

◎网络、通信、安全◎

基于改进GA的云计算任务调度算法

朱宗斌, 杜中军

ZHU Zongbin, DU Zhongjun

四川大学 计算机学院, 成都 610065

College of Computer Science, Sichuan University, Chengdu 610065, China

ZHU Zongbin, DU Zhongjun. Improved GA-based task scheduling algorithm in cloud computing. *Computer Engineering and Applications*, 2013, 49(5): 77-80.

Abstract: Cloud computing needs to manage a large number of computing tasks, while task scheduling strategy plays a key role in determining the efficiency of cloud computing. It is an important issue how to allocate computing resources reasonably and schedule tasks run effectively which can reduce the complete time and cost of all tasks. A Time and Cost constraints Genetic Algorithm(TCGA) is proposed, through which, a better scheduling result not only shortens time, but also costs less. The simulation shows that TCGA is an efficient task scheduling algorithm in cloud computing by contrast with Time constraints Genetic Algorithm(TGA) and Cost constraints Genetic Algorithm(CGA).

Key words: cloud computing; Genetic Algorithm(GA); task scheduling; time; cost

摘 要: 云计算通常需要处理大量的计算任务, 任务调度策略在决定云计算效率方面起着关键作用。如何合理地分配计算资源, 有效地调度任务运行, 使所有任务运行完成所需的时间较短、成本较小是个重要的问题。提出一种考虑时间-成本约束的遗传算法(TCGA), 通过此算法调度产生的结果不仅能使任务完成所需的时间较短, 而且成本较小。通过实验, 将TCGA与考虑时间约束的遗传算法(TGA)、考虑成本约束的遗传算法(CGA)进行比较, 实验结果表明, 该算法是云计算中一种有效的任务调度算法。

关键词: 云计算; 遗传算法; 任务调度; 时间; 成本

文献标志码: A **中图分类号:** TP393 **doi:** 10.3778/j.issn.1002-8331.1108-0132

1 引言

云计算概念提出以来就成为一个热点研究方向。文献[1]综述了当前云计算所采用的技术, 剖析其背后的技术含义以及当前云计算参与企业所采用的云计算实现方案, 并对工业界3个具体的云计算实例, 具体包括Google的云计算平台以及云计算的网络应用程序、IBM公司的“蓝云”平台产品以及Amazon公司的弹性计算云进行了研究。文献[2]对云计算的概念进行了定义, 并对云计算发展所面临的问题与机遇作了分析与描述。

云计算提供了基础设施、平台和软件的服务, 其基本原理是通过网络将计算处理任务拆分成多个较小子任务, 再由多个服务器组成的庞大系统搜索、计算分析后将处理结果回传给用户^[3]。云计算可视为分布式计算、并行计算、网络计算的发展, 并在商业领域得以实现。文献[4]全面比

较了网格计算和云计算, 并指出云计算“按需使用, 按量付费”的商业模型。

由于云计算所面对的计算任务数量十分庞大, 任务调度和资源分配问题是决定云计算效率的重点与难点。目前, 对云计算中任务调度和资源分配的相关研究不多, 而网格计算已经对相关问题进行了广泛的研究^[5-8], 在一定程度上, 两者具有相似性。网格计算中任务调度算法的一个最主要目标就是使所有任务运行完成所需要的时间最小, 大多数的任务调度算法都是以这一目标来对任务调度进行优化的。然而, 在云计算模型中, 任务执行所需要的成本也是一个不可忽略的因素, 不同计算能力的资源, 其使用成本也不同。对于时间敏感的应用, 提供较强处理能力的资源, 使得任务运行完成所需的时间较短; 对于成本敏感的用户应用, 提供较低处理成本的资源, 使得任务运行

作者简介: 朱宗斌(1988—), 男, 硕士研究生, 研究方向: 计算机网络; 杜中军(1965—), 男, 副教授, 硕士生导师, 研究方向: 计算机网络。
E-mail: zzbincd2008@163.com

