

基于粒子群算法的云计算资源配置研究^①

蔡林益

广州番禺职业技术学院 财经学院, 广州 511483

摘要: 对于云计算而言, 虚拟机资源的合理高效配置具有重要意义. 该文对粒子群方法进行到云计算资源配置的映射, 详细地设计了 3 个约束条件和目标函数. 目标函数中包含了资源利用率和迁移次数 2 个优化目标, 整个虚拟机资源的配置过程设置了 8 个步骤. 实验结果表明: 同 2 种参照方法相比, 该文所提出的基于粒子群算法的云资源配置方法完成配置后, 不仅资源利用率高、迁移次数低, 其迭代过程和迭代时间也令人满意.

关键词: 云计算; 粒子群; 资源调度; 约束条件

中图分类号: TP391

文献标志码: A

文章编号: 1000-5471(2017)09-0128-05

云计算的一个核心任务就是虚拟机的资源调度与配置, 只有这个调度和配置合理, 云服务才能获得更大的收益、云用户才能获得更高的服务效率^[1]. 当然, 虚拟机的配置过程依赖于一些重要的参数, 也依赖于配置过程的优化程度, 所以如何选择配置参数、设置优化过程, 关乎到云计算资源配置的效率^[2]. 张亮等^[3]认为云计算的资源配置要充分考虑云端的拓扑结构, 也要考虑虚拟机、物理主机的通信关系、拓扑结构等, 这关乎到能否得到更为优化的配置. Wilcox 等^[4]将遗传算法运用于虚拟机的调度, 充分考虑了云端的历史和当前情况, 进而实现以负载均衡为标志的虚拟机最小配置代价. 吉斌等^[5]指出, 对于用户、服务商等不同云端角色, 其对于云资源配置的目标也存在很大的差异, 所以合理的云资源配置就是云端不同角色博弈的结果. 王洪峰等^[6]以应用程序的最小开销为对象, 探寻云计算中合理的资源配置, 并构建了一个分层模型来实现各种资源的优化配置. 邹华^[7]认为, 云计算的资源配置应该建立在对云资源需求准确预测的基础之上, 进而充分考虑虚拟机的动态属性, 才能达成预期的配置效果. 吕晨怀^[8]认为, 云计算的优化过程本身是一个组合优化问题, 包括资源消化、虚拟机迁移成本等等. 吴丽等^[9]指出, 云资源配置为了达到利益的最大化, 应该设置多个优化指标, 进而映射成多目标优化问题. Paul 等^[10]针对多目标云资源配置, 构建了一种分阶段配置机制, 并通过两级管理机制实现虚拟机最终到物理主机的配置. 本文在多目标优化思想的指引下, 构建一种基于粒子群的云资源配置方法, 以期获得更好的虚拟机资源配置效果.

1 基于粒子群算法的云计算资源配置方案

1.1 粒子群算法

粒子群算法是一种典型的多目标优化方法, 其本质是模拟鸟群的飞行过程和觅食行为. 实际上, 鸟群的运动过程是通过个体之间的有机协调实现鸟群整体的最佳. 相比于真实的鸟群, 粒子群中的粒子个体是不考虑体积和质量的, 每个粒子都为整个种群达到最佳提供了潜在的可能. 各个粒子之间不仅可以分享彼此的信息, 还可以进行分工协作, 进而找到种群的最优位置. 当然, 每个粒子都有自己的最优位置, 这种个体的最优对于整个种群来讲是局部的; 整个种群处于最佳状态时, 各个粒子可能并不是自己的最优位置,

① 收稿日期: 2017-01-31

基金项目: 广州番禺职业技术学院“十三五”科技项目(2016KJ007); 广州番禺职业技术学院“十二五”第二批科技项目(2015KJ003).

作者简介: 蔡林益(1977-), 男, 广东广州人, 实验师, 主要从事计算机研究.

