Eric Siebert

THE EXPERT GUIDE TO VMware Data Protection and Disaster Recovery

Sponsored by VEEAM

CONTENTS

CONTENTS 2
INTRODUCTION 2
TRADITIONAL BACKUP AND RECOVERY METHODS 3
- OPERATING SYSTEM BACKUP AGENTS 3
- SCHEDULING AND PERFORMING BACKUPS 4
- RESTORING DATA 5
- TESTING AND VERIFICATION OF BACKUPS 6

VIRTUAL ENVIRONMENT BACKUP METHODS 7
- IMAGE-LEVEL BACKUPS 7
- FILE-LEVEL BACKUPS 8
- VIRTUAL MACHINE SNAPSHOTS 9
- CONSISTENT BACKUP STATES 11
- VCB & VSTORAGE APIS 13
- SCHEDULING AND PERFORMING BACKUPS 15
- VIRTUALIZATION ADVANTAGES 16

SUMMARY 17
ABOUT THE AUTHOR 18
ABOUT THE SPONSOR 18

INTRODUCTION

Backup methods have evolved over the years as new technology has become available, but the introduction of virtualization technology changes everything and introduces more options and flexibility when backing up your servers. While traditional physical backups have evolved over the last few decades, the changes have mostly been with the media being used on the target device. From the original punch card backups, to floppy disks, to magnetic tape, to optical disks and finally to hard disks, the target media has greatly improved over the years. The one area that has not changed that much is with the methods, which mainly involve using an agent installed inside the operating system that a backup server connects to over the network to copy data to the target device. With the introduction of virtualization, this method still works, but it is not as efficient because of the architectural differences in virtual environments. Companies like Veeam recognized the need for new backup and recovery methodologies for virtual environments and developed applications that were made to order for virtualization.

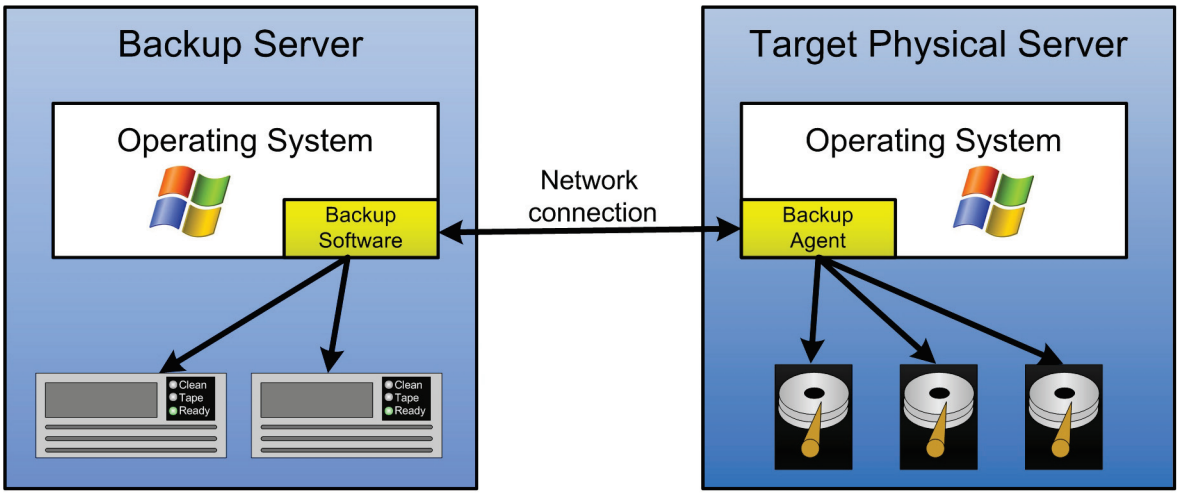
This chapter looks at traditional backup and recovery methods and why they do not work well in virtual environments, as well as the new backup and recovery methods that were developed specifically for virtualization.

TRADITIONAL BACKUP AND RECOVERY METHODS

When we refer to traditional backup and recovery methods we are talking about the methods used with a single guest operating system installed directly on a physical server, which was how things were done before virtualization was introduced by VMware for x86 servers. This section covers the methods that have been used for both backing up and restoring physical servers so you are aware of how these methods relate to those used in virtual environments.

Operating system backup agents

This method relies on a software backup agent that is installed inside the operating system. The backup server contacts the backup agent, which is always running on the server, and establishes a session over a TCP/IP port with the agent. Once communication is established, the agent can copy all the data from the operating system file system to the backup server, which can then write the data to the target media, either a tape or disk system. Because the backup agent is installed inside the operating system, it can read the file systems of all the disks that are configured on the server. It can access the Master File Table of the OS to determine all of the files that exist on the disk partitions so it can copy them one by one to the backup server. This method of backing up a server is considered a file-level backup because files are being copied at the file system level from one device to another, much like you would copy files using Windows Explorer or the command-line COPY command. To support incremental backups, an archive bit value is changed whenever a file is modified by the operating system. Once this value is changed, an incremental backup recognizes that the file was modified since the last backup and includes it in the incremental backup and then clears the archive bit.



Traditional backup method using operating system agents

Since this method works at the file level inside the OS, only active files inside the OS are backed up; all empty disk blocks and deleted files are ignored. This results in a full backup equal in size to the amount of disk space the files take up on disk regardless of how big the disk is. For incremental backups, however, this method is inefficient because if only a small part of a file changes, the entire file is backed up because it has been flagged as modified. For example, if you had a 500MB database on your disk and only a few records totaling 1MB were updated since the last backup, the entire 500MB would be backed up again because the backup agent works at the file level and not the block level. To help with this big limitation, special backup agents are

deployed for certain applications like Microsoft Exchange that know how to back up only the changed records for that application and not the entire database.

