# 实时主存数据库事务的预处理

刘云生 胡国玲 舒良才

(华中理工大学计算机系 武汉 430074)

**摘要** 本文提出了一种支持实时事务的预分析算法和基于该算法的主存数据库内外存替换策略. 它首先预分析实时事务以获得事务存取行为的知识,再在事务执行时,进行基于这些信息的存取及内外存数据交换,从而实现主存数据库管理和支持实时事务的定时限制.

**关键词** 事务预分析,实时事务,实时数据库,主存数据库.

实时数据库系统(RTDBS)就是事务和数据都可以具有定时特性或显式定时限制的数据库系统. 系统的正确性不仅依赖于事务的逻辑结果,还依赖于逻辑结果产生的时间[1],这就要求 RTDBS 能够较准确地估算事务在系统中的运行时间. 但就一般的常驻磁盘数据库系统(DRDBS)而言,事务在系统中的运行时间是无法预测的. 有许多不可预报的动态因素,其中之一便是这种数据库系统要频繁地进行内外存的数据交换,这样,磁头的定位时间、缓冲区管理和页面错误等都会导致事务的平均执行时间和最坏情况下的执行时间的估算误差很大. 如果能将实时数据库建立在主存数据库基础上,那么在事务的执行过程中就可以消除内外存的数据 I/O,这样不仅可以为系统准确估算事务的运行时间提供有力支持,同时也可以为实现实时事务的定时限制打下基础.

主存数据库系统(MMDBS)即是"当前"数据库或数据库的"工作版本"常驻主存的数据库系统. 事务在执行过程中,没有内外存的数据 I/O. 这里并没有排除主存容量有限、无法容纳整个数据库的情况. 在这种情况下,如何保证"当前"数据库或数据库的"工作版本"常驻主存,即如何保证实时事务在运行时,它所要访问的数据均在主存,如何进行内外存的数据交换,这是实现 MMDBS 的关键.

本文介绍我们研制的一个主动实时主存数据库系统 ARTs-1 的事务预分析算法和基于该算法的内外存替换策略. ARTs-1 中的实时事务具有这样的特点:在其 BOT(begin of transaction)点,它对数据库所要进行的操作类型和操作顺序都是确定的. 基于这一特点,我们可以在实时事务开始执行之前对其进行预分析和预处理以获取其存取行为的知识,再在事务执行时,进行基于这些信息的存取及内外存数据交换,从而实现主存数据库管理和支持

本文研究得到国家自然科学基金和国防预研基金资助. 作者刘云生,1940 年生,教授,博士导师,主要研究领域为现代数据库理论与技术,数据库与信息系统开发,软件方法学. 胡国玲,女,1967 年生,博士,主要研究领域为数据库与信息系统开发,智能系统. 舒良才,1971 年生,硕士,主要研究领域为数据库与信息系统开发.

本文通讯联系人:刘云生,武汉 430074,华中理工大学计算机系

本文 1996-02-07 收到修改稿

实时事务的定时限制.

## 1　ARTs-I 的事务模型

### 1.1　实时事务的形式模型

我们区分两类不同的事务："预设计"事务和"即席"事务."预设计"事务是指嵌入在应用程序中的预先设计好了的事务，在其开始执行(BOT)时，它对数据库所要进行的操作类型和操作顺序都已确定.而"即席"事务则是在其开始执行时，对数据库所要进行的操作类型和操作顺序尚未确定的事务.实时应用中的事务大多是"预设计"事务，故本文所阐述的预分析算法及内外存替换策略是针对"预设计"型的实时事务而言的.

在 ARTs-I 中，我们有

**定义** 1. 一个事务就是一个六元组$(E,A,R,\rightarrow,C,t)$

其中(1)$E=\{e_1,e_2,\ldots,e_n\}$为事件的集合，往往是外部事件.

(2)$A=\{a_1,a_2,\ldots,a_m\}$为关于数据库活动的集合.

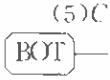
(3)$R=\{r_1,r_2,\ldots,r_n\}$为对系统环境反应的集合.

(4)$\rightarrow$为定义于 $E,A,R$ 上的偏序关系.

例如，对$\forall e\in E,a\in A,r\in R$，$e\rightarrow a\rightarrow r$，表示当且仅当事件 $e$ 发生后才能对数据库进行操作 $a$，当且仅当对数据库进行了操作 $a$ 之后，才能对系统环境输出反应信息 $r$.

在 ARTs-I 中，我们定义一个事件为一种系统行为的瞬时发生.系统行为可以是一种数据库操作(对象事件)、事务管理活动(事务事件)、时间行为(时钟事件)或与外部环境的交互作用(外部事件)，它可以由一个特定的时间点标识.

