

基于改进遗传算法的云计算任务调度研究*

屈迟文**

(百色学院数学与计算机信息工程系, 百色 533000)

摘要:针对云计算服务集群任务调度算法的效果,提出一种基于改进遗传算法的任务调度算法。该算法采用检测种群个体多样性生成初始种群,遗留优秀个体,检测个体相似度增加交叉操作的有效性等策略实现任务调度的改进。实验表明,该算法在云计算任务调度中是一种有效的算法,缩短了任务的执行时间并节约系统资源。

关键词:改进遗传算法;任务调度;云计算;相似度;交叉

中图分类号:TP393 **文献标识码:**A **doi:**10.3969/j.issn.1006-6055.2013.05.013

Research on Cloud Computing Task Scheduling Based on Improved Genetic Algorithm*

QU Chiwen**

(Department of Mathematics & Computer Information Engineering, Baise University, Baise 533000)

Abstract: The task scheduling algorithm based on genetic algorithm is discussed against the effect of cloud computing services cluster task scheduling algorithm. Task scheduling is improved with effective strategies, generating the initial population by detection on diversity of individuals in a population, preserving excellent individuals, detecting single similarity, increasing crossing operation and so on. The experiment results show the effectiveness of this method, with the task execution time and conserves system resources reduced.

Key words: improvement genetic algorithm; task scheduling; cloud computing; similarity; cross

1 引言

云计算是一种由网格计算、分布式计算以及并行计算发展起来的新型计算模式。云计算环境下将各个服务器采用虚拟化技术虚拟成一个巨大的资源池,动态调用各虚拟机提供给用户使用。如何合理的将用户任务分配给不同的资源,使得整个系统的性能达到最佳效果,是任务调度算法需要解决的关键问题。遗传算法以其算法的全局最优性以及适用的广泛性等特征,被广泛应用到自然科学和工程科学领域中。目前许多学者提出一些方法改进遗传算法并应用到云计算任务调度策略中,文献[1]提出一种基于双适应度遗传算法的任务调度策略,该算法通过增加一个适应度选择种群使任务调度获得较好效果。文献[2]采用最佳个体保留策略到下一代的改进遗传算法进行任务调度,提高了算法收敛速度。但这些算法在进化中由于交叉选择操作的有效性不足造成算法种群的多样性差,容易陷入局部优化。为了充分利用云计算环境中资源并对用户提交的任务进行合理高效的调度,笔者提出了一种通过检测种群个体多样性生成初始种群,遗留优秀个体,检测个体相似度增加交叉操作的有效性等策略的改进遗传算法的任务调度算法,并通过 CloudSim 仿真实验,该算法在减少任务完成时间以及收敛速度方面有较大的改进。

2 背景描述

云计算相关技术研究中,任务调度是个研究热点问题,也是一个 NP 问题^[3,4]。影响云计算任务分配与调度最主要的因素有:任务计算所需时间,数据在各个资源间进行数据传输的延迟时间以及数据分配方式等^[5,6]。本文主要研究任

务调度算法方面的性能,所以重点考虑总体计算能力对任务执行时间的影响。以下描述云计算平台下任务调度模型:

在某个云计算平台下,由 m 个需要调度的任务、 n 个可用的任务执行资源,需要把 m 个任务以合理高效的方式调度到 n 个可执行资源上执行,目的是获得尽可能短的总完成时间,并且且肯能提高资源的利用率和整个系统的整体性能^[7,8]。

定义 1 任务预执行时间矩阵 etc 是一个 $m \times n$ 矩阵。其中任意一元素 $etc(i, j)$ 表示任务 T_i 在服务器 P_j 上执行时间。

定义 2 任务分配矩阵

定义 3 任务调度长度

$makespan = \min \left\{ \max_{j=1}^n \left(\sum_{i=1}^m ext(i, j) \right) \right\}$, 其中 $ext(i, j)$ 为

任务分配矩阵。

3 基于改进遗传算法的云计算资源调度

3.1 编码

编码是遗传算法的第一个关键步骤,编码方式直接影响到选择、交叉和变异等遗传操作,从某种程度上决定了该遗传算法的效率。笔者根据云计算任务调度的特征,采用文献[1]任务—资源相关联的编码方式,染色体的长度为子任务的数量。其具体编码方式如下:

