基于萤火虫群优化的虚拟机放置方法

徐胜超[1]，熊茂华[1]，周天绮[2]
（1. 广州华商学院数据科学学院，广东 广州 511300;
2. 浙江药科职业大学医疗器械学院，浙江 宁波 315100）

摘　要：利用虚拟机放置策略对云数据中心的物理资源利用效率进行优化十分必要。提出了基于萤火虫群优化的虚拟机放置（glowworm swarm optimization based VM placement，Gso-wmp）方法。GSO-VMP 方法将物理主机的处理器使用效率表示为荧光素值，当一个虚拟机被放置到物理主机上时，该物理主机的荧光素值都要进行更新；能够在局部径向范围内搜索到更多的可用物理主机，完成虚拟机放置，减少了虚拟机的迁移次数，从而间接地节省了物理主机的能量消耗。使用 CloudSim 作为 GSO-VMP 的仿真环境进行仿真，实验结果表明，GSO-VMP 方法使得云数据中心的能耗降低、多维物理资源利用率提高。

关键词：智能计算；萤火虫群优化；虚拟机放置；云数据中心；低能量消耗
中图分类号: TP393
文献标志码: A
doi: 10.11959/j.issn.1000−0801.2022060

Approach of glowworm swarm optimization based virtual machine placement

XU Shengchao[1], XIONG Maohua[1], ZHOU Tianqi[2]
1. School of Date Science, Guangzhou Huashang College, Guangzhou 511300, China
2. School of medical instruments, Zhejiang Pharmaceutical Vocational University, Ningbo 315100, China

Abstract: In a cloud data center, one of the most important problems is using novel virtual machine placement strategy to promote the physical resource utilization. An approach of glowworm swarm optimization based virtual machine placement for cloud data centers called GSO-VMP was proposed. In the virtual placement, GSO algorithm was used to find a near-optimal solution. Each physical host had a luciferin value which represented the available CPU utilization. Whenever a VM was placed to a physical host, luciferin value of this physical host was updated. GSO-VMP algorithm could search the more available physical host within local range and thus the virtual migration numbers had been decreased and low energy consumption had been obtained. GSO-VMP had been evaluated using CloudSim with real-world workload data. The experimental results show that GSO-VMP has good performance in resource wastage and energy consumption.

收稿日期：2021−08−25；修回日期：2022−01−24
通信作者：周天绮，zhoutianqi_2018@126.com
基金项目：广东省高等学校科学研究特色创新项目（No.2021KTSCX167）；广州华商学院校内导师制科研项目（No.2021HSDS15）
Foundation Items: Characteristic Innovation Project of Scientific Research in Colleges and Universities of Guangdong Province (No.2021KTSCX167), Science and Research Project in Supervisor of Guangzhou Huashang College (No.2021HSDS15)

Key words: intelligent computing, glowworm swarm optimization, virtual machine placement, cloud data center, low energy consumption

0　引言

低能量消耗与物理资源的充分利用是绿色云数据中心构造的两个主要目标[1-2]。目前，国内外的研究主要采用虚拟机迁移技术来达到这两个目标[3-5]。虚拟机放置及优化是虚拟机迁移中最重要的一个阶段，属于多目标优化问题或者装箱问题[6]，它没有最优解，只能在目标函数中得到一定程度的最优，目前针对虚拟机放置有很多智能优化算法，近年来的研究主要涉及两个方面的研究进展：能量消耗模型的改进和智能算法的改进。

在能量消耗模型改进方面，早期的文献在设计云数据中心物理主机的能量消耗模型时考虑的物理资源维度比较单一，例如单一物理资源[7]只考虑处理器利用率、处理器主频大小、处理器温度大小这一个维度[8]；后续考虑问题的维度扩展到了内存利用率、磁盘空间大小、I/O 通信能力等；近 3 年，还考虑了更多的因素，如网络带宽因素、如网络设备接口能量消耗等[9]。

在智能算法改进方面，近年来研究者提出了大量的新型智能算法进行虚拟机放置阶段的优化，例如有基于遗传算法[10]、贪心算法[11]、粒子群优化算法[12]、蚁群算法[13-14]、强化学习优化算法[15]、花授粉优化算法[16]、蛙跳算法[17]、稳定匹配[18]等的虚拟机放置策略。大部分优化策略的目的是在云数据中心的总体能量消耗、虚拟机迁移次数、SLA 违规率、活动物理主机数量等目标指标上取平衡。

其中，贪心算法优化方面常见的有首次递减适用算法[19]、最佳适用算法[20]、最佳递减适用算法等，但是由于虚拟机放置是多目标优化问题，贪心算法也不可能得到最优解；在遗传算法优化方面有常见的普通遗传算法、家族遗传算法[21]、混合遗传算法[6]、改进的遗传算法[22]等，但是由于它们采用约束编程的方式，限制了搜索空间。虽然，近年来蚁群算法优化的虚拟机放置在单维物理资源的高效利用方面取得了一定的成功，它们的实验结果表明性能优于首次递减适用的贪心算法，其不足是不能充分利用多个维度的物理资源。文献[23]采用蚁群算法优化，从处理器和内存大小等多个维度的信息来考虑云数据中心的虚拟机迁移，结果表明它的性能优于遗传算法。强化学习优化算法和花授粉优化算法是2019年被提出的虚拟机放置优化算法，在云数据中心的能耗与资源利用方面都性能良好，本文重点研究的也是虚拟机放置及优化策略。

