

# 云计算环境下虚拟机部署策略研究\*

袁爱平

(长沙民政学院软件学院 长沙 410004)

**摘要** 为了提高云计算环境下的资源利用率,提出了充分考虑资源负载均衡的虚拟机部署算法。该算法在分组遗传算法的基础上,综合分析虚拟机资源提供过程中资源容量的限制以及负载均衡的需求,从空间利用率、重量和负载变化三个方面来定义评价函数,并通过加权方法定义适应度函数。为了实现快速有效地获取近似最优解,对交叉和变异等进行了优化。在 CloudSim 环境下进行了仿真,实验结果表明该算法有助于云服务资源池中虚拟机资源的均衡利用。

**关键词** 云计算;虚拟机;负载均衡;分组遗传算法

**中图分类号** TP301 **DOI:**10. 3969/j. issn. 1672-9722. 2018. 08. 024

## Research on Virtual Machine Deployment Strategy in Cloud Computing Environment

YUAN Aiping

(Department of Software, Changsha Social Work College, Changsha 410004)

**Abstract** In order to improve the resource utilization of cloud computing environment, a virtual machine resource provision load balance algorithm is proposed. The algorithm which based on group genetic algorithm gives a synthetic analysis of virtual machine resource capacity constraints and demand in the process of load balancing. The space utilization rate, weight and load changes are used to define the evaluation function and the fitness function is also defined by weighting method. The crossover and mutation are optimized to obtain the approximate optimal solution. The results of simulation in CloudSim environment show that the algorithm will help the balanced utilization of virtual machine resources in the cloud resource pool.

**Key Words** cloud computing, virtual machine, load balance, group genetic algorithm

**Class Number** TP301

### 1 引言

云计算是从分布式计算、网格计算和并行计算等概念发展而来一种新型的计算模式,它是一种基于互联网相关服务的增加、使用和交付模式。虚拟机资源是云计算环境中的主要资源,数据中心的各种硬件资源通过使用虚拟化技术形成虚拟资源池,系统动态部署虚拟机,用户透明使用。随着用户数量的持续增加,数据中心规模不断扩大,如何提高云计算环境中的虚拟机资源利用效率以及用户响应的等待时间,已成为云计算资源部署的关键问题。

虚拟机部署策略的优劣决定了数据中心云计

算系统的性能。目前,针对云计算环境中的虚拟机资源部署问题,专家和学者们进行广泛研究并提出了各种策略。文献[1]通过计算待部署虚拟机和服务性能向量的相对距离,获得待部署虚拟机的匹配向量,将匹配向量与系统负载向量综合分析得到虚拟机的部署位置;文献[2]提出了多周期分配策略,建立了多周期分配的虚拟机负载均衡问题的数学模型,利用改进的蚁群算法进行了求解;文献[3]提出了一种面向多目标优化的虚拟机初始放置方法,基于改进的分组遗传算法和多目标模糊评估方法计算虚拟机的放置问题。本文在分组遗传算法的基础上,提出了一种负载均衡的虚拟机调度算法 LBGA,综合考虑虚拟机资源提供过程中资源容量

\* 收稿日期:2018年2月15日,修回日期:2018年3月19日

基金项目:湖南省教育厅科研项目(编号:14C0062);国家自然科学基金项目(编号:61772087)资助。

作者简介:袁爱平,男,硕士,副教授,研究方向:嵌入式系统、计算机网络、知识工程。

的限制以及负载均衡的需求,从空间利用率、重量和负载变化三个方面来定义评价函数,并通过加权方法定义适应度函数。另外,对选择和变异操作进行了优化,以获得算法的近似最优解。最后通过云仿真计算器 CloudSim<sup>[4]</sup>进行了仿真,仿真结果表明本文的算法均衡了服务器资源的分配,提高了资源的综合利用率。

2 调度算法模型

我们假设有  $m$  台可用的物理服务器,定义为  $M=\{PS_i|1\leq i\leq m\}$ ,另有  $n$  个虚拟机调度请求,定义为  $N=\{VM_i|1\leq i\leq n\}$ 。虚拟机需物理服务器上的多维资源来实行运算,定义虚拟机  $i$  对资源的请求向量为  $q_i=\{q_{i,1},q_{i,2},\cdots,q_{i,k}\}$ , $q_{i,j}$  表示虚拟机  $i$  对  $j$  类型资源的需求量。同样,定义物理服务器  $i$  能提供的资源容量向量为  $c_i=\{c_{i,1},c_{i,2},\cdots,c_{i,k}\}$ , $c_{i,j}$  表示服务器  $i$  能提供的  $j$  类型资源容量。服务器资源负载等于虚拟机所占各维资源向量的和。

