

一种 Agent 数据库系统框架及其规则并行算法*

张茂元⁺, 卢正鼎

(华中科技大学 计算机科学与技术学院,湖北 武汉 430074)

An Agent-Based Database System Framework and Its Parallel Algorithm of Rules

ZHANG Mao-Yuan⁺, LU Zheng-Ding

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

+ Corresponding author: Phn: +86-27-87544285 ext 0, E-mail: coolphenix@163.com, <http://www.hust.edu.cn>

Received 2003-05-28; Accepted 2004-03-29

Zhang MY, Lu ZD. An Agent-based database system framework and its parallel algorithm of rules. *Journal of Software*, 2004,15(8):1157~1164.

<http://www.jos.org.cn/1000-9825/15/1157.htm>

Abstract: The object-oriented method is passive, so it is not consistent with the active rules and can not easily define the functions and features of the objects in a distributed and active database system. And also the active rules lead to the problems of termination and confluence. In this paper, the Agent method, distributed database, and active database are combined after the constraints of the object-oriented method are analyzed. On the basis of the combination, a distributed and active database system framework based on the Agent-oriented method is proposed. As to the system framework, a method of expanding the event-rule graph and an improved Coffman-Graham parallel algorithm are presented, and then their performances are analyzed. Shown by the analysis results, the former is used to solve the problem of database system termination, while the latter, on the foundation of remaining the confluence, improves the efficiency of parallel rule process. Besides, the database system framework is useful for applying the Agent method to the distributed and active database.

Key words: Agent-orient; active; termination; confluence; parallel

摘要: 面向对象方法描述的对象是被动的,与主动机制存在矛盾,不能很好地定义分布式主动数据库中对象的功能和特性.分布式主动数据库中的主动规则还带来了终止性、合流性的问题.首先分析面向对象方法的局限性,将 Agent 技术、分布式数据库、主动数据库相结合,给出一种面向 Agent 的分布式主动数据库系统框架.然后在这个框架基础上,提出扩展事件规则图方法和改进的 Coffman-Graham 规则并行算法,并分析它们的性能.分析结果表明,前者在一定程度上解决了数据库系统的终止性问题,后者在保持合流性的基础上提高了规则并行处理效率.这个框架对研究 Agent 技术在分布式主动数据库中的应用有一定的启示.

关键词: 面向 Agent;主动性;终止性;合流性;并行

中图法分类号: TP18 文献标识码: A

* 作者简介: 张茂元(1975—),男,广东五华人,博士生,主要研究领域为数据库,人工智能,网络信息系统;卢正鼎(1944—),男,教授,博士生导师,主要研究领域为数据库,软件工程,信息系统.

数据挖掘是指从数据库中提取先前未知的、可用的信息,并将该信息运用于制定重要的管理决策的过程^[1].当前,人们迫切要求一些系统,如外汇交易信息管理系统和网上非法药品广告稽查系统,在每条非法信息存入数据库时就能立即主动检测出来.

伴随着网上信息量的爆增,这种需求推动了分布式主动数据库的更加广泛的应用.Oracle9i^[2]采用相关对象模型来支持分布功能,并引入了较简单的主动机制.一种基于CORBA的分布式主动数据库^[3]也将分布对象技术应用于构建DADBS的设计思想中.

面向对象方法描述的对象都是被动的(passive),与主动机制存在矛盾,还不能很好地定义分布式主动数据库中对象的功能和特性.由于方法的调用对于描述交互的类型又是一个过于简单的机制^[4],而且类和模块不便表达启发式知识,因此面向对象方法还不能提供充分的概念和机制对复杂系统建模.从这些方面来看,面向对象方法与分布式主动数据库的耦合还不够紧密.

主动规则带来了与行为控制有关的问题,如终止性、合流性等^[5].针对终止性问题,触发图(triggering graph)方法^[6]未将规则间影响关系在图中表示出来;规则缩减算法^[7]结合了触发图和激活图,但还存在图的构造问题.而对保证合流性的并发控制问题而言,Lee提出的分布并行处理方法^[8]未考虑冲突的消解,而得到非合流性执行结果;基于事务树的主动规则并发控制方法^[9]将规则看作事务,但数量较多的规则回滚会降低系统并行效率.更为重要的是,其中有些方法还没有考虑终止性和合流性问题的分布性,因此它们还不适用于分布式主动数据库系统.