由于萤火虫群优化算法在解决经典装箱问题上有很好的优势[24]，本文提出了基于萤火虫群优化的虚拟机放置（glowworm swarm optimization based VM placement，GSO-VMP）方法，通过这种智能优化算法，大量的虚拟机在云数据中心的物理主机上迁移、优化放置、负载均衡，以提高物理资源的使用效率。最后与近几年的常见虚拟机放置优化策略进行比较，在各个性能指标方面完成了 GSO-VMP 的测试与性能分析，得到了理想的实验结果。

1　萤火虫群优化算法

由于萤火虫群算法的控制中心不在一个单独的点上，所以该算法具有很好的扩展性。它模拟了萤火虫的行为，一个会产生更亮的光（高荧光素）的萤火虫意味着它更接近一个实际的位置，具有更加接近目标函数的值，在 GSO-VMP 中指的是云数据中心的总体能量消耗（目标函数）最低。

根据一定的概率机制，每个萤火虫都会选择一个比自己具有更高萤光素的邻居，并且向它移动，该移动过程是基于局部径向范围的。萤火虫群算法中，在开始的时候萤火虫在一个随机的位

置，所有的萤火虫都包括一个相同的荧光素。本文将萤火虫群算法应用到虚拟机放置这个过程中，使虚拟机选择和部署在不同的物理主机，最终目的是降低云数据中心的能量消耗和减少 SLA（service level agreement）违规比率。SLA 违规比率可以通过式（1）计算。

$$\text{SLA Violation} = \text{SLATAH} \times \text{PDM} \quad (1)$$

其中，SLATAH 是 SLA 违规的在线时间，PDM 是虚拟机迁移后的性能降低，这些在虚拟机放置的经典文献都有介绍[25]。

2 GSO-VMP 工作环境与相关术语

2.1 GSO-VMP 的工作环境

云计算环境下分配可用的资源给云客户端具有各种不同的模式，这些模式都可以被认为是针对云数据中心资源池的一种基于服务的访问，虚拟机技术通过将云客户端的请求封装成虚拟机的形式来分配与访问。图 1 是 GSO-VMP 虚拟机放置策略的资源分配模型，它由 3 个主要步骤组成。

步骤 1 云客户端提交请求。

步骤 2 服务提供者处理请求。

步骤 3 云数据中心资源分配与管理。

首先，云客户端提交请求到云服务提供者，代理将返回该结果到右边的云数据中心资源管理模块。云资源管理将查询该请求，与可用资源池的资源进行比较，并做出决策。云资源管理对云客户端请求的接受都基于平台的资源可用性，如果超过了资源的可用能力，云资源管理将会把该请求传递到资源分配模式，寻找全局可用资源，该资源分配方法被传递到资源管理模块，里面包含了 GSO-VMP 等虚拟机分配与优化的相关的操作。

2.2 GSO-VMP 的相关术语

2.2.1 云客户端请求模型

云客户端请求资源通过代理或者云服务提供者来请求各种应用。云客户端的请求可以定义为 VMs。定义用户的请求为 UR。UR 按照“先来先服务”的方式来访问云数据中心的物理资源。A_i 表示为虚拟机 VMs 的组成部分。虚拟机的组成部分包括：α_s^i 为处理器需求，β_s^i 为内存需求，γ_s^i 为磁盘空间需求。相应地，i 表示资源的数量，s 表示它们各自的资源提供能力。从数学上讲，可以把资源请求表示为 $A_i \subset \text{UR}$ 并且 $\alpha_s^1, \beta_s^1, \gamma_s^1 \subset A_i$，所以有 $\alpha_s^1, \beta_s^1, \gamma_s^1 \subset A_i \subset \text{UR}$，且 $\alpha_s^1, \beta_s^1, \gamma_s^1 \subset \text{UR}$。

