

基于粒子群优化算法的虚拟机部署策略

杨靖*, 张宏军, 赵水宁, 占栋辉

(解放军理工大学 仿真与数据中心, 南京 210007)

(* 通信作者电子邮箱 262986127@qq.com)

摘要: 针对云计算基础设施即服务(IaaS)中的虚拟机部署问题, 提出一种基于粒子群优化(PSO)算法的部署策略。由于 PSO 算法在处理虚拟机部署这类大规模复杂问题时, 具有收敛速度慢且容易陷入局部最优的缺点, 首先, 引入多种群进化模式提高算法收敛速度, 并在此基础上加入高斯学习策略避免局部最优, 提出了一种多种群高斯学习粒子群优化(MGL-PSO)算法; 然后, 根据部署模型, 使用轮询(RR)算法对 MGL-PSO 进行初始化, 进而提出了一种以负载均衡为目标的虚拟机部署策略。通过在 CloudSim 中进行仿真实验, 验证了在解决虚拟机部署问题时, MGL-PSO 相比 PSO 算法, 具有更快的收敛速度, 并且负载均衡度降低了 13.1%。在两种实验场景下, 所提算法相比随机负载均衡(OLB)算法, 其负载均衡度分别平均降低了 25% 和 15%; 相比贪婪算法(GA), 使负载均衡度分别平均降低了 19% 和 7%。

关键词: 虚拟机部署; 粒子群优化; 负载均衡; 高斯学习; 多种群进化

中图分类号: TP311; TP18 **文献标志码:** A

Virtual machine deployment strategy based on particle swarm optimization algorithm

YANG Jing*, ZHANG Hongjun, ZHAO Shuining, ZHAN Donghui

(Center of Simulation and Data, PLA University of Science and Technology, Nanjing Jiangsu 210007, China)

Abstract: To solve the virtual machine deployment problem in Infrastructure as a Service (IaaS) of cloud computing, a virtual machine deployment strategy based on Particle Swarm Optimization (PSO) algorithm was proposed. Since the PSO algorithm has weaknesses of having a slow convergence speed and falling into local optimum easily when dealing with large-scale and complex problems like virtual machine deployment, firstly, a Multiple-population Gaussian Learning Particle Swarm Optimization (MGL-PSO) algorithm was proposed, with using the model of multiple population evolution to accelerate the algorithm convergence, as well as adding Gaussian learning strategy to avoid local optimum. Then according to the deployment model, with using Round Robin (RR) algorithm to initialize the MGL-PSO, a virtual machine deployment strategy aiming to load balancing was proposed. Through the simulation experiment in CloudSim, it validates that MGL-PSO has a higher convergence speed and load imbalance degree is reduced by 13% compared with PSO algorithm. In the two experimental situations, compared with the Opportunistic Load Balancing (OLB) algorithm, the load imbalance degrees of the proposed algorithm decrease by 25% and 15% respectively, and compared with the Greedy Algorithm (GA) the load imbalance degrees decrease by 19% and 7% respectively.

Key words: virtual machine deployment; Particle Swarm Optimization (PSO); load balancing; Gaussian learning; multiple population evolution

0 引言

基础设施即服务(Infrastructure as a Service, IaaS)是云计算中最为基础的服务类型。用户向 IaaS 服务商提出服务申请, IaaS 服务商通过把 CPU、内存、硬盘存储等资源通过虚拟化技术封装在虚拟机中, 并以虚拟机为单位提供给用户^[1]。云数据中心(Cloud Data Centre, CDC)大量使用这种虚拟化技术, 使多台虚拟机能同时部署在一台物理机中, 然而由于虚拟机和物理机配置存在差异性, 所以建设云数据中心需要解决虚拟机部署问题。