Another limitation with this method is with files that are open and in use when the backup runs. These files are typically locked by the OS or application, which means the backup agent cannot access them. To cope with this problem and to ensure that all files are backed up, backup applications can use special open file agents. The Microsoft Volume Shadow-copy Service (VSS) that is built into most new Windows versions is an example of this. VSS works by briefly freezing application write I/O requests while the shadow copy is created. It then flushes the file system buffers and freezes the file system to ensure that all data that is shadow-copied is written in a consistent order. VSS works at the disk block level and creates a clone or shadow copy of the disk volume that is a temporary read-only copy of all the data on the volume. Because this copy is read-only it can be read by the backup application regardless of any applications that may have the file open and are writing to them. All writes to the files happen on the original disk blocks while the shadow copy is active, but before the write happens the original disk blocks are written to a special differences area that is part of the shadow copy. Because of this the creation of the shadow copy is a quick process and does not take up much disk space as only changed disk blocks are written to the shadow copy. Once the backup completes the shadow copy is discarded.

Backups performed within the operating system have worked well for many decades with traditional physical servers. With the introduction of virtualization, however, they have become inefficient, and alternative methods that leverage the strength and architecture of virtualization were needed instead. Companies like Veeam recognized this and developed backup methods specifically designed for virtual environments that bypass the operating system layer and back up VMs at the virtualization layer instead. With virtualization becoming more and more popular the days of backing up through operating system agents is numbered as more efficient methods are used instead.

Scheduling and performing backups

Backing up a server is a resource-intensive operation especially at the network and storage layers. When performing backups of traditional physical servers, the resource usage is limited to the server being backed up. Many servers can be backed up simultaneously with the resource bottleneck typically at the backup server, which has limited tape drives and network bandwidth. Most data centers have backup windows, which is a period of time where usage is low so backups will not affect users who access the applications running on the servers being backed up. Because each server has its own resources, one server being backed up will not have any impact on another, which makes scheduling backups fairly easy.

When backups are performed the backup server connects to each individual server and copies the data from that server to the target backup device, which is typically a tape system. Tapes are a convenient backup media because they can hold a lot of data and can easily be sent offsite for storage. There have been many different types of tape formats over the years such as DLT (Digital Linear Tape), AIT (Advanced Intelligent Tape), DAT (Digtal Audio Tape) and, one of the most popular today, LTO (Linear Tape-Open). The first generation of LTO tapes (LTO-1) was introduced in 2000 and had a maximum native capacity of 100GB with a maximum backup speed of 15 MB/s. Today the 5th generation of LTO tape (LTO-5) has a native capacity of 1.5TB and a maximum backup speed of 140 MB/s.

While tapes are used by many organizations to back up their data, many companies are implementing disk systems as a backup target either in place of or to complement a tape backup system. When using a disk system as a target, data that is backed up to a disk target can be replicated offsite after the backup completes. A disk target can also be used as a holding area for data that is going to be backed up to tape; the data is quickly and easily copied to the disk target during the backup and then later copied to tape without involving the source server. Backing up to disk can also make restores much easier and quicker as you do not have to worry about finding and loading backup tapes. In addition, cheap disk systems can be used as the target storage to provide a relatively high capacity for an inexpensive price.

Backups also have a number of different backup models that can be used to back up data. The different methods are designed to reduce both the amount of time and the amount of space that backups require. Listed below are the different backup methods that are commonly used:

- **Full**—This is where all data on the source server is copied to the backup target. Full backups require the most time to perform and take up the most space. Full backups are the foundation for the other backup types.
- **Synthetic**—With synthetic backups a full backup is only done once and subsequent backups are all incremental backups. This method provides smaller backup windows and less resource consumption than traditional backups as you never have to do periodic full backups. Once an incremental backup takes place it is combined with previous backups to synthesize a full backup. This way an up-to-date full backup copy is always on hand without ever having to perform a full backup. You can still restore older data as well because all changes are backed up and saved as rollback files and historical data is used to calculate reverse increments. This method is also referred to as reverse-delta.
- **Incremental**—This method only backs up data that has changed since either the last full or incremental backup. This method relies on an archive bit that is cleared after the previous backup. Once a file is changed the archive bit is set, which indicates the file has changed since the last backup. The incremental backup then only backs up those files that have the archive bit set and then clears the archive bit once the files are backed up. While this method is quick, it can also make restores more difficult as multiple backup points may have to be used to restore files that have changed since the last full backup.
- **Differential**—This method is similar to incremental backups, but all changed data since the last full backup is backed up every time a differential backup runs. As changed files are backed up with differential backups, the archive bit is not cleared, so any file that is changed since the last full back is backed up with each subsequent differential backup. While this method takes more time and more space on the target, restores are easier since only one backup point is needed to restore files that have changed since the last full backup.

Most companies implement a backup schedule that includes periodic full backups (e.g., once a week), followed by incremental or differential backups until the next full backup occurs, and then the cycle repeats. Backup schedules will vary based on each company's requirements, which include retention timeframes, backup windows, budgets and infrastructure.

Restoring data

When it comes to restoring data from backups, there are two ways to do it: restoring individual files and directories, or restoring all the data on the server. Restoring a select group of files is the most common type of restore, which may result from a scenario where files were accidently

deleted or previous versions of a file are needed. To restore select files back to a server, a point in time is selected from the available backups that exist for the server. Once the files are selected the media that the files are backed up on must be available before the restore can begin. Oftentimes this involves locating and re-calling tapes from offsite storage; if the data is on an existing snapshot it can also be leveraged for this. Once the point-in-time version of the file is located from the restore media, it is copied either back to its original location, overwriting the existing copy or to an alternate location where it can be accessed without disturbing the original file.

