $\Omega$ 表示为

$$\Omega = \{(i, j) | i = 0, 1, 2, \dots, j = 0, 1\} \quad (1)$$

令 $\pi_{i,j}$ 表示稳态下云系统水平为 $i$ 且云系统处于状态 $j$ 的概率分布,则有:

$$\pi_{i,j} = \lim_{t \rightarrow \infty} P\{L(t) = i, J(t) = j\} \quad i = 0, 1, 2, \dots, j = 0, 1 \quad (2)$$

令 $\pi_i$ ( $i \geq 0$ )表示稳态下云系统水平为 $i$ 的概率向量, $\pi_i = (\pi_{i0}, \pi_{i1}, \pi_{i2}, \dots)$ 。则该马尔可夫链的稳态概率分布 $\pi$ 由 $\pi_i$ ( $i \geq 0$ )构成, $\pi$ 表示为 $\pi = (\pi_0, \pi_1, \pi_2, \dots)$ 。

## 2 模型分析和性能指标

### 2.1 模型分析

令 $\mathbf{Q}$ 表示连续时间马尔可夫链 $\{(L(t), J(t)), t \geq 0\}$ 的一步状态转移率矩阵, $Q_{x,y}$ 表示系统从 $x$ ( $x = 0, 1, 2, \dots$ )水平跳转至 $y$ ( $y = 0, 1, 2, \dots$ )

…水平的转移率子阵。在第2节给出的假设条件下,马尔可夫链中 $L(t)$ 的跳转只能发生在相邻水平之间。为表示方便,用 $\mathbf{B}_x$ 代替系统水平下降的子阵 $\mathbf{Q}_{x,x-1}$ ,用 $\mathbf{A}_x$ 代替系统水平不变的子阵 $\mathbf{Q}_{x,x}$ ,用 $\mathbf{C}_x$ 代替系统水平上升的子阵 $\mathbf{Q}_{x,x+1}$ 。

在多重同步休眠机制中,只有在虚拟机处于休眠状态时到达一个云用户请求后,才启动阈值为 $N$ 的计数器,故设置唤醒阈值下限 $N_{\min} = 2$ 个;为了保证所有的云用户请求都能在虚拟机唤醒以后立即被服务,故设置唤醒阈值上限 $N_{\max} = c$ 个。

考虑到唤醒阈值 $N$ 不大于系统中的虚拟机数量 $c$ ,分 $\mathbf{B}_x$ 、 $\mathbf{A}_x$ 及 $\mathbf{C}_x$ ,并给出一步状态转移率矩阵 $\mathbf{Q}$ 。

首先,根据跳转前不同的系统水平讨论一步状态转移率子阵 $\mathbf{B}_x$ 。

跳转前系统水平 $x = 0$ 时,全部虚拟机处于休眠状态,不存在一步状态转移率子阵 $\mathbf{B}_0$ 。

跳转前系统水平 $x = 1$ 时,只有1台虚拟机处于运行状态,其余的虚拟机全部处于空闲状态。处于运行状态的虚拟机完成当前云用户请求后,全部虚拟机由运行状态或空闲状态同步切换至休眠状态。

因此,一步状态转移率子阵 $\mathbf{B}_1 = \begin{pmatrix} 0 & 0 \\ \mu & 0 \end{pmatrix}$ 。

跳转前系统水平 $x \geq 2$ 时,部分虚拟机处于运行状态或全部虚拟机均处于运行状态。若跳转前系统水平 $2 \leq x \leq c$ ,有 $x$ 台虚拟机处于运行状态, $c - x$ 台虚拟机处于空闲状态,服务能力为 $x\mu$ ,一个云用户完成服务离开系统后,系统水平下降至 $x - 1$ ,当前虚拟机由运行状态切换至空闲状态;若跳转前系统水平 $x > c$ ,全部虚拟机均处于运行状态,服务能力为 $c\mu$ ,一个云用户完成服务离开系统后,系统水平下降至 $x - 1$ ,全部虚拟机仍保持在运行状态。因此,一步状态转移率子阵 $\mathbf{B}_x = \begin{pmatrix} 0 & 0 \\ 0 & \min(x\mu, c\mu) \end{pmatrix}$ 。 $x \geq c$ 时,一步状态转移率子阵 $\mathbf{B}_x$ 重复出现。将此重复的一步状态转移率子阵表示为 $\mathbf{B}$ , $\mathbf{B} = \begin{pmatrix} 0 & 0 \\ 0 & c\mu \end{pmatrix}$ 。