所以，当云客户端只发送一个请求到一个资源上时，可以按照式（2）和式（3）来表达。

$$\text{UR}^1 = A_i \quad (2)$$

$$A_i = (\alpha_s^1 + \beta_s^1 + \gamma_s^1) \quad (3)$$

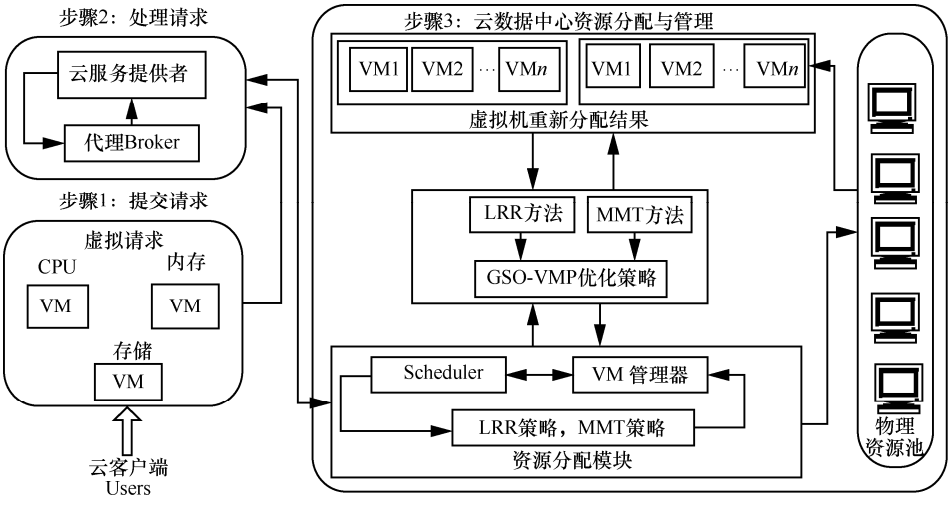


图 1 GSO-VMP 虚拟机放置策略的资源分配模型

当 $i=1$ 的时候，UR^1 表示资源的需求只有一个。

另外，如果客户端的请求数超过一个的时候，可以按照式（3）和式（4）来表达。

$$\begin{aligned}\mathrm{UR}^n &= \sum_{i=1}^{n} A_i = A_1 + A_2 + \cdots + A_n = (\alpha_s^1 + \beta_s^1 + \gamma_s^1) \\ &+ (\alpha_s^2 + \beta_s^2 + \gamma_s^2) + \cdots + (\alpha_s^n + \beta_s^n + \gamma_s^n)\end{aligned} \tag{4}$$

$$\mathrm{UR}^n = \sum_{i=1}^{n}(\alpha_s^i) + \sum_{i=1}^{n}(\beta_s^i) + \sum_{i=1}^{n}(\gamma_s^i) \tag{5}$$

2.2.2　云数据中心能量消耗模型

假设云数据中心中所有的虚拟机的集合为 $\mathrm{VM}=\{\mathrm{vm}_i, i=1,2,3,\cdots,n\}$，它们将要被分配到多个物理主机 PM_j 之上，$\mathrm{PM}=\{\mathrm{PM}_j,\ j=1,2,3,\ \cdots, m\}$。每个虚拟机的多维资源需求都按照 d 维的向量来表示[26]，$\mathrm{VM}_i=(A_{i,1}, A_{i,2},\cdots, A_{i,d})$。类似地，每个物理主机的多维资源提供能力也可以按照 d 维的向量来表示[27]，$\mathrm{PM}_j=(B_{j,1}, B_{j,2},\cdots, B_{j,d})$。本文描述的 GSO-VMP 策略中各种物理主机资源主要包括 CPU 资源、内存资源和磁盘空间。因此，$d=3$。

虚拟机的放置与优化都有开始时间和结束时间，每个虚拟机 VM_i 在一个固定的时间 t_1 开始，需要一个执行时间，因此整个虚拟机放置的时间跨度按照计算式表达为：$[t_1, t_2]$。$A_{i,s}$ 是虚拟机 VM_i 的资源请求向量，$B_{j,s}$ 是物理主机 PM_j 的资源能力提供向量，整体而言，云数据中心的资源分配问题有下列约束条件。

（1）有资源必须有能力提供给云客户端的请求；

（2）所有虚拟机的资源请求必须小于或等于整个物理主机的资源提供能力；

（3）假设 $a_j(t)$ 是被分配到 PM_j 上的一个虚拟机，则每个虚拟机只能分配到一个具体的物理主机上；

（4）对于 $(\forall s)=1,2,\cdots,d$：

$$\sum_{\mathrm{VM}_i \in a_j(t)} A_{i,s} \leqslant B_{i,s} \tag{6}$$

假设一个最优的虚拟机放置方式 R^A 为二元组 $(\mathrm{VM}_i, \mathrm{PM}_j), \forall i \in \{1,2,\cdots,n\}, \exists j \in \{1,2,\cdots,m\}$，

R^A 的目标函数是尽量提高整体物理资源的利用效率($\mathrm{RU}^{\mathrm{DC}}$)，同时降低整体云数据中心的能量消耗($\mathrm{DC}_U^{\mathrm{Energy}}$)。首先，必须最大化单台物理主机的资源利用效率 RU_j^d，可以按照式（7）描述。

$$\mathrm{RU}_j^d = \frac{\sum_{i=1}^{n} \rho_j^i \cdot \mathrm{VM}_i^d}{\mathrm{PM}_j^d}, \forall d \in \{\alpha_s^1, \beta_s^1, \gamma_s^1\} \tag{7}$$

则整个云数据中心的物理主机的资源利用效率如式（8）所示：

$$\mathrm{RU}^{\mathrm{DC}} = \int_{t_1}^{t_2} \frac{\sum_{j=1}^{m} \mathrm{RU}_j^d \alpha_s^1 + \sum_{j=1}^{m} \mathrm{RU}_j^d \beta_s^1 + \sum_{j=1}^{m} \mathrm{RU}_j^d \gamma_s^1}{|\mathrm{d}| \sum_{j=1}^{m} \rho_j^i} \mathrm{d}t, \sqrt{2} \tag{8}$$

其中，$\rho_j^i = \begin{Bmatrix} 1, \text{当UR 被指派给 } A_i \\ 0, \text{其他} \end{Bmatrix}$。

本文假设每个 PM_j 都可以容纳任何虚拟机，而且物理主机的能量消耗模型变量 $P(u(t)_j)$ 具有线性关系[28]，所以式（9）～式（10）描述了云数据中心的能量消耗模型的一些参数。

$$\mathrm{EC}_j = \int_{t_1}^{t_2} P(u(t)_j) \mathrm{d}t \tag{9}$$

$$\mathrm{DC}_U^{\mathrm{Energy}} = \sum_{j=1}^{m} \mathrm{EC}_j \tag{10}$$

其中，EC_j 是物理主机 PM_j 在 $[t_1, t_2]$ 内的能量消耗，$j \in \{1,2,\cdots,m\}$，$t \in [0, \mathrm{T}]$；$u(t)_j$ 表示在给定时刻 t 的 CPU 的利用效率，式（10）为整个云数据中心的能量消耗模型。

