

# 云计算和多维 QoS 环境中基于蚁群优化算法在虚拟机资源负载均衡问题中的研究

张 牧

(成都体育学院信息技术中心 成都 610041)

**摘 要** 针对云计算环境中虚拟机资源负载均衡问题,并为实现云计算下虚拟机资源负载均衡高效调度以满足用户的 QoS 需求,提出了一种基于多维 QoS 实现负载均衡的虚拟机资源调度方法。首先,在云计算环境下建立多维 QoS 网络环境的数学模型;然后,提出一种基于蚁群优化的优化算法,用于实现云计算环境中虚拟机资源高效调度;最后,在云仿真平台 CloudSim 上进行仿真实验。实验结果表明,相对于其他资源调度算法,所提算法能高效解决云计算下虚拟机资源调度问题,减少虚拟机资源负载均衡偏差,具有更好的性能,能完全满足云计算下和多维 QoS 环境下虚拟机资源负载均衡的需求。

**关键词** 蚁群优化,虚拟机,云计算,QoS,负载均衡

**中图分类号** TP301 **文献标识码** A

## Research of Virtual Machine Load Balancing Based on Ant Colony Optimization in Cloud Computing and Multi-dimensional QoS

ZHANG Mu

(Information Technology Center, Chengdu Sport University, Chengdu 610041, China)

**Abstract** Aiming at Virtual Machine load balancing problem in Cloud computing, in order to realize the efficient scheduling of Virtual Machine load balancing in Cloud computing and meet the user's QoS need, a Virtual Machine load balancing method based on Multi-dimensional QoS is proposed. Firstly, the mathematical model of cloud resource scheduling is built, Then, a Virtual Machine load balancing algorithm based on ant colony optimization in Cloud computing and Multi-dimensional QoS is introduced, Finally, the CloudSim simulation experiment shows the algorithm can solve the Virtual Machine load balancing problem effectively, can reduce the load balance deviation and satisfy the need of Virtual Machine load balancing in Cloud computing.

**Keywords** Ant colony optimization, Virtual machine, Cloud computing, QoS, Load balancing

## 1 引言

云计算环境作为一种基于网络分布式处理的全新的网络服务方式,由并行处理、分布式处理、网络演化而来,包含基础即服务(IaaS)、平台即服务(PaaS)和软件即服务(SaaS)以及依托于网络的各种技术和趋势<sup>[1]</sup>。云计算环境所提供的服务、资源和针对的用户群体是庞大的<sup>[2]</sup>,因此,在云环境下任务与资源调度是云计算技术的一个重要组成部分,资源调度效率直接影响到整个云环境的工作效率和性能。当前云计算环境下的任务与资源调度方法都是利用虚拟机上的资源调度策略和相关的调度技术算法相结合为虚拟机内部资源、应用和服务进行调度,实践证明,当前任务调度策略都普遍缺乏高效性和精确性<sup>[3]</sup>。蚁群算法的灵感和原理是基于蚂蚁寻觅食物的行为的模拟,其在解决 NP-hard 问题中,离散问题的方法上具有较高的协作性、并发性、鲁棒性和易扩展性,是一种以群体处理为基础的搜索自适应算法,尤其是在动态变化环境下它的健壮性和灵活性能很好地展现。

目前,针对云计算环境下的资源调度通常是采用一定的

调度策略为用户任务制定对应的虚拟机,将任务平均执行时间作为目标函数,采用蚁群算法来对云计算任务调度求解;或是构造将任务平均完成时间和总任务完成时间作为调度适应度函数,在此基础上提出了一种双适应度函数的改进遗传算法并求其解<sup>[5]</sup>。

上述工作均采用启发式算法实现对云计算环境下的资源调度问题进行求解,但仍然存在未考虑用户定义的服务质量(Quality of Service, QoS)和实现负载均衡。所以本文在上述工作的基础上,首先定义了云计算资源调度多维 QoS 数学模型,然后设计了一种改进的蚁群算法对其进行求解,最后通过仿真实验证明文中方法的优越性。

