TINTRÍ

Managing VM Data with Tintri

Powerful VM and application acceleration with Tintri VMstore™, SnapVM™, CloneVM™ and ReplicateVM™

By John Phillips

TECHNICAL WHITE PAPER

PTS DATA CENTER SOLUTIONS
For More Information:
(866) 787-3271
Sales@PTSdcs.com

Table of Contents

Intended Audience 1

Introduction 1

VM Snapshots 101 2

The Basics 2

Snapshots and Data 3

Creating snapshots of VMs 5

Differences between Software and Hardware snapshots 6

What are you Snapshotting? 7

What do Hardware Based Snapshots Capture? 7

Enter Virtualization 7

Tintri VMstore Overview 10

Tintri VMstore Snapshots 10

VM Cloning with Tintri VMstore 11

Common VM Cloning Scenarios 13

The Tintri VAAI Provider for vSphere 14

Tintri ReplicateVM 15

Configuring Replication Paths 15

Per-path Throttling 16

Replication Starts with Snapshots 16

Enabling Replication for a VM 17

Customizing the ReplicateVM Settings for a VM 18

ReplicateVM per-VM Updates 19

Replication Speed and Efficiency 19

Remote Cloning with ReplicateVM 21

Applications 22

Summary 23

Intended Audience

This paper is for readers who want to gain a better understanding of storage, snapshots, cloning and replication as it applies to virtual infrastructures, and virtual machines (VMs). A basic understanding of virtualization, networking, and storage is helpful to grasp the concepts covered in this paper. High-level technical readers should expect to learn about Tintri VMstore's working implementation of snapshots, cloning and replication and its applicability to common virtualization applications and use cases.

Introduction

Decades have passed since the inception of snapshots in open systems computing. These days, snapshots are "the norm" and the first line of defense for backup, recovery, data protection and disaster recovery scenarios. Today, there are entire ecosystems of application and data management frameworks built around snapshots.

In the early 1990's, "the snapshot star" began to rise by providing quick recovery points for user data files, documents, etc. Snapshots in various forms were progressively leveraged and integrated into operating systems, database and message applications, and into data protection and disaster recovery solutions.

No longer are snapshots merely "preferred" options. Streaming backups have been relegated to 2nd or 3rd tier data protection applications, and there are many enterprise applications that do not directly support streaming backups any longer.[1]

Snapshots are a pillar supporting the flexibility and efficiency of virtualized infrastructures, providing space efficiencies, VM cloning, rapid deployment scenarios, and checkpoints for backup and data protection applications.

Virtual Desktop Infrastructure (VDI) and End User Computing (EUC)

- Snapshots preserve the core images that comprise the virtual machines and applications, enabling administrators to define, pre-test, qualify operating system images, and settings, as well as their applications and features, and then deploy, manage and update them centrally

Server and Application Virtualization

- In addition to leveraging snapshots in the same way as VDI and EUC environments do, virtualized server applications are often mission critical "Tier1" applications, containing critical data and delivering essential services to users all over the globe

This paper will discuss snapshots and the role of snapshots in supporting these applications in a virtualized environment.

[1] The Microsoft Exchange team announced the elimination of streaming backup support in June 2009. The Volume Shadow Copy Services (VSS) snapshot framework is the entry point for all backup applications. Starting with Exchange 2010, APIs for streaming backups do not ship in the Exchange SDK. http://msdn.microsoft.com/en-us/library/dd877018(EXCHG.140).aspx.

VM Snapshots 101

The Basics

A popular analogy used to describe a "snapshot" within the realm of information technology is a camera snapshot, which captures an image at the point-in-time when the shutter button is depressed.

In our case, the image is not a picture; it is the state of a VM and its constituent files at the point-in-time of a snapshot.

