ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.6, June 2007, pp.1468−1476
DOI: 10.1360/jos181468

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于数据时态特性的实时事务并发控制*

韩启龙[1+], 郝忠孝[1,2,3]

[1](哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

[2](哈尔滨理工大学 计算机科学与技术学院,黑龙江 哈尔滨 150080)

[3](齐齐哈尔大学 计算机学院,黑龙江 齐齐哈尔 161006)

Real-Time Concurrency Control Protocol Based on Accessing Temporal Data

HAN Qi-Long[1+], HAO Zhong-Xiao[1,2,3]

[1](School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[2](School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

[3](School of Computer, Qiqihar University, Qiqihar 161006, China)

+ Corresponding author: Phn: +86-451-86390012, E-mail: hanqilong@hit.edu.cn, http://www.hit.edu.cn

Han QL, Hao ZX. Real-Time concurrency control protocol based on accessing temporal data. *Journal of Software*, 2007,18(6):1468−1476. http://www.jos.org.cn/1000-9825/18/1468.htm

Abstract: After analyzing the temporal data characteristics and the effects on scheduling transaction, a real-time concurrency control protocol is proposed, which improves the performance of real-time systems by evaluating data-deadline and transactions execution time, improving transactions validation rules, and adjusting transactions committing order. Theoretical analysis and experimental results demonstrate that the new protocol reduces the transaction restarts numbers and the miss deadline percentage, and outperforms the previous ones.

Key words: active real-time database; concurrency control; data-deadline; validation factor

摘 要: 通过对数据时态特性及其对事务调度的影响进行分析,提出了基于数据时态特性的实时事务并发控制算法.该算法根据数据截止期及事务的执行时间估算,改进了事务的验证规则,对事务的提交顺序进行调整,提高了系统的实时性能.理论分析与实验结果表明:该算法降低了事务重启个数及超截止期百分率,性能要优于已有的实时并发控制算法.

关键词: 主动实时数据库;并发控制;数据截止期;验证因子

中图法分类号: TP311 文献标识码: A

在主动实时数据库相关研究中,一个很重要的领域是事务的并发控制.一方面,主动实时数据库与实时应用密切相关,并发控制算法是否有效将严重影响实时系统的性能;另一方面,主动实时数据库中的实时数据更新可以触发新的事务,进一步增加了事务并发控制的难度.因此,既要保证数据库的一致性,又要保证事务在时间限制之内完成,一直是并发控制研究的重要问题.在这个过程中,已有的工作更多地把重点放在具有时间限制的事务上,而对于数据的实时性则考虑得不够.这主要是由于实时数据大多是由传感器动态获得的,而实时数据库中

* Supported by the Natural Science Foundation of Heilongjiang Province of China under Grant No.F00-06 (黑龙江省自然科学基金)
Received 2005-08-01; Accepted 2006-05-16

的事务只对实时数据进行读操作,不需要对其进行并发控制机制.但是,实时数据在读取其事务提交之前,有可能因超过时间限制而变得无效,无论是对整个系统还是对读取其事务都产生了重要的影响.因而无论是对单个事务调度,还是事务间并发控制,数据的实时性都是一个非常重要的问题.本文将进一步讨论有关实时数据的事务并发控制问题.

乐观的并发控制方法[1]因具有无死锁与无阻塞特性而广泛应用于实时数据库中,但延迟的冲突检测带来了巨大的重启开销.文献[2,3]提出了动态调整串行化顺序的方法,减少了不必要的事务重启个数;文献[4]根据文献[3]对数据的读写进行控制,进一步减少了事务重启个数.但在上述文献中,共同的问题是当冲突事务存取有时间限制的数据时,无法用这些方法进行控制.文献[5]详细讨论了有关实时数据与导出数据有效性的问题.文献[6]提出了实时数据相似性的概念,根据数据相似性扩展了数据的有效期以满足事务调度.文献[7]根据文献[5]提出了数据截止期的概念,用来表示数据时间限制,并采用强制等待的事务调度策略,对实时数据进行调度.文献[8]也对数据截止期及事务调度进行了讨论.但这些文献只对单个事务调度实时数据的问题进行了讨论,而没有考虑并发控制问题.文献[9]讨论了有关数据时间有效期的数据库系统,当事务所读的任何时态数据对象在事务提交之前变得无效时,事务将夭折或重启.但存在以下问题:(1) 事务在其存取时态数据对象有效期过期之前不能提交,只采取简单的夭折策略将引起不必要的夭折;(2) 由于基于优先权的调度可能引起事务抢占,导致事务所读取的实时数据过期产生时态不一致;(3) 没有考虑存取时态数据的事务的并发控制问题;(4) 没有考虑事务所读取的数据对象的版本与当前版本非常接近,即数据的两个版本相似的情况.本文基于文献[7]中的数据截止期的定义,针对文献中存在的问题进行有关实时事务并发控制的进一步研究.