基于Agent计算模型以其自然、贴切、直观以及对分布式开放环境的良好适应性而受到研究人员的广泛重视^[10].Agent技术已经应用到一些领域,如软件Agent^[11]和移动Agent^[12]的研究,但它还未与分布式主动数据库相结合.本文给出一种面向Agent的分布式主动数据库系统框架,并提出扩展事件规则图EE-RG方法和改进的Coffman-Graham并发算法.EE-RG方法可解决面向Agent分布式主动数据库系统的终止性问题,改进的并行算法在保持合流性的基础上可以有效地提高规则并行处理效率.理论分析和调度结果表明,改进的并行算法具有较优的并行效果.

1 面向 Agent 的分布主动数据库系统框架

Agent技术来源于分布式人工智能(distributed artificial intelligent,简称DAI)领域.它被誉为“软件开发的又一重大突破”、“软件界的革命”^[13].面向Agent方法是在面向对象方法的概念上发展的.由于Agent是建立在人工智能基础上的,能够表达启发式知识,因此面向Agent方法能够较好地表示主动对象.在这一点上,面向Agent比面向对象能更好地与主动数据库相结合.

将面向Agent技术与分布式主动数据库有机地结合,就可以构成面向Agent的分布式主动数据库.如图1所示,数据库系统主要由主动控制Agent、程序接口Agent、通信Agent、DBMS Agent和数据库DB组成.

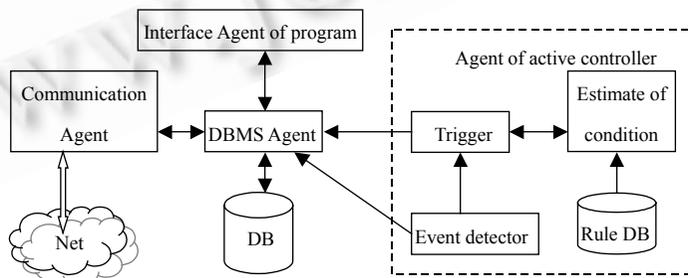


Fig.1 The Agent-oriented distributed and active database system framework

图1 面向 Agent 的分布式主动数据库框架

主动控制 Agent 主要包含触发器、事件探测器、情形评价器和规则库.事件探测器一旦探测到事件发生,就向触发器发出信号.触发器接到信号以后,就调用情形评价器.情形评价器根据所收到信号的内容(所发生的

事件),取出其中情形限制的条件谓词进行评价,将评价结果返回给触发器.最后,触发器根据评价结果决定被触发的活动是否被执行.主动控制 Agent 能主动地监视数据库状态和状态的变化,当一定的状态出现时,能及时地作相应的处理.

程序接口 Agent 为应用程序访问数据库系统提供一个公用标准接口,从而提高数据库的跨平台能力.同时,该 Agent 对程序的数据库操作请求进行控制,从而在一定程度上维护数据库的完整性和一致性.

通信 Agent 与外界交互不是通过一些预先严格确定的接口函数,而是高度抽象的消息(如通知、请求、承诺等).由于 Agent 具有自治性,可根据自己当前的状态来决定是否响应外界传来的消息,因此,Agent 比对象具有更强的封装性.正是这种更强的封装性,才使得 Agent 具有更好的分布性,使得 Agent 比对象更适用于分布式主动数据库.通信 Agent 还提供网络认证安全机制进行网络访问的智能控制,同时,将非法访问信息主动发给 DBMS Agent.

DBMS Agent 为 DB 的访问提供一个公用标准接口,从而使数据库系统访问不同结构类型的 DB.它通过消息与其他 Agent 交互,并用人工智能方法对整个数据库系统进行智能控制.

面向 Agent 的分布式主动数据库系统的数据是分布式的,计算过程是异步、并发和并行的.它不仅具有自治性、主动性、适应性和异构分布性,还有基于知识的问题求解思路.这些特性有利于系统之间的复杂合作与通信,也使得系统具有更好的应用前景.