Figure 1: Each individual VM has a unique identity and is comprised of many constituent elements

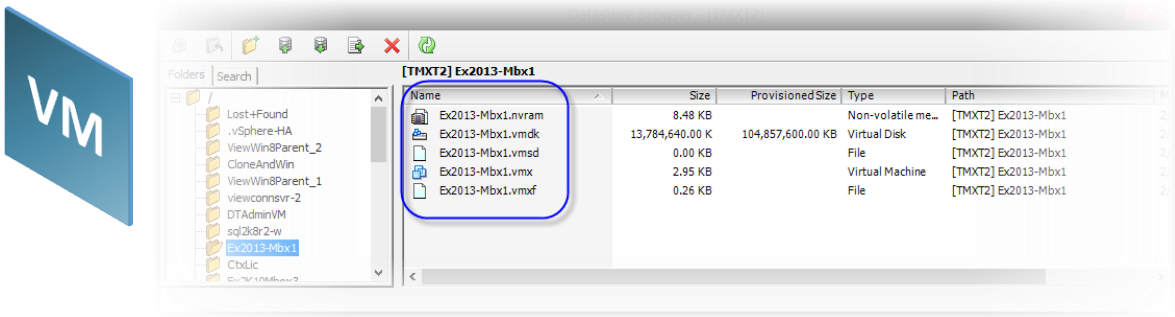
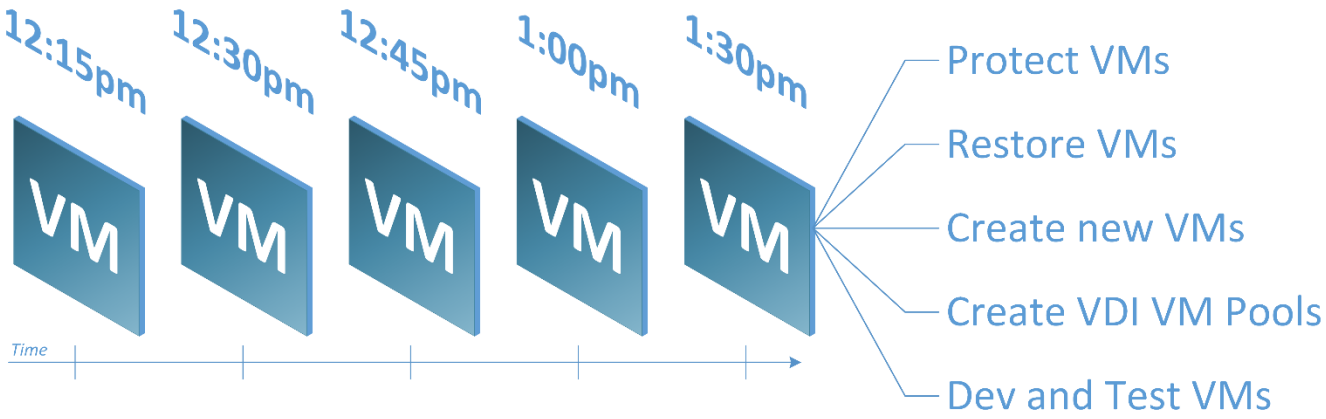


Figure 2: Scheduled snapshots of VMs and some of their common uses



Creating point-in-time snapshots of VMs provides a versioning of sorts, where you can then obtain access to a VM and its application data at the exact date and time of a given snapshot.

Snapshots provide a quick way to restore a VM and its application data back to the point-in-time captured by a VM snapshot, and to create *new* space-efficient VMs[2], which derive their initial state from a VM snapshot. The rest of this section will detail how snapshots help power server and desktop virtualization from the deployment of new VMs, to fulfilling the data protection and recovery requirements of VMs across data centers.

[2] This is otherwise referred to as "cloning" VMs

Snapshots and Data

Data or *recognizable patterns* of binary ones and zeros, "lives" within file system data-blocks ("blocks"). Blocks, when linked together by a file system constitute files. The bigger the file, the more blocks it will take to store all of a file's data. The files that make up a vSphere VM, such as one or more virtual hard disk files (.vmdk), as well as its configuration files (e.g. .vmx), and other files collectively comprise what we know as an (individual) VM.

