

基于改进粒子群算法的云计算任务调度算法

娄建峰¹,高岳林¹,李 飞¹,张克平²

(1 北方民族大学 计算机科学与工程学院,宁夏 银川 750021;2 中国移动通信集团宁夏有限公司,宁夏 银川 750021)

摘 要: 面对云计算中的大量任务,为了对其进行高效的调度,缩短任务完成时间并提高资源利用率,对基于粒子群算法的云计算任务调度算法进行了研究.首先用自然数对粒子编码表示粒子的位置,并在解空间内随机初始化种群,每个粒子的位置对应一个可行的调度方案.每次迭代更新后,对粒子进行修复操作,为降低粒子跑出解空间的概率,同时对粒子速度进行限定.针对传统粒子群算法易陷入早熟的缺陷,加入混沌扰动策略使种群跳出局部最优.通过 Cloudsim 仿真平台进行实验测试,实验结果表明该算法能取得更好的调度结果并且收敛速度更快.

关键词: 云计算;粒子群优化;离散;任务调度;混沌

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2016)08-0112-05

A Task Scheduling Algorithm Based on Improved Particle Swarm Optimization for Cloud Computing

LOU Jian-feng¹,GAO Yue-lin¹,LI Fei¹,ZHANG Ke-ping²

(1 College of Computer Science and Engineering, Beifang University of Nationalities, Yinchuan 750021, China;

2 China Mobile Group Ningxia Co.,Ltd., Yinchuan 750021, China)

Abstract: Facing with large amount of tasks in cloud computing, a task scheduling algorithm based on improved Particle Swarm Optimization(PSO) was taken into research to minimize the task completion time and maximize the resource utilization. Firstly, the velocity and position of each particle are set randomly within the search space in the initialization. A swarm of particles are used to represent the potential solutions and the position of each particle is encoded by natural number. After each iteration update, legalized method was used to repair the particle. In order to reduce the probability of particles running out of the solution space, some method was taken to limit the velocity of each particle. For overcoming precocious, chaos was combined with PSO. By using chaos disturbance, the particles can have a better position. Cloudsim was used to stimulate cloud computing environment for experimental test. Experimental results show that compared with tradition PSO the improved PSO converges faster and have a better scheduling result.

Key words: coud computing; PSO; discrete; task scheduling; chaos

1 引言

2007 年,随着 IBM-Google 的分布式计算项目,云计算的概念开始兴起并迅速发展.云计算是分布式计算、并行计算、网格计算、效用计算、网络存储、虚拟化等技术的发展融合,是已经实现的基于网络

的商业服务模式.目前,云计算服务被划分为三个层次:基础设施即服务 (Infrastructure-as-a-Service, IaaS)、平台即服务 (Platform-as-a-Service, PaaS) 和软件即服务 (Software-as-a-Service, SaaS)^[1].

云计算中软件硬件都是资源,它将计算任务分布在大量计算机构成的资源池上,并通过网络以服

收稿日期: 2015-11-17; **修回日期:** 2016-01-10

基金项目: 国家自然科学基金项目 (61561001);北方民族大学重点科研项目资助(2015KJ10);北方民族大学研究生创新项目 (YCX1562)

务的方式提供给用户. 这些资源在物理上以分布式共享方式存在,可以根据需要动态地扩展和配置. 用户可以按需获取计算能力、存储空间、网络带宽和信息服务等.

云数据中心是容纳计算设备资源的集中之地,在结构上是一个分布式网络,由计算节点、存储节点以及网络设备连接而成. 随着云计算的发展和数据中心规模的不断扩大,动态分配管理和共享虚拟资源面临新的挑战,常用的传统调度算法如轮转法、加权轮转法、最小负载优先等,并不能满足要求而且比较容易造成服务器负载不均衡. 为满足不同的业务需求,需要设计更为高效的云计算资源调度分配算法.

云计算中的任务调度是一个 NP 完全问题^[2],粒子群、差分进化、遗传等智能优化算法比较适合于求解 NP 类问题,而且这些算法已经被应用到云计算任务调度中. 文献[3]中将加入新的选择机制的 DE 算法应用到云计算任务调度中,文献[4]提出了一种基于多策略变异差分进化的任务调度算法,添加了多种变异策略和基于正态分布的分类,改善了差分算法收敛速度慢的缺点,文献[5]中运用鲑鱼粒子群算法对云任务进行调度,增强粒子群算法的全局搜索能力,文献[6]中详细分析了基于群体智能的进化算法在云计算中的应用,文献[7]介绍了智能优化算法在云计算中应用的现状.

