

# 多智能体遗传算法在云计算 负载均衡中的应用研究

付天驰

(辽宁工程职业学院, 辽宁 铁岭 112008)

**摘要** 由于云计算系统具有诸多特点, 因此使得云计算系统的负载均衡更复杂, 难以有效地实现。本文采用多智能体内遗传算法(MAGA)来解决负载均衡, 以通过参数平衡云计算中的问题。测试结果表明, 云计算系统加载MAGA构建的平衡模型可以有效地避免由过度依赖性参数指标引起的单点故障问题, 并且在系统执行效率和系统平衡方面取得了良好的效果。

**关键词** 多智能体遗传算法 云计算负载均衡 虚拟仿真技术 静态规划算法

**中图分类号:** TP3

**文献标识码:** A

**文章编号:** 1007-0745(2022)03-0016-03

云计算是基于大量计算机计算出的资源。不同的应用系统可以获得计算能力、存储能力和各种软件服务。云计算系统作为一种具有竞争性和不确定性的任务处理方式, 具有高度的动态性。因此, 资源规划和任务管理是云计算中的关键问题<sup>[1]</sup>。现有算法, 如静态投票方法和随机方法, 显然不适合这种大规模的负载均衡, 因为它易于导致一个点故障。动态算法包括UT策略和来自负载均衡策略的弹簧策略, 仅考虑单个参数的影响, 并且当影响增长时, 其执行效率将下降。此外, 虽然人们设法解决了加载余额的问题, 但随着数据和尺寸的增加, 它们的收敛速度将受到影响。遗传算法在较长的搜索时间内比较缓慢, MAGA可以在解决较大的尺寸功能时快速找到高质量的解决方案<sup>[2-3]</sup>。本文的目的是使用MAGA来解决负载均衡并平衡云计算中的问题。插入各种不同的模拟环境, 参数选择的培训是在各种测试环境中进行的, 从而获得相对最佳、更通用的校正参数, 以及获得平衡策略的负载。在整个测试中, MAGA生成的充电策略、UT策略和弹簧策略应用于相同的模拟环境, 比较最大CPU使用率、平均CPU使用率、最大内存率和平均内存使用, 分析他们的优点和缺点。测试表明, 对于大型动态云系统, MAGA产生的负载策略, CPU的最大使用和最大内存使用率较低, 并且可以保证整个系统的稳定性和强度。

目前, 云计算已经成为一种非常流行的计算机商

业模式, 它是并行计算、分布式计算和网络计算的发展。云计算是一种可以独立管理的虚拟计算资源, 因此, 绘制适当的关系和资源图, 即Rational Task Plan, 以动态有效地管理资源。根据服务水平协议, 它是云计算机器的重要组成部分。理论上, 每个用户都有一台虚拟裁缝机, 互不接触。每个物理主机将加载一个或多个虚拟机, 以确保用户应用程序的独立性。因此, 云计算的任务就是规划。在虚拟机中, 保持虚拟集群的负载均衡是一种有效的算法。

云计算虚拟集群负载均衡的研究刚刚起步。本文基于对偶遗传算法规划, 但优化的目标只是考虑负责任务的时间。使用算法完成整个任务只是最短的时间, 因为云计算的中心思想是实现廉价高效的计算, 除了与任务相关的时间, 整个计算过程都会消耗资源和能源, 系统中还存在需要进一步研究的问题。智能算法是通过模拟或揭示某些自然现象过程和思想而发展起来的, 不仅内容涉及数学, 还以物理学、生物学、人工智能、神经科学和统计力学为基础, 为解决复杂问题提供了新的思路 and 手段, 将动态联盟的思想引入到文献中, 提出了一种基于粒子群优化的任务分配算法, 该算法在网络环境中实现了多种均衡。

本文提出了一种自适应变量遗传算法(GA)来解决虚拟人群的负载问题, 使虚拟累计使用值最高。仿真结果表明, 该算法快速高效, 具有实际应用价值。

★基金项目: 辽宁省教育厅2021年度科学研究经费项目, 项目编号: LJKZ1312; 辽宁省教育科学“十三五”规划2020年度立项课题, 项目编号: JG20EB321。

表1 GA与MAGA在遗传操作上的对比

	GA	MAGA
个体	同构	可异构
信息交互方式	被选择后, 进行交叉操作	可获知四个区域信息, 并更新自身
遗传算子	选择、交叉、变异	邻域竞争、正交交叉、变异、自学习
自学习	无	有
进化	无目标进化	有目标进化
竞争	轮赌选择	与邻域发生作用

