

基于遗传加差分算法的云计算任务调度

葛君伟^{1,3}, 孙方方¹, 方义秋²

(1 重庆邮电大学 软件工程学院, 重庆 400065; 2 重庆邮电大学 计算机学院, 重庆 400065; 3 重庆邮电大学 图书馆, 重庆 400065)

摘 要: 随着云计算的不断发展, 任务调度问题成了研究的难点. 如何快速处理用户的任务请求, 使得云计算中资源达到负载均衡, 使任务的完成时间及成本达到相对最优, 通过对比分析了已有任务调度算法存在的问题提出遗传加差分算法的云计算任务调度策略, 采用种群更新的方式提高算法的鲁棒性、二级变异策略提高种群多样性, 加快算法的收敛, 加入差分算子提高算法的局部寻优能力. 通过 cloudsim 仿真实验并对比遗传算法、差分算法、min-min 算法证明了本算法的在负载均衡度, 时间和成本上有效性.

关键词: 云计算; 任务调度; 遗传算法; 差分算法

中图分类号: TP393

文献标识码: A

文章编号: 1000—7180(2016)11—0005—03

Scheduling Based on Improved Genetic Algorithm and Difference Algorithm in Cloud Computing

GE Jun-wei^{1,3}, SUN Fang-fang¹, FANG Yi-qiu²

(1 College of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2 College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 3 Library, Chongqing of University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: With the continuous development of cloud computing, task scheduling problem become a crucial aspect. How to deal with tasks quickly, not only meet the needs of users, but also to achieve load balancing and make the completion time, cost to achieve relatively optimal. By comparing and analyzing the existing task scheduling algorithms, we proposed the genetic and difference algorithm policy to solve the problem. we employs population updating scheme to improve the robustness of the algorithm. Using secondary mutation strategy to improve population diversity, accelerate the convergence speed. Adding difference operator to improve the algorithm's local search ability. The performance is analyzed using Cloudsim simulator and compared with existing GA, Min-min algorithm. Simulation results demonstrate that the proposed algorithm has better performance in load balancing, finish time and costs.

Key words: cloud computing; task scheduling; genetic algorithm; difference algorithm

1 引言

云计算是伴随着计算机技术、通信技术的飞速发展而出现的一种全新的商业计算模式. 云计算已经深入到社会生活中, 与传统行业相结合出现了金

融云, 交通云, 医疗云, 教育云等. 云计算系统的服务器规模庞大, 资源异构多样, 用户群体广泛, 应用任务类型各异, 云计算系统时刻都要处理大量的用户任务和海量数据. 因此, 如何对云系统中的资源进行合理分配管理^[1-2], 对用户任务进行高效调度执行,

这种基于服务的商业计算模式下,对其任务调度策略进行深入研究,不仅有很高的理论价值,而且具有较好的现实意义。

关于云环境的任务调度问题,已有大量研究人员对其进行深入研究,已经有一些调度算法如文献[3]提出一种同时兼顾任务完成时间最短和平均完成时间最短的两个适应度函数的遗传算法任务调度策略,在任务完成时间上达到了有效的均衡。文献[4]作者在初始化时对染色体进行相似度计算有效避免染色体在解空间分布不均,算法陷入早熟,同时在适应度函数中加入服务质量改进,使任务调度问题得到了有效改善。文献[5]提出了综合考时间、成本作为适应度函数来约束差分算法的选择机制。在交叉后对染色体分群变异操作,有效保留了交叉后优良个体,取得了良好的调度效果。文献[6]同样采用了贪婪算法处理无效染色体,在总适应度函数里加入物理机数量作为参数的遗传加贪婪算法,并分别从物理机使用数量、负载均衡度和迁移次数 3 个方面进行加权组合。但负载均衡度和迁移次数的适应度函数在不同染色体间的差异度不足,限制了算法的收敛速度;同时只通过物理机数量来衡量内存使用量是不完善的。

2 云计算任务调度模型

云环境中的任务调度一般分两层:第一层是用户任务到虚拟机的调度,将云任务分配给合适的虚拟机,在虚拟机执行范围内,分配可用的处理能力给任务;第二层调度在虚拟机到主机层,选择合适的主机来执行虚拟机中的任务^[7]。本文研究任务到虚拟机的调度。调度主要考虑任务的多维资源与虚拟机上的多维资源的映射,使任务分配合理,既不使虚拟机闲置,又可以是任务完成时间最短。本文任务调度模型如下:

用户任务调度元组可以表示为任务的多维资源请求列表 $\text{Task}\{\text{ID}, \text{CPU}, \text{mem}, \text{length}, \text{Tdata}\}$

其中第一个参数是任务编号,云计算任务调度就是把用户的任务分成多个子任务,然后让子任务在各个虚拟机上并行运行。其后两个参数是用户任务需要的处理器和内存,最后两个参数是用户的任务长度,和传输数据量。虚拟对应的资源列表 $Vm\{\text{ID}, \text{CPU}, \text{Mem}, W(i), \text{stor}, bw\}$ 分别表示虚拟机的编号,处理器,内存,价格,存储,提供的带宽。任务执行时间矩阵 $E_n(i, j)$ 表示任务 i 在虚拟机 j 上的执行时间,任务传输时间矩阵 $E_s(i, j)$ 表示任务 i

到达虚拟机 j 的时间。

通过执行时间和传输时间矩阵可求出任务完成时间

$$\text{runTime}(i, j) = E_n(i, j) + E_s(i, j) = \frac{\text{length}_i}{v_{\text{comp}}} + \frac{\text{Tdata}_i}{bw} \quad (1)$$

$$v_{\text{comp}} = \text{Num}(\text{cpu}) * \text{Mips} \quad (2)$$

式中式(2)表示虚拟机节点计算能力等于cpu的处理能力与处理机个数的乘积。 bw 表示云计算任务调度环境中的带宽。

其中虚拟机 j 的完成时间是运行在虚拟机 j 上所有任务的完成时间之和。由于各个虚拟机并行执行,所以任务的完成时间是最后一个完成任务的虚拟机的执行时间,假设初始时虚拟机的运行时间是同时开始的。

$$Vm(j) = \sum_{i=1}^k \text{runTime}(i, j) \quad (3)$$

$$\text{completeTime}(I) = \max_{i=1}^k (Vm(i)) \quad (4)$$

任务总成本代价

$$\text{TotalCost}(j) = \sum_{j=1}^m \sum_{i=1}^n \text{runTime}(i, j) \times w(j) \quad (5)$$

式中 $W(j)$ 节点 j 资源单位使用成本。

任务负载均衡度计算公式如下:

$$\begin{cases} \text{LoadB} = \sqrt{\sum_{i=1}^k \frac{(B - B_i)^2}{k}} \\ B_i = \frac{Vm(i)}{\text{RunTime}(I)}, \quad \bar{B} = \frac{\sum_{i=1}^k B_i}{k} \end{cases} \quad (6)$$

负载均衡率是考虑在整个算法运行时间里虚拟机运行的时间效率。

3 云计算任务调度算法

遗传算法是一种并行,高效,全局寻优的进化算法。通过适应度函数来控制进化方向,通过交叉与变异算子在定义域内寻找最优解。该算法是先对待求解问题进行编码,对生成的染色体进行选择、交叉、变异等操作产生出更适合环境的新的染色体。经过不停的迭代,种群规模变大,搜索到全局最优解的概率越大。但计算量和时间也会随着增加,传统的遗传算法易于陷入局部最优解和收敛速度较慢等问题^[8],难以满足云环境下对于调度资源的实时性及负载均衡大的要求。

差分进化算法(Differential Evolution, DE)是

一种简单而有效的全局优化算法,已被证明在求优过程中具有高效性、收敛性、鲁棒性等优点. DE 算法在近年来各种优化大赛比赛中取得优良成绩,尤其在解决高维问题求解上显示出卓越的特性. 差分进化的思想是利用变异策略得到新种群,通常是随机选取种群中的不同个体,用缩放因子将向量缩放后再合成待变异个体,由交叉操作得到一个由父子混合个体构成的种群,最后经选择算子根据优胜劣汰的竞争机制重新组成新一代种群. 经典差分进化算法的具体流程见文献[9]. 遗传算法的全局搜索能力较强,但局部搜索能力较差,而差分进化算法的局部寻优能力比较强,两种算法结合可以取长补短.