## 2 虚拟机资源负载均衡建模

### 2.1 性能匹配距离

在服务器上部署虚拟机时,首先要确定服务器的性能是否能满足此虚拟机的性能要求。设  $\{Ser_{1p}, \dots, Ser_{mp}\}$  为服务器  $Ser_j$  同类剩余性能集合,  $\{Vm_{1p}, \dots, Vm_{ip}\}$  为虚拟机  $Vm_i$  相对应的同类需求性能集合。如  $(Ser_{jp} - Vm_{ip}) (p=1, 2, \dots,$

本文受成都体育学院院级课题(YJ1103)资助。

张 牧(1978—),男,硕士,工程师,主要研究方向为计算机网络、软件工程, E-mail: zhangmu@cdsu.edu.cn.

k) 出现负数值( $k$  为参考个数)时,则认为服务器  $Ser_j$  无法满足虚拟机  $Vm_i$  的性能要求,从而不再将虚拟机  $Vm_i$  部署在服务器  $Ser_j$  上;如为正值,则需要量化测评服务器与各虚拟机之间的匹配关系,来为是否进行虚拟机部署提供参考依据。为此本文提出匹配距离  $Mat_{vm_i, ser_j}$  (以下简称为  $Mat_{ij}$ ) 的概念。

首先,为精确比较服务器与虚拟机,对虚拟机和服务器性能进行归一化处理。服务器性能归一化算法为:  $S_{ip} = \frac{Ser_{ip} - Ser_p^{\min}}{Ser_p^{\max} - Ser_p^{\min}}$ , 其中  $Ser_p^{\max}$  为服务器同类性能  $p$  的最大值,  $Ser_p^{\min}$  为最小值;同样虚拟机期待性能归一化算法为:  $V_{ip} = \frac{Vm_{ip} - Vm_p^{\min}}{Vm_p^{\max} - Vm_p^{\min}}$ 。若最大值与最小值相同,则参数均视为 1。

性能匹配距离是指待部署的虚拟机性能和服务器空闲之间的欧几里得距离,其算法为:  $Mat_{ij} = \sqrt{\sum_{p=1}^k (S_{ip} - V_{ip})^2}$ 。为了更好地满足下文算法,对  $Mat_{ij}$  设置下限为 0.1,若匹配距离小于下限则均视为 0.1,以防止距离为 0 的现象出现。

在批量部署中,匹配距离是一个变化的可预测值。若将虚拟机  $Vm_i$  部署在服务器  $Ser_j$  上,则在考虑  $Ser_j$  与其他虚拟机的匹配距离时,需使用  $Ser_j$  部署  $Vm_i$  后的剩余性能做计算。

## 2.2 QoS 数学模型

云环境下虚拟机资源负载均衡器即在云计算环境下根据一定的资源调度规则,在不同的执行任务  $T = \{t_1, t_2, \dots, t_n\}$  之间进行虚拟机资源  $R = \{r_1, r_2, \dots, r_m\}$  调度的过程。

将用户对任务 QoS 的需求表示成时间、费用、安全性和可靠性,并将其通过函数表示成隶属度。

(1) 时间隶属度  $Sch(i, j)$ :

$$Sch(i, j) = \begin{cases} 1 - \frac{ET(i, j)}{ET_{ij}}, & ET(i, j) > ET_{ij} \\ 1, & \text{其它} \end{cases} \quad (1)$$

式中,  $ET_{ij}$  为任务  $t_i$  在虚拟机  $r_j$  上的预期执行时间,  $ET(i, j)$  为任务  $t_i$  在虚拟机  $r_j$  上的实际执行时间。

(2) 费用效应隶属度函数  $Bug(i, j)$ :

$$Bug(i, j) = \begin{cases} 1 - \frac{Cost(i, j)}{Cost_{ij}}, & Cost(i, j) > Cost_{ij} \\ 1, & \text{其它} \end{cases} \quad (2)$$

