第7章　虚拟存储

7.1　虚拟存储概述

存储技术从DAS到NAS和SAN，存储系统的性能已获得很大提高。随着存储系统的规模逐渐扩大，由于NAS和SAN存在难以克服的不足，因而迫切需要一种新的方法来改善存储系统的管理问题，虚拟存储技术因此应运而生。它可以对现有的各种存储设备和存储子系统进行整合，对存储管理进行优化。网络存储和虚拟存储都是为了解决“如何管理好存储”这样一个最基本的问题。

另一方面，随着数据量的爆炸性增长，要满足存储的管理、异步平台数据的共享、存储系统的可用性和可扩展性方面的要求，就必须采用虚拟存储技术，虚拟存储已逐渐成为存储的发展方向。

虚拟存储是指对存储服务和设备进行虚拟化，能够在对下一层存储资源扩展时进行资源合并，降低实现的复杂度。也就是通过对底层的存储硬件资源进行抽象化，而展现出来的一种逻辑表现。它通过将实体存储空间（例如磁盘）进行逻辑的分隔，组成不同的逻辑存储空间，以一个逻辑存储实体代表底层复杂的物理驱动器，屏蔽了单个存储设备的容量、速度等物理特性，而且也屏蔽了底层驱动器的复杂性以及存储系统后端拓扑结构的多样性，极大地增强了数据的存储能力、可恢复性和性能表现。

通过虚拟存储技术，在储存系统与服务器之间添加新的软件或硬件层，使应用不再需要了解数据寄存于哪个服务器、分区或储存子系统。管理员能够识别、提供和管理分散的储存，就如同在一个单一的资源中一般。而通过储存虚拟化，可用性也得到了提高。

因此，虚拟存储技术与传统技术相比，它具有更少的运营费用和更低的复杂性，简化了物理存储设备的配置和管理任务，同时还能够充分利用现有的物理存储资源，避免了存储资源的浪费。其将存储设备的使用与管理分离的特性，为使用者提供统一的、抽象的逻辑视图，从而可以极大的提高存储系统的管理效率和物理资源利用率，支持存储系统的资源和数据共享，实现透明的可扩展和高可用，是网络存储技术中的关键技术之一。随着应用需求和网络技术的发展，虚拟存储技术也在不断的演进和发展之中。

7.2　虚拟存储的原理

随着应用需求和网络技术的发展，虚拟存储技术也在不断地演进和发展。虚拟存储实际上就是提供一个统一的逻辑存储空间给服务器，服务器所能看到的是一个逻辑的存储空间，而不知道实际物理存储设备的位置、路径和每台设备的具体特征。

使虚拟存储技术时，操作系统所看到的存储与物理存储是不同的。虚拟存储技术将存储设备虚拟成逻辑存储区间，并统一进行管理，使存储管理具有灵活的伸缩性。在虚拟存储环境下，单个存储设备的容量、速度等物理特性被屏蔽掉了，所有的存储资源在逻辑上被映射为一个整体，以单一透明的存储视图向用户呈现。存储管理的常规操作，如系统升级、建立和分配虚拟磁盘、扩展存储空间容量等可由虚拟存储管理层自动进行。

由此可见，虚拟存储的核心工作是物理存储设备到单一逻辑存储资源池的映射。通过虚拟存储技术，把各种异构的存储资源统一成对用户来说是单一视图的存储资源（storage pool），为用户或应用程

序提供虚拟磁盘或虚拟卷，并为其隐藏或屏蔽具体存储设备的各种物理特性。利用虚拟存储技术，同时采用 Striping、LUN Masking、Zoning 等技术，用户可以根据自己的需求，方便地将大的存储池分割、分配给特定的主机或应用程序，实现存储池对服务器的动态而透明地增长与缩减。

在图 7-1 中，物理存储是 Disk，虚拟化后，操作系统面对的是 LUMs，用户使用的是 Storage Group。所有的存储管理操作，如系统升级、改变 RAID 级别、初始化逻辑卷、建立和分配虚拟磁盘、存储空间扩容等比从前的任何存储技术都更容易。只有采用了虚拟存储的技术，才能真正屏蔽具体存储设备的物理细节，为用户提供统一集中的存储管理。

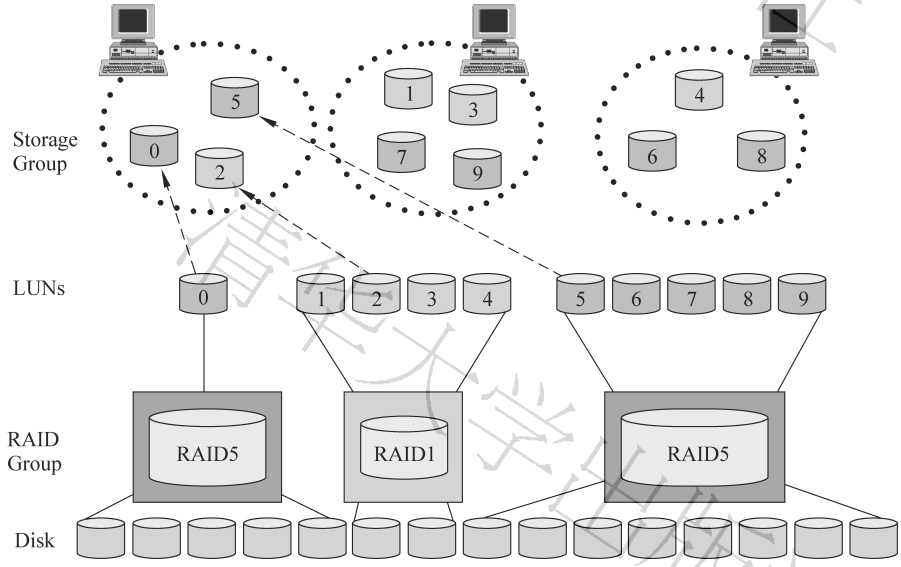

图 7-1　虚拟存储

虚拟存储技术的关键，是安全可靠的动态存储池，虚拟存储系统必须具备有的功能如下。

(1) 存储协议的自由转换(如从 SCSI 到光纤通道协议，能够支持异构存储和服务器环境)。

(2) 支持高可用性和高性能 SAN 存储配置(如指定主从镜像和空闲驱动器、产生合成式驱动器、联结多个存储子系统构成单一驱动器、实现集中治理以及灵活的存储容量扩充)。

(3) 具有可视性和可治理性，能够在更新和恢复等突发事件发生时及时通知管理员。

(4) 可以实现定时自动备份和恢复。

(5) 可以实现数据高速缓存。

(6) 可以控制主机访问不同的存储设备分区。

虚拟存储技术实际上就是把多个存储介质模块通过虚拟手段集中管理起来，所有的存储模块在一个存储池中进行统一管理，实现同构或异构的多个存储设备的统一管理，向用户提供一个大容量、高数据传输带宽的存储系统，实现了存储器物理管理与逻辑管理的分离和存储器的透明化访问。

7.3　虚拟存储的拓扑结构

目前的虚拟存储技术发展尚未形成统一标准，从虚拟存储的拓扑结构来说主要分为对称结构与非对称结构两种。对称结构是指在存储设备和应用服务器的数据路径上实现存储设备的虚拟，其数据和控制信息共用同一传输路径，其虚拟功能通过运行在虚拟控制器上的虚拟管理软件进行实现，这种结构

清华大学出版社

也称为带内存储虚拟技术，如图 7-2 所示。

非对称结构的虚拟存储由存储网络中的一台装有虚拟管理软件的独立服务器实现存储虚拟功能，该服务器完成存储设备的逻辑映射、存储分配、数据安全保障等元数据的管理，应用服务器首先通过访问元数据服务器获取映射后虚拟设备，然后通过数据通路直接访问存储设备，因此实现了数据和指令在不同的路径上的传递，这种结构也称为带外存储虚拟化技术，其结构如图 7-3 所示。

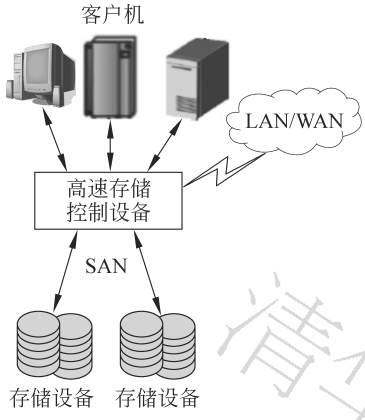

图 7-2　对称结构的虚拟存储

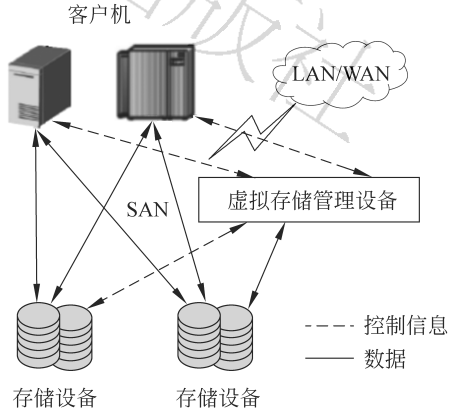

图 7-3　非对称结构的虚拟存储

虽然使用对称式结构可节省硬件投资，但是在虚拟存储控制设备点容易造成网络拥塞，形成系统瓶颈，使性能降低。因此，需要增加缓存和采用良好的缓存策略。此外，一旦出现故障，容易产生单点失效，故在实际应用中这种结构往往采用冗余配置。

使用非对称式结构时，由于数据在专用的数据通道上传输，因此减少了前端局域网的带宽占用，减少了网络延迟，提高了系统性能。此外，这种结构避免了系统的单点故障和瓶颈，只是在一定程度上增加了用户投资。

从虚拟存储的实现原理也可以分为数据块的虚拟、磁盘的虚拟、磁带库的虚拟、文件系统的虚拟与文件/记录的虚拟，如图 7-4 所示。

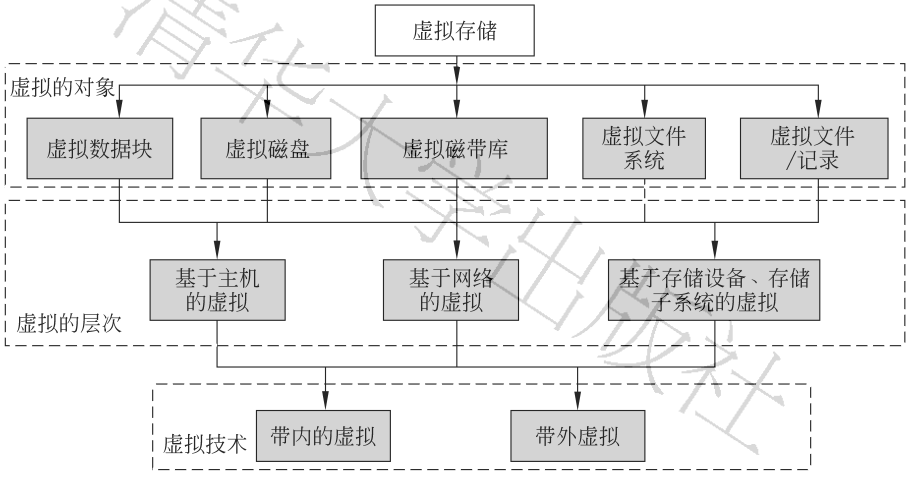

图 7-4　虚拟存储

1. 虚拟数据块

虚拟数据块是为用户的应用程序提供逻辑存储的一种存储服务，数据块级别的虚拟存储对用户抽象了存储的真实物理地址。在软件层面，它解析逻辑 I/O 请求，将其映射成物理地址。该技术能提供自由可伸缩的存储容量，对用户是透明的。

数据块级别的虚拟存储技术可以应用到直连在虚拟引擎上的内部存储，也可以应用到网络上的外部存储。这些外部存储可以是来自同一厂商的同构存储，也可以是来自不同厂商的异构存储。数据块虚拟存储方案利用虚拟的多端口并行技术，为多台客户机提供了极高的带宽，最大限度上减少了延迟与冲突的发生，在实际应用中，数据块虚拟存储方案以对称式拓扑结构为表现形式。