收稿日期: 2011-08-11 **修回日期:** 2011-10-18 **文章编号:** 1002-8331(2013)05-0077-04

CNKI出版日期: 2011-12-09 <http://www.cnki.net/kcms/detail/11.2127.TP.20111209.1002.052.html>

完成所需的成本较低。本文通过研究如何对云计算中的资源进行合理的分配,任务进行高效的调度问题,提出了一种考虑时间-成本约束的基于遗传算法的任务调度算法(Time and Cost constraints Genetic Algorithm, TCGA),并通过实验,验证了算法的有效性。

2 任务调度问题的定义

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。云计算同时需要向多个用户提供服务,要考虑到每个用户的响应时间,同时,也要考虑到服务所需的成本。已有的一些调度算法通常只考虑了其中一个因素的优化,容易造成任务完成时间较短,而成本较高的情况。因此本文提出 TCGA 对云计算中任务调度和资源分配策略进行改进,以最大限度地提高云计算效率。

云计算中的资源包括处理器、存储器、网络等,资源的使用是按需使用、按使用量付费的。本文将云计算中的资源统一视为计算资源,并作如下假设:

(1)任务的输入是多个较大的计算任务分解成的一批子任务,子任务划分的粒度均匀,即所需要运行的时间差异不大。

(2)子任务的数量远大于资源的数量。

(3)子任务在各个计算资源上运行所需的时间已知。

(4)各个计算资源节点上任务运行单元时间所需的成本已知。

用 N 表示子任务的数量, M 表示计算资源的数量,并用 $M \times N$ 的 ETC(Expect Time to Complete)矩阵^[9]($ETC(i, j)$ 表示第 j 个子任务在第 i 个计算资源上运行完成所需的时间)来计算各个计算资源上任务队列运行完成所需要的时间。用 $RCU(i)$ (Resource Cost per Unit)表示各个计算资源单元时间任务运行的成本。子任务用 T_j 表示,计算资源用 P_i 表示,那么子任务实例可以描述为 $(T_j, P_i, ETC(i, j), RCU(j))(i \in [1, M], j \in [1, N])$ 。

在以上假设条件下,可将云计算中任务调度问题定义为如何合理地将子任务分配到各个计算资源,使得任务运行完成所需的时间较短、成本较小。

3 TCGA 任务调度算法

遗传算法(Genetic Algorithm, GA)是由 Holland 教授受到生物模拟技术的启发,创造出的一种基于生物遗传和进化机制的适合于复杂系统优化的自适应概率优化技术。其本质是一种高效、并行、全局搜索的方法,它能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^[10]。

3.1 染色体的编码与解码

染色体的编码有很多种方式,可以采用直接编码(对任务的执行状态编码),也可以采用间接编码。本文采用文献[8]中提出资源-任务的间接编码方式,对子任务占用的资源编码。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,如图1所示。其中, T_i 表示任务编号, P_i 表示任务 T_i 执行时占用的资源编号。在产生初始种群时,每一个染色体中的资源编号 P_i 都是随机产生的,经过交叉、变异算子后,子任务 T_i 可能占用任何一个可用的资源,所以最优解一定对应某一个染色体编码。假设有 10 个子任务, 3 个可用资源, 则染色体长度为 10, 每个基因取值为 1~3 之间的随机数, 如随机产生如下的染色体编码:

{2, 1, 2, 1, 3, 1, 3, 3, 2, 3}

则这个染色体代表第 1 个子任务在第 2 个资源上运行, 第 2 个子任务在第 1 个资源上运行, 以此类推, 第 10 个子任务在第 3 个资源上运行。

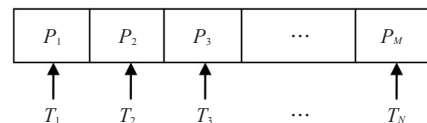


图1 资源-任务编码

产生染色体后,还必须对其进行解码,得到不同资源上子任务的分布情况。将子任务按照占用的资源分类,生成多组按照资源编号分类的子任务序列。将上述染色体解码为:

$P_1: \{T_2, T_4, T_6\}$

$P_2: \{T_1, T_3, T_9\}$

$P_3: \{T_5, T_7, T_8, T_{10}\}$

通过解码得到分配到各个计算资源上子任务的序列,利用 ETC 矩阵,可以计算得到各个计算资源完成子任务序列所需要的时间。由于云计算中各个计算资源间具有并发处理的特性,取上述计算结果的最大值,即为所有子任务运行完成的时间:

$$\text{sumTime}(i) = \sum_{j=1}^n \text{Time}(i, j), i \in [1, M] \quad (1)$$

$$\text{completeTime}(I) = \max(\text{sumTime}(i)) \quad (2)$$

其中, n 表示被分配到各个计算资源上的子任务数量, $\text{Time}(i, j)$ 表示被分配到计算资源 P_i 上的第 j 个子任务执行完成所需的时间。此时,利用上面求得的 $\text{sumTime}(i)$, 结合 $RCU(i)$ 可以求得完成所有子任务所需要的成本:

$$\text{completeCost}(I) = \sum_{i=1}^M \text{sumTime}(i) \times RCU(i) \quad (3)$$

本文提出的任务调度算法需要同时考虑所有子任务完成的时间与成本,在此利用贪心算法,求得所有子任务运行完成所需的最大时间与最大成本约束分别为:

$$\begin{aligned} \text{MaxTime} &= \text{completeTime}_{\text{MIN}} + \\ & t \times (\text{completeTime}_{\text{MAX}} - \text{completeTime}_{\text{MIN}}) \end{aligned} \quad (4)$$

$$MaxCost = completeCost_{MIN} + c \times (completeCost_{MAX} - completeCost_{MIN}) \quad (5)$$

其中, $completeTime_{MIN}$ 与 $completeCost_{MIN}$ 是根据子任务在各个计算资源上运行完成所需的最小时间与最小成本来进行贪心选择得到的结果; 而 $completeTime_{MAX}$ 与 $completeCost_{MAX}$ 则正好相反, 是根据子任务在各个计算资源上运行完成所需的最大时间与最大成本来进行贪心选择得到的结果。公式(4)中的 $t \in [0, 1]$ 为时间因子, 公式(5)中的 $c \in [0, 1]$ 为成本因子, 它们的值由用户根据需求指定, 且 t 和 c 成反比关系。对时间敏感的应用, t 应指定为较小的 $[0.2, 0.5]$ 之间的值, 而 c 应指定为较大的 $[0.5, 0.8]$ 之间的值; 对成本敏感的应用, c 应指定为较小的 $[0.2, 0.5]$ 之间的值, 而 t 应指定为较大的 $[0.5, 0.8]$ 之间的值。

3.2 初始种群的生成

取种群规模为 S , 资源数为 M , 子任务数为 N , 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 N , 基因值为 $[1, M]$ 之间的随机数。

3.3 适应度函数

遗传算法适应度函数的选取至关重要, 直接影响到遗传算法的收敛速度与最优解的查找。适应度较大的个体遗传到下一代的概率就越大; 而适应度较小的个体遗传到下一代的概率就相对小一些。任务调度需要考虑所有子任务完成所需要的时间和成本。

定义时间的适应度函数为:

$$F_{time}(I) = \frac{1}{completeTime(I)} \times u_{LB} \quad (6)$$

$$u_{LB} = \frac{\sum_{i=1}^M sumTime(i)}{M \times completeTime(I)} \quad (7)$$

公式(6)中的 u_{LB} 定义为平衡任务负载因子, 其值可由公式(7)求得, 它代表各个计算资源的利用率情况。 u_{LB} 的值越大, 表示计算资源的利用率越高, 那么 $completeTime(I)$ 的值就会相对越小。

定义成本的适应度函数为:

$$F_{cost}(I) = \frac{1}{completeCost(I)} \quad (8)$$

在只考虑时间约束的适应度函数中, 计算资源的利用率越高, 所有子任务运行完成所需时间越短的个体, 其适应度的值越大; 在只考虑成本约束的适应度函数中, 所有子任务运行完成所需成本越低的个体, 其适应度的值越大。那么, 可定义考虑时间-成本约束的适应度函数为:

$$Fitness(I) = \alpha \times F_{time}(I) + \beta \times F_{cost}(I) \quad (9)$$

其中, $\alpha \in [0, 1]$, $\beta \in [0, 1]$, $\alpha + \beta = 1$ 。取 $\alpha = 1$, $\beta = 0$ 时, 算法调度产生的结果为完成所有子任务的最短时间调度; 取 $\alpha = 0$, $\beta = 1$ 时, 算法调度产生的结果为完成所有子任务最小成本调度。