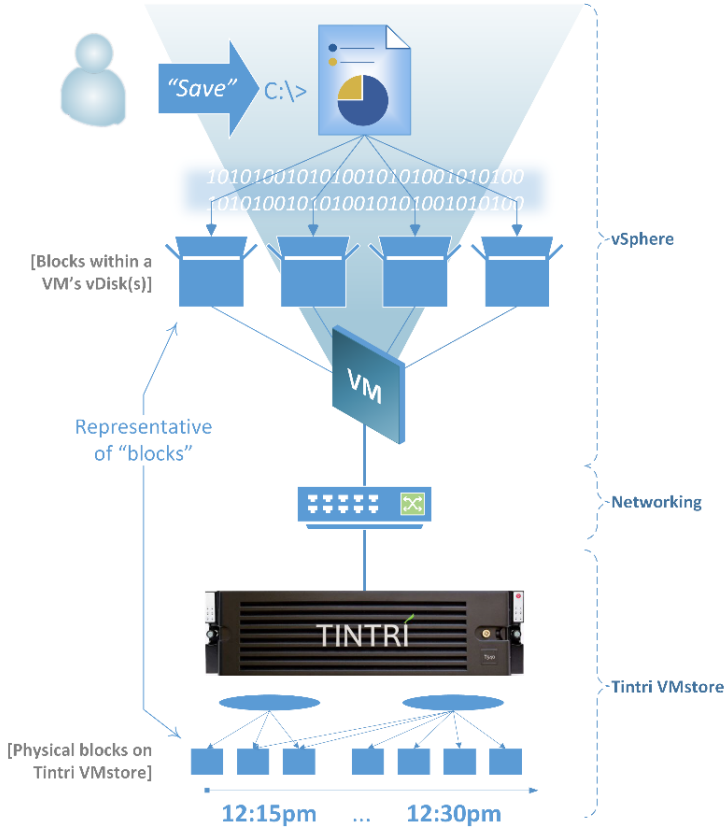
What goes into a data block, and how does it get there?

Each block "contains" data (see above) stored within a VM's files (see Figure 1).

When you save a file, you are instructing an application to save, or *persist* what you are working on. What actually happens within a VM is that the file's data is stored in one or more blocks in the VM's file system. Since we are talking about a *virtual machine*, the blocks in this case are stored in the VM's virtual hard disk files, or "vDisks" (e.g. .vmdk files). Those vDisks live on Tintri VMstore. Therefore, when saving changes to a file, the writes are to a VM's vDisk(s), which exist on Tintri VMstore.

Tintri VMstore keeps track of the changes to all of its VMs' vDisks within its flash-based file system and preserves them in snapshots according to an administrator's operations and configured snapshot schedules.

Figure 3: Data flow from an application to a VM's "C: drive" (vDisk), which actually resides on Tintri VMstore



Snapshots of a different kind

Temporarily imagine that when driving to and from work every day you passed a site over the course of a year where a building was under construction, and stopped to "snap" a picture of the progress each day.

The pictures over time would reflect the changes to the building and site along the way. People, equipment and temporary buildings would come and go. Each camera snapshot would be unique, recording different points in time during the construction process.

NOW, ADJUST YOUR VM VISION AND IMAGINE TAKING SNAPSHOTS (NOT PICTURES!) OF A VM INSTEAD OF A BUILDING.

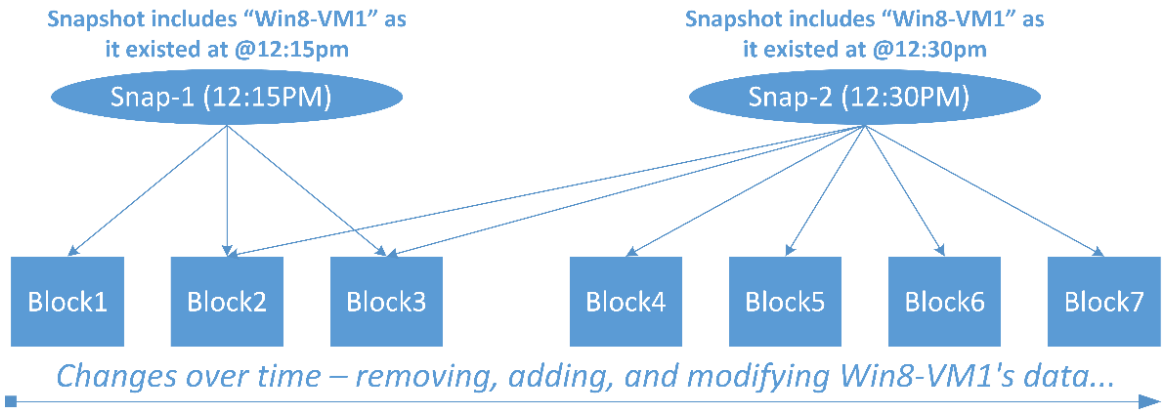
Consider the following scenario using an assumed VM named "Win8-VM1"

- The first snapshot of Win8-VM1 is created at 12:15pm
- The next snapshot is created fifteen minutes later, at 12:30pm

Snapshot space efficiencies (generalized)

In Figure 4, note that the snapshots named "Snap-1" and "Snap-2" *both* have pointers to Block2 and Block3.

