

基于改进的粒子群算法的云资源调度策略

蔡晓丽, 钱 诚

(常州工学院 计算机信息工程学院, 江苏 常州 213002)

摘 要: 云计算资源的调度是云计算中的一项关键技术. 针对粒子群算法存在易陷入局部最优解和“早熟”的缺陷, 提出一种改进的粒子群算法. 通过改进粒子迭代过程中社会项系数和认知项系数的权重变化, 使算法更符合最优解的求解规律, 避免陷入局部最优解. 仿真实验表明, 改进后的粒子群算法适应度更强、收敛速度更快, 具有更强的全局搜索能力. 该算法可以有效提高云计算资源的利用率, 具有良好的应用价值.

关键词: 云计算; 粒子群算法; 资源调度; 迭代; 适应度

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2018)06-0028-03

Cloud Resource Scheduling Strategy Based on Improved Particle Swarm Optimization

CAI Xiao-li, QIAN Cheng

(School of Information & Engineering, Changzhou Institute of Technology, Changzhou 213002, China)

Abstract: Cloud resource scheduling strategy is the most important technology in cloud computing. In order to solve the problem of being easy to fall into locally optimal solution, we proposed an improved particle swarm optimization algorithm (IPSO). Coefficient about social and personal cognition are changed according to the times of iteration, which can comply with objective rules much better and avoid falling to locally optimal solution. Simulation results show that IPSO can improve the availability of cloud resources. It has higher global search ability, better fitness, and good application value.

Key words: cloud computing; particle swarm optimization; resource scheduling; iteration; fitness

1 引言

近几年中, 云计算与物联网、移动互联网、大数据等多种技术相互依存、相互促进, 并作为一种革命性的商业计算模型得到了迅速发展. 云计算利用虚拟化技术, 将网络中千差万别的计算资源统一管理和调度, 将庞大的计算任务拆成多个小的任务, 交由云服务器集群, 经过搜寻、计算后将结果快速地反馈给用户^[1]. 在海量的数据面前, 云计算资源的快速调度成为影响云计算服务性能的关键因素. 云计算调度方法分静态和动态两种. 静态方法例如轮循调度法, 其假设各种信息是可知的. 它沿用网格计算模式, 实现简单, 但调度开销大、容易负载失衡, 难以满

足日益复杂的云计算服务要求^[2]. 云计算系统在对数量巨大的任务进行资源分配及调度时, 涉及到的用户众多、各资源节点结构不同且时刻动态变化, 这需从任务完成时间、能耗、负载均衡等多个目标综合考虑, 但目前这方面的研究较少. 动态的云计算调度方法是将云计算资源调度看成多约束的优化问题, 先将问题转化为某个数学模型, 然后找求解方法^[3]. 许多学者对云计算的资源调度问题进行了研究, 引入了不少算法, 例如蛙跳算法、遗传算法、粒子群算法、蚁群算法等^[4-5]. 它们有较好的自适应性, 但也存在着各自的不足. 其中, 粒子群算法 (particle swarm optimization, PSO) 不仅具有参数设置少, 全局搜索能力的特点, 而且算法简单易实现, 计算量小

收稿日期: 2017-04-20; **修回日期:** 2017-05-22

基金项目: 国家自然科学基金(61602063); 常州工学院课题(A3-4403-17-011)

的优点,所以成为当前云资源调度的主要研究方向^[6].但是,粒子群算法也存在一些缺点,主要是容易陷入局部的最优解及“早熟”现象.

为了解决传统粒子群算法的不足,本文通过改进其自适应能力,更新粒子速度更新公式中认知项系数与社会项系数在迭代过程中的权重,使算法更符合客观规律.经仿真实验证明,改进的粒子群算法(improved particle swarm optimization algorithm,简称 IPSO)能有效减少任务完成时间,保持节点的负载均衡,同时能有效克服局部最优及“早熟”现象,提高云计算资源的利用率.

