DATA CENTER FABRIC

Introduction to Backup and Recovery in a Virtual Iron Environment

Discusses backup and recovery of Virtual Servers (VSs) in an environment utilizing server virtualization technology from Virtual Iron®. Provides a high-level overview of Virtual Iron's server virtualization technology and the methods that can be employed for data backup and recovery.

BROCADE

CONTENTS

INTRODUCTION

If you were to ask yourself, "What is the significance of this topic?" the answer would be this: virtualized servers, while behaving like physical servers from an application perspective, have unique characteristics when it comes to storage and backup and recovery. For that reason, the backup and recovery techniques that a business currently uses in a physical server environment may not be practical (or even possible) in a virtualized environment. In addition, there may be more effective ways to perform backup and recovery in a virtualized environment. This paper will identify these differences and describe the available backup and recovery techniques when Virtual Iron® is used as the virtualization technology.

Server virtualization is gaining wide acceptance in many IT shops today, because it allows the data center to be more responsive to the needs of the business while at the same time reducing overall operating costs. Virtual Iron has chosen to work with Brocade® to develop backup and recovering techniques and best practices because of Brocade's vast storage management and backup and recovery expertise. Brocade and Virtual Iron will continue to enhance and develop these techniques as Virtual Iron's server virtualization technology evolves.

VIRTUAL IRON TECHNOLOGY

Virtual Iron (VI) is an enterprise-class server virtualization technology that fully leverages the virtualization capabilities of today's modern x86 processor chips as well as industry standards. Virtual Iron's technology can be used to address many of the key server management challenges in a data center including the following:

- Efficient utilization of server resources
- Rapid provisioning and deployment
- High availability and recovery
- Workload management
- Cost containment

VI allows a properly configured physical server to be logically partitioned to run multiple instances of Windows- or Linux-based "guest" operating systems. These guests (also called "Virtual Machines" or "VMs") each run their own operating system image and are allocated a specific number of CPUs, which are shared among the other VMs, and a fixed amount of dedicated memory. In addition, each VM has access to one or more virtual Ethernet network interfaces and storage (disks).

The entire virtual environment is managed by Virtual Iron's *Virtualization Manager™*. Virtualization Manager is used to perform all configuration and management functions in the virtual environment.

The Virtual Iron architecture consists of the following key components:

- **Managed Node**. An industry-standard server utilizing Intel or AMD processors with hardware-assist virtualization support.
- **Virtual Data Center**. A logical grouping of managed nodes.
- **Virtual machine/virtual server**. An instance of a Windows- or Linux-based operating system running on a managed node. Sometimes referred to as a "guest" operating system.
- **Virtualization Manager**. Virtual Iron management console.
- **Virtualization Manager Server**. Server host on which the Virtualization Manager application runs.
- **Virtualization Manager Client**. Host on which the Virtualization Manager GUI is executed.

- **Hypervisor**. The first software loaded when a managed node boots. Virtual Iron's hypervisor is derived from the Xen™ open source project; it manages all of the node's hardware and allocates resources to the virtual servers. Requires no disk resources; uses only a RAM-based disk for execution.
- **Service Partition**. Once the hypervisor has been loaded a virtual service partition is created on the managed node, called the service partition. This is a special "privileged" partition that has access to all the hardware and also controls the hypervisor. The operating system that runs in this partition is a very "lean" version of the Novell SUSE Linux Enterprise Server (SLES) 10 kernel. It uses a RAM disk for operation, and therefore does not need any physical disk storage. The service partition is controlled by Virtualization Manager via an embedded agent. The service partition has no console and is not user-accessible.
- **Ethernet networks**. The managed node may contain one or more Gigabit Ethernet Network Interface Cards (NICs). Virtual Gigabit NICs are assigned to each virtual server based on the server configuration.
- **Disk storage**. The managed node connects to: directed-attached, iSCSI, or Fibre Channel storage. The service partition and Virtualization Manager are used to partition the physical storage and assign one or more disk Logical Units (LUNs) to the virtual servers. From a virtual server perspective the disk LUNs appear to be attached to a SCSI adapter.

For more detailed information on the Virtual Iron architecture, visit:
http://www.virtualiron.com/products/index.cfm

VIRTUAL IRON STORAGE CONFIGURATION

Physical disk storage must first be made accessible to the managed node before it can be provisioned to the virtual servers. The storage provisioning process for the managed node is generally the same as the provisioning process for any other physical server— all the same provisioning steps are required (storage LUN configuration, zoning, LUN masking, discovery, and so on).

