

一种基于遗传算法的网格任务调度算法

肖青, 杨长兴, 杨炼

(中南大学信息科学与工程学院, 湖南长沙 410083)

摘要:任务调度算法是网格计算研究的一个重要方向,已被证明是一个 NP 完全问题。提出了一种新的网格任务调度算法。该算法基于遗传算法,为加快算法的收敛速度,在生成初始种群时优先分配关键路径上的任务;由于资源间存在着通信延迟,引入任务复制方法,并结合遗传操作控制任务复制的深度,可以减少任务之间的通信开销,缩短整个调度的完成时间;最后进行优化操作,减少冗余的任务复制。模拟实验结果表明,该算法在收敛速度和调度完成时间均优于普通遗传算法。

关键词:网格任务调度;遗传算法;关键路径;任务复制

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2008)08-0032-03

Grid Task Scheduling with an Improved Genetic Algorithm

XIAO Qing, YANG Chang-xing, YANG Lian

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: Task scheduling is an important research aim in grid computing research area, and it has been proven to be NP-complete. Presents a new grid task scheduling algorithm. The scheduling algorithm based on genetic algorithm. For improving the convergence rate, preferentially assign tasks of the critical path to resources when generating initial populations. Because of the communication delays between resources, import task duplication method and incorporate new genetic operators to control the degree of replication of tasks, which can reduce task communication delays and shorten the length of schedules. At last, use optimal operator to reduce large task replication numbers. The algorithm was compared with common genetic scheduling algorithm in terms of convergence rate and complete time. The experimental results show the effectiveness of the proposed algorithm to the scheduling problem.

Key words: grid task scheduling; genetic algorithm; critical path; task duplication

0 引言

网格计算(Grid Computing)是当前互联网研究中的一个热点,也是并行和分布处理技术的一个发展方向,任务调度是其重要的组成部分。在网格计算中,一个大型的计算任务往往被分解成若干个彼此相互联系的子任务,这些子任务向资源管理系统提出资源请求,并按照一定的任务分配策略被分配给网格资源进行处理。在通常情况下,子任务的数量多于资源数量,因此为了提高资源的利用率并使计算任务在尽可能短的时间内完成,必须对资源和任务的调度进行优化。任务调度就是逻辑地为每个任务匹配一个最合适的机器,以便最小化应用程序的执行时间和完成时间的过

程。任务调度是一个极其繁琐复杂的问题,在一般形式下是一个 NP 完全问题。

在已往的研究中,产生了许多调度算法,如 Min-min 算法、Max-min 算法、模拟退火算法、蚂蚁算法和遗传算法等^[1],这些方法被广泛地应用在任务调度算法中。但这些算法都没有考虑到任务复制的可能性。任务复制的方法是指通过把一个子任务冗余的分配到一个或多个计算资源上,可以减少任务间的通信负荷,减少它们之间的通信延迟^[2]。

提出一种带任务复制的网格任务调度遗传算法。通过 Gridsim 仿真表明,在网格环境下,与普通遗传算法相比,本算法有较好的调度结果。

1 调度模型

这里把一组有序任务表示为一个加权的 DAG 图,从而将对子任务的调度问题转化为对调度 DAG 图的问题。将子任务表示为如下 DAG 图的形式: $G = \{V, E, P, W, C\}$ 。

收稿日期:2007-10-06

基金项目:湖南省自然科学基金项目(06JJ5131);湖南省教育厅科研资助项目(07C388)

作者简介:肖青(1983-),女,硕士研究生,研究方向为网格计算;杨长兴,教授,研究方向为网格技术、医学信息表达与处理。

V : v 个子任务的集合;

E : 子任务之间有向边的集合, 有向边 (v_i, v_j) 表示任务 v_i 在任务 v_j 完成之后才能开始执行;

R : r 个计算资源的集合;

W : 一个 $v \times r$ 的矩阵, $w_{i,j}$ 表示任务 v_i 在资源 r_j 上的执行时间;

