嵌入式数据库系统的事务调度*

刘云生[1], 夏家莉[1,2], 许贵平[1]

[1](华中科学技术大学 计算机科学与技术学院,湖北 武汉 430074);
[2](江西财经大学 会计学院,江西 南昌 330013)
E-mail: xiajl65824@263.net
http://www.hust.edu.cn

摘要: 针对嵌入式数据库系统的实时性和高可预见性,提出了基于功能替代的事务模型.该模型改善了实时事务对动态实时环境的应变能力.由于受功能替代性的影响,事务调度分为内部调度和外部调度,提高了系统的成功率.研究了实时事务的可调度性分析,并给出了相应的内部调度策略,最后作出模拟性能分析.

关 键 词: 数据库系统;嵌入式系统;事务调度
中图法分类号: TP311 文献标识码: A

嵌入式数据库系统(EDBS)通常作为软件部件镶嵌于设备中或实时应用环境中,故一般也是实时数据库系统.另外,系统常常在无人工干预情况下运行,所以 EDBS 的事务调度不仅需要较高的成功率,而且应具有较强的预见能力.目前的嵌入式数据库系统,如 Empress RDBMS,ObjectivityDB,c-tree Plus 等,在模型和并发控制机制上并无新意[1].它们沿用了普通数据库系统的一些控制机制,或部加以分改造[2],没有解决系统成功率低和预见能力弱的问题.

为了在实时动态环境下提高 EDBS 的成功率,我们提出了支持功能替代性的事务模型以及相应的调度策略,其意义体现在两个方面:(1) 嵌入式系统必须具备可预见能力,在尽可能无人工干预下透明地运行[1],要求实时事务应具备较强的对实时环境的应变能力,功能替代为此提供语义支持;(2) 以替代作为并发控制和调度的基本单位可以提高事务的成功率.

1 嵌入式实时事务模型与特性

一个具有时间限制的应用为一个实时事务,它由若干任务(或事务步)组成,为适应嵌入式环境高应变能力和高可靠性的要求,每个任务又由若干功能等价的子事务组成,在每个任务(事务步)中取一个且只取一个子事务所组成的集合称为该事务的一个“替代集”,一个替代集中各子事务按相应任务在事务中的顺序执行,则是该事务的一个“替代”[3].

定义 **1**. 设 $TS=\{TK_i|1\le i\le n\}$为事务 T 的任务集,称:

① $SUB(TK_i)=\{st_{ij}|\forall j\ne k(st_{ij}\xleftrightarrow{F}st_{ik}),1\le j,k\le m_i\}(i=1,2,\ldots,n)$为任务 TK_i 的功能等价子事务集;

② $fx=\{st_{ij}|\forall TK_i\in TS(\exists|st_{ij}\in SUB(TK_i)),1\le i\le n,1\le j\le m_i\}$为 T 的一个功能替代集;

③ 四元组 $FT=\langle fx,FR,FC,<_t\rangle$为 T 的一个替代.FR,FC 和$<_t$分别为关于 fx 的资源需求集、约束集和时序.

其中,$\xleftrightarrow{F}$表示“功能等价”,$\exists|$表示“存在一个也只存在一个”.

* 收稿日期: 2001-04-18; 修改日期: 2001-08-16
基金项目: 国家自然科学基金资助项目(60073045);国家“十五”国防预研基金资助项目(00j15.3.3jw0529)

作者简介: 刘云生(1940－),男,湖南衡阳人,教授,博士生导师,主要研究领域为现代数据库理论与技术及其集成实现,数据库与信息系统开发及实时应用,软件方法学与支撑环境;夏家莉(1965－),女,湖北洪湖人,博士生,副教授,主要研究领域为主动、实时、嵌入式、内存数据库,实现技术数据库理论;许贵平(1966－),男,湖北云梦人,博士生,讲师,主要研究领域为实时数据库,嵌入式实时系统.

现在,可以给出嵌入式实时事务的形式定义.

定义 **2**. 一个实时事务是一个四元组 $T::=(FX,RS,CS,<_T)$.

$FX=\{\bigcup_i fx_i \mid fx_i$ 为 T 的一个替代,$1\le i\le n\}$;

$RS=\{\bigcup_i FR_i \mid FR_i$ 为执行 fx_i 所要求的资源,$1\le i\le n\}$;

$CS=\{\bigcup_i FC_i \mid FC_i$ 为执行 fx_i 所满足的约束,$1\le i\le n\}$;

$<_T\supseteq\{\bigcup_i <_t^i \mid <_t^i$ 为执行 fx_i 时,各子事务间的时序,$1\le i\le n\}$.