There are three options for connecting disk LUNs to a managed node:

1. **Fibre Channel (FC) Storage Area Network (SAN).** In this case, one or more FC Host Bus Adapters (HBAs) must be installed in the managed node. The managed node can then connect to any supported storage subsystem accessible in the FC fabric. Note that all the standard Fibre Channel storage provisioning tasks must be performed to make the desired LUNs available to the managed node.
2. **iSCSI.** In this case the managed node accesses Ethernet-attached storage via an iSCSI-capable NIC. Note that all the normal iSCSI storage provisioning tasks must be performed to make the desired LUNs available to the managed node.
3. **Local SATA, SAS or parallel SCSI disks.** In this case the disk is accessed via the appropriate SCSI adapter installed in the managed node, which could be internal storage or external storage.

LUNs presented to the managed node can be physical disks (for example, locally attached SATA, SAS, or SCSI disks) or abstracted LUNs from a storage subsystem controller (for example, a "slice" of group of disks in a RAID-5 configuration). In either case, the service partition on the managed node automatically discovers the LUNs and creates the corresponding SUSE Linux disk device file entries.

The disk devices that are discovered by the service partition Linux operating system are managed by the Linux volume manager. As previously mentioned, Virtualization Manager is the management interface to the service partition. Using Virtualization Manager, disks are organized into *disk groups*. Each disk can belong to only one disk group. Disks are then made accessible to virtual servers in one of two ways: as a *logical disk* or as a *raw disk*. These two storage access methods are discussed in the following sections.

Logical Disks

A logical disk represents a specific quantity of capacity (for example, 10 GB) in a disk group. When a logical disk is created via the Virtualization Manager, a Linux logical volume is created on the service partition and the logical volume is presented to the virtual server as a SCSI disk drive. Note that there is no file system layered between the logical volumes on the service partition and the disk LUNs on the virtual servers.

Compared to raw disks, logical disks have the following advantages and disadvantages:

Advantages

- Easier to manage and allocate storage resources without SAN changes. For example, in the physical world, if 10 servers each required a 10 GB LUN, the array storage would have to be divided into 10 separate LUNs and then the storage provisioning process performed for each server (zoning and LUN masking). However, with 10 virtual servers, the provisioning of the storage takes place only for the service partition (once instead of 10 times). Then logical disks can be created and assigned to each virtual server.
- Virtualization Manager can be used to:
    - Clone
    - Move
    - Import
    - Export
    - Entire virtual server can be archived, cloned or restored

Disadvantages

- Performance may be impacted due to the overhead of I/O transactions passing through the service partition's logical volume manager. However, because there is no file system used on the service partition, impact on performance is generally negligible.
- Storage subsystem tools cannot be used on individual virtual disks, since the SCSI pass-through commands are not supported.

Raw Disks

As a raw disk, a LUN on the service partition is presented completely and directly to the virtual server. In this case, the service partition's logical volume manager is not used to access the disk.

Compared to logical disks, raw disks have the following advantages and disadvantages:

Advantages

- Potentially better performance than a logical disk (although logical disks perform very well too, since they are not encapsulated within a file system on the service partition).
- Storage subsystem tools (snapshot, replicate, and so on) can be used on individual virtual server disks (LUNs).

Disadvantages

- Virtualization Manager cannot be used to perform useful functions such as snapshot, clone, move, import, and export.
- Virtualization Manager cannot be used to archive, clone, or restore entire virtual servers.

VIRTUAL SERVER BACKUP

Backup and Recovery Overview

Backup and recovery is a challenge in most IT shops. Backup is not a high-profile, revenue-generating technology; and as a result generally receives little attention and support from senior. And yet, protecting a business' information assets is a critical function of the IT organization. As a result of this thinking, backup (and recovery) is often an afterthought with little strategic planning or investment.

