

文章编号: 1007-130X(2003)04-0088-03

一个基于多 Agent 的 PCB 装配调度系统^{*} A Multi-Agent-Based PCB Assembly Scheduling System

刘 青, 余 智

LIU Qing, YU Zhi

(华中科技大学计算机学院, 湖北 武汉 430074)

(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

摘 要:PCB 装配调度问题是一个 Job Shop 调度问题。本文针对这一问题设计了一个基于多 agent 的调度系统模型, 其最大特点是将任务的优先级调度机制与基于合同网协议的投标机制相结合。文章首先说明系统的调度过程和各个 agent 的功能; 然后介绍多 agent 之间的 KQML 消息通信机制; 最后举一个实例并总结系统的特点。

Abstract:The PCB assembly scheduling problem is a Job Shop scheduling problem. To solve this problem, a scheduling system model based on multiple agents is constructed. The special feature of the system is combining the task priority scheduling mechanism with a bidding mechanism based on the Contract Net protocol. We first explain the scheduling mechanism and the function of each agent in the system, introduce the KQML message communication mechanism, and finally give an example to summarize the system's features.

关键词:agent; 多 agent 系统; Job Shop 调度; KQML; 合同网

Key words:agent; multi-agent system; Job Shop scheduling; KQML; Contract Net

中图分类号: TP391

文献标识码: A

1 引言

作业调度是给任务分配资源, 以保证任务在合理的时间内完成^[1]。它是一类应用广泛的运筹学问题, 这些问题大都被证明是不存在多项式时间纯优解法的 NP 问题^[2]。调度问题有很多研究方法, 主要有数学编程方法, 如数据建模和规划, 涉及运筹学、随机优化、控制理论等; 分布式规则; 离散仿真; 搜索方法, 如禁忌搜索、模拟退火等; 神经网络; 人工智能技术, 如专家系统、分布式人工智能等; 遗传算法等等^[3]。其中, 数学规划由

于其计算复杂性的原因, 不能获得真正的实用; 搜索法对于较大规模的问题搜索效率很低; Petri 网存在节点语义的单义性, 使所携带的系统信息量不够丰富、重用性差和难于对高级的调度规则建模等问题; 模拟退火法收敛速度较慢, 很难用于实时动态调度环境; 遗传算法的计算速度较慢, 分派式规则和神经网络等亦是如此。而基于 agent 的调度方法是一种基于知识的、分布式人工智能方法, 它利用 agent 基于知识的推理能力和多 agent 之间的协作机制可以较好地解决调度问题。它放松了对集中、规划和顺序控制, 提供了分散控制、应急和并行处理。

^{*} 收稿日期: 2002-04-01; 修订日期: 2002-06-18

基金项目: 国家 863 计划资助项目 (863-511-010-158)

作者简介: 刘青 (1967-), 女, 湖北武汉人, 副教授, 研究方向为计算机集成制造和人工智能; 余智, 硕士生, 研究方向为 MAS。

通讯地址: 430074 湖北省武汉市华中科技大学计算机学院; Tel: (027)87544285; E-mail: vzhustcs994@sina.com

Address: School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, P. R. China

目前,国内外有许多基于 agent 的调度系统的研究。张宇等提出的多代理调度系统模型是一个多 agent 系统^[4],系统分为管理 agent、生产 agent 群和工件 agent 群等。他把多 agent 的思想和基于规则调度的思想结合在一起,解决调度问题,但规则调度对多 agent 的协作求解能力体现不够,没有涉及 agent 之间的通信协议和交互语言。MetaMorphII 是一种基于混合型中介 agent 体系结构的仲裁模型,采用基于合同网协议的投标机制,进行制造调度和再调度研究^[5]。AARIA 将制造的人、机器和零件等等封装成自治的 agent,采用一种混合的启发式调度技术。

本文以国家 863 项目武汉邮科院 CIMS 二期工程为背景,为 PCB(印制电路板)装联车间构造一个基于多 agent 的 PCB 装配调度系统模型。

2 系统的多 agent 模型

PCB 装配车间的生产具有多品种、小批量、制造周期短、质量要求高、不确定性事件多等特点。其调度问题实际上是 Job Shop 调度问题,有以下三个特点:(1)有 m 类 PCB 需要装配,每类 PCB 有一定批量,根据 PCB 的组装类型和工艺流程不同,每类 PCB 操作个数和顺序也不同;(2)有 n 台机器,每台机器可完成多个操作,同一操作可在同一类的不同机器上完成,且完成时间不同;(3)每类 PCB 有交货期。在这个 Job Shop 调度问题中,称每一个 PCB 工艺流程为一个作业,称工艺流程中的每一道工序为一个任务。作业由任务的集合组成,任务之间存在时间上的偏序关系,作业任务序列可用有向活动网表示。