C : 一个 $v \times v$ 矩阵, 表示任务 v_i 和任务 v_j 之间的通信延迟, 如果这两个任务分配到同一个资源上, 则认为通信开销为 0;

如果存在一条从 v_i 到 v_j 的路, 称 v_i 是 v_j 的前驱节点, 而对于 $(v_i, v_j) \in E$, 称 v_i 是 v_j 的立即前驱节点, 记为 $v_i \in iPred(v_j)$, 称 v_j 是 v_i 的立即后继节点, 记为 $v_j \in iSucc(v_i)$ 。通信计算比 (Communication-to-Computation Ratio, CCR) 是其平均通信量与平均计算开销之比。文中任务和节点可以互换。

2 算法描述

2.1 遗传算法简介

遗传算法是由美国 Michigan 大学 J. Holland 教授于 1975 年提出的, 是一种基于生物自然选择与遗传机理的随机搜索算法, 与传统的搜索算法不同, 遗传算法从一组随机产生的初始解称为“种群 (Population)”开始搜索过程。种群中每个个体是问题的一个解, 称为“染色体 (Chromosome)”。在遗传算法中最重要的概念是染色体, 染色体通常是一串数据 (或数组), 用来作为优化问题的解的代码, 其本身不一定是解。这些染色体在后续迭代中不断进化, 称为遗传。GA 有很多明显的优势, 如: 具有领域无关的群体性全局搜索能力, 可避免一些算法容易陷入的局部最优; 使用评价函数启发搜索过程; 使用概率机制进行迭代, 有随机性; 易于与其它优化技术或已有模型结合, 可扩展性强, 应用于任务调度具有其独特的优势。

2.2 染色体编码

处理单元的分配采用直接编码, 而任务开始执行时间则采用间接编码, 假定任务已经按照其高度值由小到大排序^[2,3], 任务的高度计算方法是:

$$\text{Height}(v_i) = \begin{cases} 0 & \text{如果 } iPred(v_i) = \emptyset \\ \max_{v_j \in iPred(v_i)} \text{Height}(v_j) + 1 & \text{其他} \end{cases}$$

染色体定义为一个两元组 $\{PE, \text{Priority}\}$, 其中 PE 中每个元素代表相应任务分配到的处理单元, 任务可以分配到几个不同的处理节点上, 且 $PE = 1, 2, \dots, 2^k - 1$, 其中 k 为系统中处理单元的数目, 解码时, 将 PE 值转化为二进制序列, 即可得到处理单元号。Priority 是任务的优先权值 (随机从 1 ~ 512 中产生), 用来构成一个满足先后次序条件的优先权列表。优先权列表由

下列公式构成:

$$Q(v_i) = \begin{cases} \text{Priority}(v_i) & \text{如果 } iPred(v_i) = \emptyset \\ \max(Q(v_i) \mid v_j \in iPred(v_i) + \text{Priority}(v_j)) & \text{其他} \end{cases}$$

具有较小 Q 值的任务, 优先级较高, 如果两个任务的 Q 值相等, 规定排在前面的任务优先级较高。

2.3 初始种群

对使用任务复制的调度算法^[4]的研究表明, 复制关键路径上的任务到多个处理节点上执行, 可以缩短程序执行时间, 文中生成初始种群的步骤为:

(1) 对每个位于关键路径中的任务, 在 $[1, 2^k - 1]$ 内产生一个服从均匀分布的随机数, 作为任务的 E 值, 而对其他任务, 在 $[0, k - 1]$ 内产生一个服从均匀分布的随机数 r , 用 2^r 作为任务的 E 值。

(2) 对于每个任务, 在 $[1, 512]$ 内产生一个服从均匀分布的随机数, 作为任务的 Priority 值, 这样就生成一个个体。

(3) 重复这个过程 m 遍就可以产生具有 m 个个体的初始种群。

2.4 适应度函数

适应度函数是用来评价染色体优劣的。本算法中, 适应度函数取决于调度的完成时间, 即所有子任务执行完成所用的时间。完成时间越短, 染色体越优, 该调度序列越优。

$est(v_i, r_j)$: 表示任务 v_i 在资源 r_j 上最早开始时间;