Today, there are many backup and recovery challenges, not the least of which is *defining the requirements*. The requirements should drive decisions about the backup and recovery solutions selected and ultimately the cost of providing this service. Recovery requirements are generally defined by two parameters:

- **Recovery Point Objective (RPO).** The RPO is driven by two key parameters:
    - *The number of transactions that can be lost,* defined in terms of actual business application transactions (for example, for a transaction-oriented database) or in time units (for example, 2 hours). This determines how much data/transactions the business deems acceptable to loose
    - *The point in time it must be possible to recover data from*, that is, how far back in time must the data be recoverable. This parameter generally drives the retention settings that are used for the protected data.
- **Recovery Time Objective (RTO).** The RTO defines how quickly the business function must be restored. The RTO generally refers to the resumption of business function, not necessarily the amount of time available to perform data recovery.

More demanding requirements (that is, fewer lost transactions going back farther in time with a shorter recovery time) are more costly to satisfy per unit of storage than less demanding requirements. Therefore, the challenge for businesses is to define the requirements and then develop a strategy for meeting those requirements in a cost-effective manner. Typically, this requires multiple solutions and, because a single technology may not be able to meet the requirements of part of the business or there may be more cost-effective ways to meet those requirements.

Backup and recovery technologies can generally be grouped into the following three solution categories:

1. **Core technology.** Core technology, sometimes called "classic" backup and recovery technology, is what most businesses use today. These solutions typically consist of a "backup server" and "backup clients". Backup clients send their backup data over the network to a backup server and the backup server writes the data to storage. Managing this technology in a large shop is a sizable task. Some of the more popular solutions are: IBM Tivoli Storage Manager (TSM), Symantec NetBackup, EMC Legato NetWorker, CommVault Galaxy, CA ARCserve, and HP Data Protector. These are typical "large shop" products, but there are also a large number of products that cater to smaller environments.
2. **Storage-based technology.** Storage-based technology utilizes functions in the storage subsystem to create additional copies of the data. These functions typically include: "full-copy" data snapshots, "pointer-based" data snapshots, and synchronous and asynchronous replicated copies, typically over long distances.
3. **Specialized technology.** Specialized technology describes those technologies that tend to be newer and leading edge, technologies that have not gained industry-wide acceptance and/or are targeted at a specific application or environment. Examples of these include Continuous Data Protection (CDP) and remote office consolidation products.

Virtualized Server Backup Challenges

A virtualized server environment presents backup and recovery challenges, while at the same time providing additional capabilities that are not available in a non-virtualized environment. As with any backup and recovery design, understanding what recovery events you are designing for is a critical first. For example, are you designing a solution to recover an individual file, a complete drive/file system, an entire database, or a complete server? Then, when you know what events you are designing for, you must understand what the recovery requirements (RPO and RTO) are for each of those events.

The key challenges to be aware of in designing backup recovery solutions for a virtualized server environment are:

- Virtual servers share and compete for resources on one physical machine (the "managed node" in a Virtual Iron environment). A backup job using a core backup technology is often very resource intensive (CPU, memory, I/O, and LAN utilization). With multiple virtual servers on managed node simultaneously running backup jobs, bottlenecks may result and performance of other critical business application may be adversely affected. This is generally not an issue in a non-virtualized environment, since a server often has a significant amount of under-utilized resources available to meet the demands of the backup job. In addition, from an application server perspective, the scheduling of the backup job has to be coordinated only with the application that runs on the server itself.
- Virtual servers access virtualized storage through the service partition and not directly through a storage interface, such has an FC HBA or iSCSI NIC. As a result, some of the solutions available to a physical server may not be available to a virtual server. For example, a function to initiate a LUN snapshot on an FC-attached storage subsystem may require the host initiating the command to have a Fibre Channel HBA connection into the SAN. Because a virtual server does not have a real (or emulated) FC HBA, the solution may not function properly (depending on the storage array).

Virtual Server Backup Advantages

It is important to note that a Virtual Iron virtualized server environment provides additional capabilities over non-virtualized environments that enhance backup and recovery. These additional capabilities are derived from the fact that the storage environment is virtualized and controlled through the service partition. Through the Virtualization Manager, several Virtual Iron capabilities are available that may be used as a component in an overall backup and recovery strategy. These key capabilities are:

- **Logical disk snapshots**. Snapshot capability allows a point-in-time, pointer-based copy of a logical disk to be taken and used to supplement the backup and recovery process. For example, the snapshot can be mounted on a "proxy host" (that is, another virtual server) and backed up.
- **Logical disk export and import**. These functions allow portable logical disk images to be moved outside the Virtual Iron environment, using the industry-standard Microsoft Virtual Hard Disk (VHD) format. These VHD images can be used to archive the complete state of a virtual server at a point in time.
- **Logical disk cloning**. Disk clones can be used create a consistent copy of a logical disk in another disk group.

