

基于多线程多处理器理论的 AS/RS 调度策略研究

王海宾, 张小志, 游凯何

(邢台职业技术学院 信息工程系, 河北 邢台 054000)

摘 要: 为了优化自动化立体仓库入库调度策略, 文章引入多线程技术与多处理器理论, 并利用该思想对大型自动化立体仓库入库调度进行了深入研究与分析, 提出了基于多线程技术与多处理器理论的大型 AS/RS 入库调度策略, 有效提高了大型 AS/RS 入库效率, 通过测试用例验证了该调度策略的可行性。

关键词: 多线程; 多处理器; 调度策略

中图分类号: TP311.51

文献标识码: A

文章编号: 1000-7180(2011)01-0142-05

The Large-scale AS/RS Storage Scheduling Strategy Based on Multi-processor and Multi-threading

WANG Hai-bin, ZHANG Xiao-zhi, YOU Kai-he

(Information and Engineering Department, Vocational and Technical College of Xingtai, Xingtai 054000, China)

Abstract: In order to optimize storage scheduling strategy of the automated warehouse, a multi-processor and multi-threaded theory is introduced in the paper, using this theory a deep research and analysis for large-scale AS/RS storage scheduling is made, and a large-scale AS/RS storage scheduling strategy based on multiprocessor and multi-threading is put forward. The storage efficiency of the large-scale AS/RS is effectively improved using the scheduling strategy put forward in this paper and through the test case verified the feasibility of the strategy.

Key words: multi-threading; multi-processor; scheduling strategy

1 引言

自动化立体仓库 AS/RS (Automated Storage and Retrieval System) 是指不用人工直接处理、能自动存储和取出物料的系统。自动化仓库技术是现代物流技术的核心, 它集高架仓库及规划、管理、机械、电气于一体, 是一门综合性的技术。如何建立良好的调度策略^[1], 提高调度效率, 已成为非常重要的课题。

学术界对自动化立体仓库调度系统方法的研究很多, 文献[2]提出将专家系统的思想应用于自动化立体仓库的入出库调度控制中; 文献[3]提出了一个基于调度和控制系统的混合智能 Agent 结构并将其应用于 AS/RS 的货位拣选问题; 文献[4]运用遗传算法对自动化立体仓库调度进行了研究; Donald.

Tepas 引入一种知识信息系统来辅助决策立体仓库的入出库调度算法。但是将 Windows 内核的多线程技术与多处理器理论运用于自动化立体仓库的研究还不存在。因此, 文中从多线程的角度提出并建立了一种大型 AS/RS 入库调度策略。

2 问题描述

文中的背景是大型自动化立体仓库。一般大型 AS/RS 通常包括: 堆垛机、环形穿梭车、缓冲输送机、下货口输送机、货架、托盘等设备。货物生产完成并且组盘后通过输送线到达下货口输送机, 环形穿梭车从下货口输送机接载货物, 并通过环穿轨道将货物运动到入库口输送机, 当货物到达入库末端输送机时调度堆垛机将货物入到货架上。各设备通过系统调度协作完成每项任务, 当然所有的操作都遵

循一定的原则,其结构如图 1 所示。

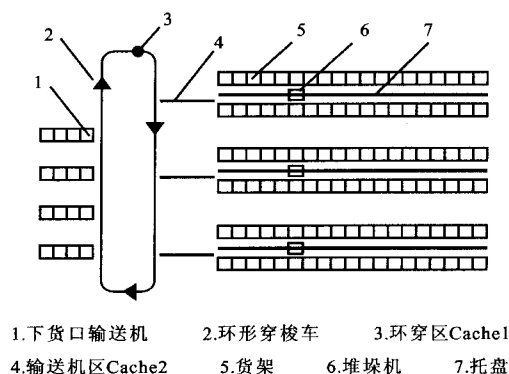


图 1 结构示意图

入库调度策略指有效调度已有设备,在最短的时间内将货物从下货口输送机送入到货架上。要使入库策略的效率最优,主要应考虑堆垛机的调度和环形穿梭车的调度。针对多台堆垛机的作业调度采用多处理器多线程思想,环形穿梭车的调度一直是入库调度的瓶颈,为了更好的调度环形穿梭车,并且使整体形成统一的调度模型,将其归纳为线程的二级缓存,其最大容量等于环形穿梭车的个数,将入库输送缓冲区归纳为一级缓存,其最大容量等于该缓冲队列长度,由此整个入库调度过程就是一个典型的多处理器多线程的调度。