$ect(v_i, r_j)$: 表示任务 v_i 在资源 r_j 上最早完成时间;

公式(1):

$$est(v_i, r_j) = \max\{\text{avail}[r_j], \max(\text{AFT}(v_m) + c_{m,i})\}$$

其中, $\text{avail}[r_j]$ 表示为在资源 r_j 的最早可用时间, 即 r_j 上前一个任务的完成时间, $\text{AFT}(v_m)$ 表示 v_m 的前驱节点的完成时间。

公式(2):

$$ect(v_i, r_j) = w_{i,j} + est(v_i, r_j)$$

根据公式(1)可以计算出子任务在不同资源上开始执行的最早时间, 根据公式(2)可以计算出每个子任务的完成时间。记资源 R_i 上最后一个子任务的完成时间为 $ect(R_i)$, 则所有的子任务完成的时间为 $\max(ect(R_i))$ 。定义适应度函数为 $\text{Fitness} = 1/\max(ect(R_i))$, 适应值越大, 染色体越优。

2.5 遗传操作

2.5.1 选择

选择算子决定了哪些染色体进入下一代。本算法

中采用“轮盘赌”的选择方式,它按照染色体的适应值大小来确定该染色体的被选择概率。如果染色体的适应值越大,其被选中的概率越大。个体 c_i 被选中的概率定义如下:

$$p(c_i) = \text{Fitness}(c_i) / \sum_{j=1}^{pSize} \text{Fitness}(c_j)$$

其中 $pSize$ 为种群大小。在确定了每个染色体的被选择概率后,系统生成一个在 $[0,1]$ 区间的随机数组,并与对应染色体的被选择概率比较,如果随机数大于染色体的被选择概率则该染色体被选择,反之被淘汰。

2.5.2 交叉

遗传算法主要利用交叉算子^[5]对问题的解空间进行搜索。由于子任务之间存在约束关系,本算法采用内部交叉算子,即在一个染色体内随机选择两个资源,交换这两个资源上的任务序列,如果产生的新染色体适应值比原种群中适应值最差的染色体优,则替换原种群中那个最差的染色体。

2.5.3 变异

变异算子的主要途径是拓展新的搜索空间,在种群局部收敛时,通过变异算子的突变作用,来保持种群的多样性。它决定了遗传算法的局部搜索能力。遗传算法中变异算子的一般原理是将染色体中的某些基因的基因值用该基因座的其他等位基因来替代,从而形成一个新个体。本算法中,将变异算子分为 3 个算子:迁移、复制、删除。随机选择一个算子执行变异操作。

①迁移算子:随机选择一个子任务,将它从原来的资源迁移到它的一个父任务所在的资源上,若父任务所在的资源上已存在此任务,则重新选择。

②复制算子:随即选择一个子任务,将它的一个副本插入到其父任务所在的资源上,若已存在,则重新选择。

③删除算子:随即选择一个子任务,如果它在其他的资源上还有副本,则删除此任务,否则重新选择。

变异后产生的新染色体适应值比原种群中适应值最差的染色体优,则替换原种群中那个最差的染色体。

2.6 优化

经过遗传操作产生的染色体中,可能会产生一些冗余的任务副本,它对缩短整个调度的执行时间并没有任何作用,可以删除这些无用的任务副本,这样可以减少网络开销,减少所占用的资源。所以我们对前面遗传操作所产生的新的染色体进行优化,删除过分冗余的任务副本,具体描述如下:

(1)选择一个子任务进行判断,看它在别的资源上是否还有副本,如果有则执行步骤(2),否则重新选择一个子任务;

(2)计算删除该子任务后染色体的适应度,如果它不小于删减前的适应度,则删除该子任务,否则,保留该子任务;

