区分移动实时事务关键性的并发控制策略

许亚梅[1],张立臣[2]
(1.广东工程职业技术学院 计算机信息系,广东 广州 510520;
2.广东工业大学 计算机学院,广东 广州 510090)

摘 要:在移动实时数据库系统中,事务的关键性代表实时事务的时间紧迫性,由于实时事务的紧迫程度不同,为了保证事务的时间正确性,因而有必要区分事务的关键性提出相应的并发控制策略。该文按满足截止时间的重要性将实时事务分为软、硬事务加以研究,在此基础上分别提出了三种区分事务关键性的优先级并发控制策略,对三种优先级策略进行了性能比较并提出了改进方法。研究结果表明,将实时事务按关键性分类有利于设计事务优先级分派及调度策略。

关键词:移动实时数据库;实时事务;关键性;截止时间;并发控制
中图分类号:TP31 **文献标识码**:A **文章编号**:1673-629X(2010)01-0090-03

Concurrency Control Strategy Distinguishing Real-time Transaction Criticality

XU Ya-mei[1], ZHANG Li-chen[2]
(1. Department of Computer Information, Guangdong polytechnic College, GuangZhou 510520, China;
2. Faculty of Com puter, Guangdong University of Technology, Guangzhou 510090, China)

Abstract: In mobile real-time database system, the criticality in real-time transaction is on behalf of the urgency of the time. Due to the urgency of real-time transaction in varying degrees, in order to ensure the time accuracy of the transaction, it is necessary to distinguish criticality of transaction and propose corresponding concurrency control strategy. tuY the real-time transaction which is divided into software and hardware transaction by the importance of meeting the deadline. Based on this, propose three priority concurrency control strategies to distinguish the criticality of transaction, compare the performance of three strategies and propose improved method. The results show that classifying real-time transaction is in favor of the designing strategy in transaction priority assignment and scheduling.

Key words: mobile real-time database; real-time transaction; criticality; dead-line time; concurrency control

0 引 言

移动实时数据库(Mobile Real-Time DataBase, MRTDB)是在移动计算环境下的分布式、实时数据库,其本质特征是实时处理、分布式计算和移动服务。移动实时数据库[1-3]是在移动计算环境下的实时数据库,事务处理是移动实时数据库中的关键问题,它解决在移动环境下保持数据一致性的问题。

在移动实时事务中除了要求数据一致性,还必须满足时间限制条件。传统数据库的并发控制并不适合实时数据库。许多依赖于事务阻塞或重启的实时并发控制方法使用优先考虑时间来迎合事务时间限制条件,但由于时间的不可预测,阻塞或重启都不是很恰当。此外实时数据库性能目标与传统性能目标有着根本的不同,它是使截止时间内完成事务的数量为最大而不是要求并发执行的事务数量保持为最大。对于一个给定的系统配置,决定实时性能的最基本因素是对数据存储进行调度的并发控制协议。该文将从实时事务的紧迫度与截止时间研究可行的并发控制协议。

1 移动实时事务按关键性分类

实时事务的关键性是指满足截止时间的重要性[4],将实时事务按关键性分类有利于设计事务优先级分派及调度策略。按关键性分类也即按事务时限的性质进行分类,可分为三种情况:

(1)硬事务:事务超过截止期将会导致不良后果,这类事务对应于安全危急性活动。

(2)软事务:事务超过截止期仍有一定的价值,但

收稿日期:2009-05-20;**修回日期**:2009-08-18
基金项目:国家自然科学基金资助项目(90818008,60774095);广东省自然科学基金资助项目(07001774)
作者简介:许亚梅(1972-),女,广东湛江人,硕士,研究方向为数据库、网络信息系统;张立臣,博士,教授,研究方向为实时系统。

价值不断下降,直到某一时刻(最终有效时间),其价值降到零,此后保持为零。

(3)固事务:一旦到达截止时间,事务价值立即降为零,此后固定为零。它是软实时事务在最终有效时间与截止时间重合情况的特例。超截止时间的事务称为过时事务。因此,文中研究的对象将把固事务归为特殊的软事务一并处理,而不单独研究。

图 1 描绘了硬、软、固实时事务的事务特性,其中,v,t 两坐标分别为价值函数与时间;d 为截止期;e 为"最终有效时间";r 为放行或启动时间,当 $t < r$ 时,$v(t) = 0$,表示在事务未准备好以前启动是无价值的。