1 系统模型正确性及实时数据的分析

1.1 系统模型的正确性

主动实时数据库包含一组对象及 ECA(event-condition-action)规则.每个对象与一个现实世界实体对应,现实世界实体的状态通常由传感器监测.在这样的数据库中,所有的数据对象分为时态与非时态对象以及基本与导出对象两大类.

时态对象的状态可能因时间过期而无效,与其相关的是时态有效期,可分为绝对有效期和相对有效期.非时态对象不存在时态有效期.时态对象的绝对有效期可表示为$(value(X_i),avi(X_i))$,其中,$value(X_i)$代表数据对象 X 的当前状态,$avi(X_i)=[avi_b(X_i),avi_e(X_i)]$,$avi_b(X_i)\leq avi_e(X_i)$为 $value(X_i)$的绝对有效期,其中,X_i 为数据对象 X 的第 i 个版本,$avi_b(X_i)$代表 X_i 的绝对有效期的开始,$avi_e(X_i)$表示 X_i 绝对有效期的结束.在 $avi_e(X_i)$之后,$value(X_i)$不再有效.时态对象的相对有效期表示为 $rvi(R)$,其中,R 为相对一致集,R 中每个元素为一个时态数据对象的版本.在时刻 $t>0$ 时,R 具有正确状态,当且仅当

(1) $\forall X_i\in R$,$value(X_i)$是逻辑一致的,即满足所有的完整性限制.

(2) R 是时态一致的:

i. $\forall X_i\in R$,$value(X_i)$是绝对一致的,即 $avi_b(X_i)\leq t\leq avi_e(X_i)$.

ii. R 是相对一致的,即$\forall X_i,Y_j\in R,|avi_b(X_i)-avi_b(Y_j)|\leq rvi(R)$.

基本对象反映环境中的特定实体,由传感器更新到数据库.根据基本对象及其他对象导出的新的数据对象为导出对象.在主动实时数据库中,当基本对象被传感器更新时,可以根据 ECA 规则触发事务.被触发事务可以更新那些从基本对象导出的数据对象的状态.本文所讨论的基本时态对象采取需要调度策略[7],当事务需要调用该时态数据时再从传感器调入.

系统中事务当且仅当满足以下条件时能够成功提交:

(1) 事务是逻辑一致的;

(2) 事务能够满足截止期;

(3) 事务读取的数据是时态一致的,并且当该事务提交时,其所读取的数据仍然有效.

1.2 实时数据对事务并发控制的影响

有关传感器事务与用来更新导出对象的被触发事务在文献[5]中有详细讨论,本文只讨论与并发控制相关的用户事务,与文献[7,8]相似,用户事务 T 具有如下属性:

$a(T)$:事务 T 的到达时间;

$s(T)$:事务的开始时间;

$d(T)$:事务的截止期;

$dd_t(T)$:事务 T 在时刻 t 的数据截止期;

$L(T)$:事务访问的数据对象数;

$L_t^{to}(T)$:时刻 t 以后事务 T 还要访问的时态数据对象数;

$L_t^{nto}(T)$:时刻 t 以后事务 T 还要访问的非时态数据对象数;

$L_t(T)$:时刻 t 以后事务 T 还要访问的数据对象数, $L_t(T)=L_t^{to}(T)+L_t^{nto}(T)$;

$E_t(T)$:在时刻 t 估计事务 T 尚需执行的时间;

$C_t(T)$:在时刻 t 估计事务 T 的完成时间,$C_t(T)=t+E_t(T)$;

$RS_t^{to}(T)$:截至时刻 t 事务 T 访问的时态数据对象集;

$P_t(T)$:时刻 t 事务的优先级.

在主动实时数据库中,时态数据具有许多特性,本节只给出有关本文所提出算法的相关特性.首先根据文献[7]介绍数据截止期的概念.

定义 **1**. 事务 T 在时刻 t 的数据截止期是指 T 截至时刻 t 访问的时序数据对象的有效期终点的最小者.记为 $dd_t(T)$,即 $dd_t(T)=\min_{X\in RS_t^{to}(T)} avi_e(X)$.