(5)C 为控制结构的集合.有 4 种控制结构：



表示一个事务执行的开始

表示一个事务执行的结束

表示路径的分支

表示事务中可以活动若干次的部分

表示事务中可以并行活动的部分

表示并行活动部分的联接

(6)$t$ 为实时事务的定时限制.

图 1 是一个典型的实时事务的模样，我们称之为 example-T，在本文的后续各节中将以该例来具体阐述我们的算法.

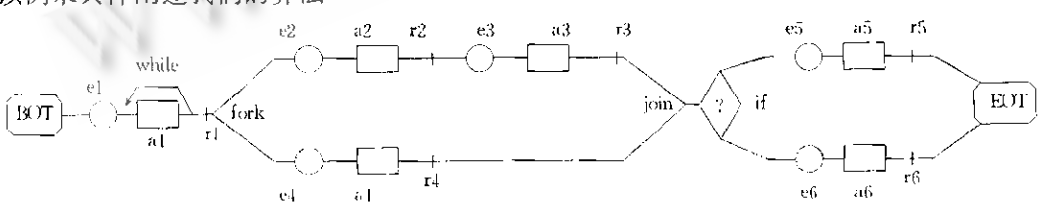



图1　一个典型的实时事务

### 1.2　ARTs-I 的事务处理

ARTs-I 事务有静态和动态 2 个方面，分别对应编译时和运行时的处理.事务处理环境

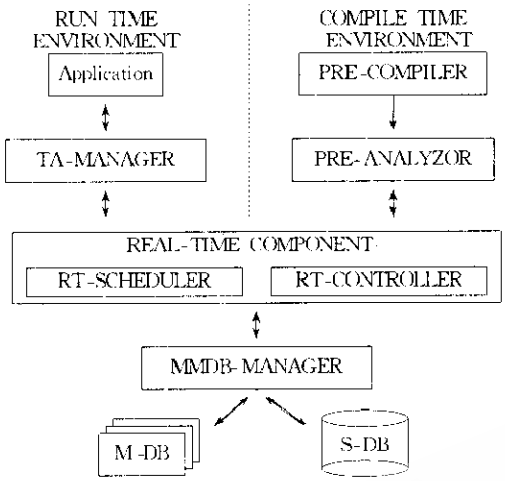



图2 实时事务的处理环境

如图2所示.其中TA-MANAGER类似于传统的事务管理程序,它处理事务的开始、提交、夭折等,但这里事务有定时特性,它要与系统的实时机构REAL-TIME COMPONENT相互作用.REAL-TIME COMPONENT实现事务的定时特性,包括"识时"的优先级分派与调度(RT-SCHEDULER)和并发控制(RT-CCONTROLLER)等.预编译器(PRE-COMPILER)包括各种语言的处理器及用户接口处理设施;预分析器(PRE-ANALYZOR)负责对实时事务进行预分析处理,下节将详细讨论.MMDB-MANAGER为ARTs-I的内存数据库管理程序.

## 2 实时事务的预分析

ARTs-I设置了编译时事务的预处理和运行前事务的预分析机制来支持实时事务定时特性的实现.

### 2.1 预编译处理

实时事务经过预编译处理后,所有的DML语句均转换为形如$p_i(s_1,s_2,\ldots,s_n)$的过程调用,$p_i(s_1,s_2,\ldots,s_n)$称为存取过程.此外,预编译器还将生成一张表,记录在一个实时事务中所有存取过程被执行的顺序,该表结构如下:

PER-STRUCT(proc_id,predecessor-id,successor-id)

其中proc-id为存取过程标识,predecessor-id为它的前置存取过程标识,successor-id为它的后继存取过程标识.图1所示的实时事务,经过预编译处理后,将转化为如图3的形式.

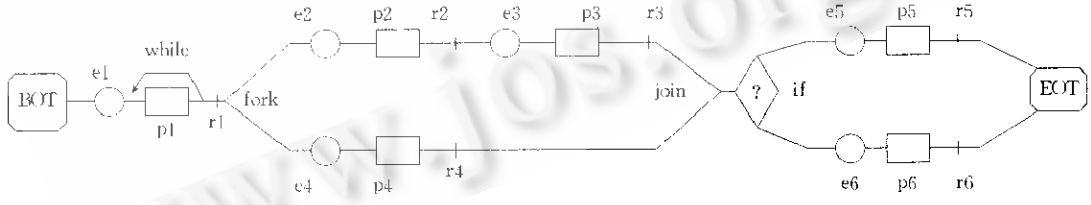



图3 预编译处理后的事务模型

我们用六元组$(E,A,P,\rightarrow,C,t)$表示经过预编译处理后的实时事务,其中$P=\{p_i \mid 0<i<(n+1)$,$p_i$为对应于$a_i$的存取过程$\}$,其余意义同定义1.