Restoring a complete copy of a server is referred to as a bare-metal restore as nothing but the bare metal of a server is needed without any requirement for an operating system being installed on the server. The restore includes the operating system along with all the applications and data that reside on it. This type of restore is typically only needed in the event of a complete hardware failure where all data on the server is lost. It is also used for disaster recovery purposes where a whole copy of a server must be restored because of a failure at the primary site. Because the backup includes the operating system, disk partitions, system state and all of its device drivers, similar server hardware is often needed for the restore to work properly. Performing a bare-metal restore can be complicated and tricky and proper planning needs to be done to ensure a successful restore operation. To complete a bare-metal restore an operating system needs to be installed on the server first so a backup agent can be installed. Once that has been completed the backup server can communicate with the agent to restore all of the files including system state information on top of the installed operating system. Some backup products can also leverage network boot capabilities using PXE to load boot images from a PXE boot server to eliminate the need for an OS to be installed first to perform a bare-metal restore.

Testing and verification of backups

While backups are a critical part of any computing department, being able to successfully restore data when needed is even more critical. Backups are pointless if you can't restore the data—imagine backing up servers for months only to find out when you need to restore some critical data that your backups have not been working properly. Backups are like a good insurance policy: you pay your premiums every month hoping you will never have to use them, but when something bad happens you need to have something to fall back on. Because of this you shouldn't just trust that your backups are working properly—you need to periodically verify this by trying to restore data successfully.

Verification of backups is more than just verifying through the backup application that the backup completed successfully and that the media is error-free. It also involves being able to properly restore files, applications, databases and whole servers when needed. The problem with verifying backups is that it can be a complicated and time-consuming process. While restoring individual files or directories to an alternate location may be simple enough, trying to restore applications or a whole server can be challenging. When you restore an application you need to ensure that it is in a working state, and restoring multi-tier applications can be even more difficult. You need a separate environment in which to perform the restore so you do not affect your production environment. This often requires having one or more physical servers available to perform the restore on; when testing bare-metal restores this can be difficult because hardware similar to the server that was backed up is needed. In data centers that have many model and generations of servers this can be even more difficult. As a result backup verification in traditional physical server environments can be a tedious and difficult process.

VIRTUAL ENVIRONMENT BACKUP METHODS

While traditional methods that are used with physical servers can be used in virtual environments, they are not very efficient. Because the architecture of a virtual environment is drastically different from a physical environment, the traditional methods can create bottlenecks, cause performance problems and take longer to complete. Virtual environments require methods that leverage the strength of the virtualization architecture to perform highly efficient backups with minimal impact on the virtual machines. In this section we detail the different backup and recovery methods that are used in virtual environments that are much different than traditional methods.

Image-level backups

With virtualization it is more efficient to back up the single large virtual disk file (vmdk) at the virtualization layer instead of going through the guest operating system to back up files individually. This type of backup is known as an image-level backup because you are backing up a whole image of the virtual machine's disk file, much like you would if you used an imaging utility like Ghost. While it is more efficient backing up just one large file instead of thousands of smaller files, there is one issue with this. Since image-level backups cannot see inside the operating system and are backing up the whole virtual disk they are also backing up empty disk blocks and deleted files as well. If a VM has an 80GB virtual disk file and only 20GB is in use, 80GB is backed up with an image-level backup; with a file-level backup only the 20GB in use is backed up. To get around this, backup vendors like Veeam that backup at the virtualization layer detect empty disk blocks that have not been written to by the operating system yet and ignore them. While checking for empty disk blocks they also use inline deduplication to detect duplicate blocks and ignore them as well.

You might wonder how an image-level backup can handle open files and avoid corruption from files that change when the backup occurs if it cannot see inside the guest operating system. This is done by first quiescing the VM using a special driver (either VMware Tools or a special driver supplied by the backup vendor) that runs inside the guest operating system that momentarily pauses the running processes on a guest and forces the operating system and applications to write any pending data to disk. Once that is complete a snapshot of the VM is taken at the virtualization layer, which creates a new temporary virtual disk file (delta) for any new disk writes that occur on the VM and prevents the original disk from being written to while the backup is running. Once the backup is completed, the temporary virtual disk file is merged back into the original disk file and the snapshot is deleted.

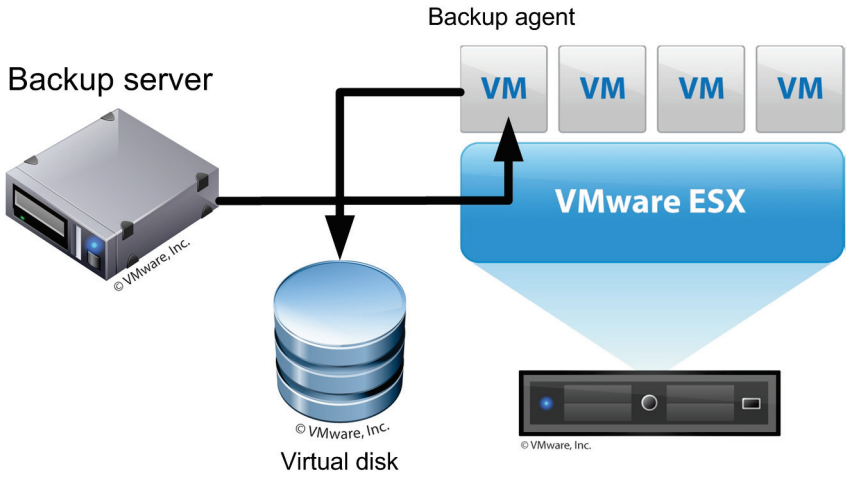
A common misconception with image-level backups is that you cannot do incremental backups because you are only backing up one large file and if any disk block changes the whole file must be backed up again. With traditional file-level backups only the files that have changed are backed up on incremental backups; this is noted by setting a flag called an archive bit that indicates that a file has changed since the last backup. Once the file is backed up the archive bit is cleared until it changes again. With image-level backups, a backup application has to keep track of all the blocks that have changed since the last backup so it knows which ones to back up when doing incremental backups. This process can increase the time of backups as the backup application must calculate a hash for each block and then scan the entire virtual disk and compare it against a hash table to see what has changed since the last backup. To speed up incremental backups, most backup vendors have taken advantage of the new Changed Block Tracking (CBT) feature that is accessible via the vStorage APIs for Data Protection. This allows the