云计算是一种通过提供服务而获取利益的商业化模式,

并通过云数据中心提供高质量服务从而满足用户需求。Wickremasinghe 等^[2]证明, 通过使用负载均衡调度策略可以改善服务质量, 并且负载均衡策略的好坏决定改善的程度。Kleinberg 等^[3]证明, 当只考虑单一 CPU 资源的负载均衡调度时, 这种负载均衡调度问题(Load-balance Scheduling Problem, LSP)可在一个多项式时间内被规约成子集求和问题。文献[4]提出在实际环境中, 需要被考虑的资源不只 CPU 一种, 问题就会变得更加复杂, 因此, 在部署虚拟机时, 如何平衡物理机资源的负载是一个 NP-Hard 组合优化问题。目前, Eucalyptus^[5]、OpenNebula^[6]等, 大多开源的 IaaS 解决方案只是简单地使用贪婪算法进行虚拟机部署, 由于贪婪算法没

收稿日期: 2015-07-27; 修回日期: 2015-09-15。

作者简介: 杨靖(1991-), 男, 江西鹰潭人, 硕士研究生, 主要研究方向: 服务计算、计算机网络; 张宏军(1963-), 男, 江苏泰州人, 教授, 博士, 主要研究方向: 军用数据与知识工程; 赵水宁(1971-), 男, 江西南昌人, 副教授, 博士, 主要研究方向: 服务计算、计算机网络; 占栋辉(1988-), 男, 江西上饶人, 博士研究生, 主要研究方向: 智能算法、数据挖掘。

有考虑虚拟机和物理机的差异性,为了平衡物理机负载,在实际使用这些解决方案时通常由使用者自己编写算法。

迄今为止,已有研究者使用智能优化算法来解决虚拟机部署问题并取得了较好的效果,文献 [7] 提出一种基于蚁群优化算法的虚拟机部署策略;文献 [8] 提出一种基于遗传算法的虚拟机部署策略。粒子群优化 (Particle Swarm Optimization, PSO) 算法 [9] 相比其他几种智能优化算法,具有概念简单、易于实现、参数较少以及无需梯度信息等优势,因此,也有研究者使用 PSO 算法解决虚拟机部署问题。例如,文献 [10] 提出一种以节能为目标的基于 PSO 算法虚拟机部署策略;文献 [11] 提出一种多目标的基于 PSO 算法虚拟机部署策略,但 PSO 算法在处理复杂的问题时,仍然存在收敛速度慢、容易陷入局部最优等缺陷,因此,本文提出多种群高斯学习粒子群优化 (Multiple-population Gaussian Learning Particle Swarm Optimization, MGL-PSO) 算法,并与轮询 (Round Robin, RR) 算法结合得到一种以物理机负载均衡为目标的基于粒子群算法的虚拟机部署策略。最后,通过仿真实验, MGL-PSO 算法相比 PSO 算法在处理虚拟机部署问题时,收敛速度和搜索精度有显著提升,并且 MGL-PSO 算法在物理机负载均衡方面优于随机负载均衡 (Opportunistic Load Balancing, OLB) 算法和贪婪算法 (Greedy Algorithm, GA) 两种经典的负载均衡算法。

1 虚拟机部署模型

1.1 部署模型

针对云环境下的虚拟机部署,通过使用 n 个虚拟机和 m 个物理机组成的五元组来描述: $Q = (V, P, D, A, S_n)$ 。其中: V 表示 n 个虚拟机组成的虚拟机集合; P 表示 m 个物理机所组成的物理机集合; D 表示虚拟机部署问题的优化目标; A 表示使用的优化算法; S_n 表示虚拟机映射到物理机的资源部署向量。

部署模型的具体特征描述如下:

1) 虚拟机集合 $V = \{Vm_1, Vm_2, \dots, Vm_n\}$, 表示一个云数据中心需要进行部署的虚拟机,还可以进一步细化为 CPU、内存、硬盘存储、网络带宽 4 种资源。 $Vm_j = \{VCPU_j, VMem_j, VDisk_j, VBW_j\}$ ($j = 1, 2, \dots, n$), 其中: j 为虚拟机编号, $VCPU_j$ 、 $VMem_j$ 、 $VDisk_j$ 、 VBW_j 分别表示虚拟机需要的 CPU 大小、内存大小、存储大小和网络带宽。

