

引用格式:黄伟建,辛风俊,黄远. 基于混沌猫群算法的云计算多目标任务调度[J]. 微电子学与计算机,2019,36(6):55-59.
[HUANG W J,XIN F J,HUANG Y. Multi-objective task scheduling based on chaos cat swarm optimization in cloud computing[J]. Microelectroncs & Computer,2019,36(6):55-59.]

基于混沌猫群算法的云计算多目标任务调度

黄伟建,辛风俊,黄 远

(河北工程大学 信息与电气工程学院,河北 邯郸 056038)

摘 要: 对于云计算中多目标任务调度问题,提出了一种基于混沌猫群算法(chaos cat swarm optimization,CCSO)的多目标任务调度模型.该模型中把任务执行时间和系统负载均衡做为优化目标.模型中使用的调度算法通过搜寻和跟踪两种模式以及 Logistic 混沌映射对实验数据进行处理,进而得到最优任务调度解集.在 CloudSim 仿真平台上,将实验结果与遗传算法和粒子群优化算法进行比较.结果表明混沌猫群算法不仅缩短了任务执行时间也使系统负载更加趋于均衡,从而能更高效的完成云计算中多目标任务调度.

关键词: 云计算;任务调度;混沌猫群算法;执行时间;负载均衡;多目标优化

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2019)06—0055—05

Multi-objective task scheduling based on chaos cat swarm optimization in cloud computing

HUANG Wei-jian,XIN Feng-jun,HUANG Yuan

(School of Information and Electrical Engineering, Hebei University of Engineering,
Handan 056038 ,China)

Abstract: Aiming at the multi-objective task scheduling problem in cloud computing, A multi-objective task scheduling model based on chaotic cat optimization algorithm (CCSO) is proposed. The scheduling algorithm used in the model processes the experimental data by searching and tracking two modes and Logistic chaotic map, and then obtaining the optimal task scheduling solution set. On the CloudSim simulation platform, making the simulation results compared with genetic algorithm(GA) and particle swarm optimization algorithm(PSO). The results show that the chaotic cat group algorithm not only shortens the time of task execution but also makes the system load more balanced, so that the multi-objective task scheduling in cloud computing can be completed more efficiently.

Key words: cloud computing; task scheduling; chaos cat swarm optimization; execution time; load balancing; Multi-objective optimization

1 引言

近年来,云计算以其计算的便捷、高效成为大家热烈讨论的话题,也成为各个互联网公司争相研究的对象^[1].云计算中的任务调度策略直接影响着云计算的效率.云计算任务调度主要研究如何为用户

提交的任务进行更为合理的资源分配,以满足用户的时间更小、负载均衡、花费更少等目的.目前研究较多的云计算任务调度算是启发式任务调度算法^[2].基于启发式任务调度算法中较为经典的遗传算法和粒子群算法,人们提出过很多任务调度模型.早期的调度模型中单一优化目标很难满足大众的需

求. 为了满足不同的需求,王国豪等提出一种改进的遗传算法,该算法实验过程中对工作流截止时间、任务跨度和执行代价进行了优化^[3]. 封良良等提出云计算环境下基于改进粒子群的任务调度算法,该算法对时间和成本起到了优化的效果^[4]. 但文献[3,4]忽略了负载均衡这一问题. 本文使用的调度算法为混沌猫群算法 (Chaos Cat Swarm Optimization, CCSO), 并以此为基础提出一种基于混沌猫群算法的多目标任务调度模型. 在其任务调度过程中将执行时间和负载均衡作为优化目标. 算法调度过程中的两种模式(跟踪模式与搜寻模式)与 Logistic 混沌映射使得全局搜索和局部寻优成为可能.

2 云计算中任务调度分析与模型建立

2.1 任务调度模型

云计算任务调度过程基于虚拟机执行任务的过程可分为两级任务调度:一级调度主要是用户与虚拟机之间的调度,二级调度主要是主机与虚拟机之间的调度. 一级调度需解决资源处理与用户任务匹配问题即任务到虚拟机的调度问题,二级调度主要解决主机与虚拟机的映射问题. 显然,本文研究的任务属于一级调度. 在此任务调度过程中,用户提交的大任务,分割成 n 个较小的子任务,然后运用任务调度策略并遵循负载均衡、最优跨度、QoS 等原则将这些子任务合理的分配给相应的 m 个计算资源节点来执行($n>m$). 调度模型如图 1 所示.