3.1 遗传加差分算法

本文所提出的遗传加差分算法流程如图 1 所示. 主要包括以下步骤:参数配置、种群初始化、适应度计算;判断是否满足结束条件、根据鲁棒阈值 ψ 判断是否需要更新种群;进行遗传算法中的选择操作等. 参数配置主要包括外界输入的任务数目及其大小、虚拟机数目 M 、平均负载 κ 、终止阈值、鲁棒阈值、种群大小、最大迭代次数、精英保留个数、交叉概率 P_c 、变异概率 P_v 和是否变异优化 θ 等.

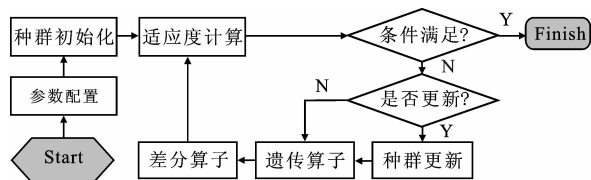


图 1 遗传差分算法总体流程

3.2 编码设计、种群初始化、适应度计算

任务调度问题可以用染色体编码表示,现有的编码方式主要有二进制编码,浮点数编码,自然数编码. 选用自然数编码,这样可以防止传统二进制编码搜索范围大,还可以减少个体解码时间.

染色体采用二级编码,基因的指标是任务的大小. 染色体片段为单个虚拟机上处理的集合,为若干个基因的集合. 染色体定义为若干染色体片段的集合. 染色体长度为执行的任务数. 每一条染色体包含 M 个与计算节点相对应的集合(染色体片段),每个集合包含若干个任务(基因). 如 10 个任务 $\{E_1, E_2, \dots, E_{10}\}$ 、3 个计算节点,一条染色体可以表示为 $\{\{E_2, E_4, E_3\}, \{E_1, E_5, E_9, E_7\}, \{E_8, E_6, E_{10}\}\}$. 解码即为虚拟机 1 上执行任务 2、3、4;虚拟机 2 上运行任务 1、5、7、9;虚拟机三上执行任务 6、8、10.

种群初始化采用随机生成的方式,即把所有任

务随机分配到各个计算节点.

适应度函数选取非常重要,直接影响到算法的收敛性和负载均衡. 本文综合考虑了任务的执行时间、成本,定义适应度函数如下:

$$f = \alpha \frac{1}{\text{completeTime}(I)} + \beta \frac{1}{\text{TotalCost}} \quad (7)$$

其中 $\alpha + \beta = 1$ 为约束函数, $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$. 即对于任何一个负载平衡分配方案,值越小表示方案越好.

3.3 遗传算子、差分算子

遗传算子包含选择、交叉、变异三种算子.

(1) 执行选择算子时,对最优的几条染色体采用精英保留策略,然后根据染色体的适应度采用轮盘赌形式进行选择,这样最优的染色体更容易被选择下来. 对给定规模为 N 种群 $(k_1, k_2, \dots, k_n)$, 染色体 k_i 的适应度为 $f(k_i)$,

$$P(k_i) = \frac{f(k_i)}{\sum_{j=1}^n f(k_j)} \quad (8)$$

(2) 交叉算子

交叉算子是产生新个体的主要方法,它决定了遗传算法的全局搜索能力.

交叉采用如下的二级部分删除交叉:

两条父代染色体的各染色体片段按顺序组合成集合(本质上为大的染色体片段) A, B ; 根据变异点,保留对应变异点间的基因,删除 A, B 中重复基因后并按顺序填入空白位置,生成新集合 C, D ; 根据染色体片段划分, C, D 复原为新的子代染色体.

(3) 变异算子设计

变异算子决定了遗传算法的局部搜索能力. 根据以上任务调度模型和编码设计,可以采用两个基因位置交换变异. 但是,变异有其自身特点. 首先,如果两个基因在同一染色体片段内,基因位置变换没有任何意义,这是由问题模型本身性质决定的. 其次,本身基因片段中基因个数差异非常大时,是很难取得较好结果. 最后,对于遗传算法本身而言,交叉算子和变异算子都是产生新子代的方法,实际上没有区别. 当变异概率较大时,直接不加区别的变异会导致种群适应度变差. 所以,在 GAPDE 中提出如下的两级变异操作:

第一级变异——随机选择两个染色体片段,随机选择选定的染色体片段中任意位置的基因进行两两交换. 这样的变异既保证了有效性,也保证了全局一致性.