2) 物理机资源集合 $P = \{Pm_1, Pm_2, \dots, Pm_m\}$, 每个物理机相互独立,物理机资源还可以进一步细化为 CPU、内存、硬盘存储和网络带宽 4 种资源。 $Pm_i = \{PCPU_i, PMem_i, PDisk_i, PBW_i\}$ ($i = 1, 2, \dots, m$), 其中: i 表示物理机编号, $VCPU_i$ 、 $VMem_i$ 、 $VDisk_i$ 、 VBW_i 分别表示物理机自身的 CPU 大小、内存大小、存储大小和网络带宽。

3) 资源部署向量 $S_n = (s_j)$ ($j = 1, 2, \dots, n$), 表示虚拟机 j 部署在物理机 s_j 上。若 $S_n = (4 \ 2 \ 4 \ 1 \ 3)$, 表示 1 号虚拟机部署在 4 号物理机上, 2 号虚拟机部署在 2 号物理机上, ..., 5 号虚拟机部署在 3 号物理机上。

4) 由资源部署向量可以得到资源使用矩阵 $E_{m \times n} = (e_{ij})$ ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 当 $i = s_j$ 且 $s_j \in S_n$ 时, $e_{ij} = 1$, 其余为 0。当 $e_{ij} = 1$ 表示虚拟机 j 部署在物理机 i 上。

5) 由资源使用矩阵得到虚拟机部署约束条件如下:

$$\begin{cases} \sum_{j=1}^n (VCPU_j \times e_{ij}) \leq PCPU_i \\ \sum_{j=1}^n (VMem_j \times e_{ij}) \leq PMem_i \\ \sum_{j=1}^n (VDisk_j \times e_{ij}) \leq PDisk_i \\ \sum_{j=1}^n (VBW_j \times e_{ij}) \leq PBW_i \end{cases}; \quad i = 1, 2, \dots, m \quad (1)$$

其含义是在物理机 i 上部署的所有虚拟机需要的 CPU 大小、内存大小、存储大小和网络带宽不能超过物理机 i 所能提供的大小。

1.2 负载均衡度量指标

由于虚拟机和物理机配置的差异性,云数据中心容易出现负载不均的情况,部分物理机上部署了过多的虚拟机,物理机可能会达到性能瓶颈,并且影响虚拟机的性能,而另一些物理机则可能负载较轻,物理机没有得到充分的利用。这种负载不均的情况直接影响云数据中心物理机的高效利用,造成 IaaS 服务商提供服务的质量下降。为了避免这种情况,需要通过合理的虚拟机部署策略保持物理机负载均衡。文献 [12-13] 提出了一些负载均衡调度的度量指标,本文通过参考这些度量指标,使用一种综合考虑 CPU、内存、硬盘存储、网络带宽 4 种物理机资源的度量指标来比较物理机负载均衡效果。为了得到度量指标,根据部署模型,考虑以下参数。

1) 当所有虚拟机部署完成时,由资源使用矩阵 $E_{m \times n}$ 物理机可以得到 Pm_i 的 CPU 利用率:

$$PCPU_i^U = \left[\sum_{j=1}^n (VCPU_j \times e_{ij}) \right] / PCPU_i \quad (2)$$

其中: Pm_i 内存利用率 $PMem_i^U$, 存储利用率 $PDisk_i^U$ 和带宽利用率 PBW_i^U 都可以用相同方式计算。

2) 所有虚拟机部署完成时,云数据中心物理机 CPU 平均利用率:

$$PCPU_{AVG}^U = \left(\sum_{i=1}^m PCPU_i^U \right) / m \quad (3)$$

云数据中心物理机内存平均利用率 $PMem_{AVG}^U$, 存储平均利用率 $PDisk_{AVG}^U$ 和带宽平均利用率 PBW_{AVG}^U 可以用相同方式计算。

3) 云数据中心物理机的 CPU 负载不均衡度:

$$NLBI_{CPU} = \sqrt{\left[\sum_{i=1}^m (PCPU_{AVG}^U - PCPU_i^U)^2 \right] / m} \quad (4)$$