定义 1:  $S=\{S_1,S_2,\cdots,S_n\}$  表示虚拟机的映射解集合, $S_i$  表示主机  $i$  上的一种虚拟机映射方案,令  $T$  为负载监测周期,根据主机负载的变化规律,将  $T$  划分为  $r$  个时间段为

$$T=[(t_1-t_0),(t_2-t_1),\cdots,(t_r-t_{r-1})]$$
 (1)

定义 2:假设虚拟机的负载在各个时间段中是稳定的,定义虚拟机  $VM_i$  在时间段  $k$  的负载为  $VL(i,k)$ ,则周期  $T$  中主机  $PS_i$  的总负载  $PL(i,T)$  为

$$PL(i,T)=\sum_{j=1}^n\sum_{k=1}^rVL(j,k)$$
 (2)

通常虚拟机配置到主机后,资源负载会发生很大的变化,因此,需通过虚拟机的动态迁移来调整资源负载以达到负载均衡,我们把虚拟机配置到物理服务器后在周期  $T$  内的负载标准偏差设置为

$$\sigma(T)=\sqrt{\frac{1}{m}\sum_{i=1}^m(PL(T)-PL(i,T))^2}$$
 (3)

其中

$$PL(T)=\frac{1}{m}\sum_{i=1}^mPL(i,T)$$
 (4)

$\sigma(T)$  反映了样本总体相对于平均值的离散程度,即表示了各物理主机负载相对于平均负载的离散程度。

由于云计算环境中虚拟机所需物理服务器的资源具有多维性(如不同的 CPU、内存和 I/O 等),各维资源的请求及性能差异很大,因此可以通过资源使用状况的负载度量来反映多维资源的利用情

况。本文将同时考虑主机的三种资源:CPU、内存和 I/O,把虚拟机负载定义为

$$VL(i,k)=1_1LI_{CPU}+1_2LI_{Mem}+1_3LI_{IO}$$
 (5)

其中  $0_1,1_2,1_3\leq 1$ ,且  $1_1+1_2+1_3=1$ ,表示资源的负载权值。 $LI_{CPU}$ 、 $LI_{Mem}$  和  $LI_{IO}$  分别表示为主机中 CPU、内存和 I/O 的利用率。我们算法的目标是启动最少的物理服务器以获得最大的资源利用率,同时确保被分配的资源不超过服务器的容量上限。

3 算法实现

如上所述,本文在设计算法时重点考虑多维资源之间的均衡负载,以实现最小化物理服务器启动数量和最大化资源综合利用率的目標。基于服务器聚合思想的虚拟机调度属于组合优化中的装箱问题,传统的遗传算法由于编码等问题对于分组问题的适应性很差<sup>[5]</sup>。因此,本文提出了一种改进的分组遗传算法来求解虚拟机的调度问题。该算法在分组遗传算法的基础上,采用负载均衡的资源分配方案,从空间利用率、重量和负载变化三个方面来定义适应度函数。另外算法采用轮盘赌法来选择个体,并优化了种群中个体之间的交叉及变异,以快速有效地获取近似最优解。其算法流程如图 1 所示。

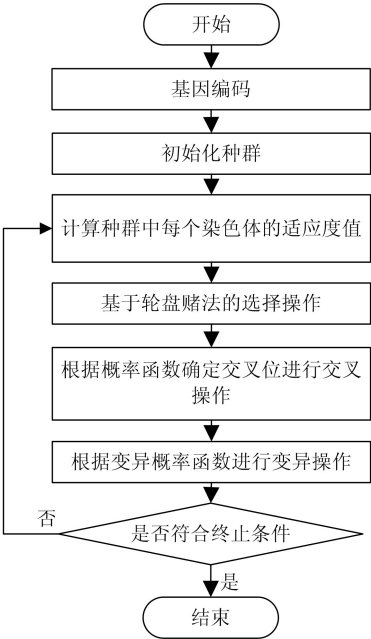


图 1 算法基本流程

3.1 基因编码

在给出的分组遗传算法中,为了避免编码的长度过长而影响计算的效率,以及后面的交叉和变异等操作不会破坏服务器的硬约束条件,我们用服务器的数量来决定染色体的基因位数。染色体中的

每一位基因代表一个物理服务器,基因的值表示该服务器上装载的虚拟机集合,也就是虚拟机的一个分组,如图 2 所示,A、B 代表物理服务器,1~5 代表虚拟机,染色体的编码为:AB(A: {1、3}, B: {2、4、5})。