3 多线程理论

多线程 (Multi-Threading, MT)^[5],是多任务的一种特殊形式。简单的说,多线程就是在同一时间同时处理多项任务。针对多处理器,多线程的实现方式主要有三种。粗粒度方式(Gorse Threading),类似传统的任务并行(TaskParallelism)模式,即基于处理器内核数量将系统划分为同等数量的子系统,每个线程负责一个子系统,所有线程按照一定策略分配到各个核^[6]。细粒度方式(Fine-grained Threading),类似传统的数据并行(DataParallelism)模式,即将问题划分为细小的、独立的任务,并将这些任务遍布于各个核^[7]。混合方式(Hybrid Threading),是以上两种模式的综合,运用粗粒度方式将系统划分为多个子系统,并用细粒度方式将各个子系统中涉及的微小任务抽象出来并线程化,散布于各个核中;同时,加入对任务线程的合理调度与分配,使得最大限度的利用多处理器资源,提高应用程序性能。

文中采用粗粒度方式来实现多线程多处理器的调度,即基于处理器内核数量将系统划分为同等数

量的子系统,每个线程负责一个子系统,所有线程按照一定策略的分配到各个核,每个核在 AS/RS 中对应每台堆垛机。每台堆垛机都可以独立完成分配到其上的线程。由于实际需要不考虑时间片轮转的情况。

4 数学模型的确定

AS/RS 中的每台堆垛机都有独立完成入库任务的能力,所以每台堆垛机相当于一个独立的 CPU,大型 AS/RS 中有多台堆垛机,这就构成了多个处理器。每个要执行的作业的货物都以托盘的方式存放,货物和托盘一起入到货架中。每一个要入库的作业就是一个线程,需要入库的作业可能同时有多个这就形成了多线程。而整个调度过程就形成了多处理器多线程的调度。

4.1 模型归纳

4.1.1 模型基础

多处理器的调度算法有多种,本模型采用处理器专派调度算法(dedicated processor assignment)。即给一个作业(线程)指派一台堆垛机(处理器),一旦作业被调度,他将一直占用这个堆垛机运行直到整个应用运行结束。这一算法是通过高速并行来达到最快的执行速度,它在整个系统的调度中避免了低级调度和切换。算法不考虑单个堆垛机使用效率,而集中关注提高总体的计算效率。

环形穿梭车虽然只起到中转的作用,但是在实际调度中应该考虑穿梭车所耗费的时间。考虑到环形穿梭车所在位置的准确性,把各个下货口点和入库输送机入口点称为稳定状态点,即与各位位置距离一定可计算。图 1 所示 AS/RS 中在整个环穿轨道上只有在各个稳定状态点才允许为载货环形穿梭车分配堆垛机。

4.1.2 组织结构与资源共享

在一个固定的自动化立体仓库中堆垛机的数量是有限的,堆垛机的运行速度也有最高极限,为了使整个立体仓库的入库效率最优,为堆垛机(CPU)设置了二级缓存。其结构如图 2 所示。其中一级缓存属于私有的,即一旦作业进入某一巷道的缓冲区,该线程的所有资源属于该堆垛机私有。每组缓冲输送机对应一台堆垛机,缓冲输送机上可能有多个作业,将其定义为缓冲队列。缓冲队列中的优先级对应缓冲输送机上货物的顺序,靠近货架处的作业在缓冲队列中对应的优先级最高。

模型所描述的二级缓存在 AS/RS 中对应环形

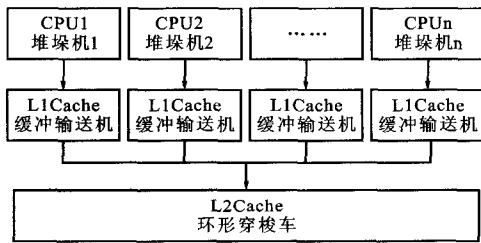


图2 逻辑结构图

穿梭车,在没有承载货物或者载货但没有分配给某个固定堆垛机之前所有资源属于共享资源,环形穿梭车的数量即为该缓存区的最大数量,一旦为作业选定了特定的堆垛机,该作业以及其所有资源将不再属于共享资源。考虑到在为每个载货的环形穿梭车选择堆垛机时需要计算堆垛机作业运行的预期时间,而如果多个环形穿梭车同时需要为其分配堆垛机将会造成数据冗余与时间计算不准确,因此,我们采用共享互斥锁的概念,即每台载货车任意时刻都可以读取与计算所有堆垛机的预期运行时间,但是在读取所有数据时为整体 Cache2 加上共享锁,一旦需要为某个载货环形穿梭车分配堆垛机时将为 Cache2 加上互斥锁也称为排他锁,此时其他载货环形穿梭车只能沿轨道继续前行,到另一稳定的可分配点判断有无互斥锁才可以为其分配堆垛机。