针对 PCB 装配的特点,结合 MetaMorphII 制造系统和 Job Shop 仿真系统^[6]的设计方法,本文提出了一个基于多 agent 的 PCB 装配调度系统模型,如图 1 所示。

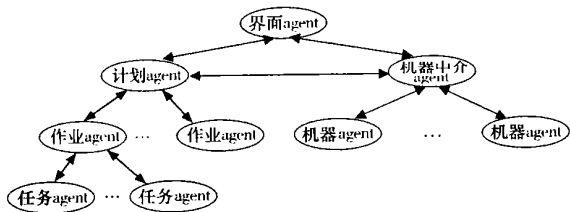


图 1 PCB 装配调度系统的多 agent 模型

它是一个层次结构模型,其中有六种 agent:

界面 agent、计划 agent、机器中介 agent、作业 agent、任务 agent 和机器 agent。计划 agent 管理多个作业 agent,每个作业 agent 管理多个任务 agent,机器中介 agent 管理多个机器 agent。

系统的运行机制是:用户通过界面 agent 输入 PCB 生产订单,界面 agent 将这个新订单信息发送给计划 agent,计划 agent 为每个订单创建一个作业 agent,作业 agent 为它的每个任务创建一个任务 agent。系统中 agent 都各自运行,并通过消息机制进行协作。首先,作业 agent 确定作业的关键路径和各个任务的优先级。然后,任务 agent 向计划 agent 发送资源请求消息,计划 agent 将所有任务请求消息按任务类型分类,并将每类中的消息按任务优先级排序,将各类型消息中优先级最高的那一条消息发给机器中介,机器中介和相应的机器 agent 用合同网协议进行求解。此过程可分为三步:机器中介向相应机器组中的所有机器 agent 招标,如图 2 所示;机器 agent 收到任务的资源请求后,根据自身完成此任务的花费时间投标;机器中介根据规则决定哪个机器 agent 被选中,并向它发布确认消息。

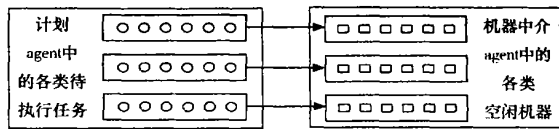


图 2 计划 agent 中的任务与机器中介 agent 中的机器的对应关系图

图 2 中的左边方框表示计划 agent,计划 agent 中任务请求按类型分组,每组中每个小圆圈代表一个任务 agent 的资源请求消息,各组中的消息按优先级排序;右边的方框表示机器中介 agent,机器中介 agent 中空闲机器也按类型分组,每组中每个小方框代表一台空闲机器。

下面分别具体说明系统中各种 agent 的功能:

(1) 界面 agent 处理用户的输入和系统的输出。用户的输入包括订单输入、订单修改、订单取消和系统设置等,界面 agent 可以收集这些数据(包括 PCB 的生产批量、装联工序、交货期或机器的特性等),并通知计划 agent 或资源中介进行相应的处理;此外,界面 agent 还负责将调度结果输出到显示器。

(2) 一个作业 agent 对应于一个作业,它包含下列信息:任务集、最迟结束时间、最早开始时间、重新规划标志和规划结束标志等。作业 agent 负责创建与其相应的任务 agent,分析作业的关键路

径和计算任务的优先级等。作业 agent 分析作业的关键路径时,各任务的执行时间按任务的平均执行时间计算,它先正向计算各任务的最早开始时间,然后反向计算各任务的最迟结束时间等等,在此基础上,计算各任务的空闲时间,以此作为任务的优先级。

(3) 一个任务 agent 对应于一个任务,它包含下列信息:任务类型、批量、直接前驱任务集、直接后继任务集、最早开始时间、最迟结束时间、空闲时间和优先级。只有当本任务的所有前驱任务都处理完之后,任务 agent 才能发资源请求消息。任务的优先级设定为任务的空闲时间(空闲时间=最迟结束时间-最早开始时间-执行时间),优先级值越小表示任务越紧迫,调度的优先级就越高。开始时,任务的优先级是按这类任务的平均执行时间估算的,当任务在机器上执行之后,用实际的执行时间再次计算剩余任务的优先级。

(4) 计划 agent 包括任务集、按类型分组的任务集和已调度结果集等信息。它将请求的任务按类型分组,各组的任务相似。组中任何一个任务可在机器中介 agent 中的相应空闲机器组中的任何一台机器上执行,如图 2 所示。计划 agent 要为每个作业确定关键路径,计算最早开始时间和最迟结束时间等。它将各组的任务按优先级排序,当机器中介的相应机器组有空闲机器时,发出资源请求,并等候机器中介通知结果。