backup application to simply query the VMkernel to find out which disk blocks have changed since the last backup, which greatly speeds up incremental backups.

Image-level backups offer some advantages over traditional file-level backups of physical servers. Having the server encapsulated into one big file makes for easy portability—the virtual disk file can easily be copied to any other storage device. For example, one could easily copy a VM from a host server to another storage device, external hard drive or flash drive for safekeeping. This makes creating ad-hoc backups of VMs a simple process. It also makes performing bare-metal restores much easier as all you need to do is restore the files that make upmakeup the VM to any host and you are up and running.

File-level backups

File-level backups are traditionally done using an agent inside the guest operating system that is aware of all the individual files. While this can be done in a virtual environment, it can cause excessive resource usage on the host server, which can negatively affect the other VMs on that host. Backups that use OS agents on virtual machines must navigate through the virtualization layer to get to the guest operating system layer to back up files as depicted below.



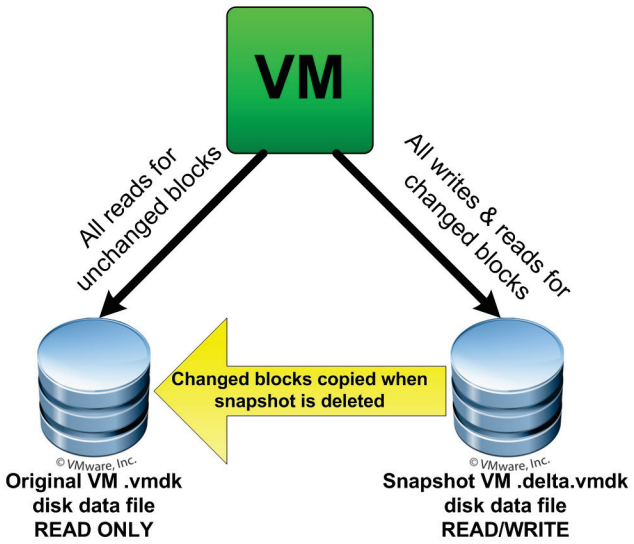
Backing up a VM using a traditional agent inside the OS

As a result, backing up a VM using a backup agent running inside the OS is not very efficient as it increases network and disk I/O as well as CPU utilization on the host while the backup is running. This results in fewer resources for the other VMs on that host. If multiple backups are running on the host, the problem will be even worse and can seriously degrade the performance of the host. This can also have an impact on your consolidation ratios as well as you will be forced to have less VMs per host to stay inside your backup windows.

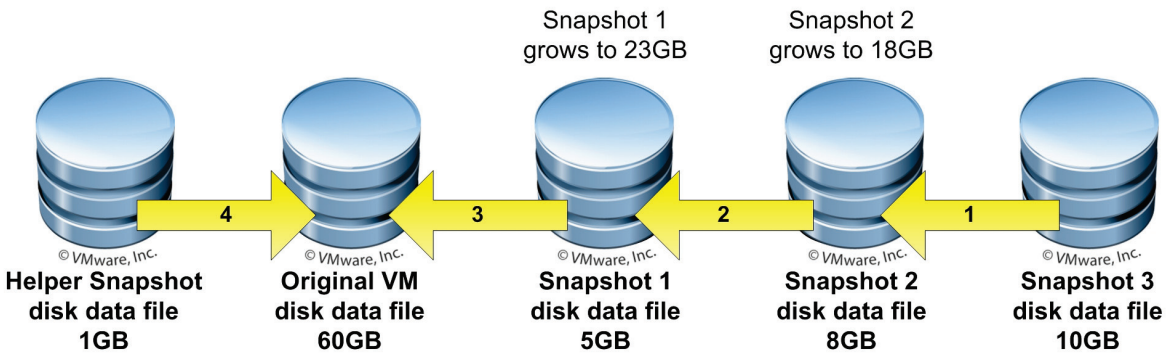
Most virtualization backup tools perform image-level backups at the virtualization layer outside of the guest OS, but they can still restore individual files if needed. This is possible because the backup software can mount the backed up virtual disks and browse the file system to access any file on it. Because of this, file-level backups are no longer needed in a virtual environment as backup applications can provide file-level restore capabilities using image-level backups for most major operating systems.