2 部分核心算法

终止性(termination)是指规则处理过程能否结束;合流性(confluence)是指规则处理结束时数据库是否处于唯一确定的终止状态.扩展事件规则图 EE-RG 方法和改进的 Coffman-Graham 并发算法分别解决面向 Agent 分布式主动数据库系统的终止性和合流性问题.

2.1 扩展事件规则图 EE-RG 方法

面向 Agent 分布式主动数据库系统与主动数据库系统有些区别.分布在不同地方的各个数据库系统可以通过通信 Agent 互相传递触发信息,具有分布性.因此,规则不仅可以触发本地数据库中的规则,也可以触发外地数据库中的规则;另一方面,规则既可以被本地数据库中的规则所触发,也可以被外地数据库中的规则所触发.

为了提高分析的准确性,提出扩展事件规则图 EE-RG(expanding event rule graph)方法.如图 2 所示,该方法基于规则执行的时间关系,引入依次关系 S(sequence)、同步关系 Y(synchronization)、并发关系 P(parallel)和迁延关系 D(deferment).上述关系的含义和 EE-RG 的构建算法如下:

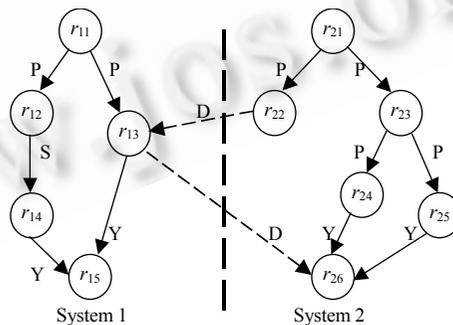


Fig.2 The expanding event rule graph

图 2 扩展事件规则图

(1) 设 r_i 和 r_j 分别为两个数据库系统中的规则.若规则 r_i 触发规则 r_j ,即 r_i 的直接后继是 r_j ,则称规则 r_i 迁延 r_j ,标记为 $r_i D r_j$.

(2) 设有规则集 $\{r_1, r_2, \dots, r_n\}$ ($n > 1$) 和规则 r .若规则集中任意一个元素 r_i 的直接后继是 r ,且它们不是迁延关系,则称规则集 $\{r_1, r_2, \dots, r_n\}$ 同步于 r ,标记为 $\{r_1, r_2, \dots, r_n\} Y r$.

(3) 设有规则集 $\{r_1, r_2, \dots, r_n\} (n > 1)$ 和规则 r . 若规则集中任意一个元素 r_i 是 r 的直接后继, 且 r_i 有唯一的直接前继 r , 同时它们不是迁延关系或同步关系, 则称规则集 $\{r_1, r_2, \dots, r_n\}$ 在 r 后是并行的, 标记为 $r P \{r_1, r_2, \dots, r_n\}$.

(4) 设规则 r_i 的直接后继是 r_j , 若它们既不是迁延关系, 也不是同步关系或并发关系, 则称规则 r_i 依次 r_j , 标记为 $r_i S r_j$.

2.2 扩展事件规则图方法的终止性分析

EE-RG 方法由最基本的时序关系引入规则图的控制结构(S, Y, P 和 D 关系), 将 E-C-A 规则按时间语义关系组成事件规则图. 这种方法是从时序角度上理解规则行为, 可以确定规则间的时序关系. 当一个规则 x 触发另一个规则 y 时, EE-RG 方法可以通过这两个规则的时序关系来判断被触发规则 y 是否已经触发过. 若在时序关系上规则 y 早于规则 x , 则表明规则 y 已被触发过, 就不触发规则 y ; 否则表明规则 y 还未被触发过, 就触发规则 y . 因此, EE-RG 方法在一定程度上可以避免规则相互触发, 从而在一定程度上保证了面向 Agent 分布式主动数据库系统的终止性.

EE-RG 方法在一定程度上体现了规则图的局部并发性, 有利于规则执行的并发研究. 而且这个方法引入迁延 D 关系, 增强了系统间协调的灵活性.