(5) 机器中介 agent 包括机器集、按功能分组的机器集和各组的空闲机器集等信息。它将机器 agent 按功能分组,组中的机器具有相似的任务处理能力。机器中介 agent 接受计划 agent 的资源请求,并将此请求多播给相应的机器,处理机器 agent 的投标,并且确定最合适的投标。

(6) 一个机器 agent 对应于一台机器,如贴片、插件或测试仪等,它是任务的执行者。机器 agent 有待处理的任务和机器特性等信息。机器 agent 估算任务执行时间,向机器中介投标。

3 Agent 之间的通信

KQML(Knowledge Query and Manipulation Language, 简称 KQML) 是美国高级研究规划局(ARPA)KSE(Knowledge Sharing Effort, 简称 KSE)设计的支持智能软件 Agent 间交互的语言^[7],它定义了一种 agent 之间传递信息的标准语法以及一

些动作。本系统采用基于合同网的 KQML 消息通信,其中有三种类型的消息:request、reply 和 inform。下面是资源请求和分配的合同网进行协商的过程中用到的几个消息:

(1) 任务 agent 请求机器中介分配资源:

```
(request resource
:sender [sender] //任务 agent 标识
:receiver [receiver] //机器中介标识
:job.id [job ID] //作业标识
:task.id [task n ID] //任务标识
:task.type [task type] //任务类型
:batch [batch] //批量
:prior [priority] //优先级
:earliestStart [earliestStart] //最早开始时间
:latestFinish [latestFinish] //最迟结束时间)
```

(2) 机器中介向有关机器 agent 宣布招标:

```
(request call for bid
:sender [sender] //机器中介 agent 标识
:receiver all related machine //机器 agent 标识
:job.id [job ID] //作业标识
:task.id [task n ID] //任务标识
:task.type [task type] //任务类型
:batch [batch] //批量)
```

(3) 机器 agent 投标:

```
(reply bid
:sender [sender] //机器 agent 标识
:receiver [receiver] //机器中介 agent 标识
:job.id [job ID] //作业标识
:task.id [task n ID] //任务标识
:batch [batch] //批量
:execTime [execTime] //执行时间)
```

(4) 机器中介通知任务 agent:

```
(inform award
:sender [sender] //机器中介标识
:receiver [receiver] //任务 agent 标识
:job.id [job ID] //作业标识
:task.id [task n ID] //任务标识
:task.type [task type] //任务类型
:batch [batch] //批量
:execTime [execTime] //执行时间
:execMachine [execMachine] //执行机器
:earliestStart [earliestStart] //更新的最早开始时间
:latestFinish [latestFinish] //更新的最迟结束时间)
```

机器中介通知机器 agent 消息,类似于机器中介通知任务 agent 消息。

4 应用实例

原型系统是在 Windows2000 平台下,使用 VC++6.0 开发工具实现的。

假设要生产两种 PCB 各 20 块,即有两个作业 J_1 和 J_2 ,车间有五台机器 M_1 、 M_2 、 M_3 、 M_4 和 M_5 。 M_1 和 M_4 是贴片机; M_2 和 M_3 是插件机; M_5 是测试仪。 J_1 有三个任务(此例中任务有所简化):贴片 T_{11} 、插件 T_{12} 和测试 T_{13} ,且 T_{11} 是 T_{12} 的直接前驱任务。 J_2 有四个任务:正面贴片 T_{21} 、反面贴片

(下转第 98 页)

(1) 如何对间接无条件分支指令进行预取。

(2) 目前, 预取指令的判断和执行机制还不是很完善, 需要进一步改进。

(3) 无论预取技术如何高明, 只能将指令读入指令 Cache。能否为执行模块提供足够多的指令供其执行, 还需要取指令机制的支持。也就是说, 能否充分发挥指令 Cache 的性能, 需要高效取指令机制的支持。

参考文献:

[1] R Colwell, R Nix, J O'Donnell, et al. A VLIW Architecture for a Trace Scheduling Compiler[A]. Proc of the 2nd Int'l Conf on Architectural Support for Programming Languages and Operating Systems[C]. 1987. 180—192.

[2] W Hwu, S Mahlke, W Chen, et al. The Superblock: An Effective Technique for VLIW and Superscalar Compilation[J]. The Journal of Supercomputing, 1993, 7, 229—248.

[3] C Xia, J Torrellas. Instruction Prefetching of Systems Codes with Layout Optimized for Reduced Cache Misses[A]. 23rd Annual Int'l Symp on Computer Architecture[C]. 1996.