2. 虚拟磁盘

虚拟磁盘是指在本地计算机中虚拟出一个远程计算机的磁盘，使用时如同是在本机上的硬盘一样。虚拟磁盘不是实际硬件。

最典型的虚拟磁盘是磁盘阵列。磁盘阵列是把多个磁盘组成一个阵列，视为单一磁盘使用，它将数据以分条的方式储存在不同的磁盘中，用户不必规划数据在各磁盘的分布。同时，采用数据校验等有效的存取控制机制，增加了数据的容错性能和安全性。存取数据时，阵列中的相关磁盘一起动作，大幅减低了数据的存取时间，提高了磁盘空间的使用率。显而易见，磁盘阵列控制器是典型的设备级虚拟存储实现。

3. 虚拟磁带库

磁带库的虚拟本质上是用磁盘阵列硬件设备通过软件模拟磁带备份的形式。对于存储管理员来说，它就是一个磁带库，对它的管理如同管理一个物理磁带库一模一样。虚拟磁带库采用基于 RAID 保护的磁盘阵列，从而将备份的可靠性较常规磁带备份提高了若干量级。封闭式结构的磁盘介质本身的 MTBF(平均无故障间隔)一般为开放式结构磁带介质的 5 倍以上。磁带库的虚拟不仅解决了传统磁带库维护负担高、备份失效率高以及备份恢复能力不佳的问题，而且也改变了磁盘备份容易被误删除或被病毒感染以及不便于在 SAN 环境中统一管理和优化使用的劣势。虚拟磁带库技术具备性能高、故障率低、可靠性高、成本投入低以及运营成本低等多项优势。

4. 虚拟文件系统

虚拟文件系统(virtual file system，VFS)是物理文件系统与服务之间的一个接口层，定义了所有文件系统都支持的基本的、概念上的接口和数据结构，允许在操作系统间使用跨文件系统的文件操作。它并不是一种实际的文件系统，只存在于内存，而不存在于外存空间。

虚拟文件系统着重解决大规模网络中文件共享的安全机制问题。通过对不同的站点指定不同的访问权限，可保证网络文件的安全。在实际应用中，虚拟文件系统存储方案以非对称式拓扑结构为表现形式。

5. 文件的虚拟

虚拟文件是对各种不同的文件系统进行聚集，通过被称为全局命名空间的逻辑层来实现不同文件系统之间无缝进行数据迁移和集中管理，用户无须关心文件所在的具体物理位置以及何时被移动到什么位置，从而极大的方便了服务器的管理、文件的重组和存储的汇聚，是非结构化数据的一条有效管理途径。

虚拟文件使物理文件服务器的基本细节和 NAS 设备抽象化，并跨物理设备创建了一个统一命名空间，通过把多个文件系统和设备都统一到一个单独的命名空间下，文件虚拟化提供了一个单一的文件和

目录。依据命名空间方式的不同，文件虚拟化可划分为集成平台的命名空间、集群存储派生的命名空间和网络虚拟的命名空间 3 种方式。

(1) 集成平台的命名空间是主机文件系统的扩展，非常适合多站点协作，但由于缺乏文件控制而往往限制在单一文件系统或操作系统内使用。

(2) 集群存储系统结合集群和先进的文件系统技术，把多文件服务器整合到一个单一的高可用系统内，创建了一个可支持不断增多的 NFS 和 CIFS 请求的模块化可扩展系统，其命名空间是一个统一的、跨所有集群结点的、共享的命名空间。

(3) 网络虚拟的命名空间是由网络设备(即网络文件管理器)创建的，这些设备在用户和存储之间展现一个虚拟的命名空间。网络虚拟的命名空间非常适用于分层存储和其他的非中断数据迁移情况。

虚拟存储技术主要应用于企业级存储用户和存储网络。它对硬盘驱动器、RAID、磁带库和光碟库等存储设备进行虚拟，使用户能够在存储容量、性能、可靠性、连通性和易管理性方面更随心所欲地使用存储系统。

7.4 虚拟存储的实现模式

虚拟存储技术可以简化存储模型，提高灵活性并支持异构的存储环境。根据虚拟技术部署方式的不同，从已推出的虚拟产品的结构来看，可以分为基于主机的虚拟技术、基于存储设备的虚拟技术和基于网络的虚拟存储技术 3 种基本结构型式。每种架构都有其独特的优势，但也都存在一定的局限性。在实际的虚拟网络存储系统中，所采用的往往是这 3 种之中的一种或几种的组合。

7.4.1 基于主机的虚拟存储

基于主机的虚拟存储是指存储设备与服务器集于一体，虚拟存储的应用通过特定的软件在主机服务器上完成，经过虚拟的存储空间可以跨越多个异构的磁盘阵列。此时服务器有 4 个层次：顶层为应用软件层(如监控系统)，次层为操作系统层(如 Linux 或者 Windows 操作系统)，接下来是虚拟管理软件层(如 Windows 操作系统的自带卷管理器)，底层为物理存储产品层(如硬盘或者磁带等)。基于主机的虚拟存储应用，可以提高灵活性、扩大存储空间，又不需要大的投入，目前应用较为广泛。

例如，有一个文件服务器，为了优化文件服务器的性能，扩大存储空间，需要其能够像多个磁盘阵列中存储、读取数据文件。这样不仅可以实现磁盘之间的负载均衡，提高文件访问的效率，而且不同磁盘阵列之间还可以实现数据的冗余校验，提高数据的安全性。像这样的应用，就可以采用基于主机的虚拟存储应用。如果磁盘阵列是异构的，即磁盘阵列是不同的类型或者所采用的存储介质是不同的，也适合基于主机的虚拟存储。支持异构的存储介质正是虚拟存储应用的一大特色。

基于主机的虚拟存储技术其核心就是位于第三层的虚拟存储技术管理软件。在现实应用中，该功能通常是由操作系统下的逻辑卷管理软件来实现。如 Windows 操作系统下面的自动卷管理软件。也可以采用第三方虚拟卷管理软件(如厂商提供的管理软件)。从在兼容性、性能上看，可优先考虑采用操作系统自带的卷管理软件。通过这些软件可以在操作系统与存储设备之间建立一个虚拟层。通过这个虚拟层，可以将存储设备组成逻辑磁盘与逻辑卷。

从功能上说，这个逻辑卷跟 Windows 操作系统下的动态磁盘很类似。动态磁盘技术就是将一块磁盘分割成多个逻辑卷。而采用逻辑卷的最大好处就在于磁盘容量的管理。如可以不用格式化，就可以调整各个逻辑卷的大小。缺点是仅用动态磁盘技术，只能够组合一块磁盘。如果想要将多块磁盘组合成一块逻辑磁盘，则需要其他技术，如磁盘阵列或者虚拟存储管理软件的支持。如果单从逻辑卷的管理上，就跟动态磁盘很类似。

在部署基于主机的虚拟存储应用时，主要是要考虑磁盘的空间规划。虽然每个逻辑卷的大小可以动态的调整，但由于主机空间的限制，没有足够大的空间来放置很多磁盘，故对于存储空间的总量需要预先规划（一般带 RAID 的服务器，受限于磁盘的体积、散热等因素，挂接的磁盘数不可能太多）。然后再根据后续的需要，来调整各个逻辑卷的大小，基于主机的虚拟存储应用其自身的实现方法决定了在性能上要比其他应用模型要逊色一些。对于性能要求特别高或者用户并发访问数量特别多的企业，可能不适合这个方案，主要原因是性能跟不上。

基于主机的虚拟可以通过在每台服务器上安装 LVM（logical volume management，逻辑卷管理）程序来实现，如图 7-5 所示。LVM 隐藏了物理存储设备的复杂性，向操作系统提供存储资源的一个逻辑视图。主机可以通过多条路径到达共享存储，存储目标也可以随意组合。

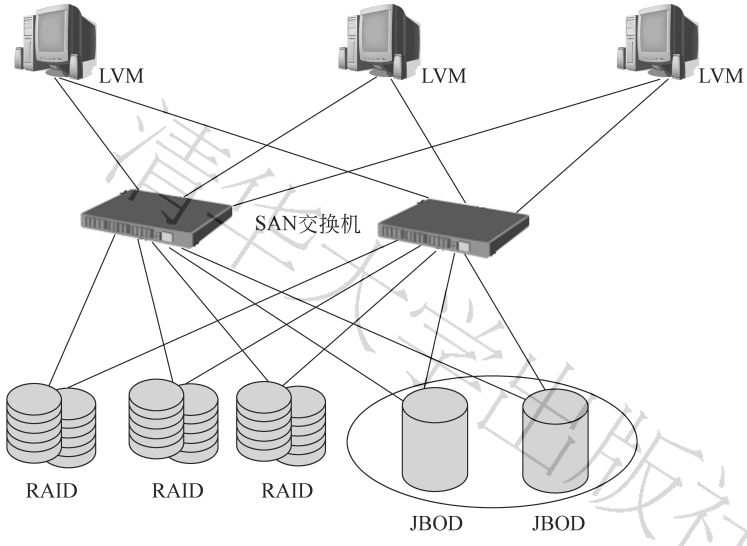

图 7-5　基于主机的虚拟存储

图 7-5 中，JBOD（just a bunch of disks，磁盘簇）是存储领域中一类重要的存储设备，是在底板上安装的带有多个磁盘驱动器的存储设备，又称为 Span。和 RAID 不同，JBOD 没有前端逻辑来管理磁盘上的数据分布，与之相反，每个磁盘进行单独寻址，作为分开的存储资源、基于主机软件的一部分或者 RAID 组的一个适配器卡。JBOD 不是标准的 RAID 级别，RAID 系统在多个磁盘上冗余地存储了同样的数据，而这多个磁盘在操作系统看来就像一个磁盘。虽然 JBOD 也让多个磁盘看来似乎只有一个，但它是通过把多个驱动器合并成一个大的逻辑磁盘来做到这一点的。JBOD 使用独立的磁盘并没有带来任何好处，也不能提供任何 RAID 所能带来的容错或是更好的性能。

由于基于主机的虚拟存储主要依赖安装在主机之上的软件或硬件来提供存储虚拟管理功能，因此必须把软件或硬件安装在每一台主机上。此种方法因各类主机和存储环境不同，不但是非常复杂和不开放，而且会产生许多安全问题。此外，基于主机的虚拟以软件方式实现时会占用主机的计算资源，会影响存储访问之外其他应用的运行，若采用硬件方式实现协议处理，往往会带来较高的成本开销，很难通过其他技术来解决。这也在很大程度上限制了其应用范围。一般来说，如果对于存储的性能要求比较高或者用户数量比较多（例如 163 等基于互联网提供邮件服务的机构），就不适合采用这个模型；如果用户比较少（例如一般企业内部自用的邮件服务器），则这个基于主机的虚拟存储应用模型在性能上已经可以满足企业的需求了。

基于主机的虚拟存储方法也有可能影响到系统的稳定性和安全性，由于有可能导致不经意间越权访问到受保护的数据。这种方法要求在主机上安装适当的控制软件，因此一个主机的故障可能影响整个 SAN 系统中数据的完整性。软件控制的存储虚拟还可能由于不同存储厂商软硬件的差异而带来不必要的互操纵性开销，所以这种方法的灵活性也比较差。

总之，在使用基于主机的虚拟存储应用模型时，要扬长避短。在合适的场合使用合适的虚拟存储模型，往往可以提到事半功倍的作用。

7.4.2 基于存储设备的虚拟存储

设备级的虚拟存储包括两个方面，其一是对设备物理特性的仿真，例如利用其他存储介质仿真一个磁盘块设备，该磁盘具有磁盘驱动器的一切特性，但是数据却存放在其他类型的设备上；其二针对的是虚拟设备的构建。从功能上看，它与主机级虚拟技术类似，不同的是它通常需要额外硬件的支持，可将磁盘驱动器、RAID、SAN 设备等组合成新的存储设备，其虚拟管理模块是嵌入在硬件中实现的。

