

Multi-Task Scheduling Algorithm Based on Genetic Algorithm in Cloud Computing Environment

Zheng Sun

College of Information Science and Engineering, Shandong University of Science & Technology, Qingdao Shandong
Email: sunzheng_good@126.com

Received: May 18th, 2016; accepted: Jun. 5th, 2016; published: Jun. 8th, 2016

Copyright © 2016 by author and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Task scheduling is a key problem in cloud environments and genetic algorithm is a good method to find a solution for this problem. For the phenomenon of genetic algorithm in task scheduling process the complexity of the task scheduling problem increased and algorithm performance declined. In this paper, a genetic algorithm based on K-means cluster method with time and load balancing constraint is proposed. This algorithm uses K-means cluster method to classify the tasks at the beginning of scheduling and uses genetic algorithm to scheduling tasks of each class. Moreover, we proposed a time-load balancing constraints fitness function and optimized the mutation operator. Experiment results show that the proposed algorithm gives a better solution.

Keywords

Task Scheduling, Genetic Algorithm, K-Means Cluster, Cloud Computing

云计算环境下基于遗传算法的优化的多任务调度算法

孙 政

山东科技大学, 信息科学与工程学院, 山东 青岛
Email: sunzheng_good@126.com

摘要

任务调度是云计算中的一个关键问题，遗传算法是一种能较好解决优化问题的算法。本论文针对遗传算法在任务调度过程中随着任务调度问题复杂度增加，算法的性能出现下降的现象，引入K-means聚类算法，提出一种基于K-means聚类和遗传算法的云计算环境下任务调度的新算法。该算法借鉴 K-means 聚类方法的思想在任务调度前对任务进行聚类预处理，然后根据遗传算法的机制进行任务调度，并提出了时间 - 负载均衡约束的适应度函数，优化了变异算子。仿真实验结果表明，该算法在云环境下任务调度中具有较高的效率和性能。

关键词

任务调度，遗传算法，K-means聚类，云计算

1. 引言

云计算[1]已经成为互联网未来的一个重要的发展趋势。云计算是一种资源交付和使用模式，指通过网络获得应用所需的资源。而这些提供资源的网络称之为“云”。从用户角度来看“云”中的资源是可以无限扩展的，并且可以随时获取。

云计算所提供的服务面向的用户群体是非常庞大的，因此云系统中的任务数量是巨大的，它每时每刻都要处理海量的任务，所以如何高效地进行任务调度是云系统的一个关键问题。目前，大部分云计算平台都采用 Google 提出的 Map Reduce 编程模型，用于大规模数据集的并行计算。通过 Map 和 Reduce 两个阶段，将较大的任务分割成为多个较小的子任务，然后分配给多个计算资源并行执行，并得到最终的运行结果。在 Map/Reduce 编程模型下，如何对数量众多的子任务进行调度是个复杂的问题。因此，对云环境下的任务调度算法[2]的研究具有重要的意义。云环境下的任务调度的目标就是按照某种调度策略，对海量任务分配处理单元，以达到任务完成时间最短，成本最低，同时高效地利用系统资源。启发式(heuristic)任务调度算法是解决云环境下任务调度的最佳途径，其中遗传算法具有较强的全局搜索能力，特别在任务调度问题的复杂度增加时[3]。

遗传算法(GA)是一类借鉴生物界中自然选择和遗传机制的随机化自适应搜索算法，最先由美国 Michigan 大学的 Holland J. H.教授提出[4]。GA 主要由选择、交叉和变异算子组成。GA 的目标是要在一定次数的迭代中进化出一个可接受的解。针对遗传算法的任务调度已经提出了很多方法，然而当任务调度问题的复杂度增加时，算法效率降低。并且遗传算法的群体中某个体适应度大大高于其他个体而迅速繁殖产生的拥挤现象导致算法的效率降低、过早收敛，对此提出了基于 k-means 的时间 - 负载均衡约束的遗传算法(KTLBCGA)。本文主要分为 5 个部分，引言，相关技术介绍，算法介绍，仿真实验，总结。

2. 相关知识与技术

k-means 算法是聚类分析方法中一种基本的且应用最广泛的划分算法。它是将各个聚类子集内的所有数据样本的均值作为该聚类的代表点，算法的主要思想是通过迭代过程把数据集划分为不同的类别，使得评价聚类性能的准则函数达到最优(平均误差准则函数 E)，从而使生成的每个聚类(又称簇)内紧凑，类间独立，以达到较优的聚类效果。K-means 算法的步骤如下所示。