其中 $NLBI_{CPU}$ 表示云数据中心物理机的 CPU 负载不均衡度,其值越小表示物理机的 CPU 负载越均衡。物理机的内存负载不均衡度 $NLBI_{Mem}$, 存储负载不均衡度 $NLBI_{Disk}$ 和带宽负载不均衡度 $NLBI_{BW}$ 可以用相同方式计算。

4) 云数据中心物理机综合负载不均衡度:

$$NLBI_{total} = \omega_1 NLBI_{CPU} + \omega_2 NLBI_{Mem} + \omega_3 NLBI_{Disk} + \omega_4 NLBI_{BW} \quad (5)$$

其中: ω_1 、 ω_2 、 ω_3 、 ω_4 分别表示物理机的 CPU 负载不均衡度的权值、内存负载不均衡度的权值、存储负载不均衡度的权值以及带宽负载不均衡度的权值。本文中 ω_1 、 ω_2 、 ω_3 、 ω_4 的初始值

为1, 根据实际情况的不同, 通过加大或者减小这些权值, 从而增强或者减弱权值对应的资源的负载性能。

$NLBI_{total}$ 就是本文用来比较物理机负载均衡效果的度量指标, 其值越小表示物理机的综合负载均衡效果越好。

5) 本文虚拟机部署问题就是求最小负载不均衡度的问题, 目标函数可以描述为:

$$DBest(s) = \min(NLBI_{total}(s)); s \in S \quad (6)$$

其中: 所有满足约束条件(1)的部署方案 s 组成部署方案域 S , $NLBI_{total}(s)$ 为部署方案 s 的综合负载不均衡度。

2 改进粒子群优化算法

2.1 算法原理

PSO 算法是一种基于迭代的启发式优化算法。算法原理可以简单描述为: 由 N 个没有体积和质量的粒子个体组成的粒子群在维度为 D 的空间进行搜索, 每一个粒子都有速度和位置两个属性, 算法通过速度来更新粒子位置, 而每个粒子位置代表了搜索空间中的一个解。本文算法考虑这些参数: 第 i 个粒子当前位置, $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$; 第 i 个粒子的速度, $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$; 粒子的历史最优位置, $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ ($pbest_i$), 以及粒子的全局最优位置, $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ ($gbest$)。在搜索过程中, 粒子每一次更新都必须参照两个最优位置, 一个是粒子个体当前发现的个体历史最优位置 $pbest_i$, 另一个是群体当前发现的全局最优位置 $gbest$, 如此重复更新, 直到满足条件, 最后找出最优解。具体更新公式表示如下:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 rand1_{id}^t (p_{id}^t - x_{id}^t) + c_2 rand2_{id}^t (p_{gd}^t - x_{id}^t) \quad (7)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (8)$$

其中: $i = 1, 2, \dots, N$; $d = 1, 2, \dots, D$; t 为迭代次数; ω 表示惯性权重; c_1 和 c_2 为加速常数; $rand1_{id}^t$ 和 $rand2_{id}^t$ 是 0 到 1 之间均匀分布的随机数; v_{id}^t 表示当前粒子 i 的速度; v_{id}^{t+1} 表示更新后粒子 i 的速度; x_{id}^t 表示当前粒子 i 的位置; x_{id}^{t+1} 表示更新后粒子 i 的位置。

2.2 多种群进化模式

MGL-PSO 算法使用多种群进化模式, 将粒子分为大小相等的 $PNum$ 个子群进行进化, 并且通过在粒子速度更新中加入所有子群的整体全局最优粒子以提高算法的收敛速度。MGL-PSO 算法的具体更新公式表示如下:

$$v_{ijd}^{t+1} = \omega v_{ijd}^t + c_1 rand1_{id}^t (p_{id}^t - x_{ijd}^t) + c_2 rand2_{id}^t \mu_1 (p_{igd}^t - x_{ijd}^t) + c_3 rand3_{id}^t \mu_2 (p_d^t - x_{ijd}^t) \quad (9)$$

$$x_{ijd}^{t+1} = x_{ijd}^t + v_{ijd}^{t+1} \quad (10)$$

其中: i, j, t 分别表示子群编号、粒子编号、迭代次数; p_{id}^t 表示在 t 次迭代后第 i 个子群中第 j 个粒子的历史最优位置; p_{igd}^t 表示在 t 次迭代后第 i 个子群的历史最优位置; p_d^t 表示在 t 次迭代后所有子群的整体历史最优位置。

