第35卷 第21期 计 算 机 工 程 2009年11月
Vol.35 No.21 Computer Engineering November 2009
·软件技术与数据库· 文章编号：1000—3428(2009)21—0036—03 文献标识码：A 中图分类号：TP311

虚拟存储技术在容灾系统中的应用

康潇文，杨英杰，杜 鑫
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：基于对虚拟存储技术及其在容灾系统中应用现状的分析，设计一个适用于容灾系统的虚拟文件系统。该系统基于Windows平台，通过在Windows存储栈的层次式结构中添加过滤驱动层来实现上层文件系统与下层卷管理器的隔离。在过滤驱动层中，结合虚拟内存的映射原理和容灾系统的应用需求，实现适用于容灾系统的虚拟存储映射机制。

关键词：虚拟存储；容灾；文件系统；过滤驱动

Application of Virtual Storage Technology in Disaster Tolerant System

KANG Xiao-wen, YANG Ying-jie, DU Xin
(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004)

【Abstract】Based on the analysis of virtual storage technology and its application in disaster tolerant system, this paper designs a virtual file system which is adapted to disaster tolerant system. This system bases on Windows platform, and adds a filter driver layer in Windows storage stack to implement the isolation between file system and volume manager. It implements virtual storage mapping mechanism which is adapted to disaster tolerant system in filter driver layer.

【Key words】 virtual storage; disaster tolerance; file system; filter driver

1 概述

随着信息化时代的到来，数据在信息系统中的地位日趋重要，而数据容灾也成为整个系统容灾、应用容灾的基础。数据的破坏和不可恢复性将导致整个容灾的失败，因此，数据存储技术成为了搭建容灾系统的基础。存储技术经历了从单个磁带、磁盘、磁带库、冗余磁盘阵列(RAID)到存储网络系统的发展历程，容灾系统的容灾能力正是随着存储技术的发展而不断提高的。但是形式各异的存储结构使得搭建容灾系统变得复杂、昂贵，并且单一的存储技术已无法满足各行业对容灾能力的需求。研究人员开始将注意力转向了虚拟存储技术，利用虚拟存储技术的优点来简化容灾系统的搭建过程，从而提高容灾的效率。但是由于业界的虚拟存储技术尚未标准化等原因，虚拟存储技术的优点还未能在容灾系统中充分体现出来。

本文针对这一情况，在分析虚拟存储技术及其在容灾系统中的应用现状的基础上，设计了一个应用于容灾服务器端的虚拟文件系统。

2 虚拟存储技术及其在容灾系统中的应用现状

近几年来，随着网络技术的迅猛发展，信息容量的急剧膨胀，数据存放也变得更加分散，促使了虚拟存储技术的快速发展。然而由于业界尚没有形成虚拟化的标准，存储厂商一般根据自己所掌握的核心技术来提供虚拟存储解决方案。从已推出的虚拟产品结构来看，可以将虚拟存储技术概括为主机级、设备级和网络级[1] 3个级别。几种虚拟存储结构如图1所示。

(1)主机级虚拟存储技术主要通过软件实现，对直接挂载在其下的磁盘和磁带设备进行统一管理。

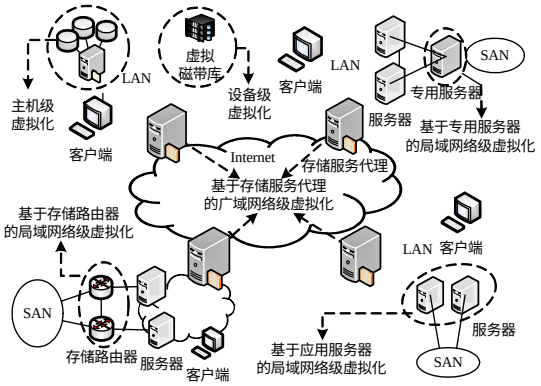

图1 虚拟存储结构

(2)设备级虚拟存储技术包括两方面：1)对设备物理特性的仿真，如利用磁带设备仿真磁盘设备，该设备具有磁盘驱动器的一切特性，而数据却存放在磁带上；2)针对虚拟设备的构建，从功能上看与主机级虚拟技术类似，不同的是其虚拟化管理模块是嵌入在硬件中实现的，如磁盘阵列RAID就是典型的虚拟化设备。

(3)网络级虚拟存储技术是基于网络实现的，通过虚拟化管理软件构建管理节点，实现资源的集中管理。根据虚拟化管理软件实现位置的不同，又可细分为服务器级、存储路由器级，以及存储服务代理级虚拟化。网络级虚拟存储被认为