3.4 遗传操作

3.4.1 选择操作

选择操作在种群中选择适应度强的个体, 以产生新的

种群的过程。根据“优胜劣汰”的原理, 个体的适应度越高, 被选择遗传到下一代的概率就越大, 使得种群中个体的适应度值不断接近最优解。TCGA 采用轮盘赌选择作为选择操作算子, 通过适应度计算公式(9)计算出个体被选择的概率。

$$P(i) = \frac{Fitness(i)}{\sum_{j=1}^S Fitness(j)} \quad (10)$$

适应度函数考虑了任务调度的时间和成本约束, 通过上述选择操作, 使得种群中既有任务完成时间较短的个体, 又有任务完成成本较小的个体。

3.4.2 交叉与变异操作

交叉操作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; 而变异操作能改善遗传算法的局部搜索能力, 维持种群的多样性, 防止出现早熟现象。TCGA 使用文献[11]提出的自适应遗传算法(AGA), 并对交叉概率和变异概率计算公式进行改进, 自适应地调整交叉和变异操作的概率。

$$P_c = \begin{cases} P_{c1} - (P_{c1} - P_{c2}) \times (f_{max} - f') / (f_{max} - f_{avg}), & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \quad (11)$$

$$P_m = \begin{cases} P_{m1} - (P_{m1} - P_{m2}) \times (f_{max} - f) / (f_{max} - f_{avg}), & f \geq f_{avg} \\ P_{m1}, & f < f_{avg} \end{cases} \quad (12)$$

式中, f_{max} 表示群体中最大的适应度值; f_{avg} 表示每代群体的平均适应度值; f' 表示要交叉的两个个体中较大的适应度值; f 表示要变异个体的适应度值^[11]。

3.5 TCGA 重新调度

在每一代遗传操作结束以后, TCGA 利用算法初始时求得的 $MaxTime$ 和 $MaxCost$ 约束与最优子任务调度结果的染色体进行比较。若 TCGA 算法运行产生的最优子任务调度结果不满足 $completeTime(I)$ 小于等于 $MaxTime$, 或 $completeCost(I)$ 小于等于 $MaxCost$ 约束条件, 则需要重新运行 TCGA 算法产生新的最优子任务调度结果。

4 实验结果与分析

本文采用 CloudSim^[12] 平台来模拟云计算环境, 在相同的条件下, 用 TCGA 与 TGA(Time constraints Genetic Algorithm)、CGA(Cost constraints Genetic Algorithm) 进行对比实验。算法中各参数的取值如表 1 所示。

表 1 实验参数设置

主要参数	算法		
	TCGA	TGA	CGA
t	0.5	无	
c	0.5		
α	0.3	1	0
β	0.7	0	1
P_{c1}	0.9		
P_{c2}	0.6		
P_{m1}	0.1		
P_{m2}	0.01		