(3)重复执行步骤(1)(2),完成染色体的优化。

3 实验结果和分析

文中用 Gridsim 和 eclipse 模拟有 30 个资源的网格环境,对本算法 TDGA、一般的遗传算法 GA 进行分析比较。算法的主要参数为:种群大小为 50,算法的交叉概率为 0.6,变异概率为 0.05。实验结果如图 1,图 2 所示。图 1 表示的是 DTAG 和 GA 在不同任务数情况下的平均调度完成时间的曲线图,可以看出,任务数越多,本算法的求解结果越优于其它两种算法。图 2 表示的是 DTAG 和 GA 随代数增加的进化求解曲线图,随着代数的增加,本算法向最优解收敛的速度快于 GA,且获得的解更优。

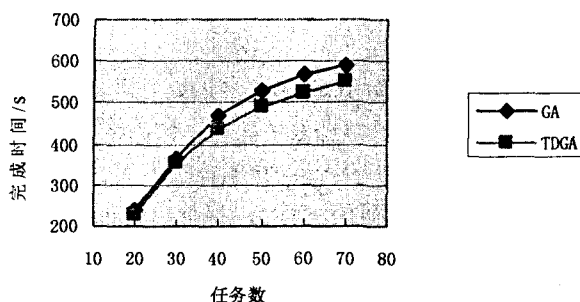


图 1 DTAG 和 GA 在不同任务数情况下的平均调度完成时间比较

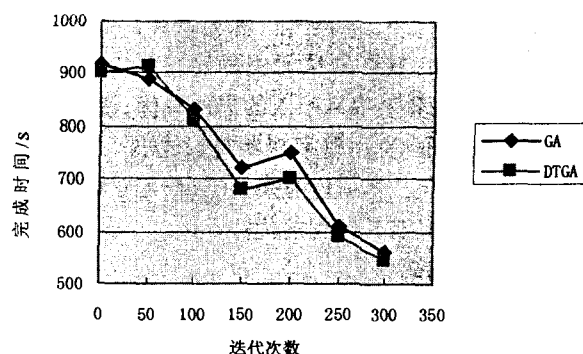


图 2 DTAG 和 GA 进化求解速度过程比较

4 结束语

提出了一种基于遗传算法的网格任务调度算法。本算法采用合适的编码方式,在产生初始种群时将关键路径上的任务优先分配到各资源上,加快了算法向最优解收敛的速度;加入了任务复制方法,并对生成的新染色体进行优化,可以有效地减少通信延迟,缩短执行时间。实验结果表明,与一般的 GA 算法相比,本算

(下转第 38 页)

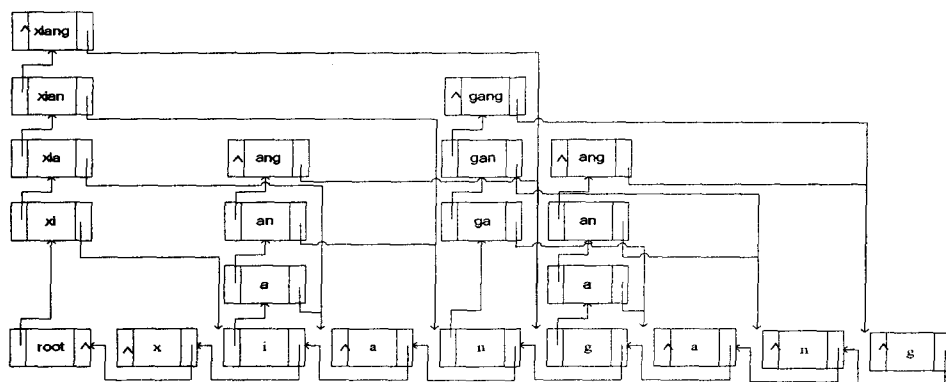


图 3 xiangang 的切分算法状态图

显然无法得到匹配的候选,而且也是一种浪费。

当用户删除最后一个音素节点,状态空间发生变化,如图 4。从音素节点 g 往前搜索,只要某合法音节节点的 pPYCharNode 指针域指向该节点,即由其产生的合法音节,在本例中有 ang 和 gang,则将其删除。最后删除该音素节点,更新中心链的尾指针。由遍历算法可知删除 g 后存在 xi'an'gan, xi'ang'an, xian'gan, xiang'an 四种切分可能。