2 云计算资源调度的数学模型

云计算资源调度中,要以尽可能短的时间完成任务.首先将任务划分成 m 个相互独立的子任务,子任务的集合用 $T = \{t_1, t_2, \dots, t_m\}$ 表示,其中第 j 个任务表示为 t_j ($1 \leq j \leq m$).每个子任务仅分配到一个虚拟资源节点运行,设虚拟资源节点共有 n 个 ($n < m$),用集合 $VM = \{vm_1, vm_2, \dots, vm_n\}$ 表示,其中 vm_i 表示第 i 个虚拟资源节点 ($1 \leq i \leq n$),子任务分配到节点上的情况可用分配关系矩阵 B 表示成:

$$B = \begin{bmatrix} b_{11} & b_{12} \dots & b_{1m} \\ b_{21} & b_{22} & b_{2m} \\ & \dots & \\ b_{n1} & b_{n2} & b_{nm} \end{bmatrix}$$

式中,任一元素 b_{ij} 代表子任务 t_j 和虚拟资源节点 vm_i 的对应关系,当第 j 个任务在第 i 个节点上运行时, b_{ij} 为 1,否则设为 0.

$$b_{ij} \in \{0, 1\} \quad (1)$$

为研究方便,设一个虚拟计算节点一次只处理一个任务,一个任务只在一个节点上执行,即:

$$\sum_{i=1}^m b_{ij} = 1 \quad (1 \leq j \leq m) \quad (2)$$

完成任务所花费的时间 TM 应该是子任务完成时间中的最大值,即:

$$TM = \max(\sum_{j=1}^m b_{ij} t_{ij}) \quad (1 \leq i \leq n) \quad (3)$$

式中, $\leftrightarrow$ 代表估计任务 t_j 在节点 vm_i 上运行的执行时间.设计一个好的云计算资源调度算法,应使完成任务花费的时间最少,即完成时间:

$$Cost = \min TM \quad (4)$$

综上所述,云计算资源调度的数学模型为式(1)~(4).

3 改进的粒子群资源调度算法

3.1 传统的粒子群算法简介

传统的粒子群算法 PSO 也称为“鸟群觅食算法”,它源于对鸟群聚集行为及迁徙行为的研究.它通过追随当前搜索到的最优值来寻找全局最优值.鸟在某区域内寻找食物的有效方法,就是搜寻目前距离食物最近的鸟的位置.经过寻找和交流,不断靠近目标,最终找到食物^[6].粒子群算法中,由 m 个粒子组成的粒子群在一个 N 维空间中,用 L_i 表示第 i 个粒子的矢量位置 $\vec{L}_i = (l_{i1}, l_{i2}, \dots, l_{iN})$.用 V_i 表示粒子的速度,即 $\vec{V}_i = (v_{i1}, v_{i2}, \dots, v_{iN})$.

当前该粒子的最佳位置为 $pbest_i$,群体最佳位置是 $gbest_i$,则该粒子的速度更新公式为:

$$V_{i+1} = \omega \cdot V_i + c_1 \cdot r_1 \cdot (pbest_i - L_i) + c_2 \cdot r_2 \cdot (gbest_i - L_i) \quad (5)$$

式中, ω 是惯性权重,代表粒子受惯性影响的因素;系数 c_1, c_2 是学习因子; r_1, r_2 是随机数,值均匀分布于 0 到 1 之间. $c_1 \cdot r_1 \cdot (pbest_i - L_i)$ 是认知项, $c_2 \cdot r_2 \cdot (gbest_i - L_i)$ 是社会项.

粒子位置的更新公式为:

$$L_{i+1} = L_i + V_i \quad (6)$$

通过公式(5)、(6)对粒子的速度、位置不断迭代更新,并计算、分析粒子的适应度,从而最终找到适应度最大的最优解.

3.2 改进的粒子群算法

在公式(5)中,学习因子 c_1 也被称为认知项系数,它与粒子个体相关; c_2 称为社会项系数,它与群体相关,它们在传统 PSO 算法中是一直保持不变的.这使得在迭代次数较少时,由于缺少群体社会经验的积累,算法受个体认知影响较大;反之,在迭代次数较大时,群体社会知识积累得越来越多,算法受社会部分影响越来越大.由于 c_1 和 c_2 值固定,难以反映上述变化,导致算法陷入局部最优解及“早熟”现象.

因此,可对迭代公式(5)进行改进,在第 m 次速度迭代过程中,公式采用:

$$V_{i+1} = \omega \cdot V_i + a \cdot \frac{1}{m} \cdot r_1 \cdot (pbest_i - L_i) + b \cdot m^2 \cdot r_2 \cdot (gbest_i - L_i) \quad (7)$$

c_1 用 $a \cdot \frac{1}{m}$ 代替,反映随着迭代次数的增加,个体认知所占比重降低, c_2 用 $b \cdot m^2$ 代替,使得迭代次数越大,群体社会项所占比重逐渐变大.在初始阶

段,令 a 取较大值, b 取较小值,这符合初始阶段个体认知项是主导因素;以及随着迭代深入,社会项比重越来越大,并逐渐成为主要影响因子的情形.这样改进速度计算公式后,更契合求解最优解的客观规律,可提高算法实用性和有效程度,防止陷入局部最优解.