式(9)与 PSO 算法速度更新式(7)相比, 增加了第4部分整体历史最优反馈。其主要目的是为了使多个子群中的全部粒子都能向整体历史最优位置移动, 从而提高算法的收敛速度, 并且为了降低当前历史最优粒子的影响, 保持粒子的多样

性, 避免陷入局部最优同时避免粒子速度过大增加影响因子 μ_1 和 μ_2 , $\mu_1 = 0.5$, $\mu_2 = 0.25$ 。

2.3 高斯学习机制

文献[4]通过采用高斯学习机制对全局最优粒子进行调整, 保持粒子的多样性避免陷入局部最优。MGL-PSO 算法通过采用高斯学习机制在多种群进化模式的基础上对每个子群的历史最优位置进行调整, 保持每个子群的粒子多样性, 避免子群陷入局部最优。

MGL-PSO 算法随机选择每个子群历史最优位置其中的一维 P_{ig}^d 进行高斯学习得到新的子群历史最优位置, 如果新的历史最优位置优于 P_{ig}^d , 则更新它; 否则保留。每一维都有相同的概率被选中, 所以在统计学上可以认为这一学习机制作用于每个子群历史最优位置的每一维。具体的高斯学习公式如下:

$$P_{ig}^d = P_{ig}^d + (Max - Min) \times \text{Gaussian}(\mu, \sigma^2) \quad (11)$$

其中: $i = \text{random}(1, PNum)$, $d = \text{random}(1, D)$, P_{ig}^d 表示第 i 个子群的历史最优位置, d 表示随机选择的某一维度, $[Min, Max]$ 为搜索区间, μ 为高斯随机数的平均值, σ^2 为高斯随机数的方差。在本算法中 $\mu = 0$, σ^2 的计算公式如下:

$$\sigma^2 = \frac{g}{2G} (Max - Min) \quad (12)$$

其中: g 为当前学习的次数, G 为每次迭代总共需要学习的次数。

3 基于 MGL-PSO 算法的虚拟机部署策略

3.1 粒子编码

在把 MGL-PSO 算法使用到虚拟机部署问题之前, 需要通过对粒子进行编码, 从而把问题的解空间和粒子的位置进行映射。在 PSO 算法中, 好的编码方式不仅能得到更好的搜索结果, 还有助于减少问题的搜索空间。PSO 算法在处理虚拟机部署问题时, 常见的编码方式有二进制编码和多值编码。在处理 n 台虚拟机在 m 台物理机上的部署问题时, 二进制编码的搜索空间为 $2^{m \times n}$, 而多值编码的搜索空间为 m^n 。由于多值编码的搜索空间更小, 并且二进制编码需要考虑同一台虚拟机是否在一个方案中被多次部署, 所以本文选用多值编码对粒子进行编码。

对粒子进行多值编码, 当第 i 个粒子当前位置为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 时, 对应的资源部署向量为 $S_n = (\text{floor}(x_{i1}), \text{floor}(x_{i2}), \dots, \text{floor}(x_{iD}))$, 本文通过这种方式, 将粒子群算法应用到虚拟机部署问题的求解中。

3.2 部署流程

根据部署模型, 式(1)为部署的约束条件, 式(5)为适应度函数, 具体的部署步骤如下:

