第33卷 第20期 计 算 机 工 程 2007年10月
Vol.33 No.20 Computer Engineering October 2007
·工程应用技术与实现· 文章编号：1000—3428(2007)20—0242—03 文献标识码：A 中图分类号：TP393

MS-Windows 异步网络备份系统的设计与实现

张晓平，刘晓洁，李 涛，赵 奎，朱国云，陈云峰，刘 锦
（四川大学计算机系，成都 610065）

摘 要：提出并实现了MS-Windows下的远程异步备份系统。该系统在卷设备驱动层监视本地应用服务器的写操作，并将相关信息封装成写操作重放记录发送到备份网关上缓存，由备份网关异步地将所缓存的记录发送到远程备份服务器上，在远程服务器上写入相应的备份卷。该系统支持Windows下的所有文件系统与存储设备，实现了对本地服务器逻辑卷的异地备份。

关键词：数据备份；设备驱动；写操作重放

Design and Implementation of Asynchronous Network Backup System Based on MS-Windows

ZHANG Xiao-ping, LIU Xiao-jie, LI Tao, ZHAO Kui, ZHU Guo-yun, CHEN Yun-feng, LIU Jin
(Computer Department, Sichuan University, Chengdu 610065)

【Abstract】 This paper presents and implements an asynchronous network backup system based on MS-Windows. This system monitors the write operations of local application server on the level of logical volume and encapsulates the information in a replay record, which is sent to the local gateway. After it is cached in the local gateway, the records are transmitted to the remote backup server asynchronously. The same write operation is carried out in the corresponding volume in the remote backup server. This system can be applied to all kinds of file systems and storage devices in windows and it realizes the remote backup of logic volumes in the local application server.

【Key words】 data backup; device driver; write operation replay

现代企业的运转日益依赖于信息技术，如果发生数据丢失和损坏，势必将造成难以估量的损失，而备份是保证数据安全的有效方法[1]。在“9.11”的灾难中，很多公司因为数据丢失纷纷倒闭，但另外一些公司却能够在两天内恢复营业，主要原因是它们将数据进行了异地远程备份，在灾难发生时及时进行了数据恢复。传统的备份技术[2~3]，如磁带备份、RAID[4]等，只能在较短的距离内实现备份，并不是真正意义上的异地备份。NAS[5]等网络存储技术可实现数据的远距离备份，但需要光纤专线，成本十分昂贵。因此，容灾及远程备份技术正成为目前的研究热点。

目前，数据容灾系统产品的研究和开发主要还集中在国外，很多国外知名大公司都有自己研制的数据备份系统，其中融合了SAN、NAS、集群等技术，虽然功能强大，但运行成本也非常高[6~8]。国内在此方面的研究才刚刚起步，几乎没有自主开发研制的容灾与远程备份类产品。

本文设计并实现了一种基于Windows的远程异步备份系统，该系统实现了本地数据的远程异步备份，可直接架构于Internet上，具有较低的系统运作成本，而且对应用程序透明，支持Windows下的所有文件系统及存储设备。

1 体系结构

如图1所示，系统被划分为3个部分，分别是本地数据中心、本地海量缓存网关以及远程数据备份中心。

本地数据中心由一个或多个本地服务器构成，每一个本地服务器都包括了为企业提供正常服务所需要的应用程序和数据；本地海量缓存网关用于集中缓存本地数据中心的数据，再将数据传送到与本地服务器相对应的远程数据备份服务器上；远程数据中心包括了多个备份服务器，每一个备份服务器都对应了一个本地服务器，用于备份相应的本地服务器上的数据。

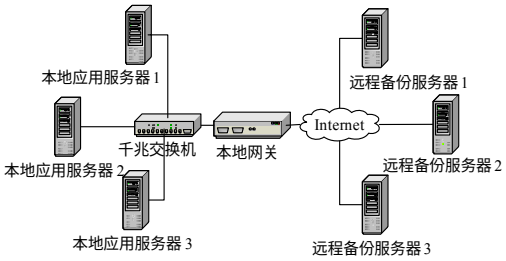

图1 体系结构

如图2所示，从功能上看，整个系统被划分成3个模块：本地服务器数据监控模块，本地网关数据缓存模块，远程写操作重放模块。数据监控模块位于本地服务器上，它以内核驱动模式运行在本地数据中心的每一台服务器中，用于记录本地服务器上数据的变化，并根据数据的变化创建写重放记录，发送给本地网关上的数据缓存模块；数据缓存模块位于