4 基于 IPSO 的云资源调度算法设计

在 IPSO 算法中,通过迭代,每个粒子都更接近最优解的值,迭代一次即搜索一次更优解,直至得到最优解.

4.1 粒子编码与适应度函数设计

设任务数 m ,虚拟资源节点数 n ,此时粒子空间维度为 $Q=m\times n$,粒子种群大小为 GroupSize.粒子的位置信息、速度信息和适应度值共同构成了粒子的编码,其中位置和速度均为 GroupSize $\times Q$ 维变量.编码时位置信息表示成 $b_{11}\cdots b_{1Q}b_{21}\cdots b_{2Q}\cdots b_{m1}\cdots b_{mQ}$,粒子速度表示成 $v_{11}\cdots v_{1Q}$,适应度为 $f()$,定义为

$$f()=\frac{\beta}{1+TM}$$

(8)

β 的取值与具体任务相关, TM 代表完成子任务所花费的最大时间,该时间越小,说明调度方案越佳.

4.2 改进算法的具体执行步骤

(1)用随机化方式初始化粒子群矩阵 SW 和最优解矩阵 OPT,任务数为 m ,虚拟节点数 n ,迭代次数 Count;

(2)计算每个粒子的适应度,存入粒子群矩阵 P 中;

(3)根据粒子群矩阵 SW 确定群体、个体最优解 gbest_{*i*}、pbest_{*i*},更新最优解矩阵 OPT 中对应的个体最优值,并将全局最优值存入 OPT 矩阵的最后一行中;

(4)计算粒子群矩阵 SW 中每个粒子的新位置及速度,判断是否达到迭代次数或达到精度要求.若是,则输出最优解;否则,执行(2).

5 实验结果与分析

为了比较改进前后的粒子群算法在任务完成速度及性能上的优劣,进行了实验验证.本实验环境为 Windows7 操作系统,处理器为 Intel Core i7,内存 8G,及 Matlab2016 软件.为合理评估算法性能,实验分小任务量和较大任务量两种情况.设标准粒子群算法中, $\omega=1,c_1=c_2=1$.为了突出 IPSO 算法中

社会项比重随着迭代次数的增加而增加,个体认知项则逐渐减少的特点,设公式(7)中 $\omega=1,a=80,b=0.001$.设计虚拟资源节点 3 个($vm_1\sim vm_3$),任务 6 个($T_1\sim T_6$),模拟小任务量时的粒子群变化.粒子空间维度为 18,粒子种群大小取 300,迭代 200 次,任务与节点交叉点表示该任务在该节点上的预计完成时间,具体如表 1 所示.

表 1 任务在各节点预计完成时间

	T_1	T_2	T_3	T_4	T_5	T_6
vm_1	6	5	13	9	24	26
vm_2	9	4	11	10	25	24
vm_3	8	3	15	11	22	21

实验结果如图 1 所示,IPSO 算法和 PSO 算法执行 200 次迭代搜寻最优解.在 80 次前,IPSO 优势不明显;随着迭代次数增至 80 次以上,IPSO 算法的适应度开始超越 PSO 算法,很快地 PSO 算法陷入局部最优解.在迭代 100 次以后,IPSO 算法最优解趋于稳定,而 PSO 在迭代 120 次左右才达到最优解.因此,IPSO 完成任务所需时间更短,适应度更高,验证了任务量较小时,改进的粒子群算法 IPSO 效果更好.

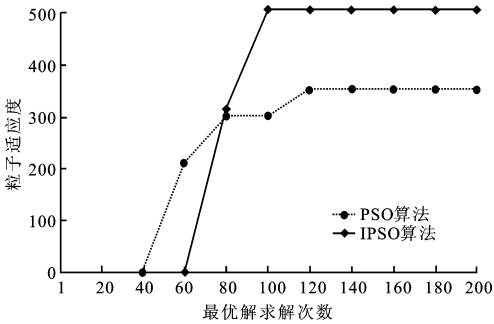


图 1 小任务量实验结果

当任务量较大,设有 600 个任务,10 个虚拟资源节点,初始状态下各任务在节点上预计的执行时间取[10,40]内的随机数,粒子种群群数 200,迭代次数 8 000 次.从实验结果图 2 所知,IPSO 的适应度一直高于 PSO 算法,IPSO 更快地找到了最优解,因此 IPSO 的收敛性也强于 PSO.