基金项目：河南省自然科学基金资助项目(0611051300)
作者简介：康潇文(1983－)，女，硕士研究生，主研方向：容灾与信息安全；杨英杰，副教授、博士；杜 鑫，硕士研究生
收稿日期：2009-04-26 **E-mail：**kangxiaowen-0829@126.com

是最具逻辑意义的虚拟化。

上述 3 种虚拟存储技术各有其特点，可以单独使用，也可以在同一个存储系统配合使用。对于大型的存储环境而言，综合应用 3 种虚拟方法能发挥整体优势，从而取得最好的效益。

将虚拟化存储技术应用于容灾系统的研究目前仍处于初级阶段，其主要应用是将设备级虚拟化存储产品应用于容灾系统，从而提高数据的迁移速度。例如，HP Storage Works EVA 企业虚拟阵列存储系统是设备级虚拟化产品在容灾系统中应用的典范[2]，它的引入可以有效提高数据的复制和恢复速度。从理论上讲，将虚拟存储技术应用到容灾系统可以在不中断应用的情况下，在线增加存储容量，进行存储设备的更换，实现数据的透明备份、恢复、迁移等，从而极大提高了容灾的有效性和灵活性。但现阶段基于这种设备级虚拟存储技术的应用仍具有一定的局限性，无法满足容灾系统中高强度数据迁移的要求，而且也未能实现真正意义上容灾系统的透明化管理。

3 基于 Windows 平台的虚拟文件系统设计

针对上述问题，本文设计了一个应用于容灾服务器端的虚拟文件系统。该文件系统基于 Windows 存储栈的层次化模型[3]，通过添加文件系统过滤驱动来实现适用于容灾系统的虚拟存储映射机制。

3.1 虚拟存储技术简述

虚拟是将一个对象(可以是产品或设备)模拟成另外一个对象或实体的操作行为[4]，其目的就是使底层设备对用户而言，达到屏蔽、透明的效果。这个概念已被用于不同的领域，而虚拟存储就是其中之一。

虚拟存储是指将不同类型的存储物理媒介和不同层面的存储子系统通过软硬件技术转化到统一的用户存储应用界面的技术[5]。后台存储设备的变更以及数据的迁移不会对前台用户的应用造成任何影响。

3.2 虚拟文件系统体系结构

在 Windows 操作系统中，将管理一个特定存储设备的过程中所涉及的设备驱动程序合起来称为存储栈[3]，见图 2。

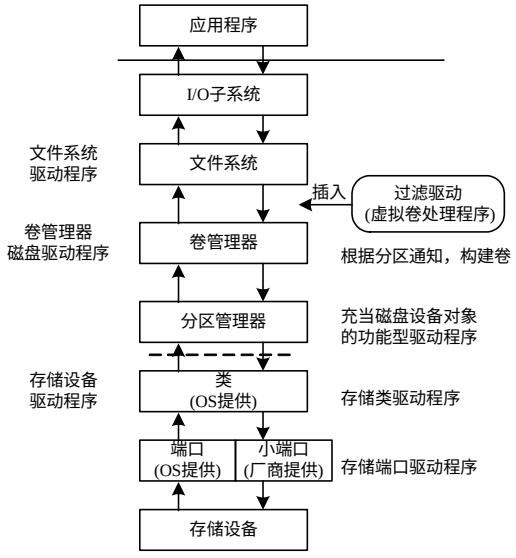

图 2 基于 Windows 存储栈的虚拟文件系统体系结构

存储端口驱动程序提供对特定总线的共有功能，由操作系统提供；小端口驱动程序用于实现存储设备的特定功能，由厂商提供；存储类驱动程序创建代表磁盘和分区的设备对象，实现对所有存储设备都共有的功能；而分区管理器则充当磁盘设备对象的功能型驱动程序，它向上层报告分区通知。以上这些统称为存储设备驱动程序[3]，其目的是将底层的硬件地址(如磁盘的柱面、扇区号)转换成连续的物理地址和偏移。

卷管理器设备驱动程序接收下层所提供的磁盘分区通知，考察所有构成卷的分区，定义相应的卷对象(对用户而言，卷就是用户通常看到的 C 盘、D 盘等)，实现从分区到卷的对应关系[3]，也就是将连续的物理地址和偏移转换成相对于卷的字节偏移。

文件系统驱动程序将无意义的字节流序列构建成一种有意义的信息单元，即文件[3]。文件的构造、命名、存储、使用、保护和实现方法都是由这部分实现的，与此同时，该模块还将相对于卷的字节偏移转换成相对于文件的字节偏移。

