

# 一种新的蚁群算法优化的虚拟机放置策略

徐胜超

(广州华商学院 数据科学学院, 广州 511300)

**摘要:** 提出了一种新的蚁群算法优化的虚拟机放置策略 ACA-VMP (Ant Colony Algorithm based virtual machine placement); ACA-VMP 以云数据中心的总体能量消耗降低、服务质量最佳及减少虚拟机迁移次数为目标函数; 根据蚁群优化算法, ACA-VMP 采用了全局最优解和局部最优解信息素强度更新规则; 全局最优解经过多次迭代后, 蚂蚁路径的多次寻优, 保证这个虚拟机放置优化策略的完成; 局部信息素强度参数更新可以补充蚂蚁其他局部最优路径的寻找, 这样也可以使得 ACA-VMP 虚拟机放置优化算法更快的接近全局最优解; 仿真结果表明: ACA-VMP 策略使得云数据中心的各类性能指标都可以改善, 该实验结果对于其他企业构造节能云数据中心有很好的参考价值。

**关键词:** 蚁群算法; 多维物理资源; 信息素强度更新; 虚拟机放置; 全局最优解

## A New Ant Colony Algorithm Optimized Virtual Machine Placement Strategy

Xu Shengchao

(School of Date Science, Guangzhou HuaShang College, Guangdong, Guangzhou 511300)

**Abstract:** A new ant Colony Algorithm optimized approach for virtual machine placement called ACA-VMP was proposed. ACA-VMP is depended on Cloudsim project which is a popular tool for cloud data center simulation. The objectives of ACA-VMP are to reduce energy consumption, minimize the number of VM migrations, and maintain the desired QoS. ACA-VMP used a vector based resource utilization capturing technique and multi-dimensional resource demands was described. The global pheromone trail evaporation rule is applied towards the end of iteration after all ants complete their migration plans. The local pheromone trail update rule is also applied on a tuple when an ant traverses the tuple while making its migration plan. The experimental results and performance analysis show our strategy leads to a further improvement compared with the old optimized algorithm strategies. ACA-VMP is valuable for other cloud providers to build a low energy consumption cloud data center.

**Keywords:** Ant Colony Optimization algorithm; multi-dimensional server resource; pheromone reinforcement update; virtual machine placement; Global best solution

## 0 引言

大数据中心的构造与建设是近年来政府与云服务提供商越来越关注的问题, 为了节省企业成本, 这类数据中心的构造主要目标是绿色节能与物理资源的充分利用<sup>[1]</sup>。为了达到这 2 个目标, 云计算中大多都使用了虚拟机迁移技术<sup>[2-3]</sup>。目前以 Cloudsim 云模拟器工具包及其研究思路为背景的大部分文献在虚拟机迁移研究方面处理世界领先地位, 它将虚拟机迁移过程划分为物理主机状态检测、虚拟机选择、虚拟机放置 3 个步骤<sup>[4]</sup>, 这 3 个步骤既具有独立性又互相关联, 而且都可以通过算法进行优化<sup>[5-8]</sup>。

本文着重考虑采用智能算法的方式来优化虚拟机放置过程。虚拟机放置有些文献把它称为多目标优化问题或者装箱问题<sup>[9]</sup>, 该过程有很多智能算法进行优化, 例如遗传算法<sup>[10]</sup>、贪心算法<sup>[11]</sup>、粒子群优化算法<sup>[12]</sup>、萤火虫群优化算法<sup>[13]</sup>、蛙跳算法<sup>[14]</sup>等。本文依托于 Cloudsim 云平台工具包, 在物理主机状态检测和虚拟机选择过程都采用 Cloud-

sim 中默认的优化策略; 然后着重考虑采用蚁群算法的方式来优化虚拟机放置过程<sup>[15]</sup>。提出一种新型蚁群算法优化的虚拟机放置方法 ACA-VMP。

ACA-VMP 虚拟机放置优化策略参考了蚁群算法的思路, 在自适应的蚁群算法中假设有大量的蚁群觅食的行为, 在寻找最优解的过程中不断的交换一种称为信息素强度的参数, 这样在一定程度上可以避免多目标优化算法中的局部最优解的“早熟”问题<sup>[16-17]</sup>。

