

文章编号: 1005-0523(2007)02-0096-04

分布式工控实时数据库的设计与实现

王安, 许哲君

(西北工业大学 自动化学院 陕西 西安 710072)

摘要:在工控领域,实时数据库的设计已成为企业现代化生产所急需解决的重要问题,该文提出了一种分布式工控实时数据库的设计,并根据内存数据库的特点,提出了其中关键问题的解决方法。

关键词:实时数据库;事务调度;内存数据库;interbase

中图分类号:TP311.131

文献标识码:A

1 引言

随着现代工业的飞速发展,现代企业的生产方式已趋向国际化、分布化。工业实时数据库作为工业控制软件的基础,处于各功能模块数据交换的中心,其性能决定了整个组态软件的可用性。本文根据现代化工业控制的要求,提出一种开放式组态实时数据库的设计思路。

2 系统整体设计

该数据库采用实时数据库与历史数据库相结合的方式,支持B/S和C/S两种结构,具有更好的开放性和数据处理能力。

2.1 整体结构

数据库分为四层:物理设备层,数据接口层,数据管理层和客户应用层,如图1所示:

2.2 系统功能介绍

(1)物理设备层,由各种现场设备组成,通过丰富的现场数据采集接口,与现场设备与PLC,DCS进行通信,为整个系统提供统一的实时数据管理平台。

(2)数据接口层,物理设备层通过数据接口层与数据库管理系统进行数据交换,创造内存缓冲区,当有多个数据写入时,按照优先级安排顺序。

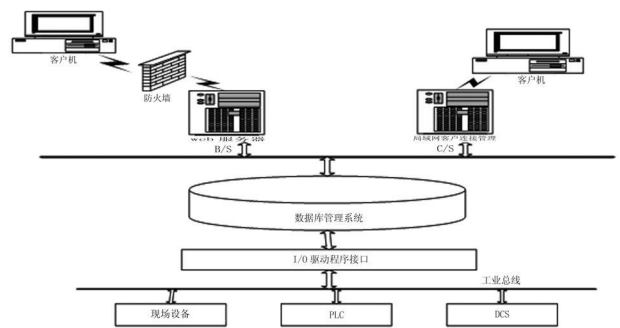


图1 数据库结构图

(3)数据库管理层,整个系统的核心和关键部分,有效地实现对实时数据的采集、处理并对大量的历史数据进行压缩,整理,存储,为系统的运行提供高效稳定的保障。

其中,实时数据库负责接收数据接口层传递的数据、分析调度、图形显示事物、报警事务、生成日志等,包括:对采集的实时数据进行合法性和一致性检查,对异常进行报警和适当处理,按照设定的方案控制设备的运行状态;实时数据可以通过API或类SQL方式所提供的类似于关系数据库的方式进行检索;将有严格时间要求的实时任务放在服务器上,由实时数据库统一调度管理;提供丰富的触发和定时机制,供各类数据处理,先进控制和优化算法使用;为保障数据一致性和时间连续性,提供实时数据的备份、转存、日志机制。

历史数据库完成历史数据的管理, 包括生成历史曲线和趋势曲线、数据查找、恢复、数据库备份及并发控制等. 它的存储、查询机制或者通过第三方大型关系数据库进行管理; 通过接口支持, 提供 ODBC、OPC 等接口, 便于第三方软件集成.

(4) 客户应用层, 主要指实时数据库浏览软件及对不同服务设备和应用环境进行配置和实时监控的组态软件等软构件在实时数据库服务层的实时应用. 客户可以通过 B/S, 从不同的地点, 以不同的接入方式对数据管理系统进行访问和操作. 这就为现代工业跨地域的信息查询和监控带来了方便, 也可以通过 C/S 访问服务器. 它是一种交互性强, 具有安全的存取模式, 响应速度快, 利于处理大量数据. 这充分适应现代工业海量数据的特点.

3 数据库管理系统的设计

3.1 数据库结构

数据库管理系统由实时数据库和历史数据库组成. 如图 2 所示.

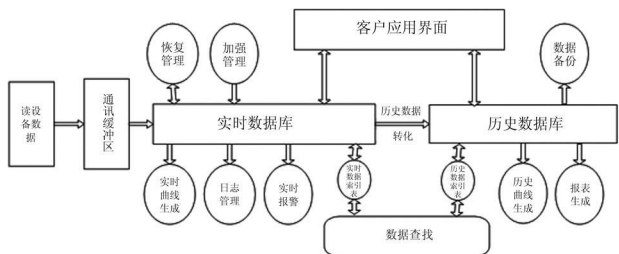


图 2 数据库管理系统结构图

实时数据库, 即事务与数据均可能有定时特性(截至期)的数据库. 工控软件需要处理大量实时事物, 应保证数据的读取速度, 存取灵活, 易于各个功能模块间的数据共享, 因此选用内存数据库(MMDB)来支持实时事务的实现, 它将数据驻留在内存当中, 消除了传统的磁盘数据库事物运行的 I/O 瓶颈. 从而获得 CPU 直接访问数据库数据的极高存取速度, 极大地提高了系统性能和吞吐量.