但这时的最优对于整个种群来讲是全局的。

假设粒子群配置了 N 个优化目标, 每个粒子都是包含了这 N 个维度的量, 可以记作 $x_i = (x_{i1}, x_{i2}, \dots, x_{iN})$; 每个粒子在更新过程中的速度参量, 可以记作 $v_i = (v_{i1}, v_{i2}, \dots, v_{iN})$; 每个粒子在更新过程中的局部最优位置参量, 可以记作 $p_l = (p_{l1}, p_{l2}, \dots, p_{lN})$; 每个粒子在更新过程中的全局最优位置参量, 可以记作 $p_g = (p_{g1}, p_{g2}, \dots, p_{gN})$ 。

假设一个粒子群中含有 m 个粒子, 对其迭代更新的处理为

$$\begin{aligned} v_m^{k+1} &= v_m^k + \eta_1 \omega_1 (p_{lm} - x_m^k) + \eta_2 \omega_2 (p_{gm} - x_m^k) \\ x_m^{k+1} &= x_m^k + v_m^{k+1} \end{aligned} \quad (1)$$

这里, $i = 1, 2, \dots, m, n = 1, 2, \dots, N, k$ 表达了迭代次数, η_1 和 η_2 表达的是 $[0, 1]$ 区间上的随机数, 通过随机设置确保了粒子群的多样性。 ω_1, ω_2 表达的是 2 个学习因子, ω_1 表达了自学习更新能力, ω_2 表达了他学习更新能力。

1.2 云计算资源配置方案

云端涵盖了多个数据中心, 数据中心又包含了多个物理主机, 物理主机对应着多个虚拟机。如果每一个物理主机的属性特征都是确定的, 虚拟机就可以考虑在不同的物理主机上进行有选择的执行。

云资源的配置算法主要就是针对虚拟机的配置而言的, 这种配置过程一方面要考虑用户的任务情况, 另一方面要考虑数据中心的资源状况。此外, 还要充分考虑已经配置好的虚拟机资源是否需要再进行重新配置。

下面根据云资源的配置需求, 进行数学描述的抽象。如果存在 N 台物理主机, 这些主机都可以提供虚拟机配置。同时, 存在 m 个虚拟机, 存在一个参数 m_r 用于表达新闲置的虚拟机数量, 存在一个参数 m_s 表达原有闲置的虚拟机数量, 三者之间符合关系 $m = m_r + m_s$ 。

物理主机的资源配置包括 3 个方面, 其中 R_i^C 表达第 i 个物理主机的 CPU 资源, R_i^M 表达第 i 个物理主机的内存资源, R_i^B 表达第 i 个物理主机的带宽资源, 3 个参数构成 1 个组合参数 $R_i = \{R_i^C, R_i^M, R_i^B\}$ 。同样地, 虚拟机也可以用 1 个组合参数表达, 即 $V_j = \{V_j^C, V_j^M, V_j^B\}$ 。

这样, 可以构建 1 个虚拟机是否都合理地配置到物理主机上的矩阵, 如公式(2)所示。矩阵中的每一个参数 η_{ij} 用于表示虚拟机的配置状态, 如果已经配置完毕就是 1, 否则为 0。

$$\eta = \begin{bmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1m} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{N1} & \eta_{N2} & \cdots & \eta_{Nm} \end{bmatrix} \quad (2)$$

各虚拟机资源都完成配置后, 配置是否合理需要进一步优化, 优化的目标函数如公式(3)所示, 同时配套的约束条件如公式(4)所示。

$$\max \varphi_1(R_i, V_j) = \frac{\sum_{j=1}^m V_j^C}{N} + \frac{\sum_{j=1}^m V_j^M}{N} + \frac{\sum_{j=1}^m V_j^B}{N} \quad (3)$$

$$\max \varphi_2(m_i) = \sum_{j=1}^m m_j$$

这里, 包含了 2 个优化目标, 一个是各个节点的最大资源利用率, 另一个是虚拟机最小的反复配置次数, 也就是最小迁移次数。

$$\begin{aligned} \sum_{j=1}^N \eta_{jk} &= 1, k = [1, 2, \dots, m] \\ \sum_{j=1}^m \eta_{ij} V_j^C &< R_i^C \end{aligned}$$

$$\sum_{j=1}^m \eta_{ij} V_j^M < R_j^M$$
$$\sum_{j=1}^m \eta_{ij} V_j^B < R_j^B$$

(4)

这里, $\sum_{j=1}^N \eta_{jk} = 1$ 用于约束虚拟机只能在一个服务器上进行配置. 后面的 3 个约束条件, 则分别标明了配置后虚拟机使用的 CPU 资源、内存资源、带宽资源不能超过物理主机的 CPU 资源、内存资源、带宽资源.

