

一种工控领域内存数据库的设计与实现

王 焱¹, 徐新国¹, 朱廷劭²

(1 华北计算机系统工程研究所, 北京 100083; 2 中国科学院研究生院 信息科学与工程学院, 北京 100080)

摘 要: 设计与实现了一种工控领域的内存数据库, 给出了各模块的具体实现方法, 重点给出了适合实时数据库特点的实时事务管理的方法和流程. 在实际应用中, 能够稳定而高效的运行.

关键词: 内存数据库; 实时事务调度; 实时事务处理; 报警

中图分类号: TP392

文献标识码: A

文章编号: 1000-7180(2012)08-0098-04

A Design and Implenmentation of Real-time Memory Database in Industrial Field

WANG Yan¹, XU Xin-guo¹, ZHU Ting-shao²

(1 National Computer Systems Engineering Research Institute of China, Beijing 100083, China; 2 Institute of Information Science and Engineering, Graduate School of Chinese Academy of Sciences, Beijing 100080, China)

Abstract: The article gives a design and implementation of real-time memory database in industrial field. It gives the concrete implementation of each module and focus on the methods and process of real-time transaction management based on the characteristics of real-time database. In practical applications, it runs stable and efficient.

Key words: real-time memory database; real-time transaction management; real-time transaction process; alarm

1 引言

实时数据库目前广泛应用于工业自动化控制领域, 在工业过程控制, 电力系统中获得了越来越广泛的应用. 实时数据库由于工作版本常驻内存极大地提高了数据处理的性能. 内存库是实时数据库的主要组成部分, 它主要用来处理实时数据, 进行事务调度和报警, 与上层应用和安全控制部分交互. 实时数据库不同于传统的关系数据库, 实时数据库的数据和事务都是具有时间维度的, 而实时事务更是具有与传统事务不同的属性^[1-3]. 这些属性使得在对实时数据处理, 实时事务处理方面都不同于以往的方法. 本文根据工控行业的需要和实时数据库的特点, 设计与实现了一种用于工业实时控制的实时数据库中的内存库部分, 根据实时数据库的特点在实时数据结构、实时事务处理和实时事务调度都采用了特别的方法. 在实际应用中, 可以稳定、高效地处理实时

数据和用户的请求操作.

2 系统的结构

本文设计的内存库主要分为以下五个部分: 实时数据结构模块、事务管理模块、时间管理模块、内存管理模块和报警库模块. 内存库的配置文件中存储了内存库厂、设备、点的结构信息和静态值, 在数据库启动的时候加载. 实时数据结构模块就是厂、设备、点在内存中组织起来的数据结构, 以及它们的位号、名称和内存地址间的映射. 事务管理模块接收用户发送的操作指令, 创建事务, 并对实时事务进行优先级分配和调度执行. 时间管理模块主要用来配合事务调度进行, 为事务监测创建时间、执行时间、截止期等. 报警库模块, 主要负责监测、记录产生单点和组合点产生的报警信息, 以及实时事务执行失败的信息. 内存库模块, 负责对整个内存库的内存进行统一管理, 它为各个功能模块申请分配内存, 减少内

存碎片,提高内存利用率.同时,内存库与上层应用结合,由数据库的安全控制模块进行权限控制和安全防护,接收从控制设备发送的实时数据和应用层发送的操作请求,并进行处理和返回结果.内存数据库向下与归档历史数据库相连,经过内存库的处理后经过归档就归入历史数据库中.内存库的结构图如图1所示.

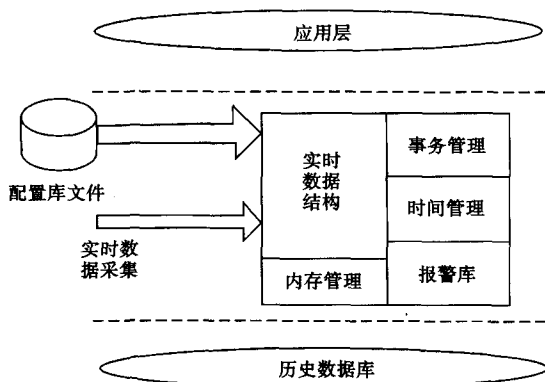


图1 内存库结构图

3 系统的设计和实现

3.1 实时数据结构模块

实时数据不同于传统关系数据库的静态数据的形式,实时数据是带有时间维度标签的,实时数据有一定的有效期,数据的价值随着时间的增长而减少^[2-4].实时数据的这个特性使得对于实时数据的处理采用不同的方法.