由此可知，在存储栈中，下层驱动程序为上层驱动程序提供了虚拟化接口，从而屏蔽了底层特性。每一层都实现了一级抽象，以流水线的方式将无意义的 01 码转换成有意义的文件。

依据该思想，本文设计的虚拟文件系统便是在文件系统驱动程序和卷管理器磁盘驱动程序之间插入一层文件系统过滤驱动程序——虚拟卷处理程序，通过创建虚拟卷来对下层卷管理器提供的数据块进行重新映射，从而屏蔽下层实际的分卷及数据分布情况。而对于提供给上层文件系统的接口完全模拟卷管理器所提供的接口，从而使文件系统感觉不到丝毫的差异。

基于这种思想设计的虚拟文件系统在容灾系统中的优势明显，主要表现在以下 4 个方面：

(1)存储位置透明：数据备份位置对用户而言是透明的，用户看到的始终是自己创建的虚拟卷。

(2)存储结构透明：用户不需要了解底层存储设备是由哪些存储介质构成的。

(3)维护管理透明：可以透明替换损坏的磁盘，同时可以动态地扩充磁盘容量。

(4)数据迁移透明：可以根据磁盘容量的需求和用户对数据的访问次数，自动调节数据的存储位置。

3.3 虚拟存储映射表的构建

在系统中，虚拟存储映射原理尤为关键，虚拟存储的映射关系如图 3 所示。

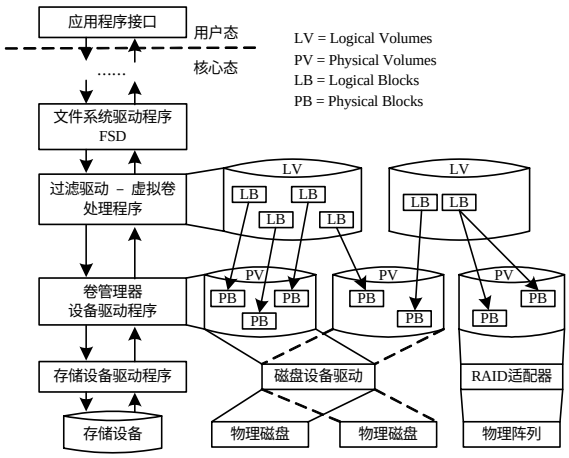

图 3 虚拟存储映射关系

虚拟卷的大小与物理卷的大小没有直接关系，用户可以根据需要创建适合的虚拟卷，而在卷内部，根据文件系统的可寻址数据块来实现映射。在未插入该过滤驱动层之前，文件系统的可寻址数据块是 Physical Blocks(PB)，而插入了过滤层之后，文件系统的可寻址数据块就变成 Logical Blocks(LB)，LB 与 PB 的数据结构是一致的，因此，对于文件系统而言，寻址 LB 和寻址 PB 没有任何区别。通过构建虚拟存储映射表，从而实现了虚拟化存储。

在构建虚拟存储映射表时，本文采用了虚拟内存映射页表[3]的设计方式。物理块是以簇为单位的，对应构建的虚拟块也以簇为单位。物理块地址与虚拟块地址的构造方式类似，都是由相应的卷号加上数据块的偏移地址。由于虚拟块可以映射到任何一个物理块上，完全不受物理卷位置的影响，因此需要卷映射表和块映射表 2 个表来实现虚拟映射，其具体的映射过程如图 4 所示。块映射表中的有效位表示该虚拟卷是否映射到有效的物理块上。

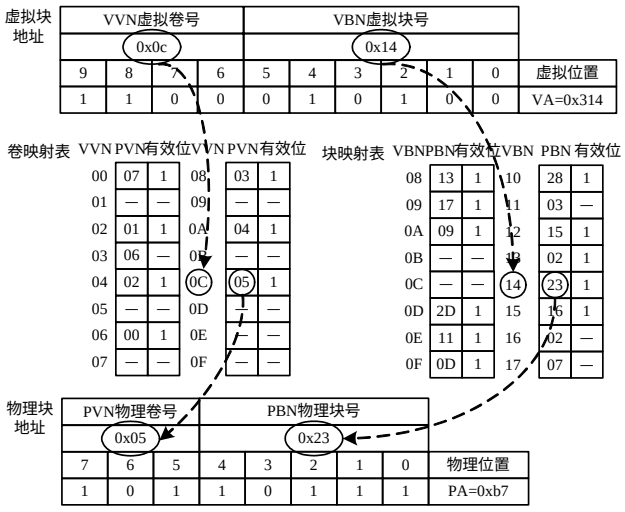

图 4　虚拟存储映射过程

一次完整的映射过程如下：虚拟卷处理程序根据虚拟块地址(0x314)中的虚拟卷号(0x0c)，查找卷映射表，得到物理卷号(0x05)；再根据虚拟块地址(0x314)中的虚拟块号(0x14)，查找块映射表，得到物理块号(0x23)，拼接到一起就得到了相应的物理块地址(0xb7)。

