

[文章编号] 1003-4684(2021)05-0055-04

# 基于改进人工蜂群算法的云任务调度

吴 傲, 陈宏伟

(湖北工业大学计算机学院, 湖北 武汉 430068)

**[摘 要]** 基于传统的优化算法在云计算资源调度中存在的较低资源利用率, 任务调度时间成本较高、服务器节点间的负载不均衡等问题, 提出了基于博弈论的改进人工蜂群算法, 以此来提高传统的资源调度算法的工作效率。仿真实验结果显示, 基于博弈论的改进人工蜂群算法较原算法和同类型算法能够显著提高资源调度的效率和有效性。

**[关键词]** 云计算资源调度; 博弈论; 人工蜂群算法

**[中图分类号]** TP399 **[文献标识码]** A

云计算技术作为一种全新的计算模式, 对于用户来讲具有成本低廉, 使用即时, 操作简单, 计算高效等特点, 因而受到业界的大量关注, 目前已经成为计算机领域热门的研究方向。随着越来越大的用户需求, 云环境下的资源分配问题成为一个急需考虑的问题, 用户日益增长的资源需求和云服务资源的高效分配之间的矛盾是目前云计算资源调度下最关键的问题。在云计算资源调度系统中, 用户可以采取租用的形式按需获取供应商提供的计算力、信息服务以及存储空间<sup>[1]</sup>, 由于在实际使用中, 云计算资源调度系统面对的用户群体十分巨大, 而且任务类型极其复杂, 在资源的调度过程中需要一个合理高效且低成本的分配方案。对于云计算任务调度。文献[2]提出了一种基于改进粒子群的云计算任务调度策略, 从负载均衡的角度和任务完成的总时间的角度这两个方面来对资源调度任务进行优化。

人工蜂群算法(artificial bee colony algorithm, ABC)于 2005 年由土耳其教授 Karaboga 基于蜜蜂种群采蜜机制而提出的一种蜂群算法<sup>[3-4]</sup>。由于 BCO 算法参数多, 而 ABC 算法参数较少, 易于实现, 因此有关 ABC 算法的研究居多。

博弈论是数学基础理论的一个分支方向, 其包含博弈者本身、博弈者策略集合、信息、收益和纳什均衡等五大因素。目前的研究已有将博弈论用于云环境下的资源调度问题中。Osorio 等<sup>[5]</sup>简要描述了当前云资源分配策略, 并且首次提出有关云资源

分配框架。Hassan 等<sup>[6]</sup>提出了用博弈论的方法来解决云环境下有关资源分配的管理问题, 他们得出了合作博弈比非合作博弈的博弈模型更适合云环境的结论。

## 1 相关技术介绍

### 1.1 人工蜂群算法模型

在 ABC 算法中, 蜂群包含 3 个小种群, 分别为雇佣蜂(也称为引领蜂), 跟随蜂和侦察蜂<sup>[7]</sup>。算法原理是将蜜蜂种群寻找食物源的过程数学化为寻优求解的过程。原 ABC 算法步骤如下:

1): 由式子(1)初始化种群, 含有  $N$  个具有  $D$  维变量的解

$$X_{i,j} = X_{\min,j} + \text{Rand}(0,1) \cdot (X_{\max,j} - X_{\min,j}) \quad (1)$$

式(1)中  $i=1,2,\dots,N, j=1,2,\dots,D$ 。  $X_{\max,j}$  为搜索解空间的上限,  $X_{\min,j}$  为搜索解空间的下限。  $\text{Rand}(0,1)$  为区间  $[0,1]$  内的随机数。每个解  $X_i (i=1,2,\dots,N)$  用一个  $D$  维向量来表示,  $D$  是待优化目标问题参数个数。

2) 引领蜂根据式(2)对食物源进行邻域搜索, 并产生候选解, 计算候选解的适应度值。基于贪婪选择原理挑选较好的食物源(解), 保留适应度值最高的食物源。

$$V_{i,j} = X_{i,j} + r_{i,j} (X_{i,j} - X_{k,j}) \quad (2)$$

其中,  $j=1,2,\dots,D, k=1,2,\dots,N, X_{k,j}, X_{i,j}$  为解空间内的随机解, 且  $k \neq j, r_{i,j}$  为  $[-1,1]$  区间内的一个随机数, 若  $V_{i,j}$  函数值优于  $X_{i,j}$ , 则前者取代

[收稿日期] 2021-05-10

[基金项目] 湖北省重点研发计划(2020BAB012)