本文设计的实时数据结构分为三级,厂、设备和点,每个厂可以是一个独立的数据库,可以有若干个厂,一个厂又包含若干个设备,每个设备都包含模拟点和数字点这两个大类,两个类中的点的类型的数据结构都是不同的.每个点中包含静态点字段和动态的点字段,静态点字段主要是一些量程,报警的限制,优先级等.动态的点主要是处理从控制设备发送来的更新的实时值、时间等信息.以对应于不同的应用需求.除了单个的点,还定义了组合点,组合点是输入和输出连接在不同的I/O点上,是几个有逻辑关系的点的组合.

实时数据结构模块在数据库启动的时候加载配置文件,根据配置文件中关于厂、设备、点三级结构的信息建立一个关于厂、设备、点结点的二叉树的数据结构.同时建立厂、设备、点名称和位号的映射,以及位号与内存地址的映射.由于我们的操作主要是对于它们的内存地址进行的,我们用这种映射的模式可以最快速度进行查找.厂、设备、点的位号和名

称在整个数据库中都是唯一的,名称采用三级名称的模式.实时数据库的内存是由内存管理模块来统一进行管理的,内存管理模块可以根据需要增加内存的容量,同时尽可能小地减少内存碎片的产生.

3.2 事务管理模块

本文设计的事务管理模块是内存中的主要功能模块.实时数据库是实时特性和数据库相结合的系统,实时数据库的数据和事务都有显式的定时限制,正确性不仅依赖于事务产生的逻辑结果,更依赖于逻辑结果产生的时间^[1].所以,实时事务不同于传统事务所具有的ACID属性,它有自己的属性,比如说实时事务的定时特性、复杂性、不可逆性、可预报性等.实时数据库事务不需要像传统事务一样具有那么严格的ACID属性,它最重要的属性就是它有定时限制的要求,实时事务都有截止期,有紧急程度.实时事务的结构和事务间的关系也更为复杂,常常有嵌套事务、父子事务等等^[3].实时事务对于事务的恢复不需要像传统事务一样严格,因为时间标签对于某些事务的恢复是无意义和无价值的.实时事务通常还要有可预报性,比如常常需要对于事务执行时间进行预估.

实时事务的这些属性使得实时事务在事务分类、事务调度、事务恢复上都有所不同.实时事务的调度需要考虑到实时事务截止期、紧迫程度,优先顺序等.实时事务的恢复要考虑到这种恢复是不是有价值,以及恢复对于本事务和其他等待执行事务的影响.

一般实时数据库的事务按照紧迫程度,分为硬实时事务、软实时事务和固实时事务^[3].按照事务执行周期,分为周期性事务、非周期性的事务和随机事务.在功能上,不同的实时数据库对于特定的功能操作都是不同的.本文的实时数据库事务分为三类,硬实时事务、软实时事务和非实时事务.当然很多事务也是周期事务,比如实时数据的采集.硬实时事务是那些实时性对于数据库影响比较大的事务,软实时事务其次.

实时事务在调度时分为两种队列、等待队列和执行队列.硬实时事务可以抢占正在执行的软实时事务而优先执行,软实时事务按照最小空余时间算法^[5-7]按照优先级排队等待.当然,系统中产生硬实时事务是很频繁的,但是不至于都是硬实时事务而影响了软实时事务的正常执行.如果实时事务等待的时间过长,比如新的数据已经到来或者事务的截止期到达了,那就失去了执行的价值,那就抛弃这个