Virtual machine snapshots

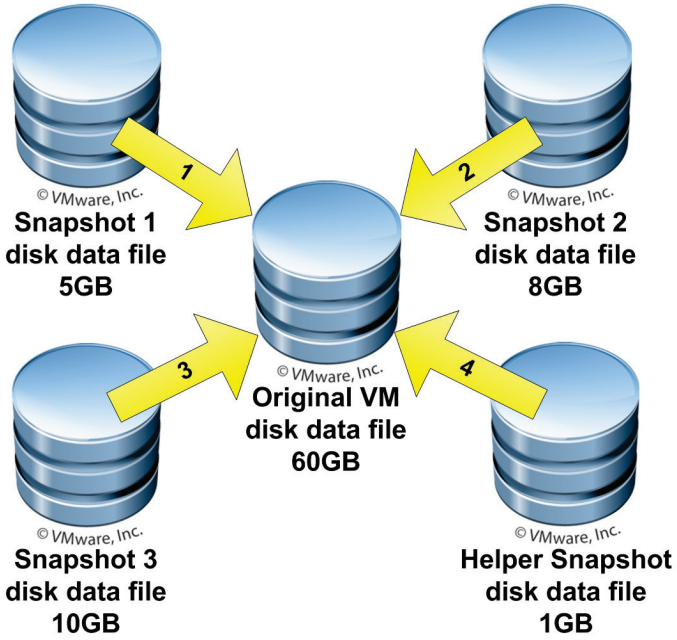
Snapshots are one of the great features that virtualization provides and can be a real lifesaver when upgrading or patching applications and servers. A snapshot is a point-in-time picture of a VM that preserves the disk file system and system memory of the VM. You can create more than one snapshot of a VM so you have multiple restore points available to recover from. When you create a snapshot all writes to the original VM disk files are suspended and they become read-only from that point on. All writes to the original disk files are deflected to new delta virtual disk files that are created when the snapshot is taken. When you delete all snapshots for a VM, all of the delta files that were created are merged back into the original vmdk disk file for the VM and then deleted when completed. If you choose to only delete an individual snapshot, then just that snapshot is merged into its parent snapshot and deleted.



The process for deleting multiple snapshots has changed across vSphere versions. In all versions, when you delete a snapshot before the data is committed back to the original disk file, a helper snapshot is first created to hold any disk writes that are made while the snapshots are being written back to the original disk. In some older 4.0 versions of vSphere, if a VM had 3 snapshots active and you deleted them all, the process would be Snapshot 3 would be copied to Snapshot 2, which would be copied to Snapshot 1, which would be copied to the original disk file and then the helper snapshot would be copied to the original disk file as outlined below.



This process would result in extra disk space being needed as each snapshot would grow as the previous snapshot was added to it. If there was not sufficient free disk space available on the datastore, the snapshots could not be committed. In later vSphere 4.0 versions and in vSphere 4.1, instead of each snapshot being merged into the previous one, each snapshot is merged directly back into the original disk in turn. So if a VM had 3 snapshots active and you deleted them all, the process would be Snapshot 1 would be copied to the original disk, Snapshot 2 would be copied to the original disk, Snapshot 3 would be copied to the original disk and then the helper snapshot would be copied to the original disk file as outlined below.



Because each snapshot is merged back into the original directly and one at a time, no extra disk space is needed except for the helper file. If you choose to revert to a snapshot, the current disk and memory states are discarded and the VM is brought back to the state of the snapshot you reverted to. Whichever snapshot you revert to then becomes the new parent snapshot. The parent snapshot is not always the most recently taken snapshot; if you revert back to an older snapshot it then becomes the parent of the current state of the virtual machine. The parent snapshot is always noted by the "You are here" label under it in Snapshot Manager.

A single snapshot file can never exceed the size of the original disk file; the reason for this is that any time a disk block is written to, it is created once in the delta file and simply updated if changed later on. If you changed every single disk block on your server after taking a snapshot, your snapshot would be the same size as your original disk file. However, the combined space of multiple snapshots could easily exceed the size of the original disk file. Snapshot files will initially be small in size (16MB) and grow larger as writes are made to the VM's disk files. Snapshots grow in 16MB increments to help reduce SCSI reservation conflicts. Whenever a request is made to change a block on the original disk, it is instead changed in the delta file. If the previously changed disk block in a delta file is changed again, it will not increase the size of the delta file because it simply updates the existing block in the delta file. The rate of growth of a snapshot will be determined by how much disk write activity occurs on your server after the snapshot is taken. Servers that have disk write-intensive applications like SQL and Exchange will have their snapshot files grow rapidly. On the other hand, servers with mostly static content and fewer disk writes like web and application servers will grow at a much slower rate. When you create multiple

snapshots, new delta files are created and the previous delta files become read only. With multiple snapshots, each delta file can potentially grow as large as the original disk file.

Virtual machine snapshots should not be considered as a primary method for backing up VMs. They are useful for short-term backups that are needed on the fly when it is necessary to preserve the state of a VM. The reason for this is that snapshots slightly degrade the performance of a VM as they grow and they also consume extra disk space on datastores. Additionally, because a VM's disk is split it creates the potential for problems and can prevent certain VM operations and features from being used. Snapshots should be closely monitored so you do not leave them running longer than necessary. The larger they grow the longer they can take to merge back into the original disk file when they are deleted. The commit operation that occurs when a snapshot is deleted can be very resource intensive on the VM while it is occurring.

While VM snapshots should not be used as a primary backup means, they are a primary enabler for performing image-level backups. Performing a snapshot before doing an image-level backup is necessary to prevent any disk blocks from changing while the virtual disk file is being backed up. The snapshot makes the VM's virtual disk read-only so the backup application can mount it and have exclusive access to it while the disk blocks are copied from it. Once the backup completes, the snapshot is deleted and all changes are written back to the original disk. Almost all virtualization backup applications rely on VM snapshots to perform image-level backups.