[第一作者] 吴 傲(1993-), 男, 湖北荆州人, 湖北工业大学硕士研究生, 研究方向为云计算

[通信作者] 陈宏伟(1975-), 男, 湖北武汉人, 工学博士, 湖北工业大学教授, 研究方向为云计算, 大数据

后者。

3) 跟随蜂根据公式(3)计算食物源的适应度值, 根据公式(4)计算得出食物源位置被选择的概率值。基于上述最优食物源, 根据式(2)展开二次邻域搜索。再计算得出新解的适应度值, 根据贪婪选择策略保留最优解。

$$f_{it_i} = \begin{cases} \frac{1}{1+f_i} & f_i > 0 \\ 1+abs(f_i) & f_i \leq 0 \end{cases} \quad (3)$$

$$P_i = f_{it_i} / \sum_{i=1}^N f_{it_i} \quad (4)$$

式(4)中:  $P_i$  表示为第  $i$  个食物源由跟随蜂选取的概率  $f_{it_i}$  为其对应的适应度值。  $f_i$  为该解的目标函数值, 等同于待优化问题的函数值。

4) 若某个解  $X_i$  在有限次循环(定义为 Limit\_time)之后并没有得到提高, 引领蜂必须放弃该较差的食物源, 且引领蜂更改职责称为侦察蜂, 按照式(5)产生随机新解。若多次循环后食物源没有被放弃, 则该食物源为最佳蜜源(即该解为问题最优解)。

$$X_{i,j} = X_{\min,j} + \text{Rand}(0,1) \cdot (X_{\max,j} - X_{\min,j}) \quad (5)$$

其中,  $X_{\min,j}$  为目前得到的第  $j$  维最小值;  $X_{\max,j}$  为得到的第  $j$  维最大值。  $\text{Rand}(0,1)$  为  $[0,1]$  区间内的随机数。通过以上种群内互相协作结合的方式, 蜜源会不断地优于上一代, 在经过多次迭代后, 达到算法所预设的次数或者达到精度范围内的所需解。以此达到求最优解的过程。

## 1.2 基于博弈论的云计算资源调度模型

在资源调度系统中, 首要保证系统资源高效利用和用户最小等待, 此时就需要采用资源调度算法将任务合理的分配给不同的服务器节点, 提升处理效率和用户满意度<sup>[11]</sup>。用户任务用  $T_i$  表示, 则  $m$  个任务可表示为  $T = \{T_1, T_2, \dots, T_m\}$ 。系统中的服务器节点为  $VM_i$ , 则  $n$  个服务器节点可表示为  $V = \{VM_1, VM_2, \dots, VM_n\}$ , 对于任意的节点  $VM_j$ 。

采用如下定义描述该模型, 调度任务定义如下:

$$T_i = \{\text{Lenth}, \text{Filesize}, \text{Cost}\}$$

式中: Lenth 为任务长度; Filesize 为任务文件大小; Cost 为任务花费。

节点模型如下:

$$VM_j = \{\text{CPU}, \text{Memory}, \text{Bandwidth}, \text{Cost}\}$$

虚拟节点考虑因素分别为 CPU、内存大小、网络带宽以及调度收益成本。云计算资源调度拓扑结构见图 1。

结合非合作博弈理论, 构建基于非合作博弈的调度模型, 博弈模型可以用一个五元组表示, 模型定义如下

$$G = \{\text{Time}, \omega, C, (S_p(t)), (P(t)_p)\}$$

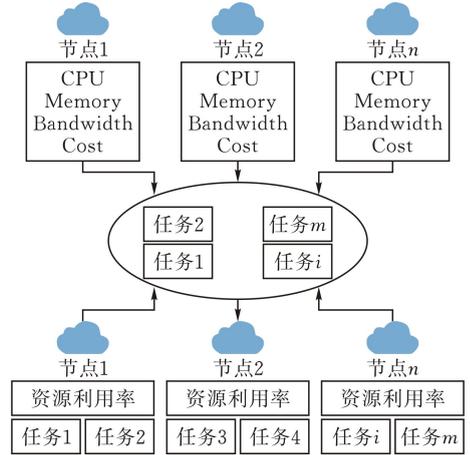


图 1 云计算资源调度结构图

其中: Time 表示资源调度中时间耗费;  $\omega$  表示节点间负载均衡状况;  $C$  表示用户任务花费;  $S_i(t)$  表示任务  $T_i$  与节点  $VM_j$  之间在  $t$  阶段的分配序列组成的全部可选策略集合; 也是任务  $T_i$  可以采取的调度策略。  $P(t)_i$  为任务在  $t$  阶段由调度成本取倒数所得到的效用函数, 用于衡量策略的可行性。全部的调度序列构成的策略集合  $S = \prod_{i=1}^m S_p$  构成博弈的策略空间。