2.3 改进的Coffman-Graham并行算法

Coffman-Graham 并行算法是一个高效的并行调度算法, 但它局限在所有规则执行的时间量相等的简单情况. 在面向 Agent 分布式主动数据库系统中, 一方面, 系统内的规则执行时间可能各不相等; 另一方面, 系统之间触发的规则执行时间容易受环境因数(如网络延迟)影响而变得不确定, 并且本系统对外系统的规则执行时间一般不清楚. 因此, 在具有时序关系的 EE-RG 规则图基础上, 本文给出改进的 Coffman-Graham 规则并行算法, 以提高面向 Agent 分布式主动数据库系统的规则处理效率.

为了表达系统间触发的规则执行时间的不确定性, 迁延预处理算法构造虚规则来替换与本系统有迁延关系的外系统规则. 其算法是: (1) 寻找本系统中具有迁延关系的所有规则; (2) 处理这些规则, 若已处理完, 则跳至步骤(5), 否则任取一个, 设为规则 r_i ; (3) 构造一个执行时间为 t_{max} 的虚规则; (4) 建立规则 r_i 与虚规则的连接, 并标记为迁延 D 关系, 转步骤(2); (5) 处理结束.

如图 3 所示, 图 2 中的系统 2 的规则时序图用迁延预处理算法可以得到等效的单系统规则时序图, 其中规则 r_{27} 和 r_{28} 是虚规则.

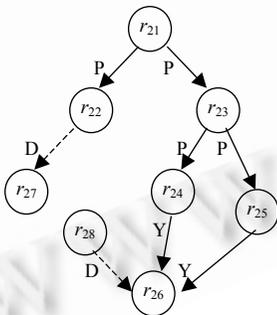


Fig.3 The expanding event rule graph of system2

图 3 系统 2 的扩展事件规则图

设 $T = \{T_1, T_2, \dots, T_n\}$ 是系统有待执行的规则任务集合(T_i 执行时间可以不相同), $<$ 是 T 上的偏序关系(前趋关系). $S(T_i) = \{T_k | T_i < T_k\}$ 表示 T_i 的直接后继的集合. $\alpha(T_i)$ 是赋给 T_i 的一个整数标号, $N_{EC}(T_i)$ 是 T_i 的执行时间加其直接后继最大的 $N_{EC}(T_j)$.

改进的 Coffman-Graham 并行算法:

(1) 把没有被标号且没有后继的所有规则任务, 组成集合 R ;

(2) for $i \leftarrow 1$ to n do

a. 对 R 中所有的 T_j , 计算 $N_{EC}(T_j)$;

b. 取 $N_{EC}(T_j)$ 中最小值的规则任务;

c. 若这样的规则任务有多个, 则采用随机策略选取一个, 并设该规则任务为 T_* ;

d. $\alpha(T_*) \leftarrow i$;

e. 在 T_* 的直接前继规则集中, 把没有被标号且后继均已被标号的规则任务添至 R ;

end for

(3) 构造 $L' = (u_n, u_{n-1}, \dots, u_2, u_1)$, 使得 $\alpha(u_i) = i (i = 1, 2, \dots, n)$;

(4) 去掉 L' 中的虚规则任务, 从而构造 L ;

(5) 给定 (T, \prec, L) , 用 Graham 表调度算法, 调度 L 中的规则任务.

其中, Graham 表调度是: 只要处理机没有工作做, 它就就从就绪任务表 L 中移动规则任务. 所移规则任务还未被调度, 且在 \prec 下的所有前趋任务均已完成执行. 如果所移规则为外系统所触发, 则须等待外系统触发消息的到达. 如果有两个或多个处理机试图执行同一任务, 则具有最低编号的处理机是成功的, 其他处理机另外寻找合适的任务.

2.4 改进的Coffman-Graham算法的并行效果分析

$N_C(T_i)$ 定义为其直接后继最大的 $N_C(T_j)$ 加 T_j 的执行时间; $N_{EC}(T_i)$ 定义为 T_i 的执行时间加其直接后继最大的 $N_{EC}(T_j)$. Coffman-Graham 算法用 $N_C(T_i)$ 计算, 而改进的 Coffman-Graham 算法用 $N_{EC}(T_i)$ 计算.