### 1 关于本研究的相关工作

虚拟仿真技术充分利用了用户的基础设施, 实现了整个应用系统。因此, 越来越多的人开始关注虚拟存储服务器的使用。为了实现各种信息、服务和资源的高可用性, 必须使用负载均衡设备来协调和管理大量的虚拟机。最终目的是研究云计算中虚拟终端的负载均衡问题。本实用新型非常有效地提高了计算机资源的利用率, 有效的负载均衡算法要求系统在高效的响应时间内智能地在不同虚拟机之间分配负载。云环境的计划与传统的资源规划有许多不同的情况。首先, 规划目标不同于传统手段的规划目标。传统规划是通过物理方式来工作, 属于精细规划; 云计算环境中的规划对象是属于原粮规划的虚拟机资源, 传输的数据量也很大。

研究人员对此主题进行了许多研究, 提供了各种静态、动态和混合资源规划策略。静态规划算法包括: ISH算法、MCP算法和ETF算法, 即BNP(Limited Number Processing SORS), 相对于具有高效网络的分布式环境, 由于虚拟云计算的应用单元, 团体需求复杂多样, 必须按要求提供服务费用。此外, 很多研究人员也针对这个课题研究了相应的启发式算法, 但随着问题的不断增多, 启发式算法的影响并不那么明显。

### 2 多智能体遗传算法

从代理商的角度来看, 遗传算法中的个体被认为是部分认知、竞争、合作和自学能力, 通过代理人、环境和每个经纪人之间实现全球最佳目标之间的互动。多智能体遗传算法的实施机制与遗传算法截然不同, 主要纳入个人之间的互动、合作和自我研究, 有学者证明, 多智能遗传算法具有更快的收敛速度, 对于求解高维函数的优化问题非常有用<sup>[4-5]</sup>。

#### 2.1 个体生存环境

与GA类似, MAGA继续在个体上工作。在MAGA

中, 每个个体都被视为一个代理, 每个代理都有自己的特点, 可以感知周围环境, 自主运作, 还可以影响和改变环境, 所有代理都位于代理网格中。

#### 2.2 遗传操作算子

MAGA中的遗传算子主要包括邻域竞争算子、变异算子、正交内插交叉算子, 邻域正交交叉算子实现了Agent之间的协作。自变异算子和自变异算子利用Agent的知识来实现Agent的行为。

#### 2.3 MAGA与GA的比较

表1对GA与MAGA在信息交互方式、遗传算子、自学习能力等方面进行了对比分析。以函数优化为例, 对MAGA与GA的性能进行比较, 取 $n=20$ , 设优化函数为:

$$F(x) = \sum_{i=1}^n x_i \sin \sqrt{|x_i|}, x_i \in [-500, 500]$$

通过对两种算法执行10次后得到的最优函数值的比较, 表明MAGA的执行效果明显优于传统的遗传算法。

### 3 负载均衡模型

每个用户请求用户信息的主要参数(ReqPerHrPerUser, ReqSize, ReqCPU, ReqMemory, Count)。其中: ReqPerHrPeruser表示每小时平均互联网用户数; ReqSize是用户组中每个用户的每个请求的大小。ReqCPU是指CPU执行请求的使用率, 以百分比表示, 与2.4GHz的CPU内核相对应; ReqMemory是指在M中执行请求所使用的内存量; 计算每分钟发送请求的平均数。

为了解决网格资源规模急剧增加的问题, 提出了一种资源调度模型。基于用户请求标准的分组策略, 组中的每个参数都有一个最大值, 每个组中所有用户的所有参数的总和不能超过由组指定的最大值, 该最大值可分为一个组, 该组可根据时间序列重新设置参数。

#### 3.1 负载均衡模型的建立

负载均衡模型主要涉及适应度函数的设计。根据

组策略, 主机上的每个虚拟机 (VM) 资源对应于用户组策略请求。主机包含多个虚拟机, 每个虚拟机可以被分配给多个组。每个数组可以被描述为一个数组 (ReqPerHrPerUser, ReqSize, ReqCPU, ReqMemory, Count)。数组中的第四个参数是数组中每个请求所消耗的内存。因此, 在执行一组任务后, 虚拟机首先加载  $u_j$  内存。

$$Ml_i = M_i + Vmp_i / Vm_i \times 100\%$$

如果  $Vmp_i$  和  $Vm_i$  是固定的, 那么第三个参数是集群中每个请求的平均 CPU 消耗量, 即使在集群任务完成后, 虚拟机也会加载 CPU。