- Step 1: 从数据中随机抽取 K 个点作为初始聚类的中心。
- Step 2: 计算数据中所有的点到这个中心的距离, 将点归到离其最近的聚类里。
- Step 3: 重新计算聚类中心。
- Step 4: 重复 Step 2, 直到误差小于阈值或者聚类中心不再改变。

3. 基于 K-means 的遗传算法的任务调度算法

算法基本思想: 首先对问题进行编码, 然后借鉴 K-means 算法思想在任务调度前对任务进行聚类预处理, 使得聚类后的每个分组内的任务具有较好的相似性, 其次对每一个任务分组进行任务调度, 其中遗传算法采用时间 - 负载均衡约束的适应度函数, 避免了早熟收敛, 从而提高了算法在任务调度中的效率。

3.1. 染色体的编码与解码

遗传算法采用固定长度的二进制符号串来表示群体中的个体。染色体的编码有很多种方式, 可以采用直接编码方式, 也可以采用间接编码方式。本文采用资源-任务的间接编码方式[5]。在这种编码方式中, 染色体的长度为子任务的数量, 染色体中每个基因的取值为该位置对应的子任务占用的资源编号。假设有 10 个任务, 3 个可用资源, 则染色体的长度为 10, 每个基因的值则为随机产生的 3 个资源的编号, 如随机产生如下的染色体编码:

$$\{2, 3, 2, 2, 1, 1, 1, 3, 2, 3\}$$

这个染色体就代表第一个任务在第二个资源上运行, 第二个任务在第三个资源上执行, 以此类推, 第十个任务在第二个资源上执行。

接下来就要进行染色体的解码, 通过解码得到不同资源上任务的分布情况。将任务按照占用的资源进行分类, 得到多组按资源编号分类的任务序列, 每个序列的编号就是某一个资源的编号, 序列中的元素就是在该资源上执行的任务编号。将上述的染色体进行解码得到 3 个资源的任务序列:

$$R1: \{5, 6, 7\} \quad R2: \{1, 3, 4, 10\} \quad R3: \{2, 8, 9, 11\}$$

根据解码得到的各个资源上的任务序列和ETC矩阵就可以计算出每个资源执行该资源上的所有子任务所需要的时间, 而云计算环境下的任务调度具有并发性, 取各个资源上执行任务的最长完成时间, 则完成所有子任务的总时间为:

$$\text{SumTime}(j) = \text{Max} \sum_{i=1}^n \text{singleTime}(i, j) \quad (1)$$

其中, n 表示分配到该资源上的子任务数量, $\text{singleTime}(i, j)$ 表示第 i 个资源上执行第 j 个子任务执行完成所用的时间。

3.2. 初始种群的生成

若种群规模为 S , 子任务总个数为 M , 资源数为 R , 则初始化描述为由系统随机产生 S 条染色体, 染色体长度为 M , 基因的取值为 $[1, R]$ 之间的随机数。

本文借鉴 K-means 算法思想对初始种群进行聚类划分,

- (1) 在 n 个任务对象中随机确定 K 个对象作为聚类中心;
- (2) 将距离初始中心最近的任务分配到该聚类中;
- (3) 计算每个聚类的中所有任务长度的均值作为聚类中心;
- (4) 重复执行(2)~(3), 直到聚类中任务不再发生改变;

(5) 对每一组任务采用遗传算法进行资源的分配。根据任务的长度进行分组后,各分组内任务之间的具有良好的相似性。

3.3. 适应度函数

遗传算法按与个体适应度成正比的概率来决定当前群体中各个个体遗传到下一代群体中的机会多少[6]。因此,适应度函数的选取是相当重要的,直接影响到遗传算法的收敛速度以及能否找到最优解[7]在任务调度中的一个重要目标就是资源上分配的所有子任务完成的总时间。因此定义适应度函数为:

$$\text{FitnessTime}(j) = 1/\text{SumTime}(j) \quad 1 \leq j \leq S \quad (2)$$

在任务调度过程中同时要考虑资源负载均衡问题,负载均衡使得资源利用率较高。本文采用染色体上各个资源上分配的任务数量的标准差来衡量负载均衡问题。染色体的各个资源上的任务分配数量的标准差适应度函数为:

$$S(j) = \sqrt{\frac{\sum_{i=1}^n (\text{subTask}_{j,i} - \text{avgTask})^2}{n}} \quad (3)$$

$$\text{FitnessBalance}(j) = 1/S(j) \quad 1 \leq j \leq S \quad (4)$$

其中, $\text{subTask}_{j,i}$ 为第 j 个染色体上第 i 个资源上所分配的任务数量。

$$\text{Fitness}(j) = \alpha \times \text{FitnessTime}(j) + \beta \times \text{FitnessBalance}(j) \quad (5)$$