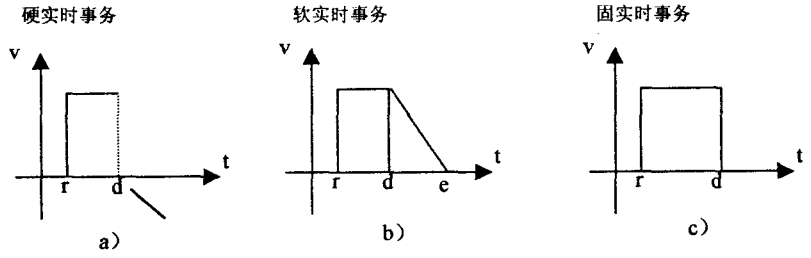


图 1　实时事务的关键性

从硬、软实时事务的特性可见,系统必须保证硬实时事务的截止期,否则该类事务的价值在截止期后为零甚至对系统产生一定的危害,而软事务超过截止期仍有一定的价值,但价值不断下降,直至到达最终有效时间时其价值降到零。因此,实时事务的并发控制策略不仅应考虑保证数据一致性,还应满足事务的截止期限要求,区别对待不同的关键性事务。

2　区分事务关键性的优先级并发控制策略

在移动实时数据库中,事务的关键性与截止时间是两个不同的概念:一个事务可能有很紧的截止时间,但超过它不会给系统造成很大的伤害[4]。紧迫度代表实时事务的时间紧迫性,截止时间表示实时事务的时间限制性,二者间不存在必然联系。一个事务被限制在一个短的截止时间内并不一定就代表该事务具有高的紧迫度,具有相同紧迫度的事务可以有不同的截止时间,截止时间一样的事务可以有不同的紧迫度。因此,具有高紧迫度的事务应该具有更大的价值[5]。

由于实时事务的紧迫程度不同,为了保证事务的时间正确性,因而有必要区分关键性对事务的并发控制提出相应的方法。由于固事务是软事务在最终有效时间与截止时间重合情况的特例,因此下面从硬事务、软事务的角度[6]研究并发控制策略。以下算法假定优先级的设定方式是动态设置的,当发生"优先级倒置"时,首先判断请求锁的事务 T_R 和持有锁的事务 T_H 是硬实时事务还是软实时事务,然后从以下三种情况分别考虑实时事务并发控制策略:

(1)T_R 是硬事务,T_H 是软事务。

此时考虑采用"高优先"法[7],文中记为 C_1 - HP,在事务执行过程中无条件夭折事务 T_H。该策略的算法为:

```
IF pr(T_R) > pr(T_H)
    aborts T_H; /* T_H 被夭折,而 T_R 执行 */
ELSE
    T_R waits;
ENDIF
```

(2) T_R 是软事务,T_H 是硬事务。

此时考虑 T_R 无条件等待,算法(C_2 - HP)如下:

```
IF pr(T_R) > pr(T_H) /* 事务的紧迫度占主导地位 */
    T_R waits;
    Pr(T_H) := pr(T_R); /* T_H 继承 T_R 的优先级 */
ENDIF
```

(3) T_R 和 T_H 都是硬事务或 T_R 和 T_H 都是软事务。

考虑对第(1)种情况的 C_1 - HP 作适当的改进:设 S_R 表示 T_R 的空闲时间,$E_H - P_H$ 表示 T_H 还需要执行的时间,若 T_H 即将完成,则继承 T_R 的优先级继续执行,T_R 等待;否则夭折 T_H,算法(C_3 - HP)如下:

```
IF pr(T_R) > pr(T_H)
    IF S_R >= E_H - P_H
        T_R waits;
      Pr(T_H) := Pr(T_R);
    ELSE
      aborts T_H;
    ENDIF
ELSE
    T_R waits;
ENDIF
```

3　并发控制策略的性能分析与改进方法

3.1　三种优先级策略的比较

对于策略(1),(2),即 T_R 是硬事务,T_H 是软事务或 T_R 是软事务,T_H 是硬事务:存在硬、软事务时,采取的措施是优先考虑硬实时事务,基本符合对硬实时事务的测度标准。具体策略上,采取了"高优先"法(C_1 - HP、C_2 - HP),夭折低优先级的事务而让高优先级事务执行。这种方法可以消除死锁,但也存在一定的问题:对那些已执行了很久时间的事务来说,夭折的代价较大,另外被夭折重启的事务可能会有更高的优先级,因而会导致循环夭折。