$$Cl_i = C_i + Vmc_i / Vc_i \times 100\%$$

其中:  $Vmc_i$  和  $Vc_i$  是常数,  $C_i$  为  $Vm_i$  在执行任务之前剩余的 CPU 百分比。

基于内存消耗与 CPU 消耗,  $Vm_i$  上的总负载:

$$Vl_i = w \times Ml_i + v \times Cl_i$$

其中,  $w$  和  $v$  是权重因子, 且满足  $w+v=1$ 。

### 3.2 编码

假设有 10 个用户组 {Group0, Group9, ……Group9} 和 30 个虚拟机 {VMU}, 每个用户组被视为一维。

让另一个 10 维数据 ( $x_0 \ x_1 \dots x_9$ ), 其中  $x_i$  对应一个面板, 然后  $x_i$ , 可以选择 30 和 30 的虚拟 25, 然后采取每一  $x_i$  长度编码 5, 其中:

00001—VM\_0

00010—VM\_1

11110—VM\_29

若采用二进制多维编码方式, 那么对整个系统的编码即是 {00000,0000,0000}, 则对拥有 10 个用户组、30 台 VM 资源的系统的一个可行解如下:

{00001,00100,10010,00110,10010,

11000,11100,00010,01000,10000}

故对于一个含有  $n$  个用户组、 $M$  个 VM 虚拟资源的系统, 问题的求解维数即是  $n$ , 每维个体的编码长度为  $\log_2 M$ 。

### 4 算法步骤

Step1: 随机生成  $L_{SIZE}^2$  个 Agent, 初始化  $L^0$ , 更新  $Best^0$ , 令  $t \leftarrow 0$ 。

Step2: 对  $L^1$  中的每个 Agent 执行邻域竞争算子得到  $L^{1+1/3}$ 。

Step3: 对  $L^{1+1/3}$  中的每个 Agent, 若  $U(0,1) < P_c$ , 则将邻域正交交叉算子作用在其上, 得到  $L^{1+1/3}$ 。

Step4: 对  $L^{1+1/3}$  中的每个 Agent, 若  $U(0,1) < P_e$ , 则将

变异算子作用在其上, 得到  $L^{t+1}$ 。

Step5: 从  $L^{t+1}$  中找出  $CBest^{t+1}$ , 并将自学习算子作用在其上。

Step6: 若  $Energy(CBest^{t+1}) > Energy(Best^t)$  成立, 则令  $Best^{t+1} \leftarrow CBest^{t+1}$ ; 否则  $Best^{t+1} \leftarrow Best^t$ ,  $CBest^{t+1} \leftarrow Best^t$ 。

Step7: 如果终止条件满足, 输出  $Best^t$  并停机; 否则, 令  $t \leftarrow t+1$  并转 Step2。

其中,  $L^t$  表示第  $t$  代的 Agent 网格,  $L^{t+1/3}$  和  $L^{t+1}$  是  $L^t$  和  $L^{t+1}$  的中间代 Agent 网格;  $Best^t$  是  $L^t$  中最优的 Agent,  $CBest^t$  表示  $L^t$  中最优的 Agent;  $P_c$  和  $P_m$  为预先设定的参数, 分别是邻域正交交叉算子和变异算子的执行概率。

### 5 结语

本文提供了一种在云计算中使用魔法算法构建负载均衡模型的方法。当解决功能最佳问题时, MAGA 具有更快的收敛速度。特别是在大数据集上, 魔法算法保持不错。比较 UT 策略生成的加载模型, 弹簧策略和魔法算法策略, 可以发现 MAGA 生成的加载平衡模型更能加载以平衡 UT 策略和弹簧策略生成的加载平衡模型, 特别是在任务数量很大的情况下, MAGA 生成的云计算加载平衡模型明显优于其他两个算法, 并且该模型不易引起系统的“单次故障”, 这表明该方法是可行且有效的构建云计算负载均衡策略。

### 参考文献:

- [1] 卢虎, 蒋小强, 闵欢. 具有通信约束的分布式 SOR 多智能体轨迹估计算法 [J]. 航空学报, 2019, 40(10): 166-178.
- [2] 徐向艺, 陈东方. 浅析信息受限下分布式多智能体优化算法设计与分析 [J]. 信息技术与信息化, 2019(12): 118-120.
- [3] 金凯琪. 多智能体系统的一致性研究及其在电力系统中的应用 [D]. 华北电力大学, 2019.
- [4] 盛喆. 基于多智能体的泵站多机组优化调度系统研究 [J]. 水资源开发与管理, 2019, 45(10): 67-73.
- [5] 田承东, 杨壁竹. 基于事件触发的聚类分析方法在多智能体任务协同中的研究 [J]. 新电脑, 2019, 51(02): 9-14, 26.