1) 设用户提交的任务有 T 个,第 t 个任务有子任务 $sub_T(t)$,子任务总数量 $m = \sum_{t=1}^T sub_T(t)$,系统资源为 n 个,取值在 $[1, n]$ 之间的正整数构建染色体 $G \{g_1, g_2, g_3, \dots, g_m\}$,其中 m 为染色体的长度,大小等于子任务总数量。

2) 对于每个染色体 G 的每个基因 $G(i)$,下标 i 代表任务的编号, $G(i)$ 的值代表编号为 i 的子任务占用的资源编号。如:假设染色体的长度为 12,每个基因的取值为 $1 \sim 6$,产生的一条染色体为 $\{3, 2, 1, 4, 2, 6, 5, 6, 1, 2, 3, 5\}$,则这条染色体的含义表示为:第 1 个子任务在第 3 个资源机器上执

* 广西高校科研项目 (2013YB247), 百色学院一般科技项目 (2010KB16) 资助

** E-mail: 375200865@qq.com; Tel: 0776-2848110

行,第2个子任务在第2个资源机器上执行,⋯,第12个子任务在第5个资源机器上执行。

3.2 适应度函数

遗传算法通过适应度来决定种群个体生存能力,适应度函数的选择决定了种群的进化方式。适应度函数的选择的规则是适应度函数的优化方向要与适应度值增加的方向一致并且适应度值是非负的。对于云计算任务调度的问题,本文采用的个体适应度函数为:

$$fit(x) = \frac{1}{\max_{j=1}^n (\sum_{i=1}^m ext(i,j))} \quad (1)$$

这样个体执行时间越小,该染色体适应度的值就越大,那么该个体的性能就越好,被遗传到下一代的概率比较大。

3.3 初始种群的生成

遗传算法对初始种群非常敏感,对算法的收敛速度和是否是全局优化产生较大的影响。传统遗传算法采用随机方式产生初始种群,容易产生局部优化,本文为了加快算法的收敛速度以及产生全局的最优解,引入了个体间相似度保证初始种群均匀的分布在解空间上。

定义4 设种群规模为 N ,个体染色体的长度为 L ,资源机器个数为 n :

两个个体染色体海明码距离:

$$D(i,j) = \sum_{m=1}^L |G_{im} - G_{jm}| \quad (i,j) \in (1,2,\dots,n); \quad (2)$$

其中, $|G_{im} - G_{jm}| = \begin{cases} 1, G_{im} = G_{jm} \\ 0, G_{im} \neq G_{jm} \end{cases}$

$$\text{相似度为: } Sim(i,j) = \frac{D(i,j)}{L} \quad (3)$$

检测阈值: $\mu = \frac{L-C}{L}$, C 表示调节参数;

必须满足相似度 $Sim(i,j) > \mu$ 的两个个体染色体才能够进入初始种群。

通过以上方法产生的初始种群,可以保证种群中的各个个体之间具有较大的区别,当种群规模较大时,能够较大范围且均匀的分布在解空间上,从而较少了局部优化的几率,提高了全局搜索的能力。

3.4 选择操作

传统遗传选择通常采用轮盘赌算法进行个体选择,在交叉和变异过程中容易把一些局部最优个体被淘汰掉,降低了算法的收敛程度^[2]。本文采用一种最优保存策略实现选择操作。在进行选择操作时,计算出种群每个个体的适应度并按照从大到小的次序进行排序,父代种群中适应度最高的10%个体直接入选到子代种群中,剩下90%的个体按照轮盘赌算法进行个体选择。此算法能够保证最优局部个体不会在遗传操作的交叉和变异操作中被破坏,保证了子代的多样性,增强了算法的全局搜索能力。

3.5 交叉与变异

在标准遗传算法中,常采用的交叉操作有单点交叉、两点交叉等多种方式。在某些个体相同位数较多时,容易产生无效的交叉操作。如:两个染色体 $A(3,2,1,4,2,6,5,6,1,$

$2), B(3,2,1,2,3,5,5,6,1,2)$ 有3个基因位不同,分别是第4~6位,当随机交叉点产生的位置为7时,产生的子代与父代是相同,产生本次交叉操作无效。一般地,当种群进化到后期阶段时,种群中出现相同或相似度比较高个体,从而使大量的交叉操作的无效,导致算法的收敛速度缓慢,在解空间中找不到全局最优解^[5]。本文从交叉个体选择以及交叉点两个方面着手来提高交叉效率。