对于策略(3),即 T_R 和 T_H 同时是硬事务或 T_R 和

T_H 同时是软事务。

当硬事务与硬事务或软事务与软事务并存时，判断空闲时间与执行时间的大小，对事务进行有条件的夭折，该算法(C_3 - HP) 实际上综合了“优先级继承(PI)”法和“高优先”法(HP)。但该策略存在两个问题：一是假设只有一个事务 T_H 需要继续运行到完成，然而实际上可能存在 T_R 被阻塞的时间任意长，即形成阻塞链，那样的话，要使 T_R 不超过截止期，算法中比较的则不应是 $S_R >= E_H - P_H$，而是与链中所有事务比较，即：$S_R >= \sum(E_{H_i} - P_{H_i})$。这个问题如果不解决，那么会造成 T_R 在等待中，不自觉地超过截止期。二是 S_R 和 $E_H - P_H$ 的有效性，可能决定着该算法的成功率。

3.2 改进方法

(1)防止死锁问题的对策。

只要存在针对共享数据资源的大规模并发访问的情况，那么死锁是不可避免的。理论上，预防死锁的最好的途径是：给每一个事务设定一个优先级，同时确保较低优先级的事务不必等待较高优先级事务释放共享资源，反过来，也确保较高优先级的事务能够立刻取得相应的资源[8]。如果开发人员无法判别事务的优先级，那么可以考虑在每个事务开始时赋予它一个时间戳，依据时间戳的先后来判定事务的优先级。但是，假设由于网络故障而导致高优先级的事务无法提交或者回滚，那么是否其他低级别事务便要一直等待或者被抛弃？死锁出现的因果关系让我们意识到开发人员在设计数据库过程中，应尽可能地预见到各种并发访问的关联因素。

这些因素包括：

①尽可能缩短事务。在同一数据库系统中并发执行多个需要长时间运行的事务时，发生死锁的概率较大。事务运行时间越长，其持有锁的时间便越长，从而堵塞了其它活动并可能导致死锁。保持事务在一个批处理中，可以最小化事务的网络通信往返量，减少完成事务可能的延迟并释放锁。同时，涉及多个表的查询更新操作，若比较耗时，尽量不要放在一个事务内处理，能分割便分割。若不能分割，便尽可能使之在业务量较小的时间(例如子夜或者午餐时间)执行。

②尽可能按同一顺序访问数据对象。如果所有并发事务按同一顺序访问对象，则发生死锁的可能性会降低。

③ 避免编写包含用户交互的事务。因为运行没有用户交互的批处理的速度要远远快于用户手动响应查询的速度，若用户不能及时反馈，则此事务将挂起。因而将严重降低系统的吞吐量，因为事务持有的任何锁只有在事务提交或回滚时才会释放。即使不出现死锁的情况，访问同一资源的其它事务也会被阻塞，等待该事务完成。

④ 使用低隔离级别。确定事务是否能在更低的隔离级别上运行。执行提交读允许事务读取另一个事务已读取(未修改)的数据，而不必等待第一个事务完成。使用较低的隔离级别(例如提交读)而不使用较高的隔离级别(例如可串行读)可以缩短持有共享锁的时间，从而降低了锁定争夺。

⑤ 将经常更新的数据库和查询数据库分开。定期将不变的数据导入查询数据库中，这样查询和更新就可以分开进行，而降低死锁机率。

(2)改进死锁或阻塞链的算法。

为防止死锁和阻塞链的形成，考虑把 C_3 - HP 算法进行改进，基本思想是每一个数据设置一个“优先级顶”[7,9]，它定义为要存取该数据的事务的优先级最高者。一个事务要获得对一个数据的锁，它的优先级必须严格地高于当前由其他事务锁住的各数据的最高优先级顶(记为 HPC)，否则它就被锁住最高优先级顶的数据的事务(记为 T_{HPC})所阻塞。假定在每一事务执行就知道其要存取的数据，并且对每一数据维护一个包含要存取它的各事务及其优先级的表。

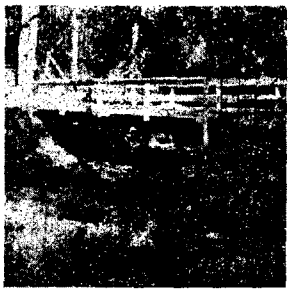
算法如下：

```
IF pr(T_R)>HPC
  T_R is granted to lock;/* T_R 获得了处理数据的锁 */
ELSE
  T_R waits;
  IF pr(T_R)>pr(T_HPC)
     THEN pr(T_HPC):= pr(T_R);
  ENDIF
ENDIF
```

移动实时数据库系统 MRTDBS(Mobile Real - TimeDatabase System)是运行在移动计算环境中的实时数据库系统。由于无线网络的频繁断接、不可靠性和不可预测等特点，使得移动实时事务特性不同于一般的实时事务，例如：事务有移动性；事务多为长事务；无线网络的通信延迟使得事务截止期更难满足等。因此，移动计算环境中的实时事务大多为软截止期事务，即允许事务有限的超截止期，而不会对系统造成危害[10]。