1.3 云计算资源配置步骤

经过上述的方案设计, 云计算资源配置已经映射成为一个多目标优化问题. 对于这种问题的求解, 一般可以采用穷举法. 但是, 如果云端的物理主机和虚拟机数量过于庞大, 穷举法的实现就比较困难. 粒子群算法属于启发式方法范畴, 但传统粒子群算法的优化过程一般都是针对单一优化目标的, 本文针对已经建立的多目标云资源配置过程, 设置完整的步骤.

- 根据云计算资源的粒子群配置方案, 本文的配置流程如下:
- 第 1 个环节, 为粒子群设置初始规模, 并用参数 X 表示; 在种群内随机生成粒子位置和粒子速度, 这 2 个值都分别用参数 $x_i = (x_{i1}, x_{i2}, \cdots x_{iN})$ 和 $v_i = (v_{i1}, v_{i2}, \cdots v_{iN})$ 表示, 同时要设置判断阈值和迭代次数的上限.
- 第 2 个环节, 计算目标函数一 $\varphi_1(z_i)$ 和目标函数二 $\varphi_2(z_i)$ 的最佳 z_i 值.
- 第 3 个环节, 根据已经在上个环节计算出来的 z_i 值, 分别求取局部最佳值和全局最佳值.
- 第 4 个环节, 从计算出来的所有局部最佳值中, 选取出 2 个最好的.
- 第 5 个环节, 将上个环节的 2 个最佳值加和取平均, 并和第三步的全局最佳值进行相似性测算, 测算的标准是欧式距离.
- 第 6 个环节, 按照上述 2 个环节, 实现对粒子群中每个粒子局部最佳的距离测算.
- 第 7 个环节, 根据迭代所得的全局最佳和局部最佳, 不断更新种群中每个粒子的速度信息和位置信息.
- 第 8 个环节, 当迭代过程满足收敛条件时, 输出迭代结果.

2 云计算资源配置实验结果与分析

为了验证本文提出的基于粒子群算法的云计算虚拟机资源配置方法的有效性, 本文进行相关的验证性实验对设计出的算法性能进行验证和分析.

整个实验的参数设计: 物理主机的数量为 100 个, 参与虚拟机配置的虚拟机数量为 200 个, 全部物理主机的资源配置情况一致, 都包含了 10 核 CPU、20 GB 大小的内存、100 M 大小的带宽. 这 200 个虚拟机资源则有所不同, 一共可以分为 4 个类型, 每个类型的虚拟机数量为 25 个. 任何一个虚拟机的任务请求, 都要在算法的优化过程执行之后到达.

对于本文提出的算法的设计: 粒子群的种群规模为 50 个, 即包含的粒子数量为 50 个; 迭代过程的上限为 40 次; 学习因子 w_1 设置为 0.5, 学习因子 w_2 设置为 0.5. 同时, 如果物理主机的资源利用率降低了 10%, 那么就看作虚拟机配置发生了一次迁移.

选择了 2 个单目标优化方法, 作为本文粒子群虚拟机配置方法的对比方法. 第 1 个是以资源利用率为优化目标的虚拟机调度方法, 命名为单目标方法一; 第 2 个是以迁移次数为优化目标的虚拟机调度方法, 命名为单目标方法二. 对于本文的方法, 将资源利用率和迁移次数的 2 个优化目标都设置相等的权重, 即 0.5.

上述 3 种方法完成虚拟机配置后, 资源利用率如图 1 所示.

从图 1 的结果可以看出, 3 种方法针对相同的任务数量完成虚拟机资源配置后, 本文方法的资源利用率最高, 既高于以资源利用率为优化目标的虚拟机调度方法, 也高于以迁移次数为优化目标的虚拟机调度方法.

进一步比较 3 种方法在迁移次数方面的情况, 其实验结果如图 2.