其次,根据跳转前不同的系统水平讨论一步状

态转移率子阵 $\mathbf{A}_x$ 。

跳转前系统水平 $x = 0$ 时,全部虚拟机处于休眠状态。若没有云用户请求到达,系统水平不变,且虚拟机仍保持在休眠状态。因此,一步状态转移率子阵 $\mathbf{A}_0 = \begin{pmatrix} -\lambda & 0 \\ 0 & 0 \end{pmatrix}$ 。

跳转前系统水平 $1 \leq x \leq N - 1$ 时,在系统水平不变的情况下,仅能有长度为 $T$ 的休眠计时器超时,休眠参数为 $\theta$ 。若休眠计时器超时,全部虚拟机由休眠监听状态同步切换至空闲状态或运行状态。因此,一步状态转移率子阵 $\mathbf{A}_x = \begin{pmatrix} -(\lambda + \theta) & \theta \\ 0 & -(\lambda + x\mu) \end{pmatrix}$ 。

跳转前系统水平 $x \geq N$ 时,部分虚拟机处于运行状态或全部虚拟机均处于运行状态。若没有云用户请求到达,也没有云用户请求离开,系统水平不变,且虚拟机仍保持在运行状态。因此,一步状态转移率子阵 $\mathbf{A}_x = \begin{pmatrix} 0 & 0 \\ 0 & -(\lambda + \min(x\mu, c\mu)) \end{pmatrix}$ 。 $x \geq c$ 时,一步状态转移率子阵 $\mathbf{A}_x$ 重复出现。将此重复的一步状态转移率子阵表示为 $\mathbf{A}$ , $\mathbf{A} = \begin{pmatrix} 0 & 0 \\ 0 & -(\lambda + c\mu) \end{pmatrix}$ 。

最后,根据跳转前不同的系统水平讨论一步状态转移率子阵 $\mathbf{C}_x$ 。

跳转前系统水平 $x = 0$ 时,全部虚拟机处于休眠状态。到达一个云用户请求后,系统水平上升至1,且全部虚拟机由休眠状态同步切换至休眠监听状态。因此,一步状态转移率子阵 $\mathbf{C}_0 = \begin{pmatrix} \lambda & 0 \\ 0 & 0 \end{pmatrix}$ 。

跳转前系统水平 $1 \leq x < N - 1$ 时,有可能全部虚拟机均处于休眠监听状态;也有可能部分虚拟机处于空闲状态,而另一部分虚拟机处于运行状态。此时,到达一个云用户请求,系统水平上升至 $x + 1$ 。当全部虚拟机均处于休眠监听状态时,虚拟机仍处于休眠监听状态;当部分虚拟机处于运行状态,而另一部分虚拟机处于空闲状态时,一个虚拟机切换至运行状态。因此,一步状态转移率子阵 $\mathbf{C}_x = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$ 。

跳转前系统水平  $x = N - 1$  时, 有可能全部虚拟机处于休眠监听状态; 也有可能虚拟机部分处于运行状态, 而另一部分虚拟机处于运行状态。此时, 到达一个云用户请求, 系统水平上升至  $N$ 。当全部虚拟机均处于休眠监听状态时, 全部虚拟机由休眠监听状态同步切换至空闲状态或运行状态; 当部分虚拟机处于运行状态, 而另一部分虚拟机处于运行状态时, 处于空闲状态的一个虚拟机切换至运行状态。

因此, 一步状态转移率子阵  $\mathbf{C}_{N-1} = \begin{pmatrix} 0 & \lambda \\ 0 & \lambda \end{pmatrix}$ 。

跳转前系统水平  $x \geq N$  时, 部分虚拟机处于运行状态或全部虚拟机均处于运行状态。到达一个云用户请求后, 系统水平上升至  $x + 1$ 。当部分虚拟机处于运行状态时, 处于空闲状态的一个虚拟机切换至运行状态; 当全部虚拟机均处于运行状态时, 全部虚拟机仍保持在运行状态。因此, 一步状态转移率子阵  $\mathbf{C}_x = \begin{pmatrix} 0 & 0 \\ 0 & \lambda \end{pmatrix}$ 。 $x \geq N$  时, 一步状态转移率子阵  $\mathbf{C}_x$  重复出现。将此重复的一步状态转移率子阵表示为  $\mathbf{C}, \mathbf{C} = \begin{pmatrix} 0 & 0 \\ 0 & \lambda \end{pmatrix}$ 。