ACA-VMP 虚拟机放置策略考虑了物理主机的处理器、内存大小、网络带宽等多个资源维度的因素。与其他的虚拟机放置优化的智能算法比较起来, ACA-VMP 的目标函数综合考虑整个大数据中心的总体能量消耗和云服务器端的服务质量 Qos 提高等目标<sup>[18-20]</sup>。

## 1 ACA-VMP 建模与工作场景

### 1.1 虚拟机放置的问题描述

物理主机的描述向量如下:  $P_j = (id_j, cpu_j, mem_j,$

收稿日期: 2021-02-20; 修回日期: 2021-03-15。

基金项目: 广州华商学院校内导师制科研项目(2020HSDS04)。

作者简介: 徐胜超(1980-), 男, 湖北武汉人, 硕士、讲师, 主要从事并行分布式处理软件方向的研究。

引用格式: 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.

$network_j$ ),  $M$  表示了物理主机的标识符,  $cpu_j$  表示了物理主机的处理器的计算能力, 有时候也用 (Millions of Instructions Per Second, MIPS) 来表示,  $mem_j$  表示了物理主机的可用内存大小,  $network_j$  表示了物理主机所拥有的网络带宽大小, 每个虚拟机也可以通过下面的向量来表示:  $V_i = (id_i, cpu_i, mem_i, network_i)$ ,  $id_i$  表示了虚拟机的标识符,  $cpu_i$  表示了虚拟机的表示了对处理器的需求能力,  $mem_i$  表示了虚拟机对内存的需求能力,  $network_i$  表示了虚拟机对网络带宽的需求能力。

虚拟机放置问题的含义是寻找一个最优的虚拟机到物理主机的映射, 就把虚拟机迁移过程中前一虚拟机选择阶段的大量的虚拟机重新分配到大量物理主机之上, 图 1 描述了这种在云客户端访问大数据中心的时候平台的虚拟机迁移机制开始工作后的场景。在这种最优映射条件下满足物理主机的资源利用效率是最高; 或者整个云数据中心的能量消耗最小; 或者是保证云数据中心的的服务质量 QoS 较好, 最好的情况是达到 Pareto 最优解。

### 1.2 ACA-VMP 的工作场景

ACA-VMP 依托了 Cloudsim 模拟器工具包, 该工具包在运行模块中主要包括: 全局代理 Global Broker、本地代理 Local Broker、虚拟机管理器 Virtual Machine Manager<sup>[5]</sup>。

图 2 显示了 ACA-VMP 虚拟机放置策略的工作模块。在 Cloudsim 中每个物理主机上都运行有一个本地代理 Local Broker, 它监视物理主机的软硬件资源, 同时决定是否完成虚拟机迁移的动作, ACA-VMP 优化策略的实现代码主要在本地代理模块中完成。本地代理根据物理主机的处理器的利用率情况把其划分为 4 类:  $P_{normal}$ 、 $P_{over}$ 、 $P_{over}$ 、 $P_{under}$ , 正常主机、超负载主机、预测超负载主机、低负载主机等, 一般 CPU 的利用率低于或者高于一个阈值边界都处于  $P_{over}$  或  $P_{under}$  状态, 其他的为正常状态  $P_{normal}$ ,  $P_{over}$  是经过预测后的处理器的利用率。利用率阈值边界具体取值多少可以通过实验来测试来调整, 例如高于 90% 或者低于 10% 等。虚拟机管理器 Virtual Machine Manager 用来配合本地代理 Local Broker 具体的完成虚拟机迁移的启动与关闭, 这个工作场景在文献都有描述<sup>[5]</sup>。