由于虚拟化管理软件嵌入在硬件实现，可提高虚拟化处理和虚拟设备 I/O 的效率。这种方法能获得较高的性能和可靠性，管理方便，缺点是成本过高。

最典型的虚拟存储设备是磁盘阵列(RAID)。RAID 的虚拟化是由 RAID 控制器实现的，它提供硬件 RAID 或软 RAID 技术，将多个物理磁盘按不同的分块级别组织在一起，通过板上 CPU 及 RAID 管理固件来控制及管理硬盘，解释用户的 I/O 指令并将它们发给物理磁盘执行。从而屏蔽了具体的物理磁盘，为用户提供了一个统一的具有容错能力的逻辑虚拟磁盘，用户对 RAID 的存储操作就像对普通磁盘一样。

基于存储设备的虚拟方法依靠存储设备子系统提供虚拟存储管理功能(多为硬件实现)，主要包括将一个物理设备虚拟成为多个逻辑存储设备(如分区)和将多个物理存储设备虚拟成为一个逻辑存储设备(如 RAID)等。这种虚拟化技术在存储设备中，特别在高端存储阵列设备中已被广泛支持。

类似于虚拟服务器，虚拟存储是将物理存储系统抽象化，隐藏复杂的物理存储设备，将来自多个网络存储设备的资源整合为资源池，对外部来说，相当于单个存储设备，连同虚拟化的磁盘、块、磁带系统与文件系统。虚拟存储的优势之一就是该技术可以更好地管理存储设备，提高执行诸如备份、恢复、归档任务的效率。

虚拟存储架构维护着一份虚拟磁盘与其他物理存储的映射表。虚拟存储软件层(逻辑抽象层)介于物理存储系统与运行的虚拟服务器之间。当虚拟服务器需要访问数据时，虚拟存储抽象层提供虚拟磁盘与物理存储设备之间的映射，并在主机与物理存储间传输数据。

虚拟存储与基于主机的存储虚拟的区别仅在于采用怎样的技术来实现。实现虚拟存储可能直接通过存储控制器也可能通过 SAN 应用程序。同样的，某些部署存储虚拟化将命令和数据一起存放(带内模式)而其他可能将命令与数据路径分离(带外模式)。

虚拟存储设备的典型例子是在智能磁盘子系统中的块级虚拟，这些存储系统采用 LUN 掩码和 RAID，通过 I/O 通道向多个服务器提供它们的存储容量。存储设备把物理硬盘集成在一起，形成虚拟盘，供服务器使用诸如 SCSI、光纤通道或 iSCSI 协议进行访问。基于存储设备的存储虚拟化如图 7-6 所示。

通常的企业级存储阵列已经通过 RAID 和镜像提供虚拟化。基于阵列的虚拟化可以对服务器完全透明，无须在服务器上安装任何代理软件。与基于阵列的虚拟化不同，存储阵列位于 SAN 的后端而不是中间，因此在性能上不会产生影响 SAN 的瓶颈。

基于存储设备的虚拟通常关注的是性能，虽然性能一般可以得到保障，但这些专用系统过于单一

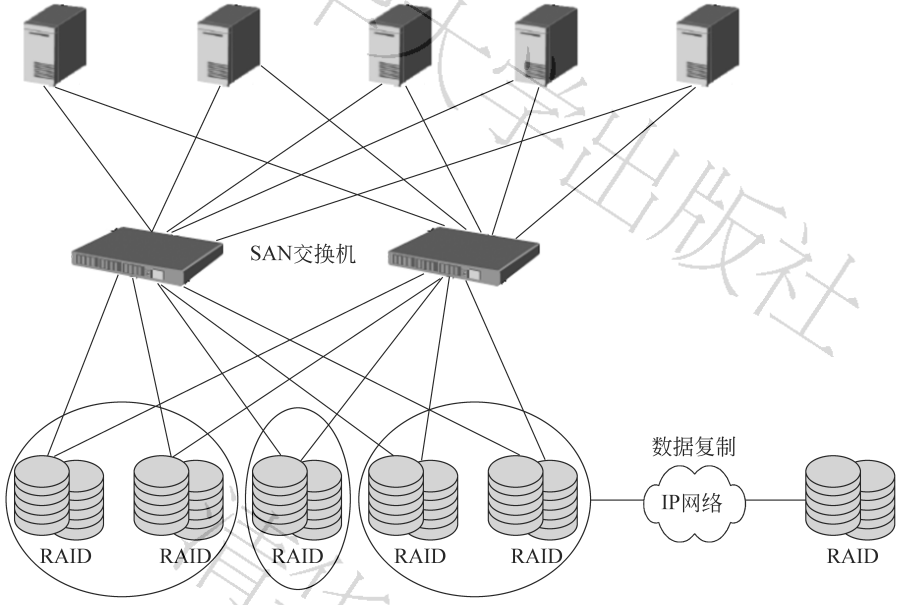

图 7-6　基于存储设备的虚拟存储

化，单台设备的升级空间不大，当管理多个这样的设备时，仍产生额外的管理成本；此外，为了确保性能通常会采用单一厂商的产品，多家厂商的产品很难共同使用，这会对系统扩容和升级带来不便。此解决方案比较适合小型系统使用。

作为虚拟服务器存储问题的解决方案，存储虚拟可以减少数据中心开支，提高商业灵活性并成为任何私有云的重要组件之一。

7.4.3　基于网络的虚拟存储

在基于网络的虚拟由网络本身来实现，把虚拟引擎移到 SAN 的心脏，作为一个独立的设备运行，或者用一个增强模块插入光纤通道 SAN 交换机、千兆位以太网交换机或路由器中。

在传统的网络存储访问中，服务器和存储目标之间的路径如图 7-7 所示，目标同时承载着数据在设备上如何存储的信息（称为元数据）和数据本身。例如，文件系统可能位于一个磁盘阵列上，阵列中包含元数据，例如文件名、属性和每个文件的数据所在的块地址列表。阵列的其余部分存储文件系统元数据所指向的数据块。服务器通过在它和存储设备之间的 SAN 中的传输路径访问元数据和数据本身。

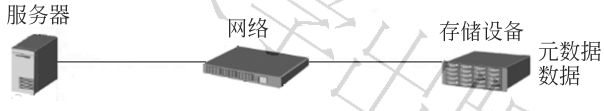

图 7-7　对元数据和数据本身的传统一访问方式

基于网络的虚拟是通过存储网络中的专用服务器或网络设备（交换机或路由器）来实现的。它能够支持多种网络（LAN、SAN、MAN、WAN）和网络传输协议（TCP、IP），可以将不同厂商、不同设备品牌、不同连接方式的磁盘阵列组成一个虚拟的存储池，并将虚拟存储池的存储空间映射给网络上的应用服务器使用。此外，它还可以根据应用的变化在线调整虚拟存储空间和数据传输通道。

基于网络的虚拟存储是在主机和存储设备子系统之间的网络上实现虚拟存储功能，根据数据通路

和管理通路的耦合情况可分为带内(in band)和带外(out of band)两种模式,如图 7-8 所示。

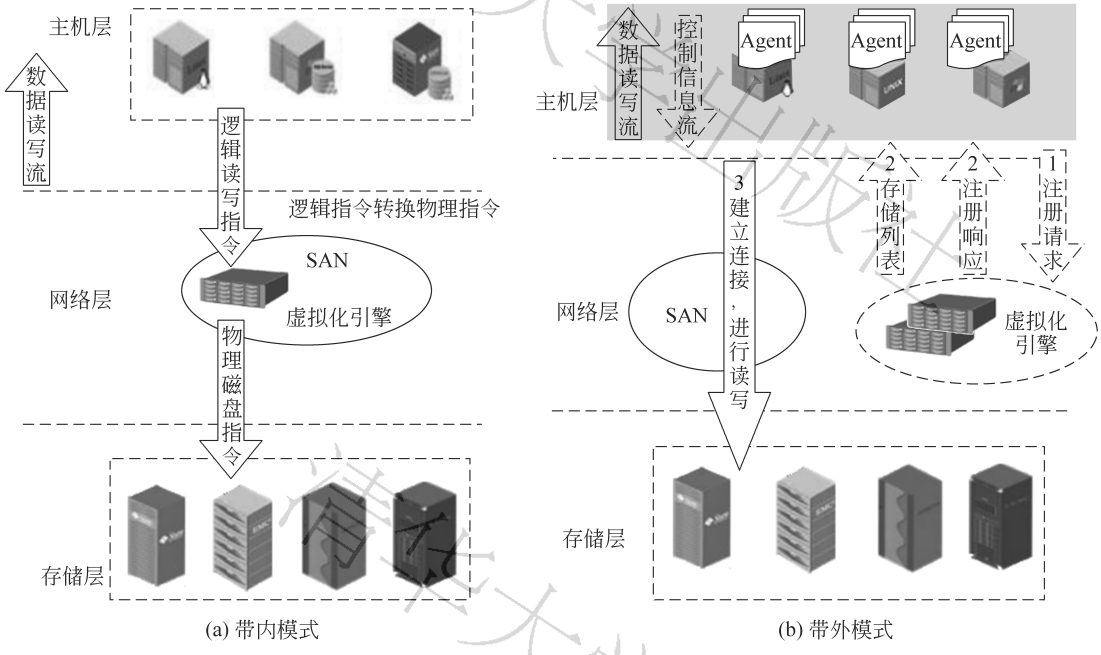

图 7-8 带内虚拟化与带外虚拟化

带内虚拟技术是在数据读写的过程中,在主机到存储设备的路径上实现存储虚拟化;带外虚拟技术,是在数据读写之前,就已经做好了虚拟工作,而且实现虚拟的部分并不在主机到存储设备的访问路径上。所以带内虚拟技术可以基于主机、设备和网络实现,而带外虚拟技术则只能是基于存储网络实现。

在带内模式中,负责虚拟化管理和控制的设备被置于应用服务器和存储设备之间的数据通路上,其最大的好处就是简化拓扑结构,可以集中管理多种连接设备。但是,带内虚拟存储结构由于控制命令和存储数据共用一条通路,因此虚拟存储层设备处容易造成网络拥塞而形成系统瓶颈,降低系统性能,而且严重的限制了系统的可扩展性。另外,这种结构的虚拟存储系统容易出现服务器到存储设备的单点故障,因此带内虚拟存储结构在实际使用中往往要做冗余配置。

在带外虚拟模式中,负责虚拟管理和控制的模块处于应用服务器和存储设备之外的独立控制通路上。带外虚拟存储结构中,存储数据在专用的数据通道上传输,而控制管理信息通过管理通道传输。这样,减少了网络延迟,增加了带宽的可升级性,从而提高了系统性能。同时这种结构还避免了系统的单点故障和瓶颈。然而,这种方式又引出了一些主机系统的易操作性问题。也就是说,需要加载、维护和修改主机系统软件。

为解决主机系统的易操作性问题,出现了一种基于交换机的带外虚拟化方式。这一方式利用智能 SAN 交换机作为平台,构建以网络为基础的存储虚拟化。这些带有专门的端口级处理器的交换机能快速检验并重新分配 I/O 指令(从逻辑地址转换到物理地址)。以前由主机接口程序管理的主机元数据被加载到智能端口的闪存上,不再需要主机系统软件。元数据服务器不与主机交换信息,而是和智能端口交换信息,确保总能为通过这些端口存储信息的主机提供正确的映射信息,从而使其易操作性大大提高。这种基于交换机的结构方式具有很强的可扩展性,可以满足大规模数据中心推广存储虚拟化的需求。

网络级的虚拟存储是基于网络实现的，可简化大规模存储系统的管理。通常，这种虚拟实现方式也需要硬件的支持，即通过集成虚拟化管理软件构建管理结点，实现资源的集中管理。网络虚拟化可确保存储技术能够跨不同厂商的设备工作，让不同的存储空间看上去和工作起来就像统一的存储资源一样。