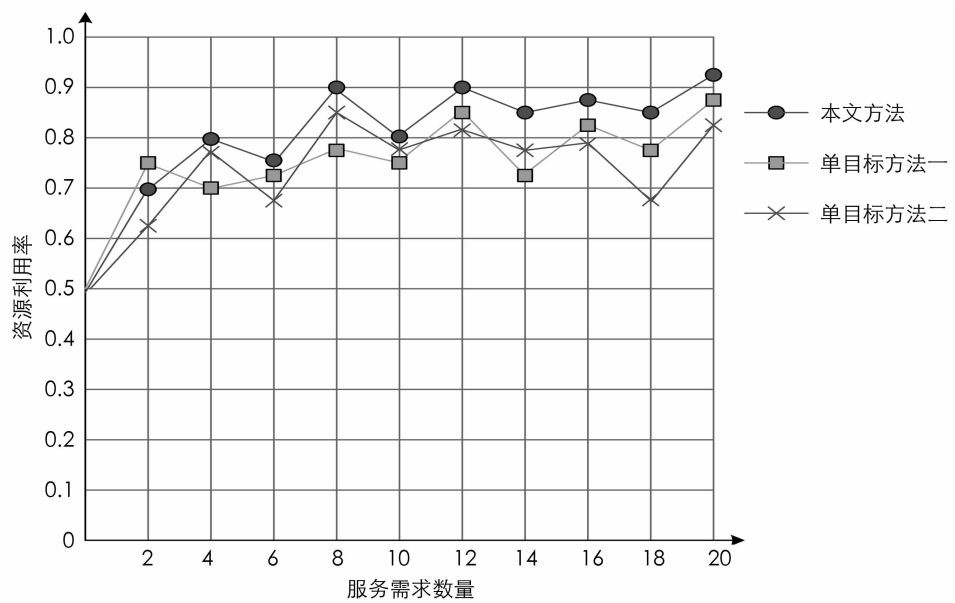


图 1 3 种方法的资源利用率对比

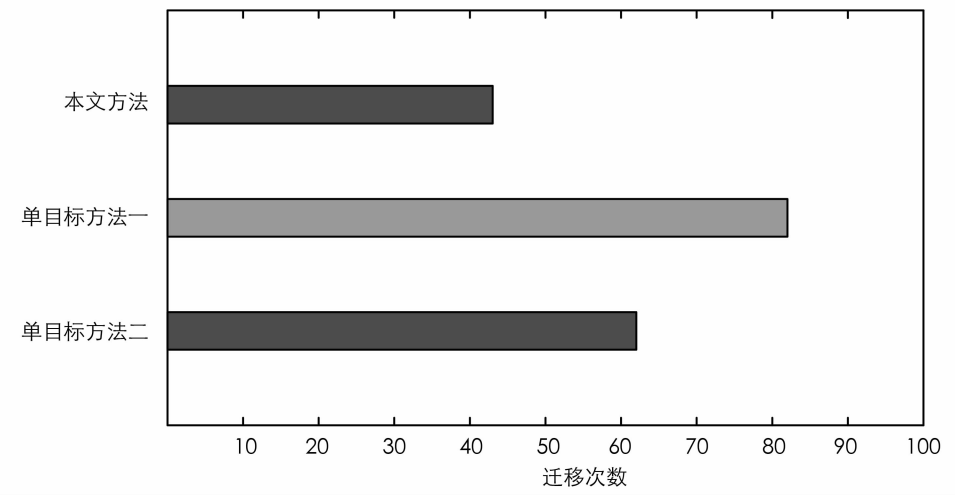


图 2 3 种方法的迁移次数对比

从图 2 的结果可以看出，本文方法在迁移次数方面也优于其他 2 种方法。对于同样的实验条件，本文方法的迁移次数仅为 43 次，而单目标方法一和单目标方法二的迁移次数分别为 82 次、63 次。