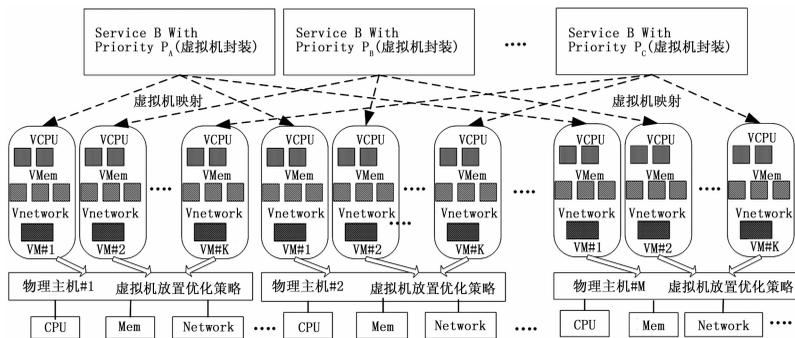


图 1 云数据中心虚拟机放置问题描述

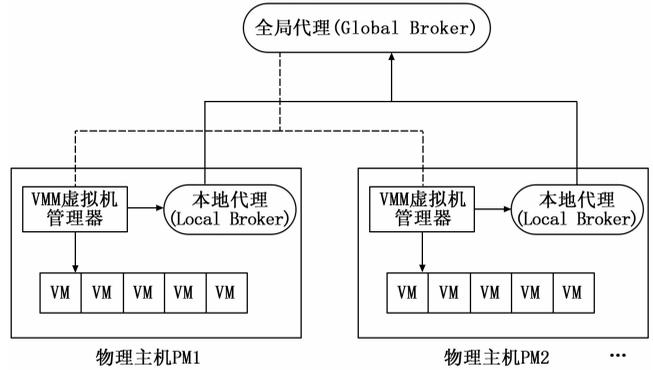


图 2 ACA-VMP 蚁群虚拟机放置优化策略实现模块

## 2 ACA-VMP 的虚拟机放置过程描述

对于一个上面容纳了

多个虚拟机的物理主机, 它随时都可能成为源主机  $P_{so}$ 。所有的源主机和源虚拟机都通过它们自身的物理资源利用效率进行描述: 比如处理器、内存大小、网络带宽大小。类似的每个虚拟机都可能成为潜在的源虚拟机, 根据它自身的资源利用率情况, 每个物理主机也可能成为潜在的目标物理主机  $P_{de}$ 。所以 ACA-VMP 优化算法给每一次虚拟机放置动作都设计了一个元组, 该元组由 3 个参数组成。

$$T = (p_{so}, v, p_{de}) \quad (1)$$

由于虚拟机放置这里元组可以类似于分布式旅行商问题中的连接各个城市的边。为了降低 ACA-VMP 优化算法的运行时间, 必须降低的  $|T|$  值。所以在构造元组的时候, 必须注意两个约束条件为真:

$$(P_{so} \in P_{over}) \vee (P_{so} \in P_{over}) \vee (P_{so} \in P_{under}) \quad (2)$$

它可以保证预测超负载的物理主机  $P_{over}$ 、超负载物理主机  $P_{over}$ 、低负载物理主机  $P_{under}$  只能作为源物理主机, 这样的好处可以降低超负载物理主机的 SLA 违规率, 同时把那些低负载的主机切入到睡眠或者关闭状态, 降低活动物理主机数量, 间接的降低云数据中心的能量消耗。第二个约束条件为是:

$$(P_{de} \notin P_{over}) \wedge (P_{de} \notin P_{over}) \quad (3)$$

这个条件为真可以保证预测超负载的物理主机  $P_{over}$  和超负载物理主机  $P_{over}$  不可能成功目标物理主机。通过上面的这两个约束条件之后, 整体上保证了 ACA-VMP 优化算法的运行时间能够很快的减少。

根据蚁群觅食寻优路径的分析, ACA-VMP 优化算法的目标函数是:

$$f(M) = |P_s|^\gamma + \frac{1}{M} \quad (4)$$

$M$  表示的是整个虚拟机迁移过程,  $P_s$  表示的是在执行虚拟机迁移过程  $M$  后, 可以转入睡眠模式的所有物理主机的集合, 根据  $|M|, \gamma$  是与睡眠

数量  $P_s$  的相关的参数，它可以设置为常量。由于最终的目标是降低活动物理主机的个数，ACA-VMP 的目标函数的定义可以根据睡眠主机数  $|P_s|$  来确定。

在 ACA-VMP 优化算法快结束的时候，它可以通过把虚拟机迁移到已经存在的正常状态的活动物理主机上从而降低整个云数据中心的物理主机数量，因为这个时候睡眠模式的物理主机可以进入关闭状态。这里有一个新的约束条件：

$$P_s = \{ \forall p \in P \mid V_p = \varphi \} \quad (5)$$

该条件可以保证当所有的虚拟机都被迁移出去之后，物理主机将被切换到睡眠模式。

ACA-VMP 算法中蚁群的信息素被分解成元组。每个蚂蚁使用一个状态比例规则完成对下一个元组的选择，该原则在蚁群算法中称为伪随机比例规则。蚁群算法中根据这个规则，第  $k$  个蚂蚁根据下面的公式选择一个元组：

$$s = \begin{cases} \operatorname{argmax}_{u \in T_k} \{ [\tau_u] * [\eta_u]^\beta \} & \text{if } q \leq q_0 \\ S & \text{otherwise} \end{cases} \quad (6)$$