实时事务 T 任一替代的成功意味着 T 的成功,即:通过任何一个 fx_i 按子时序$<_t^i$执行来实现,因此,实时事务处理的根本问题就是如何选择一个 fx_i 在满足 FR_i 和 FC_i 的前提下按$<_t^i$执行并使其成功.替代成为事务调度和并发控制的基本单位.

根据以上定义,嵌入式实时事务具有如下一些新的特性:

① 功能等价性.功能等价性表示若实时事务 T 的任何一个替代成功执行,则 T 成功,一个替代夭折不代表该事务夭折,只有当全部替代都失败或超过(或必定会超过)截止期时,该事务才夭折.由上述定义直接可得出下面的定理.

定理 **1**. 同属于一个实时事务 T 的所有替代都是功能等价的,等价于该实时事务.

② 结构同构性.由于实时事务 T 的任意一个替代都是由 T 的各任务步中选取一个子事务而组成,因此,这些替代是同构的.

定理 **2**. 一个实时事务的所有替代在结构上是同构的,都同构于该事务.

证明:对于实时事务 $T=\langle TS,R,<_t,C\rangle$,首先定义一个代数系统:$V1=\langle AA,<_t\rangle$,$AA=TS+R+C$,其含义是带有资源请求 R 和约束 C 的任务集,任务集间的时序为$<_t$,$V1$ 实际上是实时事务 T 的空间.

在该空间上定义函数 h:$h(TK_i+R_i+C_i)=st_{ir}+R_{ir}+C_{ir}$.

定义代数系统:$V2=\langle BB,<_t\rangle$,$BB=ST+SR+SC$.

$$ST=st_{ir},SR=R_{ir},SC=C_i(i=1,2,\ldots,n,r=1,2,\ldots,m_i).$$

因为从各任务中选取的子事务的执行依然保持任务间的时序,故有

$$h(TK_i+R_i+C_i<_t TK_j+R_j+C_j)=h(st_{i1}+R_{i1}+C_{i1})<_t h(st_{j1}+R_{j1}+C_{j1}),$$

即 $h(AA_i<_t AA_j)=h(AA_i)<_t h(AA_j)$,且 $h(AA)=BB$.所以,$V1$ 与 $V2$ 是同态.

因为 $TK_i+R_i+C_i\ne TK_j+R_j+C_j$ 时,$h(st_{i1}+R_{i1}+C_i)\ne h(st_{j1}+R_{j1}+C_j)$,所以 h 是双射,于是 $V1$ 与 $V2$ 同构.

由定义 1 可知,$V2$ 构成 T 的一个替代 fx_r,$V1$ 与 $V2$ 是同构,意味着实时事务 T 与其中一个替代 fx_r 同构.再由于 r 的任意性,得到不同的替代都与实时事务 T 同构,由此,定理得证. □

③ 性能差异性.由于组成替代的子事务不同,它们在执行时间、占用的系统资源等方面一定存在差异,因此不同的替代之间存在着性能差异.

2 嵌入式实时事务的可调度性分析

2.1 实时事务预分析

实时事务预分析在传统的预分析之中增加了实时事务分解,其步骤为:

① 静态分析:实时事务 T 经过预编译后,所有的 DML 语句都变为函数调用,同时,为了提高嵌入式数据库系统的事务处理效率,系统采用内存数据库技术,保证在事务处理过程中没有 I/O(或将 I/O 减至最少),因此还需提取有关存取行为的知识(操作数据集、时间限制等),以便在执行时进行基于上述提取知识的内外存数据交换,从而支持事务的定时限制.

② 事务分解:根据实时事务的同构性分解该实时事务得到替代.

③ 替代预分析:在动态环境下分析替代的行为特性,如分析替代的估算执行时间、事务间的时间相关性等.

关于实时事务预分析的策略和算法将另文详述.

2.2 可调度性分析

由于各个替代的性能不同,它们的可调度性也存在差异,系统应分析各替代的可调度性,得到可调度集.

定义 **3**. 在不考虑资源竞争时,如果实时事务 T 能满足相应约束地执行,则称该实时事务是可调度的,表示为 $SC(T)$.

定义 **4**. 事务 T 的一个任务 TK_i 是可调度的,乃指$\exists st_{ij}\in SUB(TK_i)SC(st_{ij})$.

由定义 3、4 和定理 1、2 直接得出下列引理和定理.