再次考察本文方法的迭代过程，即迭代次数的多少和迭代时间的长短，结果如表 1 所示。

表 1 配置过程的迭代结果

虚拟机数目		迭代次数和迭代时间的结果					
虚拟机数目	迭代次数	10	15	20	25	30	35
	时间/s	1.287	3.109	8.072	12.488	17.936	20.688
20	迭代次数	10	15	20	25	30	35
	时间/s	5.322	6.924	11.283	14.351	20.921	25.121
30	迭代次数	10	15	20	25	30	35
	时间/s	12.188	13.872	16.931	18.964	24.765	28.012
40	迭代次数	10	15	20	25	30	35
	时间/s	15.326	16.863	19.765	21.423	25.283	30.001
50	迭代次数	10	15	20	25	30	35
	时间/s	17.279	18.875	21.868	24.182	27.951	30.882

从表 1 的结果可以看出,随着粒子群规模不断扩大、粒子数目不断增加,本文所构建的虚拟机配置方法迭代次数也在增加,迭代时间也在变长,但增加和延长都在可以接受的范围内。

3 结 论

对于云计算问题而言,各种资源调度尤其是虚拟机资源的配置具有非常重要的意义。本文在粒子群算法的基础上,同时设置了资源利用率和迁移次数 2 个优化目标,并分别设置了虚拟机到物理主机的 CPU 资源约束条件、内存资源约束条件、带宽资源约束条件,进而给出了配置过程的 8 个实施步骤。同 2 种参照方法的对比实验表明,本文方法完成虚拟机资源配置后,资源利用率更高,迁移次数更低,迭代过程和迭代时间比较理想。

参考文献:

[1] HUNG N O, NIEN P D, NAM N H, et al. A Genetic Algorithm for Power-Aware Virtual Machine Allocation in Private Cloud [M]. Berlin: Springer Berlin Heidelberg, 2013.

[2] 谢文静,唐 卓,杨 柳,等. 基于随机规划的云计算中虚拟机分配优化研究 [J]. 计算机工程与科学, 2012, 34(5): 95—100.

[3] 张 亮,张曦煌. 一种面向云计算虚拟机资源拓扑结构的任务调度 [J]. 计算机应用研究, 2015, 32(12): 3738—3741.

[4] WILCOX D, MCNABB A, SEPPI K. Solving Virtual Machine Packing with a Reordering Grouping Genetic Algorithm [J]. Evolutionary Computation, 2011, 30(1): 362—369.

[5] 吉 斌,杨 杨,钟泽伟. 一种基于进化博弈论的云计算虚拟计算资源配置模型 [J]. 北京交通大学学报(自然科学版), 2013, 37(5): 75—79.

[6] 王洪峰,任国恒. 混合云计算环境下利用 AMPL 建模的应用程序计算开销最小化 [J]. 西南师范大学学报(自然科学版), 2015, 40(7): 41—49.

[7] 邹 华. 基于云计算的资源预测与配置的优化 [J]. 电子技术与软件工程, 2015, 21(9): 24—24.

[8] 吕晨怀. 云计算环境下资源需求分析与配置策略研究 [J]. 无线互联科技, 2016(18): 120—121.

[9] 吴 丽,余文春. 基于多服务器最优配置的云计算利润最大化技术研究 [J]. 计算机应用研究, 2015, 32(1): 194—197.

[10] PAUL I, YALAMANCHILI S, JOHN L K. Performance Impact of Virtual Machine Placement in a Datacenter [J]. Performance Computing & Communications Conference, 2013, 326(2): 424—431.

On Cloud Computing Resource Allocation Based on Particle Swarm Optimization Algorithm

CAI Lin-yi

Finance and Economic College, Guangzhou Panyu Polytechnic Institute, Guangzhou 511483, China

Abstract: For cloud computing, the rational and efficient allocation of virtual machine resources is of great significance. In this paper, the particle swarm optimization (PSO) has been used to map the resource allocation of cloud computing, and three constraints and objective functions been designed in detail. The objective function includes two optimization objectives; the resource utilization rate and the number of migration, and the whole virtual machine resource allocation process is divided into 8 steps. The experimental results show that, compared with the two reference methods, the proposed allocation of cloud resources allocation method based on particle swarm optimization algorithm is completed, not only the resource utilization rate, migration times low, the iterative process and the iteration time is satisfactory.

Key words: cloud computing; particle swarm; virtual machine scheduling; DP algorithm

责任编辑 夏 娟