分布式实时库的一致性实现

高春雷[1],尹燕敏[2]
(1. 国电自动化研究院,江苏 南京 210003;
2. 河海大学 计算机及信息工程学院,江苏 南京 210098)

摘要: 为了满足数据采集和监控 SCADA(Supervisory Control And Data Acquisition)系统中分布式实时数据库对关键数据的强一致性和节点数据完整性的要求,提出采用高速网络通信技术和网关模式相结合的方案。介绍了分布式实时数据库系统 DRTDBS(Distributed Real-Time DataBase System)关键数据强一致实现及网关模式实现 DRTDBS 中各节点数据的完整性。用 100 Mb 以太网和 Intel 100/1000 Mb 网卡进行的性能测试表明:所提出的方案客户端接收信息的速度能及时地响应 SCADA 系统实时库更新操作;方案已投运现场。

关键词: 分布式实时数据库; 高速网络通信; 同步复制
中图分类号: TM 76;TP 393 **文献标识码**: A **文章编号**: 1006-6047(2005)04-0083-03

1 分布式实时数据库系统

随着计算机软硬件技术的飞速发展,在电力系统数据采集和监控 SCADA(Supervisory Control And Data Acquisition) 系统中, 分布式实时数据库系统 DRTDBS(Distributed Real-Time DataBase System)已逐步进入实际应用阶段。

在开发 DRTDBS 应用时,为了适应 SCADA 系统中关键数据的强一致性,及系统中各个节点数据完整性的要求,需要进一步提高同步复制效率。本文通过在 DRTDBS 中运用高速网络通信技术和基于网关模式的主副本更新技术[1]实现了 SCADA 系统中对 DRTDBS 的一致性和完整性的要求。

在实现 SCADA 中的 DRTDBS 子系统时,主要有以下 3 个问题需要解决:

a. 如何提高主副本节点与副本节点之间关键数据的强一致性,防止关键数据出现不一致;

b. 如何保证 DRTDBS 中各个节点的数据完整性,即解决同步更新问题;

c. 各个事务如何在提交时进行有效性检测。

本文主要解决前两个问题,采用高速网卡驱动方式提高主副本节点与副本节点之间的连通性和传输的实时性以确保关键数据的强一致性,以及采用网关模式保证各个节点的数据完整性。至于问题 **c**,可以考虑采用有效性检测算法[2]处理事务冲突问题。

2 DRTDBS 关键数据强一致性实现

2.1 相关知识

对于 DRTDBS 中的关键数据,为了提高系统各个节点之间的数据强一致性,通过采用高速网络通信方式完成。由于高速网络的带宽增长迅速,从 100 Mb 以太网到 1 Gb 以太网以至 10 Gb 以太网,现有的计算机网络处理结构和处理能力跟不上网络发展的要求。而分布式实时数据库属于专用系统,不需要标准的、完备的协议处理,因此可根据系统实际情况,开发专用驱动程序,提供用户程序和网卡之间零拷贝数据包收发,保证复制同步程序及时处理网络包,以尽量减少通信中间环节、达到充分利用系统资源的目标。

常规的 OS 网络数据包接收过程如下:

a. 网卡接收数据包,通过 DMA 传送到主机;

b. 网卡向主机发送硬件中断;

c. 硬件中断程序将数据包转入接收队列;

d. 软中断处理程序对数据包按协议分发;

e. 用户程序通过系统调用将数据包拷贝到用户空间;

f. 用户程序进行分析处理。

为了减少通信中间环节,提高通信传输效率,可以采用改进系统提供的网卡驱动程序并且直接使用用户进程缓冲区,减少核心到用户空间内存拷贝,免去核心频繁的内存分配和不必要的协议处理。同时,由于使用了用户缓冲区,可以极大地扩充系统可用缓冲区。通过使用大缓冲区(可到 GB 级),可以有效地减少高峰期丢包率。

在 Windows 平台上,用户可以通过微软提供的设备驱动程序资源包 DDK(Driver Development Kit)开发包与网络驱动程序接口规范 NDIS(Network Driver Interface Specification)系统库进行底层驱动程序的开发。NDIS 程序库(NDIS.sys)提供了一个面向网络接口卡 NIC(Network Interface Card)驱动程序的完全抽象的接口,网卡驱动程序与协议层驱动

收稿日期:2004-07-15;修回日期:2004-11-02

程序及操作系统通过这个接口通信。NDIS库还参与管理操作系统与网络有关的特定任务,管理所有底层的NIC驱动程序的绑定与状态信息。NDIS支持微端口、中间层和协议3个驱动程序[3]。