按照系统水平从低到高的顺序, 系统一步状态转移率矩阵  $\mathbf{Q}$  写为

$\mathbf{Q} =$

$$\left\{ \begin{array}{c} \mathbf{A}_0 \quad \mathbf{C}_0 \\ \mathbf{B}_1 \quad \mathbf{A}_1 \quad \mathbf{C}_1 \\ \ddots \quad \ddots \quad \ddots \\ \mathbf{B}_{N-1} \quad \mathbf{A}_{N-1} \quad \mathbf{C}_{N-1} \\ \mathbf{B}_N \quad \mathbf{A}_N \quad \mathbf{C} \\ \ddots \quad \ddots \quad \ddots \\ \mathbf{B}_{c-1} \quad \mathbf{A}_{c-1} \quad \mathbf{C} \\ \mathbf{B} \quad \mathbf{A} \quad \mathbf{C} \\ \ddots \quad \ddots \quad \ddots \end{array} \right\} \quad (3)$$

由一步状态转移率子阵  $\mathbf{Q}$  的结构可知, 马尔可夫链  $\{(L(t), J(t)), t \geq 0\}$  是一种拟生灭过程, 该过程正常返的充分必要条件是矩阵二次方程  $\mathbf{R}^2 \mathbf{B} + \mathbf{R} \mathbf{A} + \mathbf{C} = 0$  的最小非负解  $\mathbf{R}$  (称为率阵  $\mathbf{R}$ ) 的谱半径  $SP(\mathbf{R}) < 1$ 。

转移率子阵  $\mathbf{A}, \mathbf{B}$  及  $\mathbf{C}$  都是右下角只有一个非零元素的方阵, 因此, 率阵  $\mathbf{R}$  也必然有相同的结构。设

$$\mathbf{R} = \begin{pmatrix} 0 & 0 \\ 0 & r_{22} \end{pmatrix} \quad (4)$$

其中,  $r_{22}$  是实数。

求解方程  $r_{22}^2 - r_{22}(\lambda + c\mu) + \lambda = 0$ , 得最小非负解  $r_{22} = \frac{\lambda}{c\mu} = \rho < 1$ 。

构造方阵  $\mathbf{B}[\mathbf{R}]$  如下:

$$\left\{ \begin{array}{c} \mathbf{A}_0 \quad \mathbf{C}_0 \\ \mathbf{B}_1 \quad \mathbf{A}_1 \quad \mathbf{C}_1 \\ \ddots \quad \ddots \quad \ddots \\ \mathbf{B}_{N-1} \quad \mathbf{A}_{N-1} \quad \mathbf{C}_{N-1} \\ \mathbf{B}_N \quad \mathbf{A}_N \quad \mathbf{C}_N \\ \ddots \quad \ddots \quad \ddots \\ \mathbf{B}_{c-1} \quad \mathbf{A}_{c-1} \quad \mathbf{C}_{c-1} \\ \mathbf{B} \quad \mathbf{A} + \mathbf{R}\mathbf{B} \end{array} \right\} \quad (5)$$

基于平衡方程及正规化条件, 建立方程组:

$$\left\{ \begin{array}{l} (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots, \boldsymbol{\pi}_c) \mathbf{B}[\mathbf{R}] = 0 \\ (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots, \boldsymbol{\pi}_{c-1}) \mathbf{e} + \boldsymbol{\pi}_c (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e}_1 = 1 \end{array} \right. \quad (6)$$

其中,  $\mathbf{e}$  是维数为  $2 \times c$  的全 1 列向量,  $\mathbf{e}_1$  是维数为 2 的全 1 列向量。

利用高斯-赛德尔法<sup>[9]</sup> (Gauss-Seidel method) 求解方程组式(6), 可得  $\boldsymbol{\pi}_i (i = 0, 1, \dots, c)$  的数值解。结合矩阵几何解方法, 给出  $\boldsymbol{\pi}_i = \boldsymbol{\pi}_c \mathbf{R}^{i-c}$ ,  $i > c$ , 可进一步求得  $\boldsymbol{\pi}_i (i = c+1, c+2, \dots)$  的数值解。

## 2.2 性能指标

云用户请求数量定义为从云用户请求到达云系统开始到服务完成为止的时间长度<sup>[10]</sup>。一个云用户请求数量包含在系统缓冲区中的等待时间及在虚拟机上的服务时间。运用 Little 公式<sup>[11]</sup>, 云用户请求数量平均响应时间  $D$  的表达式为

$$D = \frac{1}{\lambda} \left( \sum_{i=0}^{\infty} i (\boldsymbol{\pi}_{i,0} + \boldsymbol{\pi}_{i,1}) \right) \quad (7)$$