$q$  是均匀分布在区间  $[0, 1]$  之间的随机数， $q_0$  是在区间  $[0, 1]$  的一个参数。 $\tau$  描述了信息素变量， $\eta$  是与元组  $T = (p_w, v, p_{dw})$  相关的当前启发式参数值。 $\beta$  是一个用来确定信息素变量和启发式参数值的相关性的一个非负参数。 $T_k \subset T$  表示了经过了第  $k$  蚂蚁路径寻优之后的剩余的元组的集合。 $S$  是一个根据公式 (6) 的概率分布来确定的随机变量。第  $k$  个蚂蚁选择元组  $s$  的概率表示为  $p_s$ ，它可以通过公式 (7) 来计算：

$$p_s = \begin{cases} \frac{[\tau_s] * [\eta_s]^\beta}{\sum_{u \in T_k} [\tau_u] * [\eta_u]^\beta} & \text{if } s \in T_k \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

一个元组的启发式参数值  $\eta_s$  可以根据蚁群算法<sup>[21-22]</sup>中的规则定义如下公式 (8)：

$$\eta_s = \begin{cases} (|C_{p_s} - (U_{p_s} + U_v)|_1)^{-1} & \text{if } U_{p_s} + U_v \leq C_{p_s} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

$C_{p_s}$  表示了目标物理主机的整体资源提供能力； $U_{p_s}$  表示了目标物理主机的整体资源利用效率； $U_v$  表示了元组  $s$  上的虚拟机的利用效率；这里  $\eta_s$  是一个多维物理资源利用效率向量，它支持通过虚拟机迁移的方式完成低负载物理主机数量的降低。 $U_{p_s} + U_v \leq C_{p_s}$  这个约束条件可以避免虚拟机迁移之后目标物理主机的超负载。ACA-VMP 优化算法中包括 3 个维度的物理资源：处理器、内存大小和网络带宽大小，根据蚁群算法的公式 (6) 和公式 (7) 的条件完成优化之后，基本可以保证高信息素变量的集中和大量的物理主机资源释放。

在公式 (7) 中，指数函数有利于局部最优路径的快速收敛，保证了一个广泛的搜索路径。根据经典的蚁群优化算法，ACA-VMP 采用了全局最优解和局部信息素更新规则，全局最优解采用更新规则，经过多次迭代后，蚂蚁路径的多次寻优，保证这个虚拟机放置优化策略的完成。信息素强度变量的更新规则如下：

$$\tau_s = (1 - \alpha) * \tau_s + \alpha * \Delta_\tau^+ \quad (9)$$

$\Delta_\tau^+$  是应用到每个虚拟机-物理主机映射元组对的信息素强度参数： $\Delta_\tau^+$  的计算公式如下：

$$\Delta_\tau^+ = \begin{cases} f(M^+), & \text{if } s \in M^+ \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

变量  $\alpha$  的含义是全局信息素延迟参数， $0 < \alpha < 1$ ， $M^+$  表示了蚁群算法的虚拟机放置全局最优解。局部最优解也采用更新规则，规则如下：

$$\tau_s = (1 - \rho) * \tau_s + \rho * \tau_0 \quad (11)$$

变量  $\rho$  的含义是全局信息素参数  $\alpha$  类似的参数， $0 < \rho < 1$ ， $\tau_0$  是初始信息素级别变量，计算的公式如下：

$$\tau_0 = (|M| * |P|)^{-1} \quad (12)$$

ACA-VMP 虚拟机放置优化算法的伪随机比例规则和全局信息素强度变量更新法则的目的都是要使得蚂蚁的搜索更加直接，伪随机比例规则更趋向于使虚拟机-物理主机元组  $T$  能够获得更高的信息素级别和高的启发式参数值。因此，蚂蚁就可以在更加邻近的区域寻找它们的全局最优解。