综上所述，整个云数据中心的资源利用效率 RU 的最大值 R^A 和整个云数据中心的能量消耗利用效率可以描述为式（11）～式（12），这也是下面萤火虫群虚拟机放置算法的目标函数。

$$\mathrm{Maximize} \Rightarrow \mathrm{RU}^{\mathrm{DC}}, \forall d \in \{\alpha_s^1, \beta_s^1, \gamma_s^1\} \tag{11}$$

$$\mathrm{s.t.Minimize} \Rightarrow \mathrm{DC}_U^{\mathrm{Energy}} \tag{12}$$

3 GSO-VMP 虚拟机放置策略

3.1 GSO-VMP 总体描述

下面分别具体描述 GSO-VMP 的 4 个阶段。

（1）萤火虫初始化阶段

初始种群由大量的萤火虫组成，这些被认为是虚拟机放置阶段的候选解决方法。一个萤火虫可以认为是一个元素向量，每个元素都表示一个虚拟机。元素的值采用被分配了虚拟机后物理主机的 ID 来表示。

初始种群的产生基于最小负载的物理主机，从物理主机状态检测阶段产生的候选迁移物理主机列表 HostsToMigrateList 中获取。

每个物理主机 j 有一个处理器的利用效率 $u_j(t)$、一个荧光素值 $\ell_j(t)$，$\ell_j(t)$ 的值通过式（13）来计算。荧光素值表示了可利用的物理主机的 CPU 使用效率。

$$\ell_j(t)=\mathrm{EC}_{\max_j}-\mathrm{EC}_j \tag{13}$$

其中，$\mathrm{EC}_{\max_j}$ 是物理主机 j 在完全满负载工作时的能源消耗，EC_j 是当前物理主机 j 通过 CPU 利用效率计算获得的能量消耗。

另外，物理主机 j 的局部径向范围 $\omega_d^j(t)$、物理主机 j 的局部径向范围的最大传感范围 ω_s、移动的步长尺寸 s、期待的邻居的数目 k_t、荧光素衰减系数 p $(0<p<1)$、荧光素增强系数 ω、相邻区域范围的变化率 θ 都进行了初始化。

（2）荧光素值更新阶段

最开始的时候，每个物理主机都有自己的荧光素值 $\ell_j(t)$，每个物理主机根据自己的处理器的利用效率 $u_j(t)$ 计算得到目标函数值 $\mathrm{DC}_U^{\mathrm{Energy}}=\sum_{j=1}^{m}\mathrm{EC}_j$，更新后的荧光素值 $\ell_j(t+1)$ 通过式（14）进行计算。

$$\ell_j(t+1)=(1-p)\ell_j(t)+\omega P\left(u_j(t+1)\right) \tag{14}$$

其中，$\ell_j(t)$ 是物理主机 j 在 t 时刻的荧光素值，p 是荧光素衰减系数（$0<p<1$），ω 是荧光素增强系数，$P(u_j(t+1))$ 是式（9）提到的在时刻 $t+1$ 的目标函数值。

因此，物理主机的荧光素值是根据目标函数值的变化而改变。无论什么时候当一个虚拟机被放置到物理主机上时，该物理主机的荧光素值都要进行更新。如果荧光素值比较高，意味着它可以更好地降低物理主机的能量消耗，荧光素值也随着时间的变化而衰减。

（3）萤火虫移动阶段

在移动阶段，物理主机 j 上的虚拟机在局部径向范围 $\omega_d^j(t)$ 内，朝着它 k 个邻居中的其中一个进行移动，移动方向是朝着比较高的荧光素亮度值，也就是说有更大的处理器可用效率的物理主机，移动阶段又可以进一步划分为下面 3 个小步骤。

1）发现邻居区域

在局部径向范围 $\omega_d^j(t)$ 的物理主机 j 的邻居集 $N_j(t)$ 可以通过式（15）进行计算。

$$N_j(t)=\left\{k:\left\|u_k(t)-u_j(t)\right\|\leqslant\omega_d^j(t);\ell_j\leqslant\ell_n(t)\right\} \tag{15}$$

其中，k 是最接近物理主机 j 的物理主机的编号，$u_k(t)$ 和 $u_j(t)$ 分别表示其对应的物理主机处理器的利用效率，$\ell_n(t)$ 和 $\ell_j(t)$ 表示其对应的物理主机的荧光素亮度值，$\|u\|$ 表示变量 u 的欧几里德范式，$\omega_d^j(t)$ 表示局部径向范围。

2）计算概率

对于每个物理主机 j，概率表示了它上面运行的虚拟机朝着高亮度的邻居的移动方向，该概率 $p_{jn}(t)$ 可以通过式（16）进行计算。

$$p_{jn}(t)=\frac{\ell_n(t)-\ell_j(t)}{\sum_{k\in N_j(t)}\ell_k(t)-\ell_j(t)} \tag{16}$$

其中，$p_{jn}(t)$ 表示虚拟机从物理主机 j 到物理主机 n 的概率，物理主机 j 上的虚拟机从邻居集中，一直选择一个比较合适的物理主机进行放置，这种循环最后的终止条件是 $p_{jn}(t)$ 最高。

3）萤火虫移动

在移动阶段的最后，根据它所选择的物理主

机，虚拟机的资源使用效率也发生了变化，虚拟机的状态通过式（17）进行计算。

$$u_j(t+1)=u_j(t)+s\left(\frac{u_n(t)-u_j(t)}{\left\|u_n(t)-u_j(t)\right\|}\right) \quad (17)$$