云系统节能水平定义为单位时间内云系统因休

眠机制的引入而减少的能量消耗。云系统处于唤醒状态时消耗正常的能量,处于节能状态时节省能耗;由节能状态切换至唤醒状态的过程会增加额外的能耗。云系统节能水平 $S$ 的表达式如下:

$$S = \sum_{i=0}^{N-1} \pi_{i,0} \times (w_h - w_s) \\ - \left( \sum_{i=1}^{N-1} \pi_{i,0} \times \theta + \pi_{N-1,0} \times \lambda \right) \times w_a \quad (8)$$

其中,  $w_h$  表示云系统处于唤醒状态时单位时间内的能量消耗,  $w_s$  表示云系统处于节能状态时单位时间内的能量消耗,  $w_a$  表示云系统由节能状态切换至唤醒状态的能量消耗。

### 3 系统实验

为了在不同服务率 $\mu$ 下进一步研究唤醒阈值 $N$ 及休眠参数 $\theta$ 对虚拟机调度策略的影响,进行数值实验及仿真实验。数值实验基于式(7)和式(8)用Matlab R2016a进行;仿真实验基于( $N, T$ )休眠机制采用Java语言在MyEclipse2014下进行。

在云系统稳态条件 $\frac{\lambda}{c\mu} < 1$ 下,设定虚拟机数量 $c = 20$ 台,云用户请求到达率 $\lambda = 3$ 个/ms,服务率 $\mu$ 的变化范围为 $1.3 \sim 2.3$ 个/ms。假设云系统能量消耗与虚拟机服务能力成正比,云系统处于唤醒状态时单位时间的能量消耗 $w_h = 10.0 \times \mu$ mW,云系统处于节能状态时单位时间的能量消耗 $w_s = 0.5 \times \mu$ mW,云系统由节能状态切换至唤醒状态的能量消耗 $w_a = 0.2$ mJ/次。

图2刻画了不同唤醒阈值 $N$ 及服务率 $\mu$ 下云用户请求平均响应时间 $D$ 随休眠参数 $\theta$ 的变化趋势。

由图2可知,若服务率 $\mu$ 及唤醒阈值 $N$ 固定,随着休眠参数 $\theta$ 的增大,云用户请求平均响应时间 $D$ 整体呈下降趋势。休眠参数越大,休眠定时器长度越短,云用户请求在系统节能状态的滞留时间越少,因而,云用户请求平均响应时间减小。特别地,当休眠参数 $\theta = 0.0$ 时,系统退化为纯 $N$ 策略,云用户请求平均响应时间较长;当唤醒阈值 $N = \infty$ 时,系统退化为纯 $T$ 策略,云用户请求平均响应时间也较长。与传统的纯 $N$ 策略及纯 $T$ 策略相比,( $N, T$ )策略下

的云系统休眠机制改善了云系统的响应性能。

从图2可以看出,当 $\mu = 1.3, \theta < 4.0$ 时,云用户请求平均响应时间 $D$ 随着唤醒阈值 $N$ 的增大呈现出增加的趋势;当 $\mu = 1.3, \theta > 4.0$ 时,唤醒阈值 $N$ 对云用户请求平均响应时间 $D$ 无影响。

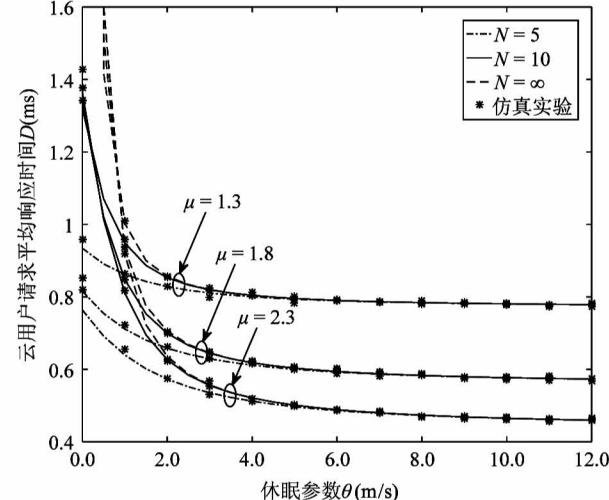


图2 休眠参数对云用户请求平均响应时间的影响

为了准确了解唤醒阈值 $N$ 对云用户请求平均响应时间 $D$ 的影响,图3刻画了不同休眠参数 $\theta$ 及服务率 $\mu$ 下云用户请求平均响应时间 $D$ 随唤醒阈值 $N$ 的变化趋势。