其中, $\alpha \in [0,1]$, $\beta \in [0,1]$, $\alpha + \beta = 1$ 。适应度函数既考虑了任务调度的时间,又考虑了资源的负载均衡。

3.4. 遗传操作

3.4.1. 个体选择操作

选择操作的目的是为了从当前群体中选出优良的个体。根据各个个体的适应度值,按照一定的规则或方法从上一代群体中选择出一些优良的个体遗传到下一代群体中。KTLBCGA 采用轮盘赌选择作为选择操作算子,通过适应度计算公式计算出个体被选择的概率。根据个体适应度计算公式(2),则个体 i 被选取的概率为:

$$P(i) = \frac{\text{Fitness}(i)}{\sum_{j=1}^S \text{Fitness}(j)} \quad (6)$$

3.4.2. 交叉与变异操作

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。将原有的优良基因遗传给下一代个体,并生成包含更优良基因结构的新个体。交叉算子从全局出发找到了一些比较优秀的个体,此时已经接近最优解,但是交叉算子无法对搜索空间的细节进行局部搜索,这时采用变异算子可以拓展新的搜索空间,在种群局部收敛时,通过变异可以保持种群多样性,防止出现早熟现象。变异算子使得算法在接近最优解邻域时能加速向全局最优解收敛。

4. 仿真实验

本文采用 Cloudsim 模拟云计算环境,在相同的条件下,将算法 TGA [8]与 KTLBCGA 进行对比实验。算法的初始条件:种群规模 $S = 100$,资源数量 $R = 5$,子任务总个数 $M = 100$ 。虚拟机取[500,1000]之间的随机数,任务长度取[10000,11000]之间的随机数。算法的终止条件设为到达一定的进化代数,进化结

束后，从最后一代个体中选择适应度值最大的染色体，将染色体解码后得到任务调度的最优调度方案。KTLBCGA 算法的初始聚类中心用 K 表示， K 取 2。根据大量的实验数据显示， $K=2$ 时，效果最优秀。

图 1 表示了 TGA 和 KTLBCGA 两种算法的任务完成时间随进化代数的变化曲线。从图中可以看出，在遗传算法的进化初期，KTLBCGA 算法优于 TGA 算法，而在进化的中期 TGA 与 KTLBCGA 相接近，随着进化代数的增加 KTLBCGA 又明显优于 TGA，从整个进化的过程来看，KTLBCGA 整体优于 TGA。

图 2 显示了不同数量的任务的完成时间。通过图 2 可以看出，KTLBCGA 算法相对与 TGA 算法在任务完成的总时间上具有优势，并且随着任务数量的增加，KTLBCGA 算法在任务完成总时间上的优势越来越显著。而云计算环境下的任务调度通常需要对大量的任务进行处理，所以 KTLBCGA 算法相比 TGA 算法更具有优势，在云计算环境下采用该算法更能满足用户的需求。

5. 总结

本文提出了一种基于时间和负载均衡的适应度函数的遗传算法，并用 k-means 算法对初始种群进行聚类，然后对每一类任务进行调度，还对变异算子进行了优化。仿真实验结果表明，该算法具有较好的

