

基于改进遗传算法的网格任务调度研究

叶春晓^{1,2} 陆杰¹

(重庆大学计算机学院 重庆 400044)¹ (重庆大学软件学院 重庆 400044)²

摘要 网格任务调度是一个 NP 完全问题,它关注大规模的资源 and 任务调度,要求采用具有高效性的调度算法。提出了一种基于改进遗传算法的网格任务调度算法,在算法初始化种群产生时引入 min-min 算法和 max-min 算法,从而提高初始化种群的质量;算法迭代过程中采用了一种新的局部收敛判断以及改进的变异操作来防止局部收敛。仿真结果表明,该改进算法能更有效地解决网格任务调度问题。

关键词 网格,任务调度,遗传算法,局部收敛

中图分类号 TP309 **文献标识码** A

Grid Task Scheduling Based on Improved Genetic Algorithm

YE Chun-xiao^{1,2} LU Jie¹

(College of Computer, Chongqing University, Chongqing 400044, China)¹

(College of Software, Chongqing University, Chongqing 400044, China)²

Abstract Grid task scheduling is a NP-complete problem which concerns the scheduling of tasks and resources in a large scale, and thus a scheduling algorithm of high efficiency is required. A grid task scheduling algorithm based on GA was proposed. In the process of population initialization, a new method which combines the min-min algorithm and the max-min algorithm was addressed, and in the evolution of the population a new criterion predicting the premature convergence was presented and the corresponding improved mutation was designed to avoid premature convergence. The simulation results show that this improved algorithm can solve the problem of grid task scheduling more effectively.

Keywords Grid, Task scheduling, Genetic algorithm, Premature convergence

网格作为一种特殊的分布式系统,在提供前所未有的计算能力的同时,也面临着一些新的问题。其中,最主要的问题是如何调度应用程序到各种异构资源,即网格任务调度问题。由于网格系统的异构性和动态性,以及运行于网格系统之间的应用程序对于资源的不同需求,使得任务调度变得极其复杂;不好的任务分配策略,将会增加任务的总执行时间,从而降低整个网格系统的吞吐性能。

以时间跨度(makespan)为优化目标的任务-资源映射是一个 NP 完全问题,解决这类问题通常采用启发式算法。目前已有较多的任务调度算法,如基于遗传算法的任务调度、基于模拟退火算法的任务调度、基于蚁群算法的任务调度等。本文在研究现有算法的基础上,提出了一种基于改进遗传算法的任务调度算法。

1 网格任务调度问题及描述

影响分配和调度的因素主要有:任务计算时间、资源之间的数据传输延迟以及数据分配策略等。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。网格任务调度问题可描述如下:

(1) N 个任务的集合 $T = \{T_1, T_2, \dots, T_N\}$;

(2) M 个资源的集合 $R = \{R_1, R_2, \dots, R_M\}$;

(3) 一个 $N \times M$ 的矩阵 $\text{Time}[N, M]$, $\text{Time}[i, j]$ 表示任务 T_i 在处理机资源 R_j 上的运行时间。

网格任务调度以总的调度时间长度为优化性能指标,则任务分配和调度的目标是将这 N 个任务分配到 M 个资源上,并安排好它们的执行顺序,使任务的总体完成时间最短。

2 网格环境下遗传调度算法的改进

2.1 网格任务调度的编码和解码

目前遗传算法常用的编码方式是采用二进制编码,其具有编码解码操作简单、交叉和变异等遗传操作较容易实现等优点,但是有时却不能反映所求解问题的特点。编码方式对于算法的性能(如搜索能力和种群多样化)影响比较大,一般来说,浮点数编码比二进制编码在变异操作上更能保证种群的多样性。本文针对网格任务调度的特点,采用任务-资源相对应的浮点数编码方式,对每个任务占用的资源进行编码,每条染色体的长度等于任务的总体数量,染色体中的每一位都是正整数,代表的是任务编号,而该基因位置上的正整数代表

到稿日期:2009-10-05 返修日期:2009-12-01 本文受国家“十一五”科技支撑计划项目(2006BAH02A20),国家自然科学基金(60803027)资助。