下面给出非合作博弈模型和相关的表达式<sup>[8]</sup>。

1) 资源调度时间耗费 在资源调度系统中,  $T_i$  表示第  $i$  个任务,  $VM_j$  表示第  $j$  个节点, 因此任务  $T_i$  在节点上  $VM_j$  会对应一个时间耗费, 用  $\text{Time}(T_i \rightarrow VM_j)$  来表示。那么对于一个完整的执行序列  $K$  必然对应一个总时间耗费 Time, 如

$$\text{Time}_k = \max_{j=1}^m \sum_{i=1}^m \text{Time}(T_i \rightarrow VM_j) \quad (6)$$

式中,  $m$  表示任务数量,  $n$  表示服务器节点数量。

2) 资源调度节点负载均衡 节点负载均衡同样是纳入考虑的重要因素, 用  $\omega$  表示, 即

$$\omega = \sqrt{\frac{\sum_{j=1}^n (VMl_j - aVMl_j)^2}{n}} \quad (7)$$

式中,  $VMl_j$  表示节点  $VM_j$  负载,  $aVMl_j$  表示节点的平均负载,  $\omega$  值越小说明系统负载均衡度越好, 此时效率越高。

3) 资源调度中用户任务需求花费 另一个重要因素是任务花费, 用  $C$  表示用户支付费用, 如

$$C = \sum_{j=1}^n P_j \cdot (Ccpu_j + Cram_j + Cbw_j), \quad P_j = \text{sum}(j)$$

式中:  $Ccpu_j$ ,  $Cram_j$ ,  $Cbw_j$  为悉尼节点的特征量。分别代表节点 cpu 花费成本, 内存花费成本, 网络带宽花费成本。该式体现了资源调度服务按需付费的原则, 合理地将任务分配到不同的计算节点上, 使用

户以最低的成本来达到目的。

4)效用函数 在博弈中,对于某个特定的策略序列,该序列下的单位调度成本的倒数为效用函数,用于衡量该调度的效用:

$$p_i(k) = 1/c_i^k, \quad \forall k \in n$$

其中:  $S$  表示调度策略序列集合,  $c_i^j$  表示节点  $VM_j$  分配给任务  $T_i$  的计算资源成本,所有的节点中心  $VM_j$  分配给调度任务  $T_i$  的资源的效用满足叠加原则,如:

$$p_i(N) = \sum_{k=1}^m p_i(k), \quad \forall k \in m$$

5)目标函数 资源调度系统的最大目的为保持各个节点之间公平均衡调度的前提下实现系统最低调度成本和最大化总效用。定义表述目标函数如:

$$F(p_i) = \max \sum_{i=1}^m p_i(N)$$

### 1.3 基于 Nash 均衡算法求解

Nash 均衡点在非合作博弈下具有以下几个特点,任何博弈参与者在单独改变行动策略的前提下都不会增加效用函数,因此利用 Nash 均衡求解能得到云计算资源调度的最优调度序列<sup>[8-9]</sup>。

定理 1:在任务  $m$  的需求非合作博弈中,假如不同的任务最佳反应函数  $O_i(S_{-i})$  满足如下条件:

1)对于任务  $q$  的调度策略  $S_q (\forall q \neq i)$ ,  $O_i(S_{-i})$  是  $S_q$  的可微函数;

$$2) \sum_{i \neq 1, i \neq q}^m \left| \frac{\partial O_i(S_{-i})}{\partial S_q} \right| \leq \gamma_0 < 1$$

则此次任务  $T_i$  的非合作博弈必定存在 Nash 均衡点。

证明:先做以下定义,  $S = R^i$  中的距离,挑选任意两点  $s^{(1)} = \{s_1^{(1)}, s_1^{(1)}, \dots, s_i^{(1)}\}$ ,  $s^{(2)} = \{s_1^{(2)}, s_1^{(2)}, \dots, s_i^{(2)}\}$ , 因此两点间距离为  $d_{S^{(1)}}^{S^{(2)}} = \sum_{i=1}^m |s^{(1)-(2)}|$ 。