粒子群算法具有参数简单、早期收敛速度快等优点,本文将加入混沌扰动的离散粒子群算法应用到云计算任务调度中,使任务的总完成时间最短,同时实现负载均衡. 对于粒子群后期种群多样性减少易陷入早熟问题,通过加入混沌扰动策略,避免了种群过早的收敛,取得了更优的调度结果.

2 云计算任务调度描述

在云计算中,往往要将逻辑上完整的大任务划分成若干个子任务,然后根据适当的策略将这些子任务分配到不同的资源节点上. 目前,大部分的云计算平台采用的是 Google 提出的 MapReduce^[8]编程模式. MapReduce 是一种处理海量数据的并行编程模式,用于大规模数据集的并行运算. 每个 Map 函数对其中一部分原始数据进行处理,Map 之间相互独立可以并行计算. Reduce 操作就是对每个 Map 操作产生的部分中间结果进行合并,所有 Reduce 结果简单连接就形成了完整的结果集. MapReduce 运行模型如图 1 所示.

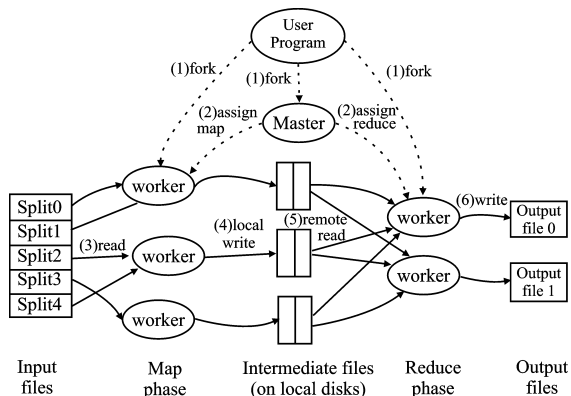


图 1 MapReduce 的运行模型

3 标准粒子群算法

粒子群优化算法^[9]是由美国的 Kenney 和 Eberhart 受鸟群觅食行为的启发提出的一种基于群体智能理论的全局优化方法,算法通过群体中粒子间的信息共享提供群体智能优化搜索. 该算法模拟鸟类觅食过程,采用速度-位置搜索模式,每个粒子根据对个体和群体的飞行经验的综合分析来动态调整自己的速度和位置,在解空间中进行搜索,根据适应度函数,通过迭代找到最优解.

一个由 NP 个粒子组成的群体在 D 维搜索空间以一定的速度飞行,群体中任一粒子由 3 个 D 维向量组成,第 i 个粒子的当前位置 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 、历史最优位置 $P_{bi} = (p_{i1}, p_{i2}, \dots, p_{iD})$ 、速度 $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. 每个粒子在更新自己的速度和位置时都根据自己搜索到的历史最好点和群体内其他粒子搜索的历史最好点两个值来更新,粒子根据如下公式更新自己的速度和位置:

$$v_{ij}^{k+1} = \omega v_{ij}^k + c_1 r_1 (pbest_{ij}^k - x_{ij}^k) + c_2 r_2 (gbest_{ij}^k - x_{ij}^k) \quad (1)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2)$$

式中, ω 代表惯性权重, v_{ij}^{k+1} 表示第 i 个粒子在第 $k+1$ 次迭代时在 j 维的速度; c_1, c_2 表示粒子个体的学习因子和粒子群体的学习因子; r_1, r_2 为 0 到 1 之间均匀分布的随机数; x_{ij}^{k+1} 表示第 i 个粒子在第 $k+1$ 次迭代时在 j 维的位置; $pbest_{ij}^k$ 表示第 i 个粒子在第 k 次迭代时个体历史最优位置; $gbest_{ij}^k$ 表示第 k 次迭代时的全局最优位置. 其次为了防止粒子飞出最大解空间,通常限制 $v_{ij} \in (-v_{max}, v_{max})$.

4 基于改进粒子群算法的云任务调度

4.1 粒子编码

将粒子群算法应用到云计算任务调度中,首先

要做的是对粒子进行编码,将云任务调度与粒子的位置、速度等结合起来.由于云计算中任务只能取离散值,对粒子的编码也只能是离散的数值.本文中对粒子采用自然数编码,粒子位置的编码长度为子任务数量,即粒子的维度等于调度的任务数.设有 task 个任务,共划分为 m 个子任务,云环境下有 n 个资源,则粒子可编码为式子(3) 所示的 n 维向量:

$$\boldsymbol{P} = \{p_1, p_2, p_3, \cdots, p_m\} \tag{3}$$

式中, $1 \leq p_i \leq n$, 粒子每一维坐标表示一个子任务编号,任意一维分量 p_i 表示分配给此子任务的资源号.当任务个数 task = 3,子任务个数 $m = 10$,云环境资源个数 $n = 5$ 时,即 3 个任务被划分为 10 个子任务,分配到 5 个资源上,粒子可编码为(3,2,5,4,3,2,1,1,5,3),如表 1 所示.该粒子代表一个可行的调度方式,对粒子解码即可知任务分配情况,如表 2 所示.其中子任务 7,8 分配到 1 号资源上,子任务 2,6 分配到 2 号资源上,子任务 1,5,10 分配到 3 号资源上,子任务 4 分配到 4 号资源上,子任务 3,9 分配到 5 号资源上.