事务并向报警库中产生一个报警信息。

由于对于实时数据库会产生大量的操作,有大量的事务需要处理,为了避免频繁的事务创建,销毁而耗费系统资源和时间,根据系统的需要,设计了一种高效的事务调度器,事务调度器中预先设置好等待执行事务的线程,事务调度器根据上面的调度的策略进行调度执行,执行完毕后重新回到等待的状态。根据对于时间的管理,在某些时刻需要大量处理事务时扩大事务调度器的容量;在比较空闲的时刻,缩小事务调度器的容量。事务调度器也方便了我们对于事务的操作内容的修改,我们只用把需要执行的事务逻辑功能传给调度器,调度器就会有效地运行,比较灵活。事务调度器的结构如图 2 所示。

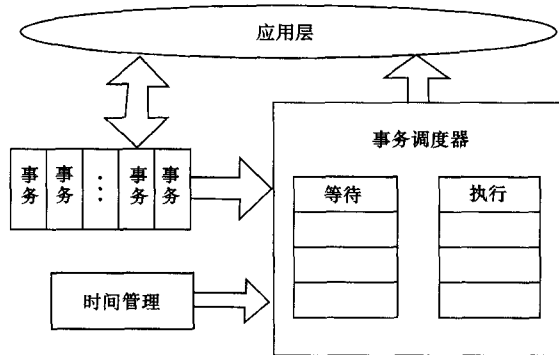


图 2 事务调度器结构图

对于实时事务来说,并不同于传统事务的原子策略的情况。我们要充分考虑到事务的恢复而耗费的时间和资源对于正在执行事务和等待的其它事务的影响^[8]。实时事务保证原子性策略一般有以下三种:一种是将事务恢复到原状态,一种是只产生报警信息,最后一种是什么也不做。根据上面定义的事务,我们的原子策略有两种,对于实时数据的操作由于有时间限制,所以如果事务失败,就放弃此实时事务,同时产生报警信息到报警库中。另一种是我们将事务恢复到原状态,但是要对恢复需要的时间进行预估,以此来确定带来影响的大小,如果对于其它事务影响不是太大就考虑恢复。事务回滚,采用内存私有日志的方法,每个执行的事务都有一个私有日志记录自己的操作。整个内存数据库有一个全局日志。私有日志在事务提交后都提交给全局日志。全局日志根据内部检查点,定时将全局日志缓冲区刷新到磁盘中。定义的私有日志是在内存中数据结构,里面保存有操作对象的地址信息,标识信息等。

本文采用加锁方式来保证事务的并发性,定义了两种锁:共享锁和排它锁。在需要对点进行修改,

也就是对点进行写操作的时候,我们对点进行加锁,避免其他操作访问同一个数据结构的内容所来的并发问题。当然事务的并发性一定会给事务的处理带来影响^[9],但是这种代价的付出从目前来说也是不可避免的。系统可以容许数据库的状态在短时间内的不一致性,用以提高事务的并发度,满足事务的截止期。

我们对于事务的处理流程是这样的,根据上层应用对于数据库的操作产生事务,将事务放入事务调度器,如果是硬实时的事务就抢占正在执行的软实时事务而优先执行,如果是软实时事务就根据优先级分配算法,动态的调整优先级并且按照优先级进行排队。执行事务时结合并发策略,在需要写操作的地方加锁。同时私有日志记录下重要的操作。最后,事务执行完毕,就把私有日志写入全局日志等待写回磁盘。如果执行失败或者执行时间超时,如果有恢复价值的事务就执行回滚,将操作对象回滚到起始的状态。事务处理大致流程如图 3 所示。