叶春晓(1973—),男,博士,副教授,主要研究方向为网络信息安全、行业信息化等;陆杰(1986—),硕士生,主要研究方向为遗传算法、网格计算。

该任务所占用的资源的编号,如图1所示。

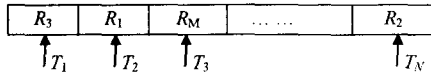


图1 任务-资源编码

其中 T_i 表示第 i 个待调度的任务, R_j 为任务 T_i 执行时所占用的第 j 个资源。染色体都是以数组形式存放的,在这种编码方式中,每个染色体向量被编码成一个与解向量长度相同的浮点数组,那么,在执行时遗传空间就是问题空间,染色体能够直接反映问题的规律和特性。

2.2 适应度函数的设定

适应度函数的选取要依据具体的调度目标而定。由于进化的准则是“优胜劣汰”,即遗传算法向着适应度增加的方向进化,因此适应度函数的寻优方向最好与适应度值增加的方向一致,这是适应度函数设计的先决条件。另外,为便于理论分析,最好保证适应度函数非负,所以对适应度为负的情况要进行适当的转换。由于本文主要的调度目标是最优跨度,也就是使所有任务的总执行时间(makespan)最小,而时间问题函数是非负的,我们只需要做适当转换就能够使得适应度函数的寻优方向与适应度函数值增加的方向一致。

定义适应度函数 $f(x)$ 为总执行时间的倒数,即 $f(x) = 1/\text{makespan}$ 。这样执行时间越短的染色体的适应度值越大,也就是说适应度值越大的个体的性能越好。于是问题就转换成求解一个染色体使其适应度值最大。

2.3 改进的种群初始化

初始种群的产生是遗传算法首先要解决的重要问题,如果初始种群具有较高的平均适应度值,则迭代过程中那些较优的个体就能引导种群向着好的方向进化,从而使迭代过程能够尽快收敛到最优解。因此,构建一个具有较高平均适应度值的初始种群,能够在很大程度上减少迭代次数,节省计算时间。

由于 Min-Min 算法和 Max-Min 算法是已经被证明的调度性能较好的搜索算法,利用这两种算法产生的个体质量较优,相应地对这些个体进行交叉和变异得到的个体也是较优的。改进后的种群初始化方法结合了 Min-Min 算法和 Max-Min 算法的优点,还采用海明距离(Hamming)保证了初始种群的多样性。所以用改进后的新方法产生的初始种群既可以保证种群的平均适应度值较高,又由于有随机产生的其余个体而可以保证种群的多样性。

常用的多样性保持策略是小生境技术,其主要思想是利用共享函数来限制相似个体的选择概率。但是,小生境技术对小生境半径非常敏感,小生境半径的略微偏移就会导致共享函数取值的极大偏差。因此,本文采用基于个体间的海明距离来保持种群的多样性,它不存在参数的估算和敏感问题。初始种群的建立采用海明距离控制个体差异,防止近亲繁殖。在产生种群的过程中,若新个体与前面产生的某一个体的海明距离小于某一设定值,则重新产生新的个体,直到得到所有个体之间均有一定差异的种群。种群的个体解遍及整个解空间,因此不仅能够很好地反映搜索空间,加快搜索速度,还能够很好地避免局部最优。改进后的种群初始化方法如图2所示。

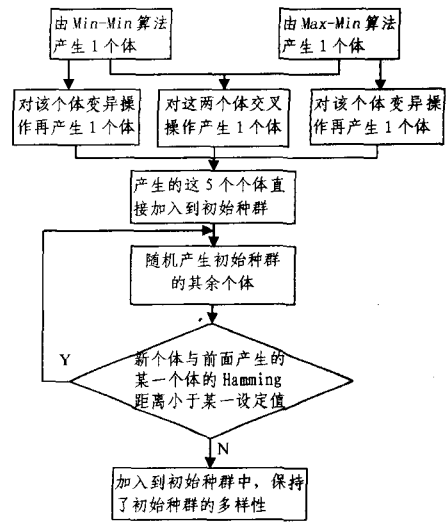


图2 改进后的种群初始化

2.4 改进的早熟局部收敛判断标准及相应操作

我们知道,进化后期的种群中适应度较大的那些个体要么重复,要么趋同,而遗传算法中适应度值大的个体的交叉和变异的概率又都很小,甚至为零,所以导致局部最优个体占据了整个种群而使得进化停止。于是,问题的关键就转化为需要判断最优个体的重复率以及趋同率。为此,本文提出了一种新的早熟局部收敛判断标准。