其中 $u_j(t+1)$ 和 $u_j(t)$ 分别表示了虚拟机在物理主机 j 上放置的新状态，s 表示移动的步长。

（4）局部径向范围更新阶段

局部径向范围 $\omega_d^j(t)$ 通过式（18）来完成更新，这是根据萤火虫群优化算法来完成计算的，给物理主机的下一步的虚拟机放置增加了更多的灵活度。

$$\omega_d^j(t+1)=\min\left\{\omega_s,\max\left\{0,\omega_d^j(t)+\theta\left(k_t-\left|N_j(t)\right|\right)\right\}\right\} \quad (18)$$

其中，θ 是相邻区域范围的变化率，k_t 是最佳的邻居的数目，$N_j(t)$ 是物理主机 j 的邻居集。GSO-VMP 虚拟机放置策略工作流程如图 2 所示。

3.2　GSO-VMP 放置策略的实现

GSO-VMP 萤火虫群虚拟机放置优化整个过程的伪代码如下。

算法 1

输入　物理主机列表 PhList,虚拟机选择阶段产生的侯选虚拟机列表 VMsToMigrateList，初始参数

输出　已经完成好的虚拟机放置列表 Placement of VMs

Initialize parameters θ, p, s, k_t, ω_d^j

g = newGlowworm(vm, PhList)

PhId = luciferin.getPh()

currentluc=g.calculateLuc(PhId) by 式（13）

while $i < N$ **do**

for j = 1 to M **do**

newluc=g.calculateNewLuc(PhId) by 式（14）

neighbours = g.getNeighbours (PhId, neighbourSize) by 式（14）

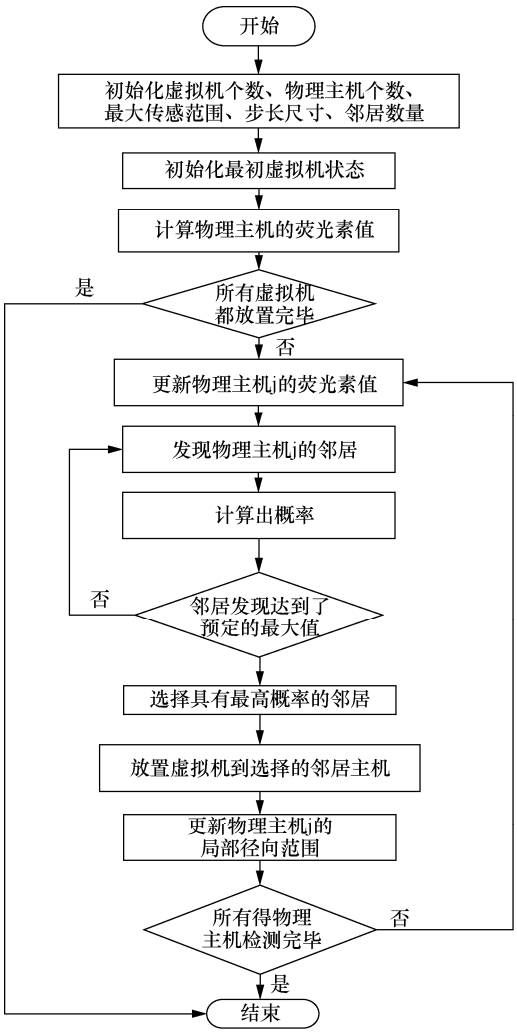


图 2　GSO-VMP 虚拟机放置策略工作流程

while j < neighbours.size() **do**

neighbourId = neighbours.get(j)

neighbourluc=g.calculateLuc (neighbourId)

if　neighbourluc = 0 **then**

currentluc = neighbourluc

neighbourPhId = neighbours.get(j)

j= neighbours.size()

end if

luc = currentluc- neighbourluc

if　luc > luciferin **then**

if neighbourPhId < ω_d^j **then**

luciferin = luc

currentluc = neighbourluc

```
            neighbourPhId = neighbours.get(j)
            end if
          end if
        end while
        allocatedPh = PhList.get (neighbourPhId)
        update radial range by 式（18）
      end for
    end while
```

4 GSO-VMP 仿真实验与性能分析

4.1 实验环境

本文参考了 Cloudsim4.0 工具包，同时依据图 1 中的功能模块，实现了基于 Java 语言的云资源管理方法，根据算法 1 在该模块中实现了萤火虫群算法的优化代码。表 1 显示了云数据中心的物理主机和虚拟机的参数配置情况，为了使描述简单，这些物理主机都是相同的配置，表 2 显示了 GSO-VMP 算法与其他常见的 Benchmark 智能算法的参数设置，这些参数设置是虚拟机放置优化算法的常见经验设置，用来比较不同的虚拟机放置算法的性能优劣[4]。

表 1 云数据中心的物理主机的参数设置

云数据中心	参数	具体值
物理主机	内存	8 GB
	磁盘	1 TB
	操作系统	Linux
	网络带宽	1 Tbit/s
	处理器	Intel CoreTM i7
	虚拟机管理器	Xen
	CPU 能量消耗模型	SpecPower