网络级虚拟存储通过在主机、交换机或路由器上执行虚拟化模块实现。网络是实现虚拟存储中最具有逻辑含义的部分，被认为能实现最为“开放”的虚拟化。

网络级虚拟方法要比通过纯软件实现的主机级虚拟方法方便。主机级虚拟方法需要在每台主机上安装、配置和维护软件，在管理上比较累赘。网络级虚拟存储可以提供一种中央虚拟化方式将网络中的存储资源集中起来进行管理，仅需要少量的管理人员。

网络级虚拟方法有下面几种实现方式。

(1) 基于互连设备的虚拟化。基于互连设备虚拟化方法可以是对称的，也可以是不对称的。在对称的方式下，控制信息和数据走在同一条通道上，互连设备之间的网络通道有可能成为瓶颈。在非对称的方式下，控制信息和数据走在不同的路径上。与对称方式相比，它更具有可扩展性。这种处理方法需要有一台设备充当控制结点，负责全局存储空间的管理和配置。除此以外，每台设备当中还需要嵌入或运行一个代理程序，它一方面与控制结点通信实现虚拟化配置，另一方面与存储用户通信实现 I/O 命令的处理。

(2) 基于交换机的虚拟化。基于交换机的虚拟化存储，虚拟化功能模块嵌入到交换机的固件中或者运行在与交换机相连的服务器上，对与交换机相连的存储设备进行管理。优点是实现比较简单，而且可在异构设备环境中实现互操作。缺点是由于多种虚拟功能集中在交换机处实现，可能会形成处理上的瓶颈，而且一旦交换机出现故障，整个系统就无法正常运行。

(3) 基于路由器的虚拟化。基于路由器虚拟化存储，通过嵌入在路由器固件上的虚拟化功能模块实现。路由器位于用户到存储网络的数据通道中，可截取发往存储网络的 I/O 命令，并对它进行处理。大多数控制模块存在于路由器的固件中，相对基于互连设备的虚拟方法，这种方法的性能更好。另外，由于不依赖于每个主机上运行的服务程序，这种方法比主机级或设备级虚拟方法更安全。当用户到存储网络的路由器出现故障时，将导致数据不能被访问，此时受影响的只有联结于故障路由器的用户，其他用户仍然可以通过其他路由器访问存储系统。由于路由器支持动态多路径配置，因而可通过路径切换的方法从其他的路由器寻求支持。

以上 3 种虚拟化架构各有其优缺点。主机级虚拟对于某些应用来说魅力最大，因为它们不需要附加任何硬件，但是对于追求高性能的应用场合，设备级虚拟显然更受欢迎。在网络级虚拟存储方法中，对于那些要求最大限度进行互操作的企业，基于交换机的虚拟方法比较适合。对于既要求高可靠性，又要求高性能的应用，基于路由器的方法是最优选择。基于互连设备的方法处于两者之间，它解决了一些安全性问题，能减轻单一主机的负载，同时又具有很好的可扩展性。

3 种虚拟存储技术可以单独使用，也可以在同一个存储系统中配合使用。对于一个大型的存储环境，综合应用 3 种虚拟方法能发挥整体优势，取得最好的效益。

7.5　虚拟存储管理工具

7.5.1　LVM

1. LVM 介绍

LVM(logical volume manager，逻辑卷管理器)是 Linux 环境下对磁盘分区进行管理的一种机制。LVM 是在磁盘分区和文件系统之间添加的一个逻辑层，为文件系统屏蔽下层磁盘分区布局，提供一个

抽象的存储卷，在存储卷上建立文件系统。通过 LVM 提高磁盘分区管理的灵活性。例如，将若干个磁盘分区连接为一个整块的卷组（volume group），形成一个存储池。可以在卷组上随意创建逻辑卷（logical volume），并进一步在逻辑卷组上创建文件系统。通过 LVM 可以让分区变得弹性，能方便的调整存储卷组的大小，并且可以对磁盘存储按照组的方式进行命名、管理和分配，前提是该分区是 LVM 格式的。

例如，按照使用用途定义 development 和 sales，而不是使用物理磁盘名 sda 和 sdb。而且，当系统添加了新的磁盘，通过 LVM，可直接扩展文件系统跨越磁盘，而不必将磁盘的文件移动到新的磁盘上，这样就可以充分利用新的存储空间。

LVM 通常用于装备大量磁盘的系统，但它同样适于仅有一两块硬盘的小系统。

2. 基本概念

Linux 环境下，每一个物理卷都被分成几个基本单元，即物理盘区（physical extent，PE）。PE 的大小是可变的，但是必须和其所属卷组的物理卷相同。在每一个物理卷里，每一个 PE 都有一个唯一的编号。PE 是一个物理存储里可以被 LVM 寻址的最小单元。与 LVM 相关的术语还有物理存储介质、物理卷、卷组、逻辑卷、物理盘区和逻辑盘区等。

（1）物理存储介质（physical storage media）。物理存储介质是指系统的物理存储设备——磁盘，例如/dev/hda、/dev/sda 等，它是存储系统最底层的存储单元。

（2）物理卷（physical volume，PV）。物理卷是指磁盘分区或从逻辑上与磁盘分区具有同样功能的设备（如 RAID），是 LVM 的基本存储逻辑块，与基本的物理存储介质（如分区、磁盘等）相比，包含有与 LVM 相关的管理参数。硬盘分区后（还未格式化为文件系统）使用 pvcreate 命令可以将分区创建为 PV，要求分区的 system ID 为 8e，即为 LVM 格式的系统标识符。

（3）卷组（volume group，VG）。类似于非 LVM 系统中的物理磁盘，卷组由一个或多个物理卷（PV）组成。可以在卷组上创建一个或多个 LV（逻辑卷）。将多个 PV 组合起来，使用 vgcreate 命令创建成卷组，这样卷组包含了多个 PV 就比较大了，相当于重新整合了多个分区后得到的磁盘。虽然 VG 是整合多个 PV 的，但是创建 VG 时会将 VG 所有的空间根据指定的 PE 大小划分为多个 PE，在 LVM 模式下的存储都以 PE 为单元，类似于文件系统的 Block。

（4）逻辑卷（logical volume，LV）。类似于非 LVM 系统中的磁盘分区，逻辑卷建立在卷组（VG）之上。在 LV 之上可以建立文件系统。如果说 VG 相当于整合过的硬盘，那么 LV 就相当于分区，只不过该分区是通过 VG 来划分的。VG 中有很多 PE 单元，可以指定将多少个 PE 划分给一个 LV，也可以直接指定大小来划分。划分为 LV 之后就相当于划分了可管理的分区，只需再对 LV 进行格式化即可成为普通的文件系统。

（5）物理盘区。每一个物理卷 PV 被划分为称为 PE 的基本单元（实际存储的数据都是存储在这里面的），具有唯一编号的 PE 是可以被 LVM 寻址的最小单元。PE 的大小是可配置的，默认为 4MB。所以物理卷（PV）由大小等同的基本单元 PE 组成。

（6）逻辑盘区（logical extent，LE）。LE 是逻辑存储单元，也即为 LV 中的逻辑存储单元，和 PE 的大小是相同的，并且一一对应。从 VG 中划分 LV，实际上是从 VG 中划分 VG 中的 PE，只不过划分 LV 后它不再称为 PE，而是成为 LE。

LVM 之所以能够伸缩容量，其实现的方法就是将 LV 里空闲的 PE 移出，或向 LV 中添加空闲的 PE。

如图 7-9 所示 LVM 抽象模型，展示了与 PV、VG、LV 三者之间以及与文件系统、硬盘关系。

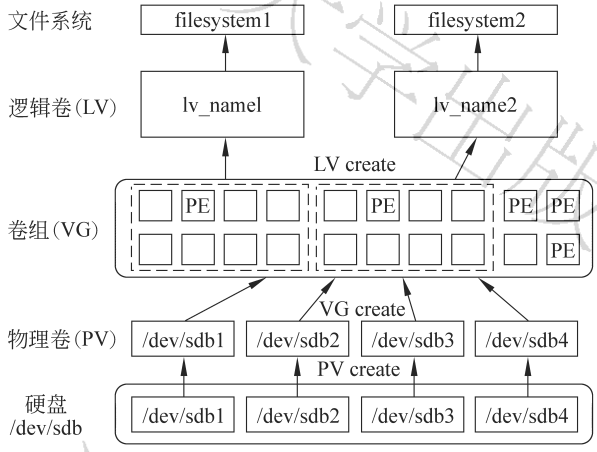

图 7-9　LVM 与 PV、VG、LV 三者之间关系

和非 LVM 系统将包含分区信息的元数据保存在位于分区的起始位置的分区表中一样，逻辑卷以及卷组相关的元数据也是保存在位于物理卷起始处的 VGDA（卷组描述符区域）中。VGDA 包括以下内容：PV 描述符、VG 描述符、LV 描述符、和一些 PE 描述符。

系统启动 LVM 时激活 VG，并将 VGDA 加载至内存，来识别 LV 的实际物理存储位置。当系统进行 I/O 操作时，就会根据 VGDA 建立的映射机制来访问实际的物理位置。

物理服务器存储设备一般由多个硬盘组成，例如在 Linux 下，有 sda、sdb、sdc，它们是不同的物理硬盘，但是可以通过创建物理卷（PV），将一块硬盘分成不同的物理卷。然后，卷组（VG）再将不同物理卷统一管理起来，形成一个大的磁盘空间，最后，在 VG 的基础上 LVM（Logical Volume Manager）划分出不同的逻辑卷，并进行使用。相对来说，使用 VG 和 LVM 有以下优势。

（1）扩容方便。VG 能够支持不同类型的物理硬盘，当存储空间不够时，可以增加新的硬盘到 VG 中。

（2）LVM 能够灵活调整大小。LVM 是裸设备，而不是文件系统，而 Libvirt（用于管理虚拟化平台的开源的 API，后台程序和管理工具）等虚拟化技术支持对 LVM 的读写，与文件系统相比，它的效率更高。

3. 常用命令

1）物理卷的一般维护命令

常用命令物理卷命令如表 7-1 所示。

表 7-1　物理卷命令

命　　令	作　　用
pvscan	在系统的所有磁盘中搜索已存在的物理卷
pvdisplay 物理卷全路径名称	用于显示指定物理卷的属性
pvdata 物理卷全路径名称	用于显示物理卷的卷组描述区域信息
PvchangeCx|--allocation {y|n}物理卷全路径名	用于改变物理卷的分配许可设置，物理卷的创建与删除命令

续表

命　　令	作　　用
pvcreate 设备全路径名	用于在磁盘或磁盘分区上创建物理卷初始化信息，以便对该物理卷进行逻辑卷管理
Pvmove 源物理卷全路径[目的物理卷全路径名]	用于把某物理卷中的数据转移到同卷组中其他的特别卷中

2）卷组命令

常用的卷组命令如表 7-2 所示。

表 7-2　卷组命令

命　　令		作　　用
vgscan		检测系统中所有磁盘
vgck [卷组名]		用于检查卷组中卷组描述区域信息的一致性
vgdisplay [卷组名]		显示卷组的属性信息
vgrename 原卷组名新卷组名		对已存在的卷组进行改名
vgchange -a y|n [卷组名]		改变卷组的相应属性。是否可分配
vgchange -l 最大逻辑卷数		卷组可容纳最大逻辑卷数
vgchange -x y|n [卷组名]		卷是否有效
vgmknodes [卷组名|卷组路径]		用于建立(重新建立)已有卷组目录和其中的设备文件卷组配置的备份与恢复命令
vgcfgbackup [卷组名]		把卷组中的 VGDA 信息备份到/etc/lvmconf 目录中的文件
vgcfgrestore -n 卷组名物理卷全路命名		从一个文件备份中恢复卷组的元数据。可以指定备份文件，如果没有指定备份文件，使用最近的一次
vgcreate 卷组名物理卷全路径名[物理卷全路径名]		用于创建 LVM 卷组
vgmove 卷组名		删除卷组
卷组的扩充与缩小命令	vgextend 卷组名物理卷全路径名[物理卷全路径名]	
	vgreduce 卷组名物理卷全路径名[物理卷全路径名]	
卷组的合并与拆分	vgmerge 目的卷组名源卷组名	合并两个已经存在的卷组，要求两个卷组的物理区域大小相等且源卷组是非活动的
	vgsplit 现有卷组新卷组物理卷全路径名[物理卷全路径名]	从源卷组(物理卷)移动到新的或现有的卷组中
卷组的输入与输出命令	vgexport 卷组名	允许设置系统未知的非活动卷组名称
	vgimport 卷组名卷组中的物理卷[卷组中的物理卷]	导入卷组。从不同的系统移动导出物理卷之后，vgimport 命令配合相应 map 文件可以让系统再次认出导出的卷组