定义1 假设种群的大小为 N ,由个体 $x_1, x_2, \dots, x_N$ 构成,它们的适应度值分别为 $f_1, f_2, \dots, f_N$,将它们按从大到小的次序重新排列为 $f'_1, f'_2, \dots, f'_N$ 。取前 $N/2$ 个适应度值较大的个体,其平均适应度值为 $\bar{f}' = \frac{2}{N} \sum_{i=1}^{N/2} f'_i$,则定义 $\Delta = \sqrt{\frac{2}{N} \sum_{i=1}^{N/2} (f'_i - \bar{f}')^2}$ 来表征种群早熟局部收敛的产生。

这个定义之所以只计算前 $N/2$ 个适应度值较大的个体的差异,是基于以下考虑的:第一,这样可以排除适应度值低的个体对种群分布的影响,更能直接地反映较优个体的重复率以及趋同率。第二,遗传进化中的交叉操作总是两两配对的,而且较优个体的适应度值较大,被选中进入下一代的概率也就较大,若个体越优,其进行交叉和变异的概率也就越小,这样种群中 $N/2$ 个较优个体重复或者趋同进化,就会因为失去种群多样性而停滞不前,从而发生早熟局部收敛。

对于局部收敛发生后的变异,因此时已经发生了早熟而使种群失去了多样性,所以要使用相对“盲目”的随机变异来对局部收敛的种群作大规模调整,以增加种群多样性和避免早熟,不再采用本文后面提到的基于动态自适应变异概率的变异操作,而是进行“倒置大变异”。所谓“倒置大变异”,即指将染色体的首尾基因位相应倒置,从而产生新的个体,如图3所示。

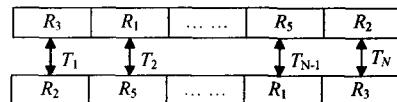


图3 “倒置大变异”示意

2.5 网格任务调度的遗传操作

2.5.1 选择操作

我们采用最优个体保护策略,即父代群体中的最优个体

直接进入子代群体中,这样可以防止最优个体由于遗传算子的偶然性而被破坏掉,且能以接近 1 的概率来找到全局最优解;也可将父代与子代一起排序,按群体规模截取前 N 个个体进入匹配集,作为下一步交叉操作的对象,这样可以使每一代中的优良个体得到保护,而淘汰那些不良个体,以期通过交叉等操作产生更优的个体,从而提高寻优速度。

2.5.2 自适应交叉和变异操作

普通自适应算法中,个体适应度值越接近最大适应度值,交叉概率与变异概率就越小;当它等于最大适应度值时,交叉概率和变异概率为零。这种调整方法对于种群处于进化后期比较合适,但对于进化初期不利,因为进化初期种群中的较优个体,尤其是适应度值接近种群最大适应度值的个体,几乎处于一种不发生变化的状态。这增加了进化走向局部最优的可能。因此,需要做进一步修改,使种群的最大适应度值的个体的 P_c 和 P_m 不为零。修改后的公式为:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}, & f' > f_{avg} \\ P_{c1}, & f' \leq f_{avg} \end{cases}$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f_{max} - f')}{f_{max} - f_{avg}}, & f' > f_{avg} \\ P_{m1}, & f' \leq f_{avg} \end{cases}$$

当 $f' = f_{max}$ 时, $P_c = P_{c2} > 0$; 当 $f = f_{max}$ 时, $P_m = P_{m2} > 0$ 。这样,种群中的较优个体相对于普通自适应遗传算法,拥有更高的交叉概率与变异概率。同时,为了保证每一代的最优个体不被破坏,采用最优保存策略,将其直接复制到下一代中。

上述公式中,一般取 $P_{c1} = 0.9$; $P_{c2} = 0.6$; $P_{m1} = 0.1$; $P_{m2} = 0.001$ 。

3 仿真实验结果

作者采用 Gridsim 网络任务调度模拟器,对改进后的遗传算法 IGA 进行了仿真,并将 IGA 与传统自适应遗传算法 AGA 就两种情况进行了试验:一种模拟轻负载情况;另一种模拟重负载情况,试验结果如图 4 和图 5 所示。