此外,  $\forall S = (s_1, s_2, \dots, s_m)$ , 根据文中定义可知  $S$  为策略空间,定义  $K_s = (O_1(S_{-1}), O_2(S_{-2}), \dots, O_m(S_{-m}))$ , 于是有

$$d_{KS^{(1)}}^{KS^{(2)}} = \sum_{i=1}^m |O_i^{(1)} - O_i^{(2)}|$$

由拉式中值定理可知:

$$O_i^{(1)} - O_i^{(2)} = \sum_{q \neq 1, q \neq i}^m \frac{\partial O_i(\varphi_i)}{\partial S_q} \Big|_{S_{-i} = \varphi_i(s_q^{(1)} - s_q^{(2)})}$$

式中,  $\varphi_i$  是  $s_q^{(1)}$  和  $s_q^{(2)}$  连线上的某一点。即:

$$d_{KS^{(1)}}^{KS^{(2)}} = \sum_{i=1}^m \left| \sum_{q \neq 1, q \neq i}^m \frac{\partial O_i}{\partial S_q} S_{-i} = \varphi_i(s_q^{(1)}, s_q^{(2)}) \right| \ll \sum_{i=1}^m \sum_{q \neq 1, q \neq i}^m \frac{\partial O_i}{\partial S_q} |s_q^{(1)} - s_q^{(2)}| = \sum_{q=1}^m \sum_{i \neq 1, i \neq q}^m \frac{\partial O_i}{\partial S_q} |s_q^{(1)} - s_q^{(2)}| \ll$$

$\sum_{q=1}^m \gamma_0 |s_q^{(1)}, s_q^{(2)}| = \gamma_0 \sum_{q=1}^m |s_q^{(1)} - s_q^{(2)}| = \gamma_0 d(s^{(1)}, s^{(2)})$   
由上述条件可知  $0 \ll \gamma_0 \ll 1$ , 因此  $K$  是  $S$  到自身的一个缩小因子;在  $S$  中一定有一个满足  $K_s^* = s^*$  的点  $s^* = \{s_1^*, s_2^*, \dots, s_m^*\}$ , 因此可知  $O_i(s_{-i}^*) = s_i^* (i = 1, 2, \dots, m)$ , 故得到  $s^*$  为任务  $q$  的非合作博弈的 Nash 均衡点<sup>[10]</sup>。

## 2 仿真实验结果分析

实验环境:Inter i7-8086k CPU 4.0GHZ; 系统平台为 Windows7 操作系统; 软件为 MATLAB2018, Cloudsim3.0 仿真平台。本文实验使用谷歌 Cluster trance 数据作为参考。表 1 收集了谷歌集群 5 天内的各项参数变化。

表 1 节点中心部分参数

编号	工作量	任务量	任务运行时间/s	CPU 需求	内存需求/GB	硬盘需求/GB
1	8	280	[0,66]	[0,4]	[0,5]	[0,35]
2	9	190	[0,91]	[0,4]	[0,7]	[0,27]
3	12	175	[0,75]	[0,5]	[0,9]	[0,38]
4	13	210	[0,110]	[0,5]	[0,10]	[0,19]
5	10	230	[0,55]	[0,4]	[0,8]	[0,15]

本实验将经典的群智能优化算法的 ACO 算法(蚁群优化算法), PSO 算法(粒子群优化算法)和本文结合博弈论的改进 ABC 算法(后文统一用 BG-ABC 算法作描述)做对比, 并从多个角度比较分析两个算法在实际运用中的不同点。实验结果见图 2、图 3。

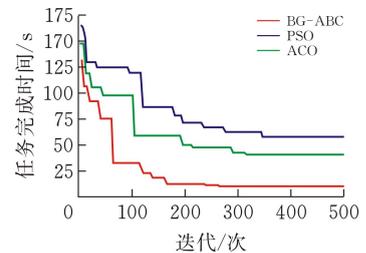


图 2 任务完成总时间对比图

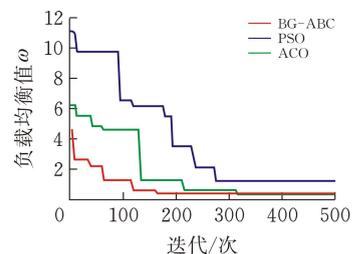


图 3 节点间负载均衡值 ω 对比图

由图 2 可知,对于资源调度任务完成总时间, BG-ABC 算法较 PSO 算法和 ACO 算法具有更高的速度, PSO 算法和 ACO 算法相差不大。由图 3 可知,采用 BG-ABC 算法的云计算资源调度系统,其