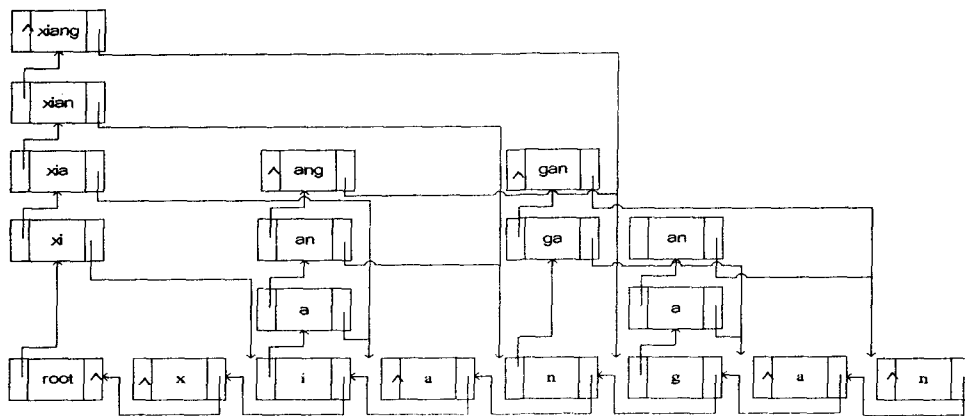


图 4 删除最后一个音素的状态空间

4 结束语

音节切分是整句拼音转换的前提和基础。为避免

由于切分错误较早地将正确的语句排除在外,提出一种音节切分算法,通过插入、遍历操作可得到所有的切分可能,再由统计和语法方法可选出全局最优的候选语句。该算法虽然应用于音节切分解决歧义问题,但其方法适用于所有需要全局考虑的问题,将所有的可能性都包

含其中。例如手写笔画输入法^[6]中,通过方向码构造汉字时,该算法可获得所有可能的笔画序列,再从字库里查找对应的汉字。

参考文献:

- [1] 吴根清. 统计语言模型研究及其应用[D]. 北京:清华大学, 2004.
- [2] 王晓龙,王开铸,李仲蓉,等. 最少分词问题及其解法[J]. 科学通报, 1988(11): 1030-1032.
- [3] 徐志明,王晓龙,姜守旭. 一种语句级汉字输入技术的研究[J]. 高技术通讯, 2000, 10(1): 51-55.
- [4] 张小衡. 不完全拼音码的模版处理——兼谈微软拼音输入法的进一步完善[J]. 计算机工程与应用, 2005(20): 74-101.
- [5] 李 炜,贾庆成,刘政治. 汉语拼音输入法中拼音流的切分[J]. 现代计算机(专业版), 2007(8): 11-13.
- [6] 樊庆林. 基于笔画的联机手写汉字识别系统的研究与实现[D]. 合肥: 安徽大学, 2007.

(上接第 34 页)

法具有更好的调度结果,是一种有效的网格任务调度算法。

参考文献:

- [1] Di Martino V. Scheduling in a grid computing environment using genetic algorithms[C]// In: Mililotti M. the 16th and Distributed Processing Symp(IPEPS 2002). Florida USA: [s. n.], 2002.
- [2] Xu Zhihong, Hou Xiangdan, Sun Jizhou. Ant algorithm - based task scheduling in grid computing[C]// In Proc of 2003 Canadian Conf on Electrical and Computer Engineering. Montreal, Canada: IEEE Computer Society Press, 2003: 1107-1110.
- [3] Di Martino V, Mililotti M. Suboptimal scheduling in a grid using genetic algorithms[J]. Parallel Computing, 2004, 30: 553-565.
- [4] Czajkowski K, Fitzgerald S, Foster I, et al. Grid Information Services for Distributed Resource Sharing[C]// In Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10). [s. l.]: IEEE Press, 2001: 181-195.
- [5] 钟求喜,谢 涛,陈火旺. 基于遗传算法的任务分配与调度[J]. 计算机研究与发展, 2000, 37(10): 1197-1203.