

文章编号: 1001-2486(2002)04-0069-04

基于进化的多任务分配与调度*

钟求喜, 胡华平, 祁悦

(国防科技大学计算机学院, 湖南长沙 410073)

摘要: 在考虑多种约束关系基础上, 提出一种并行与分布式系统中多约束关系的任务分配与调度的共同进化遗传算法。仿真试验结果表明所给算法比传统单种遗传算法更能有效地进行多任务分配与调度, 具有一定的工程价值。

关键词: 并行与分布式系统; 任务分配与调度; 进化计算; 合作式共同进化计算模型

中图分类号: TP301.6 文献标识码: A

Tasks Matching and Scheduling Based on Coevolution Computation

ZHONG Qiu-xi, HU Hua-ping, QI Yue

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Considering the multi-restraints among sub-tasks in parallel and distributed systems, a coevolutionary genetic algorithm (CGA) for tasks matching and scheduling is proposed based on computational model of cooperative coevolution. Simulation results show that the CGA is more effective than the conventional single population-based genetic algorithm (SGA) for multi-tasks matching and scheduling, and the algorithm is of practical use in engineering.

Key words: parallel & distributed systems, task matching and scheduling; evolution computation; computational model of cooperative coevolution

针对并行与分布式系统中任务分配与调度这一类 NP 问题^[1], 基于遗传算法^[2]等进化计算方法的研究工作已有很多^[3-8], 如基于同构系统任务分配与调度的遗传算法^[3,5], 动态的分配与调度算法^[6], 以及基于合作式共同进化计算模型的共进化方法^[4]。这些研究大都是基于同构系统或作了大量简化的异构系统。但是, 实际中各种计算机通过高速网络互连所组成的异构网络系统中, 任务分配与调度策略所依赖的约束关系是十分复杂的。主要的约束关系包括: 网络拓扑结构、网络连接节点间的通信机制与通信波特率、子任务在各处理机上的计算时间、子任务之间的数据输入输出逻辑关系及数据流量大小等等。

1 问题描述

基于异构网络计算环境的并行与分布式系统中独立多任务的分配与调度问题可由以下因素来描述:

① 设一个应用任务 $T_k = \{T_1^k, T_2^k, \dots, T_n^k\}$ 已经被分解为 n_k 个子任务, T_i^k 为任务 T_k 的第 i 个子任务。 n 个相互独立应用任务的集合记为 $T = \{T_1, T_2, \dots, T_n\}$

② 设 $P = \{P_1, P_2, \dots, P_m\}$ 为 m 个处理机 (异构/同构) 组成的处理机集合, P_i 为第 i 个处理机。

③ 设 $time$ 为子任务在各处理机上估计执行时间矩阵, $time [T_i^k, P_j]$ 表示子任务 T_i^k 在处理机 P_j 上的估计执行时间。

④ 网络的拓扑结构, 如全互连结构、星形网结构、环状结构及超立方体结构等。

⑤ 设 cw 为各处理机间的通信波特率矩阵, $cw [P_i, P_j]$ 表示处理机 P_i 与处理机 P_j 间的通信波特率。矩阵 cw 与网络拓扑结构相关。

* 收稿日期: 2002-01-10

基金项目: 国防科技大学预研基金资助 (cx-02-06-015)

作者简介: 钟求喜 (1969-), 男, 助理研究员, 博士。

⑥ 数据包的大小数组 $data$, $data [T_i^k]$ 表示子任务 T_i^k 所产生数据量的大小。

⑦ 一个应用任务中各子任务间的数据依赖约束关系通常用有向无回路图 (DAG) 表示。设应用任务 T_k 的 DAG 图为 $G_k = \langle T_k, E_k, time \rangle$, 其中, T_k 为子任务集, 一个子任务 T_i^k 就是图 G_k 中的一个节点, E_k 是任务依赖约束关系图中的有向边集, $time$ 为估计执行时间矩阵。 $\langle T_i^k, T_j^k \rangle \in E_k$, 表示在子任务 T_i^k 未处理完之前, 子任务 T_j^k 不可执行。这时将子任务 T_j^k 定义为子任务 T_i^k 的一个前继 (PRED), 将子任务 T_i^k 定义为子任务 T_j^k 的一个后继。

不妨设 s 为任务集合 T 上的一个分配与调度策略, $T_s(P_i)$ 表示在分配与调度策略 s 下处理机 P_i 完成其最后一个子任务所花费的时间, $\mu(s)$ 为整个分配与调度策略 s 所花费的时间, 则有

$$\mu(s) = \max_{1 \leq i \leq m} (T_s(P_i))$$

定义 $\mu(s)$ 为分配与调度策略 s 的调度长度, 则任务分配与调度的目标是在满足各任务中所有子任务间数据依赖约束关系的前提下, 为任务集合 T 寻找一个分配与调度策略使得其调度长度最小, 即

$$\min_s (\mu(s)).$$

2 任务分配与调度的共同进化遗传算法

共同进化计算模型借鉴了生物界中的共同进化机制, 是在现有进化计算的基础上形成的一种解空间分离编码的动态搜索和优化方法^[4,9], 能有效克服传统进化算法完整解编码技术的不足 (如完整编码对解的较好部分的利用可能会被其它较差的部分所掩盖)。