引理 **1**. 一个任务是可调度的,当且仅当它至少包含一个可调度的子事务.

定理 **3**. 当且仅当实时事务的所有任务(步)是可调度的,则该实时事务是可调度的.

由于实时事务的成功率与系统的实时环境有关,并不是每个替代都能成功执行,因此,我们必须结合替代本身的特点及系统的因素,排除那些不具备可调度性的替代,将内部调度的结果缩小到一个较小的有效范围,从而提高实时事务的成功率.

定理 **4**. 当且仅当实时事务中至少存在一个可调度的替代时,该实时事务是可调度的.

证明:设实时事务 $T=(FX,RS,CS,<_t)$,$FX=\{fx_i \mid 1\le i\le D(GT)\}$($D(GT)$为替代个数).

1) 当$\exists fx_d\in T\ SC(fx_d)$时,由定义 1 中的②,则有$\forall st_{ij}\in fx_d(SC(st_{ij}))$.再由引理 1 和定理 3 得:$SC(T)$.

2) 当 $SC(T)$时,

$$T=\langle TS,R,<_t,C\rangle;TS::=\{TK_i|i\in[1,n]\}.$$

根据定理 3,

$$SC(T)\Rightarrow\forall i\in[1,n]\ SC(TK_i).$$

根据引理 1,

$$\forall i\in[1,n]\ SC(TK_i)\Rightarrow\forall i\in[1,n](\exists st_{ij}\in TK_i\cap SC(st_{ij})).$$

由定义 1 的②及定理 3 得 $SC(fx_i)$.

定义 **5**. 实时事务调度集由该事务所有的可调度替代组成.

3 基于功能替代的二次调度策略

根据替代之间的性能差异,对该事务的所有可调度的替代做出调度,此调度有别于传统的实时事务调度,称为内部调度.内部调度的目的是在一个实时事务中选取最佳的替代,它实际上包括预分析和预调度两部分.外部调度就是传统的实时事务调度[4,5],针对于多个实时事务,目的是在多个实时事务中分配系统资源(包括 CPU、数据对象等).

设有实时事务集合 $TT=\{T_i|1\le i\le m\}$,$T_i=\{fx_{ij}|1\le j\le D(GT_i)\}$,用 Si 和 Se 分别表示内部调度和外部调度,则:

对于 T_i 的一个内部调度为

$$Si(T_i)=(\ fx_{iq_1},\ fx_{iq_2},\ldots,\ fx_{iq_c},\ldots,\ fx_{iq_p}),\ fx_{iq_1}\in T_i,1\le q_p\le D(GT_i).$$

对应的一个外部调度为

$$Se(TT)=scheduling(Si(T_1),Si(T_2),\ldots,Si(T_i),\ldots,Si(T_m))=(\ fx_{a_1},\ fx_{a_2},\ fx_{a_3},\ldots,\ fx_{a_k},\ldots,\ fx_{a_u}).$$

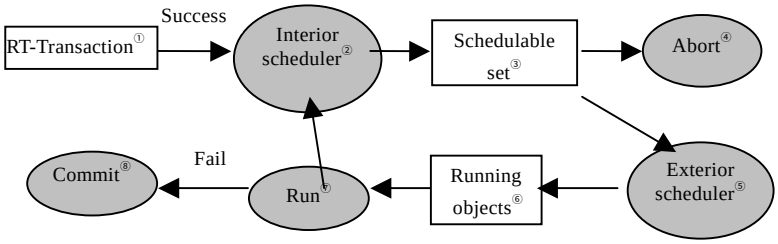
满足条件:

$$\forall a_i(\ fx_{ak}\in Sche\text{-}ex(TT))\exists j(\ fx_{ak}\in Sch\text{-}in(T_j))\cap\forall a_i,a_j(a_i\ne a_j)(\ fx_{ak}\in Sch\text{-}in(T_{j1}))\ fx_{ak}\in Sch\text{-}in(T_{j2})(T_{j1}\ne T_{j2}).$$

内部调度与外部调度紧密联系,内部调度是基础,只有通过内部调度才能选出一个可调度的替代参与外部调度.外部调度是最终目的,只有通过外部调度,实时事务才能真正投入执行.

如图 1 所示,当内部调度的结果为空集时,该实时事务不可调度,当事务执行失败时,不能立即夭折,必须重新转入内部调度,停止调度活动的原因有:

(1) 事务的截止期到;(2) 由特殊操作强迫停止;(3) 该事务的所有替代都夭折;(4) 有一个替代成功执行并提交.



