

粒子群算法在云计算任务调度中的应用

苏淑霞

(北方民族大学计算机科学与工程学院,宁夏 银川 750021)

[摘要] 针对云计算的任务调度问题,在传统 PSO 算法的基础上,提出了一种改进粒子群算法的云计算任务调度方法.首先描述了云计算任务调度的数学模型和粒子群算法的基本原理,在此基础之上,采用间接编码方式对资源进行编码,定义了适应度函数,确立粒子速度和位置的更新方法.仿真实验表明,文中的方法获得了较好的调度结果.

[关键词] 粒子群,任务调度,适应度,云计算

[中图分类号] TP311 [文献标志码] A [文章编号] 1001-4616(2014)04-0145-05

Application of Particle Swarm Optimization Algorithm on Cloud Computing Task Scheduling

Su Shuxia

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

Abstract: Aiming at the problem of task scheduling of cloud computing, a task scheduling method was introduced based on particle swarm algorithm. Firstly, the mathematical model for cloud computing task scheduling and the principle of particle swarm optimization algorithm were described, and then the paper introduces the encoding of each subtask takes by indirect encoding, defines the fitness function, establish the particle velocity and position updating method. Experimental results show that the method of the paper obtained good scheduling results.

Key words: particle swarm, task scheduling, fitness, cloud computing

云计算作为一种新的基于服务的资源提供模式^[1],有着非常广泛的用户群体,时刻都在处理海量的任务.如何高效地调度用户提交的任务成为云计算中所要解决的重要问题之一.云计算的任务调度,即在一个特定的云环境中,根据一定的条件规约,将系统资源在不同的资源使用者之间进行调整的过程.

当前,云环境中的任务调度算法主要有传统任务调度算法、Hadoop 模型中的任务调度算法、智能化的任务调度算法等.前两类算法较为简单,但性能不佳,往往不能满足用户的需求.后一类算法通常存在收敛性能和搜索全局最优解方面能力较低以及调度目标单一的问题.

考虑到现有任务调度算法存在的不足,本文对传统的任务调度方法和现有的任务调度方法进行了深入的研究,提出了一种基于粒子群算法的任务调度方法,仿真实验验证了其良好的性能.

1 云计算任务调度研究现状

1.1 云计算任务调度的描述

在云计算环境中,一个大规模的计算任务的处理是一个分布式的并行处理过程.系统首先将逻辑上完整的大任务切分成若干个子任务,再根据任务的特征采取适当的调度算法,把若干子任务分配到不同的资源节点上运行,最后将各子任务的运行结果汇总,传给用户.

假设云计算中任务调度的 n 个任务和 m 个资源满足以下条件^[2]:①任务之间没有依赖关系;②资源是独占的,只有当任务执行完成或失败后,资源才会被重新分配.

若将云环境描述为^[3]:资源总数为 m ,对应集合 $R=\{r_1, r_2, r_3, \dots, r_m\}$,用户共提交了 P 个作业,表示为 $J=\{J_1, J_2, J_3, \dots, J_P\}$. Map/Reduce 模型将 P 个作业切分成 n 个任务,即 $T=\{T_1, T_2, T_3, \dots, T_n\}$,则云计算任务调度的实质就是将 n 个相互依赖的并行任务分配到有限的 m 个资源上高效地处理,使任务的总完成时间最小,同时系统的资源得到充分利用.云计算任务调度的目标就是对用户提交的任务实现最优调度.

1.2 云计算任务调度算法

目前,云计算中的任务调度算法可简单划分为3类.

1.2.1 传统的任务调度算法

传统的任务调度算法主要指来源于网格计算的任务调度算法,如 Min-min 算法、Max-min 算法、Sufferage 算法等. Min-min 算法^[4]的基本思想是将需调度的任务分配给具有最早执行开始时间和最快执行速度的资源. Max-min 算法同 Min-min 的调度方法类似,但又具有自身的特点,Min-min 算法先完成执行时间短的任务,后完成执行时间长的任务,而 Max-min 算法则相反. Sufferage 算法^[5]将任务的最早完成时间与次早完成时间的差值定义为 Sufferage 值,用该算法调度时总将资源分配到 Sufferage 值较大的资源节点上,避免遭受最大的损失.

由于网格计算和云计算具有一定的相似性,因而借鉴网格计算中的任务调度算法研究云计算中的任务调度具有一定的可行性.