另外一个方面，局部信息素强度参数更新可以补充蚂蚁其他局部最优路径的寻找，这样也可以使得 ACA-VMP 虚拟机放置优化算法更快的接近全局最优解。这是因为无论何时蚁群在利用元组寻优的过程中，元组将失去它们自己的信息素，同时对其他的蚂蚁缺少更多的吸引力。因此这样它就有利于避免对局部最优解的“早熟”的情况。

ACA-VMP 虚拟机放置优化算法的伪代码如下面的 Algorithm1 所示：

Algorithm 1 ACA-based VM placement

- 1:  $M^+ = \varphi, MS = \varphi$
- 2:  $\forall t \in T \mid \tau_t = \tau_0$
- 3: for  $i \in [1, nI]$  do
- 4: for  $k \in [1, nA]$  do
- 5:  $M_k^m = \varphi, M_k = \varphi, Scr_k = 0$
- 6: for  $t \in T$  do
- 7: generate a random variable  $q$  with a uniform Distribution between 0 and 1
- 8: if  $q > q_0$  then
- 9: compute  $p_s \forall s \in T$  by using (7)
- 10: end if
- 11: choose a tuple  $t \in T_k$  to traverse by using (6)
- 12:  $M_k^m = M_k^m \cup \{t\}$
- 13: apply local update rule in (11) on  $t$
- 14: update used capacity vectors  $U_{p_s}$  and  $U_{p_s}$  in  $t$
- 15: if  $f(M_k^m) > Scr_k$  then
- 16:  $Scr_k = f(M_k^m)$
- 17:  $M_k = M_k \cup \{t\}$
- 18: else
- 19:  $M_k^m = M_k^m \setminus \{t\}$
- 20: end if
- 21: end for
- 22:  $MS = MS \cup \{M_k\}$

23: end for  
 24:  $M^+ = \operatorname{argmax}_{M_k \in MS} \{f(M_k)\}$   
 25: apply global update rule in (9) on all  $s \in T$   
 26: end for

它利用公式 (4) 在  $M_k^m$  之上建立了 ACA-VMP 优化算法的目标函数, 如果它的值在循环过程中只要大于全局最优解, 参数  $t$  将加入到一个具体的蚁群优化步骤  $M_k$  之中。否则该元组  $t$  将从临时的虚拟机放置优化步骤中删除。

最后在迭代过程中当所有的蚂蚁完成它们的路径寻优步骤后, 程序将转入到 MS 虚拟机迁移过程中。每一次循环,  $M_k \in MS$  都要经过公式 (4) 的目标函数进行评估;  $M^+$  表示了全局最优的虚拟机放置步骤。全局信息素参数规则根据公式 (9) 和公式 (10) 完成更新, 当所有的迭代完成或者运行特定的时间之后结束, 算法将输出全局最优问题解  $M^+$ 。为了描述方便, 算法 Algorithm1 中涉及到的公式与符号都在表 1 中详细列出。

表 1 ACA-VMP 虚拟机放置策略的相关符号描述

符号	具体含义
MS	虚拟机放置过程功能函数
T	虚拟机-物理主机放置的元组的集合
$T_k$	第 $k$ 个蚂蚁寻优后的元组 T
$P_s$	处于睡眠状态的物理主机
$C_{p_s}$	目标物理主机
$P_{de}$	的资源提供能力
M	虚拟机放置过程集
$M^+$	全局最优虚拟机放置结果集
$M_k$	第 $k$ 个蚂蚁的虚拟机放置过程
$q$	均匀分布的随机变量
$q_0$	是在区间 $[0, 1]$ 的一个参数
S	由公式(6)选择后产生的随机变量
$Scr_k$	第 $k$ 个蚂蚁路径寻优后的最高分
$U_{p_s}$	目标物理主机的资源提供能力向量
$U_{p_e}$	源物理主机的资源提供能力向量
$P_{de}$	元组 T 中的目标物理主机
$P_{so}$	元组 T 中的源物理主机
$\alpha$	全局信息素变量更新中的强度延迟参数
$\beta$	与启发式参数值 $\eta$ 相关的参数
$\gamma$	与睡眠主机数 $ P_s $ 相关的参数
$\rho$	局部信息素变量更新中的强度延迟参数
nA	蚂蚁并行寻优过程中的数量
nI	ACA-VMP 优化算法的循环次数