[4] N Jouppi. Improving Direct-Mapped Cache Performance by the Addition of a Small Fully Associative Cache and Prefetch Buffers[A]. Proc of the 17th Annual Int'l Symp on Computer Architecture[C]. 1990.

[5] J Pierce, T Mudge. Wrong-Path Instruction Prefetching[A]. 29th Int'l Symp on Microarchitecture[C]. 1996. 165—175.

[6] D Joseph, D Grunwald. Prefetching Using Markov Predictors[A]. 24th Annual Int'l Symp on Computer Architecture[C]. 1990.

(上接第 90 页)

T_{22} 、插件 T_{23} 和测试 T_{24} , 且 T_{21} 和 T_{22} 是 T_{23} 的直接前驱任务。机器执行各个任务的估计时间、各机器的估计准备时间和各个作业的最迟结束时间如表 1 所示。

表 1 作业、任务和机器的有关时间属性表

		机器 (准备时间)					
		$M_1(10)$	$M_2(9)$	$M_3(8)$	$M_4(11)$	$M_5(15)$	
作业 (交货期)	任务 (执行时间)						
	$J_1(100)$		$T_{11}(0.3)$	$T_{12}(0.4)$	$T_{12}(0.5)$	$T_{11}(0.2)$	$T_{13}(0.2)$
	$J_2(70)$		$T_{21}(0.2)$	$T_{23}(0.5)$	$T_{23}(0.4)$	$T_{21}(0.2)$	$T_{24}(0.2)$
			$T_{22}(0.3)$			$T_{22}(0.4)$	

表 1 假设同一机器对于不同任务的准备时间相同, 在任务到达后机器才开始准备, 并规定批量的作业任务是一次执行不能中断。例如, 完成作业 J_1 的 20 块 PCB 的装配任务 T_{11} 需要的时间为 $10 + 0.3 * 20 = 16$ 。系统产生的调度结果如图 3 所示。

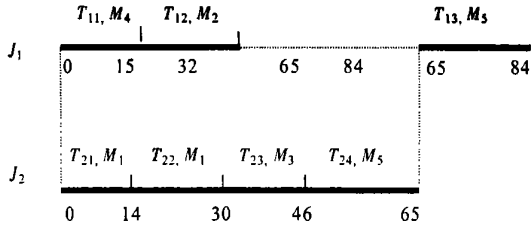


图 3 作业调度的甘特图

5 结束语

基于多 agent 的 PCB 装配调度系统具有以下四个特点: (1) 动态可扩展性。多 agent 系统的拓扑结构可以动态改变, 系统在调度过程中, 可以动态增加作业 agent 或者机器 agent。 (2) 较好的自适应性。当出现作业取消、延期、机器故障或资源短缺时, 系统仍然照常运行。 (3) 采用基于合同网协议扩展的 KQML 消息通信机制。 (4) 采用将作业的优先级与合同网的协商相结合的调度机制, 避开了组合爆炸问题; 不必进行大量的计算, 能在较短的时间内获得调度问题的近优解, 既满足客户的交货期需求又保证有较高的机器利用率; 产生的调度方案优于人工调度方案, 可以用于指导生产。当然, 系统还有一些需要完善的地方, 进一步的工作是用 agent 进行生产控制, 集成包括机器、元器件、工具、工人、AGV 等所有生产资源, 使它们协同工作, 实现智能制造。

参考文献:

[1] 熊锐, 吴澄. 车间生产调度问题的技术现状与发展趋势[J]. 清华大学学报(自然科学版), 1998, 38(10): 55—60.

[2] 熊有伦, 赵东波. 制造自动化系统优化调度、体系结构和关键技术基础[J]. 世界科技研究与发展, 1997, 19(5): 30—40.

[3] L C Rabelo Jones. Survey of Job Shop Scheduling Techniques[EB/OL]. <http://www.nist.gov/msdlibrary/doc/jobshop1.pdf>, 2001—10—12.

[4] 王艳红, 尹朝万, 张宇. 基于多代理和规则调度的敏捷调度系统研究[J]. 计算机集成制造系统, 2000, 6(4): 45—60.

[5] F Maturana, W Shen, D H Norrie. MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing[J]. International Journal of Production Research, 1998, 37(10): 2159—2174.

[6] Jaap A Ottjes, Hans P M Veeke. Production Scheduling of Complex Jobs with Simulation[EB/OL]. <http://www-itt.wbmt.tudelft.nl/users/duinker/papers/was0004c.pdf>, 2001—10—15.

[7] Tim Finin, Richard Fritzson, Don McKay, et al. KQML as an Agent Communication Language[A]. Proc of the 3rd Int'l Conf on Information and Knowledge Management[C]. Gaithersburg, Maryland, 1994. 122—131.