1.2.2 Hadoop 模型中的任务调度算法

目前,云计算中大部分仍采用 Google 提出的 Map/Reduce^[6]编程模型. Hadoop 是云计算现有的代表技术之一,是对 Map/Reduce 编程模型的开源实现. Hadoop 系统中使用最多的是 FIFO 调度算法,该算法根据用户提交作业的先后时间和优先级的高低来进行调度.在新版本的 Hadoop 中,Facebook 和 Yahoo 的工程师又提出了两种算法即公平调度算法和计算能力调度算法.其中,Yahoo 提出的公平调度算法(Fair Scheduling)^[7]调度的核心目标是尽可能满足不同需求的用户能够有平等的机会使用系统资源,而 Facebook 的计算能力调度算法(Capacity Scheduling)^[8]的核心思想是通过建立作业队列管理和维护作业,保证每个作业都能占用各自需要的资源.

1.2.3 智能化的任务调度算法

近年来,国内外的专家学者们提出了很多启发性的智能算法,如遗传算法、蚁群算法、粒子群算法,以及这些算法的互相融合.

遗传算法(Genetic Algorithm, GA)^[9]是由 Holland 教授创造出的一种基于生物遗传和进化机制、适合复杂系统优化的自适应概率优化的算法.该算法在搜索过程中自动获取和积累有关搜索空间的知识,并自适应地控制搜索过程以求得最优解.

蚁群算法和粒子群算法都是模拟自然界动物觅食行为的一种新型模拟进化算法,是群体智能理论研究领域所研究的主要算法.所谓的群体智能是指具有简单智能的主体,通过合作而表现出复杂智能行为特征.本文通过深入研究粒子群算法的基本原理,并将其应用到云计算的任务调度中.

2 粒子群算法

粒子群算法(Particle Swarm Optimization, PSO)^[10]是一种仿生优化算法,该算法模拟于鸟类的觅食过程,算法采用速度-位置搜索模型,每个粒子代表解空间的一个候选解,算法的优劣由选择的适应度函数决定.在连续空间坐标系中,粒子群算法的数学描述如下^[11]:

粒子群中的任意一个粒子都是由3个 D 维向量组成,分别为:当前位置 $\mathbf{x}_i=(x_{i1}, x_{i2}, \dots, x_{iD})$ 、历史最优位置 $\mathbf{p}_{bi}=(p_{bi1}, p_{bi2}, \dots, p_{biD})$ 和速度 $\mathbf{v}_i=(v_{i1}, v_{i2}, \dots, v_{iD})$.

一个由 m 个粒子(particle)组成的群体以一定的速度在 D 维搜索空间飞行,每个粒子在搜索时根据两个值来改变自己的位置.两个值分别为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.

该算法在每一次迭代中,当前位置 \mathbf{x}_i 作为问题解被评价,如果 \mathbf{x}_i 好于历史最优位置 \mathbf{p}_{bi} ,则用 \mathbf{x}_i 重置 \mathbf{p}_{bi} .同时将整个粒子群搜索到的最好位置记为 \mathbf{g}_{bi} ,表示为 $\mathbf{g}_{bi}=(g_{bi1}, g_{bi2}, \dots, g_{biD})$.在标准粒子群算法中,每一个粒子速度和位置的更新如下.

$$V_{id} = \omega \cdot v_{id} + c_1 \cdot \text{rand}() \cdot (Pb_{id} - x_{id}) + c_2 \cdot \text{rand}() \cdot (gb_{id} - x_{id}). \quad (1)$$

$$X_{id} = x_{id} + v_{id}. \quad (2)$$

其中, c_1 和 c_2 是学习因子, 是两个非负数. $\text{rand}()$ 是范围在 $[0, 1]$ 内的一个随机数. ω 是惯性权重, 它决定粒子的先前速度对当前速度的影响程度, 起到平衡算法全局搜索和局部搜索能力的作用^[12].

粒子群算法的实现过程描述如下^[13]:

Step 1 粒子群初始化. 产生原始种群, 选取算法中所需参数, 在问题空间中产生粒子的位置与速度.

Step 2 评价粒子. 对每一个粒子, 评价 D 维优化函数的适应度值.

Step 3 更新最优.

(1) 比较粒子适用值及其个体最优值 p_{best} , 若粒子当前适用值优于 p_{best} , 则用当前粒子的位置重置 p_{best} 的值.

(2) 比较粒子的适用值及群体最优值 g_{best} , 若粒子适用值好于 g_{best} , 则用当前粒子位置重置 g_{best} 的值.

Step 4 更新粒子的速度和位置. 按照式(1)和式(2)改变粒子的速度和位置.

Step 5 停止条件. 循环回到 Step 2 直到满足循环终止条件. 通常是满足适用值和最大的迭代次数.

粒子群算法采用的速度-位置模型, 操作较简单, 避免进行复杂的遗传操作, 通过跟踪当前整个种群的最优粒子来找到问题解. 传统的粒子群算法在求解云计算任务调度问题上有着天然的优势.