轮询算法

1) 随机生成 N 个虚拟机请求序列, 通过 RR 算法分配这些虚拟机请求序列, 得到 N 种虚拟机部署方案, 从而得到 N 个粒子组成的初始种群。

2) 将粒子分为 $PNum$ 个子群, 每个子群 Num 个粒子。

3) 根据式(5), 计算初始种群的适应值, 得到个体最优解、子群最优解和全局最优解。

4) 根据式(9)和式(10)计算粒子的速度和位置。

5) 判断每个粒子的部署方案是否满足式(1)的约束条件,如果满足,则更新粒子的位置,并计算适应值,更新个体最优解和子群最优解;否则粒子保持原位置。

6) 对每个子群进行高斯学习,并更新全局最优解。

7) 判断是否达到最大迭代次数,如果是,则算法结束,并输出全局最优解,即虚拟机部署方案;否则返回步骤4)。

MGL-PSO 算法使用 RR 算法生成粒子,在 RR 算法的基础上进行粒子搜索,并通过在多种群进化基础上的高斯学习策略增加搜索的精度和收敛速度。另外,式(1)的约束条件保证了粒子的位置的可行性,确保输出的虚拟机部署方案为可行方案。

4 仿真实验

本文使用澳大利亚墨尔本大学的网格实验室和 Gridbus 项目推出的开源云计算仿真平台 CloudSim^[3]进行仿真实验参考亚马逊 EC2(Elastic Compute Cloud)^[4],并在此基础上进行扩展,得到云数据中心的物理机配置如表 1 所示,虚拟机配置如表 2 所示。

表 1 物理机配置

物理机类别	CPU (核心数)	内存/GB	硬盘 存储/GB	带宽/(Mb·s ⁻¹)
1	16	30	3 380	1 000
2	24	45	5 070	1 500
3	32	60	6 760	2 000

表 2 虚拟机配置

虚拟机类别	CPU (核心数)	内存/GB	硬盘 存储/GB	带宽/(Mb·s ⁻¹)
1	1	1.70	160	50
2	1	3.75	320	100
3	2	7.50	320	200
4	4	3.75	640	200
5	4	7.50	640	300
6	6	7.50	850	400
7	8	15.00	1 690	500

本文进行了两种仿真实验,分别对比 MGL-PSO 算法和原 PSO 算法在对虚拟机部署方案搜索时的收敛速度和精度,以及对比 MGL-PSO 算法、OLB 算法和 GA 在处理虚拟机部署问题时,物理机负载均衡的效果。参考文献[4]的分析,在这两种仿真实验中 MGL-PSO 算法参数为: $\omega = 0.5, c_1 = c_2 = c_3 = 2$ 。

4.1 收敛性比较

为了验证本文算法的收敛性,在 CloudSim 中建立一个由 50 台物理机所组成的数据中心,随机生成 100 台虚拟机,物理机按照表 1 的配置,虚拟机按照表 2 的配置。设置初始粒子数为 20,子群数为 4,MGL-PSO 算法和 PSO 算法都进行 500 次迭代,重复 10 次实验取平均值。图 1 为每 50 次迭代,PSO 算法与 MGL-PSO 算法的适应值对比。

从图 1 中可以看出 MGL-PSO 算法收敛速度明显快于 PSO 算法,并且最终的负载不均衡度比 PSO 算法降低了 13.1%,因此,在处理虚拟机部署问题时,通过引进多种群进化模式和高斯学习机制,能够提高粒子的收敛速度并且避免粒子陷入局部最优,从而获得更好的搜索结果。

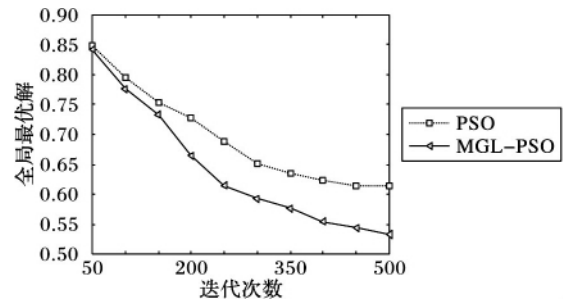


图 1 算法收敛性比较

4.2 综合负载不均衡度比较

为了验证算法的负载均衡效果,分别进行两种不同的场景下的仿真实验。

场景 1 20~200 个虚拟机需要在数据中心进行部署,物理机的数量设为虚拟机的一半;