### 3 仿真实验与性能分析

#### 3.1 仿真环境与性能比较对象

因为 ACA-VMP 基于蚁群算法优化的虚拟机放置是在虚拟机迁移过程中运用的, 所以进行 ACA-VMP 实验分析, 必须构造云数据中心的虚拟机迁移场景, 本文参考了 Cloudsim3.0 工具包, 同时依据图 2 中的功能模块, 实现了基于 Java 语言的局部代理, 根据 Algorithm1 在该代理中实现了蚁群算法的优化的代码。

ACA-VMP 依托于 Cloudsim 云平台工具, 在物理主机状态检测和虚拟机选择过程都采用 Cloudsim 中默认的优化策略: 鲁棒局部回归物理主机状态检测 (local regression robust, LRR) 策略和最短时间虚拟机选择策略 (minimum migration time, MMT);

为了体现蚁群智能优化虚拟机放置策略的高效性与优秀性能, 与 ACA-VMP 同时做性能比较的虚拟机放置的智能优化算法包括如下:

- 1) 遗传算法 GA 优化的虚拟机放置<sup>[10]</sup>;
- 2) 粒子群算法 PSO 优化的虚拟机放置<sup>[12]</sup>;
- 3) 萤火虫群算法 GSO 优化的虚拟机放置<sup>[13]</sup>;
- 4) 自适应的最大最小蚁群算法 MMVMC<sup>[24]</sup>;

被模拟的云数据中心物理服务器总数为 800 个, 物理服务器配置如表 2 所显示。

表 2 云数据中心物理服务器硬件配置

主机个数	CPU 提供能力 (MIPS)	内存大小/GB	磁盘空间/GB	网络带宽/(GB/s)
200	1 600	4	1 000	1
200	1 600	4	500	1
200	1 600	8	500	1
200	2 200	8	500	1

虚拟机迁移周期设置为 10 分钟一次, 一共运行 24 个小时, 每次统计一天内的能量消耗, 在一周内重复运行五次取平均值, 一周内每天虚拟机请求的个数如表 3。

表 3 ACA-VMP 虚拟机放置策略测试环境

日期	虚拟机个数	物理主机个数	运行时间
周一	1 000	800	24 小时
周二	1 000	800	24 小时
周三	1 000	800	24 小时
周四	1 000	800	24 小时
周五	1 000	800	24 小时

虚拟机的放置中有一个装箱效率的概念<sup>[19]</sup>, 可以认为是虚拟机粒度, 我们设置为至少 4 个, 一共 4 个类型, 装箱效率分别为 10, 6.75, 5, 4, 这样设置的原因是为了体现云平台有不同资源需求大小尺寸的虚拟机来访问云数据中心。例如装箱效率如果为 10, 这样说明这个虚拟机尺寸比较小, 一个物理主机上最多可以容纳 10 个虚拟机, 那么最理想装箱效率就为 10。ACA-VMP 蚁群算法的其他参数初始化设置如表 4。

这里比较了 4 个主要性能指标: 1) 云数据中心的能量消耗; 2) 装箱效率; 3) 虚拟机迁移次数; 4) 云数据中心的 SLA 违规率;

#### 3.2 仿真结果与性能分析

##### 3.2.1 云数据中心的能量消耗和装箱效率

为了实验测试的需要, 表 5 显示了在 1 000 个虚拟机个数情况下 ACA-VMP 蚁群优化的虚拟机放置算法与 4 个优化算法的能量消耗、装箱效率的性能比较结果: 从表 5 可以

表 4 ACA-VMP 虚拟机放置优化策略参数设置

参数	取值
$nA$	10
$nI$	2
$w$	$2^{-7}$
$\alpha$	0.1
$\beta$	0.9
$\gamma$	5
$\rho$	0.1
$q_0$	0.9

看出，ACA-VMP 优化算法在各个方面都优于其他 4 个算法。ACA-VMP 优化算法的装箱效率最接近理想值，这也是物理主机负载均衡的一个重要体现，针对不同的虚拟机粒度，相对于其他智能优化算法，ACA-VMP 优化可以节省大约 10~20% 左右的能量消耗。