Figure 4: Basic snapshot "block pointer" diagram for 'Win8-VM1'



When creating Snap-2, it would be inefficient to make additional copies of Block 2 and Block3 just to include them in Snap-2. Instead, Snap-2, like Snap-1, records *pointers* to the locations of the existing blocks 2 and 3.

A small example: Imagine that sometime between Snap-1 and Snap-2, at 12:20pm, that you open a spreadsheet, replace a cell value, and then save those changes. Let us declare that the "old" cell value is in Block1, safely protected by Snap-1[3]. Assume that when you save the spreadsheet, the *new* cell-value lands in Block4.

Notice in Figure 4 that Snap-2 does NOT include a pointer to Block1. Why? The 2-part answer:

- Writes always occur to *new* blocks, as opposed to returning to Block1 to modify its data. This technique accelerates I/O operations and allows the Tintri file system to quickly record pointers to snapshotted blocks
- The old value (stored in Block1) is not germane to Snap-2. Consequently, Snap-2 does not record a reference to Block1. The prior version of the VM and its spreadsheet as captured at 12:15pm will remain available through Snap-1

IN THIS EXAMPLE, IF YOU WERE TO "SEE" WIN8-VM1 THROUGH HD GLASSES IN EACH OF THESE TWO SNAPSHOTS, YOU WOULD "SEE" WIN8-VM1 AS IT WAS CAPTURED AT 12:15 IN SNAP-1 (A "MAP" CONSISTING OF BLOCKS 1-3), AND AT 12:30 IN SNAP-2 (A "MAP" CONSISTING OF BLOCKS, 2, 3, 4, 5, 6 AND 7).

[3] By design, it is not possible to modify a block referenced by snapshots. Block1 is protected by the Tintri file system from overwrites or deletion so long as any VM or snapshot has a reference to it. Only when a given block's reference count drops to zero (e.g. no VM or snapshots "point to it"), will Block-1's location be marked as "free", allowing new writes to it.

Creating snapshots of VMs

To create a snapshot of a VM, all of its files are collectively "snapped" at the same point in time, as quickly and as efficiently as possible.

First, know that there are two common ways to approach the *creation* of snapshots for VMs:

- **Crash-consistent**: Create a snapshot of the VM without taking extra measures to coordinate the snapshot with the VM's guest operating system and its applications
- **VM-consistent**: Create a snapshot while also taking further steps to coordinate the construction of the snapshot with the hypervisor (e.g. vSphere ESX/ESXi), and the guest operating system and its applications

While both crash-consistent and vm-consistent snapshots can be equally utilized by an administrator, the VM's state captured within a crash-consistent snapshot is considered "unknown". That does not mean it is inaccurate or unusable, just that the VM's condition was captured "on-the-fly."

One way to help understand "crash-consistent" snapshots

Imagine that rather than "shutting down" your laptop, you just powered it off while it was running. When powered on, the operating system will have not recorded a normal shut down. Consequently, it will examine its logs and settings, taking any necessary recovery steps if needed.

When restoring a VM into operation from a "crash-consistent" snapshot, the VM's guest operating system will take steps similar to the ones noted above in the laptop example.

More on VM-consistent snapshots

When a VM is actively performing work, one of its applications may have data "in-flight", i.e. not yet written to its vDisk(s), at various points in time. VM-consistent snapshots utilize software (see below) within a VM's guest operating system to synchronize a "flush" of application data writes from the VM's memory to its vDisk(s) when creating a VM-consistent snapshot. Consequently, the VM commits any volatile data residing within its memory to its vDisk(s) and therefore the VM snapshot. Hence the term, "VM-consistent."

In vSphere environments, VMware Tools™, installed within each VM's guest operating system, allows the hypervisor to "reach into a VM", to interact with specific aspects of the VM's guest operating system, such as the "freeze/thaw" operations associated with Microsoft Volume Shadow Copy Services (VSS) for Windows. VM-consistent operations with pre-defined scripts are also part of VMware Tools for Linux guest VMs.