①实时事务,②内部调度器,③可调度集,④夭折处理,
⑤外部调度,⑥运行集,⑦运行,⑧提交处理.

Fig.1 Real-Time transaction schedule

图 1 实时事务调度

内部调度机制采用基于优先级的调度策略对事务的可调度集进行调度,相应的优先级分派策略有:

(1) 执行时间最短优先

该策略将最高优先级分派给具有最短执行时间的替代,其优点是:① 执行时间最短空余时间就最长,成功的机会就更多;② 即使运行失败,运行对象也尽可能少地占有 CPU,减轻 CPU 负担,同时将发现错误的时间尽可能提早,会有较多的时间重新调度其他对象.

(2) 资源需求种类最少

该策略将最高优先级分派给具有最少资源需求种类的替代,因为较多的资源需求蕴涵着较多的资源竞争和受阻,从而使因资源等待而超过截止期的机会增大,该策略的优点是降低资源等待风险.

(3) 嵌套层次最少优先

事务的嵌套层次过多增加了调度的难度和资源负担,考虑到这方面的因素,该策略将最高优先级分派给具有最少嵌套层次的替代,此策略一般作为辅助策略使用.

综合考虑上面的因素,依应用语义的要求构造一个优先级函数:

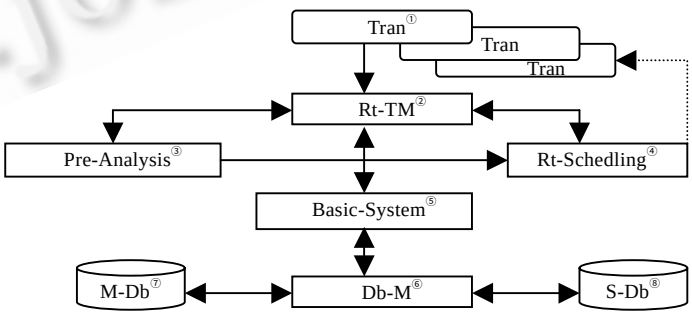
$$P(fx_i)=(\alpha_1 et_i+\alpha_2 R_i/R+\alpha_3 * ne)/(\alpha_1+\alpha_2+\alpha_3),$$

其中$\alpha_1,\alpha_2,\alpha_3$是权值,$ne$ 是 fx_i 的嵌套层次,它们由应用的语义确定.et_i 和 R_i 分别是 fx_i 的估算执行时间和资源需求量,通过分析处理决定,R 是系统可以提供的资源量.

4 性能分析

4.1 实验系统

模拟实验系统结构如图 2 所示,实时事务管理器(Rt-TM)负责事务的接纳、并发控制、分派优先级等;预分析处理器(pre-analysis)负责对实时事务进行预分析;调度程序(rt-scheduler)负责事务的内部调度和外部调度调度;基本系统(basic-system)实现数据的操纵与控制;数据库管理程序(db-manager)处于系统的最底层,实际上是一个内存数据库管理系统,物理地管理内存数据库(M-db)与外存数据库(S-db).



①实时事务,②实时事务管理器,③预分析,④实时调度,⑤基本系统,
⑥数据库管理系统,⑦内存数据库,⑧外存数据库.

Fig.2 Architecture for experiment system

图 2 实验系统的体系结构

模拟实验系统采用 Client/Sever 结构,模拟城市交通管理系统,在 20 台客户机上发出汽车运行事务或道路维护事务,汽车运行事务根据当前城市路况选择从起点到终点的路径,道路维护事务

可以封闭(如修路)或增加(如开放新路)道路等,服务器模拟交通管理中心,假设每个汽车要么停止,要么保持匀速运行,这便于估算事务的执行时间.实验系统的有关参数见表 1.实时事务的静态分析在客户机上进行,而系统的当前资源信息已经记录在服务器端的系统中,所选取的替代是较优的(理论上是最优的),因此在同等系统条件(资源条件、调度策略、并发控制策略等)下,系统的成功率得到提高.

Table 1 System parameters

表 1 系统参数

Parameters	Value
The average number of alternative sets including by each transaction①	3
The average number of relations accessed by each transaction②	3
The average number of pages of each relation accessed by each transaction③	5
The possibility of each page which be updated④	0.5
The fraction of slack time of processing time⑤	0.5

①每个事务中包含的替代平均数,②每个事务访问的关系的平均数,③被每个事务访问的关系的平均页数,④每页被修改的概率,⑤空余时间与处理时间的比.