数据截止期的引入显式地表达了数据与事务之间的时间关系.但数据截止期的引用增加了事务调度及并发控制的难度,因为没有达到截止期的事务可能因为数据截止期而重启或夭折.下面举例说明数据截止期对事务并发控制的影响.

例 1:事务 T_1:$w_1(x)r_1(y)$;T_2:$w_2(y)r_2(z)$.

T_1 的截止期为 t_7,预计完成时间为 t_6;T_2 的截止期为 t_8,预计完成时间为 t_6.时态数据 z 的有效期为$[t_4,t_6]$.

如图 1 所示,在 t_5 时刻 T_2 进入验证阶段,此时 $WS(T_v)\cap RS(T_a)=\{y\}\neq\varnothing$.根据文献[2,3],执行顺序为 $T_1\rightarrow T_2$,即 T_2 将延迟提交.若 T_1 在时刻 t_6 提交,此时,T_2 因时态数据 z 超过截止期将夭折.

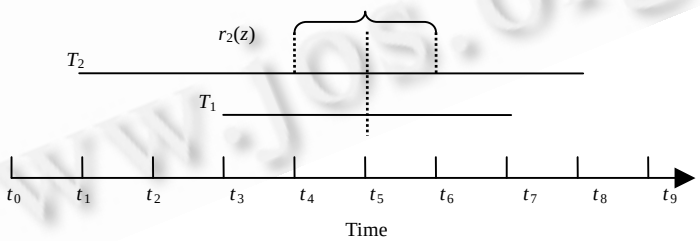


Fig.1 Execution of transactions

图 1 事务执行情况

如果不考虑数据的实时性,则根据传统的乐观方法[2,3]将成功调度 T_1,T_2,但是增加了数据的实时特性,事务 T_2 因无法满足数据截止期而产生夭折.例 1 说明了数据的时态特性对事务并发控制的影响,而现有的并发控制策略并没有考虑到有关存取时态数据的事务发生冲突的情况.

除了数据截止期以外,另一个重要的特性是数据的稳定性.实时数据有着不同的变化速率,有的随着时间的推移而频繁地变化,有的很长时间才发生变化.若$|avi_e(X)-avi_b(X)|<k$,则称时态数据 X 为易变的,否则称时态数据 X 为稳定的.根据事务的不同时间限制,k 的值随着事务对时态数据的访问动态地变化,在第 2 节中有详细说明.

总之,数据的实时特性对数据库事务并发控制会产生严重的影响,我们正是从实时数据的这些特性对事务并发控制的影响这一角度做进一步的研究.

2 相关定义及检查机制

2.1 相关定义

数据截止期作为事务的另一种时态限制,如果数据截止期短于事务截止期,则必须考虑数据截止期,否则,事务在提交时可能读取了时态不一致的数据.为了保证数据的时态一致性,使事务所读取的数据在事务完成时是有效的,必须对事务的执行时间进行检测与估算.

定义 **2**. 事务的截止期为 $d(T)$,预计完成时间为 $C_t(T)$,则事务的可延迟时间为 $sd_t(T)=d(T)-C_t(T)$.因为验证事务在验证时刻已完成所有读写操作,故可延迟时间为 $sd_t(T_v)=d(T_v)-t$.

定义 **3**. 若 $L_t^{to}(T)=\varnothing$,则称 $tsd_t(T)$为事务 T 的时态可延迟时间.

$$tsd_t(T)=\begin{cases} d(T)-C_t(T), & dd_t(T)\geq d(T) \\ dd_t(T)-C_t(T), & dd_t(T)<d(T) \end{cases}.$$

定义 **4**. 在时刻 t,事务的开始时间为 $s(T)$,截止期为 $d(T)$,则 $TFD_t(T)=t-s(T)/d(T)-s(T)$为时刻 t 的事务完成度.

定义 **5**. 设验证事务 T_v 的冲突事务集为 $CTS(T)$,事务 T_v 的验证因子为验证时刻 t 的验证事务的完成度与 $CTS(T)$中事务完成度的比值,表示为 $VF_t(T_v)=TFD_t(T_v)/TFD_t(T),T\in CTS(T)$.如果在时刻 t,$VF_t(T_v)>1$,则说明验证事务要比冲突事务更接近完成,否则为冲突事务接近完成.