There are also other, application specific (VSS) components for Microsoft Exchange, Microsoft SQL Server, and many others, which provide specialized functionality and extended integration capabilities with various third party tools. Those components work together with VMware Tools to facilitate backups and data protection, data management operations, etc. While the applications vary, application specific components play a pivotal role in the VM-consistent snapshots required for advanced database and server VMs.

Differences between Software and Hardware snapshots

For the purposes of this document, software based snapshots are implemented by a hypervisor server, such as vSphere ESXi.

Hardware based snapshots are features implemented in storage systems with capabilities that extend beyond the basic functionality of Direct Attached Storage (DAS) or "Just a Bunch of Disks" (JBOD) storage.

Per-VM, software based snapshots

Software based snapshots such as vSphere's native or "built-in" snapshots are typically inclusive to the hypervisor server software (i.e. "free") and exceedingly easy to use. Software snapshots are also VM-specific.

VMware vSphere ESX/ESXi snapshots implement a series of related virtual hard disk files, or "snapshot chains"[4] to manage and track a VM's snapshots. Unfortunately, the practical applications of software snapshots are limited due to the extraordinary and widespread I/O activity associated with a VM's disk chain. The penalties even on fast (i.e. Flash/SSD) drives can be serious and costly given the amount of flash/SSD storage space consumed by the disk chain's snapshot files.

VMware's best practices recommend no more than three (software) snapshots at a time of a VM, and you should avoid using and retaining them past 24-72 hours to avoid significant performance complications.

Hardware based snapshots

The common attribute that all storage based snapshots share is that they allow hypervisor host servers to delegate the heavy lifting of creating and managing the snapshots to the storage, freeing host server resources.

WHAT DEFINES EACH STORAGE SYSTEMS' HARDWARE BASED SNAPSHOTS?

Hardware based snapshots rely solely on the individual characteristics of the implementing storage system.

While hardware based snapshots are much faster than software based snapshots, they are not necessarily easier to use. Traditionally structured shared storage systems, even those that employ flash/SDD drives can simulate "per-VM" snapshots with specialized software or plugins. Unfortunately, the deployment and management complexities, and the limitations of the underlying hardware are unavoidable.

Tintri VMstore is the only storage appliance that actually creates snapshots of VMs, rather than creating snapshots of arbitrary storage configurations where a VM's files happen to be located. There is a stark contrast between these two approaches. The entire lifecycle, from the acquisition and deployment costs, to the manageability and TCO of a virtualization storage platform necessitates a solid understanding of how these choices affect your organization.

[4] http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1015180

What are you Snapshotting?

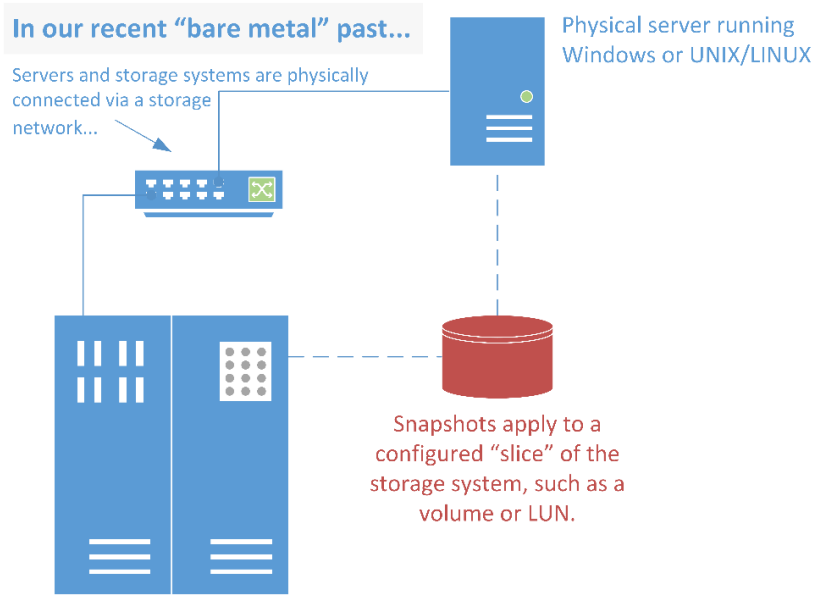
Storage hardware accelerated snapshot functionality was available long before virtualization was a mainstream technology.