While these features themselves do not represent a comprehensive backup and recovery solution, they can be combined with other technologies (core and storage-based) to provide an enhanced capability.

BACKUP STRATEGIES WITH VIRTUAL IRON

Overview

There are three basic backup and recovery techniques that can be deployed in a Virtual Iron virtual server environment. Depending on business requirements, these techniques can be used independently or combined to build a comprehensive backup and recovery solution. The basic techniques are:

1. Install a standard software backup agent on each virtual server and utilize the native functionality of the backup agent to perform LAN-based backup and recovery.
2. Utilize Virtual Iron's logical disk snapshot capability to take a snapshot of a virtual server's logical disks and back up the snapshots on a "proxy backup host," using the standard backup agent (running on the proxy server).
3. Utilize Virtual Iron's raw disks and the native backup and recovery functions of the storage array.

Each of these techniques is discussed in more detail in the following sections.

Backup Agent on Each Virtual Server

This technique requires that the client software for the core backup and recovery technology (TSM, NetBackup, NetWorker, and so on) be installed on each virtual server, just as would be done in a non-virtualized server environment. The installation requirements would be based on the guest operating system (Windows 2003, Windows Vista, SUSE Linux Enterprise Server 9, Red Hat Enterprise Linux 4, and so on). In general all the constraints and limitations associated with a non-virtual server apply to a virtual server.

Advantages

- This method of backup and recovery is well understood and most easily integrated into the current operations, since it is the primary technique used in the non-virtualized server environment. Backup administrators and operations staff will notice very little, if any, changes to their current processes and procedures.
- The granularity of backups is controlled by the client backup software and therefore can be customized to meet the desired recovery objectives. For example, image backups can be done to provide faster recovery of an entire drive of a file system while file-based backups can be done to provide recovery of specific files and directories.
- Most, if not all, of the backup client software functionality can be utilized on the virtual servers. The logical drives and file systems on the virtual servers are not altered in any way by Virtual Iron and therefore are completely compatible with the backup client software.
- Additional backup application agents ("API" agents) can be easily integrated to provide application-aware backup and recovery. For example, the TSM API agent for Oracle functions transparently on a virtual server.

Disadvantages

- Each virtual server requires a backup client license and software installation and customization, potentially increasing overall costs.
- Some advanced functionality may be lost if a specific physical connection to networked storage (iSCSI or Fibre Channel) is required. For example, the TSM agent for Microsoft Exchange has very specific storage and connectivity requirements if it is integrated with Microsoft Virtual SourceSafe (VSS) and storage-based snapshots.

Other Discussion Points

- Installation of the backup client software can be made part of the "master" operating system images that are cloned for use as new virtual servers are installed, thereby mitigating the software installation effort.
- License ramifications will depend on the software and the current licensing agreement. For example, a shop with a site license may not incur additional software licensing costs.
- Backups have to be LAN based since direct attachment of tape (or virtual tape) devices is not supported.

Snapshots and a Proxy Backup Host

Virtual Iron's snapshot functionality provides the ability to make a single (that is, you cannot create multiple snapshots of the same logical disk) point-in-time copy of a logical disk. This snapshot can be used when significant changes are going to be made to a logical disk (for example, during an operating system upgrade) as a means of providing a quick method of rolling back the changes.

However, the snapshot can also be used as the source of a backup when the snapshot is made accessible to and mounted on an alternate virtual server. This alternate virtual server, also called a "Proxy Backup Host" (PBH), mounts the logical disk and then performs a backup of the disk on behalf of the primary server. This enables the backup processing workload to be off-loaded from the primary server.

The steps in the process are as follows:

1. A snapshot of the primary volume is taken. Note that this is a "copy-on-write" snapshot function and therefore it is completed in a matter of seconds.
2. The snapshot logical disk is then assigned to the PBH (another virtual server).
3. The PBH discovers the new logical disk. Today this requires the PBH to be rebooted.
4. The PBH runs the backup of the logical disk. This can be a file-based backup job or an image-based backup job.
5. Once the backup job has been completed, the logical disk is unassigned from the PBH.
6. The snapshot is then deleted.