场景 2 在 100 台物理机中部署 100~200 个虚拟机。

在这两种场景下,分别采用 MGL-PSO 算法、OLB 算法和 GA 对虚拟机进行部署,OLB 算法随机选择一台物理机进行虚拟机部署,GA 每次选取负载最小的物理机进行虚拟机部署。在 MGL-PSO 算法中,粒子的初始数为 20,子群数为 4,迭代次数为 500 次,场景 1 的实验结果如图 2 所示,场景 2 的实验结果如图 3 所示。从图 2 中可以看出,在解决不同规模的虚拟机部署问题时,OLB 算法的综合负载不均衡度最大,GA 优于 OLB 算法,而通过本文提出 MGL-PSO 算法进行虚拟机部署,其物理机负载不均衡度比 OLB 算法平均降低 25% 和比 GA 平均降低 19%。从图 3 中可以看出,在物理机数量相同的情况下部署不同数量的虚拟机时,MGL-PSO 算法的负载均衡效果比另外两种算法要好,其负载不均衡度比 OLB 算法平均降低 15% 和比 GA 平均降低 7%,因此,在处理虚拟机部署问题时,MGL-PSO 算法比这些经典的负载均衡算法具有更好的负载均衡效果。

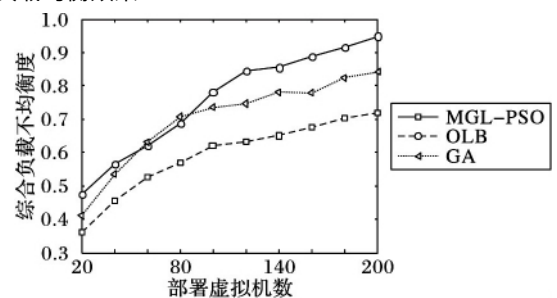


图 2 场景 1 物理机综合负载不均衡度比较

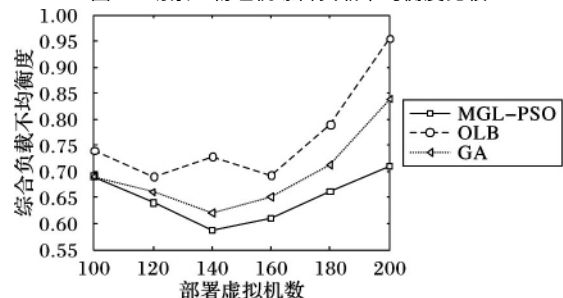


图 3 场景 2 物理机综合负载不均衡度比较

5 结语

本文在 PSO 算法的基础上,通过使用多种群进化模式提

高粒子的收敛速度,并在此基础上对子群进行高斯学习,防止搜索陷入局部最优,提出一种 MGL-PSO 算法。为了解决以物理机负载均衡为目标的虚拟机部署问题,把 RR 算法加入到 MGL-PSO 算法的初始化中,提出一种新的虚拟机部署策略。通过实验验证,该算法比 PSO 算法有更快的收敛速度和更好的搜索结果,并且它在综合负载均衡度方面,要优于 OLB 算法和 GA 这两种经典负载均衡算法,因此,基于 MGL-PSO 算法的虚拟机部署策略能够较好地解决云环境下的虚拟机部署问题。

参考文献:

- [1] OSTERMANN S, IOSUP A, YIGITBASI N, et al. A performance analysis of EC2 cloud computing services for scientific computing [C]// CloudComp 2009: Proceedings of the First International Conference on Cloud Computing. Berlin: Springer, 2010: 115 – 131.
- [2] WICKREMASINGHE B, CALHEIROS R N, BUYYA R. CloudAnalyst: a CloudSim-based visual modeller for analysing cloud computing environments and applications [C]// Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications. Piscataway, NJ: IEEE, 2010: 446 – 452.
- [3] KLEINBERG J, TARDOS É. Algorithm design [M]. Upper Saddle River, NJ: Pearson Education, 2006: 352 – 367.
- [4] NAKADA H, HIROFUCHI T, OGAWA H, et al. Toward virtual machine packing optimization based on genetic algorithm [C]// IWANN 2009: Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. Berlin: Springer, 2009: 651 – 654.
- [5] CARON E, DESPREZ F, LOUREIRO D, et al. Cloud computing resource management through a grid middleware: a case study with DIET and Eucalyptus [C]// Proceedings of the 2009 IEEE International Conference on Cloud Computing. Piscataway, NJ: IEEE, 2009: 151 – 154.
- [6] MILOJICIC D, LLORENTE I M, MONTERO R S. OpenNebula: a cloud management tool [J]. IEEE Internet computing, 2011, 15 (2): 11 – 14.
- [7] GAO Y, GUAN H, QI Z, et al. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing [J]. Journal of computer and system sciences, 2013, 79(8): 1230 – 1242.
- [8] 袁爱平, 万灿军. 云环境下基于改进遗传算法的虚拟机调度策略 [J]. 计算机应用, 2014, 34(2): 357 – 359. (YUAN A P, WAN C J. Virtual machine deployment strategy based on improved genetic algorithm in cloud computing environment [J]. Journal of computer applications, 2014, 34(2): 357 – 359.)
- [9] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE, 1995: 1942 – 1948.
- [10] JEYARANI R, NAGAVENI N, RAM R V. Design and implementation of Adaptive Power-Aware Virtual Machine Provisioner (APA-VMP) using swarm intelligence [J]. Future generation computer systems, 2012, 28(5): 811 – 821.
- [11] XU B, PENG Z, XIAO F, et al. Dynamic deployment of virtual machines in cloud computing using multi-objective optimization [J]. Soft computing, 2014, 19(8): 2265 – 2273.
- [12] WOOD T, SHENOY P J, VENKATARAMANI A, et al. Black-box and gray-box strategies for virtual machine migration [C]// Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation. Berkeley, CA: USENIX Association, 2007: 17 – 17.
- [13] TIAN W, XU M, CHEN Y, et al. Prepartition: a new paradigm for the load balance of virtual machine reservations in data centers [C]// Proceedings of the 2014 IEEE International Conference on Communications. Piscataway, NJ: IEEE, 2014: 4017 – 4022.
- [14] ZHAN Z H, ZHANG J, LI Y, et al. Adaptive particle swarm optimization [J]. IEEE transactions on systems, man, and cybernetics, part B: cybernetics, 2009, 39(6): 1362 – 1381.
- [15] CALHEIROS R N, RANJAN R, BELOGLAZOV A, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: practice and experience, 2011, 41(1): 23 – 50.
- [16] Amazon EC2[EB/OL]. [2015-06-20]. <http://aws.amazon.com/ec2/>.

Background

YANG Jing, born in 1991, M. S. candidate. His research interests include service computing, computer network.

ZHANG Hongjun, born in 1963, Ph. D., professor. His research interests include military data and knowledge engineering.

ZHAO Shuining, born in 1971, Ph. D., associate professor. His research interests include service computing, computer network.

ZHAN Donghui, born in 1988, Ph. D. candidate. His research interests include intelligent algorithm, data mining.

(上接第 112 页)

- [22] SINGH M, SURI P K. QPSMax-MinMin-Min: a QoS based predictive Max-Min, Min-Min switcher algorithm for job scheduling in a grid [J]. Information technology journal, 2008, 7(8): 1176 – 1181.
- [23] 涂雪芹. 基于蜂群算法的多目标加权优化负荷频率控制研究 [J]. 计算机测量与控制, 2015, 32(2): 648 – 651. (TU X Q. Research of multi-objective weighted sum optimization for load frequency control based on artificial bee colony algorithm [J]. Computer measurement and control, 2015, 32(2): 648 – 651.)

Background

This work is partially supported by the National Natural Science Foun-

dation of China (71271071), the Key Project of National Natural Science Foundation of China (71490725) and the National High Technology Research and Development Program (863 Program) of China (2011AA040501).

NI Zhiwei, born in 1963, Ph. D., professor. His research interests include artificial intelligence, machine learning, cloud computing.

LI Rongrong, born in 1990, M. S. candidate. Her research interests include swarm intelligence algorithm, cloud computing.

FANG Qinghua, born in 1990, M. S. candidate. Her research interests include swarm intelligence algorithm, data mining.

PANG Shanshan, born in 1989, M. S. candidate. Her research interests include swarm intelligence algorithm, data mining.