

# 实时数据库系统关键技术及实现<sup>\*</sup>

叶建位, 苏宏业

(浙江大学 先进控制研究所 工业控制技术国家重点实验室, 浙江 杭州 310027)

**摘要:** 简要地论述实时数据库的基本概念和特点, 分析了实时数据库与传统关系数据库的差异。同时, 重点研究了基于 Windows 平台下实时数据库的实现中关键技术的设计思想和技术路线。根据上述设计思想研制开发的 ESP-iSYS 实时数据库已经在多个工业控制现场得到成功应用。

**关键词:** 实时数据库; 组件技术; 数据高效存取技术; 事务调度模式

中图法分类号: TP311.13 文献标识码: A 文章编号: 1001-3695(2005)03-0045-03

## Key Technology of Real-Time DataBase System

YE Jian-wei, SU Hong-ye

(National Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Hangzhou Zhejiang 310027, China)

**Abstract:** The conception and characteristic of real-time database and the difference between real-time database and relational database are discussed briefly. The philosophy of design and the key technology to implement real-time database on Windows platform are studied in this paper. ESP-iSYS system, which is developed according to technology above, has been applied to several practical projects successfully.

**Key words:** Real-Time DataBase (RTDB); Component Technology; High Performance Data Access; Task Scheduling

### 1 引言

过程工业在过去的十几年内得到了巨大的发展, 其中重要的改进是通过将已经得到广泛应用的集散控制系统 (Distributed Control System) 与新兴的计算机技术进行结合, 实现过程控制逐渐从底层向上层渗透, 形成集控制、优化、管理、经营于一体的综合自动化新模式<sup>[1]</sup>。但是由于技术发展的原因, 控制网络上分布着不同厂商的不同种类的监控系统, 每个系统只能采集和管理相应装置或设备在运行过程中产生的部分实时数据。因此迫切需要一个统一的数据平台来集成各种控制系统的数据库, 同时该平台应当为先进控制、实时在线优化和生产执行系统 (Manufacturing Executing System, MES) 等提供实时数据的支持<sup>[2]</sup>。

实时数据库系统 (Real-Time DataBase System, RTDBS) 正是为实现此类需求而产生的。近年来实时数据库系统的应用成功地实现了现场控制系统过程生产数据的集成, 建立了企业管理系统与底层控制系统之间的数据通信, 为全厂过程实时信息的综合集成奠定了基础。

### 2 实时数据库系统特点

传统的关系数据库系统长于处理稳定的数据, 强调维护数据的完整性、一致性, 其性能目标是高系统吞吐量和低代价, 但

对处理的定时限制没有严格要求。而传统的实时系统 (RTS) 虽然支持任务的定时限制, 但它针对的是结构关系简单、稳定不变和可预测的数据, 不涉及维护大量共享数据及它们的完整性和一致性, 尤其是时间一致性。

实时数据库的数据和事务均有显式的时间限制, 系统的正确性不仅依赖于事务的逻辑结果, 而且依赖于该逻辑结果所产生的时间<sup>[3]</sup>。实时数据库不是数据库和实时系统的简单结合, 它需要在数据模型、体系结构、事务处理模式、数据存储方式等诸多方面重新进行研究和开发<sup>[4]</sup>。由此可知实时数据库在设计 and 实现时需要注意以下几个方面:

(1) 实时性。RTDBS 作为外部系统的一个客观反映, 它表示了外部系统的当前状态, 只有数据与外部系统的实际情况相吻合时, 数据才有意义。所以要求 RTDBS 必须高效, 能够实现实时反应。

(2) 容错性。由于工业控制现场的情况复杂, 各种干扰较为常见, 可能导致采集的数据被污染。这就要求 RTDBS 须具备一定的容错性, 防止出现数据败坏 (Data Corrupt)。

(3) 稳定性。任何数据库系统都要求稳定性, 但由于现实中直接基于 RTDBS 的应用往往一样强调实时性 (基于 RTDBS 的典型应用如先进控制软件和在线实时数据优化等), 所以系统的稳定性被提到了更高的高度, RTDBS 是绝对不能轻易重新启动的。

(4) 鲁棒性。过程控制应用中, RTDBS 多应用于分布式环境与多个数据源连接, 工业现场的环境容易导致个别数据源出现数据流波峰或者通信受阻。因此 RTDBS 必须能承受数据流量冲击保证系统的实时性和稳定性。