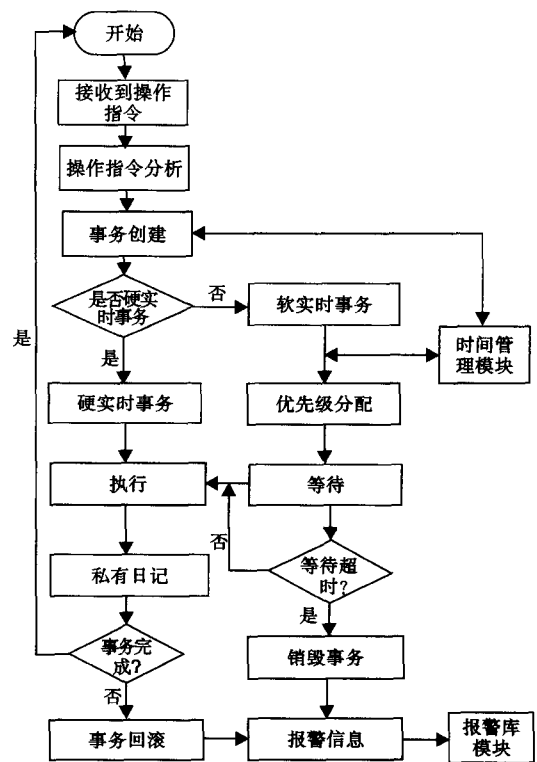


图 3 事务处理流程图

3.3 时间管理模块和报警库模块

时间管理模块是本文为了满足实时系统对于时间的要求而设计的独立模块,时间管理模块跟事务管理模块有着密切联系,它为事务管理模块在优先级分配时提供事务的到达时间,系统的当前时刻,计

算出事务的截止期,为低实时事务执行的先后提供了重要参考.同时,时间管理模块有监视线程,在事务调度器繁忙的时候扩充调度器的容量,在事务调度器空闲的时候,根据事务平均运行时间,减少不必要的容量.

报警库模块也是内存库中重要的部分,负责为单点、组合点以及事务异常报警.根据我们定义的模拟点的报警上下限,报警上上限、报警下下限,数字量的正常值,分为四种类型的报警,包括限值报警、变化率报警、偏差报警和数字量的变位报警.限值报警是对于点的当前值超过报警限值产生的报警,变化率报警是对于模拟量在单位时间内变化量超过设定值产生的报警,偏差报警是对于模拟量的值相对于设定值上下波动范围超过一定量时产生的报警条件.数字量的变位报警主要是由于数字量只有0或1两种状态,只有一种状态是正常状态的报警.对于组合点的报警,我们采用一种倒排索引的方法,计算设定的组合点的当前值,并根据报警条件进行报警.

报警库模块从内存数据库更新后开始检查点的实时当前值是否满足报警条件,如果产生报警条件,报警库就通知上层用户界面进行处理,并将异常值和时间信息等写入报警缓冲区,当报警缓冲区即将写满时再整体刷新到磁盘的报警文件中.

4 结束语

本文根据工控系统的需求设计了满足实时性的内存库的各个模块,根据实时数据库的特性,设计出满足实时数据库特性的实时数据结构、实时事务调度和实时事务处理的方法.在实际应用中,能够快速

有效的处理实时数据,能够对于内部产生的实时事务进行正确和有效的调度,对于异常值进行及时的报警和处理.

参考文献:

- [1] 刘云生.实时数据库系统[J].计算机科学,1994(3):21-26.
- [2] 刘云生,李国徽.实时数据库数据特征对事务处理的影响[J].计算机研究与发展,1999,36(3):364-368.
- [3] 刘云生.关于实时数据库事务[J].软件学报,1995,6(10):614-622.
- [4] 张武. ATRs-EDB 系统的实时数据定义与操作[D]. 武汉:华中科技大学,2010.
- [5] 王成光.流程工业大型实时数据库理论、技术与应用[D].浙江:浙江大学,2003.
- [6] 韩凯.实时数据库的算法研究与系统开发[D].北京:清华大学,2005.
- [7] 刘志华.实时数据库内核[D].河北:华北电力大学,2005.
- [8] 王永炎.实时事务并发控制算法优化[D].北京:中国科学院软件研究所,2004.
- [9] 廖国琼.嵌入式实时数据库 ARTs-EDB 事务调度实现技术[D].武汉:华中科技大学,2005.

作者简介:

王焱 男,(1985-),硕士研究生.研究方向为实时系统与实时数据库.

徐新国 男,(1966-),博士,高工.研究方向为实时系统与实时数据库.

朱廷劭 男,(1972-),博士,教授.研究方向为数据挖掘与计算网络心理学.