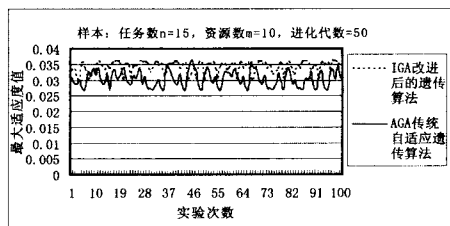


图4 轻负载(任务数 $n=15$, 资源数 $m=10$)的情况

由上述仿真实验结果分析可知,改进后的算法既具有较好的全局收敛能力,又有较快的收敛速度,能为网络任务调度节约时间,且改进后的算法对于规模大的网络任务调度效果更好,所以实现了本文以网络任务调度最优跨度为目标的实

验目的。

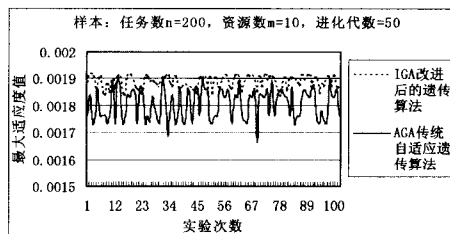


图5 重负载(任务数 $n=200$, 资源数 $m=10$)的情况

结束语 本文提出了一种基于改进遗传算法的网络任务调度算法,改进后的算法产生的初始种群既可以保证种群的平均适应度值较高,又由于有随机产生的个体而可以保证种群的多样性。算法迭代过程中采用了一种局部收敛判断的新标准以及改进的变异操作来防止局部收敛。仿真结果表明,该改进算法具有良好的效率和收敛性,能更有效地解决网络任务调度问题。

参考文献

- [1] Abraham A, Buyya R, Nath B. Nature's heuristics for scheduling jobs on computational grids[C]// The 8th IEEE International Conference on Advance Computing and Communications India, 2000
- [2] Karonls NT, Tonnen B, Foster I. MPICH-G2: A Grid Enabled Implementation of the Message Passing Interface[J]. Journal of Parallel(JPDC), 2003, 163(5): 551-563
- [3] Foster I, Kesselman M. The Grid: Blueprint for a Future Computing Infrastructure[M]. USA: Morgan Kaufmann Publishers, 1999
- [4] Beman F, Fox G, Tony H. Grid Computing-making the Global Infrastructure a Reality[M]. USA: John Wiley and Sons Ltd, 2003: 65-100
- [5] Spooner D P, Jarvis S A. Local Grid Scheduling Techniques using Performance Prediction[J]. IEEE Proceedings-Computers and Digital Techniques, 2003, 150(2): 87-96
- [6] 杨博, 陈志刚. 一种基于双层进化结构的网络任务调度算法[J]. 计算机工程与应用, 2006, 42(15): 8-10
- [7] 都志辉, 陈渝, 刘鹏. 网络计算[M]. 北京: 清华大学出版社, 2002
- [8] 王小平, 曹立明. 遗传算法——理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002
- [9] 袁禄来, 曾国荪, 姜黎立, 等. 网络环境下基于信任模型的动态级调度[J]. 计算机学报, 2006, 29(7): 1217-1224
- [10] 吴高峰, 蒋玉明, 等. 基于 QoS 改进的 Min-Min 网络调度算法[J]. 微计算机信息, 2009, 3-9: 110-112
- [11] 李敏, 吴浪, 张开碧. 求解旅行商问题的几种算法的比较研究[J]. 重庆邮电大学学报: 自然科学版, 2008, 20(5): 624-626

(上接第 155 页)

- [10] Sam Y, Boucelma O. Customizable Web Services Description, Discovery and Composition: An Attribute Based Formalism[C]// Proc. of 2007 IEEE/WIC/ACM International Conference on Web Intelligence, 2007
- [11] Liang Q, Miller S, Chung J. Service Mining for Web Service Composition[C]// Proc. of IEEE International Conference on Information Reuse and Integration (IEEE IRI-2005). 2005

- [12] Gaaloul W, Bhiri S, Godart C. Research Challenges and Opportunities In Web Services Mining[C]// Proc. of System and Information Service Web, INFORSID2006. 2006
- [13] Zhuge H. The Knowledge Grid Environment, IEEE Intelligent [J]. Systems, 2008, 23(6): 63-71
- [14] 代宇, 刘宴兵, 程瑶. 基于异步 Web Service 调用的 Web 应用程序研究[J]. 重庆邮电大学学报: 自然科学版, 2008, 20(6): 746-748