3）逻辑卷命令

常用的逻辑卷命令如表 7-3 所示。

清华大学出版社

表 7-3　逻辑卷命令

命　　令	作　　用
一般命令	lvscan
	lvdisplay 逻辑卷全路径名[逻辑卷全路径名]
	lvrename 旧逻辑卷全路径名新逻辑卷全路径名
	lvrename 卷组名旧逻辑卷名新逻辑卷名
	lvchange
	e2fsadm -L ＋|-逻辑卷增减量逻辑卷全路径名
逻辑卷的创建与删除命令	lvcreate
	lvremove
逻辑卷的扩充与缩小命令	lvextend -L|--size ＋逻辑卷大小增量逻辑卷全路径名
	lvreduce q -L|--size ＋逻辑卷减小量逻辑卷全路径名

4）逻辑卷管理命令

常用的逻辑卷管理命令如表 7-4 所示。

表 7-4　逻辑卷管理命令

命　　令	作　　用
lvmdiskscan	检测所有的 SCSI、IDE 等存储设备
lvmchange -R|--reset	复位逻辑卷管理器
lvmsadc [日志文件全路径名]	收集逻辑卷管理器读写统计信息，保存到日志文件中
lvmsar 日志文件全路径名	从 lvmsadc 命令生成的日志文件中读取并报告逻辑卷管理器的读写统计信息

4. 命令实例

（1）查看当前系统是否装有 lvm。

```
#rpm -qa|greplvm
```

（2）创建物理卷（LVM 允许 PV 建立在几乎所有块设备上，如整个硬盘、硬盘分区、Soft RAID）。

```
# pvcreate /dev/sda
# pvcreate /dev/sdb1
```

（3）创建卷组。

```
# vgcreate test_vg /dev/sda /dev/sdb /dev/sdc/
```

（4）查看、验证卷组信息。

```
# vgdisplay
```

（5）创建逻辑卷。

清华大学出版社

① 创建线性 lv

```
# lvcreate -L 1G -n test_lv test_vg
```

②创建交错 lv

```
# lvcreate  -i 3 -I 4 -L 1G -n  test_lv test_vg
```

-L：指定逻辑卷大小，单位为 K、M、G；
-l：指定逻辑卷大小，单位为 PE；
-i：交错单位，本例中需要将此逻辑卷建立到 3 个 PV 上；
-I：交错参数：本例中交错参数为 4KB。
(6) 创建文件系统。
①创建 Ext2/Ext3 系统。

```
# mke2fs /dev/test_vg/test_lv
```

② 创建 reiserfs 文件系统(常用)。

```
# mkreiserfs /dev/test_vg/test_lv
```

③ 使用整个 vg 创建逻辑卷。

```
#vgdisplay test_vg|grep "TotalPE"
#lvcreate -l 45230 test_vg -n test_lv
```

(7) 挂接文件系统。

```
# mkdir  /data/wwwroot
# mount  /dev/test_vg/test_lv  /data/wwwroot
```

(8) 激活 VG。
① 激活指定 VG。

```
vgchange -a y testvg
```

② 激活所有 VG。

```
# vgchange -a y
```

(9) 去激活 VG。

```
# vgchange -a n testvg
```

(10) 移除 VG。

```
# vgchange -a n testvg
# vgremove testvg
```

清华大学出版社

7.5.2　VMware vSphere

1. VMware vSphere 简介

vSphere 是 VNware 公司在 2001 年基于云计算推出的一套企业级虚拟化解决方案、核心组件为 ESXi。经过不断改进，已经实现了虚拟化基础架构、高可用性、集中管理、性能监控等一体化解决方案。它分为许多系列产品，例如服务器虚拟化、桌面虚拟化、应用程序虚拟化等。

vSphere 的基本架构如图 7-10 所示。

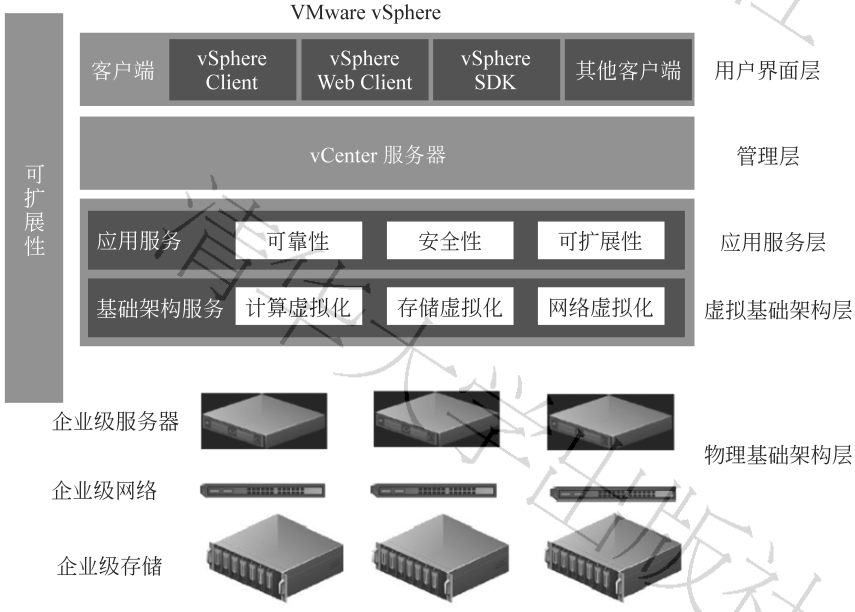

图 7-10　VMware vSphere 的基本架构

vSphere 将应用程序和操作系统从底层硬件分离出来，应用程序可以看到专有资源，服务器则可以作为资源池进行管理。

VMware 公司的 vSphere Essentials 和 Essentials Plus 套件专为工作负载不足 20 台服务器的环境而设计，结合使用 vSphere Essentials Plus 与 vSphere Storage Appliance 软件，无须共享存储硬件即可实现业务连续性。

vSphere Essentials 和 Essentials Plus 专门为刚开始体验虚拟化的小型组织而设计。两个版本都提供最多 3 台服务器主机的虚拟化和集中化管理。vSphere Essentials 可以整合服务器以充分利用硬件。Essentials Plus 添加了 vSphere Data Recovery 等功能，可以实现数据和虚拟机的无代理备份。它还包括业务连续性功能，如 vSphere High Availability（用于在检测到服务器故障时自动重启应用程序）和 vSphere vMotion（可完全消除用于服务器维护的计划内停机）。由此可以建立一个始终可用的 IT 环境，此环境更经济高效、恢复能力更强，并且能更好地响应业务需求。

VMware 公司的 vSphere Essentials 和 Essentials Plus 包括以下主要功能和组件。

（1）VMware ESXi 虚拟化管理程序体系结构提供强健的、经过生产验证的高性能虚拟化层，允许多个虚拟机共享硬件资源，性能可以达到甚至在某些情况下超过本机吞吐量。

（2）VMware vCenter Server for Essentials 通过内置的物理机到虚拟机（P2V）转换和使用虚拟机模板进行快速部署，可为所有虚拟机和 vSphere 主机提供集中化管理和性能监控。

(3) vSphere 虚拟对称多处理(SMP)能使用拥有多达 4 个虚拟 CPU 的超强虚拟机。

(4) vSphere vStorage Virtual Machine File System(VMFS)允许虚拟机访问共享存储设备(光纤通道、iSCSI 等),而且是其他 vSphere 组件(如 Storage vMotion)的关键促成技术。

(5) vSphere vStorage Thin Provisioning 提供共享存储容量的动态分配,允许 IT 部门实施分层存储战略,同时将存储开支削减多达 50%。

(6) vSphere vStorage API 可提供与受支持的第三方数据保护的集成。

(7) vCenter Update Manager 可自动跟踪、修补和更新 vSphere 主机以及 VMware 虚拟机中运行的应用程序和操作系统。

(8) vCenter Converter 允许 IT 管理员将物理服务器和第三方虚拟机快速转换为 VMware 虚拟机。

(9) vSphere VMsafe API 支持使用与虚拟化层协同工作的安全产品,从而为虚拟机提供甚至比物理服务器级别更高的安全性。

(10) 硬件兼容性可兼容最广泛的 32 位和 64 位服务器和操作系统、存储和网络设备以及企业管理工具。

此外,VMware vSphere Essentials Plus 还包括为实现始终可用的 IT 而提供的以下业务连续性功能和组件。

(11) vSphere vMotion 支持在不中断用户使用和不丢失服务的情况下在服务器间实时迁移虚拟机,从而无须为服务器维护安排应用程序停机。

(12) vSphere High Availability 可在硬件或操作系统发生故障的情况下在几分钟内自动重新启动所有应用程序,实现经济高效的高可用性。

(13) vSphere Data Recovery 可为小型环境中的虚拟机提供简单、经济高效、无代理的备份和恢复。

vSphere 服务器虚拟化产品主要有两块,一块为 ESX SERVER,一块为 ESXi SERVER。ESX 和 ESXi 主要的区别在于是否有 Server Console,ESXi 相对于内核比较精简,所以性能会比 ESX 出色,但是 ESX SERVER 的优势主要体现在安全方面以及排错也会相对于简单些。

2. 存储设备的选择

在虚拟化项目中,推荐采用存储设备而不是服务器本地硬盘。在配置共享的存储设备时,只有虚拟机保存在存储时,才能快速实现并使用高可用(high availibility,HA)、容错(fault tolerance,FT)、计划内迁移(vMotion)等技术。在使用 VMware vSphere 实施虚拟化项目时,一般作法是将 VMware ESXi 安装在服务器的本地硬盘上,该硬盘可以是一个固态盘(约 10GB),也可以是一个 SD 卡(约 8GB),甚至可以是 1GB 的优盘,如果服务器没有配置本地硬盘,也可以从存储上为服务器划分 8～16GB 的分区用于启动。

在选择存储设备时,要考虑整个虚拟系统中需要用到的存储容量、磁盘性能、接口数量、接口的带宽。对于容量来说,整个存储设计的容量必须是实际使用容量的 2 倍以上,例如,整个数据中心已经使用了 1TB 的磁盘空间(所有已用空间加到一起),则在设计存储时,要至少设计 2TB 的存储空间(是配置 RAID 之后而不是没有配置 RAID、所有磁盘相加的空间)。

在存储设计中另外一个重要的参数是 IOPS(input/output operations per second,每秒进行读写操作的次数),多用于数据库等场合,衡量随机访问的性能,存储端的 IOPS 性能和主机端的 I/O 是不同的,IOPS 是指存储每秒可接受多少次主机发出的访问,主机的一次 I/O 需要多次访问存储才可以完成。例如,主机写入一个最小的数据块,也要经过发送写入请求、写入数据、收到写入确认这 3 个步骤,

也就是 3 个存储端访问，每个磁盘系统的 IOPS 是有上限的，如果设计的存储系统，实际的 IOPS 超过了磁盘组的上限，则系统反应会变慢，影响系统的性能。简单来说，15000RPM 的磁盘的 IOPS 是 150，10000RPM 的磁盘的 IOPS 是 100，普通的 SATA 硬盘的 IOPS 是 70～80。一般情况下，在做桌面虚拟化时，每个虚拟机的 IOPS 可以设计为 3～5 个；普通的虚拟服务器 IOPS 可以视实际情况规划为 15～30 个，当设计一个同时运行 100 个虚拟机的系统时，IOPS 则至少要规划为 2000 个，如果采用 10 000RPM 的 SAS 磁盘，则至少需要 20 个磁盘，当然这只是简单的测算，在真正实施时需要考虑多方面的因素。