式中,  $Cost_{ij}$  是任务  $t_i$  在虚拟机  $r_j$  产生的预期费用,  $Cost(i, j)$  是任务  $t_i$  在资源  $r_j$  产生的实际费用。

(3) 安全性隶属度函数  $Sa \int(i, j)$ :

$$Sa \int(i, j) = \begin{cases} 1 - \frac{TSec_{ij} - Sec_{ij}}{Sec_{\max} - Sec_{ij}}, & TSec_{ij} > Sec_{ij} \\ 1, & \text{其它} \end{cases} \quad (3)$$

式中,  $TSec_{ij}$  是任务  $t_i$  对虚拟机  $r_j$  提出的期望安全等级,  $Sec_{ij}$  为虚拟机  $r_j$  真实能为任务  $t_i$  提供的安全等级,  $Sec_{\max}$  为任务设置的最高安全等级。

(4) 可靠性隶属度函数  $Reb(i, j)$ :

$$Reb(i, j) = \begin{cases} 1 - \frac{\exp(SRel_{ij})}{\exp(TRel_{ij})}, & TRel_{ij} > SRel_{ij} \\ 1, & \text{其它} \end{cases} \quad (4)$$

式中,  $TRel_{ij}$  是任务  $t_i$  对虚拟机  $r_j$  上的期望可靠性,  $SRel_{ij}$  是虚拟机  $r_j$  实际能为任务  $t_i$  提供的实际可靠性。

(5) 4 维-QoS 目标函数:

$$fit(i, j) = w_1 Sch(i, j) + w_2 Bug(i, j) + w_3 Sa \int(i, j) + w_4 Reb(i, j) \quad (5)$$

式中,  $w_1, w_2, w_3, w_4$  分别表示时间、费用、安全性、可靠性。

## 3 蚁群优化算法

### 3.1 蚁群算法基本原理

蚁群算法(Ant Colony Algorithm, ACA),是一种在图中寻找最优化路径的启发式智能学习算法。该算法通过设定的智能蚂蚁进行网络节点间路由寻址,遴选出源节点到目标节点间的最短路径,以最大化地节约网络资源。但是网络情况不完全同于真实蚂蚁寻找食物的路径状况。当大量的模拟蚂蚁在相同的网络路径上留下信息素,就会造成该条路径上信息素过多,使得后续的模拟蚂蚁不会选择新的路径,而全部选择到此条路径上。如此,当算法完成一次路由后,找到的最优路径也许只是局部最优,而不是全局最优<sup>[6]</sup>。为了避免以上情况发生,经科学分析:动物界蚂蚁分泌液随着浓度的逐渐增大,挥发速度会迅速加快。整个算法进行过程中,使用禁忌表  $tabu_k (k=1, 2, \dots, m)$  来记录蚂蚁  $K$  所走过的路径,并随着蚂蚁不断运动变化过程做动态调整,用  $P_{ij}^k(t)$  表示第  $K$  只蚂蚁在  $t$  时刻选择下一目标网络节点  $j$  的状态转移概率<sup>[7]</sup>。

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^k(t) \eta_{ij}^k(t)}{\sum_{n \in allowed_k} \tau_{in}^k(t) \eta_{in}^k(t)}, & j \in allowed_k \\ 0, & \text{other} \end{cases} \quad (6)$$

### 3.2 蚁群优化算法

本文在蚁群算法的基础上,为其增加一个网络节点审查机制,提出了一种能够得到网络节点的最优路径的蚁群优化算法。改进后的蚁群算法步骤如下:

Step1 对各节点信息素进行初始化,将蚂蚁任意地放置在各独立的网络节点上;

Step2 蚂蚁按照迁移概率规则,进入下一个网络节点,将选中的网络节点加入表  $tabu_k$  中,最后将所有的网络节点都加入该表;