3 任务调度中的粒子群算法

云计算中由于需要调度的任务量和资源数量相当庞大, 因而任务策略的好坏直接影响着整个系统的性能. 若算法的调度性能不佳, 不仅会增加任务的执行时间, 也会降低整个云环境的性能和用户的满意度. 目前常用的任务调度算法多是以任务的完成时间作为调度目标. 本文提出的粒子群算法, 通过对算法中参数的优化设置, 改善传统粒子群算法的性能, 尽量减少优良因子的丢失, 在一定程度上提高了云计算任务调度的效率和用户的满意度.

3.1 粒子群的初始化

粒子群算法实现的关键在于将粒子的位置、速度和算法的规则与所要研究的领域相结合, 并用恰当的方式表示出来. 对于云计算环境下的 n 个任务在 m 个处理节点的任务调度问题, 其编码方式可采用直接编码方式, 也可采用间接编码方式. 本文采用的是间接编码方式, 即对每个子任务占用的资源进行编码, 编码的长度取决于子任务的数量.

假设系统随机产生 S 个粒子, 第 i 个粒子的位置用向量 x_i 表示, 速度用向量 v_i 表示. 系统资源总数为 RC 个, 有 $TASK$ 个任务, 每个任务又分为若干个子任务, 子任务的总数为 $STASK$, 且满足 $STASK \geq RC$.

例如, 当 $TASK=3$, $RC=4$, $STASK=10$ 时, 粒子 $(4, 2, 1, 3, 2, 4, 1, 2, 1, 3)$ 就是一个可行的调度策略. 其表示的含义为: 3 个任务被划分成 10 个子任务; 粒子中的每个数值表示给予任务分配的资源号; 子任务 1、6 在 4 号资源上调度, 子任务 2、5、8 在 2 号资源上调度, 子任务 3、7、9 在 1 号资源上调度, 子任务 4、10 在 3 号资源上调度.

3.2 适应度函数

在粒子群算法中, 对应每个粒子的分配方案, 算法使用适应度函数来衡量粒子当前位置和速度的优劣. 适应度函数有效地反映出每个粒子与问题的最优解粒子之间的差距. 每次迭代时, 根据适应度值来不断寻找和更新个体最优和全局最优, 直到算法结束. 通常, 适应度函数的值越大, 解越优.

在任务调度问题上, 一般常用任务的总完成时间、任务的平均完成时间或任务总完成成本等来定义适应度函数. 本文是以任务的总完成时间最短作为调度的目标, 因而用任务的总完成时间来定义适应度函数.

本文使用 **ETC** 矩阵^[14]来记录任务的执行时间, $ETC[i, j]$ 表示第 i 个子任务在第 j 个资源上运行的时间. 完成所有任务的总时间如下:

$$SFT = \max_{r=1}^s \sum_{i=1}^n RT(r, i). \quad (3)$$

其中, S 为粒子群的规模, $RT(r, i)$ 表示第 i 个子任务在第 r 个资源上运行所用的时间. 文中采用的适应度函数定义为:

$$Fit(i) = \frac{1}{SFT_i}, \quad 1 \leq i \leq S. \quad (4)$$

算法在执行时,任务的总完成时间越小的粒子其适应度值越大,粒子越容易被选择.

3.3 权重因子

从式(1)可以看出,在粒子群算法中,权重因子 ω 用以控制前一次粒子的速度对当前粒子速度的影响,算法在执行时 ω 值的大小直接影响算法的寻优能力和收敛速度^[15]. 当 ω 较大时,粒子在当前解的附近进行搜索,因而具有较强的局部搜索能力;当 ω 较小时,则可扩大搜索范围,使解的多样性增加,提高算法的全局寻优能力.

本文为降低算法的复杂程度,通过多次试验,并参照文献^[16]将 ω 的值设定为 0.9.

3.4 粒子位置和速度的更新

粒子群在迭代过程中,粒子受个体最优和群体最优的指导,粒子会根据自身的历史最优位置和群体最优位置来更新粒子的当前速度和位置.

算法在执行时,会根据粒子的当前位置 $\mathbf{v}_i(t)$ 计算一个适应度值,只有当粒子的当前位置比粒子的历史最好位置具有更好的适应度值时,用粒子的当前位置来更新粒子的历史最优位置;当粒子的当前位置比群体粒子的历史最优位置具有更好的适应度值时,用粒子的当前位置值来更新群体的历史最优位置.

每一次迭代时,每一代的粒子要根据如下公式来更新自己的速度和位置:

$$\mathbf{v}_i(t+1) = \omega \times \mathbf{v}_i(t) + c_1 \times \text{rand}() \times (\mathbf{p}_{bi}(t) - \mathbf{x}_i(t)) + c_2 \times \text{rand}() \times (\mathbf{g}_{bi}(t) - \mathbf{x}_i(t)) \quad (5)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (6)$$

式中, t 表示迭代次数. 在整个迭代过程中,粒子的速度和位置被限定在一个特定的范围内,同时 \mathbf{p}_{bi} 和 \mathbf{g}_{bi} 不断更新,最后输出的 \mathbf{g}_{bi} 为全局的最优解.