由大小不同规模的实验得知,无论在小规模,还是大规模下,IPSO 算法在搜寻最优解的速度和质量上均高于传统 PSO 算法,IPSO 算法收敛性更强,提高了全局搜索能力,防止陷入局部最优解,减少了云计算资源调度的时间,是比传统 PSO 算法更优的调度方案.

[4] 张征凯,谷立臣,曹向辉,等. 基于最优小波包的电机定子电流故障特征提取方法[J]. 煤矿机械, 2016, 37(8):149-151.

[5] 王少福,张金磊,赵仕俊,等. 广义回归神经网络的改进及在预测控制中的应用[J]. 微电子学与计算机, 2009, 26(6):32-35.

[6] 郁磊. MATLAB 智能算法 30 个案例分析. 第 2 版[M]. 北京航空航天大学出版社, 2015.

[7] 洪波,刘龙,王涛. 修正型果蝇算法优化 GRNN 的大梁自动焊障碍预测[J]. 焊接学报, 2017, 38(1): 73-76.

[8] Pan W T. Pan, W. T. A new Fruit Fly opti- mization Algorithm; Taking the financial distress model as an example. Knowledge-Based Systems 26, 69-74 [J]. Knowledge-Based Systems, 2012, 26(2):69-74.

[9] 祁丽婉,梁庚,童国炜. 基于果蝇算法优化 BP 神经网络

的齿轮箱故障诊断[J]. 电网与清洁能源, 2014(9): 31-36.

[10] 田源,张彼德,刘代伟,等. 基于果蝇优化算法的 GRNN 水电机组状态趋势预测[J]. 水电能源科学, 2012(12):127-129.

作者简介:

郭庆 男,(1962-),教授,硕士生导师. 研究方向为嵌入式测控系统、微弱信号检测.

吴汝琴 女,(1992-),硕士研究生. 研究方向为心电检测与分类.

徐翠锋(通信作者) 女,(1977-),硕士,高级实验师. 研究方向为信号处理、测控技术等.

E-mail:343183324@qq.com.

(上接第 30 页)

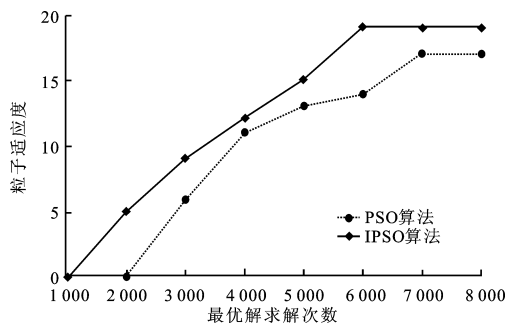


图 2 大任务量实验结果

6 结束语

改进的粒子群算法 IPSO,通过改进 PSO 算法中个体项和社会认知项公式,更能反映粒子位置、速度更新与迭代次数之间的关联性. 通过不同规模实验的对比,验证了 IPSO 算法的收敛性、适应性强于 PSO 算法,能有效解决云计算资源调度中易陷入局部最优解的问题. 改进的 IPSO 算法节约了云资源分配、完成任务的时间,是较理想的方案.

参考文献:

[1] Lee Y C, Zomaya A Y. Energy efficient utilization of

resources in cloud computing systems[J]. Journal of Supercomputing, 2012, 60(2):268-280.

[2] Iosup A, Ostermann S, Yigitbasi M N, et al. Performance analysis of cloud computing services for many-tasks scientific computing[J]. IEEE Transactions on Parallel & Distributed Systems, 2011, 22(6): 931-945.

[3] 孙大为,常桂然,李风云,等. 一种基于免疫克隆的偏好多维 QoS 云资源调度优化算法[J]. 电子学报, 2011, 39(8):1824-1831.

[4] 孟令玺,李洪亮. 基于 CA-PSO 算法的云计算资源调度策略[J]. 计算机仿真, 2013, 30(10):406-410.

[5] 刘卫宁,靳洪兵,刘波. 基于改进量子遗传算法的云计算资源调度[J]. 计算机应用, 2013, 33(8): 2151-2153.

[6] 袁正午,李君琪. 基于改进粒子群算法的云资源调度[J]. 计算机工程与设计, 2016, 37(2):401-404.

作者简介:

蔡晓丽 女,(1981-),硕士,副教授. 研究方向为云计算、网络安全. E-mail:caixl@czu.cn.

钱 诚 男,(1981-),博士,讲师,研究方向为云计算、物联网和移动计算.