Consistent backup states

Whenever a snapshot of a VM is taken prior to backing it up, it is important to ensure it is in a consistent state so data can be properly restored if needed. This is especially important for transaction-sensitive applications like Active Directory, Exchange and SQL Server. When a snapshot of a VM is taken, its disk is frozen so no more writes occur to it while it is being backed up. At the point that the disk is frozen, however, there may be pending transactions or data in memory that has not yet been written to disk. This missing data can cause part of the backup data to be corrupt or incomplete. Because of this, before the snapshot is taken it is important to quiesce the VM, which temporarily halts the operating system and applications while they write any pending data to disk. Quiesce is a term used to describe the operation of pausing a computer while all outstanding writes are flushed to disk. Once this operation completes, the snapshot is taken and the operating system and applications can proceed as usual. This operation guarantees that the now read-only disk is in a state where applications and data can be properly restored if needed. There are several different states that a server can be in when the snapshot is taken, as outlined below:

- **Crash consistent**—this state is the same as if a VM hard crashed or had its power turned off without being properly shut down. All pending transactions and data in the VM's memory are lost and not written to disk. This is the default state if you take a snapshot of a VM without quiescing it first.
- **File system consistent**—in this state the operating system is quiesced, which allows for the operating system to write any pending data to disk before the snapshot is taken. This state does not take into account any applications that may be running and that may need to take additional steps to properly write all data to disk. This state is better than crash consistent but only ensures that the operating system is in a proper state to be backed up and not the applications running on it.

- **Application consistent**—in this state both the operating system and applications are quiesced so the VM is in a state where both the operating system and applications can be properly restored. This is the best possible state a VM can be in to ensure good backups. This type of quiescing only works with applications that specifically support being told to pause and write pending data when necessary. Typically this includes transaction-based applications like databases, email servers and financial systems.

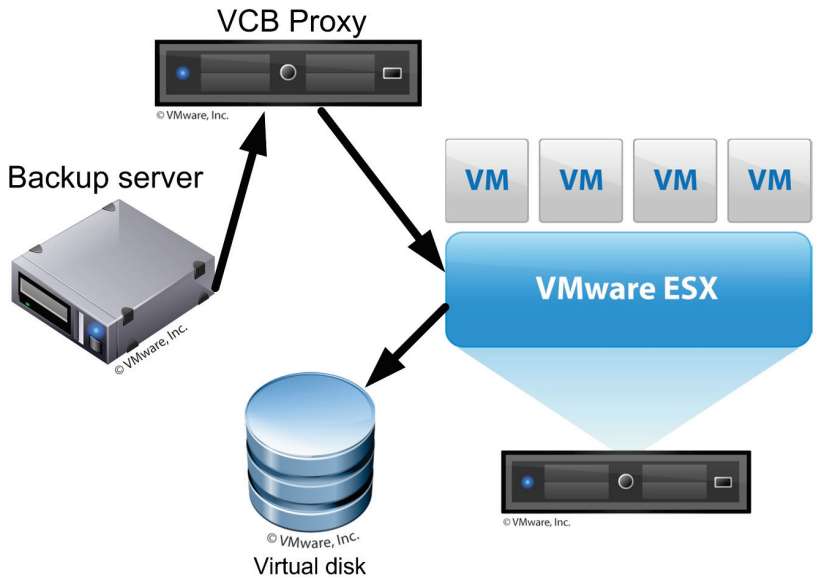
For VMs running Windows the quiescing is done through Microsoft VSS service, which works with the file system and applications to pause them and put them in a proper state for a snapshot and backup. The VSS service has several components that work together to take a shadow copy and put the disk in a consistent state, as outlined below:

- **Requestor**—this is the component that initiates the request to the VSS service. This is typically a backup application like Veeam Backup & Replication. Requestors also work with writers to collect information about the data to be backed up.
- **Writers**—this is the component that is part of applications and services that are designed to work with the VSS service. The writers work with VSS to prepare applications to quiesce their data and to ensure that no data is written until the shadow copy is created. Doing this ensures they are in a proper state to be backed up and all data in memory is written to disk.
- **Providers**—this is the component that does the actual work of creating the shadow copy. Once the writer has done the work to ensure that applications are quiesced, the provider creates and maintains the shadow copies until they are no longer needed. There are different types of providers that can be used, including hardware providers that offload the shadow copies to hardware storage devices, software providersthat work at the software layer to intercept and process I/O requests and write them to any type of storage device, and system providers that are built into the Windows OS and write to a NTFS volume on the system.

Having application-consistent backups is critical to ensure that your data is properly backed up. Most virtualization backup applications leverage the VMware Tools application that is installed inside the VM to quiesce the operating system and applications prior to creating the snapshot for backups. Instead of relying on VMware Tools, some vendors like Veeam also include their own agent that can run inside the VM and work directly with the backup application to quiesce the VM when needed. The reason for this is that VMware can be slow to update the integration with VSS in VMware Tools as changes are made to Windows. By not relying on VMware Tools to handle the quiescing, Veeam can be quicker to respond to changes or new versions in the Windows OS and doesn't have to wait for VMware to update its VSS integration in VMware Tools.

VCB & vStorage APIs

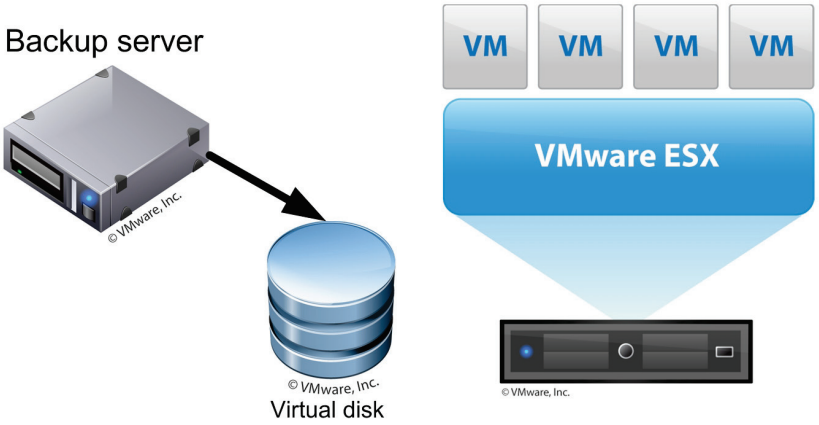
VMware recognized that a more efficient way of doing backups was necessary and developed methods to perform backups at the virtualization layer and never enter the guest operating system. VMware's first attempt at this was the VMware Consolidated Backup (VCB) framework included with VI3. VCB acted as a proxy server to offload backups from the virtual machine by mounting the virtual disk on the VCB server and then doing an image-level backup of it without involving the host or the VM. This shifted the backup overhead from the VM and the host to the proxy server instead. While this was a step in the right direction, it required a middleman between the backup device and the target disk as shown below:



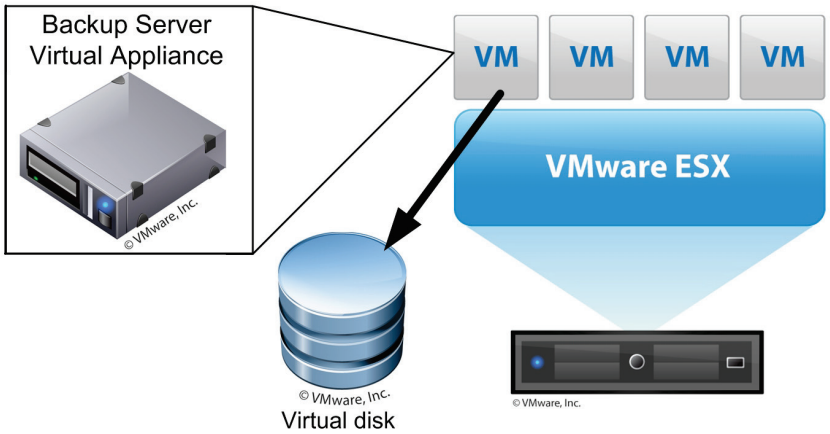
Backing up a VM using a VCB Proxy offload

With the vSphere release VMware eliminated VCB and the proxy that was used and instead leveraged APIs and SDKs so backup vendors could directly connect to virtual storage targets to backup VMs. The new vStorage APIs for Data Protection (VADP) include the functionality that was available in VCB and also added new functionality such as Changed Block Tracking (CBT) and the ability to directly interact with the contents of virtual disks. Doing this was much more efficient and offered more features than using VCB to back up virtual machines.

The two methods for using VADP to back up VMs are having the backup server directly access VM datastores and using a virtual appliance to hot-add a VMs disk file directly from its datastore as shown below:



Backing up a VM using VADP with direct disk access



Backing up a VM using VADP with a virtual appliance

The vStorage APIs existed in VI3 but were referred to as the VCB Backup Framework; however, unlike VCB they are not a separate standalone application and instead are built directly into ESX(i) and require no additional software installation. While the VCB Backup Framework still exists in vSphere and can be used by backup applications, the vStorage APIs are the successor to VCB and will eventually completely replace it.

The vStorage APIs are not really a single API, and the term is basically just a name for a collection of interfaces that can be utilized by 3rd party applications to interact with storage devices in vSphere. These interfaces consist of various SDKs that exist in vSphere and also the Virtual Disk Development Kit (VDDK). The VDDK is a combination API and SDK that enables vendors to develop applications that create and access virtual disk storage. The VDDK is used in conjunction with other vStorage APIs to offer a complete integrated solution for management of storage in vSphere. For example, while VM snapshots can be managed using the SDK functionality, other operations like mounting virtual disks are handled through the VDDK.

The vStorage APIs are broken into 4 groups that have different types of functionality, with the vStorage APIs for Data Protection being most beneficial to backup and replication applications. Changed Block Tracking is the standout feature as it allows 3rd party applications to query the API to find out which disk blocks have changed in a virtual machine's disk file since the last backup operation. Without this feature applications would have to figure this out on their own, which can be time-consuming; now with CBT they can instantly find this out so they know exactly which disk blocks need to be backed up. This enables much faster incremental backups and also allows for near-continuous data protection when replicating virtual disk files to other locations.

CBT is supported on any storage device and datastore in vSphere except physical mode Raw Device Mappings, which includes iSCSI, VMFS, NFS and local disks and it also works with both thin and thick disk types. As CBT is a new feature to vSphere, it does require that the virtual machine hardware be version 7, which is the default in vSphere. CBT is disabled by default as there is a very slight performance penalty that occurs when using it. It can be enabled on select VMs by adding parameters (ctkEnabled=true and scsi#:#.ctkEnabled=true) to the configuration file of the VM; backup applications can also enable it using the SDKs. Once CBT is enabled, a VM must go through something called a stun/unstun cycle for it to take effect; this cycle happens during certain VM operations including power on/off, suspend/resume and create/delete snapshot. During this cycle a VM's disk are re-opened, which allows a change tracking filter to be inserted into the storage stack for that VM.

The CBT feature stores information about changed blocks in a special "-ctk.vmdk" file that is created in each VM's home directory. This file is fixed length and does not grow; the size will vary based on the size of a virtual disk (about .5MB per 10GB of virtual disk size). Inside this file the state of each block is stored for tracking purposes using sequence numbers that can tell applications if a block has changed or not. One of these files will exist for each virtual disk for which CBT is enabled.

The vStorage APIs for Data Protection and the CBT feature make backups quicker and easier in vSphere and are a big improvement over VCB. Veeam was quick to recognize the many advantages that the vStorage APIs offered, and Veeam Backup & Replication was the first backup application to fully embrace and make use of the vStorage APIs.