历史数据库我们选用 interbase 数据库. 这是安装最快最简单的大型数据库, 提供强大的自动化和人工智能, 可以自行管理、配置数据库以及日志数据库. 系统将需要频繁读取的组态信息和实时数据存储在内存中, 实时数据的处理也在内存中完成. 一段时间后, 将失去实时意义的数据成批地写入历史数据库中.

3.2 内存数据库技术实现

3.2.1 数据的存储结构

内存数据库的存储空间由 4 层组成, 如图 3 所示:

其中 M1 和 M2 分别为易失内存和不易失内存, M3 和 M4 分别为磁盘存储器 and 档案式存储器. 在整个 MMDB 中, M1 是主存储器, 它存放支持各事务的工作数据. 它由事务自接存取, 一般事务也只与它打交道. 由于 M1 易失性, 需要一定量的不易失内存 M2 作为临时的固定存储器, 用以存储部分活动的临时性数据. 其中的数据再依其永久性要求的不同决定是否移到磁盘存储器 M3 上. 当内存放不下整个数据库时, 磁盘存储器 M3 不仅用来存放不在内存的数据库部分, 还要存放用作恢复的数据库备份. 档案式存储器 M4 一般为磁带, 是脱机的, 它来存储数据库以前某一时刻完整状态的映象, 负责安全保护和长期档案保存目的.

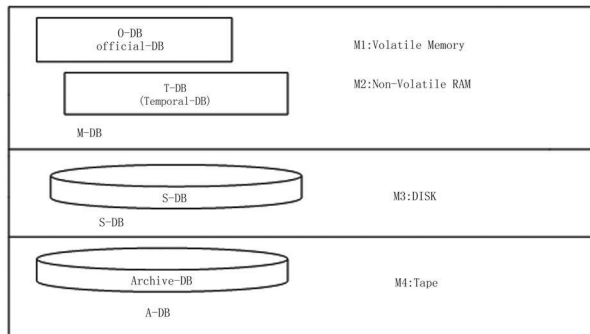


图 3 内存数据库存储结构图

3.2.2 数据的索引机制

数据的存取我们采用网格+T 树的方法. 这是一种典型的基于哈希的存取方法. 将整个记录集按照某种范围划分到不同的网格单元中, 每个网格单元都通过网格目录与一个装载记录的数据桶相联系, 每个单元对应着一个数据桶. 对于内存数据库将这些数据单元按某一字段建立一棵 T 树, 从而使每个桶的大小变为可动态分配的. 网格目录通常由一个 K 维(关键字的维数)数组来表示, 这些数组单元所代表的字段范围称为刻度. 查询非常简单. 当进行精确查询时, 首先用这些刻度来定位要查找记录所在的网格, 然后通过网格目录找到要查询的 T 树, 并按 T 树搜索方式进行搜索.

3.2.3 实时事务的分类及其调度

对于实时数据库而言, 达到最大的事务的吞吐量和最短的事务平均响应时间是追求的目标. 因此, 内存数据库需要提供统一的方法来保证数据的时间性和一致性. 系统把所有操作封装为两大类: 实时优

先任务和常规任务。

(1)实时数据采集、实时报警、实时数据查询和分析,这些任务都是针对实时数据的,本身具有较高的实时要求。由于数据模型的特殊性和实时数据存储、索引的高效性,这些任务的执行时间短,速度快,可以准确地预测运行时间,因此它们运行在较高的优先级上,运行时不可被中断和回滚。这类任务叫做实时任务,在系统中是预定义的。

(2)实时数据向历史数据的转化、远程客户应用的数据访问等任务没有严格的时间限制,超过了时间限制不会产生重要影响。这些任务在系统运行时随机发生,运行在较低的优先级上,可被中断,重起和回滚。把任务叫做常规任务。

在任务调度上,将实时任务和常规任务分别进行串行化,形成两个调度序列。在调度时优先执行实时任务序列,保证紧迫任务的按时完成。

对于实时事务,在其调度上我们进一步考虑了在事务无法满足截止期的情况下的行为以及事务在错过截止期或操作失败时能够自动回滚和强制回滚操作。非预先安排的事务夭折所引起的回滚和重启会造成系统资源的巨大浪费,也会影响事务的实时性。因此,我们提出了在事务执行前预分析的概念,其步骤包括:

静态分析:提取有关存取行为的知识(操作数据集、时间限制等),以便在执行时进行基于上述提取知识的内外存数据交换,从而支持事务的定时限制。

事务分解:根据实时事务的同构性分解,该实时事务得到替代。

动态预分析:在动态环境下分析替代的行为特性,如分析替代的估算执行时间、事务间的时间相关性等。

3.2.4 事件触发机制

在内存数据库中,用户可以根据事件一条件一动作的形式预先定义实时规则。在运行过程中,当事件到来时,如果满足条件,将会触发一系列推理和动作。通过这样的规则触发形式,系统可以完成实时控制逻辑和实时规则推理。在设计上系统采用事件驱动和数据上推的处理模式可以提高实时数据库服务器核心的处理能力,避免在服务器端发生低效率的设备轮询,有效地将对数据设备的轮询以及数据点的灵敏度判断等任务分布到实时数据设备接口站点。为更有效地实现事件触发与数据流动,系统专门开发了高效率的线程间管道、进程间管道和实时数据通信服务中间件等通信软件构件。