4.2 性能比较

衡量实时系统性能的标准通常采用系统成功率.在实验系统中,性能比较结果如下(支持替代用 GT 表示,不支持替代用 NGT 表示):

图 3 表示在不同的系统负载上系统的成功率比较,从图中可见,GT 时系统的成功率更高,且高出的幅度较大.这是由于当事务失败时,如果 NGT,则该事务夭折,如果 GT,只要截止期未到,系统可能选取另外的替代,提高了该事务成功执行的可能性,从而提高了系统的成功率.图 4 表示在不同的系统负载时,CPU 有效率的比较,从图中可见,在 GT 和 NGT 两种情况下,CPU 有效率非常接近,并且,随着系统负载的增大,GT 时系统的 CPU 有效率更小.这是因为对于某事务而言,GT 意味着系统可能要处理多个替代,而只有花费在成功提交的替代上的时间才是有效的,随着系统负载的增大,冲突增多,系统要处理更多的替代,有效 CPU 时间相对较少.

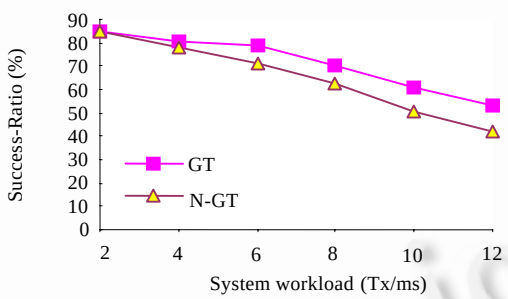


Fig.3 Success-Ratio comparing at difficult system workload

图 3 在不同系统负载下的成功率比较

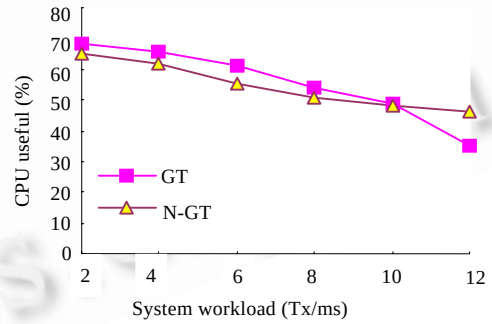


Fig.4 Useful CPU comparing at difficult system workload

图 4 在不同系统负载下的 CPU 有效率比较

5 结 束

本文所作的主要贡献是:① 分析了基于功能替代的嵌入式实时事务模型及事务特性,该模型支持嵌入式实时应用的多方案选择,有利于处理复杂实时环境下的实时事务,使实时事务具有更丰富的含义,改变了传统的实时事务应变能力差、成功率低的弱点;② 提出实时事务内外二重调度的新策略,并讨论了调度的具体步骤和策略;③ 在一个模拟系统中进行性能分析与比较.

References:

[1] Olson, M.A. Selecting and implementing an embedded systems. Computer, 2000,33(7):27~34.

[2] Vrbsky, S.V., Tomic, S. Satisfying temporal consistency constraints of real-time databases. The Journal of Systems and Software, 1999,45:45~60.

[3] Liu, Yun-sheng, Hu, Guo-ling. Transaction pre-processing in real-time main memory database. Journal of Software, 1997,8(3):204~209 (in Chinese).

[4] Konana, P., Ram, S. Transaction management mechenism for active and real-time databases: a comprehensive protocol and a performance study. The Journal of System and Software, 1998,42:205~228.

[5] Son, S.H. Issues and approaches to supporting timeliness and security in real-time database systems. Journal of Systems Architecture, 2000,46:397~410.

附中文参考文献:

[3] 刘云生,胡国玲.实时主存数据库的事务预分析.软件学报,1997,8(3):204~209.

Transactions Schedule of Embedded Database Systems*

LIU Yun-sheng[1], XIA Jia-li[1,2], XU Gui-ping[1]

[1](*College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074, *China*);

[2](*School of Accountancy, Jiangxi University of Finance and Economics, Nanchang* 330013, *China*)

E-mail: xiajl65824@263.net

http://www.hust.edu.cn

Abstract: A transaction model is proposed based on function alterative characteristic, which improves the ability of transaction suiting for real-time environment. As the characteristic, transaction schedule is divided into interior schedule and exterior schedule, which increases the systems success ratio. The other main contribution is to provide more abundance transaction semantic for embedded database systems.

Key words: database system; embedded system; transaction schedule

* Received April 18, 2001; accepted August 16, 2001

Supported by the National Natural Science Foundation of China under Grant No.60073045; the Ministry & Commission-Level Research Foundation of China under Grand No.00j15.3.3jw0529