系统的负载均衡度值相较于 PSO 和 ACO 算法更小,说明其系统负载均衡度更好,相关公式参见式(2-7)。

为进一步验证改进算法的高效性,这里增加实验,使用 Schaffer 测试函数来验证,理论上最优解为 0,即当实验结果越接近 0 表示越精确。Schaffer 函数具体表达式如下:

$$f = 0.5 + \frac{\sin \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001 \times (x_1^2 + x_2^2))^2}$$

实验对比见图 4。实验结果表明,BG-ABC 算法在测试函数求最优值上具有高效性和精准性,精确度高于 ACO 算法和 ABC 算法。

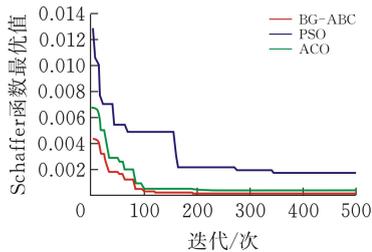


图 4 Schaffer 函数寻优结果对比

### 3 结束语

本文提出了基于博弈论的云计算资源调度优化模型,并以人工蜂群算法为前提。该模型考虑了调度时间,节点间负载均衡度,资源调度花费,效用函数和目标函数。将改进后的 GB-ABC 和传统 PSO 算法、ACO 算法同时运用在资源调度下,并对实验结果进行对比,同时利用测试函数进一步证明 BG-ABC 算法的有效性。结果表明,BG-ABC 算法较同类型的其他算法效果更佳。

#### [ 参 考 文 献 ]

[1] 王波,张晓磊.基于粒子群遗传算法的云计算任务调度

研究[J].计算机工程与应用,2015,51(6):84-88.

- [2] 王虎,雷建军,万润泽.基于改进的粒子群优化的云计算资源调度模型[J].Journal of Central China Normal University, 2018, 52(6):560-562.
- [3] KASABOGA D.An idea base on honey bee swarm for numerical optimization[J].Techn. Rep. TR06. Kayseri: Erciyes University,2005,(14):1899-1902.
- [4] KASABOGA D. A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm [J]. Journal of Global Optimization, 2007,39(3):459-471.
- [5] OSORIO I,ZAVERI H,FREI MG, et al.Epilepsy: the intersection of neurosciences, biology, mathematic, engineering and physics[M].Florida: Sanibel Symposium and IWSP4.CRC,2017:127-132.
- [6] HASSAN M,SONG B,HUB E N.Game based distributed resource allocation in horizontal dynamic cloud federation platform[M].Berlin: Algorithms and Architectures for parallel Processing. Springer, 2019: 194-205.
- [7] LIU HONGZHI, GAO LIQUN, KONG XIAN-GYONG, et al.modified artificial bee colony algorithm based on subpopulation[J].Journal of Northeast University(Natural Science Edition), 2018, 35(9): 1239-1243.
- [8] VOUK M A. Cloud Computing-Issues, research and implementations[J].Journal of Computing and Information Technology,2018,4:235-246.
- [9] 龚小庆.合作演化复杂性[M].杭州:浙江工商大学出版社,2017:12.
- [10] SCHARLEMANN J P W, ECKEL C C, KACELNIK A, et al. The value of a smile: Game theory with a human face[J]. Journal of Economic Psychology, 2007, 22(5):617-640.
- [11] 徐昕.基于博弈论的云计算资源调度方法研究[D].上海:华东理工大学,2015.

## Research on Application of Improved Artificial Bee Colony Algorithm in Cloud Computing Resource Scheduling

WU Ao, CHEN Hongwei

(School of Computer Science, Hubei Univ. of Tech., Wuhan 430068, China)

**Abstract:** There exist some problems with traditional optimization algorithms, such as low utilization rate, high task scheduling cost and unbalanced load among server nodes, in cloud computing resource scheduling. Therefore, this paper proposes an improved artificial bee colony algorithm based on game theory to improve the efficiency of traditional resource scheduling algorithms. The simulation experiment results show that the improved artificial bee colony algorithm based on game theory can significantly improve the efficiency and effectiveness of resource scheduling compared with the original algorithm and the same type of algorithm.

**Keywords:** cloud computing resource scheduling; game theory; artificial bee colony algorithm

[责任编辑:张岩芳]