4.1.3 调度层次

当货物到达下货口调度环形穿梭车接载货物后创建线程(CreateThread)将线程置于初始状态,在调度系统中称为作业的未执行状态;当每一个线程到达入库调度稳定状态点时为线程分配 CPU (OpenThread);穿梭车将货物放到入库输送机上如果已有缓存则将线程置于挂起状态,调度系统称其为作业的挂起状态;如果缓冲区空闲但是 CPU 不空闲时将线程置于等待状态,调度系统称其为作业的未执行状态;当 CPU 空闲时在缓存队列中唤醒优先级最高的线程将其置于运行状态,调度系统称其为作业的正执行状态;当堆垛机将货物成功入库且完成后线程结束,调度系统称其为作业的完成状态。线程状态在调度系统中的表现形式如图3所示。

4.2 变量定义

为了模型的简洁,不考虑堆垛机伸叉收叉的时间,为了模型结构化,文中做如下定义:

(1) AS/RS 中有 N 台堆垛机,即对应 N 个 CPU, N 个入库输送机入库点,分别称为 $N1$ 、 $N2$... NN 。 M 个下货输送机下货点,分别称为 $M1$ 、 $M2$ 、... MM 。

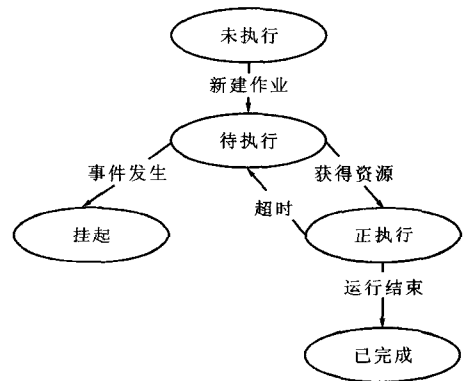


图3 作业状态图

(2) 每组缓冲输送机最多可存放 C 盘货物,即一级缓存的缓冲队列的长度最大为 C ,实际缓冲队列中值得个数定义为 C_{buffer} 。

(3) 堆垛机运行的平均速度定义为 V_{ddj} ,环形穿梭车运行平均速度定义为 V_{hxc} 。

(4) 两巷道之间距离定义为 $D_{distanc}$,环穿轨道一周长度为 D_{circle} 。

(5) 图1所示该系统中共有 $M+N$ 个稳定状态点,固定即只有环形穿梭车运行到这些点时才能为其分配堆垛机。因为所有下货点对于所有的入库口点是均等的,即距离差仅是各入库口之间的距离,因此可以动态的设所有的 M 个点到的距离均为 $D_{regular}$ 。

(6) 单位货格长度为 D_{grid} 。

(7) 由货位分配算法得到第 $Line$ 个作业所对应的货位为该巷道的第 L 列,记作 L_{line} , $line \in (1, 2, \dots, C_{buffer})$ 。定义队列 Q 存放上一作业货位的列值, Q_{end} 为队尾值,即上一货位对应列,队列最大长度为 $C+1$ 。

(8) 完成一个缓冲队列中单个作业程所需时间定义为 T :

$$T = \frac{D_{grid} \times (L_{line} + Q_{end})}{V_{ddj}}$$

(9) 完成一个缓冲队列中所有作业所需时间定义为 A_i :

$$A_i = \sum_{i=1}^{i \leq C_{buffer}} \frac{D_{grid} \times (L_{line} + Q_{end})}{V_{ddj}}$$

$$i \in (1, 2, \dots, C_{buffer})$$

(10) 环形穿梭车的有效运行时间(载货分配堆垛机后运行时间)的计算可分为呈现为分段函数,下货口点和入库输送机口点两段,定义为 T_{hxcx} 。

(11) 每个堆垛机运行所有作业所需时间和环形穿梭车从稳定状态点运行到对应入库输送机所需

时间之和称为堆垛机运行的带权时间,定义为 S_i .

4.3 模型确定

4.3.1 模型原则

(1) 当存在空闲堆垛机时,为一个作业选择一个堆垛机.即当有空闲堆垛机时优先选择空闲的堆垛机运行待执行作业.