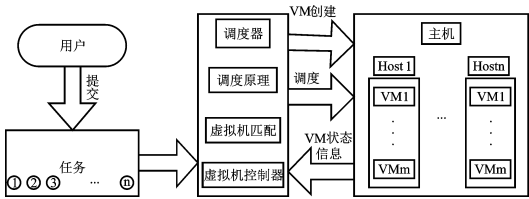


图 1 任务调度模型

如图 1 中任务调度过程中涉及到用户、任务、调度器、虚拟机. 用户提交任务,数据中心根据任务调度机制来分配虚拟机数量及任务执行次序,虚拟机按要求完成任务调度. 这里所说的调度机制就是用户所使用的不同调度算法.

2.3 服务质量

云计算过程中,为了满足客户的要求,必须按照用户所描述的服务质量协议中的服务质量(Quality of Service, QoS)要求来提供服务^[5]. 当前服务质量要求中的主要问题之一是按用户的期望的 QoS 级别在满足最小执行成本和执行时间的前提下提供所

需服务. 另一个问题是云计算过程中消耗大量资源,随着社会资源的多元化,云资源管理的复杂性也日益加剧,需制定更有效、合理的管理计划. 为制定的管理计划更为合理,需从资源的流向上多加关注. 资源的进一步调度时需在调度策略的支配下完成资源的调度即任务调度. 调度策略中,可将资源消耗、成本、执行时间、系统负载均衡作为 QoS 考虑的参数,从而有效的规范和调度资源.

3 混沌猫群算法下的任务调度

3.1 算法简介

混沌猫群算法 (Chaos Cat Swarm Optimization, CCSO) 是在猫群算法 (Cat Swarm Optimization, CSO) 的基础上改进得来. 猫群算法由 Chu Shu-An 等人在 2006 年首次提出^[6]. 猫群算法具有易陷入局部极值点的“早熟”和跟踪精度差的问题, Yang Shi - Da 等(2013 年)提出一种混沌猫群算法,利用不同混沌映射改进搜寻模型的步长,研究 7 种不同的混沌映射,找到最佳的物流与正弦图^[7]. 算法中 logistic 混沌映射^[8]来改变每次迭代过程中猫群的位置信息,避免了 CSO“早熟”问题的出现,使得整个种群能够在全局搜索能力和局部搜索能力之间达到一个平衡,而且具有良好的收敛速度. 目前,猫群算法及其改进算法在函数优化、图像处理、模式识别、神经网络训练等领域获得了一定应用并且取得了较好的效果. 对于混沌猫群算法,提出的时间较晚,在实际应用方面较少. 聂晓华等将混沌猫群算法应用于光伏最大功率的求解上^[9]. 在多目标云计算任务调度中,极少的涉及到混沌猫群算法.

3.2 算法的执行过程

混沌猫群算法的求解过程如下:

- (1)初始化猫群;其中包括初始化猫群的位置:位置和速度分量、种群大小为 N 、最大迭代次数 K_{max} , 分组率 (Mixture Ratio, MR) 等. 其中猫群的活动范围为 $[a, b]$. 分组率 MR 的大小决定了处于跟踪模式的猫的个数多少,根据实际情况下大多数猫处于慵懒的状态所以大部分猫为搜寻模式少数为跟踪模式,从而 MR 的值一般设置的较小.
- (2)计算种群中所有猫的适应度值并记录其中最大适应度值.
- (3)根据 MR 将猫群随机分组;判断猫下一步处于跟踪模式还是搜寻模式.
- (4)跟踪模式
- (5)搜寻模式

(6)计算各个体适应度值,保留种群中适应度最优的猫。

(7)判断是否满足终止条件,若满足则输出最优解,结束程序;否则,利用混沌映射更新猫的位置重复流程(3)~(6)进行寻优迭代处理.一般算法的终止条件有两种:迭代次数达到上限和最优解满足要求,本文采用的是执行到最大迭代次数。