定理 1. 在无迁延关系的规则时序图中, 改进的 Coffman-Graham 算法与 Coffman-Graham 算法具有等效的并行效果.

证明: 由于 Coffman-Graham 并行算法只适合规则执行时间相等的情况, 所以, 为了比较两种算法的并行效果, 将任意一个规则时序图转换成等效的且规则执行时间相等的时序图. 如图 4 所示, 对规则时序图中任意两个规则任务 T_i 和 T_j , 若其执行时间分别为 m 和 n , 分别用 m 和 n 个执行时间为单位时间的串行规则序列替换规则 T_i 和 T_j .

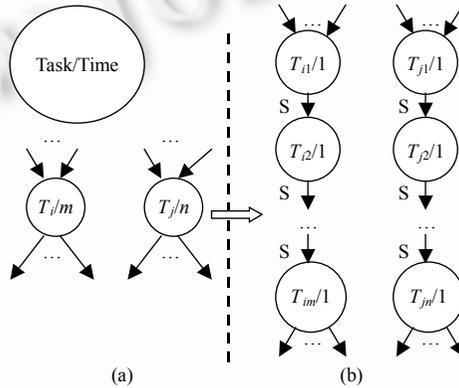


Fig.4 The time-unity transformation of a rule graph

图 4 一个规则图的时间单位化转换图

用改进的 Coffman-Graham 算法和 Coffman-Graham 算法分别对原图和转换图进行调度, 得到调度表 L_{EC} 和 L_C . $L_C(T_i)$ 用来表示 T_i 在调度表 L_C 中的调度位置.

(1) 由 $N_{EC}(T_{i1})=N_{EC}(T_{im})+(m-1)$ 及 $N_{EC}(T_{im})=N_C(T_{im})+1$, 可得

$$N_{EC}(T_{i1})=N_C(T_{im})+m \tag{1}$$

由 $N_{EC}(T_i)=N_C(T_i)+m$ 及 $N_C(T_i)=N_C(T_{im})$, 可得

$$N_{EC}(T_i)=N_C(T_{im})+m \tag{2}$$

联合式(1)和式(2), 得

$$N_{EC}(T_i)=N_{EC}(T_{i1}) \tag{3}$$

同样可得

$$N_{EC}(T_j)=N_{EC}(T_{j1}) \tag{4}$$

(2) 若 $L_C(T_{i1}) < L_C(T_{j1})$, 则 $N_C(T_{i1}) > N_C(T_{j1})$, 可得 $N_{EC}(T_{i2}) > N_{EC}(T_{j2})$.

因此 $N_{EC}(T_{i2})+1 > N_{EC}(T_{j2})+1$, 即

$$N_{EC}(T_{i1}) > N_{EC}(T_{j1}) \tag{5}$$

将式(3)和式(4)代入, 得 $N_{EC}(T_i) > N_{EC}(T_j)$.

由此可得 $L_{EC}(T_i) < L_{EC}(T_j)$.

同样, 若 $L_C(T_{i1}) > L_C(T_{j1})$, 也可推出 $L_{EC}(T_i) > L_{EC}(T_j)$. 因此, 调度表 L_C 中的 T_{i1} 和 T_{j1} 的前后位置关系与调度表

L_{EC} 中的 T_i 和 T_j 的前后位置关系是一致的.又因为原图中 T_i 和 T_j 分别对应于转换图中 T_{i1} 和 T_{j1} ,所以两种调度算法的调度效果是一致的. □

推论 1. 在有迁延关系的规则时序图中,改进的 Coffman-Graham 算法与 Coffman-Graham 算法具有等效的并行效果.

证明:由于迁延关系,规则时序图可能有多个终结点(如虚规则).这样,规则时间单元化的转换图就不能用 Coffman-Graham 算法,所以要将有迁延关系的规则时序图转化成只有 1 个终结点的规则时序图.如图 5 所示,添加一个执行时间为 1 的规则任务 T_e 作为新的终结点,使原来的终结点都与它同步,并将迁延关系依情况转换成同步、并行或依次关系中的一种.