2.2 专用驱动程序开发设计

实现DRTDBS关键数据的一致性时,采用如下方式设计NDIS协议驱动程序。

每当上层应用程序请求读取数据时,就为该请求生成1个相应的I/O请求包IRP(I/O Request Packet),并将此IRP置于1个读取等待队列中,当NIC从网络上接收到数据包时,由NDIS负责调用相应的接收函数。接收函数从读取等待队列的首部取出1个IRP,并将从网络上接收的数据包拷贝到由该IRP所指定的缓冲区中。驱动程序负责维护1个接收缓冲区,该缓冲区以双缓冲区的方式,以提高数据传递的可靠性和响应速度。DRTDBS中各个节点之间采用系统约定的网络数据包处理:先发送信息包,报告下一或多包数据的信息,然后发送具体的数据包。

上层应用程序通过共享内存直接与NDIS协议栈相交互。通过自旋锁[3](spin lock)和事件在两个正在执行的线程之间实现同步操作。定义、分配NDIS包和填充数据并且将它传递到下层的NDIS驱动程序,从而将数据包发送到网络上(自旋锁是由内核定义的内核模式仅有的一种同步机制,它以不透明类型KSPIN_LOCK向外界输出)。

对于协议驱动程序开发中的内存管理,需要解决在内核模式下使用用户模式下的内存空间的问题。

协议驱动程序通常运行在内核模式,可以按需分配额外的系统空间内存,而且可用内核栈在调用内部驱动程序例程时传递少量的数据。由于它们运行在内核模式,驱动程序不能分配用户空间的虚拟内存。此外,驱动程序不能通过用户模式的虚拟地址访问内存,除非它正运行在引起驱动程序当前I/O操作的用户模式线程环境中,而且它正在使用此线程的虚拟地址。驱动程序可以在为低层驱动程序建立IRP之前调用MmProbeAndLockPages以锁住用户缓冲。最低层驱动程序和为缓冲或直接I/O建立设备对象的中间层驱动程序,可以依赖I/O管理器或最高层驱动程序,在IRP中传递对被锁用户缓冲或系统空间缓冲的合法访问。

具体的实现步骤如下:

a. 调用IoAllocateMdl分配描述用户缓冲的MDL;

b. 调用MmProbeAndLockPages锁住用户缓冲;

c. 给用户缓冲传输数据;

d. 调用MmUnlockPages,并做下列事情之一,如果驱动程序在步骤**a**中分配的MDL非常大,足够下次传输,调用MmPrepareMdlForReuse并重复步骤**b**到**c**,否则,调用IoFreeMdl并重复步骤**a**到**d**;

e. 所有数据传输结束后,调用MmUnlockPages和IoFreeMdl释放被锁住的内存和用户缓冲描述符。

2.3 应用程序与驱动程序的交互

在成功地开发完成协议驱动程序后,还需要在用户程序中实现与底层协议驱动程序的交互,这样才能充分利用专用的协议驱动程序提高节点之间的通信传输效率。在Windows平台中,设备是被当成一个文件进行操作的。对于任何用户模式代码可以直接递交I/O请求的设备,无论它是物理的、逻辑的或虚拟的,其驱动程序都必须为它的用户模式客户提供某种名字。用这个名字,用户模式应用程序就可以识别出请求I/O的那个设备。设备对象被划分成类,每一类都与1个全局唯一标识符GUID(Globally Unique IDentifier)相关。系统在设备特定的头文件中为公共设备接口类定义GUID。当驱动程序注册了1个设备接口,I/O管理器将其设备接口类GUID与1个符号链接名相关。链接名保存在注册表中,它在系统启动期间一直存在。使用此设备接口的应用程序可以查找它的符号链接名,并将它保存用作I/O请求的目标。用户态下的应用程序可以通过:CreateFile,CloseHandle,ReadFile,WriteFile和DeviceIoControl完成底层驱动程序实现PacketOpen,PacketClose,PacketRead,PacketWrite,PacketIoControl对应的功能。

3 网关模式

根据DRTDBS的数据完整性的要求,运用网关模式[4]实现DRTDBS中各个节点数据的完整性。模式的体系结构见图1。

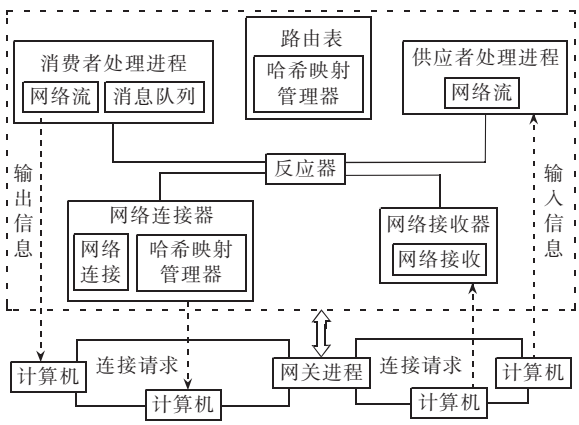

图1 网关模式的体系结构

Fig.1 The architecture of gateway pattern

在DRTDBS中,由于系统中节点较多,而且系统中数据更新频繁导致网络中通信数据量很大。因此,在运用网关模式实现DRTDBS通信模块时,采用多线程实现主动对象网关,并使用非阻塞缓冲式I/O模式提高更新数据在主副本节点之间的传输效率。同时,消息排队组件中的消息队列(message queue)采用管程对象(monitor object)[5]模式实现,使得通信模块可在多线程环境中高效地运行。