混沌猫群算法流程图如图2。

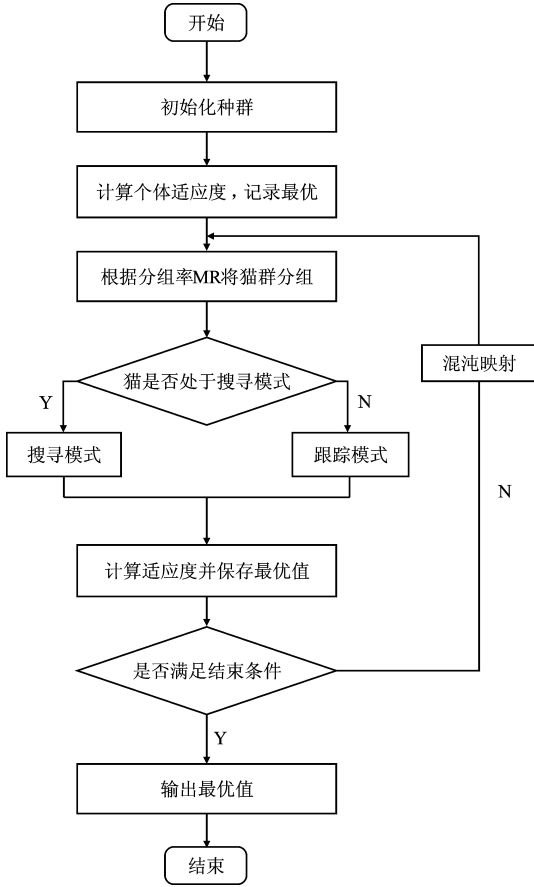


图2 算法流程图

混沌猫群算法的伪代码:

开始

初始化种群,随机产生 n 个猫,最大迭代次数设为 $kmax$

计算个体适应度,记录最优适应度值

While(目前迭代次数<设定的最大迭代次数)do

根据分组率将猫分组

for 猫群处于搜寻模式 do

进入搜寻模式更新猫的速度和位置

end for

For 猫未处于跟踪模式

进入跟踪模式更新猫的速度和位置

end for

计算更新后的猫群适应度并保存最优值

混沌映射猫群位置

end while

输出最优解

结束。

3.3 混沌映射

混沌猫群算法算法中的映射采用 logistic 混沌映射,其表达式为:

$$x(n+1) = u \times x(n) \times (1 - x(n)) \quad (1)$$

式中, n 为迭代次数; u 是混沌参数; $u \in [0, 4]$, $x \in [0, 1]$. 经典 logistic 映射中 u 是一个变量,从公式(11)中可以看出 u 影响着 logistic 映射的状态. u 的取值也受到 x_n 的影响,当 u 增加时, x_n 的取值范围也增大,且分布较均匀. 当 u 的值增加到 4 时, logistic 映射也趋向于满射,此时的 x_n 序列分布最为均匀. 进一步通过计算证实这一点. 经计算当 $u = 4$ 时且 $x \in [0, 1]$, 且 $x \notin \{0.25, 0.50, 0.75\}$ 时,系统所处的混沌状态是安全的. 设常规变量为 $Cx(n)$, 把此变量用公式(12), (13) 与混沌变量进行映射变换,其公式如下:

$$x(n) = [Cx(n) - a] / (b - a), n = 1, 2, \dots, N \quad (2)$$

$$Cx(n) = a + x(n) \times (b - a), n = 1, 2, \dots, N \quad (3)$$

式中, a, b 为猫群活动范围; $Cx(n)$ 表示为猫群所处的位置。

logistic 混沌映射的加入避免了“早熟”问题的出现,使得整个种群能够在全局搜索能力和局部搜索能力之间达到一个平衡,在资源分配任务调度方面有良好的优势。

3.4 两种更新模式

本文所提算法中涉及到的两种模式:跟踪模式和搜寻模式。

3.4.1 跟踪模式具体执行操作

(1)用公式(4)和(5)更新每只猫的速度变量并与设定的速度边界值比较,如果超出范围则将其设置为其邻近的边界值。

$$V_{k,d,t+1} = V_{k,d,t} + r_1 \times c_1 \times (X_{best,d,t} + X_{k,d,t}), \quad d = 1, 2, \dots, M \quad (4)$$

$$V_{k,d,t+1} = \begin{cases} V_{max,d,t} & V_{k,d,t} > V_{max,d,t} \\ V_{min,d,t} & V_{k,d,t} < V_{min,d,t} \end{cases} \quad (5)$$

(2)用公式(6)和(7)更新每只猫的位置变量并与设定的位置边界值比较,如果超出范围则将其设置为其邻近的边界值。

$$X_{k,d,t} = X_{k,d,t-1} + V_{k,d,t} \quad (6)$$

$$X_{k,d,t+1} = \begin{cases} X_{max,d,t} & X_{k,d,t} > X_{max,d,t} \\ X_{min,d,t} & X_{k,d,t} < X_{min,d,t} \end{cases} \quad (7)$$

式中, $X_{\text{best},d,t}$ 是第 k 个猫的适应度函数最优值, r_1 是常量, c_1 是变量, 通常取值范围为 $[0,1]$.