3.2.5 与 windows 系统的融合

在 Windows 系统下,一般情况,系统中每个线程被 CPU 调度的时间是 20ms,采用分类调度事务的方法,实时数据库根据 windows 的特点和对变量采集时间的估计,合理分配线程及其中变量的数目,可以提高线程处理的效率。在实时数据向历史数据库转化中,以大约 300kb 的速度量写入硬盘,可以减少磁盘读写的 I/O 操作,在内存管理中按照 4kb 大小来分配内存,可以加快数据的访问速度,节省内存的占用空间。

3.3 历史数据库技术实现

当历史数据来临时,历史数据首先存储在历史数据缓冲池中。缓冲存满后被成块存入以队列形式存放的当前历史文件中。存满后,选择最早的历史数据文件作为可用的空历史文件,在实现时采用异步存储方式。即数据存储操作只需将历史数据追加在历史数据缓冲队列的末尾就立即返回,然后再由系统的内部工作线程将历史数据缓冲队列的数据写入历史数据缓冲池,这种异步操作方式避免了文件操作和数据封锁带来的时间不可预测性。

由于实时数据库系统中的历史数据量非常庞大,因此需要采用压缩的方法对其进行保存。对于大量连续变化的浮点型数据,一般采用依照变化趋势抽取的压缩方式。

4 结束语

实时数据库是现代工业控制的软件核心,基于内存的实时数据库具有内存快速访问的特性,通过实时事务调动的封装分类技术和数据检索方法,加快了工作速度,满足工控软件对实时性的要求。另外,C/S 和 B/S 结构的实现,满足了现代工业分布式的需求。

参考文献:

- [1]Hnang J,Slankovic J,Towslev D,Ramamrilham K,Purimetla B. Priority inheritance in soft real-time database [J]. Real-time System, 1992, 12(3):243-268.
- [2]R. Abott, H. Garcia-Molina. Scheduling real-time transactions: a performance evaluation [J]. ACM Transactions on Database Systems, 1992, 17(3):513-560.
- [3]A. Bestavros, K. Lin, S. H. Son Managing Contention and Timing Constraints in a Real-Time Database Systems [J]. Theory and Practice, 1996, (3):45-51.
- [4]郭巍,赵英,曲延涛.组态软件关键技术研究[J].电测与

仪表, 2006, 43(483): 49—52.

型计算机系统, 2003, 24(6): 1021—1024.

[5] 李国徽, 工洪亚. 分布式实时数据库并发控制[J]. 小型微

A Case for Distributed Real-Time Database for Industrial Control System

WANG An, XU Zhe-jun

(Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In the field of Industrial Control, the design of the real-time database has been a very important issue. This paper introduces a design of the Distributed Real-time database for Industrial Control System. To compensate the feature of the memory database, this paper introduces the ways to solving the key problems.

Key words: Real-Time Database; transaction scheduling; memory database; interbase

简 讯:

华东交通大学 2007 年 ACM 程序设计竞赛圆满落幕

为提高大学生运用计算机来充分展示自己分析问题和解决问题的能力, 并为了选拔我校新一批 ACM 集训队的队员, 由教务处主办、信息工程学院承办的华东交通大学 2007 年 ACM 程序设计竞赛于 4 月 21 日和 22 日成功地举办了预赛和决赛。日前竞赛结果揭晓, 评出一等奖 7 人、二等奖 13 人、三等奖 26 人。

ACM/ICPC (ACM International Collegiate Programming Contest, 国际大学生程序设计竞赛) 是由美国计算机学会 (ACM Association for Computing Machinery) 举办的一项历史悠久, 迄今规模最大, 能充分展示大学生计算机程序设计水平的世界性竞赛。

我校 2007 年 ACM 程序设计竞赛分预赛和决赛两个阶段: 预赛采用网上提交的形式, 共 8 道题, 比赛时间为 4 个小时。决赛采用现场测试结果的形式, 共 6 道题, 比赛时间为 5 个小时。经过激烈的角

逐, 初赛 200 余名参赛选手中, 有 80 余人进入第二天的决赛, 最后 46 人获奖。

此次竞赛反映了参赛选手的真实水平, 充分激发了同学们的学习兴趣。其中倪虎(电气学院)、袁超和王辉(信息学院)、余双剑和邓冰(软件学院)表现尤为出色, 他们均正确解出了决赛的所有题目并获得一等奖。这五名同学都是我校 ACM 集训队的队员, 其中参加训练时间长的有两年多, 短的有近半年。另两名获一等奖的同学是基础学院包五平, 软件学院胡楚斌。

我校的 ACM 集训队自 1998 年起由周尚超教授担任教练至今, 已经成功地培养出一批又一批的人才, 获得了 ACM/ICPC 国际大学生程序设计竞赛的诸多奖项, 为我校争得了荣誉。毕业的队员们都成功走向社会, 发挥着计算机程序设计特长。

(王辉供稿)