Scheduling and performing backups

With physical environments servers don't share resources like VMs do so you can typically schedule backups without regard for other backups that may be running at the same time. Scheduling backups in a virtual environment does require some planning though. The reason for this is you don't want to put too much resource strain on hosts and shared storage datastores that may negatively impact VMs that need resources to function properly. How much resource strain you will encounter depends on the method used to back up the VMs. When backing up VMs using agents inside the guest OS, the resource usage on the host will be the greatest. If VMs are being backed up using image-level backups, the resource usage on the host will be lessened. When leveraging the new features in the vStorage APIs, the resource usage will be even further reduced.

Depending on the backup method you use you should take care to make sure you do not back up too many VMs simultaneously on the same host and shared datastores. When using agent based or over the network backups you can put too many resource constraints on the host. When using direct to datastore backup methods you can put too many resource constraints on shared datastores. This can cause your VMs to be resource starved while backups are occurring and VM performance can suffer greatly as a result. Doing some basic planning of backup schedules can help ensure that backups are staggered to not put too much burden on a single resource point at

the same time. Also be aware that snapshots are taken and deleted on all VMs being backed when doing image-level backups, so this can also cause additional resource constraints in the environment, particularly with storage.

Leveraging the vStorage API methods to streamline backups and backing up to disk targets mean everything moves at a very fast pace. Perhaps one of the biggest bottlenecks can be the backup server that is handling all the backup coordination tasks. When backup operations are running, there is more to it than just copying data from point A to point B. Backup operations also do a lot of CPU processing to determine what data to back up and what not to back up, deduplicate data, and also compress data that is written to the target. Having an undersized server especially in the CPU area can greatly reduce backup performance. Therefore, it is important to not skimp on resources for your backup server; running on a physical server or VM with at least 4 CPUs is necessary to get the best backup performance possible.

Virtualization advantages

Because of its unique and flexible architecture, virtualization provides many advantages over traditional physical servers when it comes to backup and recovery. These advantages can help companies save time and money as well as provide new capabilities to the data center. Listed below is a summary of the advantages that virtualization can provide related to backup and recovery:

- **Easier bare-metal restores**—Because VMs are all presented with the same virtual hardware, bare-metal recovery is much easier as VMs will always see the same virtual hardware regardless of the physical hardware that the host is using.
- **Easier backup verification**—Backed up VMs can easily be restored to any host or workstation to be verified without disrupting the original running VMs. SureBackup Recovery Verification simplifies and automates this and allows you to do this directly from the backup target datastore.
- **Backup usability**—With Veeam vPower you can actually put your backups to work instead of leaving them sit there taking up space until a restore is needed. Backed up VMs can be brought online for testing and troubleshooting purposes without affecting the original VMs.
- **Multiple backups methods**—Traditional servers typically need to do separate image-level for bare metal restore capabilities in addition to standard file-level backups. Virtualization only needs to back up once at the image level but can provide both image-level and file-level restores from that one backup.
- **Agentless**—Backup agents do not need to be installed on each server to be backed up as the backup is performed at the virtualization layer without going through the guest OS.
- **Resource-free backups**—The VM's virtual disk is accessed directly when backed up without involving the VM's guest OS, so no resources are tied up on the VM while the backup is running.
- **Network-free backups**—The backup server can connect directly to the VM's virtual disk using the storage network, so data being backed up does not need to be dragged across the network where it can impact other servers communicating on the network.
- **Easy snapshots**—VM snapshots provide an easy short-term backup solution for VMs and also enable easy image-level backups by freezing the VM's disk during backups.
- **Easy application-item recovery**—With Veeam vPower, application-level items such as database records and emails can easily be recovered from an isolated environment and copied as needed back to the original environment.

SUMMARY

In this chapter we compared traditional backup methods to those used in virtual environments. Implementing virtualization requires a different mindset and techniques as traditional methodologies are not as efficient and practical in a virtual environment. Virtualization architectures can be a big enabler for backup and recovery in the data center and open the door for new and more efficient ways of doing things. There are some people that continue doing backup and recovery the way they always have after implementing virtualization despite it being less efficient. It's important to leverage the strengths of virtualization and change your methodologies so you can take advantage of the great features that virtualization has to offer to improve your backup and recovery methods. After all, backup and recovery are never easy, so why make it harder than it needs to be.

ABOUT THE AUTHOR

Eric Siebert is an IT industry veteran, author and blogger with more than 25 years of experience, most recently specializing in server administration and virtualization. He is a very active member of the VMware VMTN support forums, where he's attained the elite Guru status by helping others with their virtualization-related challenges.

Siebert has published books including his most recent, **"Maximum vSphere"** from Pearson Publishing, and has authored training videos in the Train Signal series. He also maintains his own VMware VI3 information website, vSphere-land, and is a regular blogger and feature article contributor on TechTarget's SearchServerVirtualization and SearchVMware websites. Siebert has presented at VMworld in 2008 & 2010 and has been recognized as a vExpert by VMware in 2009 & 2010.

ABOUT THE SPONSOR

Veeam Software, a premier-level VMware Technology Alliance Partner, develops innovative software to manage VMware vSphere. Veeam vPower™ provides advanced Virtualization-Powered Data Protection™ and is the underlying technology in Veeam Backup & Replication™, the #1 VMware backup solution. Veeam ONE™ provides a single solution to optimize the performance, configuration and utilization of VMware environments and includes: Veeam Reporter™ - VMware capacity planning, change management, and reporting and chargeback; Veeam Business View™ - VMware business service management and categorization; and a choice of VMware monitoring options including the nworks Management Pack™ - VMware management in Microsoft System Center, the nworks Smart Plug-in™ - VMware management in HP Operations Manager, and Veeam Monitor™ - framework-independent VMware monitoring. Learn more about Veeam Software by visiting www.veeam.com.