· 算法框架

采用共同进化计算模型, 异构网络环境中多约束关系的独立多任务的分配与调度算法 (CGA) 的基本框架如下:

(1) 依次读入每个应用任务的 DAG 图并生成其初始解群 $Pop_i(t)$, $t=0$, (i 为子种群的编号, t 为算法进化的代数);

(2) 计算初始解群中个体的适应值;

(3) 依次对每个子种群 $Pop_i(t)$ 分别进行选择、交叉^[4]和变异操作, 生成下一代解群;

(4) 根据各子种群状态采用合作式的共同进化方法计算解群中每个解的适应值, 并采用精英策略保存最好解;

(5) 如果算法终止条件满足, 转步骤 (6); 否则 $t = t + 1$, 转步骤 (3);

(6) 输出最好解, 算法终止。

由于各独立应用任务间没有约束关系, 算法中每个应用任务用一个子种群来对之进行求解, 所有子种群采用同步进化的方式依次各进化一代 (将所有子种群依次进化一代定义为共同进化算法的一代), 但子种群的同步进化要考虑到其它子种群的进化状态。在算法的并行实现时, 各子种群可采用异步方式的进化, 即各种子种群的进化速度可以不同。

· 编码方式

所有子种群中解个体染色体的编码技术均采用直观自然的列表方式, 即将分配到处理机 P_i 上的所有子任务按其在图 DAG_k 中高度值 $height$ 的升序排列成一个任务列表 l_i (一个处理机对应一个任务列表), 所有处理机的任务列表组成一个对该图 DAG_k 的分配与调度策略 $s_k = (l_1, l_2, \dots, l_m)$ 。图 DAG_k 中一个子任务 T_i^k 的高度值 $height(T_i^k)$ 可定义如下:

$$height(T_i^k) = \begin{cases} 0, & \text{如果 } PRED(T_i^k) = \emptyset, PRED(T_i^k) \text{ 表示子任务 } T_i^k \text{ 的前继节点集合} \\ 1 + \max_{T_j^k \in PRED(T_i^k)} (height(T_j^k)), & \text{其他} \end{cases}$$

· 适应值计算

子种群中个体的优劣是通过其适应值来评估的。显然, 子种群中个体适应值的计算依赖于该子种

群与其它子种群的相互作用方式,其个体的优劣取决于它与其它子种群中个体相互作用后对问题求解的贡献程度。

计算子种群 $\text{Pop}(t)$ 中个体 s_k (单个应用任务的子分配与调度策略) 的适应值时,必须考虑其它子种群的进化状态。最简单的方式是从其它各子种群中选择一个最好的个体作为代表,将所有代表与个体 s_k 组成一个完整的分配与调度策略 s , 则个体 s_k 的适应值 $f(s_k)$ 定义为

$$f(s_k) = c - f(s)$$

其中, c 是一个足够大的常数以保证 $f(s_k) \geq 0$, 例如可令

$$c = \sum_{i=1}^n \sum_{k=1}^{n_k} \left(\max_{1 \leq j \leq m} \left\{ \text{time}[T_i^k, j] + \frac{\text{data}[T_i^k]}{\text{cu}[P_i, P_j]} \right\} \right)$$

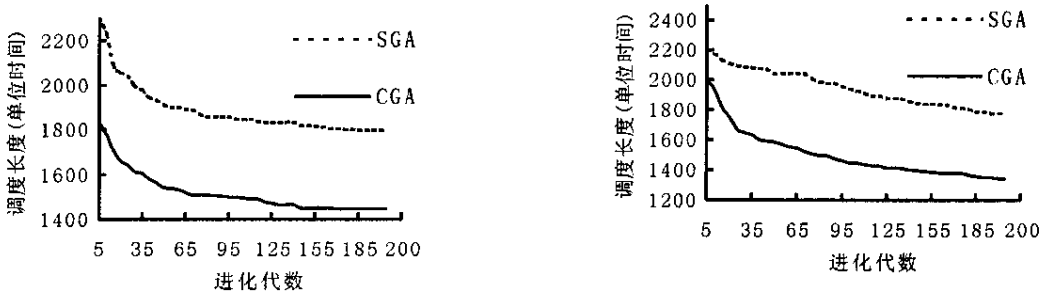
· 遗传操作

算法框架中的交叉算子采用改进的交叉算子、内部交叉算子和迁移算子^[4]。对于给定的分配与调度策略 s_1 和 s_2 , 交叉算子操作过程为:首先,生成随机数 q ($0 \leq q \leq h$, h 是相应 DAG 图的最大高度值), 在 s_1 和 s_2 的所有任务列表中选择交配点, 以便能将每个任务列表分成前后两部分。交配点的选择要满足两个条件, 一是所有紧跟交配点后子任务的高度值与 q 不等, 二是所有紧接交配点前子任务的高度值小于或等于 q 。其次, 相互交换 s_1 和 s_2 对应任务列表的后半部分所有子任务, 生成新的分配与调度策略 s'_1 和 s'_2 。