(2) 当没有空闲堆垛机时,为一个作业选择一个堆垛机.即当所有堆垛机都未处于空闲状态时,根据计算利用整体最优策略选择一台堆垛机,将对应作业插入到该堆垛机所相应的私有缓存队列中.

(3) 选择一作业让它在指定的堆垛机上运行.即根据一定原则唤醒某个作业,使其在堆垛机上执行.

4.3.2 决策函数

在 Windows 内核的多 CPU 多线程调度理论中,在 CPU 数量(NumCPUs)固定的情况下,线程的数量(NumThreads)严重制约着系统的调度效率,因此文中所设计的调度策略中作业的数目也不会无限增多^[8].当达到一定比例即使下货口有货环形穿梭车也不会去接货,即不会产生新的未执行作业.我们称其为作业饱和状态,直到原有作业中某一个或多个执行完毕,释放资源后才会产生新的未执行作业.也就是说,作业数最好等于堆垛机数与作业任务要花费在实际工作上的时间百分比的比率,其中 BP 是任务将阻塞的时间百分比.

NumTreads=NumCPUs/(1-BP) (1)

在 Windows 内核的多 CPU 多线程调度理论中,共享缓冲区的竞争调度是多处理器调度的瓶颈问题,同样环形穿梭车的调度一直是大型自动化立体仓库入库调度的瓶颈问题,环形穿梭车的有效运行时间 T_{hcx} 的计算是最棘手也是最关键的,结合上述理论可以得到 T_{hcx} 的计算方法为:

$$T_{hcx} = \begin{cases} \frac{D_{regular} + (j-1) \times D_{distance}}{V_{hcx}} \\ j \in (N_1, N_2, \dots, N_n) \\ hcx \in (N_1, N_2, \dots, N_n) \\ T_N = \begin{cases} \frac{(j-1) \times distance}{V_{hcx}} \\ hcx \in (N_1, \dots, N_j) \\ \frac{D_{circle} - (N-j+1) \times D_{distance}}{V_{hcx}} \\ hcx \in (N_j, \dots, N_n) \end{cases} \end{cases} \quad (2)$$

为一个未执行作业选择堆垛机的原则决定于 S_i 的大小.即比较所有堆垛机的带权执行时间,最

小者将被选中.

$$S_j = T_{hcx} + A_i$$
$$j \in (1, 2, \dots, N), i \in (1, 2, \dots, C_{butter}) \quad (3)$$

5 调度策略

综合上述多处理器多线程的调度策略如下:

Step1: 根据决策函数判断是否可以继续产生未执行作业.

Step2: 如果 Step1 被允许则在下货输送机口环形穿梭车接上货物后产生未执行作业.

Step3: 如果堆垛机有空闲则直接将空闲堆垛机分配给该作业直接执行 Step5.

Step4: 环形穿梭车到达各稳定状态点时读取资源信息与互斥标志,如果没有其他作业占用资源的标志,即存在互斥标志,判断 S_i 大小,根据模型原则和决策函数为作业选择堆垛机,如果其他作业已经占用资源,则待到环形穿梭车运行到下一稳定状态点继续读取判断.

Step5: 环形穿梭车将货物送到入库输送机上时再次计算该巷道对应缓冲区的 A_i 值,根据 A_i 值决定该线程的下一个状态.

Step6: 实时获取资源信息,决定作业下一状态.

Step7: 当该作业获得所有必须资源时开始执行,如果在执行过程中被出库任务抢占则作业挂起,继续等待回到 Step6.

Step8: 判断作业是否正常结束,如果结束释放资源进入下一作业的判断.

6 算例分析

某自动化立体仓库有 5 台堆垛机;缓冲区最多可缓存 3 盘货物;巷道之间距离 3m;单位货格宽度 2m;环穿轨道的整体长度为 50m;堆垛机运行的平均速度为 10m/s;环穿小车运行的平均速度为 5m/s;任务阻塞的时间百分比为 50%.当前缓冲区所对应的缓冲队列中已有线程情况如表 1 所示.

表 1 当前作业表

巷道	1 巷道	2 巷道	3 巷道	4 巷道	5 巷道
货位列	20,21	18,19	18	15	10
CPU S_i	17.0	16.0	9.0	8.4	6.6

由式(1)得 NumThreads=5/(1-50%)=10,所以可以继续创建线程,如果下货口输送机有新任务,则环形穿梭车可直接接取,并创建相应线程,当

线程运行到稳定状态点后,由式(2)与式(3) $S_j = T_{hexx} + A_i$ 可分别计算出各 CPU 的带权运行时间,数据如表 2 所示.