在规划存储时，还要考虑存储的接口数量及接口的速度，通常来说，在规划一个具有 4 主机、1 个存储的系统中，采用具有 2 个接口器、4 个 SAS 接口的存储服务器是比较合适的，如果有更多的主机，或者主机需要冗余的接口，则可以考虑配 FC 接口的存储，并采用光纤交换机连接存储与服务器。

3. 安装和配置

ESXi 对主机、网络有一定要求，如果不能达到，安装难以进行。

（1）主机的基本配置如下：

服务器所用的 CPU 为 64 位双核以上的 x86 CPU，如果存在多个 ESXi，应当选择同一供应商；支持开启硬件虚拟化功能。至少 4GB 物理内存，需要一个或多个千兆以太网控制器。

（2）网络配置如下：

2 个用于业务网络的网络接口卡（光纤接口）；2 个用于管理网络的网络接口卡（电口）[①]；2 个用于 vMotionnic 的功能的网络接口卡（电口）；2 个用于 iSCSI 网络的网络接口卡（2 电口）。

由于 ESXi 安装时会将整张硬盘覆盖，建议通过不用的优盘或硬盘来安装。整个安装和配置过程较为复杂、费时。如果按最简单的要求来安装，大致如下。

开机插入 ESXi 安装盘，按交互式流程完成安装。安装时系统会自动检查可用存储设备，可自行在该界面选择安装的磁盘位置。

安装重启之后，由于硬盘中已经有了 ESXi 系统，ESXi 服务器启动后是全文本界面。但实际操作时很少在服务器上进行操作，相关的操作如创建管理虚拟服务器等，都可以在浏览器/vSphere Client 上进行（6.5 之后官方开始推荐直接浏览器界面配置）。不过在此之前，需要对 ESXi 的网络进行一些配置。vSphere Client 对其的控制也是通过网络进行的，必须事先为 ESXi 配置好访问 IP。

设置网络配置的界面选项如下。

（1）Configure Password：配置 root 密码。

（2）Configure Management Network：配置网络。

（3）Restart Management Network：重启网络。

（4）Test Management Network：使用 ping 测试网络。

（5）Network Restore Options：还原配置。

（6）Troubleshooting Options：故障排查选项。

（7）View System Logs：查看系统日志。

（8）Reset System Configuration ESXi：出厂设置。

根据实际环境正确设置后，把 PC 与 ESXi 服务器设置在同一网段就能通过 IP 地址在浏览器中访问服务器。图 7-11 所示为 ESXi 服务器登录界面，图 7-12 所示为 ESXi 服务器主界面。

---

① 电口是指 RJ-45 等各类双绞线接口的总称。

清华大学出版社

图 7-11　ESXi 服务器登录界面

图 7-12　ESXi 服务器主界面

vSphere 的物理架构如图 7-13 所示。ESXi 服务器就安装在虚拟计算机上，访问 ESXi 服务器的是图中笔记本计算机，也就是通过它管理整个虚拟化平台。

7.5.3　Openfile

在第 5 章曾经介绍过 Openfile，并用它进行 NAS 实验。实际上，Openfile 也是虚拟存储开源软件，其软件接口是基于使用开放源码的第三方软件来提供虚拟存储功能。

Openfiler 所提供的强大的虚拟存储功能特性，使其在以动态、灵活、可伸缩为特质的云计算环境中，成为了一个非常有力的云端存储解决方案之一。同时，它也可以作为实验环境下的一种高效的存储模拟解决方案之一，具有很高的成本节约优势和实用价值。可以通过基于 Openfiler 的虚拟镜像创建和配置，创建基于自己云端环境需求的自定制虚拟存储解决方案。

当 Openfiler 系统安装并重启完成后，就可以进行镜像系统配置。对系统所有后续的配置过程，都是以 Web 方式配置完成的。这种全部基于 Web 的系统配置方式，使得配置过程变得更加简易，同时也

清华大学出版社

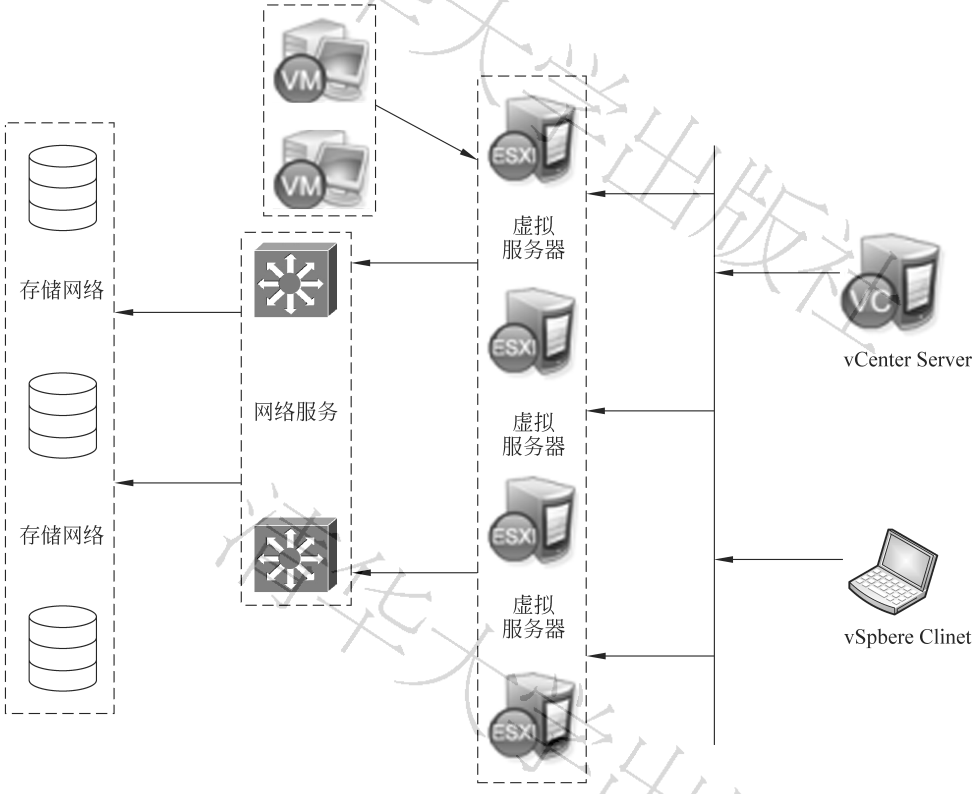

图 7-13　vSphere 物理架构

带来了良好的用户体验。首先打开浏览器，输入地址 https://IP:446。注意，此处的 IP 为在系统安装时所配置的固定 IP 地址。然后使用系统初始默认的用户名和密码进行登录。用户名为 openfiler，密码为 password。系统初始的用户名和密码可以在第一次登入系统后进行重新设置。如图 7-14 所示。

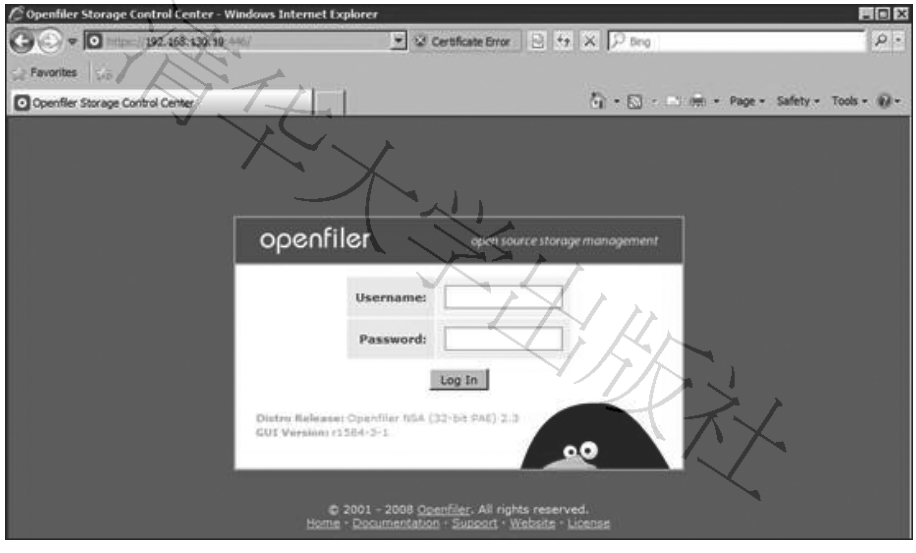

图 7-14　登录系统管理界面

录入系统后，可以看到当前系统的各种初始化配置信息。

在 System 菜单下，可以检查当前系统的 IP 等网络信息设置情况。如果想重新设置网络信息，可以单击 Configure 功能连接对系统网络进行重新配置。

通过 System|Network Access Configuration 菜单，配置允许访问 Openfiler 系统的安全访问控制列表。只有加入到 Openfiler 的网络访问控制列表中的网络或者主机地址，才允许访问 Openfiler 系统所提供的虚拟存储服务。在网络访问控制列表中，既可以配置网段地址也可以配置单个主机地址。例如可填入 192.168.130.0(假设)网段，设置类型为 Share 方式，添加完成后单击 Update 按钮，完成系统配置更新，如图 7-15 所示。

图 7-15　配置系统安全访问控制列表

然后单击 Network ACL 标签，将访问控制从默认禁止访问更新为允许访问 Allow，最后单击 Update 按钮，更新系统配置。

当配置好系统的安全访问控制后，接下来配置系统磁盘，实现虚拟存储服务。在 Openfiler 中，有 3 类存储概念：Block Device、Physical Volume 和 Volume Group。其中，Block Device 表示的是实际的物理磁盘；Physical Volume 表示的是物理磁盘的分区，它是组成 Volume Group 的单元；Volume Group 则是由一个或多个物理磁盘分区(Physical Volume)组成，它又是组成 Logical Volume 的单元。

首先选中 Volumes|Volumessection|Block Devices 菜单项，然后系统会显示当前所挂载的硬盘信息。然后单击磁盘/dev/sda，系统会显示当前磁盘的详细分区信息。

现需要创建一个新的分区。首先在 Create a partition in /dev/sda 处设置 Partition Type 属性的值为 Physical volume，在 Ending cylinder 属性处选择默认值，从而设置当前所有的剩余空间划为一个分区，最后单击 Create 按钮，如图 7-16 所示。

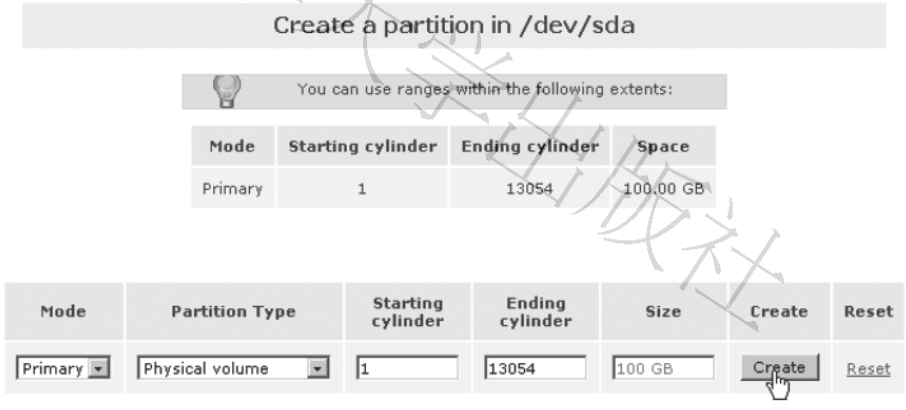

图 7-16　创建新分区

通过 Volumes|Volumes|Volumes section|Volume Groups 菜单创建一个卷组。在区域 Create a new volume group 处填写卷组 volume group 的名称为 volume_group_iscsi，同时勾选刚刚在上一步所创建的物理卷/dev/sda4，最后单击 Add volume group 按钮。