Step3 记录完所有蚂蚁的表  $tabu_k$  中节点路径顺序,进行对比,得出本次迭代最优路径;

Step4 不断执行前面 3 步操作,直到迭代次数达到  $NC_{MAX}$  次,结束循环迭代;

Step5 通过节点审查机制,判断节点是否退出网络,如果有,则将路径中包含有退出节点的解从解集中删除;如果没有,则进入下一步;

Step6 在所有的解中,进行路径长度比较,路径最短的解作为全局最优解。

其流程图如图 1 所示。

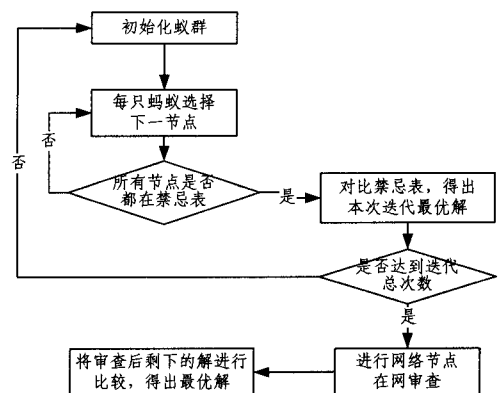


图 1 蚁群优化算法程序的流程图

为了更好的贴近网络实时性,使用  $Test=rand\_per(NC\_MAX)$  产生与迭代次数等量的随机数组,将这随机数组  $Test$  中的数值与最优路径中的节点标识比较,如果相等,则认为该节点退出网络。

### 3.3 基于多维 QoS 资源调度算法

根据以上蚁群优化算法推导得出如下基于多维 QoS 的资源调度算法,算法输入:任务集  $T=\{t_1, t_2, \dots, t_n\}$ 、虚拟机集  $R=\{r_1, r_2, \dots, r_m\}$ 、蚁群  $Ant$ , 蚂蚁的种群规模  $K$ , 最大迭代次数  $I_{max}$ , 4 维-QoS 目标函数权值  $w_1, w_2, w_3$  和  $w_4$ , 虚拟机选择概率参数  $a$  和  $b$ , 信息素挥发率  $\eta$ 。

算法输出:一个  $1 * n$  的向量,向量的每个分量值对应了分配给该任务的虚拟机号;

初始化:初始迭代次数  $I=1$ ;

(1)对所有虚拟机  $r_j (1 \leq j \leq m)$  信息素进行初始化;

(2) $K$  只蚂蚁随机选择某台虚拟机分配给第一个任务;

(3)第  $k$  只蚂蚁根据式(6)为下一个任务分配虚拟机;

(4)当第  $k$  只蚂蚁完成所有任务  $t_i (1 \leq i \leq n)$  调度时,对所调度的所有虚拟机进行局部信息素更新;

(5)当  $K$  只蚂蚁均完成所有任务调度时,对最优分配方案选择的虚拟机进行全局信息素更新;

(6)当某只蚂蚁对应的任务调度方案计算的 QoS 目标值连续两次迭代未发生明显变化时,采用遗传变异因子对其进行变异;

(7)对当前迭代次数  $I$  进行判断:

如果达到最大值  $I_{max}$ , 则算法结束,输出最优解;

否则  $I=I+1$ , 返回(2)重新开始新一轮的迭代。

## 4 仿真实验与分析

本文采用澳大利亚墨尔本大学的网络实验室和 Gridbus 项目提出的云仿真平台 CloudSim 作为试验仿真工具,使用版本为 CloudSim-2.1.1。通过对 VmAllocationPolicy 类中的管理虚拟机放置方法 `allocateHostForVm(Vm vm)` 进行编写,实现虚拟机放置算法,并同时需要相应修改 Vm 和 Datacenter 类。重新编译 CloudSim, 获得仿真环境,并在此基础上编写仿真程序。算法参数设置如下:蚁群规模为 20, 4 维-QoS 目标函数权值  $w_1=0.2, w_2=0.3, w_3=0.22$  和  $w_4=0.28$ , 资源选择概率参数  $a=0.7, b=0.3$ , 信息素挥发率  $\eta=0.4$ , 迭代次数  $I_{max}=100$ 。