表 2 不同优化策略性能分析相关的参数设置

算法	参数	具体值
GSO-VM	p 荧光素衰减系数(0< p <1)	0.4
	物理主机 j 的局部径向范围的最大传感范围 ω_s	0.6
	相邻区域的变化率 θ	0.08
	最佳的邻居的数目 k_t	5
	移动的步长尺寸 s	0.03
	荧光素初始值 ℓ_0	0.05
FPA[4]	种群大小	50, 100, 150, 200
	标准函数 gamma	1.5
	随机步长 L	[0, 1]
	转换概率 p	0.9
	最大迭代次数	1 000
GA[22]	种群大小	50, 100, 150, 200
	交叉率	0.5
	变异率	0.01
	最大迭代次数	1 000
ACO[29]	蚁群算法的蚂蚁个数	50, 100, 150, 200
	挥发因子 p	0.4
	信息素追踪权重 α	0.3
	启发式信息权重 β	1
	信息素更新常量 Q	100
	最大迭代次数	1 000

4.2　客户端虚拟机类型设置

实验中与虚拟机相关的参数都来自最常见的 CoMon 项目，它是由 PlanetLab 开发的一个项目[30]。虚拟机迁移周期设置为 10 min 一次，一共运行 24 h，每次统计一天内的能量消耗，在一周内重复运行 5 次取平均值，一周内每天虚拟机请求的个数见表 3，不同的虚拟机粒度见表 4。

表 3　周一到周五 PlanetLab 项目的设置

时间	虚拟机个数/个	物理主机个数/个	运行时间/*h*
周一	1 000～5 000	800	24
周二	1 000～5 000	800	24
周三	1 000～5 000	800	24
周四	1 000～5 000	800	24
周五	1 000～5 000	800	24

表 4　CoMon 项目不同虚拟机粒度的参数设置

名称	处理器个数/个	内存需求/GB	磁盘需求/GB
类型 1 微型	2	3.75	2×8
类型 2 小型	4	7.5	2×16
类型 3 中型	8	15	2×40
类型 4 较大	16	30	2×80
类型 5 大型	32	60	2×160

4.3　仿真结果与性能分析

4.3.1　物理资源的利用效率

GSO-VMP 多维物理资源利用效率分析如图 3 所示，显示了各个智能算法优化针对云数据中心的所有物理主机的处理器资源、内存资源和存储资源的平均利用效率情况，也对包括云数据中心在内的所有活跃物理主机情况进行了分析。CPU 利用效率随虚拟机个数的变化如图 4 所示，显示了随着虚拟机请求个数的增加，物理资源利用效率的变化情况。这些实验结果表明，GSO-VMP 策略可以使得云数据中心的物理主机资源利用率平均达到 91%以上，而且处理器的利用效率最高。

在相同的软硬配置和虚拟机请求的条件下，花授粉优化算法（flower pollination algorithm，FPA）、遗传算法（genenic algorithm，GA）以及蚁群优化（ant colony optimization，ACO）算法等虚拟机放置优化策略只能使得云数据中心的物理资源达到 72%的利用效率，GSO-VMP 策略能够高效地利用物理资源。分析原因是 GSO-VMP 能够在局部径向范围内搜索到更多的可用物理主机，完成虚拟机放置，通过定义最佳邻居数量，经过萤火虫群优化后，各个物理主机的处理器的利用效率提高，从而间接地节省了物理主机的能量消耗。

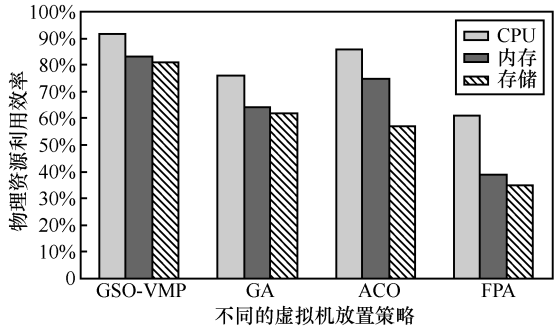


图 3　GSO-VMP 多维物理资源利用效率分析

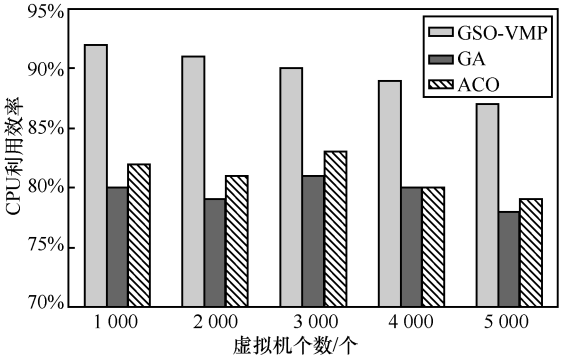


图 4　CPU 利用效率随虚拟机个数的变化

另外一个原因是 GSO-VMP 中虚拟机迁移都是基于虚拟机对处理器利用效率需求情况。GSO-VMP 可以很好地分配虚拟机到目标物理主机，整体来讲，所有上面的实验结果都证明 GSO-VMP 是一个可操作的，成功的和高效的大规模云资源分配和优化策略。

4.3.2　云数据中心的总体能量消耗

图 5 和图 6 显示了 GSO-VMP、GA、ACO、FPA 等虚拟机放置策略在不同的虚拟机请求下的

云数据中心的能量消耗情况。总体来说，随着云客户端的虚拟机个数的增加，能量消耗都在增长。与 GA、ACO 和 FPA 策略比较起来，GSO-VMP 策略的能量消耗最小。