3.4.2 搜寻模式具体执行操作

(1)复制副本:创建记忆池可放置最大数目的当前猫的副本,并判断布尔型变量 SPC 的值,SPC 决定了此刻的位置下一刻是否会有猫转移至此,若 SPC 为真,创建猫的副本数减一,并记录此位置为记忆池 SMP 的候选位置.

(2)改变:根据随机数与维度该变量 CDC 大小对比,决定对上述产生的副本增大或减小 SRD 比率的值来更新位置大小.在此之前初始化猫群的活动维度.

(3)计算每一个产生的副本的适应度值,如若都不同根据公式(8)计算其被选择的概率,若适应度值都相同,则选择的概率设置为 1.

$$P_1 = \frac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, 0 < i < j \tag{8}$$

式中, FS 代表适应度; FS_b 代表最优的适应度. 如果 FS 是求解极小值,则 $FS_b = FS_{\max}$, 否则 $FS_b = FS_{\min}$.

(4) 根据适应度大小,选择适应度较大的代替当前猫的位置.

3.5 算法的编码

算法在模拟任务调度的过程中,采用不同的编码方式来实现调度. 本文采用的是二维数组编码. 这种编码方式简单易懂,在操作过程中降低了算法的复杂度. 如本文在算法初始过程中产生个体对应的解集 $\{\{1,2\},\{1,1\},\{2,2\},\{3,2\},\{4,3\},\{5,4\},\{6,3\},\{7,4\},\{2,1\}\{8,2\}\}$,解集中的 10 个解对应初始化过程中的 10 个个体,每一个解中第一个数值代表对应的任务编号号,第二个数值代表虚拟机编号. 如其中 $\{3,2\}$ 表示 3 号任务分配到 2 号虚拟机上,其它与此解释类似.

3.6 多目标优化目标度函数设计

本文采用混沌猫群算法对多目标函数进行优化,函数的优化目标为执行时间和负载均衡. 在种群选取过程中,适应度作为选择的条件,适应度大的个体更优良,而适应度函数是根据用户需求设定的. 根据适应度原理设 x_i^t 为第 i 只猫经过 t 次迭代得到的位置,则设计适应函数来计算其值大小:

$$F_1(x_i^t) = \frac{1}{\log_2 Exetime(x_i^t)} \tag{9}$$

$$F_2(x_i^t) = \frac{1}{\log_2 VMload(x_i^t)} \tag{10}$$

式中,公式(9)是某一个体的执行时间的的适应度,公式(10)是某一个体的负载计算的适应度函数. 为

将两个相差较大的函数整合到一起,把每一个函数值取对数再取倒数最后相加得到目标函数为:

$$Fitness(x_i^t) = F_1(x_i^t) + F_2(x_i^t) \tag{11}$$

4 实验结果与分析

4.1 实验参数的设置

本文实验过程中涉及到的三种算法分别是遗传算法、粒子群优化算法、混沌猫群算法. 将遗传算法与粒子群优化算法作为比较算法是因为启发式算法中较为经典的算法有遗传算法和粒子群优化算法,后来提出的算法大都是在此基础上的改进或融合如文献[10],且遗传算法与粒子群算法优化在多目标任务调度方面应用较为广泛且效果较好. 三种算法主要涉及到的参数及其大小如下表 1:

表 1 各算法参数设置

算法	参数	取值
GA	交叉率	0.6
GA	变异率	0.001
PSO	学习因子 C1	2
PSO	学习因子 C2	2
PSO	惯性权重 W	0.8
CCSO	追踪模式比率 MR	0.1
CCSO	记忆池长度 SMP	5
CCSO	维度变化范围 SRD	0.1
CCSO	维度变异率	0.8
CCSO	Logistic 映射控制参数 U	4

除算法本身的参数不同,三种算法在相同的环境下运行. 即在 Cloudsin 仿真平台上进行仿真实验,实验模拟了将 8 个任务分配到 4 个虚拟机上运行. 实验环境: Intel® Core™ i3—2350M, 2.3 GHz 主频, 4.0 GB 内存, MATLAB2016a. .