定义 **6**. 事务的最小运行时间是指从当前时刻 t 到事务尚需执行的时间.

如果事务读取的时态数据对象的有效期小于事务的最小运行时间,则事务不可能在数据有效期内提交,这样的有效期称为不可能有效期.因此,在事务读取数据之前,采用一个检查机制来检查数据的有效性,以阻止事务读取无效数据,或者是存在不可能有效期的数据.

2.2 检查机制

在事务执行前预先声明事务需要访问的时态数据集 $RS_{s(T)}^{to}(T)$,时态数据的个数为 $L_{s(T)}^{to}(T)$,在事务开始执行后依次访问时态数据,在执行过程中动态检查数据的有效性,如果 $dd_t(T)>d(T)$,且 $L_t^{to}(T)=\varnothing$,则数据截止期比事务的截止期要长,且事务不会再访问其他时态数据,因此不会对事务的调度及并发控制产生影响.如果 $dd_t(T)>d(T)$,但 $L_t^{to}(T)\neq\varnothing$,事务还将访问其他时态数据,因此必须检查访问了所有的时态数据时是否能够满足 $dd_t(T)>d(T)$.检查算法(CHECKING algorithm)描述如下:

CHECKING Algorithm ().

INPUT: $RS_{s(T)}^{to}(T)=\{X^1,X^2,\ldots,X^m\}$;

OUTPUT: k, $L_t^{to}(T)$.

{$k=\infty$;$N=L_{s(T)}^{to}(T)$; i=1;

While ($RS_{s(T)}^{to}(T)\neq\varnothing$)

{T accesses X^i from $RS(T)$;

$RS_t^{to}(T)=\{X^1,X^2,\ldots,X^m\}-\{X^i\}$;

if ($|avi_e(X^i)-avi_b(X^i)|<k$)

then if $dd_t(T)<C_t(T)$

then Abort(T);

else $k=k\bigcap_{j=1}^{i}|avi_e(X^j)-avi_b(X^j)|$;

}

$L_t^{to}(T)=N-1$;

$i=i+1$;

}

事务在访问每个时态数据前调用检查算法,以保证所读取时态数据的有效性.同时,通过对 k 值的动态调整,保证了易变数据的时态一致性.

引理 **1**. 检查算法能够保证易变数据的时态一致性.

若$|avi_e(X)-avi_b(X)|<k$,则称时态数据 X 为易变的;否则,称时态数据 X 为稳定的.k 值代表了事务绝对有效期的长度,随着事务对时态数据访问的动态变化,k 值越小,说明时态数据越不稳定.若事务所要访问的下一个时态数据要比当前的时态数据易变,即$|avi_e(X^i)-avi_b(X^i)|<k$,则重新检测此时事务是否能够在数据截止期之前提交,如果可以满足时态一致性,则更改 k 值为该时态数据的绝对有效期长度.因此,检查算法对每一个易变数据进行时态一致性检查,以保证事务能够正确提交.

定理 **1**. 检查算法能够保证事务调度时态数据的时态一致性.

证明:对事务 T 所需访问的时态数据个数 $n(T)$进行归纳,

(1) 当 $n(T)=1$ 时,命题显然成立,否则事务将夭折.

(2) 设 $n(T)=m$ 时,命题成立.

当 $n(T)=m+1$ 时,可分为两种情况:

(1) 当$|avi_e(X^i)-avi_b(X^i)|\geq k$ 时,一定可以满足数据时态一致性.因假设事务 T 在时刻 t 访问时态数据 X^i,根据采用的需要调度策略,$avi_b(X^i)=t$,此时,$avi_e(X^i)\geq dd_t(T)$.而根据假设,当 $n(T)=m$ 时,命题成立,即 $dd_t(T)\geq C_t(T)$,故$avi_e(X^i)\geq C_t(T)$.因此,在事务完成之前,所有的数据都可以满足时间限制.命题成立.

(2) 当$|avi_e(X^i)-avi_b(X^i)|<k$ 时,即事务所读取的下一个时态数据的绝对有效期的长度小于 k.这说明此时访问的数据为易变数据,需要重新计算 $dd_t(T)$的值,根据引理 1,也可以满足数据的时态一致性.命题成立.