基金项目：国家自然科学基金资助项目(60373110, 60573130, 60502011)；教育部新世纪优秀人才计划基金资助项目(NCET-04-0870)；教育部博士点基金资助项目(20030610003)；四川省应用基础研究计划基金资助项目(05JY029-021-1)

作者简介：张晓平(1981－)，男，硕士研究生，主研方向：网络安全技术及应用；刘晓洁，副教授；李 涛，博士生导师、教授；赵 奎，博士研究生；朱国云、陈云峰、刘 锦，硕士研究生

收稿日期：2006-11-19 **E-mail：**xpingc@gmail.com

本地网关上，它将接收数据监控模块所发送的写重放记录，并将这些数据缓存在其上的磁盘海量缓存器中，然后发送给对应的远程备份服务器；远程写操作重放模块位于远程备份服务器上，它接收数据缓存模块所发送的写重放记录，再写入到相应的设备对象中。

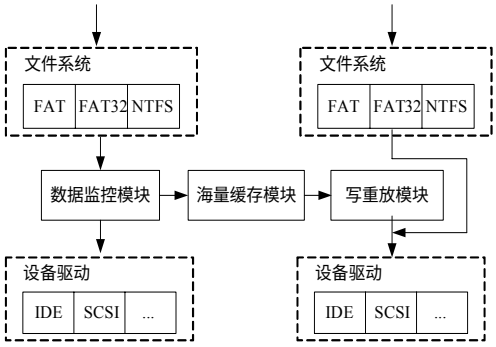

图 2　模块结构

从上面的描述可以看出，在实际的传送过程中，本地服务器上所监控到的数据将先被发送到本地海量缓存器中，然后才会被发送到对应的远程备份服务器上。之所以采用两层的数据传送结构，主要是考虑到真正意义上的异地备份需要使用 Internet 连接，而 Internet 连接的数据传输率远远小于本地磁盘的写入速率。如果直接连接本地服务器和远程备份服务器，本地服务器的数据写入速度将受到较大影响，而且在每一台本地服务器上都需要有一个较大的缓存来存放写操作重放记录，这无疑影响了本地服务器的性能。引入本地海量缓存器网关后，它和本地服务器之间采用千兆位以太网进行连接，其数据传输的平均速度可以达到 60Mb/s 左右，这样，只需要在本地服务器的内存中设置较小的缓存就可以将数据传送到本地网关上的海量缓存器中，不会给本地服务器的性能带来多大影响。

2　系统设计与实现

2.1　本地数据监控模块

本地数据监控模块以内核驱动程序模式运行在本地服务器上，其要实现的功能有两个：监控本地写操作数据，将监控到的数据封装成写重放记录；发送到本地网关的海量缓存器中。数据监控是数据监控模块所要实现的核心功能，需要了解 Windows 下 I/O 操作的处理流程。

2.1.1　Windows 下的 I/O 操作处理流程

Windows 下的所有应用程序进行的 I/O 操作都会转化为调用 NtReadFile 和 NtWriteFile 这两个系统调用函数。它们将调用相应的设备驱动程序中的例程来完成 I/O 操作，再将结果返回给应用程序。而在调用驱动程序的相应例程前，它们首先会调用操作系统 I/O 管理器中的相应模块生成一系列 I/O 请求包(IRP)来描述这个 I/O 操作。一个 I/O 请求包描述了 I/O 操作对应的设备对象，读出或是写入的数据在设备中的偏移，以及需要读入的数据将传输到的内存中的位置或者是需要写入设备的数据在内存中的位置等信息。

Windows采用多层结构来处理一个磁盘设备的I/O请求操作。当需要写入磁盘上的某个文件时，通常需要经过文件系统驱动程序、逻辑卷管理驱动程序以及磁盘驱动程序。文件系统驱动程序接收NtReadFile和NtWriteFile所产生的I/O请求包，它将针对文件的偏移转换成针对逻辑卷的偏移，再调用逻辑卷驱动程序；逻辑卷驱动程序又将I/O请求包中针对卷的偏移转换成针对磁盘的偏移，调用磁盘驱动程序；最终，磁盘驱动程序在接收到该I/O请求包后，将实际完成该I/O请求，然后将结果逐层返回，这一过程如图 3 所示[9]。