4 仿真实验

4.1 仿真实验的环境及参数设置

为评价和分析本文的算法,实验采用云计算仿真工具 CloudSim 进行仿真实验. 算法中涉及的参数设置为:种群规模为 100,迭代次数为 100,惯性权重因子 ω 为 0.9, c_1 和 c_2 取值为 1,资源数为 50,任务数分别为 50 和 100.

4.2 实验的性能分析

任务数分别为 50 和 100 时,随迭代次数的变化,任务的总完成时间如图 1 所示.

由图 1 可知,随着粒子迭代次数的增大,本文算法所得到的任务完成时间在不断减少. 当任务数由 50 增加到 100 后,任务的总完成时间变化较大. 当任务数为 50 时,迭代到 20 次后,算法具有较好的收敛性;当任务数为 100 时,迭代 40 次后,算法具有较好的收敛性.

但该算法在运行过程中,由于无法有效地跳出局部寻优的搜索状态,导致算法过早地收敛到局部最优的调度结果上.

5 结语

本文针对云计算的编程模型,根据粒子群算法的基本思想,通过参数的优化设置,提出一种云计算的任务调度方法. 在进行任务调度时,充分考虑了任务的总完成时间. 实验结果表明,该算法在云计算环境下实现了较为理想的调度结果. 但算法只定义了一个适应度函数,出现了算法过早收敛到局部最优的调度结果. 下一步工作是研究如何提高任务调度的效率.

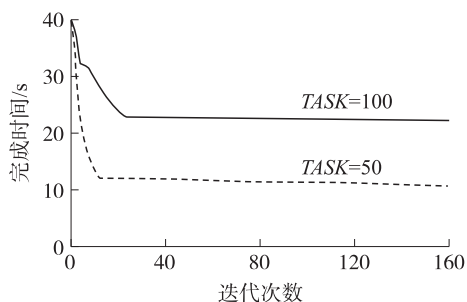


图1 任务的总完成时间

Fig. 1 The total completion time of the TASK

[参考文献]

- [1] 刘鹏. 云计算[M]. 北京:电子工业出版社,2007:2-3.
- [2] 查英华,杨静丽. 改进蚁群算法在云计算任务调度中的应用[J]. 计算机工程与设计,2013,34(5):1 716-1 719.
- [3] 熊聪聪,冯龙,陈丽仙,等. 云计算中基于遗传算法的任务调度算法研究[J]. 华中科技大学学报:自然科学版,2012,40(增刊):1-4.
- [4] Braun T D, Siegel H J, Beck N. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(1):810-837.
- [5] 郑爱卿. 基于执行时间方差的元任务网格调度算法研究[D]. 北京:北京交通大学电子信息工程学院,2008.
- [6] 封良良,张陶,贾振红,等. 云计算环境下基于粒子群的任务调度算法研究[J]. 计算机工程,2013,39(5):183-186.
- [7] Isard M, Prabhakaran V, Currey J, et al. Fair scheduling for distributed computing clusters[C]//Proceedings of the 22nd ACM SIGOPS Symposium on Operating Systems Principles. New York:ACM,2009:261-276.
- [8] 遆鸣. 云计算下计算能力调度算法的研究和改进[D]. 太原:太原理工大学计算机科学与技术学院,2012.
- [9] 王小平,曹立明. 遗传算法—理论、应用于软件实现[M]. 西安:西安交通大学出版社,2002:10-50.
- [10] Bratton D, Kennedy J. Defining a standard for particle swarm optimization[C]//Proc of IEEE Swarm Intelligence Symposium. Honolulu,2007.
- [11] Kennedy J. The Particle swarm: social adaptation of knowledge [C]//IEEE International Conference on Evolutionary Computation:Indianapolis,1997:303-308.
- [12] Shi Y, Eberhart R. A modified particle warm optimizer[C]//IEEE World Congress on Computational Intelligence. Anchorage, USA,1998:69-73.
- [13] 刘建华. 粒子群算法的基本理论及其改进研究[D]. 湖南:中南大学信息科学与工程学院,2009.
- [14] Ali S, Siegel H J, Maheswaran M, et al. Representing task and machine heterogeneities for heterogeneous computing systems[J]. Journal of Science and Engineering,2000,3(3):195-207.
- [15] 沈恺涛. 基于云计算和改进粒子群算法的任务调度研究[J]. 计算机测量与控制,2012,20(11):3 070-3 072.
- [16] 李斌,李文峰. 基于仿真的优化的粒子群算法参数选取研究[J]. 计算机工程与应用,2011,47(33):30-35.

[责任编辑:严海琳]