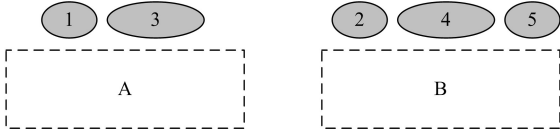


图 2 算法中的染色体基因编码示例

3.2 适应度函数

适应度函数的选取直接影响到分组遗传算法的收敛速度以及能否找到最优解,为了达到最小化物理服务器启动数量和最大化资源综合利用率的的目标,适应度函数的设计应综合考虑物理服务器的使用数量和多维资源的均衡分配。考虑到虚拟机资源提供过程中资源容量的限制以及负载均衡的需求,从空间利用率、重量和负载变化三个方面来定义评价函数,并通过加权方法定义最终的算法适应度函数。

1)空间利用率函数

空间利用率为虚拟机体积占主机体积的比率,定义为

$$f(x)=\frac{\sum_{i=1}^nBV_i}{CV} \tag{6}$$

其中,  $BV_i$  表示放入主机的第  $i$  个虚拟机的体积,  $CV$  表示主机的体积,  $n$  为主机中的虚拟机数量。第  $i$  个虚拟机的体积定义为

$$BV_i=LI_{CPU}^iLI_{Mem}^iLI_{IO}^i \tag{7}$$

2)重量函数

虚拟机的重量以其负载  $VL$  表示,定义为

$$f(y)=\begin{cases} \frac{\sum_{i=1}^nVL_i}{PL}, & \sum_{i=1}^nVL_i \leq PL \\ 0, & \sum_{i=1}^nVL_i > PL \end{cases} \tag{8}$$

其中,  $VL_i$  表示第  $i$  个虚拟机的重量,  $PL$  表示主机的最大承载重量,若虚拟机的总重量超过主机的最大载重,则函数的值为 0。

3)负载变化函数

$$f(z)=\Phi(\sigma(S,T)-\sigma_0)=\begin{cases} 1, & \sigma(S,T)-\sigma_0 \leq 0 \\ e, & \sigma(S,T)-\sigma_0 > 0 \end{cases} \tag{9}$$

$\sigma_0$  表示实现主机负载均衡所允许的负载变化约束值,  $\sigma(S,T)$  表示主机在  $T$  时间段内实际的负载变化值,  $\Phi(x)$  为惩罚函数,当个体满足负载变化

约束时,其值为 1,否则为  $e$ 。

根据惩罚函数和处理多目标优化的加权系数方法,本文定义适应度函数为

$$F(S,T)=Af(x)+Bf(y)+C \times 1/f(z) \tag{10}$$

其中,  $A$ 、 $B$  和  $C$  表示加权因子,  $0 \leq A, B, C \leq 1$ , 且  $A+B+C=1$ 。

3.3 选择

选择操作就是对每次迭代产生的个体进行整体评估,然后选择适应度值较大的个体作为下一代种群的初始集合。为了避免仅挑选适应度较大的个体而使解陷入局部最优,我们采用轮盘赌法来选择种群的个体,其基本原理是根据每个染色体适应度的比例来确定该个体的选择概率。具体步骤如下:

步骤一:根据式(10)定义的适应度函数计算每个个体的适应度值和种群的适应度总和。

步骤二:计算个体的适应度值在总和中所占的比例,作为该个体的选择概率  $p_s(a_i)$ 。

步骤三:产生一个 0~1 之间的随机数  $e$ , 若  $p_s^*(a_{k-1}) < e < p_s^*(a_k)$ , 则选择个体  $k$ 。其中,  $p_s^*(a_k) = \sum_{i=1}^k p_s(a_i)$ , 表示第  $k$  个个体的累计选择概率。

可以看出,这种选择算法随机性较强,适应度值越大的个体被选择的概率越大,同时具有较小适应度值的个体也有被选择的机会。

3.4 交叉

交叉操作通过选择两个父代个体的部分结构进行交换以生成新的个体,可以极大提升遗传算法的全局搜索能力。但如随机选择交叉位,则会造成全局搜索效率的下降;如每次选择评估值较高的基因进行交叉,则算法很容易陷入局部最优并出现早熟收敛现象。因此我们引入可调参数的概率函数,既保证了评估值较高的基因被选择以进行交叉,同时制造了全局搜索的随机性。我们首先计算染色体中的基因评估值并排序,再定义排在第 1 位的基因被选择为交叉位的概率为  $p(q)$ ,服从  $p(q) \sim q^{-\varepsilon}$  的分布函数,其中,  $\varepsilon \geq 0$ , 且取值为 2。可以看出,概率的随机性保证了全局搜索的性能,同时评估值较高的基因具有较大的概率被选择为交叉位。选择基因进行交叉操作之后,去掉含有相同基因值的服务器编码,同时采用 FFD<sup>[6]</sup>方式回填因消除服务器而失去从属关系的虚拟机。