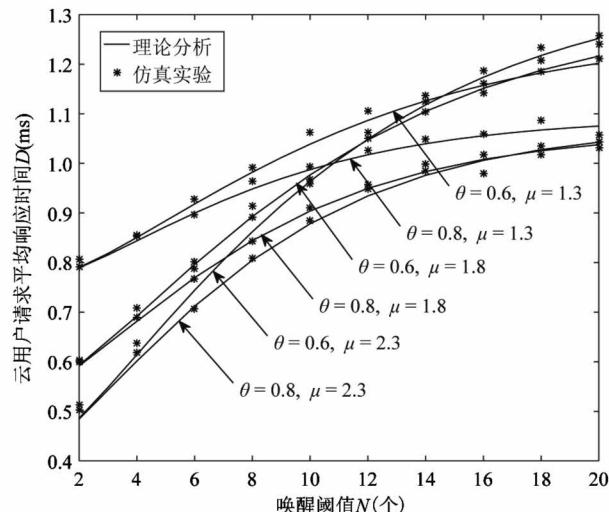


图3 唤醒阈值对云用户请求平均响应时间的影响

由图3可知,若服务率 $\mu$ 及休眠参数 $\theta$ 固定,随着唤醒阈值 $N$ 的增大,云用户请求平均响应时间 $D$

整体呈增加趋势。唤醒阈值越大,节能状态下云用户请求数达到唤醒阈值所需的时间越长,因而云用户请求平均响应时间增加。

图 4 刻画了不同唤醒阈值  $N$  及服务率  $\mu$  下云系统节能水平  $S$  随休眠参数  $\theta$  的变化趋势。

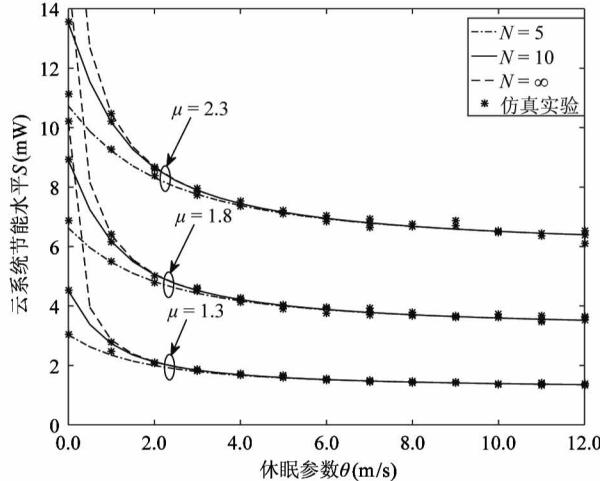


图 4 休眠参数对云系统节能水平的影响

由图 4 可知,若服务率  $\mu$  及唤醒阈值  $N$  固定,随着休眠参数  $\theta$  的增大,云系统节能水平  $S$  整体呈下降趋势。休眠参数越大,休眠定时器长度越短,云用户请求在系统节能状态的滞留时间越少。由于云系统处于节能状态时单位时间内的能量消耗低于系统唤醒状态时的能量消耗,因而,云系统节能水平降低。特别地,当休眠参数  $\theta = 0.0$  时,系统退化为纯  $N$  策略,云系统节能水平较高;当唤醒阈值  $N = \infty$  时,系统退化为纯  $T$  策略,云系统节能水平也较高。 $(N, T)$  策略下云系统休眠机制响应性能的改善是以牺牲一定的云系统节能水平为代价的。

图 4 可以看出,当  $\mu = 1.3, \theta < 3.0$  时,云系统节能水平  $S$  随着唤醒阈值  $N$  的增大呈现出增加的趋势;当  $\mu = 1.3, \theta > 3.0$  时,唤醒阈值  $N$  对云系统节能水平  $S$  无影响。

为了准确了解唤醒阈值  $N$  对云系统节能水平  $S$  的影响,图 5 刻画了不同休眠参数  $\theta$  及服务率  $\mu$  下云系统节能水平  $S$  随唤醒阈值  $N$  的变化趋势。

由图 5 可知,若服务率  $\mu$  及休眠参数  $\theta$  固定,随着唤醒阈值  $N$  的增大,云系统节能水平  $S$  整体呈增加的趋势。唤醒阈值越大,系统节能状态内云用户