通过 Volumes|Volumes section|Add Volume 菜单，在刚刚创建好的卷组 volume_group_iscsi 中新创建一个 iSCSI 卷。注意在属性 Filesystem/Volume type 中设置属性值为 iSCSI。

通过 Volumes|Volumes section|Manage Volumes 菜单，可以看到刚刚创建的 iSCSI 卷信息。

当 iSCSI 卷创建完成后，接下来是开启 iSCSI target server 系统服务。通过 Services|Services section|Manage Services 菜单，将系统服务列表中的 iSCSI target server 设置为 Enabled 状态，从而使得系统能够对外提供基于 iSCSI 协议的虚拟存储服务。

接下来是添加一个 iSCSI Target。首先通过 Volumes|Volumes section|iSCSI Targets 菜单。再选中 Target Configuration|Add new iSCSI Target|Target IQN。最后单击 Add 按钮，从而添加了一个 iSCSI Target。注意，此处的 Target IQN 信息在后面的对 ESX 服务器的存储配置过程中会用到。如图 7-17 所示。

图 7-17　添加 iSCSI Target

选中 LUN Mapping 子菜单，保持其余默认选项，单击 Map 按钮。从而实现从 LUN 到刚刚配置好的 iSCSI Target 之间的映射。如图 7-18 所示。

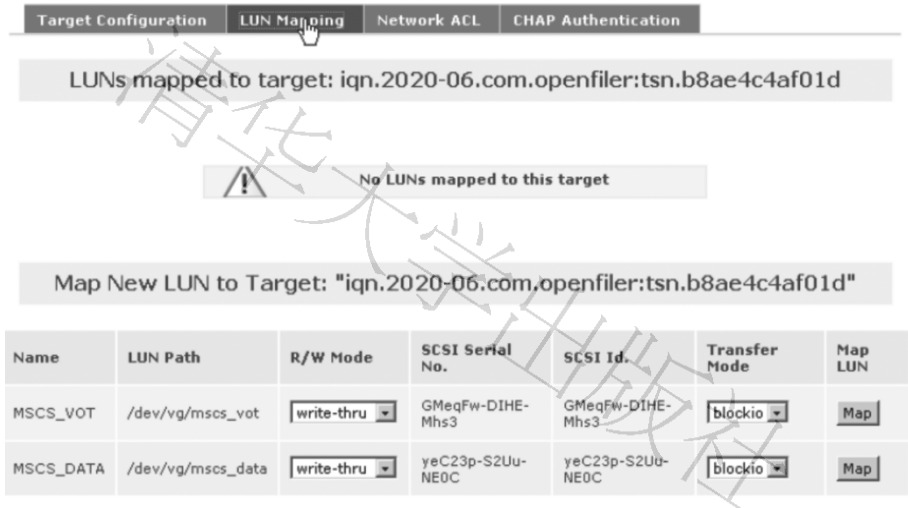

图 7-18　配置 LUN Mapping

至此，在 Openfiler 系统上对 iSCSI 虚拟存储的配置过程就完成了。

上面对虚拟存储及开源虚拟存储软件 Openfiler 进行了介绍，同时通过基于 Openfiler 软件的 iSCSI

存储服务的创建和配置，充分享受虚拟存储所具有的独特优势。

习题 7

一、选择题

1. 虚拟存储的原动力包括(　　)。
   A. 空间资源的整合　　B. 统一数据管理
   C. 标准化接入　　D. 使数据自由流动
2. 从实现位置来看，虚拟存储技术可分为(　　)。
   A. 基于主机的虚拟
   B. 基于网络的虚拟
   C. 基于存储设备、存储子系统的虚拟
   D. 虚拟带外
3. 对于基于主机的虚拟技术，下列说法正确的是(　　)。
   A. 使服务器的存储空间可以跨越多个异构的 RAID
   B. 需占用主机资源，并且导致主机升级、维护和扩展复杂
   C. 存在操作系统和应用的兼容性问题
   D. 只能通过操作系统下的逻辑卷进行，别无他法
4. 对于基于网络的虚拟技术，下列说法正确的是(　　)。
   A. 必须在主机端安装代理程序，才能够实现
   B. 能够支持异构主机、异构存储设备
   C. 使不同存储设备的数据管理功能统一
   D. 基于网络的虚拟技术，就是带外虚拟技术
5. 关于带内、带外虚拟技术，正确的是(　　)。
   A. 带外虚拟设备发生故障，整个系统将不会中断
   B. 带内虚拟设备发生故障，整个系统将不会中断
   C. 带外虚拟必须在服务器端安装代理程序
   D. 带内虚拟必须在服务器端安装代理程序
6. 存储虚拟技术可以实现存储和管理的好处是(　　)。(多选)
   A. 提升存储管理员的生产能力
   B. 提升存储容量利用率，提高资源利用率
   C. 提升应用的可用性
   D. 以上都不是
7. 同时利用服务器和存储虚拟的优势是(　　)。
   A. 可以降低每台服务器的专用存储容量
   B. 可以消除额外存储管理员的需求
   C. 可以降低对额外网络交换机的需求
   D. 可以降低对日常磁带备份的需求
8. 以下选项中，(　　)准确描述了虚拟化。
   A. 提供按需计量服务

清华大学出版社

B. 将物理资源抽象化为逻辑资源

C. 共用逻辑资源以提供数据完整性

D. 支持跨数据中心的分散式管理

9. 关于虚拟技术的描述，不正确的是(　　)。

A. 虚拟是指计算机元件在虚拟的基础上而不是真实的基础上运行

B. 虚拟技术可以扩展硬件的容量，简化软件的重新配置过程

C. 虚拟技术不能将多个物理服务器虚拟成一个服务器

D. CPU 的虚拟技术可以单 CPU 模拟多 CPU 运行，允许一个平台同时运行多个操作系统

10. 虚拟化资源指一些可以实现一定操作具有一定功能，但其本身是(　　)的资源，如计算池、存储池和网络池、数据库资源等，通过软件技术来实现相关的虚拟化功能包括虚拟环境、虚拟系统、虚拟平台。

A. 虚拟　　B. 真实　　C. 物理　　D. 实体

11. 关于 VMware vSphere 的核心组件 VMware ESXi 虚拟化技术的说法错误的是(　　)。

A. VMware ESXi 是一款可以独立安装和运行在裸机上的系统

B. VMware ESXi 与 VMware Workstation 功能类似，都是宿主模型虚拟化

C. 在 ESXi 安装好以后，可以通过 vSphere Client 远程连接控制，在 ESXi 服务器上创建多个虚拟机(VM)

D. ESXi 可以从内核级支持虚拟硬件

二、简答题

1. 如图 7-19 所示，假设某虚拟存储系统的存储监测模块能够获得存储系统的监测数据，当得知存储系统的存储空间不足时，就需要为存储系统分配虚拟存储资源以扩大其存储空间。请讨论该虚拟存储系统的虚拟硬盘资源分配的流程，并画出处理流程图。

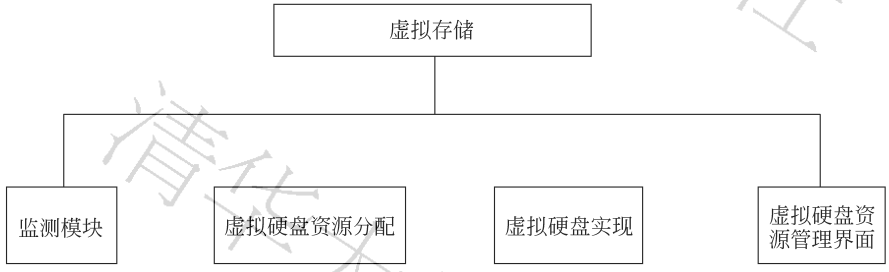

图 7-19　第 1 题图

2. 简要说明虚拟服务器、虚拟存储和虚拟网络都有哪些实现方式？

3. 虚拟存储的实现方式有哪 3 种？原理分别是什么？各自的优缺点是什么？

4. 虚拟架构的分类，根据在整个系统位置的不同可以分为哪几类？

5. 简要描述物理体系结构与虚拟体系结构的差异。

三、实验题

1. LVM 基础实验。

【实验目的】

(1) 了解 LVM。

清华大学出版社

(2) 能进行 LVM 部署和简单操作。

【实验环境】

一台 PC,要求配有两块硬盘或将一块硬盘划分为两个分区(如/dev/sdc(200G) & /dev/sdd(300G))作为逻辑卷。然后对逻辑卷并进行扩容、缩减操作等操作。

【实验内容】

按以下要求写出操作命令,贴出截图。

(1) 创建 PV(physical volume)。

(2) 创建 vg(volume group)。

(3) 创建 lv(logical volume)。

(4) 创建文件系统(filesystem)。

(5) 建立 LVM 线性卷,镜像卷,带区卷,快照。

(6) 扩容 20GB、缩减 20GB。

(7) 恢复快照。

(8) LVM 线性卷转换成镜像卷。

2. LVM 综合实验。

【实验目的】

(1) 了解 LVM。

(2) 掌握 mdadm 的使用方法。

【实验环境】

一台安装了 Ubuntu Linux 的 PC。

【实验内容】

下面进行软 RAID 实验,主要完成 4 项任务。

(1) 将 sdb、sdc、sdd3 块硬盘组成 RAID 5 模式。

(2) 建立 LVM。

(3) 模拟故障,sdc 出故障,删除该硬盘,再重新添加硬盘,恢复 RAID 5。

(4) 增加 LVM 容量。

操作过程如下。

1) 组成 RAID 5 模式

(1) 分别格式化 3 块硬盘(命令:fdisk /dev/硬盘)。

通过虚拟机建立 3 块硬盘作为阵列卡使用,创建后,分别格式化 sdb、sdc、sdd 这 3 块盘。

(2) 查看分区情况(命令:fdisk -l)。

(3) 建立 RAID 5(命令:mdadm,带多个参数) 例如:

```
mdadm --create /dev/md0 --level=5 --raid-device=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

表示创建 RAID 5,/dev/md0 阵列设备名,level=5 阵列模式 RAID 5,raid-device=3raid 有 3 块硬盘。

(4) 查看数据同步情况(命令:cat /proc/mdstat)。

(5) 查看系统日志(命令:tail /var/log/messages)。

(6) 建立 RAID 5 的配置文件。例如:

```
echo device /dev/sdb1 /dev/sdc1 /dev/sdd1 &gt; /etc/mdadm.conf
mdadm --detail --scan >>/etc/mdadm.conf
cat /etc/mdadm.conf
```

(7) 启动和停止阵列。举例如下。

停止阵列：

```
mdadm -S /dev/md0
```

启动阵列：

```
mdadm -As /dev/md0
```

2) 建立 LVM

LVM 创建的顺序是 PV→VG→LV。

(1) 建立 PV(命令 pvcreate /dev/md0)。首先下载 lvm2,然后把物理盘数据写入/dev/md0,要确保写入成功。

(2) 建立 VG(命令 vgcreate lvm1 /dev/md0)。

(3) 查看 VG(命令 vgdisplay)。

(4) 建立 LV(命令 lvcreate)。例如：

```
lvcreate -L 500m -n web1 lvm1        //建立名为 web1,大小为 500MB 的 LV
lvcreate -L 500m -n web2 lvm1        //建立名为 web2,大小为 500MB 的 LV
```

(5) 格式化 web1、web2。例如：

```
mke2fs -j /dev/lvm1/web1            //格式化 web1
```

格式化逻辑盘 web1,格式化 web2 的过程相同。

(6) 建立目录 web1、web2,并挂载到 web1、web2 下。例如：

```
mkdir /web1
mkdir /web2
mount /dev/lvm1/web1 /web1          //挂载 web1
mount /dev/lvm1/web2 /web2          //挂载 web2
```

编辑/etc/fstab,让系统启动时自动挂载 vi /etc/fstab。

建立目录 web1、web2,并挂载到 web1、web2 下。

通过 vim 编辑/etc/fstab 让系统启动时自动挂载 web1、web2。

(7) 查看/web1 的容量。例如：