该虚拟映射表驻留在当前实现虚拟化的第一个物理磁盘的尾部，随着数据量不断的增大，该映射表(特别是块映射表)也会不断扩大，但由于表结构简单，每条记录仅限于字节量级，因此相比于庞大的磁盘容量，基本不会造成影响。

4　虚拟存储映射机制在容灾系统中的应用

本文设计的虚拟文件系统实现于容灾服务器端，客户端通过 Windows 远程文件系统访问服务器，但客户端只能看到系统为用户创建的一个自定义大小的虚拟卷。在该虚拟卷上存放着用户的备份数据，其他用户的备份数据对该用户是透明的，这样就保证了各个用户备份数据的独立性，从而在一定程度上保障了数据安全。

在上述机制中，文件系统只与虚拟卷进行通信，实现了文件与虚拟块之间的连接，而虚拟块与物理块之间的连接对文件系统来说是完全透明的，因此，可以很轻松地修改虚拟块与物理块之间的映射关系而对文件系统不产生任何影响，这对于实现容灾的数据迁移等功能提供了便利，主要体现在以下 2 点：

(1)损坏磁盘的透明替换

在传统的容灾系统中，服务器端的存储磁盘损坏之后，会将哪些用户备份了哪些数据到服务器上等相关信息完全丢失。用户需要根据自己的备份需求重新执行备份，构建一个全新的数据备份视图。这个过程难度较大并且极其繁琐，严重影响了用户的日常工作。

本文提出的解决方法为：在用户创建虚拟卷时，将虚拟存储映射表、容灾策略等元数据在其他容灾服务器上进行多副本备份，并且每执行一次备份都更新相关副本。当磁盘损坏后，将执行以下几步操作：

1)容灾服务器请求复制虚拟存储映射表等元数据副本到新更换的磁盘上。

2)根据这些元数据执行数据重构。

3)在数据重构执行完毕后，修改相应的卷映射表和块映射表。

在整个过程中，用户看到的备份视图与原来的视图一样，而且也不会对用户的任何应用操作(包括用户的备份操作)产生影响。

(2)备份数据分级、智能管理

在一个容灾系统中，每个容灾服务器需要管理多个客户端，因此，备份数据的存储量会很大。常规的处理方法有：1)提供多个大容量磁盘或者磁盘阵列；2)按照备份的时间顺序将部分数据迁移到离线的磁带设备上。这 2 种方法都有其明显的缺点，前者增大了容灾的成本，不适合大量数据的备份存储；后者会对数据恢复速度造成一定的影响。

本文提出的解决方法为：在上述第 2 个方法的基础之上，对数据迁移实施智能化处理。数据在磁盘、磁带之间的存储策略模仿高速缓存的缓存策略[3]。每个用户在容灾服务器端占用的磁盘容量都有上限值。当需要实现数据迁移时，执行以下几个步骤：

1)判断当前用户在服务器端的磁盘占用量是否超出最大值。未达到，不执行任何操作；达到，执行 2)。

2)计算超出的容量大小。

3)按照“用户修改次数最少”策略，找出相应容量的数据块。

4)执行数据迁移，并做好相关记录。

5)修改虚拟存储映射表，标明该虚拟块上的数据被换出了(在图 4 中，虚拟块地址有对应的物理块地址，但有效位为 0，表明物理块被换出到磁带上了)。

在整个过程中，磁盘设备在系统中起到高速缓存的作用，配合“用户修改次数最少”的换入换出策略，极大地提高了用户访问备份数据的速度和数据存储容量的大小。

5　结束语

本文针对现有容灾系统存在的问题，设计了一个应用于容灾服务器端的虚拟文件系统，并结合具体的应用实例对该文件系统中的虚拟存储映射机制进行描述。该系统主要具有以下特点：(1)客户端只能访问容灾服务器为其创建的虚拟卷，保证了各个用户备份数据的独立性；(2)系统中的虚拟存储映射机制模仿虚拟内存映射原理，并根据容灾系统的实际情况进行了改进，实现了容灾系统的透明化管理；(3)采用了“用户修改次数最少”策略实现磁盘与磁带设备之间的换入换出，提高了容灾系统的有效性和灵活性。但该系统还未能实现数据的透明备份与恢复，这也是下一步需要解决的问题。

（下转第 41 页）