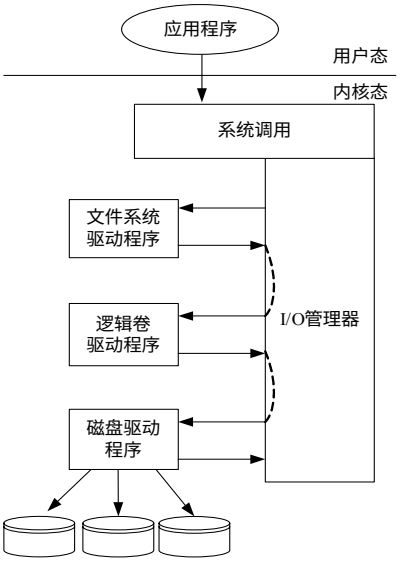

图 3　磁盘设备驱动程序层次

Windows不仅提供了层次化的驱动程序模型，而且还提供了相应的方法在已有的设备驱动层次结构中加入新的驱动程序。可以通过在磁盘设备驱动程序的层次结构中加入监控驱动程序的方法来实现本地数据的监控。考虑到监控程序的灵活性、文件系统的复杂性以及某些数据库应用程序有可能直接对没有文件系统的逻辑卷进行操作等因素，监控驱动程序最终被加载到了逻辑卷驱动程序和文件系统驱动程序之间[10]，也就是监控逻辑卷的写操作。

2.1.2　本地数据监控模块的实现

从监控到本地服务器上的数据到将数据发送给本地网关上的海量缓存器这一过程将采用半异步的方式来实现。当监控驱动程序加载到需要监控的逻辑卷上时，监控驱动程序会首先创建一个内核系统线程，用于以后发送数据到本地网关。随后，每当监控程序收到一个写操作 I/O 请求包时，它将分配一块和写入数据大小相同的内存区来拷贝需要写入的数据，然后将该内存区的地址、写入数据在卷中的偏移以及写入到远程备份服务器的卷名称等信息打包成写操作重放记录存放到内存中的写操作重放队列中。此时，刚才所创建的系统线程将被唤醒，它会从写操作重放队列中逐个取出每个记录，然后通过在内核中所实现的套接字，使用 TCP 协议将数据发送给本地网关。而原始的 I/O 请求包将继续在调用该驱动程序写操作处理例程的线程中被完成，无须等到写入的数据发送到本地网关上，这样就不会影响本地写入的速度。

显然，当本地写入的速度达到一定程度或是本地网关出现故障时，本地监控程序有可能会出现内存溢出的问题，需要对本地监控程序写操作重放队列的大小进行限制，一旦它超过了该限制，所有调用该监控驱动程序写操作处理例程的线程将被阻塞，直到当前所缓存的数据全部被发送到本地网关。如果本地网关出现了故障，或是需要关闭本地服务器，那么所有写操作重放队列中的数据将被存放在本地服务器上的一个临时文件中，在本地网关恢复正常或是本地服务器重新启动后，监控驱动程序首先会将该临时文件中的数据发送到本地网关中，这样就保证了数据备份的可靠性。

2.1.3　备份节点队列大小的设置

由于本地磁盘的写入速度通常是网络传输速度的几倍，因此合理设置本地服务器上写操作重放队列的大小直接影响到了本地服务器的性能以及备份的速度。假设本地需要写入大小为m的数据，如果直接在本地进行写入，需要时间t_1，如果同时还需要传递到本地网关上，需要的时间为t_2，令$\lambda=t_2/t_1$。假设本地磁盘写入的速度为v_{in}，从本地服务器到本地网关网络传输的速度为v_{out}，令$k=v_{in}/v_{out}$，显然$\lambda>k>1$。再假设本地写操作重放队列的大小限制为s，则经过时间$s/(v_{in}-v_{out})$，该队列将被装满。此时，写入该卷的线程将被阻塞，同时另一个线程将被调入CPU执行。假设被调入的线程也需要写入数据到该监控卷，而线程切换需要的时间为δ_t，那么将大小为m的数据写入本地卷的同时再发送到本地网关所需要的时间t_2为

$$t_2=\frac{v_{in}-v_{out}}{s}\cdot t_2\cdot\delta_t+\frac{m}{v_{out}}$$

经计算，可以得出 s 的值为

$$s=\frac{k-1}{1-k/\lambda}\cdot v_{out}\cdot\delta_t$$

根据上面的式子不难得出，s 随 λ 的增大而变小。也就是说，本地服务器的缓冲区越大，数据备份到本地网关的时间与数据写入本地所需要的时间两者相差越小。假设本地磁盘写入的速度为 120Mb/s，本地服务器与本地网关之间的网络传输速率为 60MB/s，则 k=2，再假设 λ=2.2，另外对于 Windows 系统而言，线程之间切换需要的时间通常不会超过 10ms，按照这样的假设值，可以得出 $s\approx$3M。可见，这样大小的缓冲区对于本地服务器的运行完全没有多大的影响。