第二级变异——随机选择两个染色体片段进行合并,随机产生分割点,根据分割点生产两个新的染色体片段.这样的变异本质上是染色体片段长度的变异.

(1) 在进行第一、二级变异时,当 θ 为TRUE时,根据进行择优保留策略.当变异后得到的子代染色体的适应度如果比父代染色体优良的话,则保存子代染色体,否则不变异.

(2) 为提高种群的多样性,采用如下策略对 θ 赋值:当迭代次数为奇数时, θ 取TRUE;当迭代次数为偶数时, θ 取FALSE.两级变异加快了算法的收敛.

(4) 差分算子 执行差分算子时,差分算子的变异策略是在两个以上的染色体上进行变异,增加算法的局部寻有能力.差分进化多种变异策略采用如下方式进行:

$$X_{\text{offspring}} = X_{\text{best}} \oplus (X_{\text{parent1}} \oplus X_{\text{parent2}}) \quad (9)$$

为提高种群的多样性, X_{best} 为随机选择的三个染色体中适应度值最高的一条,而标准差分进化算法使用的是全局最优染色体. $\oplus$ 表示交叉操作.

3.4 鲁棒性设计与终止条件

遗传算法和差分算法具有全局寻优能力,但是也存在早熟现象,鲁棒性不行. GAPDE 算法采用如下策略提高鲁棒性.首先,保留全局最优染色体及最优适应度值.其次,记录最优适应度值重复次数,当最优适应度值重复次数大于一定阈值时,重新生成全部染色体.

迭代自然终止条件主要有两种方式:

第一种是迭代次数超过设定的最大迭代次数 200,迭代自然终止.

第二种是全局最优染色体的适应度值 F_{opt} 满足一定要求时终止.

$$\text{abs}(\kappa - 1/F_{\text{opt}})/\kappa < \varepsilon \quad (10)$$

4 实验与仿真

本实验采用 cloudsim 平台进行仿真,在实验中模拟 100~500 个任务调度,并对比 Min-min, GA 实验.初始条件:虚拟机 8 台,最大迭代次数为 200,交叉概率 0.05,变异概率 0.85,鲁棒性阈值 20,种群规模为 64.将本算法与 min,标准遗传算法做对比,在参数相同情况下,比较任务的执行时间,负载均衡,以及成本.实验结果如图 2、图 3 所示.

首先对三批任务调度,任务数分别为 100、300、500,看算法的成本消耗.从实验效果看,本文所提算

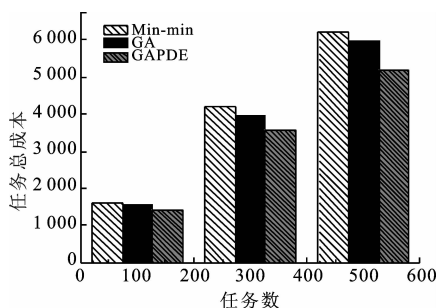


图2 任务成本

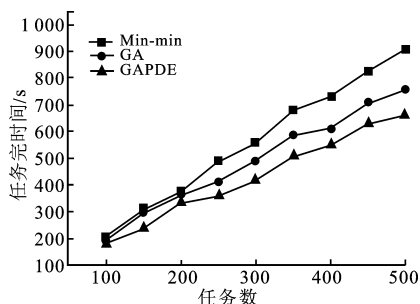


图3 任务完成时间

法在同等参数设置下,成本均低于其他两个算法.在 500 批任务时本文算法比 GA 算法节约成本约 14%,比 Min-min 算法节约成本约 18.7%.如图 2 所示.

对 100 到 500 个任务,每 50 个任务递增,可以看出本文算法在完成时间上均优于其他两种算法.任务完成时间如图 3 所示.

改进后的算法,采用两级变异,收敛速度要快于 GA,GA 收敛较慢,本文算法在迭代四十次后算快速收敛,随着迭代次数的增加,趋于平稳,负载度趋于 1%.如图 4 所示.