### 2.2 实时事务的预分析处理

设经过预编译了的实时事务$T=(E,A,P,\rightarrow,C,t)$.

**定义2(存取集).** 对$\forall\ p_i\in P$,称它所要存取的数据集为$p_i$的存取集,记为$D(p_i)$.

要保证$T$在存取过程中没有I/O,就必须保证在每一$p_i\in P$开始执行时,$D(p_i)$均在主存.预分析的任务就是确定每个存取过程$p_i\in P$的存取集$D(p_i)$.为此,它为每一$p_i$构造另一存取过程$p'_i$,它具有如下性质:

(1)$D(p_i')\supset D(p_i)$;

(2)$p_i'$不对$D(p_i')$进行更新操作.

我们称$p_i'$为$p_i$的"超存取过程"(Super Access Process).性质(2)决定了$p_i'$只对数据库进行检索操作.下面我们将详细讨论怎样使$p_i'$满足性质(1).

设我们采用的数据库操纵语言是基于元组关系演算的(如SQL,QUEL等),$t$表示元组变量,$t[i]$表示元组$t$的第$i$个分量.我们可将对应于$p_i'$的DML语句中的各种条件表达式$Q(p_i)$(一阶谓词公式)转换成析取范式的形式(DNF):

$$\mathrm{DNF}[Q(p_i)]=\bigvee_{0<k<p}\left(\bigwedge_{0<j<r}(q[k,j])\right);$$

其中$q[k,j]$为原子公式,每一$q[k,j]$均与相应的$p_i$所定义的对象事件$e_i$相联,称$e_i$为$q[k,j]$的"相联"事件.$q[k,j]$可以具有4种类型:

(1)$R(t)$.其中$R$为关系名,该原子公式表示$t$是$R$中的一个元组.

(2)$t[k]\theta u[j]$.其中$t$和$u$均为元组变量,$\theta\in\{\leqslant,\geqslant,\neq,=,<,>\}$.它表示元组$t$的第$k$个分量与元组$u$的第$j$个分量之间满足$\theta$关系.

(3)$t[k]\theta C$.它表示元组$t$的第$k$个分量与常数$C$之间满足$\theta$关系.

(4)$t[k]\theta v$.它表示元组$t$的第$k$个分量与程序变量$v$之间满足$\theta$关系.

设$\mathrm{QUAL}[p_i]$为对应于$p_i$的DML语句中条件表达式$Q(p_i)$的DNF中原子公式的集合　　$\mathrm{QUAL}[p_i]=\{q[k,j]\mid \mathrm{DNF}[Q(p_i)]=\bigvee_{0<k<p}\left(\bigwedge_{0<j<r}(q[k,j])\right)\}$;

在$\mathrm{QUAL}[p_i]$中的原子公式可以分为2类:一类是其真假值由相联事件$e_i$确定的原子公式.如若程序变量$v$的值由$e_i$确定,则原子公式$t[i]\theta v$的真假值也由事件$e_i$确定.另一类是其值在相联事件发生之前已确定的原子公式.如原子公式$R(t)$.因此,我们也可以相应地将$\mathrm{QUAL}[p_i]$分为2个子集:

$\mathrm{QUAL}_1[p_i]=\{q[k,j]\mid 0<k<p\wedge 0<j<r\wedge q[k,j]$的值由相联事件$e_i$确定$\}$

$\mathrm{QUAL}_2[p_i]=\{q[k,j]\mid 0<k<p\wedge 0<j<r\wedge q[k,j]$的值在相联事件$e_i$发生之前已确定$\}$

通过下列算法可以构造对应于$p_i$的超存取过程$p_i'$的查询条件表达式.

```
PROCEDURE QUAL
输入:QUAL[pi]—对应于 pi 的 DML 语句中条件表达式的 DNF 中原子公式的集合;
输出:SQUAL[pi']—对应于 pi' 的 DML 语句中条件表达式的 DNF 中原子公式的集合;
{SQUAL[pi']=QUAL[pi];
    for(k=1;k<p;k++)
    for(j=1;j<r;j++)
    if(q[k,j]∈QUAL1[pi])
      remove q[k,j] from SQUAL[pi'];/* 因为QUAL1[pi]中原子公式的值由相联事件确定 */
    compare QUAL[pi] and SQUAL[pi'];
    if(∃ tuple variable t∈QUAL[pi]∧ t∉SQUAL[pi'])
       disjunction add t.A=* to SQUAL[pi'];
}
```

其中$t$为条件表达式中的元组变量,$A$为$t$所属关系的主关键字,$t.A=*$表示$t$所属关系的所有元组的谓词.从上述算法描述中可以看出,$\mathrm{SQUAL}[p_i']$中原子公式的个数不多于

QUAL[$p_i$]中原子公式的个数，因而由SQUAL[$p'_i$]中原子公式构成的析取范式确定的存取数据集$D(p'_i)$一定包含$p_i$对应的DML语句中条件表达式所确定的数据.