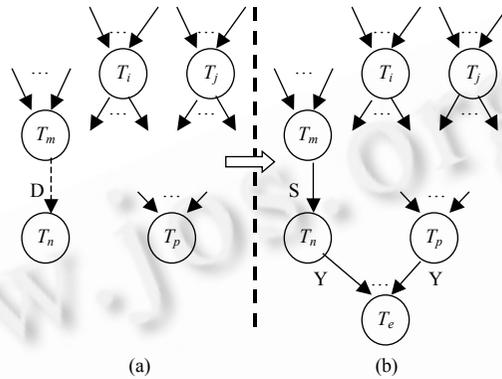


Fig.5 The one-ending-nod transformation of a rule graph

图 5 一个规则图的终结点归一化转换图

用改进的 Coffman-Graham 算法分别对原图和转换图进行调度,得到调度表 $L1_{EC}$ 和 $L2_{EC}$. $N1_{EC}(T_i)$ 表示对原图计算的 $N_{EC}(T_i)$, $N2_{EC}(T_i)$ 表示对转换图计算的 $N_{EC}(T_i)$.

$$\forall T_i, T_j \in \{G_a \cap G_b\}, \text{ 则 } T_i \neq T_e, T_j \neq T_e.$$

$$\text{如果 } L2_{EC}(T_i) < L2_{EC}(T_j), \text{ 则 } N2_{EC}(T_i) > N2_{EC}(T_j).$$

$$\text{又 } N2_{EC}(T_i) = N1_{EC}(T_i) + 1, N2_{EC}(T_j) = N1_{EC}(T_j) + 1.$$

$$\text{所以 } N1_{EC}(T_i) > N1_{EC}(T_j).$$

$$\text{由此可得 } L1_{EC}(T_i) < L1_{EC}(T_j).$$

这就表明,在对原图和终结点归一转换图用改进的 Coffman-Graham 算法得到的两个调度表中, T_i 和 T_j 调度前后位置关系是一致的.因此,改进的 Coffman-Graham 算法对两图的调度效果是一致的.

终结点归一转换图是一个无迁延关系的规则时序图,因此由定理 1 可知,改进的 Coffman-Graham 算法对终结点归一转换图的调度效果与 Coffman-Graham 算法对时间单位化转换图的调度效果是一致的.综上可知,改进的 Coffman-Graham 算法对有迁延关系的原图的调度效果与 Coffman-Graham 算法对时间单位化转换图的调度效果是一致的. □

引理 1^[14]. 当处理机数 $p=2$ 时, Coffman-Graham 算法是最优调度.

推论 2. 当处理机数 $p=2$ 时,改进的 Coffman-Graham 算法是最优调度.

证明:由推论 1 和引理 1 可以推出,当处理机数 $p=2$ 时,改进的 Coffman-Graham 算法是最优调度.

改进的 Coffman-Graham 规则并行算法,其实质是在分析各规则执行的时序关系基础上,并行处理无时序制约的多个规则.这个方法保持各规则执行的时序关系,从而使结果是正确而又唯一的.这就使得系统既保持流动性,又提高了规则处理的效率.

2.5 改进的Coffman-Graham算法的调度例子分析

如图 6 所示的两个规则图,用改进的 Coffman-Graham 算法分别进行并行规则处理.其中图 6(b)中 T_9 和 T_{10} 是虚规则,其执行时间设定为最大值 $A=t_{max}$,并假定外系统对规则 T_7 能及时触发.调度结果如图 7 所示.