综合(1),(2),命题成立. □

3 基于数据时态特性的实时并发控制 RTCC-DD

3.1 RTCC-DD验证调整规则

数据库中的并发控制机制必须保证数据库的一致性,可串行性是数据库中并发控制是否正确的一个标准.本文所讨论的方法基于乐观的并发控制策略,在事务读取阶段通过 CHECKING Algorithm 算法保证事务存取时态数据的时态一致性.当事务进入验证阶段时进行读写调整检测,在该阶段,事务所做的修改对数据库是有效可读的,冲突事务可以读取该事务的私有缓存来得到该事务所做的修改.当所有冲突事务都串行化在该事务之后时,提交该事务.具体的调整规则如下:

规则 **1**. 在时刻 t,当 $L_t(T)=0$ 时,为事务 T 分配串行序号 $ser(T)$并进入验证阶段,$ser(T)$的取值范围为 1,2,…,n.第 1 个进入验证阶段的事务分配序号为 1,以后进入的事务依次累加.

规则 **2**. 在时刻 t,$avi_e(X_i)<C_t(T)$且 X 第 i 个版本的值与下一个版本的值相似,则调整 X 的时态有效期为下一个版本的时态有效期,即时态对象为$(value(X_i),avi(X_i))$,$avi(X_i)=[avi_b(X_i),avi_e(X_{i+1})]$.

规则 **3**. 在时刻 t,有 $VF_t(T_v)<1$ 且 $tsd_t(T_v)>tsd_t(T_i)$,其中 $T_i\in CTS(T_v)$,则调整事务串行化顺序为 $ser(T_i)=ser(T_v)$,$ser(T_v)=ser(T_v)+1$.如果 $RS(T_v)\cap WS(T_i)\neq\varnothing$,则 T_v 读取 T_i 的私有缓存中的数据.

规则 **4**. 在时刻 t,提交事务 T,当且仅当$\forall T_i\in CTS(T)$,$ser(T)<ser(T_i)$.

规则 1 保证在事务完成对数据的访问后才进入验证阶段并分配串行序号.规则 2 保证了在下一个版本数据与当前版本相似的情况下,延长时态数据的时态有效期,减少不必要的事务重启.规则 3 保证了对冲突事务比验证事务接近完成,且验证事务可延迟到冲突事务提交后,调整事务的执行顺序,优先执行冲突事务;当验证事务与冲突事务发生读写冲突时,允许验证事务读取冲突事务所更新的数据.规则 4 则保证了事务提交的可串行性.

3.2 RTCC-DD方法描述

RTCC-DD 方法在事务执行前预先声明需要访问的数据集,利用检查算法保证事务所存取时态数据的有效性.采用乐观的方法,调整事务验证阶段的验证规则.在验证事务与其他事务发生冲突时,通过对验证因子的判断,优先调度即将完成的事务;同时,考虑验证事务的时态可延迟时间,动态调整事务的执行顺序,以保证调度时态数据的事务能够满足时态一致性.在保证事务及数据时间限制的同时,最大程度上减少了不必要重启的事务.

RTCC-DD 并发控制方法描述如下:

$\forall X_i \in RS_t^{to}(T)$, if X_i is similar to $X_{i+1} \wedge dd_t(T) < d(T)$

then $avi(X_i)=[avi_b(X_i), avi_e(X_{i+1})]$

endif

if T_v conflicts with T_i, $T_i \in CTS(T_v)$

if $RS(T_v) \cap WS(T_i) \neq \varnothing$ then

if $(VF_t(T_v)<1) \wedge (tsd_t(T_v)>tsd_t(T_i))$

then adjusts the execute order to T_i, T_v

$ser(T_i)=ser(T_v)$, $ser(T_v)=ser(T_v)+1$;

else the execute order is T_v, T_i

$ser(T_i)=ser(T_v)+1$;

else if $RS(T_i) \cap WS(T_v) \neq \varnothing$ then

if $(VF_t(T_v) \geq 1) \wedge (tsd_t(T_i)>tsd_t(T_v))$

then adjusts the excute order to T_v, T_i

$ser(T_i)=ser(T_v)+1$;

else the execute order is T_i, T_v

$ser(T_i)=ser(T_v)$, $ser(T_v)=ser(T_v)+1$;

else if $WS(T_i) \cap WS(T_v) \neq \varnothing$ then

if $(VF_t(T_v)<1) \wedge (tsd_t(T_v)>tsd_t(T_i))$

then adjusts the execute order to T_i, T_v

$ser(T_i)=ser(T_v)$, $ser(T_v)=ser(T_v)+1$;

else the execute order is T_v, T_i

$ser(T_i)=ser(T_v)+1$;

endif

endif

RTCC-DD 方法通过检查算法动态调整数据的 k 值,对每一个易变的时态数据都进行计算以保证对易变数据的调度;对与下一个版本相似的不能满足时态限制的数据延长了其有效期截止时间;同时,采用乐观的并发控制方法,不会产生优先权倒置及阻塞问题;并且解决了有关访问时态数据的事务并发控制问题.完善了文献中所讨论的有关访问时态数据的事务调度问题.