采用文中蚁群优化算法、经典任务调度算法 Min-min 算法、Max-min 算法以及一种基于免疫克隆的算法中方法对实例进行仿真,对所得出的负载均衡离差进行比较,负载均衡离差反映了系统资源根据自身性能而实现的利用率,目的是实现高效的负载均衡。负载均衡离差可以通过式(7)算出。

$$\phi = \sqrt{\frac{\sum_{i=1}^m (I B_i - L \bar{B}_i)^2}{m-1}} \quad (7)$$

仿真得到的负载均衡离差如图 2 所示。

从图中可以看出,随着任务数的增加,负载均衡离差呈增加趋势,但文中蚁群优化算法所得到的值较其他 3 种方法偏

低,平均值为 0.175, Min-min 方法、Max-min 方法以及一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法得到的平均负载均衡离差分别为 0.218、0.209 和 0.195, 文中方法相对其分别提高了 19.7%、16.3% 和 10.3%, 说明了文中方法较好地实现了资源负载均衡。

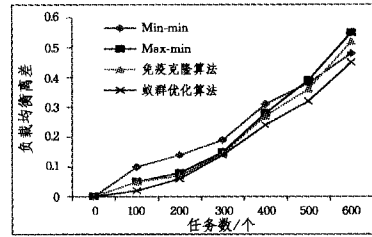


图 2 负载均衡离差

**结束语** 虚拟机资源负载均衡是云计算环境下的一个关键问题, QoS 是用户定义的服务质量, 因此, 针对具有 QoS 约束的虚拟机资源负载均衡问题, 首先定义了虚拟机资源负载均衡问题的数学模型, 然后提出了一种基于蚁群优化算法用于实现虚拟资源的调度方法。通过 CloudSim 平台的仿真实验证明文中方法有效地实现了云环境下虚拟机资源负载均衡, 并具有较高的执行效率和较小的负载均衡离差。然而, 本文算法只考虑了虚拟机的负载均衡问题, 而在实际应用中情况比较复杂, 还有更多因素需要考虑, 比如任务的执行速度、调度的时间、计算成本、用户请求的类型对虚拟机资源的调度影响以及能耗问题, 因此还有待作进一步的研究, 以使在云计算和多维 QoS 环境中虚拟机资源的调度在综合性能上达到最优。

## 参考文献

- [1] 华夏渝, 郑骏, 胡文心. 基于云计算环境的蚁群优化计算资源分配算法[J]. 华东师范大学学报: 自然科学版, 2010, 1(1): 127-134
- [2] Tian Wen-hong, Zhao Yong, Zhong Yuan-liang, et al. Dynamic and integrated Load-Balancing Scheduling Algorithm for Cloud Data Centers[C]// Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on. IEEE Conference Publications, 2011: 311-315
- [3] 《虚拟化与云计算》小组. 虚拟化与云计算[M]. 北京: 电子工业出版社, 2009
- [4] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240-1245
- [5] 刘万军, 张孟华, 郭文越. 基于 MPSO 算法的云计算资源调度策略[J]. 计算机工程, 2011, 37(11): 42-48
- [6] 史恒亮, 白光一, 唐振民, 等. 基于蚁群优化算法的云数据库动态路径规划[J]. 计算机科学, 2010, 37(5): 143-145
- [7] Wu Yong, Ma Lin, Yang Jing. Ant colony algorithm for optimal resources configuration of equipment main-tenance support[D]. Department of System Engineering of Engineering Techology Beijing University of Aernautics and Astronautics, 2011: 793-795
- [8] 孙大为, 常桂然, 李凤云, 等. 一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法[J]. 电子学报, 2011, 39(8): 1824-1831