1) 交叉个体的选择:设种群中个体 m 与被选择交叉的其它个体池 $\{m_1, m_2, m_3, \dots, m_n\}$, 选择一个与 m 染色体进行交叉操作。算法要求被选择个体与 m 个体的不相关性较大的个体赋予较大的选择几率,以增加交叉操作有效性的几率。其计算方式如下:

$$p(m_i | m) = \begin{cases} \frac{1}{Lth} (1 + \gamma \frac{sim(m, m_i) - sim_{average}}{sim_{max} - sim_{min}}) & sim_{max} \neq sim_{min} \\ \frac{1}{Lth} & sim_{max} = sim_{min} \end{cases} \quad (4)$$

式中, γ 为调控参数, $\gamma \in (0,1)$, Lth 为染色体的长度, $sim(m, m_i)$ 为染色体 m 与 m_i 的相似度,其计算方式如式(2), $sim_{average}$ 为被选择个体池与 m 染色体的平均值, sim_{max} 为被选择个体池与 m 染色体的最大值, sim_{min} 为被选择个体池与 m 染色体的最小值。

2) 交叉点的选定:设两个染色体 $m_i \{g_{i1}, g_{i2}, g_{i3}, \dots, g_{iLth}\}$ 和染色体 $m_j \{g_{j1}, g_{j2}, g_{j3}, \dots, g_{jLth}\}$, 若选择的交叉点不恰当,则可能引起本次交叉操作的无效,因此交叉点的选择在交叉操作过程中显得至关重要。论文算法确定一个交叉点有效区域,然后在该有效区域内选择交叉点,从而保证了交叉操作的有效性。设交叉区域为 $(left, right)$, 其中, $left = \min\{k | g_{ik} \neq g_{jk}\}$, $k \in \{1, 2, 3, \dots, Lth\}$ $right = \max\{k | g_{ik} \neq g_{jk}\}$, $k \in \{1, 2, 3, \dots, Lth\}$ 。

算法对剩余90%的种群个体进行交叉操作,其交叉操作算法步骤如下:

Step1: 计算染色体 m 与被选个体池 m_i 的相似度,即 $sim(m, m_i)$;

Step2: 计算被选池中个体 m_i 可被选定的概率,即 $p(m_i | m)$;

Step3: 采用轮盘赌算法从被选池中选择一个染色体 m_i ;

Step4: 计算染色体 m 和 m_i 的交叉点选择区域,在该区域内随机生成一个插入点 $crosslocal$, 交换染色体基因段。

对于变异方式,染色体根据变异概率确定变异位置,明确染色体变异的基因位,对指定变异点的基因值为除该基因位原值以外的随机产生的 $[1, n]$ 之间的值取代。

4 算法仿真结果与分析

4.1 实验参数

为了验证本文所给算法在云计算任务调度中的有效性和优越性,实验在云计算仿真平台 CloudSim 环境^[9]中进行仿真。实验任务预期完成时间 ETC 随机生成取值范围为 $[1, 100]$ 之间 60×16 的测试数据,并且根据需要选定对应的行和列,选取文献1、文献2与本文算法进行比较,对每种情

况运行 20 次,取其中的平均值作为此情况的调度结果,交叉概率选取 0.9,变异概率选取 0.05,种群规模大小为 80,最大迭代次数为 100。

4.2 实验结果与性能分析

4.2.1 寻优性能比较

1) 资源和任务不变实验比较:实验以资源为 6 个,子任务数 60 个进行测试,从图 1 中看出,本文算法在种群初始化是引入个体相似度检测保证了种群的多样性,在进化的初始就比另外两种算法的任务完成时间要少;随着进化的深入,从交叉个体选择以及交叉点两个方面提高交叉效率,增加了种群的多样性,避免算法陷入局部最优解。