请求数达到唤醒阈值所需的时间越长,即系统处于节能状态的时间越长,因而,云系统节能水平提高。

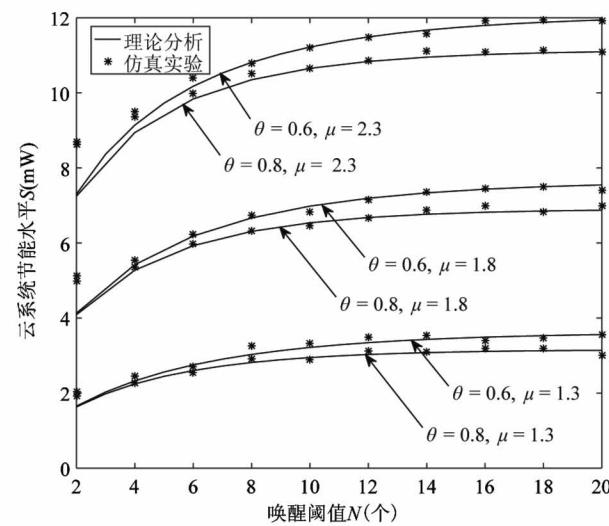


图 5 唤醒阈值对云系统节能水平的影响

综合图 2 和图 3 揭示出的云用户请求平均响应时间的变化趋势与图 4 和图 5 揭示出的云系统节能水平的变化趋势,可以看出在设置休眠参数  $\theta$  及唤醒阈值  $N$  时,云用户响应性能及云系统节能效果之间存在折衷关系。过大的休眠参数  $\theta$  可以降低云用户响应时间,但降低了云系统节能水平。过大的  $N$  可以提高云系统节能水平,但又影响了云用户响应性能。为此,面向  $(N, T)$  休眠机制的虚拟机节能策略,需要联合优化休眠参数  $\theta$  及唤醒阈值  $N$ 。

#### 4 系统优化

为了寻找云用户响应时间  $D$  及云系统节能水平  $S$  之间的平衡点,需要联合优化休眠参数  $\theta$  及唤醒阈值  $N$ 。

由图 2~图 5 的实验结果,给出不同服务率  $\mu$  下云用户请求平均响应时间的最大值  $D_{\max}$  及云系统节能水平最大值  $S_{\max}$ 。无量纲化云用户请求平均响应时间与云系统节能水平,采用权重算法<sup>[12]</sup>,建立系统成本函数  $H$  为

$$H = f_1 \times \frac{D}{D_{\max}} - f_2 \times \frac{S}{S_{\max}} \quad (9)$$

其中,  $f_1$  与  $f_2$  表示权重系数,且  $f_1 + f_2 = 1$ 。

由2.2节的理论分析可知,云用户请求平均响应时间 $D$ 与云系统节能水平 $S$ 均很难用休眠参数 $\theta$ 与唤醒阈值 $N$ 显示表示,系统成本函数 $H$ 的封闭解也很难求出。因此,利用传统的理论分析方法难以给出系统成本函数 $H$ 的优化结果。

群体智能优化算法具有操作简单、不受目标函数形式影响的特点,但容易陷入局部最优。本文以提高全局搜索能力为目的,综合使用对数螺旋函数及正弦函数更新飞蛾的当前位置,改进飞蛾扑火优化(moth-flame optimization, MFO)算法<sup>[13]</sup>,用于优化( $N, T$ )休眠机制的系统参数。该算法的主要步骤如算法1所示。

### 算法1 ( $N, T$ )休眠机制系统参数的优化算法

**Input:** 种群数量 $n$ ,最大迭代次数 $I_{\max}$ ,唤醒阈值下限

$N_{\min}$ 及上限 $N_{\max}$ ,休眠参数下限 $\theta_{\min}$ 及上限 $\theta_{\max}$

**Output:** 最小成本 $H^*$ ,最优参数组合 $(N^*, \theta^*)$

```

1:   for  $N = N_{\min} : N_{\max}$ 
2:     构建  $n \times 3$  二维表  $M$ 
3:      $M(i,1) = N, i \in \{1, 2, \dots, n\}$ 
4:      $M(i,2) = \text{rand} \cdot (\theta_{\max} - \theta_{\min}) + \theta_{\min}, i \in \{1, 2, \dots, n\}$  % 初始化休眠参数
5:      $M(i,3) = H(M(i,1), M(i,2)), i \in \{1, 2, \dots, n\}$  % 由式(4)计算系统成本函数
6:     创建与表  $M$  内容相同的表  $F$ ,并对表  $F$  按最后一列值升序排序,构造表  $X$  存放表  $F$  第一行的内容
7:      $Iter = 1$ 
8:      $f = \text{round}\left(n - \frac{Iter}{I_{\max}}(n - 1)\right)$ 
9:     for  $Iter = 2 : I_{\max}$ 
10:     $M(i,1) = N, i \in \{1, 2, \dots, n\}$ 
11:     $t = -\left(2 + \frac{Iter}{I_{\max}}\right) \cdot \text{rand} + 1$  %  $t$  为  $[-2, 1]$ 
之间的随机数
12:     $D_i = |(M(i,1), M(i,2)) - (F(i,1), F(i,2))|, i \in \{1, 2, \dots, f\}$  % 计算表  $M$  中前  $f$  个系统参数与表  $F$  中前  $f$  个系统参数之间的欧氏距离
13:     $M(i,2) = D_i \cdot e^{bt} \cdot \sin(2\pi t) + F(i,2), i \in \{1, 2, \dots, f\}$  % 利用螺旋函数及正弦函数更新表  $M$  中前  $f$  个休眠参数
14:     $D_i = |(M(i,1), M(i,2)) - (F(f,1), F(f,2))|, i \in \{f+1, \dots, n\}$  % 计算表  $M$  中后  $(n-f)$  个系统参数与表  $F$  中最后一行系统参数之间的欧氏距离