与传统数据库不同,实时数据库并发控制方法的性能指标不再是系统吞吐量与平均响应时间,而是满足时间限制的事务个数.而影响乐观并发控制方法性能的主要因素之一是不必要的事务重启问题,因此,如何降低不必要的事务重启个数是衡量并发控制算法性能的关键.通过以下分析可知,RTCC-DD 优于传统并发控制方法.

首先,RTCC-DD 方法采用的是乐观的思想,如果所调度的数据没有时间限制,则 RTCC-DD 方法将退化为传统的乐观并发控制方法,因此,传统的乐观并发控制方法可调度的事务情况也可以用 RTCC-DD 方法来调度.

其次,传统的并发控制方法在事务提交时,若其访问的实时数据已过期,则在这种情况下,由于传统并发控制方法没有考虑数据截止期问题,因此在事务提交后将产生错误的结果,使系统处于不一致的状态.为了保证系统的正确性,必须在每个事务提交时利用文献[7]中的调度方法进行判断,而访问的实时数据已经过期,因此要重

后该事务.从例 1 来看,由于传统的并发控制方法没有考虑数据截止期,因此导致了不必要的事务重启,增加了事务重启个数.而 RTCC-DD 并发控制方法由于在事务间发生冲突时通过对事务执行情况及实时数据的时态特性进行分析,优先考虑了事务与数据的时态特性,降低了事务重启的个数,因此要优于传统的并发控制方法.

3.3 例 子

例 2:与例 1 中的事务及执行情况相同,事务 T_1:$w_1(x)r_1(y)$;T_2:$w_2(y)r_2(z)$.T_1 的截止期为 t_7,预计完成时间为 t_6;T_2 的截止期为 t_8,预计完成时间为 t_7.时态数据 z 的有效期为$[t_4,t_6]$.

在 t_5 时刻 T_2 进入验证阶段,此时,$WS(T_v)\cap RS(T_a)=\{y\}\neq\varnothing$.根据 RTCC-DD 方法,$VF_t(T_v)=\frac{4}{7}\Big/\frac{2}{4}=\frac{8}{7}>1$,$tsd_{t_5}(T_1)=t_7-t_6$, $tsd_{t_5}(T_2)=t_6-t_6$,因此满足条件$(VF_t(T_v)\geq 1)\wedge(tsd_t(T_i)>tsd_t(T_v))$,则调整事务执行顺序为 T_2,T_1.在 T_2 提交之后,T_1 也可以满足截止期限制,成功调度 T_2,T_1.因此,利用 RTCC-DD 方法能够避免事务的不必要重启.

3.4 正确性证明

可串行性是评价数据库并发控制是否正确的一个标准,RTCC-DD 方法在保证存取时态数据的事务之间的并发控制的同时,也满足事务的可串行性.下面讨论有关优越性与正确性问题.

定理 **2**. RTCC-DD 方法能够保证事务调度的可串行性.

证明:根据规则 1,只有当 $L_t(T)=0$ 时为事务 T 分配串行序号 $ser(T)$,保证在事务进入验证阶段时才分配序列号,并且是唯一有序的,通过规则 3 对冲突事务进行动态调整,并根据执行情况调整事务串行序号,保证序号最小的事务最先提交.根据规则 4,只有当所有冲突事务都串行化在该事务之后时才提交该事务.因此,对于事务的提交是严格按照事务的串行序号进行的,保证了事务提交的串行性. □

4 实验分析

4.1 实验参数

我们对文中所讨论的方法通过仿真实验进行测试,利用仿真软件包 SIM++用 C++程序实现.假定数据库由许多页组成,每页包含固定数目的记录,事务由若干读写操作组成,事务所处理的记录在数据库中均匀分布.表 1 给出了系统的基本参数.

主要性能参数是事务重启百分率.计算公式如下:

事务重启百分率(miss percentage)=超截止期重启事务数/进入系统的事务总数.