### 3 实时数据库系统关键技术

#### 3.1 基于组件技术的模块化体系

组件模型(Component)是构造二进制兼容软件的规范,利用组件技术构建的软件具备更好的可伸缩性和可扩展性,并且组件模型能方便地解决分布式环境中的共享和协作问题。基于 Windows 平台的组件技术是微软公司力推的 COM/DCOM 技术。COM/DCOM 作为软件组件之间互相通信的一种标准,可以实现二进制兼容和位置透明(即无论对方位于另外一个进程甚至另外一个机器都可以透明地进行通信)<sup>[5]</sup>。

实时数据库中核心业务封装在不同的独立进程 COM 服务器(Out-Of-Process COM Server)中,彼此之间通过 COM 进行接口调用而不是简单的数据交换。COM 所提供的 LPC(Local Procedure Calls)通道经过特别的优化,在执行效率和数据通信带宽方面均优于常规 IPC(Interprocess Communications)方式,使得实时数据库的实时性得以保证。而且 COM 本身提供了线程管理模式,使得核心业务模块可以避免手动处理复杂的线程同步和数据共享,所有的接口访问都被自动同步和序列化(Serialize),保证数据安全并且防止访问冲突。同时,当前在数据通信接口方面 OPC 规范日益流行,并已经成为工业控制界的事实标准,大量设备及控制系统均采用 OPC 接口,而 OPC 规范本身就是基于微软的 COM/DCOM 技术设计的。实时数据库的数据采集模块同样采用 COM/DCOM 技术,能够实现与底层 OPC 服务器无缝整合(Seamless Integration),使得实时数据库可以方便地实现分布式构架而不必手动处理网络通信。

#### 3.2 被动式内核结构

实时数据库为了保证鲁棒性和稳定性,采用类似操作系统中内核体系的结构,并将各核心模块尽量独立,即便特定模块出现故障系统也可以自动重新启动该模块并且将系统恢复到故障前的状态。由于将内核模块与直接面向用户的常规业务应用(如流程图、趋势图、系统组态、数据查询和报表等)以及数据采集模块(从底层数据源采集数据,如 OPC 接口、DDE 接口、TCP/IP 接口、RS-232 接口等)隔离开来,大大提高了系统的稳定性和鲁棒性。并且内核服务全部注册成系统服务运行,这种方式至少具备以下优点:

- (1) 即便服务器没有完成登录,系统服务已经可以运行,该情况在出现系统由于硬件故障或断电后重启时尤其有用;
- (2) 可以让低级别的用户(如操作员)以非管理员身份在服务器登录并且进行操作而不至于因为登录用户权限不足导致系统无法运行;
- (3) 系统服务只能运行一个实例;
- (4) 系统服务方便管理员进行控制,可以避开常规 COM 应用计数来控制服务器的启动和停止。

事实上采用系统服务方式运行数据库系统核心的做法几乎是所有大型数据库系统(典型的如 Oracle 和 OSISoft PI)的标准方式。

内核模块采用被动模式工作,即内核模块本身不主动从数据源读取数据,而是通过数据采集模块完成数据读取并且将数据 Push 到内核模块中,并且外部应用则通过主动读取内核模块的方式获得数据,内核模块本身不提供主动数据通知服务。

同时,将数据采集模块管理功能隔离在独立的内核模块,当某个数据采集模块出现故障时可以自动重新启动该采集模块。虽然这种结构存在着降低系统实时性的风险,但是可以将内核模块隔离在复杂的现场环境之外,即便由于数据源本身行为失当导致对应的数据采集模块崩溃也不至于危及到其他数据采集模块以及数据库系统内核服务。图 1 描述了实时数据库内核体系的详细结构。

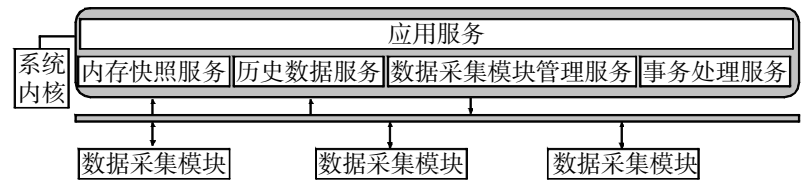


图 1 实时数据库内核体系结构

#### 3.3 高效数据存取技术

实时数据库必须实现高效的数据存储,才能在应对海量数据的同时保证系统的实时性。过程工业中以位号方式标志数据,与此相应,位号是实时数据库中数据存储的基本单位。实际应用中,实时数据库中往往有上万甚至十几万个位号,如何在这种情况下实现高效率数据存取是保证实时性的关键。在实时数据库中实现高效数据存取必须解决以下三点:

- (1) 快速位号检索。能够在上万或更多位号中快速定位指定的位号。
- (2) 访问同步以及数据共享。考虑到实时性,不能单纯地采用锁定来处理并发访问,同时又要确保数据一致性。
- (3) 高速数据读写。能够快速存取数据。

##### 3.3.1 快速位号检索

实时数据库必须面对海量的位号,位号的检索速度对于保证实时数据库实时性至关重要。工程经验表明实际应用中大量外围业务和应用软件均会频繁存取位号,大量 CPU 时间被用于检索位号。

工业控制领域使用名称标志位号,但字符串处理速度极慢,所以实时数据库为每个组态的位号分配一个 ID(64 位整数),并保证 ID 不重复(即便位号被删除,其 ID 也不能被复用,64 位整数足够长可以确保在正常的操作模式下不会用尽)。测试及现场运行证明基于 ID 的检索比基于字符串的检索速度快一个数量级以上。

系统在创建位号实例时动态地为新生成的位号分配一个索引号(32 位整数,不是 ID)。该索引从 0 开始按位号实例的创建次序递增,当组态删除该位号时,位号的实例并没有被释放而是进入了回收池。在回收池中的位号其 ID 及其他状态被复位但索引号保持不变,用户组态新位号时回收池中的位号被移出重新加以利用。这种方式可以避免频繁的内存分配和释放,另外可利用动态分配的索引号实现高速检索。

实时数据库为加速位号的检索,采用了双索引结构。主索引数据结构为平衡二叉树,以位号的 ID 作为标志,树的叶子内容为指向位号实例的指针。另外一套为辅助索引,其数据结构为可扩展线性数组,以位号的索引号作为数组下标,数组单元内容为指向位号实例的指针。结合上述位号回收利用机制,数组长度总是等于系统中已经组态位号的数目,而不至于无限膨胀。外围业务及应用首先根据位号的 ID 从主索引中取得位号的索引号,然后利用位号的索引号能够在辅助索引中快速定位位号实例(定位位号仅需一个跳转指令,速度极快且每次检索

时间恒定)。采用双索引结构的代价是系统必须保证两套索引的一致性,为此实时数据库在组态过程中增加了校验功能用于防止出现索引混乱。

### 3.3.2 访问同步及数据共享

实时数据库需要面对大量并发访问,而同时数据采集模块读取的大量数据也需要更新到内核模块。如果为防止访问冲突而简单地采用整体锁定方式,势必导致效率急剧下降,难以保证实时性。另外 RTDBS 需要处理的位号数量巨大,如果为每个位号单独分配一把锁进行同步同样会导致效率低下,这种方式更加不利的一面是造成过度的资源消耗。

负责数据服务的内核模块(内存快照服务和历史数据服务)均采用 COM 的 MTA (Multi-Threaded Apartments) 模式,允许许多客户端并发访问,由于大型 RTDBS 运行的服务器多为 MPS (Multi-Processor System) 系统,采用 MTA 模式可较好地提高并发访问性能。注意到工业控制中多数位号都是从属于特定数据源的事实,实时数据库依照数据源对系统中的位号进行分组,每个组持有一个共享锁。当特定数据采集模块完成从指定数据源数据扫描后即试图锁定该组位号,然后进行批量更新,锁定后客户端无法读取该组位号中的任何一个。当批量数据更新完成以后即解除共享锁,此时多个客户端可同时访问该组中任何位号。如果必要还可以在数据源分组基础上进一步分组以提高效率,在此不作详细讨论。

### 3.3.3 高速数据读写

无论中间过程如何,实时数据库最终都需要从存储介质中存取位号的数据。为实现高速数据读写所需的数据应当驻留内存,但物理内存容量有限而实时数据库所处理的数据极其庞大(尤其是位号的历史数据,有时需要保持位号长达数月的历史数据),所以无法将所有的数据驻留在内存。

考虑到实际应用中位号的即时值(又称实时数据)访问频率极高,实时数据库将所有位号即时值的管理独立成内存快照服务,该模块属于系统内核。内存快照根据位号的数目预分配大块内存进行优化管理,可以有效应对大规模并发访问。

对于位号的历史数据,由于总体容量巨大而必须将大部分转入磁盘存储器。但是磁盘的 I/O 速度大大慢于内存(速度相差几个数量级以上),为提高存储效率必须采用缓冲技术。实时数据库对于数据请求采用可组态缓冲技术,缺省的缓冲管理策略采用优先级机制。系统建立优先级队列管理缓冲区中的缓冲块,依据每个缓冲块的命中次数动态调整优先级,在队列达到指定长度时淘汰优先级最低的缓冲块。同时采用智能预读技术,可利用系统的空闲时间将位号的数据预先载入到缓冲区。结合智能预读与缓冲技术,能够极大地提高系统对数据请求的响应速度。