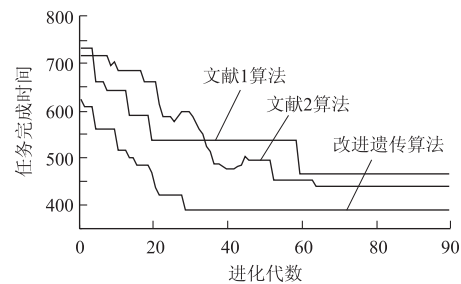


图 1 总任务完成时间
Figure 1 Total time to complete the task

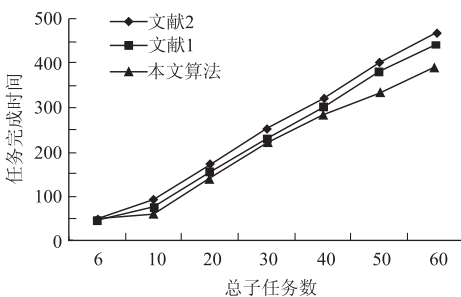


图 2 执行任务完成时间比较
Figure 2 The comparison of task completion time

2) 资源数不变任务数不断增加时实验比较:实验中资源为 6 个保持不变,子任务数从 6 个增加到 60 个,从图 2 中看出,在资源数不变,随着任务数的不断增加,本文算法执行任务所消耗的时间其它另外两种算法要少。任务数较少时,算法的效果并不明显,但任务数大于 40 时,比另外两种算法的执行任务完成要较少得更加剧烈。因此,在资源数不变,随着任务数的增加,该算法比其它两种算法在执行任务所消耗的时间差距不断的增大,该算法在任务调度中是有效的。

3) 任务数不变资源数不断增加时实验比较:实验子任务数为 60,资源数从 6 个不断增加到 16 个进行测试,对三种算法进行比较。从图 3 中表明,随着资源数的不断增加,算法比另外两种算法的任务完成时间要少,特别在资源数比较少时,优势更加明显。因此该调度算法在任务调度中也符合要求。

4.2.2 负载均衡的比较

实验选取资源数为 6 个不变,子任务数不断增加下在各个资源上的负载情况进行比较。如表 1 所示。表中描述 3 中方法,分别对应每个计算节点上任务的具体分布情况,第

一行表示不同的任务数,其中 C1 表示文献 1 算法,C2 表示文献 2 算法,O 表示文本改进遗传算法。为更加清楚表示算法的区别,负载均衡度采用式(3)相对标准差(RSD)的值进行比较^[10-12]。

$$RSD = \frac{\sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}}{\bar{x}} \tag{3}$$

其中,n 表示节点号,xi 为 i 节点的负载,xi 负载平均值。结果如图 4 所示。由图看出,改进的遗传算法的相对标准差要比另外两种要小,且随着任务数的增加偏差值越来越小。因此,该算法的负载均衡度较高,是一种有效的云计算任务调度算法。

表 1 不同任务下各计算节点任务分配具体情况
Table 1 Different tasks is assigned in each compute node under the specific circumstances

资源	10			20			30			40		
	C1	C2	O	C1	C2	O	C1	C2	O	C1	C2	O
1	30	90	39	143	153	145	202	198	190	267	198	245
2	26	43	25	125	87	140	132	150	146	234	230	190
3	65	30	25	55	45	126	214	147	215	145	278	260
4	77	56	50	83	146	137	225	251	145	300	269	288
5	40	65	48	155	170	81	187	230	130	258	253	221
6	68	25	60	98	109	74	145	217	187	270	320	280

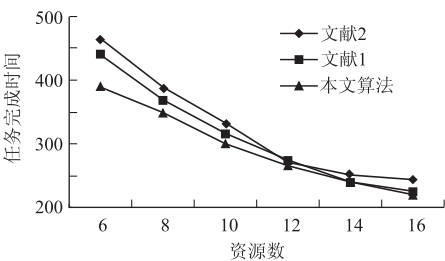


图 3 执行任务完成时间比较
Figure 3 The comparison of task completion time

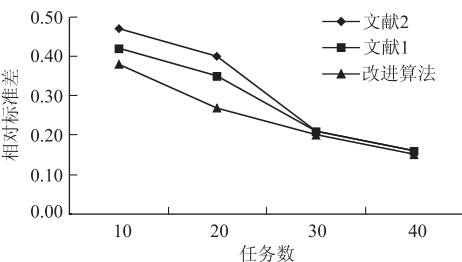


图 4 任务分配相对标准差
Figure 4 Relative standard deviation of task allocation

5 结束语

本文通过分析遗传算法的一些特性,提出了一种基于改进遗传算法的任务调度算法,该算法从种群个体多样性生成初始种群,在选择遗传操作中采用优秀个体遗留,检测个体相似度增加交叉操作的有效性等策略方面对算法进行改进。通过在云计算仿真平台 CloudSim 环境中进行仿真。实验表

明,该算法在云计算任务调度中是一种有效的算法,缩短了任务的执行时间并节约系统资源。

参考文献

[1]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.

[2]任萱萱. 基于 Hadoop 平台的作业调度研究[D]. 天津:天津师范大学,2011.