Table 1 Simulation parameter

表 **1** 仿真参数

Parameter	Meaning	Value
N_{DB}	Database size	400
ET_T	Transaction execute time	100ms
CT_{CPU}	CPU computation time	10ms
Slack	Transaction slack	8~12
N_{perT}	Operation number per transaction	5~50
RT_{OV}	Restart overhead	10
AT_{MT}	Mean transaction arrive time	10~200ms
N_{to}	Number of temporal data	100
N_{nto}	Number of non-temporal data	500

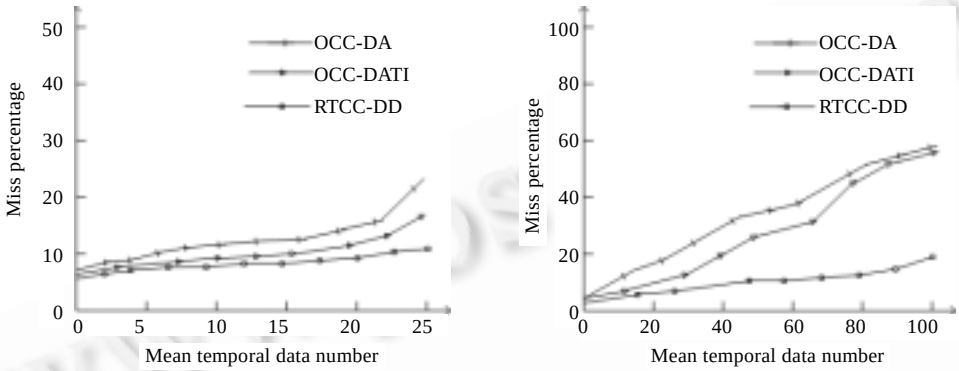
事务 T 的长度即其操作个数,记为 $L(T)=U(5,50)$,其中,$U[i,j]$代表在区间$[i,j]$内随机取的整数值.若 $L_t(T)$代表事务 T 在时刻 t 的剩余长度,则在时刻 t,事务 T 剩余时间估算值为 $E_t(T)=a(T)+L_t(T)\times CT_{CPU}$.

4.2 性能分析

实验主要通过对 RTCC-DD 方法与虽然在单个事务调度时考虑数据截止期,但仍采用传统的乐观并发控制方法的 OCC-DA 及 OCC-DATI 进行比较,进而对事务平均访问时态数据个数发生变化时,不同的并发控制策略

进行对比.

实验结果如图 2 所示,图 2(a)中事务所存取的时态数据平均个数由 0 到 25 进行变化,RTCC-DD 方法的性能要优于 OCC-DA 和 OCC-DATI 方法.因为事务调度中考虑了数据截止期问题,因此当事务因提交时已经超过数据截止期,事务将重启.OCC-DA 和 OCC-DATI 方法都没有考虑数据的时态限制,当事务存取少量时态数据时,性能差别不是很大.但如图 2(b)所示,当系统中存在大量时态数据时,RTCC-DD 方法则明显要优于未考虑数据时间限制的传统并发控制方法,也就是说,随着系统中时态数据的增加,数据的时间限制对事务的并发控制影响增大,有更多的事务因无法满足数据截止期而重启.



(a) Little number temporal data
(a) 具有较少时态数据

(b) Much number of temporal data
(b) 具有较多时态数据

Fig.2 Transactions miss percentage

图 2 事务超截止期百分率

5 结束语

并发控制问题是影响实时数据库系统的关键因素,乐观的方法因不存在死锁和阻塞而广泛应用于实时数据库系统中,不必要的事务重启与即将完成的事务重启是影响乐观方法的核心问题.因此,所有的乐观并发控制方法都是围绕如何减少不必要的事务重启和即将完成的事务重启问题而进行的.当前最有效的是基于动态调整串行化顺序的方法,但当事务存取具有时间限制的时态数据时,传统的并发控制方法因没有考虑数据截止期而无法对这样的事务进行有效调度.本文通过对事务验证阶段的规则进行改进,提出了基于时态数据的乐观并发控制方法,该方法充分考虑了时态数据的时间限制及其对事务的影响,有效解决了这一问题.实验证明,RTCC-DD 能够有效减少因数据截止期而引起不必要的事务重启个数,性能比现有并发控制方法更加优越.

References:

[1] Liu YS. Advanced Database Technology. Beijing: National Defence Industry Press, 2001 (in Chinese).