```

```

15:            $M(i,2) = D_i \cdot e^{bt} \cdot \sin(2\pi t) + F(f, 2), i \in \{f+1, \dots, n\}$  % 利用螺旋函数及正弦函数更新表  $M$  中后  $(n-f)$  个休眠参数
16:            $M(i,3) = H(M(i,1), M(i,2)), i \in \{1, 2, \dots, n\}$ 
17:           结合表  $M$  与表  $F$ ,构造表  $R$ ,并对表  $R$  按最后一列值升序排序
18:            $f = \text{round}\left(n - \frac{Iter}{I_{\max}}(n - 1)\right)$ 
19:           清空表  $F$ ,抽取表  $R$  中前  $f$  行的内容,放入表  $F$  中
20:           endfor
21:           if  $F(1,3) < X(1,3)$ 
22:              $X(1,1) = F(1,1)$ 
23:              $X(1,2) = F(1,2)$ 
24:              $X(1,3) = F(1,3)$ 
25:           endif
26:           endfor
27:            $H^* = X(1,3)$  % 输出成本最小值  $H^*$ 
28:            $(N^*, \theta^*) = (X(1,1), X(1,2))$  % 输出最优参数组合  $(N^*, \theta^*)$ 

```

利用改进的飞蛾扑火优化算法进行实验。实验过程中的休眠参数范围为  $[\theta_{\min}, \theta_{\max}]$ 。考虑休眠定时器长度为无穷大,设置休眠参数下限为  $\theta_{\min} = 0.0$  m/s。由图2与图4可知,当休眠参数  $\theta > 11.0$  m/s时,随着休眠参数  $\theta$  的增大,云用户请求平均响应时间及云系统节能水平的变化很小,因此,设置休眠参数上限为  $\theta_{\max} = 12.0$  m/s。算法所需其他实验参数设置如表1所示。

表1 实验参数设置

参数名称	参数取值
虚拟机数量 $c$ (台)	20
到达率 $\lambda$ (个/ms)	3
唤醒阈值下限 $N_{\min}$ (个)	2
唤醒阈值上限 $N_{\max}$ (个)	20
权重系数 $f_1$	0.7
权重系数 $f_2$	0.3
种群数量 $n$ (个)	30
最大迭代次数 $I_{\max}$ (次)	500

将上述实验参数代入算法 1 给出的算法中, 在不同服务率  $\mu$  下寻找出最小系统成本  $H^*$ , 并给出唤醒阈值  $N$  及休眠参数  $\theta$  的最优组合  $(N^*, \theta^*)$ 。系统优化数值结果如表 2 所示。

表 2 给出的系统优化数值结果是在表 1 所设权重系数下得到的。在实际应用中, 对于实时性要求高的云系统, 通常设置较大的权重系数  $f_1$ , 而对于追求高节能效果的云系统, 则应设置较小的权重系数  $f_2$ 。

表 2 系统优化数值结果

服务率 $\mu$ (个/ms)	唤醒阈值及休眠 参数的最优组合 $(N^*, \theta^*)$	最小系统成本 $H^*$
1.3	(5, 0.002)	0.1293
1.4	(5, 0.001)	0.1054
1.5	(4, 0.002)	0.0839
1.6	(4, 0.002)	0.0652
1.7	(4, 0.001)	0.0490
1.8	(4, 0.002)	0.0347
1.9	(4, 0.002)	0.0221
2.0	(3, 0.001)	0.0100
2.1	(3, 0.002)	-0.0010
2.2	(3, 0.001)	-0.0110
2.3	(3, 0.002)	-0.0201

## 5 结 论

折衷考虑云数据中心的节能水平与云用户请求的响应性能, 提出了基于  $(N, T)$  休眠机制的云计算中心节能策略。结合唤醒阈值及休眠定时器, 建立了一种多重同步休假随机模型, 并给出了 2 维马尔可夫链的一步状态转移率矩阵。利用矩阵几何解方法, 给出了云系统节能水平与云用户请求平均响应时间等性能指标。在 Matlab R2016a 平台下基于性能指标的理论分析进行了数值实验, 在 Myeclipse2014 平台下采用 Java 语言进行了仿真实验。实验结果表明, 较大的休眠参数可以减少云用户响应时间, 但降低了云系统节能水平; 较大的唤醒阈值可以提高云系统节能水平, 但又影响了云用户响应性能。权衡不同性能指标, 建立系统成本函数, 通过