文献[11]提出了 SSRTD 协议，采用“扩展截止期”方法进行移动实时事务调度。文献[11]中“扩展截止期”根据软实时事务本身的截止期而定，是固定不变的，同时该方法它也能够有效支持无线网络的频繁断接。

(下转第 96 页)


(c1)椒盐攻击


(d1) 椒盐攻击


秘密图像(a)的还原图像


秘密图像(b)的还原图像

图 5 椒盐噪声攻击

4 结束语

讨论了一个具有对称性的四维动力系统的基本性质,并通过相图和最大李雅普诺夫指数判定该系统是否具有混沌性质。提出了一种新的数字图像隐藏算法。该算法利用 Arnold 变换做置乱,并用可逆矩阵做两幅图像的交叉分解,结合一个新的具有对称特性的四维混沌系统对秘密图像进一步加密,最后采用基于像素灰度值的位平面嵌入的隐藏方法做秘密图像的隐藏。该算法实现了秘密图像和载体的无损复原。实验结果表明,文中的算法具有很好的隐藏效果,具有较高的安全性和较好的抗攻击稳健性。

参考文献:

[1] 王兴元,骆 超. Lorenz 系统走向混沌的道路[J]. 大连理工大学学报,2006,46(4):582-587.

[2] Qi Guoyuan, Du Shengzhi, Chen Guanrong, et al. On a four-dimensional chaotic system[J]. Chaos, Solitions and Fractals, 2005,23:1671-1682.

[3] Nikolov S, Clodong S. Occurrence of regular, chaotic and hyperchaotic behavior in a family of modified Rossler hyperchaotic systems[J]. Chaos, Solitions and Fractals, 2004,22:407-431.

[4] Qi Guoyuan, Chen Guanrong. Analysis and circuit implementation of a new 4D chaotic system[J]. Physics letters A, 2006,352:386-397.

[5] 王兴元,王明军. 超混沌 Lorenz 系统[J]. 物理学报, 2007, 49(56):5136-5141.

[6] 郗坤洪,叶瑞松. 一种改进的基于混沌序列的图像加密算法[J]. 计算机技术与发展,2008,18:48-50.

[7] 眭新光,罗 慧. 基于矩阵分解的数字图像分存技术[J]. 计算机工程与应用, 2004,40(32):96-98.

[8] 王继军,张显全,张军洲,等,一种新的数字图像分存方法[J]. 计算机工程与应用, 2007,43(31):79-81.

(上接第 92 页)

4 结束语

移动实时数据库满足事务截止期的同时,还要考虑事务的紧迫度,这就要求实时事务在并发控制等处理技术方面有别于传统数据库的处理技术。

(1)给出了移动实时事务按关键性分类的方法;

(2)提出了区分事务关键性的优先级并发控制策略;

(3)对不同关键性的实时事务并发控制方法作了性能比较。

参考文献:

[1] 肖迎元,刘云生,廖国琼. 移动实时数据库系统综述[J]. 计算机工程与应用,2004(35):173-177.

[2] Lam K Y, Kuo T W, Tsang Wai Hung, et al. Concurrency control in mobile distributed real-time database systems[J]. Information Systems,2000(6):261-286.

[3] Liao Guo qiong, Liu Yun sheng, Yang Jin cai. A concurrency control mechanism for mobile real-time nested transactions [J]. Chinese Journal of Computers,2003(10):1326-1331.

[4] 刘云生. 现代数据库技术[M]. 北京:国防工业出版社, 2001.

[5] 韩 凯,王京春. 实时数据库调度策略综述[J]. 计算机工程与应用,2004(32):172-176.

[6] 魏志东,魏洪波. 面向混合实时数据库系统的调度与并发控制[J]. 计算机工程,2003,29(7):73-75.

[7] 何新贵,唐常杰,李 舜,等. 特种数据库技术[M]. 北京:科学出版社,2000.

[8] Xiao Qiao. 细化解析:不同类型数据库的死锁问题[DB/OL]. 2007. http://www.sudu.cn/info/html/edu/20070803/311782.html.

[9] Lam K Y, Kuo T W, Tsang Wai Hung. The Reduced Ceiling Protocol for Concurrency Control in Real-time Databases with Mixed Transactions[J]. The Computer Journal,2000,43(1):65-80.

[10] Lindstrom J. Distributed Optimistic Concurrency Control for Real-Time Database Systems[M]. Helsink: Helsinki Institute for Information Technology(HIIT), Advanced Research Unit (ARU),2000.

[11] 刘云生,王丽娜,廖国琼. 支持断接的移动实时事务调度[J]. 计算机科学,2005,32(4):155-158.