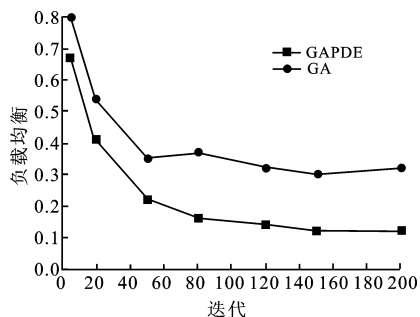


图4 负载均衡

5 结束语

本文探讨了云计算任务调度的研究进展及特点,对遗传算法的原理及改进策略进行了研究,提出

了遗传加差分算法的云计算任务调度策略,改进的遗传算法对进化具有导向作用,明显减少搜索的盲目性。也可避免遗传算法的欺骗性问题,陷入局部最优解。通过实验仿真算法在成本,任务完成时间,负载均衡度上效果均好于遗传算法,Min-min,证明了本算法的有效性。实际中云计算的调度环境更为复杂,下一步会考虑任务间依赖关系,以及大任务量的调度,以及用户满意度等因素。

参考文献:

- [1] Mezma M, Melab N, Kessaci Y, et al. A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems[J]. Journal of Parallel & Distributed Computing, 2011, 71 (11): 1497-1508.
- [2] 刘鹏. 云计算[M]. 北京: 电子工业出版社, 2011.
- [3] 李建锋,彭舰. 云计算环境下基于改进遗传算法的任务调度算法[J]. 计算机应用, 2011, 31(1):184-186.
- [4] 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(Sup1):1-4.
- [5] 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, 54 (1): 101-106.

- [6] Chen S, Wu J, Lu Z. A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness[C]// Computer and Information Technology, International Conference on. Chittagong, Bangladesh, IEEE, 2012:177-184.
- [7] Calheiros R N, Ranjan R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software-Practice and Experience, 2011, 41(1):23-50.
- [8] 李亚洲,郑晓军,张强等. 基于改进初始解的遗传算法的布局设计方法[J]. 计算机工程与应用, 2013, 49(8): 245-248.
- [9] Epitropakis M G, Tasoulis D K, Pavlidis N G, et al. Enhancing differential evolution utilizing proximity-based mutation operators[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1):67-98.

作者简介:

葛君伟 男, (1961-), 博士, 教授. 研究方向为云计算软件工程.

孙方方(通讯作者) 女, (1987-), 硕士. 研究方向为云计算软件工程. E-mail: sunfangfang12@126. com.

方义秋 女, (1963-), 博士, 副教授. 研究方向为云计算数据库.

(上接第 4 页)

- [M]. 北京:清华大学出版社, 2011:63-96.
- [2] Finisar. The path from 3Gb/s to SATA 6Gb/s; how to migrate current designs to the SATA revision 3. 0 specification[EB/OL]. [2006-02-15]. <https://sata-io.org/sites/default/files/documents/SATA-6-Gbs-The-Path-from-3gbs-to-6gbs.pdf>.
- [3] 刘芬. 刀片服务器的发展[J]. 科技视界, 2014, 3 (24):97.
- [4] Daniel P Bovet, Marco Cesati. Understanding the Linux Kernel[M]. Cambridge: O'Reilly, 2000.
- [5] Seagate Technology LLC. Barracuda 7200. 11 serial ATA product manual revision B [M]. USA: Seagate, 2008.
- [6] 欧阳科文. 基于 FPGA 的固态硬盘控制器设计[D]. 湖南: 湖南大学, 2009.
- [7] Wu wei, Su haibing. A high performance serial ATA

host controller[C]//Proc. of 2008 IEEE I on Computer Science and Software Engineering. [S. 1.]: IEEE Press, 2008: 18-23.

- [8] Louis Woods, Ken Oguro. Groundhog a serial ATA host bus adapter (HBA) for FPGAs [EB/OL]. [2016-02-19]. [http://people.inf.ethz.ch/lwoods/publications/Groundhog-A_Serial_ATA_Host_Bus_Adapter_\(HBA\)_for_FPGAs.pdf](http://people.inf.ethz.ch/lwoods/publications/Groundhog-A_Serial_ATA_Host_Bus_Adapter_(HBA)_for_FPGAs.pdf).

作者简介:

王振升 男, (1989-), 硕士研究生. 研究方向为嵌入式系统及分布式存储. E-mail: tsuzs@163. com.

张刚 男, (1953-), 教授, 博士生导师. 研究方向为基于嵌入式系统的多媒体开发应用.