图4所示为$p_i$的超存取过程$p_i'$的导出过程.

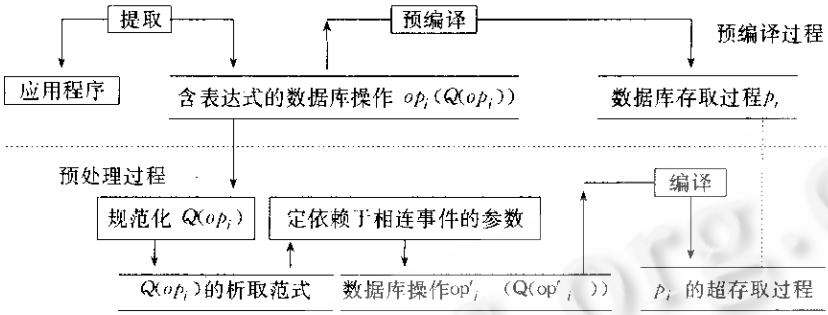



图4 预处理过程

在为每个$p_i$构造了$p_i'$之后，预分析器还将构造相应于实时事务$T$的另一系统事务$T'$，称其为超系统事务，该事务由二元组$(P',\rightarrow)$标识，其中$P'=\{p'_i|0<i<(n+1)$，$p_i'$为相应于$P_i$的超存取过程$\}$，$\rightarrow$为定义于$P'$上的偏序关，即若$p_1'\rightarrow p_2'$，则$p_1'$执行之后才执行$p_2'$，它就是$T$中各$p_i$的执行顺序. 图5为example-T经过预分析处理后所形成的超事务模型.

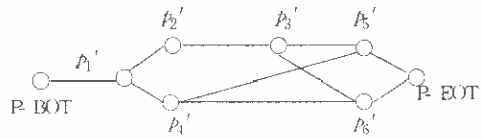



图5 example-T的超事务模型

实时事务经过预分析处理后，还将形成一张表，记录每个实时事务标识符及其相应的超事务标识符.

## 3 基于实时事务的内外存交换策略

当实时事务$T$开始执行时，系统同时也启动了其超系统事务$T'$，这2个事务在系统中是同步运行的：在$T$等待某个事件$e_i$发生而其后继的存取过程尚未执行时，$T'$执行其相应于$p_i$的超存取过程$p'_i$，这样便可将$p_i$的存取集$D(p_i)$调入主存数据库中(如果$D(p_i)$不在M-DB中)，在$p_i$执行之前，$D(p_i)$被定位在M-DB中以免被交换出去，$p_i$执行完之后，MMDB_MANAGER便可不再对其进行定位，如果需要，还可将其交换到S-DB中.

## 4 结束语

本文研究了主存数据库系统中在主存无法容纳整个数据库的情况下，为了消除事务运行中的I/O而进行的事务的预分析和预处理及基于它的内外存替换策略. 常规数据库管理系统的缓冲区替换算法(如LRU算法[23])都着眼于以往事务访问数据库的信息(页面被事务访问的次数). 本文所提出的内外存数据调度策略则着眼于事务将要访问的数据集合，该策略以实时事务的预分析算法为依据，能有效地实现主存数据库的目标和支持实时事务的定时特征.

## 参考文献

1　刘云生. 关于实时数据库事务. 软件学报，1995，6(10)：614～622.

2　Effelsberg W, Haerder T. Principles of database buffer management. ACM TODS, Dec. 1984, 19(4): 560～595.

# TRANSACTION PRE-PROCESSING IN REAL-TIME MAIN MEMORY DATABASE

LIU Yunsheng　HU Guoling　SHU Liangcai

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

**Abstract**　In this paper, the authors present transaction pre-analysis and data I/O strategies of a main memory database to support real-time transactions. A real time transaction is firsly pre-analyzed to get the information about access behavior of the transaction, the system performs data I/O between main memory and secondary storage (if necessary) based on the information. Therefore, the main memory database requirements are implemented and thus real-time transactions are supported.

**Key words**　Transaction pre-analysis, real-time transaction, real-time database, main memory database.