表 1 粒子编码示例

子任务号	1	2	3	4	5	6	7	8	9	10
粒子编码 (资源号)	3	2	5	4	3	2	1	1	5	3

表 2 粒子解码示例

资源号	1	2	3	4	5
分配的任务号	7 8	2 6	1 5 10	4	3 9

4.2 种群初始化

种群规模设为 NP,对 m 个子任务在 n 个资源进行调度,则种群初始化时系统随机产生 NP 个粒子,每个粒子的位置由向量 $\boldsymbol{P}$ 表示,第 i 个粒子:

$$p_i = \{p_{i1}, p_{i2}, \cdots, p_{ij}, \cdots, p_{im}\},$$

式中, $1 \leq p_{ij} \leq n$, p_{ij} 表示任务 j 分配到第 p_{ij} 号资源上.由于任务数是离散的,初始化时将 p_{ij} 的值取为 1 到 n 之间的随机整数.粒子速度由向量 $\boldsymbol{V}$ 表示:

$$v_i = \{v_{i1}, v_{i2}, \cdots, v_{ij}, \cdots, v_{im}\}$$

式中, $1 \leq i \leq NP$, $-n \leq v_{ij} \leq n$, v_{ij} 初始化为 $(-n, n)$ 之间的随机数.

4.3 适应度函数

适应度函数是评价粒子好坏的标准,粒子以适应度函数值的大小为目标,不断迭代更新,直到满足停止条件.本文使用 $m \times n$ 阶矩阵 ETC 来表示子任务在云资源上运行的时间,其中,ETC(i, j) 表示子任务 i 在云资源 j 上运行的时间,子任务 i 不在资源 j 上运行时,ETC(i, j) = 0. 根据粒子解码结果和 ETC 矩阵,可以得出资源 j 执行完分配到该资源的

所有子任务的时间 Time(j):

$$\text{Time}(j) = \sum_{i=1}^m \text{ETC}(i, j), j = 1, 2, \cdots, n \tag{4}$$

各个资源上的任务都运行完成后,则所有任务都执行完毕,任务的总完成时间 FTime:

$$\text{FTime} = \max(\text{Time}(j)), j = 1, 2, \cdots, n \tag{5}$$

定义适应度函数为

$$\text{Fitness} = \text{FTime} \tag{6}$$

4.4 混沌扰动

混沌搜索是一种看似无规则的搜索机制,但其内在规律,能够随机不重复地遍历所有状态.本文利用 Logistic 映射产生混沌序列,这是一个典型的混沌系统.其方程为

$$z_{n+1} = \mu z_n (1 - z_n) \tag{7}$$

当粒子种群迭代一定次数之后,收敛速度开始变慢,用混沌序列对全局最优粒子进行扰动.先将最优粒子的每一维映射到(0,1) 区间,生成一个 m 维取值范围为(0,1) 的向量 $\boldsymbol{D} = (d_1, d_2, \cdots, d_m)$, $d_i = (\text{gbest}_i - a)/(b - a)$,其中 d_i 表示粒子的第 i 维,gbest _{i} 表示最优粒子的第 i 维, a 和 b 分别表示粒子每一维取值的下界和上界.然后以向量 $\boldsymbol{D}$ 为初始值,利用式(7) 迭代产生新的序列: $z_1 = (z_{11}, z_{12}, \cdots, z_{1m})$,将 z_1 作为新的粒子,计算适应度值,若 z_1 的适应度值优于目前的全局最优位置,则将 z_1 更新为全局最优位置.

4.5 更新位置和速度

每一次迭代,粒子的速度和位置按式(1)、(2) 进行更新.由于任务调度是离散问题,采用了自然数编码方式,位置按式(2) 更新后,粒子位置取值会变为非自然数值,某些维度可能超出实际的取值范围,需要对这些粒子进行离散化处理和越界处理,本文对 p_{ij} 采用取绝对值、取下整、求余的方式:

$$p_{ij} = \begin{cases} \lfloor |p_{ij}| \rfloor & 0 \leq |p_{ij}| < n \\ \lfloor |p_{ij}| \rfloor \% n & |p_{ij}| \geq n \end{cases},$$

式中, $\lfloor |p_{ij}| \rfloor$ 表示对 p_{ij} 取绝对值并向下取整, $\lfloor |p_{ij}| n \rfloor \% n$ 表示对 $\lfloor |p_{ij}| \rfloor$ 除以 n 求余.