2.2　本地网关海量缓存器模块

本地网关为每一个本地服务器都准备了一个磁盘缓存器，而海量缓存器模块的作用就是接收本地服务器数据监控模块所发送的写操作重放记录，并按序存放在与该本地服务器相对应的磁盘缓存器中。当本地网关和远程备份服务器的网络连接正常时，海量缓存模块从缓冲器中取出数据，并按序发送给远程备份服务器。

该模块全部的功能都是由用户态程序来实现，启动时它将从本地网关上的配置文件中读取相应的启动参数。配置文件中的启动参数包括需要备份的本地服务器的个数、与每个本地服务器所对应的远程服务器的地址、每个本地服务器上的监控卷所对应的远程备份服务器上的备份卷的名称等信息。当启动成功后，它会创建一个写操作重放记录接收线程T_R，用于接收所有本地服务器所发送来的数据；此外，它还为每个本地服务器建立一个记录发送线程T_i，用于将缓存到本地网关的写重放记录发送到相应的远程服务器上。

2.3　远程备份服务器写操作重放模块

远程备份写操作重放模块以用户态应用程序的形式在远程服务器上运行，它根据接收到的本地网关海量缓存器模块所发送的写操作重放记录中的数据，将本地服务器某个卷上的写操作重放到其上的某个备份卷上，完成数据备份的最后一个阶段。该模块以直接写入逻辑卷的形式来完成重放过程，它根据写操作重放记录中所指定的备份卷的名称，将备份卷打开，然后根据写操作重放节点中所指定的偏移量，直接写入到备份卷中。

3　结束语

本文设计并实现了一种 MS-Windows 下的远程异步备份系统，通过在设备驱动层截获本地服务器写操作并在远程服务器上重放，实现了对本地卷设备的远程物理备份。该系统支持 Windows 下的所有文件系统与存储设备，可直接架构于 Internet 上，为企事业的数据安全提供了一套合理的远程数据备份方案。

参考文献

1 李　涛. 网络安全概论[M]. 北京: 电子工业出版社, 2004-08.

2 Hutchinson N C, Manley S, Federwisch M. Logical vs. Physical File System Backup[C]//Proceedings of the 3rd USENIX Symposium on Operating System Design and Implementation. 1999-02.

3 Cunhua Q, Syouji N, Toshio N. Optimal Backup Policies for a Database System with Incremental Backup[J]. Fundamental Electronic Science, 2002, 85(4): 1-9.

4 Stefano T, Source C W. The Distributed Data Center: Front-end Solutions[J]. IT Professional, 2004, 6(3): 26-32.

5 Lo Chi-chun. A Novel Approach of Backup Path Reservation for Survivable High-speed Networks[J]. IEEE Communications Magazine, 2003, 41(3).

6 Hayes P E, Disaster H A. Recovery Project Management[C]//Proc. of IAS Annual Meeting(IEEE Industry Applications Society). 2000: 2814-2821.

7 Suzuki J, Suda T. Middleware Support for Disaster Response Infrastructure[C]//Proc. of the 1st IEEE Workshop on Disaster Recovery Network, 2002-06.

8 Toigo J W. Disaster Recovery Planning——Strategies for Protecting Critical Information Assets[M]. [S.l.]: Prentice Hall, 2000.

9 Solomon D, Russinovich M. Microsoft Widows Internals[M]. Washington: Microsoft Press, 2005.

10 Oney W. Programming the Microsoft Windows Driver Mode[M]. Washington: Microsoft Press, 2003.

(上接第 241 页)

7　结语

本文针对使用硬件仿真器对多 DSP 进行调试的不足，阐述了在使用总线与主机连接的多 DSP 系统上构建软件调试器的方法，提出了动态 monitor 的方式使调试器对 DSP 的资源占用降到极低的程度。测试和使用表明，基于总线的软件调试器与硬件仿真器相比，具有更好的性能、更高的性价比和更好的多处理器支持。以后的工作中需要进一步完善软件调试器的兼容性，在多处理器调试方法方面做进一步的研究。

参考文献

1 Earnshaw R W, Smith L D, Kevin V. Challenges in Cross-development[J]. IEEE Micro, 1997, 17(4).

2 吴　疆, 田金兰, 张素琴. 面向多目标机的交叉调试器的研究与设计[J]. 清华大学学报(自然科学版), 2003, 43(1).

3 熊　竞. 嵌入式操作系统的调试[Z]. (2000-06). http://www2900.ibm.com /developerWorks/linux/embed/debug/index.shtml#author1.