改进飞蛾扑火优化算法, 给出了云数据中心节能策略的联合优化结果。

在下一步的工作中, 将引入延迟休眠机制, 进一步研究响应性能更高的云计算中心的节能策略。

## 参 考 文 献

- [1] Gourisaria M K, Patra S S, Khilar P K. Energy saving task consolidation technique in cloud centers with resource utilization threshold [C] // Proceedings of the 2016 International Conference on Automatic Control and Information Engineering, Hong Kong, China, 2016: 655-666
- [2] 金顺福, 郝闪闪, 王宝帅. 融合双速率和工作休眠的虚拟机调度策略及参数优化 [J]. 通信学报, 2017, 38(12): 10-20
- [3] Fayyaz A, Khan M U S, Khan S U. Energy efficient resource scheduling through VM consolidation in cloud computing [C] // Proceedings of the International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2015: 65-70
- [4] Zhou Z, Hu Z, Yu J, et al. Energy-efficient virtual machine consolidation algorithm in cloud data centers [J]. Journal of Central South University, 2017, 24(10): 127-137
- [5] Tang X, Liao Xi, Zheng J, et al. Energy efficient job scheduling with workload prediction on cloud data center [J]. Cluster Computing, 2018, 21(3): 1581-1593
- [6] Fan L, Gu C, Qiao L, et al. Green-sleep: a multi-sleep modes based scheduling of servers for cloud data center [C] // Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIG-COM), Chengdu, China, 2017: 368-375
- [7] Jin S, Hao S, Yue W. Energy efficient strategy with a speed switch and a multiple-sleep mode in cloud data centers [C] // Proceedings of the International Conference on Queueing Theory and Network Applications, Tsukuba, Japan, 2017: 143-154
- [8] Hou C, Zhao Q. A new optimal algorithm for energy saving in embedded system with multiple sleep modes [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2015, 24(2): 706-719
- [9] Edalatpour V, Hezari D, Salkuyeh D K. A generalization of the Gauss-seidel iteration method for solving absolute

- value equations [ J ]. *Applied Mathematics and Computation*, 2017, 293: 156-167
- [ 10 ] 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(11-12): 907-914
- [ 11 ] 孙健, 丁日佳, 陈艳艳. M/M/c型与M/M/1型排队系统对比仿真 [ J ]. 北京工业大学学报, 2016, 42(9) : 1324-1331
- [ 12 ] 周晓彦, 嵇福高, 刘文杰, 等. 一种酉权重量子感知机 [ J ]. 小型微型计算机系统, 2018, 39(3) : 534-538
- [ 13 ] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm [ J ]. *Knowledge-Based Systems*, 2015, 89: 228-249

## Energy saving strategy and optimization of cloud computing centers based on ( $N, T$ ) sleep mechanism

Wang Xiaochen \* \*\* , Wang Yuting \* \*\* , Zhang Liyuan \*\*\* , Jin Shunfu \* \*\*

( \* School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004)

( \*\* Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao 066004)

( \*\*\* Science School, Langfang Normal University, Langfang 065000)

### Abstract

With the increase in the variety and number of cloud applications, the demands of cloud users for response performance show more diverse and the requirements of cloud computing centers for energy conservation level present more higher. Under the premise of guaranteeing the energy saving effect of cloud computing centers and improving the response performance of cloud users, ( $N, T$ ) sleep mechanism is introduced to the cloud computing centers, and a virtual machine based energy saving strategy is proposed. Combining the wake-up threshold  $N$  and the sleep timer with time length  $T$ , a stochastic model with multiple synchronous vacations is constructed to characterize the stochastic behavior of the energy saving strategy. The stochastic model is theoretically analyzed by using the quasi birth-and-death process and the matrix-geometric solution, the sleep mechanism is simulated by the Monte Carlo method, and then, the influence of sleep parameters on the system energy conservation level and the average response time of cloud users are quantitatively evaluated for different service rates. Based on the trade-off between different performance measures, a system cost function is established, a type of the moth-flame optimization (MFO) algorithm is improved, and then, the energy saving strategy is jointly optimized.

**Key words:** cloud computing, sleep mechanism, wake-up threshold, sleep timer, multiple synchronous vacations, moth-flame optimization (MFO) algorithm