粒子速度按公式更新后,也可能超出范围使粒子直接飞出可行范围,需要设置最大飞行速度 $V_{\max}$,当 $v_{ij} > V_{\max}$ 时,取 $v_{ij} = V_{\max}/2$.

4.6 算法描述

基于改进的粒子群算法的云计算任务调度算法实现流程如下所示.

Step1 种群初始化,在问题解空间里随机初始化 NP 个粒子的位置和速度,设定算法参数.

Step2 对粒子解码得出该粒子的分配方式,根据任务列表和资源列表计算出 ETC 矩阵.

Step3 由 ETC 矩阵算出每个粒子的适应度函数值.

Step4 根据粒子适应度函数值更新粒子的个体最优值和全局最优值.若全局最优值更新,则更新全局最优位置 gbest.

Step5 当种群迭代次数 $t > 30$ 且最优位置连续 5 次没有更新时执行 Step6,否则执行 Step7.

Step6 对全局最优位置进行混沌扰动,先将最优粒子的每一维映射到 $(0,1)$ 区间,产生一个 m 维的向量 $D=(d_1,d_2,\cdots,d_m)$,然后按照混沌映射公式,产生一个新的粒子 $z_1=(z_{11},z_{12},\cdots,z_{1m})$.

Step7 计算新产生粒子的适应度值,并与全局最优值比较,若新粒子的适应度值更优,则将新粒子位置更新为全局最位置.

Step8 判断迭代次数 t 数是否达到最大迭代次数 MaxNum,若 $t > \text{MaxNum}$,执行 Step10.

Step9 按照式(1)、(2)更新粒子的速度和位置,执行 Step2.

Step10 输出全局最优位置解码结果,作为最佳调度方案.

5 仿真实验

CloudSim^[10]是由墨尔本大学 Rajkumar Buyya 教授领导开发的云计算仿真环境,用户可以通过 Cloudsim 在虚拟环境中实现并测试自己的调度算法.本文使用 CloudSim-3.0.3 模拟云环境进行调度算法的测试实验,通过修改 DatacenterBroker 类中的方法,实现了基于混沌扰动的离散粒子群算法在云任务调度中的应用,并进行测试实验.实验中只考虑资源的处理速度和待处理任务的长度,并以所有任务的总完成时间作为评价指标.以传统的粒子群算法作为对比,在同等的实验条件下进行测试.

本文取种群规模为 60,算法主要参数设置如表 3 所示.

表 3 算法主要参数	
参数	取值
种群规模 NP	60
计算资源数量 n	5
任务数量 m	20,50,200,500
惯性因子 ω	0.9
学习因子 c_1	2
学习因子 c_2	2

实验中分别测试在任务数为 20、50、200 和 500 时,不同迭代次数情况下,按照两种算法得到的调度策略进行调度的任务总完成时间,其中任务数为 500、200、50 时任务的大小是由函数在 $[500 \text{ MI}, 5\,000 \text{ MI}]$ 范围内随机产生的一组数据,当任务数为 20 时,为增加算法的对比度,任务的大小由函数在 $[5\,000 \text{ MI}, 50\,000 \text{ MI}]$ 范围内随机产生,计算资源的处理能力由函数在 $[200 \text{ MIPs}, 400 \text{ MIPs}]$ 范围内生成.每个迭代次数下都运行 50 次取平均值,实验结果如图 2~5 所示.