Advantages

- The backup processing workload is removed from the primary application virtual server.
- The snapshot provides a "crash-consistent" copy (that is, the state of the disk is the same as if the server had experienced a crash) of the logical disk.
- Backup jobs are run from a fewer number of backup clients (that is, one or more PBHs).

Disadvantages

- Additional complexity is added to the backup process.

Other Discussion Points

- As previously mentioned, currently the PBH has to be rebooted in order to discover newly assigned logical disks (snapshots). This may make scripting the overall process somewhat cumbersome.
- Once a snapshot logical disk is mounted on the PBH, there are two possibly ways in which it can be backed up:
  - *As a mounted file system/logical drive.* For example, in a Windows environment the drive could be mounted as the g: drive. Note that this method requires the PBH and the source server to be running the same operating system (for example, Windows).
  - *As a raw block-level backup*. For example, TSM's "image" backup capability can be used to back up a logical disk, which eliminates the need to inspect each file and directory object on the disk. This has the advantage of allowing, for example, a Windows logical disk snapshot to be mounted on a Linux PBH and an image backup performed. This backup technique was tested in Virtual Iron's lab as follows:
    - On a Windows 2003 virtual server a snapshot of the c: drive was created using Virtualization Manager. Virtualization Manager was then used to assign the snapshot to a Linux virtual server (the PBH)
    - The snapshot was then discovered by the Linux PBH (via reboot). The "fdisk –l" command showed the existence of a new disk "/dev/sdb".
    - The snapshot was backed up to the TSM server as an image backup:
      tsm> backup image /dev/sdb
    - The logical disk was removed from the Linux PBH and the server rebooted. Virtualization Manager was then used to delete the snapshot
    - A new logical disk was created, equal to the size of the logical disk on the original Windows server from which the snapshot was taken.
    - This new logical disk was assigned to the Linux PBH and discovered by the server. The existence of the new logical disk was verified (as device "/dev/sdb"):
      # fdisk -l
    - The image backup previously taken was restored to the new disk device using TSM and verified:
      tsm> restore image /dev/sdb /dev/sdb
      # fdisk -l
    - The logical disk was removed from the Linux PBH and the server rebooted
    - A new Windows virtual server was created and assigned the new logical disk (to which the restore operation was directed) as its boot disk.
    - The new Windows virtual server was able to boot from and mount the logical disk (an exact duplicate of the c: drive of the original Windows virtual server).
- In order for this method to be practical in a large production shop, scripting must be used to automate the backup process. Virtual Iron supports an API to the Virtualization Manager via open source Python™ from Python Software Foundation (PSF) and Java. See the appendix for an example of Virtual Iron's scripting to create a snapshot of a virtual server.
- To practically remove the backup workload from a managed node, the PBH should be a virtual server on a different managed node in the same Virtual Iron Virtual Data Center (VDC).

- A backup using a PBH, even in a server environment that is not virtualized, presents challenges in the naming consistency of the items being backed up. For example, with TSM, the high-level naming component (called the "filespace") of a Windows disk device generally contains the drive letter designation. However, the drive letter assigned to the snapshot logical disk on the PBH may not be the same as that assigned to the primary logical disk on the source host. Keeping the names consistent, regardless of the backup application, is important to ensure that backups are consistent and to avoid confusion. These issues should be addressed in the scripts that manage the mounting of the snapshots. It may also be possible to use functions of the backup client software to ensure consistent naming (for example, the TSM "snapshotroot" function).

Raw Disk and Storage Subsystem Functions

Today, many businesses rely on the unique backup and recovery functions in storage subsystems as part of their data protection strategy. In particular, snapshot and mirror (local and remote) functions are frequently used. These functions typically operate at the LUN level and as a result require that the following rules be followed to ensure that these additional copies of the data (snapshots, remote mirrors, and so on) are consistent and meaningful:

- Entire LUNs must be assigned to specific hosts (or clusters)/applications.
- Hosts access LUNs directly without any intervening virtualization or encapsulation.

Therefore, in order for virtual servers to leverage these functions the virtualization layer must allow LUNs to be accessed directly by the virtual servers. Virtual Iron accomplishes this via the use of the "raw disk." Virtual Iron's raw disk allows networked (iSCSI or Fibre Channel) storage LUNs to be accessed directly by the virtual servers without passing through the Linux logical volume manager of the service partition.