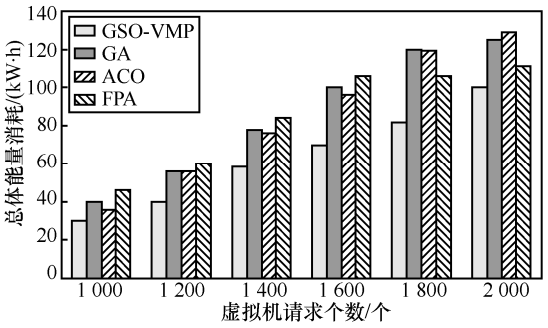


图 5　GSO-VMP 随虚拟机个数的能量消耗情况

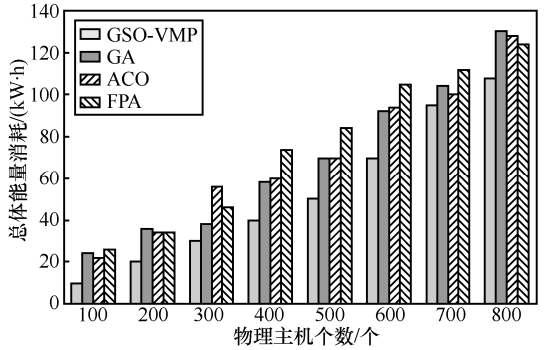


图 6　GSO-VMP 随主机个数的能量消耗情况

因为客户端的虚拟机请求个数的增加，越来越多的物理主机将被分配虚拟机，当活动主机数目不够的时候，云数据中心还会启动处于睡眠状态的物理主机。GSO-VMP 能够获得这样的良好性能也是因为萤火虫群算法的虚拟机放置优化策略以及该策略在对云数据中心总体能量消耗的目标函数设计之上。

表 5 显示了随着物理主机数量的变化，整个云数据中心不同算法的总体能量消耗与资源利用情况。从表 5 可以看出，在相同的虚拟机请求个数条件下，GSO-VMP 策略的最大能量消耗只有 100.3 kW·h，GA 策略是 125 kW·h。ACO 策略是 129 kW·h，FPA 策略是 111 kW·h，因此 GSO-VMP 虚拟机放置策略比常见的优化策略可以节省 20% 的能量消耗，间接降低了企业成本。

5　结束语

本文提出了云数据中心基于萤火虫群优化的虚拟机放置策略——GSO-VMP。GSO-VMP 通过定义云数据中心总体能量消耗的目标函数，将物理主机处理器的使用效率作为萤火虫群的荧光素亮度值的函数参数，将萤火虫群优化算法的 4 个阶段运用到虚拟机的放置过程中，最终将大量的候选迁移虚拟机重新高效地放置到最合适的物理主机之上。实验结果表明，GSO-VMP 在能量消耗与资源利用方面有更加优秀的性能。下一步工作是将 GSO-VMP 放置策略运用到容器云资源分配之中。

参考文献：

[1] 陈双喜, 赵若琰, 刘会, 等. 基于 KVM 的虚拟机 Post-Copy 动态迁移算法稳定性优化[J]. 电信科学, 2021, 37(7):57-66.
CHEN S X, ZHAO R Y, LIU H, et al. Stability optimization of dynamic migration algorithm for Post-Copy of virtual machine based on KVM[J]. Telecommunications Science, 2021, 37(7):57-66.

[2] 黄丹池, 何震苇, 严丽云, 等. Kubernetes 容器云平台多租户方案研究与设计[J]. 电信科学, 2020, 36(9): 102-111.
HUANG D C, HE Z W, YAN L Y, et al. Research and design of multi-tenant scheme for Kubernetes container cloud platform[J].

表 5　不同算法的总体能量消耗与资源利用情况

虚拟机放置策略	资源利用	能量消耗/kw·h	活动物理主机数/个	虚拟机资源请求数/个
GSO-VMP	91.4%	100.3	450	2 000
GA	65.2%	125	736	2 000
ACO	71.5%	129	712	2 000
FPA	80%	111	606	2 000

Telecommunications Science, 2020, 36(9): 102-111.

[3] SHI T, MA H, CHEN G. Energy-aware container consolidation based on PSO in cloud data centers[C]//Proceedings of 2018 IEEE Congress on Evolutionary Computation. Piscataway: IEEE Press, 2018: 1-8.

[4] USMAN M J, ISMAIL A S, CHIZARI H, et al. Energy-efficient virtual machine allocation technique using flower pollination algorithm in cloud datacenter: a panacea to green computing[J]. Journal of Bionic Engineering, 2019, 16(2): 354-366.

[5] ARIANYAN E, TAHERI H, KHOSHDEL V. Novel fuzzy multi objective DVFS-aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers[J]. Journal of Network and Computer Applications, 2017(78): 43-61.

[6] KAAOUACHE M A, BOUAMAMA S. Solving Bin packing problem with a hybrid genetic algorithm for VM placement in cloud[J]. Procedia Computer Science, 2015, 60(1): 1061-1069.

[7] BELOGLAZOV A, BUYYA R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[8] WANG J V, CHENG C T, TSE C K. A power and thermal-aware virtual machine allocation mechanism for cloud data centers[C]//Proceedings of 2015 IEEE International Conference on Communication Workshop. Piscataway: IEEE Press, 2015: 2850-2855.

[9] 刘开南. 云数据中心基于遗传算法的虚拟机迁移模型[J]. 计算机应用研究, 2020, 37(4): 1115-1118.
LIU K N. Virtual machine migration model in cloud data centers based on genetic algorithm[J]. Application Research of Computers, 2020, 37(4): 1115-1118.