```
df -h /web1
pvdisplay
```

3) 模拟故障

(1) 标记/dev/sdc1 为故障盘。例如：

```
mdadm /dev/md0 -f /dev/sdc1          //标记/dev/sdc1 为故障盘
```

清华大学出版社

查看 more /proc/mdstat 请写出执行过程，并略加说明。

(2) 移除故障盘。例如：

```
mdadm /dev/md0 -r /dev/sdc1          //移除故障盘
```

(3) 查看阵列情况。例如：

```
more /proc/mdstat                    //查看阵列情况
```

(4) 查看 PV 情况。例如：

```
pvdisplay /dev/md0                   //查看 PV 情况
```

(5) 重新格式化 sdc，重新添加进阵列。例如：

```
fdisk /dev/sdc                       //重新格式化 sdc
mdadm /dev/md0 -a /dev/sdc1          //重新添加进阵列
more /proc/mdstat                    //开始同步数据
pvdisplay
```

```
vgdisplay lvm1
df -h /web1
```

再次使用 pvdisplay、vgdisplay、df 查看信息与之前结果是否完全一致，以此证明移除添加同一块硬盘对 RAID 5 的各项数据没有影响。

4) 增加 LVM 容量

(1) 增加。例如：

```
lvextend -L +50M /dev/lvm1/web1          //增加 web1 50MB
```

成功添加 50MB，容量变为 552MB，确实比初始设置的 500MB 增加了 50MB。

(2) 刷新。例如：

```
resize2fs /dev/lvm1/web1
```

(3) 查看。

```
df -h /web1
```

5) 实验思考

LVM 卷管理系统与虚拟化存储在存储资源管理上有着截然不同的效果，试就本例讨论其特点。

3. VSphere 平台虚拟系统综合实验。

【实验目的】

(1) 了解 VSphere。

(2) 掌握虚拟平台的搭建方法。

清华大学出版社

【实验环境】

(1) 建立一台 Windows 主机(IP:192.168.12.10),要求具有不少于 16GB 的运行内存,40GB 的硬盘 1 用于独立存储,150GB 的硬盘 2 用于独立存储。

(2) 建立一台 ESXi 主机(IP:192.168.12.2),要求具有不少于 16GB 的运行内存,40GB 的硬盘 1 用于独立存储。

(3) 克隆刚建立的 ESXi 主机,配置 IP:192.168.12.3。

(4) 建立一个名为 MEVMNET 的虚拟网络,用于虚拟系统数据传输。

(5) 建立一个名为 MEVMK_NET 的虚拟网络,用于虚拟系统主机集的管理和传输。

【实验内容】

在 VSphere 上搭建一套 VSphere 环境平台。需要完成下面 6 项实验,请画出拓扑图,给出操作截图,并附必要的说明。

(1) 在 Windows 主机上安装 VCenter。

① 安装 VCenter。

② 建立数据中心。

③ 添加 IP:192.168.12.2、192.168.12.3 两台主机。

④ 建立群集,移入上述两台主机。如果直接加入主机会报错,提示容错资源不足。

⑤ 为 192.168.12.2、192.168.12.3 两台主机添加 NFS 共享存储和私有存储。

(2) 在 Windows 主机上建立 NFS,NFS 是 Linux 系统上的文件共享服务器。建立后可以为虚拟机平台提供共享存储,提供主机集。

① 为该主机添加一块 150GB 的虚拟磁盘,即为共享盘。

② 选中"管理工具"|"服务器管理",通过左边的"角色",添加角色并选择文件服务。

③ 在共享盘添加一个名为 NFS 的文件夹并右击,从弹出的快捷菜单中选中"属性"|"NFS 共享"|"管理 NFS 共享"选项,打开 NFS 高级共享。

④ 启用无服务器身份验证,启用未映射用户访问,选择"允许未映射的用户 UNIX 访问"。

⑤ 单击"权限"按钮。添加允许的计算机的 IP,此处为 192.168.12.2 和 192.168.12.3。

(3) 把安装系统的 ISO 文件传递到 NFS。

① 在 Windows 主机再添加一块网卡,IP 地址配置为所在主机网络的 IP 地址,通过 ping 命令确认从操作物理终端机连接正常。

② 通过 FTP、Windows 共享等方式从用户本地上传文件。使用 Windows 共享方式,把该虚拟机的 NFS 文件夹共享出来。

(4) 在 192.168.12.2 建立一台虚拟机。并安装 Windows 系统,要求具有 8GB 的运行内存和 20GB 的硬盘。

① 在 192.168.12.2 建立一台安装 Windows 系统的虚拟机。

② 将文件储存在 NFS 共享上。

③ 安装 Windows 系统。

(5) 测试主机集的特性。

① 启动上述 Windows 虚拟机。

② 关闭 192.168.12.2 的电源。等待一会儿,查看是否启动了 Windows 虚拟机。

③ 其他测试,请自行设计测试用例。

清华大学出版社

4. 用 Openfiler 配置 iSCSI。

【实验目的】

使用 Openfiler 搭建 iSCSI 网络共享存储。

【实验环境】

一台 PC,要求具有 2GB 的内存和 4 块硬盘。实验可选择在实体机或虚拟机上进行。

设定 Openfiler 服务器 IP 地址为 192.168.1.254、掩码为 255.255.255.0。

【实验内容】

1) 安装 Openfiler

(1) 初始化硬盘并删除所有数据。将 Openfiler 系统安装到第 1 块硬盘 sda 上。

(2) 设置网卡地址、掩码。

(3) 设置网关、DNS。

(4) 设置 root 账户密码。

(5) 系统重启。

2) 创建物理卷(PV)和卷组(VG)

(1) 在一台连接正常的 Openfiler 机(如果是虚拟机则将主机网卡设置为同一网段)通过浏览器 https://192.168.1.254:446/访问 Openfiler。

(2) 在登录界面输入 Openfiler 系统默认的初始密码登录系统。

(3) 创建 PV。

① 将第 2 和第 3 块硬盘(/dev/sdb、/dev/sdc)组成新的 VG。

② 选择/dev/sdb,此时会进入另一个页面,在这个页面最下方有一个 Create a partition in /dev/sdb。

③ 在 Mode 的下拉列表框中选择 Primary,Partition Type 中选择 Physical Volume。由于要使用整个扇区,因此直接单击 Create 按钮即可。

④ 观察这个分区是否已经创建成 PV 的成员。

⑤ 重回 Block Devices 页面,重复上面的步骤可以将/dev/sdc 也创建成另一个 PV。

(4) 将 PV 组合成 VG。

① 将 PV 创建好之后,就可以利用这些 PV 创建一个 VG。选择右侧的 Manage Volumes。

② 为 VG 命名,将两个 PV 选中,并单击 Add volume group 按钮。

③ 通过 Volume Group Management 中观察列出的 VG。

3) 创建 iSCSI 的分区和连接,划分出具有 iSCSI 连接能力的 LUN(逻辑分区)

(1) 开启 iSCSI Target Server 功能。Services 选项卡下,将原来 Disabled 的 iSCSI target server 的 Enable 按钮单击。

(2) 创建 iSCSI 逻辑分区 LUN。

① 选择 Volumes 中的 Manage Volumes。

② 选择 Volume Groups,可以看到目前已有一个刚创建好的 VG,上面还没有任何 LUN。

③ 单击 Add Volume 按钮,就会进入加入新 LV 的画面。最下面有一个创建 LV 的地方,输入 LV 的名称(如 LUN01),输入描述、大小,并且在 Filesystem/volume type 下拉列表框中选择 block (iSCSI, FC,etc)。之后单击 Create 按钮。至此 iSCSI 的 LUN 已经创建完成。

4) 配置 Openfiler 网络

(1) 进入 Openfiler 中的 System,并且直接拉到页面的下方,在 Network Access Configuration 的

清华大学出版社

地方输入这个网络访问的名称，如 VM。

(2) 输入主机的 IP 段。注意不可以输入单一主机的 IP，这样会都无法访问。当输入 192.168.1.0 时，表示从 192.168.1.1 一直到 192.168.1.254 都能访问。在 Netmask 中选择 255.255.255.0，并且在 Type 下拉列表框中选择 Share，之后即可以单击 Update 按钮。

5) 配置 Openfiler iSCSI 目标

(1) 进入 Openfiler 的 Volumes 中，在右侧选择 iSCSI Targets。会看到有一个 Target IQN 的字段，这个字段称为 iSCSI 合格证(iSCSI Qualified Number)，是每一个 iSCSI 唯一的编号，也是在网络上辨认 iSCSI 设备的唯一编号。这个号码由系统产生，可以不需要更动。在此单击 Add 按钮。

(2) 在新增之后，可以在方块下方看到完整的 IQN 参数，这个参数暂时不用更动，使用默认值即可。

(3) 在 IQN 创建之后，将 LUN 映射到这个 IQN 上。此处 Openfiler 一般已做好。只要进入 LUN Mapping 的选项卡，选择刚才创建的 LUN，并且单击 Map 按钮即可。

(4) 映射完成后，确认已经正常，随时可以再 Unmap。

(5) 选择 Network ACL 选项卡，将 Access 改成 Allow，单击 Update 按钮。

6) 在 Windows 系统中测试 iSCSI 连接

(1) 调出“控制面板”，选中“管理工具”，在弹出的“管理工具”窗口中选中“服务中启动 Microsoft iSCSI Initiator Service”。

(2) 运行 iSCSI 发起程序。通过“发现”选项卡、“发现门户”按钮新增 iSCSI Target，指定在 Openfiler 的设置的 IP(192.168.1.254)。

(3) 在“目标”选项卡中，观察有否出现对应的 LUN、什么状态、可否使用？

(4) 通过“连接”按钮，将这个 LUN 加入计算机中。

(5) 观察状态，LUN 是否已经连接上？有什么新的改变？

(6) 在 Openfiler 的网页上，单击 Status，单击右侧 iSCSI Target，观察连接情况。

(7) 在 Windows 系统下，打开“磁盘管理”，观察是否新增了一块磁盘，观察大小、状态等情况。

(8) 性能测试。利用 Iometer 测试工具对 Openfiler 的存储性能、传输带宽及反应能力、网络吞吐量、硬件性能等进行测试，对测试结果进行分析。

5. 数据中心存储服务器虚拟化存储设计实践。

【实验目的】

(1) 通过自己动手设计搭建一独立小型存储服务器，采用多种存储技术软硬件搭配，实现对数据的存储、管理与使用。

(2) 掌握 RAID 技术以及存储虚拟化技术并对存储服务器在安全可靠性方面进行改进。

【实验环境】

需要的实验设备包括计算机、磁盘、RAID 卡、虚拟化存储磁盘管理软件、服务器端。整个环境根据设计方案自行写出。

【实验内容】

(1) 采用不同的 RAID，测试其对存储服务器性能的影响。

(2) 在服务器客户端中安装使用存储虚拟化管理软件，掌握存储虚拟化在存储空间的分配管理以及客户的使用权限方面的解决方案。

(3) 对本存储服务器进行性能改进，如数据读写保护权限的管理、磁盘防插拔方案。

具体步骤如下。

清华大学出版社

(1) 完成对存储服务器的搭建以及管理软件的安装。

(2) 使用虚拟化存储方式,多台计算机与服务器进行网络连接并对服务器发出存储空间申请,服务器对其进行存储空间以及权限的分配与管理。

(3) 以多种方式搭建 RAID,并在服务器端与客户计算机端上进行数据的传输存储与数据备份速度的测试。

(4) 在存储服务器端观察服务器对客户机逻辑存储空间与物理存储空间的分配情况,进行存储虚拟化下服务器负荷能力测试。

(5) 在客户端计算机上进行应用测试,主要包括数据传输速度、数据访问与改动权限测试。

(6) 对存储服务器在数据的可靠安全性方面进行测试,主要进行数据备份方式、模拟数据灾难并进行恢复、入侵拦截、防恶意插拔等安全测试。

(7) 通过以上的实验,提出对存储服务器的改进建议,尤其在数据的可靠安全性方面进行改进并再次进行测试。

* 要求将本实验的过程录制视频。

清华大学出版社