4.2 实验结果及分析

实验过程中设种群的规模为 10,种群的迭代次数为 100,将三种算法在此环境下运行 20 次,得到的实验结果即为每次最优适应度值. 其值如图 3 所示.

从图 3 可以得出本文提出的算法在相同的实验次数下可得到最优适应度值的次数远远超出其对比算法,也证明了该算法具有更强的稳定性.

实验可知遗传算法和粒子群算法的执行时间在迭代初期小于本文算法,但当迭代次数超过 70 代之后,本文算法的执行时间明显小于其它两种对比算法. 在迭代的后期,本文算法与遗传算法相比执行时

间减少了 20.67%,与粒子群优化算法比减少了 37.78%.

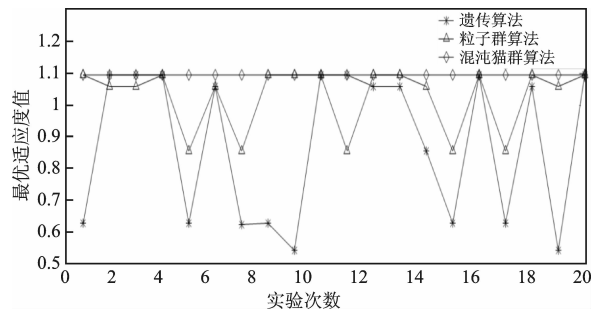


图 3 三种算法下最优适应度值对比

三种算法在随着迭代次数的增加系统负载值有不同程度的降低. 相比较于遗传算法和粒子群算法,混沌猫群算法的系统负载值是在较小的范围内波动,且在迭代后期其负载值明显小于对比算法的负载值.

5 结束语

提出了将混沌猫群算法应用到云计算中多目标任务调度中,并建立多目标任务调度模型. 调度模型中将执行时间和负载均衡作为优化对象,并以此为基础设计出目标函数. 调度算法采用混沌猫群算法,分别从理论分析与实验实验结果上证明了算法的可行性. 实验结果表明算法在迭代次数较大的时候,执行时间和负载均衡度均优于其对比算法. 但算法在实际实验过程中会产生一些异常数值,所以下一步考虑算法的改进. 优化目标方面考虑加入成本、资源利用率等.

参考文献:

[1] SHEN W,DOU W C, WU F, et al. Combined cloud;a mixture of voluntary cloud and reserved instance marketplace[J]. 计算机科学技术学报(英文版), 2016, 31(6):1110-1123.

[2] NIU W, FENG Z, CHENG C, et al. Parallel multi-objective optimal operation of cascaded hydropower system[J]. Journal of Hydraulic Engineering, 2017, 48(1):104-112.

[3] 王国豪,李庆华,刘安丰. 多目标最优云 workflow 调度进化遗传算法[J]. 计算机科学, 2018, 45(5): 31-37.

[4] 封良良,张陶,贾振红,等. 云计算环境下基于改进粒子群的任务调度算法[J]. 计算机工程, 2013, 39(5): 183-186.

[5] SINGH S, CHANA I. Resource provisioning and scheduling in clouds: QoS perspective[J]. Journal of Supercomputing, 2016, 72(3):926-960.

[6] CHU S C, TSAI P, PAN J S. Cat swarm optimization [J]. Lecture Notes in Computer Science, 2006(6): 854-858.

[7] Yang S D, Yi Y L, Shan Z Y. Chaotic cat swarm algorithms for global numerical optimization[J]. Advanced Materials Research, 2012(5): 602-604.

[8] Chen S, Zhang S F. Improved algorithm for digital chaos sequences in full mapping[J]. Journal of Computers, 2017, 28(6):100-113.

[9] 聂晓华,王薇. 混沌改进猫群算法及其在光伏 MPPT 中的应用[J]. 中国电机工程学报, 2016, 36(22): 6103-6110.

[10] MANASRAH A M, ALI H B. Workflow scheduling using hybrid GA-PSO algorithm in cloud computing [J]. Wireless Communications & Mobile Computing, 2018(3):1-16.

作者简介:

黄伟建 男,(1964-),博士,教授. 研究方向为云计算、并行计算、信息管理与信息系统.

辛风俊(通讯作者) 女,(1992-),硕士研究生. 研究方向为云计算. E-mail:1941127187@qq.com.

黄 远 男(满族),(1987-),博士,讲师. 研究方向为数据挖掘、信息处理、预测算法研究.