表 5 各类虚拟机放置优化算法的性能比较

虚拟机粒度	优化算法	能量消耗 (单位:KWh)	装箱效率 PE (单位:无)
10	ACA-VMP	106	10
	GA	108	9.71
	PSO	113	9.26
	GSO	120.6	8.55
	MMVMC	121.4	8.47
6.75	ACA-VMP	166	6.55
	GA	172	6.13
	PSO	174.4	5.99
	GSO	184.4	5.62
	MMVMC	183	5.65
5	ACA-VMP	226	4.85
	GA	234	4.42
	PSO	246	4.17
	GSO	248	4.13
	MMVMC	248.6	4.13
4	ACA-VMP	282	3.80
	GA	297	3.50
	PSO	316.6	3.23
	GSO	305	3.38
	MMVMC	305	3.38

但是随着虚拟机粒度参数的变化，在比较小（粒度为 10 和 6.75）的资源需求参数情况下，ACA-VMP 优化算法优化的性能超过 MMVMC 优化算法和 GSO 算法，性能提高比例比较大。在比较大的资源需求参数情况下（粒度为 5 和 4），ACA-VMP 优化算法性能超过粒子群优化 PSO 算法的比例比较大。

分析原因是 ACA-VMP 启发式虚拟机放置算法具有很好的灵活度，很容易把虚拟机尺寸比较小的多个虚拟机放置到同一个物理主机上，如果虚拟机尺寸太大，反而效果不明显。另一方面，在虚拟机尺寸比较大的情况下，萤火虫群优化 GSO 算法也可以获得比较高的整体物理资源利用

效率，需求的活动物理主机个数比较少。

### 3.2.2 虚拟机迁移次数和 SLA 违规率

表 6 显示了在 4 种虚拟机粒度下 ACA-VMP 的虚拟机优化策略的迁移次数低于 PSO 策略、GSO 迁移策略、MMVMC 策略，随着粒度的增强，虚拟机迁移次数逐渐减少；而遗传算法 GA 优化的策略虚拟机迁移次数在某些条件下还要小于 ACA-VMP 策略。

表 6 各类虚拟机放置优化算法的性能比较

虚拟机粒度	优化算法	迁移次数 (次)	SLA 违规率 (%)
10	ACA-VMP	10 500	0.001 5
	Genetic	11 000	0.001 0
	PSO	13 000	0.001 8
	GSO	14 000	0.002 4
	MMVMC	18 000	0.003 0
6.75	ACA-VMP	10 000	0.001 1
	Genetic	11 500	0.000 9
	PSO	12 000	0.001 4
	GSO	13 000	0.001 8
	MMVMC	17 000	0.002 4
5	ACA-VMP	9 800	0.001 0
	Genetic	10 000	0.000 9
	PSO	11 500	0.001 3
	GSO	12 500	0.002 0
	MMVMC	15 500	0.002 1
4	ACA-VMP	9 600	0.000 9
	Genetic	8 800	0.000 9
	PSO	10 500	0.001 1
	GSO	11 000	0.001 9
	MMVMC	14 500	0.001 7

原因是 ACA-VMP 策略中设计的能量消耗模型考虑问题的维度比较广泛，包括处理器、内存、磁盘和网络带宽，局部和全局的信息素强度更新变量可以使得虚拟机放置过程不断优化，提高了资源的利用效率，使得负载比较均匀的放置在各个活动物理主机之上，迁移次数降低。但是在这个过程中也不一定是最优的，ACA-VMP 优化策略立足于能量消耗最小而不是立足于迁移次数最小，所以遗传算法 GA 的迁移次数少于 ACA-VMP 优化策略也是可以接受的。

当一个云客户端提交作业到云计算平台的时候，如果云平台没有提供合适的物理资源，就会出现服务协议 (Service Level Agreement, SLA) 违规。从表 6 可以看出，不同虚拟机粒度条件下，ACA-VMP 优化策略的 SLA 违规率比 PSO 优化策略、GSO 优化策略、MMVMC 策略要低，随着粒度的增强，SLA 违规率逐渐减少；有这样的实验结果的原因是 ACA-VMP 中的各类资源约束条件在物理主机状态检测阶段尽量避免物理资源使用率超过 100% 的情况，从而间接的避免 SLA 违规事情的发生。大粒度虚拟机访问一次需要稳定的资源，云服务器虚拟机迁移次数少，SLA 违规率自然降低。

## 4 结束语

本文采用蚁群优化启发式智能算法, 针对云数据中心的绿色节能和高服务质量两个目标, 提出了一种新的蚁群算法优化的虚拟机放置方法 ACA-VMP。实验结果表明和常见的智能优化的虚拟机放置策略比较起来, ACA-VMP 在总体能量消耗、装箱效率、虚拟机迁移次数, SLA 违规率等方面性能较好。ACA-VMP 策略的目前只是在虚拟机放置阶段, 它还可以和虚拟机选择阶段的优化策略结合起来使用, 这个是本文的后续工作。