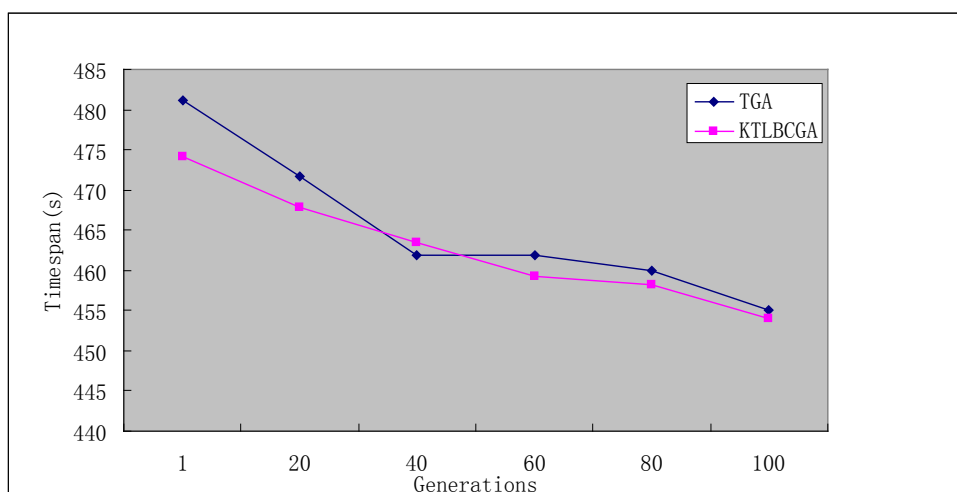


Figure 1. Generations vs. timespan of tasks

图 1. 进化代数与任务完成时间关系

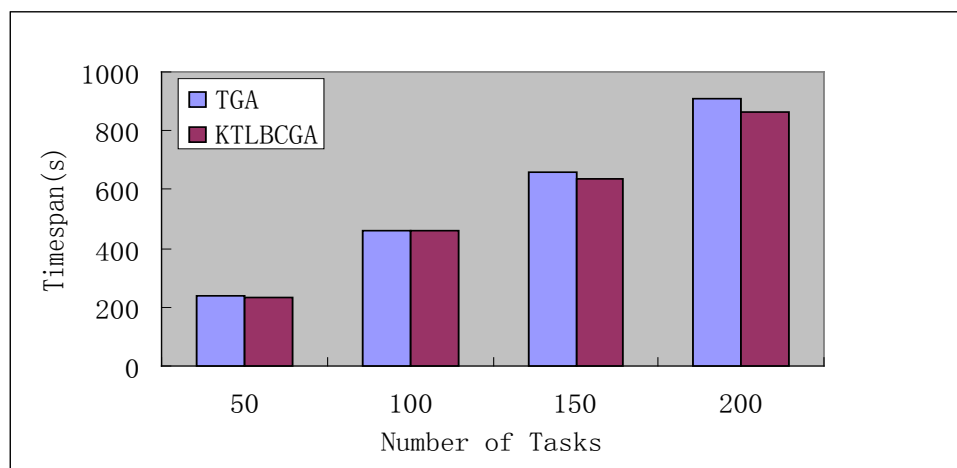


Figure 2. Number of tasks vs. timespan of tasks

图 2. 任务数量与任务完成时间关系

最优解搜索能力和较高的收敛性，在云计算环境下的任务调度中有比较理想的效果。

在未来的工作中，将考虑任务调度中的经济因素，进一步研究云环境下任务调度中的可靠性和安全性。

参考文献 (References)

- [1] Botta, A., de Donato, W., Persico, V., *et al.* (2016) Integration of Cloud Computing and Internet of Things: A Survey. *Future Generation Computer Systems*, **56**, 684-700. <http://dx.doi.org/10.1016/j.future.2015.09.021>
- [2] Kampas, S.R., Tarkowski, A.R., Portell, C.M., *et al.* (2016) System and Method for Cloud Enterprise Services. US Patent 9,235,442.
- [3] Haque, M.N., Noman, N., Berretta, R., *et al.* (2016) Heterogeneous Ensemble Combination Search Using Genetic Algorithm for Class Imbalanced Data Classification. *PLoS One*, **11**, 1-28. <http://dx.doi.org/10.1371/journal.pone.0146116>
- [4] Bahrami, M. and Singhal, M. (2015) The Role of Cloud Computing Architecture in Big Data. Information Granularity, Big Data, and Computational Intelligence. Springer International Publishing, Berlin Heidelberg, 275-295.
- [5] 朱宗斌, 杜中军. 基于改进 GA 的云计算任务调度算法[J]. 计算机工程与应用, 2013, 49(5): 77-80.
- [6] Demir, Y. and İşleyen, S.K. (2014) An Effective Genetic Algorithm for Flexible Job-Shop Scheduling with Overlapping in Operations. *International Journal of Production Research*, 1-17. (Ahead-of-Print).
- [7] Xu, Y., Li, K., Hu, J., *et al.* (2014) A Genetic Algorithm for Task Scheduling on Heterogeneous Computing Systems Using Multiple Priority Queues. *Information Sciences*, **270**, 255-287. <http://dx.doi.org/10.1016/j.ins.2014.02.122>
- [8] Pop, F., Cristea, V., Bessis, N., *et al.* (2013) Reputation Guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments. 2013 *27th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, IEEE, 772-776.