[10] 徐胜超. 利用遗传算法完成虚拟机放置策略的优化[J]. 计算机与现代化, 2020(12): 25-31, 42.
XU S C. Using genetic algorithm for virtual machine placement optimization[J]. Computer and Modernization, 2020(12): 25-31, 42.

[11] 徐胜超. 贪心算法优化云数据中心的虚拟机分配策略[J]. 计算机系统应用, 2021, 30(3): 134-141.
XU S C. Greedy algorithms optimized virtual machine allocation for cloud data centers[J]. Computer Systems & Applications, 2021, 30(3): 134-141.

[12] XIONG A P, XU C X. Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center[J]. Mathematical Problems in Engineering, 2014: 816518.

[13] 徐胜超. 一种新的蚁群算法优化的虚拟机放置策略[J]. 计算机测量与控制, 2021, 29(5): 235-240.
XU S C. A new ant colony algorithm optimized virtual machine placement strategy[J]. Computer Measurement & Control, 2021, 29(5): 235-240.

[14] 陈艳, 周天绮, 徐胜超. 利用蚁群算法完成虚拟机放置的优化[J]. 计算机工程与设计, 2021, 42(5): 1229-1234.
CHEN Y, ZHOU T Q, XU S C. ACO-VMP: using ant colony optimization algorithm for virtual machine placement[J]. Computer Engineering and Design, 2021, 42(5): 1229-1234.

[15] DUGGAN M, FLESK K, DUGGAN J, et al. A reinforcement learning approach for dynamic selection of virtual machines in cloud data centres[C]//Proceedings of 2016 Sixth International Conference on Innovative Computing Technology (INTECH). Piscataway: IEEE Press, 2016: 92-97.

[16] 戴娇, 张明新, 孙昊, 等. 花朵授粉算法的优化[J]. 计算机工程与设计, 2017, 38(6): 1503-1509.
DAI J, ZHANG M X, SUN H, et al. Optimization of flower pollination algorithm[J]. Computer Engineering and Design, 2017, 38(6): 1503-1509.

[17] LUO J P, LI X, CHEN M R. Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers[J]. Expert Systems With Applications, 2014, 41(13): 5804-5816.

[18] WANG J V, FOK K Y, CHENG C T, et al. A stable matching-based virtual machine allocation mechanism for cloud data centers[C]//Proceedings of 2016 IEEE World Congress on Services. Piscataway: IEEE Press, 2016: 103-106.

[19] WOOD T, SHENOY P, VENKATARAMANI A, et al. Sandpiper: black-box and gray-box resource management for virtual machines[J]. Computer Networks, 2009, 53(17): 2923-2938.

[20] MISHRA M, SAHOO A. On theory of VM placement: anomalies in existing methodologies and their mitigation using a novel vector based approach[C]//Proceedings of 2011 IEEE 4th International Conference on Cloud Computing. Piscataway: IEEE Press, 2011: 275-282.

[21] JOSEPH C T, CHANDRASEKARAN K, CYRIAC R. A novel family genetic approach for virtual machine allocation[J]. Procedia Computer Science, 2015(46): 558-565.

[22] VASUDEVAN M, TIAN Y C, TANG M L, et al. Energy-efficient application assignment in profile-based data center management through a repairing genetic algorithm[J]. Applied Soft Computing, 2018(67): 399-408.

[23] LIU X F, ZHAN Z H, DENG J D, et al. An energy efficient ant colony system for virtual machine placement in cloud computing[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(1): 113-128.

[24] ALBOANEEN D A, TIANFIELD H, ZHANG Y. Glowworm swarm optimisation algorithm for virtual machine placement in

cloud computing[C]//Proceedings of 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress. Piscataway: IEEE Press, 2016: 808-814.

[25] ZHOU Z, HU Z G, LI K Q. Virtual machine placement algorithm for both energy-awareness and SLA violation reduction in cloud data centers[J]. Scientific Programming, 2016: 5612039.

[26] JAMIL M, YANG X S. A literature survey of benchmark functions for global optimisation problems[J]. International Journal of Mathematical Modelling and Numerical Optimisation, 2013, 4(2): 150.

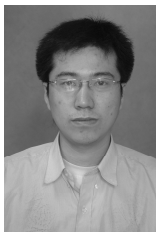
[27] WANG R, ZHOU Y Q. Flower pollination algorithm with dimension by dimension improvement[J]. Mathematical Problems in Engineering, 2014: 481791.

[28] LIN W W, XU S Y, HE L G, et al. Multi-resource scheduling and power simulation for cloud computing[J]. Information Sciences, 2017, 397/398: 168-186.

[29] FARAHNAKIAN F, ASHRAF A, PAHIKKALA T, et al. Using ant colony system to consolidate VMs for green cloud computing[J]. IEEE Transactions on Services Computing, 2015, 8(2): 187-198.

[30] SPEC. Benchmarks, standard performance evaluation corporation[S]. 2021.

[作者简介]



徐胜超（1980– ），男，广州华商学院数据科学学院讲师，主要研究方向为并行分布式处理软件。

熊茂华（1958– ），男，广州华商学院数据科学学院教授、硕士生导师，主要研究方向为嵌入式与物联网、智能控制、人工智能技术。

周天绮（1976– ），男，浙江药科职业大学医疗器械学院副教授，主要研究方向为图像处理、医疗大数据和云计算。