表 2 带权时间表

线程数	巷道	1 巷道	2 巷道	3 巷道	4 巷道	5 巷道
		2	2	1	1	1

由此当环形穿梭车载货到达线程调度起点 P 时,系统将通知其将货物直接运送到 5 巷道对应的入库输送机,并且堆垛机会在相对最短的时间内将其执行完成,这将是一条最优执行路径.

7 结束语

文中研究了大型 AS/RS 中调度策略的综合性能要求,提出了一种优化利用所有资源,提高整体入库效率的多处理器多线程的综合调度策略,并给出调度策略的测试用例. 主要成果包括:(1)将 Windows 内核调度理论引入自动化立体仓库调度领域;(2)采用了 Windows 内核多处理器多线程调度中线程数量控制函数;(2)给出了在大型 AS/RS 中为环形穿梭车选择适当入库输送机的函数;(3)提出了综合考虑多种因素的多处理器多线程调度策略,并给出对其详细的分析论证. 该研究成果可应用于对大型自动化立体仓库调度系统效率的优化.

参考文献:

[1] 江勇,吴建平. 多性能目标分组调度策略[J]. 软件学报, 2002,13(8):1514-1520.

[2] Linn R, Wysk R At. An ExPert system based controller for AS/RS[J]. Intemational Journal of Production Research,1990,32(2):176-191.

[3] Kim Byung-In, Heragu Sunderesh S, Graves Robert, et al. A hybrid scheduling and control system aichitecture for warehouse management [J]. IEEE Transactionson Robotics& Automation, 2003,12(6): 6-11.

[4] Krishnaiah Chetty M OV, Sarveswar Reddy. Genetic algorithms for studies on AS/RS integrated with machines[J]. Int J Adv Manuf Technol, 2003(22): 932-940.

[5] Love R. Linux 内核设计与实现[M]. 陈莉君,译. 北京: 机械工业出版社,2004.

[6] 赵树林,张璟,李军怀,等. 基于工作流的办公自动化系统的研究与实现[J]. 微电子学与计算机,2009,26(5):135-141.

[7] 赵佳华,伦立军. 软件体系结构层的耦合性度量研究 [J]. 微电子学与计算机,2009,26(5):103-109.

[8] Russinovich M E, Solomon D A. 深入解析 Windows 操作系统, Microsoft windows server 2003/windows XP/windows 2000 技术内幕[M]. 4 版. 潘爱民,译. 北京:电子工业出版社,2007.

作者简介:

王海宾 男,(1982-),硕士研究生,教师,CCF 会员. 研究方向为信息系统与决策支持、数据库与软件工程技术.

(上接第 141 页)

表 2 三种算法的兼容度

方法	兼容度
算法(19)	0.6006
算法(20)	0.8504
文中算法(18)	0.9585

4 结束语

文中在算法中引入了语义树中有向边的深度、密度和强度对边的权值的影响以及概念语义重合度和概念层次差对语义相似度的直接影响来计算概念间的语义相似度. 并抽取 WordNet2.0 中关于计算机的局部语义树来进行实验,根据具体情况设定了调节因子. 实验结果比较理想,但有些地方仍然不够完善. 进一步的研究方向是考虑影响语义相似度的其它更多因素,并将概念语义相似度的计算应用到有关信息检索应用系统中.

参考文献:

[1] 陈沈焰,吴军华. 基于本体的概念语义相似度计算及其应用[J]. 微电子学计算机,2008,25(12): 96-99.

[2] Agirre, Rigau. A proposal for word sense disambiguation using conceptual distance[C]// International Conference of Recent Advances in Natural Language Processing. Borovets, Bulgaria, 2003.

[3] 王进,陈恩红,施德明,等. 一种基于语义相似度的信息检索方法[J]. 模式识别与人工智能,2006(6):2-6.

[4] 江敏,肖诗斌,王弘蔚,等. 一种改进的基于《知网》的词语语义相似度计算[J]. 中文信息学报,2008,22(5):84-89.

[5] 张选平,蒋宇. 一种基于概念的信息检索查询扩展[J]. 微电子学与计算机,2006,23(4): 110-114.

[6] 夏天. 汉语词语语义相似度计算研究[J]. 计算机工程, 2007,33(6):191-194.

作者简介:

安建成 男,(1964-),副教授. 研究方向为语义网、数据库.

武俊丽 女,(1982-),硕士. 研究方向为数据库.