供应者处理进程(supplier handler):负责将DRTDBS 中传递来的更新数据作为到来的外部消息路由给它们的目的地(即与该主副本节点相关的各个副本节点)。

消费者处理进程(consumer handler):负责将路由消息(DRTDBS 中传递来的更新数据)可靠地递送给它们的目的地(即与该主副本节点相关的各个副本节点)。该处理进程采用非阻塞缓冲式 I/O 模式实现以防止在与某个节点通信时发生阻塞而影响整个系统的通信。

消费者处理进程采用多线程结构的主动对象模式实现,同时,在供应者处理进程中也采用同样的基于多线程结构的主动对象模式,由于采用了多线程结构,系统可以在硬件许可的情况下提高响应速度。

4　性能测试

测试环境是:局域网为 100 Mb 以太网,系统中的机器都是 Intel 100/1000 Mb 网卡,P41.7G,操作系统为 WIN2K SP4 和 WIN XP。测试达到的指标是:采用普通 Windows socket 方式能够正确接收 64 Byte 的 $1\sim2\times10^4$ 包/s。实时库进行 1×10^3 条记录的同表插入所需时间为 51 ms,若全部通过网络发送给客户,则客户端接收速度需达到 64 Byte 的 4×10^4 包/s。通过采用网关模式及高速网卡驱动技术后,客户端的接收速度达到了 64 Byte 的 5×10^4 包/s,能够及时地响应整个 SCADA 系统中的实时库更新操作。

5　总结和展望

本文的设计思想已经成功地用于电力系统的 SCADA 系统中。实际运行效果表明,所述的基于高速网络通信技术和网关模式的 DRTDBS 设计思想是可行的。整个 SCADA 系统对在传输系统关键数据时能迅速准确同步更新到网络中各个节点,整个 SCADA 系统在 DRTDBS 的基础上能稳定可靠地长时间运行。

参考文献:

[1] KEMME B,ALONSO G. A new approach to developing and implementing eager database replication protocols[J]. **ACM Trans. on Database Systems**,2000,25 (3):333-379.

[2] CERI S,OWICKI S. On the use of optimistic methods for concurrency control in distributed system[A]. **Proc 6 th Int. Conf. on Distributed Data Management and Computer Networks** [C]. Berkeley,California:[s.n.],1986.213-226.

[3] Microsoft Corporation. Windows 2000 驱动程序开发大全:第 1 卷 设计指南[M]. 冯博琴,朱丹军,薛　涛,等译. 北京:机械工业出版社,2001.

[4] SCHMIDT D C,STAL M,ROHNERT H,*et al*. Pattern-oriented software architecture:Patterns for concurrency and distributed objects[M]. New York,USA:Wiley & Sons,2000.

[5] 缪国钧,葛晓霞,林中达. B/S 架构的电厂实时 MIS 系统的分析与研究[J]. 电力自动化设备,2003,23(7):23-27.
MIAO Guo-jun,GE Xiao-xia,LIN Zhong-da. Research on power plant real-time MIS with B/S architecture [J]. **Electric Power Automation Equipment**,2003,23(7):23-27.

[6] 陈素芬,张　波,李　萍,等. Windows 操作系统下电力系统实时数据采集的 VXD 技术 [J]. 电力自动化设备,2003,23(3):13-15.
CHEN Su-fen,ZHANG Bo,LI Ping,*et al*. VXD technology for power system real-time data collecting in Windows operating system[J]. **Electric Power Automation Equipment**,2003,23(3):13-15.

(**责任编辑**: 汪仪珍)

作者简介:

高春雷(1973-),男,江苏常州人,工程师,硕士,主要研究方向为分布式计算和计算机网络(**E-mail**:gaocl@nari-china.com);

尹燕敏(1975-),女,山东泰安人,讲师,硕士,主要研究方向为网格计算和数据挖掘(**E-mail**:yinym@vip.sina.com)。

Implementation of consistency in distributed real-time database management system

GAO Chun-lei[1],YIN Yan-min[2]

(1. Nanjing Automation Research Institute,Nanjing 210003,China;
2. Hohai University,Nanjing 210098,China)

Abstract: A solution combining high-speed network communication and gateway pattern is presented to realize the hard consistency and integrity of distributed real-time database system in SCADA (Supervisory Control And Data Acquisition). Its implementation is introduced. A performance test is carried out with 100 Mb Ethernet and Intel 100/1000 Mb net card. The results show that the speed of receiving client information matches the updating of real-time database in SCADA. The scheme has been practiced.

Key words: distributed real-time database system; high-speed network communication; eager replication