[3]FOSTER I,YONG ZHAO,RACIU I,et al. Cloud Computing and Grid Computing 360-Degree Compared[C]. Grid Computing Environments Workshop,2008:1-10.

[4]HEWITT C. ORGs for Scalable,Robust,Privacy-Friendly client cloud computing[J]. Internet Computing,2008,12(5):96-99.

[5]BELOGLAZOV A,ABAWAJY J,BUYYA R. Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing[J]. Future Generation Computer Systems,2012,28(5):755-768.

[6]刘万军,张孟华,郭文越. 基于 MPSO 算法的云计算资源调度策略

[J]. 计算机工程,2011,37(11):43-45.

[7]卢厚清,陈亮,宋以胜. 一种遗传算法交叉算子的改进算法[J]. 解放军理工大学学报:自然科学版,2006,8(3):250-251.

[8]ZAHARIA M,BORTHAKUR D,SARMA J S,et al. Job Scheduling for Multi-User MapReduce Clusters[R]. Berkeley:EECS Department University of California,2009.

[9]CALHEIROS R N,RANJAN R,BUYYA R,et al. CloudSim:a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience,41(1):23-50.

[10]程国建,刘丽景,石彩云,朱凯. 一种混合遗传算法在云计算负载均衡中的应用研究[J]. 西安石油大学学报(自然科学版),2012,27(2):93-97.

[11]肖斐. 虚拟化云计算中资源管理的研究与实现[D]. 西安:西安电子科技大学,2010.

[12]彭海云,李骞,李强. 网格环境下资源负载均衡和优化调度研究[J]. 计算机工程与应用,2009,45(19):104-106

作者简介

屈迟文(1979-),男,硕士,讲师,主要研究向:软件测试,云计算,智能计算。

(上接第 593 页)

5 结论

本文针对煤尘的危害性和国内外对综掘面粉尘治理研究的现状,根据风幕控尘的原理以及风幕集尘除尘系统的风幕形成条件和风流结构创造性的设计了风幕集尘射流装置。实验证明本文研制的倒“U”型射流腔与三角形射流箱构成一体的集尘射流装置较为合理,该装置所占用的空间可达最小,只要位置放置得当就不影响机体的三维移动与司机操作,另外采用三角形射流箱,不仅阻力较小,而且可以实现多级、多段调节并且能够控制射流的方向与风速的大小。从而可以控制综掘工作面的粉尘扩散,改善综掘工作面条件,净化了综掘巷道的风流。

参考文献

[1]孙明波,董曰喜. 现代化矿井采掘防尘技术的研究与应用[J]. 煤矿安全,2004,35(1):18-20.

[2]张大明,马云东. 矿井粉尘污染防治新技术浅析[J]. 辽宁工程技

术大学学报:自然科学版,2009,20(2):22-24.

[3]刘建,姚海飞,魏传光,等. 掘进工作面湿式离心除尘器的结构优化及数字模拟[J]. 煤炭学报,2010,35(3):424-428.

[4]杜翠凤,王辉,蒋仲安,等. 长压短抽式通风综掘工作面粉尘分布规律的数值模拟[J]. 北京科技大学学报,2010,32(8):957-961.

[5]刘荣华,王海桥,施式亮,等. 压入式通风掘进工作面粉尘分布规律研究[J]. 煤炭学报,2002,27(3):233-236.

[6]李雨成. 基于风幕技术的综掘面粉尘防治研究[D]. 阜新:辽宁工程技术大学,2010.

[7]LI Yucheng,LIU Jian,LIU Bo. Study on dust collection and removal systems in heading face based on air curtain technology[C]//International Mining Forum London;Taylor & Francis Group,2010:299-305.

[8]陈彩云. 转载点粉尘扩散模式与综合治理方案研究[D]. 阜新:辽宁工程技术大学,2008.

[9]MA Heng,LI Yucheng,LIU Jian. Development of dust collection and removal technology of comprehensive mechanized excavation face[J]. Journal of Coal Science & Engineering,2008,(12):567-570.

[10]张东辉,鲁民,关中吉. 利用雾化射流技术治理输煤皮带转运点粉尘污染[J]. 黑龙江电力,2004,26(1):61-63.

作者简介

杨靖(1990-),女,硕士在读,主要研究方向:粉尘防治;

李雨成(1978-),男,博士,副教授,讲师,主要研究方向:矿井通风,粉尘防治。