算法的初始条件: S 取值为 60, M 取值为 5, N 取值为 1 000, ETC 矩阵和 RCU 数组由系统随机生成。

算法的终止条件: 考虑到用遗传算法对任务进行调度自身运行的开销, 设置最大进化代数 ($MaxGn$) 为 100, 而不等待算法自身收敛, 算法终止。

实验结果如图 2、图 3 所示。

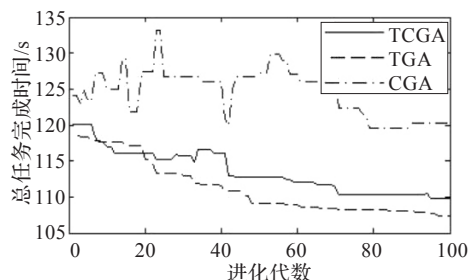


图2 总任务完成时间对比

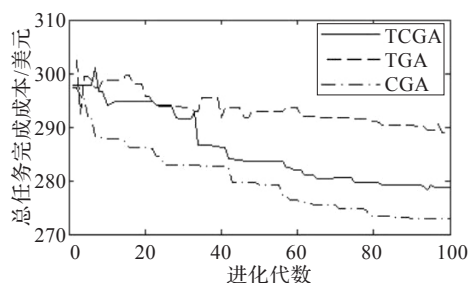


图3 总任务完成成本对比

从图 2 可以看出, 在遗传算法进化初期, TCGA 与 TGA、CGA 运行产生的最优子任务调度结果所需的总任务完成时间相差不大。随着进化代数的增长, TCGA 和 TGA 调度结果对总任务完成时间的优化越加明显, 而 CGA 调度结果对总任务完成时间无明显的优化作用。

从图 3 可以看出, 在遗传算法进化初期, TCGA 与 TGA、CGA 运行产生的最优子任务调度结果所需的总任务完成成本相差不大。随着进化代数的增长, TCGA 和 CGA 调度结果对总任务完成时间的优化越加明显, 而 TGA 调度结果对总任务完成时间无明显的优化作用。

综上所述, TGA 得到的子任务调度结果可以取得最短的总任务完成时间, 而对任务完成成本的优化作用不明显; CGA 得到的子任务调度结果可以取得最小的总任务完成成本, 而对任务完成时间的优化作用不明显; TCGA 同时考虑了时间-成本对算法的约束, 故得到的子任务调度的结果, 使得总任务完成时间较短、成本较小。

5 结束语

任务调度算法通常以总任务完成时间作为标准对任

务进行调度, 本文结合云计算的特点, 提出了一种考虑时间-成本约束的遗传算法的任务调度算法, 该算法把总任务完成时间和总任务完成成本同时作为标准, 对任务进行调度。实验结果表明, 通过本算法可以在云计算平台中实现较为合理的任务调度, 并产生理想的任务调度结果。

在未来的工作中, 将把重点放在云计算中动态任务调度的资源负载均衡上, 并考虑云计算中其他资源, 如网络服务质量、数据分布等因素对任务调度结果的影响。

参考文献:

- [1] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.
- [2] Armbrust M, Fox A, Griffith R, et al. Above the clouds: a Berkeley view of cloud computing[EB/OL]. [2009-02-10]. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>.
- [3] Wikipedia. 云计算[EB/OL]. [2010-06-26]. <http://zh.wikipedia.org/wiki/%E4%BA%91%E8%AE%1%E7%AE%97>.
- [4] Foster I, Zhao Y, Raicu I, et al. Cloud computing and grid computing 360-degree compared[C]//Proceedings of the 2008 Grid Computing Environments Workshop. Washington, DC: IEEE Computer Society, 2008: 1-10.
- [5] 罗红, 慕德俊, 邓智群, 等. 网格计算中任务调度研究综述[J]. 计算机应用研究, 2005, 22(5): 16-19.
- [6] Buyya R. Economic-based distributed resource management and scheduling for grid computing[D]. Australia: School of Computer Science and Software Engineering, Monash University, 2002.
- [7] Abraham A, Buyya R, Nath B. Nature's heuristics for scheduling jobs on computational grids[C]//The 8th International Conference on Advanced Computing and Communications. New Delhi: Tata McGraw-Hill Publishing, 2000: 45-52.
- [8] 林剑钊, 吴慧中. 基于遗传算法的网格资源调度算法[J]. 计算机研究与发展, 2004, 41(12): 2195-2199.
- [9] Braun T D, Siegel H J, Beck N, et al. 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(6): 810-837.
- [10] 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.
- [11] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Trans on SMC, 1994, 24(4): 656-667.
- [12] Calheiros R N, Ranjan R, de Rose C A F, et al. CloudSim: a novel framework for modeling and simulation of cloud computing infrastructures and services[EB/OL]. [2009-09-03]. <http://www.cloudbus.org/reports/CloudSim-ICPP2009.pdf>.