### 参考文献:

- [1] Ariyan 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 & Computer Applications*, 2017, 78 : 43 - 61.
- [2] 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.
- [3] Jin X, Zhang F, Wang L, et al. Joint Optimization of Operational Cost and Performance Interference in Cloud Data Centers [J]. *IEEE Transactions on Cloud Computing*, 2017, 1 (99): 697 - 711.
- [4] 徐胜超. 虚拟机迁移中一种新的物理主机异常状态检测算法 [J]. *计算机测量与控制*, 2020, 28 (10): 241 - 246.
- [5] Cloudsim [EB/OL]. <http://github.com/Cloudslab/cloudsim/>, 2019.
- [6] Zhou Z, Zhigang H, Keqin L. Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers [J]. *Scientific Programming*, 2016, 1 - 11.
- [7] Zoltán Mann. Interplay of Virtual Machine Selection and Virtual Machine Placement [A]. *European Conference on Service-oriented & Cloud Computing*. Springer International Publishing [C]. 2016.
- [8] Joseph C T, Martin J P. Task Dependency Aware Selection (TDAS) in Cloud [J]. *Procedia Computer Science*, 2016, 93: 269 - 275.
- [9] 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.
- [10] Meera Vasudevan, Yu-Chu Tian, Maolin Tang, et al. Energy-efficient Application Assignment in Profile-based Data Center Management Through a Repairing Genetic Algorithm [J]. *Applied Soft Computing*, 2018, 67 (1): 1 - 10.
- [11] Rasoul Beik. Green Cloud Computing: Greedy Algorithms for Virtual Machines Migration and Consolidation to Optimize Energy Consumption in a Data Center [J]. *International Journal of Digital Application & Contemporary research*, 2014, 2 (9): 1 - 9.
- [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, 1 - 8.
- [13] Alboaneen D A, Tianfield H, Zhang Y. Glowworm Swarm Optimisation Algorithm for Virtual Machine Placement in Cloud Computing [A]. *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 IEEE [C]*. 2016.
- [14] 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.
- [15] Dorigo M, Birattari M, Stutzle T. Ant colony optimization [J]. *Computational Intelligence Magazine, IEEE*, 2006, 1 (4): 28 - 39.
- [16] Joseph C T, Chandrasekaran K, Cyriac R. A Novel Family Genetic Approach for Virtual Machine Allocation [J]. *Procedia Computer Science*, 2015, 46: 558 - 565.
- [17] Feller E, Morin C, Esnault A. A case for fully decentralized dynamic VM consolidation in clouds [A]. in *Cloud Computing Technology and Science (CloudCom)*, 2012 IEEE 4th International Conference on [C]. 2012, 26 - 33.
- [18] Wang J V, Cheng C T, Tse C K. A power and thermal-aware virtual machine allocation mechanism for Cloud data centers [A]. *IEEE International Conference on Communication Workshop, IEEE [C]*. 2015.
- [19] Ferdaus M, Murshed M, Calheiros R, et al. Virtual machine consolidation in cloud data centers using ACO metaheuristic [A]. in *Euro-Par 2014 Parallel Processing*, ser. *Lecture Notes in Computer Science*, Silva F, Dutra I, Santos Costa V, Eds, Springer International Publishing [C]. 2014, (8632): 306 - 317.
- [20] 沈华峰. 一种云计算环境下的虚拟机动态迁移策略 [J]. *计算机测量与控制*, 2019, 27 (3): 159 - 163.
- [21] 姜栋瀚, 林海涛. 基于布谷鸟搜索的虚拟机放置算法 [J]. *电信科学*, 2017 (10): 96 - 104.
- [22] Ashraf A, Porres I. Using ant colony system to consolidate multiple web applications in a cloud environment [A]. in *22nd Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP) [C]*. 2014, 482 - 489.
- [23] Liu X F, Member S, 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] Eugen Feller, Louis Rilling, Christine Morin, et al. Energy-aware ant colony based workload placement in clouds [A]. In: *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing [C]*. IEEE Computer Society, 2011, 26 - 33.