3.5 变异

变异算子通过赋予个体以很小的概率产生转



变而形成新的个体,增加了种群的多样性,使得遗传算法在交叉算子决定的全局搜索能力的基础上,还具有局部的随机搜索能力,以防止出现非成熟收敛。为了保证算法的局部搜索能力,变异率不能过大,如果变异率过大,则算法就退化成随机搜索,因此设置变异概率函数如下:

$$p_m = \begin{cases} \frac{k_1(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_2, & f < f_{\text{avg}} \end{cases} \quad (11)$$

式中,  $f$  为要变异的个体适应度值,  $f_{\text{avg}}$  为当代种群的平均适应度值,  $f_{\max}$  为种群中最大的适应度值。当  $f \geq f_{\text{avg}}$ , 表示个体适应度值较大时则应该让  $p_m$  较小, 防止适应度值较大的个体被破坏; 当  $f < f_{\text{avg}}$ , 表示个体适应度值较小时则应该让  $p_m$  较大, 以开辟新的搜索区域。在本文中设定  $k_1 = 0.003$ ,  $k_2 = 0.0042$ 。通过引入变异概率函数, 根据种群适应度值的情况调整变异概率, 不仅加快了算法的收敛速度, 同时防止出现非成熟收敛而陷入早熟情况。

3.6 终止条件

算法的终止是防止出现迭代次数过多的情况, 本文基于最优跨度的适应度函数值标准差来决定算法的终止, 如式(12)所示:

$$\sqrt{\frac{\sum_{i=1}^N (f(i) - f_{\text{avg}})^2}{N}} < \zeta, \quad \zeta \in (0, 1) \quad (12)$$

其中,  $f(i)$  表示第  $i$  个个体的适应度值,  $f_{\text{avg}}$  为当代种群的平均适应度值,  $N$  为种群的个体数量,  $\zeta$  为收敛的阈值, 在这里取  $\zeta = 0.1$ , 当算法的标准差小于这个阈值时终止迭代, 否则继续进行迭代进化。

4 仿真实验

为了验证负载均衡调度算法 LBGA 与传统算法相比在效率上的提高, 本文采用 CloudSim 工具进行仿真, 通过与该领域的服务器聚合算法 FFD 以及多维资源调度算法 DRF<sup>[7]</sup> 进行比较。实验运行的工作负载规模分为 100、300、500、800 及 1000 个虚拟机这五个实例, 参数设置参考 Feitelson<sup>[8]</sup> 和 Mishra<sup>[9]</sup> 等方法, 我们把每个实例中的算法均运行 30 次, 取其平均值作为性能分析的依据。

物理服务器的使用数量是衡量服务器资源均衡调度性能的重要指标<sup>[10]</sup>。LBGA 算法、DRF 算法和 FFD 算法在物理服务器使用数量方面的比较如图 3 所示。通过分析可以看到, 由于 LBGA 算法综

合考虑了资源负载的均衡, 因此对于实验运行的各个工作实例都只需要启动最少的物理服务器, LBGA 算法相比 FFD 算法、DRF 算法分别节省 11.2% 和 4% 的服务器使用量。LBGA 算法由于在调度中综合考虑了虚拟机的多维资源需求及服务器的资源容量, 通过虚拟机的合理部署使服务器的空余资源达到均衡, 为后续虚拟机的资源使用提供了更多的可能性, 相应地减少了服务器的使用数量。