Advantages

- All data movement functions are off-loaded to the storage subsystem.
- Existing processes and procedures for storage subsystem-based functions can still be used, easing implementation and overall costs.
- Clustered applications are supported with raw device access.

Disadvantages

- With raw disks, Virtual Iron's advanced logical disk functions (snapshot, clone, archive, and so on) cannot be used.

Other Discussion Points

- The technique used to execute and manage storage array functions is a function of the specific array. For example, some arrays require software to be installed on the application server and have a Fibre Channel connection to the storage array. In other cases, all that is required is an IP connection to the storage array controller.
- With Virtual Iron, the virtual servers do not have a physical connection to networked storage arrays (iSCSI or Fibre Channel). The disks just look like locally attached storage. Therefore, it is important to understand the manner in which storage array functions are controlled when considering using them for virtual servers.

Other Backup and Recovery Options

There are a variety of other options, or combination of the above options, that can be used for backup and recovery in a virtualized server environment. For example, a business may not want to install the client agent of their core backup and recovery technology on every virtual server. This might be especially true for the class of servers whose data does not change very often or those servers that have a more relaxed RPO. For these servers it may be acceptable to perform full backup only once a week and therefore not worth the time and effort to install and manage the client backup agent.

Instead, they may chose to execute a home-grown script on the virtual server that makes an image copy of the virtual server's volume to a disk, generally on a network file server, for example, Network-Attached Storage (NAS). No backup agent software is required to perform this backup (a Linux virtual server could use the "dd" or "tar" command). If desired, these backup images on the NAS storage could then be backed up by a proxy backup host running the client backup agent. Recovery in this situation would be a two-step process in which the proxy backup host first restores the image file to the NAS storage and then virtual server performs a restore from the NAS storage.

SUMMARY

Server virtualization has a significant impact on physical storage – especially as users take server virtualization into production environments. Virtualized servers have unique characteristics when it comes to storage and backup and recovery and the techniques that a business currently uses for their physical server environment may not be best in a virtualized environment. It's important that users understand the implications of virtualization on their backup and recovery infrastructure.

Virtual Iron (VI) is an enterprise-class server virtualization technology that is designed to integrate easily and effectively with physical storage environments and enables flexible backup and recovery. It fully leverages industry standards and the virtualization capabilities of current modern x86 processor chips and can be used to address many of the key server management challenges within a data center.