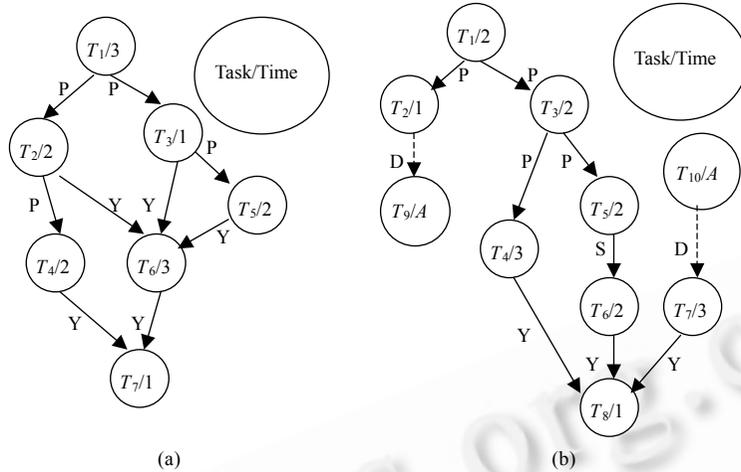


Fig. 6 Non-Deferment rule graph and deferment rule graph

图 6 无迁延关系规则图和有迁延关系的规则图

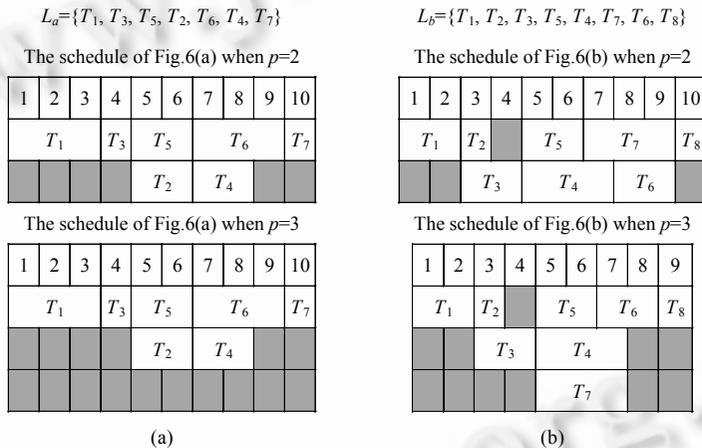


Fig. 7 The scheduling results of improved Coffman-Graham algorithm

图 7 改进 Coffman-Graham 算法的调度结果

给定了一个任务图和处理机数,最优调度是使得总的执行时间最短.最优调度所需时间应最接近 $t_o = t_s +$

$\left[\left(\sum_{i=1}^n t_i \right) - t_s \right] / p$, 其中 t_s 是必须串行的操作时间, p 是处理机个数.对于图 6(a)而言,总操作时间是

$\sum_{i=1}^n t_i = 14, t_s = t_1 + t_7 = 4$.对于图 6(b)而言,总操作时间是 $\sum_{i=1}^n t_i = 16, t_s = t_1 + t_8 = 3$.

(1) 对图 6(a)的调度结果分析

当 $p=2$ 时, $t_o = 4 + 10/2 = 9$.改进的 Coffman-Graham 算法的调度时间 $t_{p2} = 10$, 则 $(t_{p2} - t_o) = 1$.尽管 T_2 和 T_3 没有同时运行,但其最后调度效果相同,其总运行时间最接近 t_o .可见,对于这个例子,当 $p=2$ 时,改进的 Coffman-Graham 算法是最优调度.

当 $p=3$ 时, $t_o = 4 + 10/3 = 7.33$.改进的 Coffman-Graham 算法的调度时间 $t_{p3} = 10$, 则 $(t_{p3} - t_o) = 2.67$.可见,对于这个例子,当 $p=3$ 时,改进的 Coffman-Graham 算法不是最优调度.

(2) 对图 6(b)的调度结果分析

当 $p=2$ 时, $t_o = 3 + 13/2 = 9.5$,改进的 Coffman-Graham 算法的调度时间 $t_{p2} = 10$, 则 $(t_{p2} - t_o) = 0.5$.其总运行时间最接

近 t_0 , 所以对于这个例子, 当 $p=2$ 时, 改进的 Coffman-Graham 算法是最优调度.

当 $p=3$ 时, $t_0=3+13/3=7.33$, 改进的 Coffman-Graham 算法的调度时间为 $t_{p3}=9$, 则 $(t_{p3}-t_0)=1.67$. 可见, 对于这个例子, 当 $p=3$ 时, 改进的 Coffman-Graham 算法是较好的调度, 但不是最优调度.