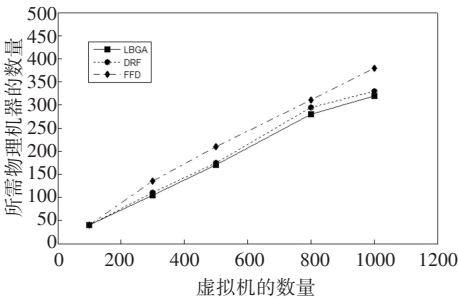


图3 服务器使用数量比较

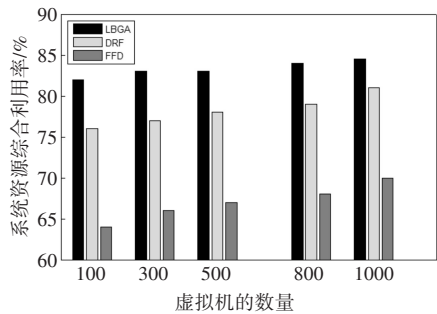


图4 系统资源综合利用率比较

系统资源的综合利用率也是衡量调度算法性能优劣的重要依据<sup>[11]</sup>。通过图 4 中 LBGA 算法、DRF 算法和 FFD 算法的比较可以看到, LBGA 算法相比 DRF 算法和 FFD 算法在资源综合利用率方面取得了 6.5%~12.1% 的提高, 并且由于 LBGA 算法具有较小的标准差, 算法在稳定性方面也具有一定优势。由于 LBGA 算法在虚拟机部署过程中注重物理服务器资源之间的负载均衡, 每次虚拟机部署时尽量满足服务器多维资源之间的互补关系, 防止出现因某种资源过早饱和而抑制其它资源使用的现象, 因此从整体上提高了服务器资源的综合利用率。

5 结语

本文针对传统的遗传算法在解决装箱问题上效率的不足进行了改正, 并将其应用于云环境中虚拟机的部署问题。该算法在分组遗传算法的基础上, 从空间利用率、重量和负载变化三个方面来定义适应度函数, 并通过加权方法定义适应度函数。

同时,算法对选择和变异操作进行了优化,实现了数据中心服务器资源的高效均衡分配。最后,通过与该领域的聚合调度算法 FFD 和多维资源调度算法 DRF 比较,证明了该调度算法的有效性和高效性。

参 考 文 献

[1] 杨星,马自堂,孙磊. 云环境下基于性能向量的虚拟机部署算法[J]. 计算机应用,2012,32(1):16-19.  
YANG Xing, MA Zitang, SUN Lei. Performance vector-based algorithm for virtual machine deployment in infrastructure clouds[J]. Journal of Computer Applications, 2012,32(1):16-19.

[2] 马华伟,吴晓莉. 云计算环境下虚拟机负载均衡问题研究[J]. 微电子学与计算机,2014,31(4):10-12.  
Ma Huawei, WU Xiaoli. Research of Virtual Machine Load Balancing in the Cloud Computing Environment[J]. Microelectronics & Computer,2014,31(4):10-12.

[3] 李淑英,潘亚,费薇,等. 基于分组遗传算法的虚拟机放置方法[J]. 南京理工大学学报,2016,40(3):322-327.  
LI Shuying, PAN Ya, FEI Wei, et al. Virtual machine placement method based on grouping genetic algorithm [J]. Journal of Nanjing University of Science and Technology, 2016, 40(3):322-327.

[4] Calheiros R N, Ranjan R, de Rose C A F, et al. CloudSim: a novel framework for modeling and simulation of cloud computing infrastructures and services [EB/OL]. [2009-09-03]. <http://www.cloudbus.org/reports/Cloud-Sim-ICPP2009.pdf>.

[5] Falkenauer E, Delchambre A. A genetic algorithm for bin packing and line balancing [C]//Nice, France: Proceedings of IEEE International Conference on Robotics and Automation, 1992.

[6] S.Martello and P. Toth. "Knapsack problems: algorithms and computer implementations," Wiley, 1990.

[7] A.Ghods, M.Zaharia, A.Konwinski, et al. Dominant resource fairness: Fair allocation of heterogeneous resources in datacenters[C]. In NSDI 2011: Proceedings of the 8th USENIX conference on Networked systems design and implementation, Boston, MA, 2011:2-3.

[8] D.G.Feitelson. Workload modeling for computer systems performance evaluation [EB/OL]. <http://www.cs.huji.ac.il/~feit/wlmod/>.

[9] A.Mishra, J.L.Hellerstein, and WalfredoCirne. Towards characterizing cloud backend workloads: Insights from google compute clusters[J] ACM Sigmetries Performance Evaluation Review, 2009,37(4):34-41.

[10] Zhong M, Shen K, Seiferas J. Replication Degree customization for high availability[J]. Operating System Review, 42(4), 2008:55-68.

[11] Kusic D, Kephart J O, Hanson J E, et al. Power and performance management of virtualized computing environments via lookahead control [J]. Cluster Computing, 2009,12(1):1-15.

[12] Beloglazov A, Buyya R. Energy Efficient Resource Management in Virtualized Cloud Data Centers[C]. Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, Australia, 2010: 826- 831.

[13] Beloglazov A, Buyya R. Energy Efficient Allocation of Virtual Machines in Cloud Data Centers[C]. Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010:577-578.

[14] Beloglazov A, Abawajy J, Buyya R. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing[J]. Future Generation Computer Systems, 2012, 28(5):755-768.

[15] Beloglazov A, Abawajy J, Buyya R. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing[J]. Future Generation Computer Systems,2012,28(5):755-768.