### 3.4 事务调度模式

实时数据库不同于传统的关系数据库,其数据结构和事务模型均相对简单,但是同样提出了对于实时性的要求,需要系统作出快速的响应。鉴于实时数据库需要同时处理多个事务的现实,系统必须具备良好的事务调度模式使得各个事务均得到最快的处理,为此必须实现多事务并发处理及负载平衡。另外事务本身具备优先级,如何在并发处理中进行优先级的调度也是设计中需要考虑的。

实时数据库内部采用线程池(Threads Pool)技术来实现多个事务的并发处理,并且根据每个线程处理事务时候所花费的

时间计算权值,系统根据每个线程的权值实现负载平衡。作为系统内核的事务处理服务,内部持有工作线程池,缺省的线程总数为 CPU 数目的两倍,所有的线程均在系统初始化时被创建,然后系统将它们挂起等待处理事务。

当事务被触发时,系统调用事务处理服务的接口函数,由于采用了 MTA 模式,COM 会自动地从内置线程池(此处所指不是实时数据库的线程池,而是 COM 库为 RPC Channel 所准备的线程池)选取合适的线程进入事务处理模块,多个事务可以同时进入而不至于被阻塞。在事务处理服务模块中,首先根据各个线程的权值以及事务本身的优先级从线程池中选取最合适的处理线程并激活该线程,然后将事务打包并发送到该线程处理并立即返回。每个工作线程均装备事务队列,发送过来的事务根据其优先级插入队列中合适的位置,高优先级的任务总是位于前面,工作线程依次处理这些事务,这样可以确保事务优先级。

工程实践验证了以上模型的可靠性,在处理多用户访问和多任务(实时数据库支持脚本语言编写的任务)并发的时候表现出较好的性能和稳定性。ESP-iSYS 实时数据库的负荷测试总共组态 5 000 个位号,测试设定平均每秒有 150 个以上脚本编写的内部任务同时执行,系统仍然保持稳定并能够迅速响应 30 个以上的客户端通过网络方式并发访问。对比国外先进的实时数据库系统如 OSISoft PI, ESP-iSYS 实时数据库在处理多任务及并发访问方面具有更好的表现。

## 4 结束语

随着过程工业信息化改造进程的不断推进,实时数据库作为全厂信息集成的基础平台必将越来越受到重视。依托实时数据库所实现的先进控制、全厂物料平衡、数据挖掘等应用已经且将继续为企业创造价值。

本文分析了实时数据库的特点以及在设计中面临的难点,对实现实时数据库所需要的关键技术进行了详细的阐述并给出解决方案。基于本文思路开发的 ESP-iSYS 实时数据库已经在多个工业控制现场得到成功应用,基于 ESP-iSYS 实时数据库进行的先进控制、生产管理等系统的实施均取得了良好的应用效果。

### 参考文献:

- [1] 王成光. 流程工业大型实时数据库理论·技术与应用[D]. 杭州: 浙江大学, 2003.
- [2] 林伟璐, 钱宇, 李秀喜, 等. 化工过程集成运行系统的研究[J]. 化工自动化及仪表, 2000, 27(1): 1-3.
- [3] Rohan F M Aranha, Venkatesh Ganti, Srinivasa Narayanan, *et al.* Implementation of A Real-Time DataBase System[J]. Information System, 1996, 21(1): 55-74.
- [4] 刘云生, 卢炎生, 王道忠. 实时数据库系统(RTDBS)及其特征[J]. 华中理工大学学报, 1994, 22(6): 66-70.
- [5] 潘爱民. COM原理与应用[M]. 北京: 清华大学出版社, 1999.
- [6] 刘云生. 现代数据库技术[M]. 北京: 国防工业出版社, 2001.
- [7] Ben Kao, Hector Garcia-Molina. An Overview of Real-Time DataBase Systems[M]. NJ, USA: Prentice-Hall, Inc, Upper Saddle River, 1995. 463-486.

### 作者简介:

叶建位(1978-), 男, 浙江台州人, 硕士研究生, 主要研究方向为大型实时数据库整体构建、实时数据库实现; 苏宏业(1969-), 男, 江苏常州人, 教授, 博士生导师, 主要研究方向为时滞、非线性系统控制理论与应用、鲁棒、变结构控制理论与应用。