APPENDIX: SNAPSHOT SCRIPT

```
#
# vsSnapshot - snaphot a VirtualServer
#
from com.virtualiron.vce.mgmt.api import VirtualizationManager
from com.virtualiron.vce.mgmt.api.physical import
from com.virtualiron.vce.mgmt.api.virtual import *
import java.lang
import os
import string
import sys
import traceback

def Usage():
       if os.name == 'nt':
              command = 'runner.bat'
       else:
              command = 'runner.sh'
       print 'Usage: %s --inputfile=vsSnapshot.py --vs="Virtual Server" --
bootDevice="dsl.iso"' % (command)
       sys.exit(1)

#
# parse command line arguments
#
vsName   = None
bootName = None
for arg in sys.argv[1:]:
       if string.find(arg, "--vs=") != -1:
              tokens = string.splitfields(arg, "=")
              vsName = tokens[1]
       if string.find(arg, "--bootDevice=") != -1:
              tokens = string.splitfields(arg, "=")
              bootName = tokens[1]

#
# check if required arguments were specified
#
if vsName is None or bootName is None:
       Usage()

#
# get connection to Database
#
configurationManager = VirtualizationManager.getConfigurationManager()
foundry = configurationManager.getFoundryContext()

vs = configurationManager.findObject(VirtualServer, vsName)
if vs is None:
       print 'FAIL to find VirtualServer %s' % (vsName)
       sys.exit(1)

bootDevice = configurationManager.findObject(StorageDevice, bootName)
       print 'FAIL to find bootDevice %s' % (bootName)
       sys.exit(1)

# set bootType depending on type of bootdevice (nbd or disk)
if NetworkBlockDevice.isAssignableFrom(bootDevice.getKind()):            # nbd iso
       bootType = VirtualServer.BootInfo.VS_BOOT_DEVICE_CDROM
elif LvmStorageDevice.isAssignableFrom(bootDevice.getKind()):            # virtualdisk
or logicaldisk
```

```
        bootType = VirtualServer.BootInfo.VS_BOOT_DEVICE_DISK
else:
        print 'FAIL: %s is not a VirtualDisk, LogicalDisk or NBD-Device' % (bootName)
        sys.exit(1)

#
# snapshot VS
#
try:
        jobName = java.lang.Long.toString(configurationManager.getLocalTime())
        job = configurationManager.createJob(jobName)
        job.begin()

        snapvs = vs.snapshot()
        job.addOperationDescription("Snapshot " + vs.getName(), vs, vs, vs)

        job.commit()

except java.lang.Throwable, throw:
        job.abort()         # if job fails, rollback
        throw.printStackTrace()
except:
        job.abort()         # if job fails, rollback
        traceback.print_exc()

if snapvs is None:
        print 'FAIL to snapshot VS'
        sys.exit(1)

print "Snapshot:", vs.getSnapshot()

#
# change VS snapshot bootDevice
#
try:
        jobName = java.lang.Long.toString(configurationManager.getLocalTime())
        job = configurationManager.createJob(jobName)
        job.begin()

        snapvs.addStorageDeviceFirst(bootDevice)
        job.addOperationDescription("Set BootDevice to " + bootDevice, snapvs, snapvs,
snapvs)

        snapvs.setBootType(bootType)
        job.addOperationDescription("Set BootType to " + bootType, snapvs, snapvs,
snapvs)

        job.commit()

except java.lang.Throwable, throw:
        job.abort()         # if job fails, rollback
        throw.printStackTrace()
except:
        job.abort()         # if job fails, rollback
        traceback.print_exc()

#
# start VS snapshot
#
try:
        jobName = java.lang.Long.toString(configurationManager.getLocalTime())
        job = configurationManager.createJob(jobName)
```

```
	job.begin()

	snapvs.start()
	job.addOperationDescription("Start VirtualServer " + snapvs.getName(), snapvs,
snapvs, snapvs)

	job.commit()

except java.lang.Throwable, throw:
	job.abort()		# if job fails, rollback
	throw.printStackTrace()
except:
	job.abort()		# if job fails, rollback
	traceback.print_exc()

#
# wait for VS Snapshot to go Running
#
EventLog = foundry.getEventLog()

count = 0
found = 0
while not found and count < 10:
	count += 1
	Events = eventLog.listenForEvents(VirtualServerEvent, snapvs, 60*1000)
	if Events is None:
		print 'No Events'
	else:
		for event in Events:
			print "Rcvd:", event
			if VirtualServerRunningEvent.isAssignableFrom(event.getKind()):
				found = 1

if not found:
	print 'VS not RUNNING', vs.getStatusEvent()

#
# wait for VS snapshot to shutdown, wait for VS Stopped event
#
found = 0
while not found and count < 10:
	count += 1
	Events = eventLog.listenForEvents(VirtualServerEvent, snapvs, 60*1000)
	if Events is None:
		print 'No Events'
	else:
		for event in Events:
			print "Rcvd:", event
			if VirtualServerStoppedEvent.isAssignableFrom(event.getKind()):
				found = 1

if not found:
	print 'VS not RUNNING', vs.getStatusEvent()

#
# delete VS snapshot and all its snapshot logical disks
#
try:
	jobName = java.lang.Long.toString(configurationManager.getLocalTime())
	job = configurationManager.createJob(jobName)
	job.begin()
```

```
      Disks = snapvs.getStorageDevices()
      for snapdisk in Disks:
            if LogicalDisk.isAssignableFrom(snapdisk.getKind()) and
snapdisk.getSnapshotParent() is not None:
                        lvmDiskGroup = snapdisk.getParentLvmDiskGroup()
                  lvmDiskGroup.deleteLogicalDisk(snapdisk)
                  job.addOperationDescription("Delete Snapshot Disk" + snapdisk,
snapdisk, snapdisk, snapdisk)
            else:
                  print "WARNING: %s not deleted" % (snapdisk)

       vs.deleteSnapshot(snapvs)
      job.addOperationDescription("Delete Snapshot " + snapvs.getName(), snapvs,
snapvs, snapvs)

      job.commit()

except java.lang.Throwable, throw:
      job.abort()         # if job fails, rollback
      throw.printStackTrace()
except:
      job.abort()         # if job fails, rollback
      traceback.print_exc()
```