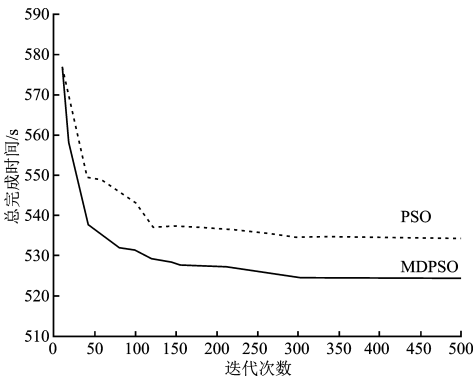


图 2 任务为 20 的总完成时间

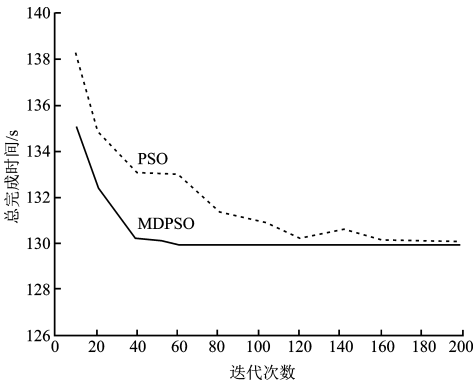


图 3 任务为 50 的总完成时间

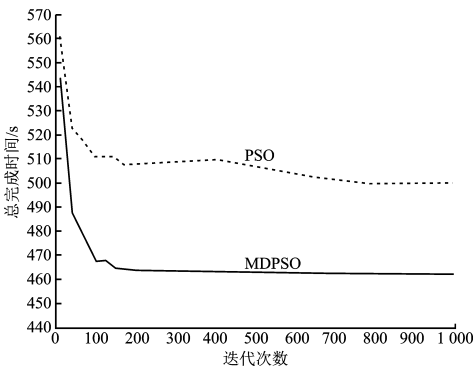


图 4 任务为 200 的总完成时间

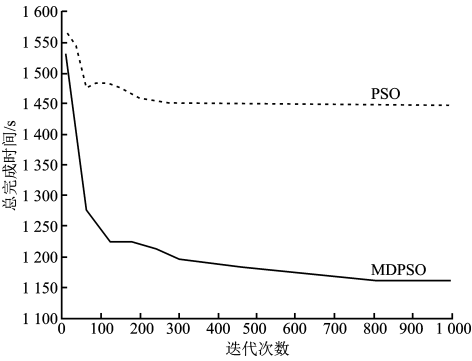


图 5 任务为 500 的总完成时间

由以上实验结果可以看出,传统粒子群算法初期收敛速度较快,但到后期收敛速度变慢且无法跳出局部极值过早的收敛.同传统 PSO 算法相比,当任务数较多或任务比较大时,改进的离散粒子群算法(Modified Discrete PSO,MDPSO)收敛速度更快而且收敛结果更好;当任务较小且数量较少时,几乎能达到同样的收敛结果,但 MDPSO 收敛速度更快且更稳定.

6 结束语

针对云计算中的任务调度问题,本文在传统 PSO 算法的基础上进行改进,对粒子位置和速度进行限定,并加入混沌扰动策略,以任务总完成时间作为算法的适应度函数,通过实验证明,改进的算法能够有效地使种群跳出局部极值并加快了收敛速度.本文以任务总完成时间为目标,而实际应用中还有计算成本、带宽、能耗等因素要考虑,有待进一步的研究.

参考文献:

[1] 刘鹏. 云计算[M]. 2 版. 北京:电子工业出版社,2011: 3-15.

[2] Arfeen M A, Pawlikowski K, Willig A. A Framework for Resource Allocation Strategies in Cloud Computing

Environment[C] // Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual. [s. l.]:IEEE,2011,261-266.

[3] 朱宇航,郑丽英,邬开俊. 基于改进 DE 的云计算任务调度算法[J]. 兰州交通大学学报, 2013, 32(1): 101-106.

[4] 董丽丽,黄贲,介军. 云计算中基于差分进化算法的任务调度研究[J]. 计算机工程与应用, 2014, 50(5): 90-95.

[5] 张彬桥. 云环境下计算资源调度策略与仿真研究[J]. 计算机仿真, 2013, 30(11):392-395.

[6] Bilgaiyan S, Sagnika S, Das M. An analysis of task scheduling in cloud computing using evolutionary and swarm-based algorithms[J]. International Journal of Computer Applications, 2014, 89(2):11-18.

[7] MG Huang, ZQ Ou, GY Song, et al. Review of task scheduling algorithm research in cloud computing[J]. Advanced Materials Research, 2014 (926/930): 3236-3239.

[8] Jeffrey D, Sanjay G. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1):107-113.

[9] Bratton D, Kennedy J. Defining a standard for particle swarm optimization[C] // Proc of IEEE Swarm Intelligence Symposium, Honolulu: IEEE, 2007.

[10] Calheiros R N, Ranjan R, Beloglazov A, et al. Cloud-Sim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2010, 41(1):23-50.

作者简介:

娄建峰 男,(1993-),硕士研究生.研究方向为智能信息处理与网络计算. E-mail:liffrank@163.com.

高岳林 男,(1963-),教授,博士生导师.研究方向为智能计算与智能信息处理、最优化理论方法及其应用.

张克平 男,(1963-),博士,教授.研究方向为网络管理.

李飞 女,(1991-),硕士研究生.研究方向为智能信息处理与网络计算.