按照上述分析方法, 对大量的例子进行调度模拟, 同样得到当 $p=2$ 时改进的 Coffman-Graham 算法是最优调度, 而当 $p=3$ 时不能保证改进的 Coffman-Graham 算法是最优调度的结论.

3 结束语

面向 Agent 是在面向对象的概念上发展的, 具有主动性和更强的封装性, 因而更适合与分布式主动数据库结合. 扩展事件规则图方法和改进的 Coffman-Graham 规则并行算法确实是解决面向 Agent 的分布式主动数据库的终止性和合流性问题的有效方法之一. 面向 Agent 的分布式主动数据库是数据库技术的一个未来发展模式, 它的通信机制等其他问题还有待研究.

References:

- [1] Karim KH. Discovering data mining: From concept to implementation. ACM SIGKDD Explorations Newsletter, 1999,1(1):44~45.
- [2] Lenore ML. Oracle9i Database Concepts. Redwood: Oracle Corporation, 2001. 21~24.
- [3] Wang GC, Pan SL, Wang SK. A CORBA based distributed active database system. Journal of Computer Research and Development, 2001,38(9):1092~1096 (in Chinese with English abstract).
- [4] Nicholas RJ. An Agent-based approach for building complex software system. Communications of the ACM, 2001,44(4):35~41.
- [5] Alexander A, Jennifer W, Joseph MH. Behavior of database production rules: termination, confluence, and observable determinism. In: Michael S, ed. Proc. of the 1992 ACM SIGMOD Int'l Conf. on Management of Data. San Diego: SIGMOD Record, 1992,21(2):59~68.
- [6] Anca V, Stella G, Klaus RD. Investigating termination in active database systems with expressive rule languages. In: Andreas G, Mikael B, eds. Proc. of the 3rd Int'l Conf. on Rules in Database Systems. London: Springer-Verlag, 1997. 149~164.
- [7] Zhou ZK, Luo H. An algorithm of the refined rules for active rule termination analysis. Journal of Beijing Institute of Technology, 2001,21(4):485~489 (in Chinese with English abstract).
- [8] Minsoo L, Stanley YW, Herman L. Parallel rule processing in a distributed object environment. Technical Report, TR99-014, Gainesville: Database Systems R&D Center, University of Florida, 1999. 410~416.
- [9] Zuo WL, Jin Y, He FL. Concurrency control and deadlock recover of active rules. Mini-Micro System, 2002,23(10):1181~1184 (in Chinese with English abstract).
- [10] Mao XJ, Wang HM, Chen HW, Liu FQ. A non-terminating active computing model in multi-Agent systems. Journal of Computer Research and Development, 1999,36(7):669~775 (in Chinese with English abstract).
- [11] Hyacinth SN, Divine TN. A perspective on software Agents research. Knowledge Engineering Review, 1999,14(2):125~142.
- [12] Fritz H. A framework to protect mobile Agents by using reference states. In: Wu CL, Hruhisa I, eds. 20th IEEE Int'l Conf. on Distributed Computing Systems. Taipei: IEEE Computer Society, 2000. 410~417.
- [13] Wooldridge M, Jennings NR. Intelligent Agents: Theory and practice. Knowledge Engineering Review, 1995,10(2):115~152.
- [14] Quinn MJ. Parallel Computing: Theory and Practice. 2nd ed., New York: McGraw Hill Book Company, 1994.

附中文参考文献:

- [3] 王广昌, 潘善亮, 王申康. 一个基于 CORBA 的分布式主动数据库系统. 计算机研究与发展, 2001,38(9):1092~1096.
- [7] 周志远, 罗红. 主动规则可终止性分析的规则缩减算法. 北京理工大学学报, 2001,21(4):485~489.
- [9] 左万利, 金阳, 赫枫龄. 主动规则的并发控制与死锁处理. 小型微型计算机系统, 2002,23(10):1181~1184.
- [10] 毛新军, 王怀民, 陈火旺, 刘凤歧. 多 Agent 系统的非终止主动合作计算模型. 计算机研究与发展, 1999,36(7):669~775.