[2] Haritsa JR, Carey MJ, Livny M. Dynamic real-time optimistic concurrency control. In: Proc. of the 11th Real-Time Symp. IEEE Computer Society Press, 1990. 94−103. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=128734

[3] Lindstrom J. Optimistic concurrency control methods for real-time database systems [Ph.D. Thesis]. Helsinki: University of Helsinki, 2003.

[4] Wang YY, Wang Q, Wang HA, Dai GZ. Dynamic adjustment of execution order in real-time database. In: Proc. of the 18th Int'l Parallel and Distributed Processing Symp. New Mexico: IEEE Computer Society Press, 2004. 1219−1225. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1303028

[5] Adelberg B, Kao B, Garcial-Molina H. Database support for efficiently maintaining derived data. In: Apers PMG, ed. Proc. of the Extending Database Technology. Avignon: Springer-Verlag, 1996. 223−240.

[6] Xiong M, Ramamritham K, Stankovic JA, Towsley D, Sivasankaran R. Scheduling transactions with temporal constraints: Exploiting data semantics. IEEE Trans. on Knowledge and Data Engineering, 2002,14(5):1155−1166.

[7] Xiong M, Sivasankaran RM, Stankovic JA, Ramamritham K, Towsley D. Real-Time Systems: Issues and Applications. Dordrecht: Kluwer Academic Publishers, 1997. 167−191.

[8] Liu YS, Li GH. The effect of real-time database data characteristics on transactions. Journal of Computer Research & Development, 1999,36(3):364−368 (in Chinese with English abstract).

[9] Kuo T, Mok AK. Real-Time data semantics and similarity-based concurrency control. IEEE Trans. on Computers, 2000,49(11): 1241−1254.

附中文参考文献:

[1] 刘云生.现代数据库技术.北京:国防工业出版社,2001.

[8] 刘云生,李国徽.实时数据库数据特征对事务处理的影响.计算机研究与发展,1999,36(3):364−368.

韩启龙(1974－),男,黑龙江肇东人,博士生,主要研究领域为实时主动数据库,事务处理.

郝忠孝(1940－),男,教授,博士生导师,主要研究领域为数据库系统理论与应用,空间数据库.

****

第 7 届全国虚拟现实与可视化学术会议(CCVRV 2007)

征文通知

由中国计算机学会虚拟现实与可视化技术专业委员会、中国图像图形学会虚拟现实与可视化技术专业委员会和中国系统仿真学会虚拟现实技术专业委员会主办，北京航空航天大学承办的第 7 届全国虚拟现实与可视化技术及应用学术会议将于 2007 年 10 月在北京举行。本次会议将集聚国内从事虚拟现实与可视化技术的研究人员和工程技术人员，广泛开展学术交流、研究发展战略、推动成果转化、共同促进虚拟现实与可视化技术的发展与应用。

本次大会录用的学术论文将在核心期刊《系统仿真学报》(增刊)发表。会议将邀请国内外著名专家作专题报告，同时将举办科研成果和最新产品展示会，为各研究开发单位及有关厂商展示自己的成果、产品提供场所。欢迎大家积极投稿。

一、征文范围(包括但不限于)

建模技术、动画技术、可视化技术、多媒体技术、人机交互技术、虚拟制造、仿真技术、分布式系统、空间化声音、模式识别应用、图形平台、网络技术、遥操作技术、VRML 技术、逼真图形图像技术、增强现实、协同操作、数字博物馆、网络游戏、图象绘制技术、可视化地理信息系统、基于图像的视景生成技术、虚拟现实与可视化应用系统……

二、征文要求

1、论文未被其他会议、期刊录用或发表，不超过 10 页；2、要求接受电子投稿(同时提交 Word 与 Pdf 格式文件)；3、论文包含：题目、中英文摘要、正文、参考文献等；4、正式论文格式见论文录用通知；5、投稿者请在论文最后务必写清姓名、单位、通信地址、电话及 E-mail 地址。

三、重要日期

征文截止日期：2007 年 6 月 15 日(收到日期)

录用通知日期：2007 年 7 月 15 日(发出日期)

四、来稿联系方式(请注明 CCVRV07 征文)

联系单位：北京航空航天大学 6863 信箱(邮政编码：100083)

联 系 人：伍潇潇，胡勇，范志强，王正光　　电话：13161965259

电子邮件: ccvrv07@vrlab.buaa.edu.cn

五、会议网站

http://vrlab.buaa.edu.cn 欢迎上网查询大会各项文件和最新通知