内部交叉算子的操作过程与改进的交叉算子相似, 但只在一个分配与调度策略内部相互交换两个任务列表上的后半分子任务。作为变异的迁移算子, 不同于标准变异算子在一个分配与调度策略内相互交换两个等高度值的子任务, 而是将一个子任务从一个任务列表迁移到另一个任务列表的相应位置上。

3 仿真实验

文献[4]的数学分析表明:基于独立多任务分配与调度的共同进化遗传算法中, 高于平均适应值的模式在算法后续代中试验次数的递增指数高于传统遗传算法的递增指数, 说明共同进化算法的效率优于传统单种群遗传算法。为检验文中算法对多约束关系的独立多任务分配与调度问题的有效性, 分别采用 CGA 方法和 SGA 方法对不同规模的问题作了对比仿真实验。两种方法中使用相同的遗传算子, 并采用轮盘赌规则的按比例选择和精英保留策略。实验中的重要参数有, CGA 中子种群的种群规模皆为 30, SGA 中的种群规模是 $30n$, n 为相应的独立任务个数, 交叉算子和内部交叉算子的发生概率分别为 0.2 和 0.8, 迁移概率为 0.2。文献[3]指出算法通常在 100 代左右时就收敛了, 故将算法的最多迭代代数设为 200。所有任务的 DAG 图是随机生成的, 每个任务的子任务个数为 10 至 50 个, 每个子任务



(a) $n=5, m=3$ 全互连网络结构

(b) $n=5, m=5$ 环状网络结构

图1 静态性能曲线

Fig.1 Schedule length vs. generation

在各处理机上的估计执行时间(单位时间)为1到100之间的一个随机数。处理机间通信波特率矩阵的数据是10到50间的随机数(单位波特率),数据包大小数组的数据是10到50间的随机数(单位大小)。

表1列出了仿真比较结果(表中完成时间为10次独立运行结果的平均值),其数据表明CGA比SGA能找到更好的解,并且,随着独立任务和子任务数量的增多,算法效率也随之提高。图1中给出了算法在不同进化代找到最好解的静态性能曲线,其收敛曲线说明CGA比SGA的收敛性好。以上仿真结果均表明,多约束关系的独立多任务分配与调度的共同进化遗传算法的效率高于传统单种群遗传算法。

表1 仿真比较结果

Tab.1 Simulation results

拓扑结构	任务个数 n	子任务总个数	调度长度(单位时间)		$\frac{SGA - CGA}{SGA} \times 100\%$
			SGA	CGA	
全互连	2	50	746.31	668.59	10.4
	5	120	1797.68	1448.40	19.4
	10	230	3837.17	2877.92	25.0
环状网	2	80	1055.54	917.93	13.0
	5	150	1770.45	1340.35	24.3
	10	320	3468.08	2421.77	30.2

4 结论与展望

针对并行与分布式系统中的任务分配与调度问题,考虑与实际系统相近的多种约束关系基础上,本文提出一种多约束关系的任务分配与调度的合作式共同进化方法。仿真结果说明文中算法比传统单种群进化类算法更能有效地进行多任务分配与调度,因而具有一定的实际工程价值。

网络计算中多约束关系增加了任务分配与调度的困难。而网络环境中应用任务的难以预测,更是增加了调度问题的难度和复杂性。这就必须综合考虑到各处理机的负载平衡,研究网络计算中任务的动态分配与调度技术,以适应发展越来越快的异构网络计算的需求。

参考文献:

- [1] Coffman E G. Computer and Job - Shop Scheduling Theory [M]. John Wiley & Sons , Inc. 1976.
- [2] Holland J H. Adaptation in Natural and Artificial Systems [M]. The University of Michigan Press , 1975.
- [3] Edwin S H , Nirwan Ansari. Genetic Algorithm for Multiprocessor Scheduling [J]. IEEE trans. on Parallel and Distributed systems , 5 (2) , Feb. 1994.
- [4] 钟求喜. 网络计算中任务分配与调度的遗传算法研究 [D]. 国防科技大学研究生院,长沙,2000 (10)
- [5] Intiaz Ahmad , Muhammad K. Multiprocessor Scheduling in a Genetic Paradigm [J]. Parallel Computing , 22 : 395 - 406 , 1996.
- [6] Gayathri Krishnamurthy. Scheduling Aairiable - width Parallelism Workstations [D]. Purdue University , 1998.
- [7] Lee Wang , Howard Jay Siegel , et al. Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic - Algorithm - Based Approach [J]. J. of Parallel and Distributed Computing , 47 (1) , Nov. 1997.
- [8] Yu - Kwong Kwok , Ishfaq Ahmad. Efficient Scheduling of Arbitrary Task Graphs to Multiprocessors Using a Parallel Genetic Algorithm [J]. J. of Parallel and Distributed Computing , 47 (1) , Nov. 1997.
- [9] Mitchell A , Potter The Design and Analysis of a Computational Model of Cooperative Coevolution [D]. George Mason University , 1997.