Hardware based snapshots were immediately applied to virtualization by storage vendors to help overcome the performance limitations of software based snapshots. However, while the seemingly universal adaptability of legacy hardware snapshots is significant, so also are the entanglements and limitations of their bonds with their respective storage technologies.

What do Hardware Based Snapshots Capture?

The endearing answer is that it depends. Fundamentally, the design of general purpose shared storage and SAN systems serves "bare metal" or physical servers and PCs.

Figure 5: Snapshots with "bare metal" physical servers, before virtualization



Prior to virtualization, the applicability of hardware snapshots was favorable with physical servers and their applications.

Snapshots accelerated the rate at which servers and their application data could be 'snapped' for quick recovery operations.

Enter Virtualization

Hypervisor server software runs on "bare metal"[5], or in other words, on the physical server hardware. Examples of hypervisors are VMware vSphere (ESX/ESXi), Microsoft Hyper-V, and Citrix XenServer. For the examples in this paper, we are concentrating on VMware vSphere.

One of VMware's objectives for the vSphere hypervisor (ESXi) is to reduce the amount of datastores that customers must manage. It is a clear objective evidenced by the evolution of the VMFS file system.

---

[5] Experimental nested virtualization notwithstanding

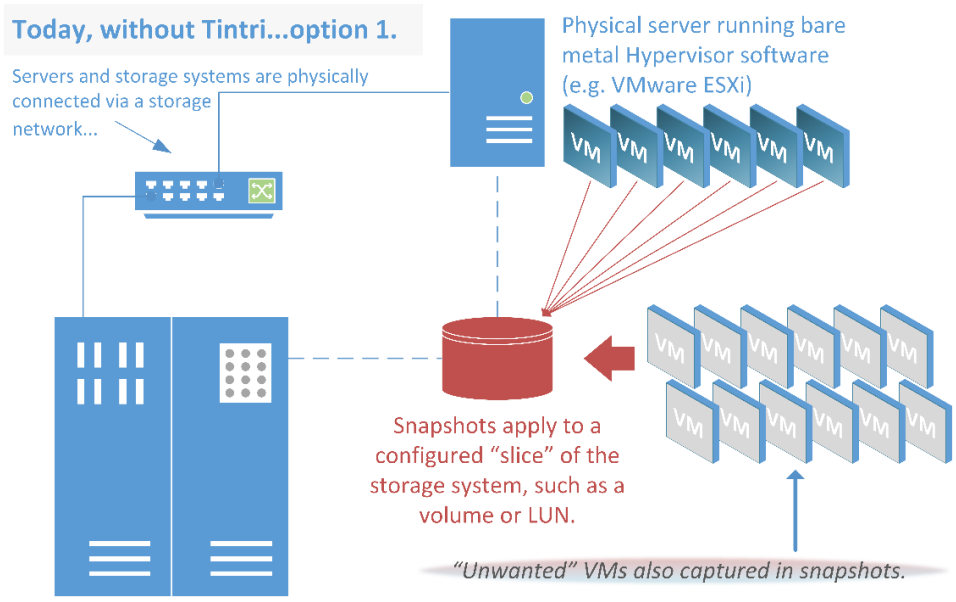
With traditionally structured shared storage and modern versions of ESXi, you can provision larger datastores using SAN technologies than ever before due to improvements in VMFS. The tradeoff is that when it comes to hardware-accelerated snapshots using LUNs as datastores, you must create a snapshot of the entire datastore and all of the VMs contained within a given datastore.

Therefore, you must choose between tradeoffs:

- Provision a large number of datastores, *one for each VM* (via storage system operations) so that you can target individual VMs for snapshots (more specifically, target their containing datastores)
- Place several VMs in fewer datastores to simplify management, sacrificing any and all semblances of "per-VM" snapshot fidelity

Moreover, it is a best practice to limit the VMs or vDisks per LUN in SAN environments due to factors such as per-LUN or target queue depths, protocols, switch fabrics, etc. The impositions associated with these storage related factors dictates VM and application management. This class of constraints is becoming increasingly unacceptable in the software-defined datacenter.

Figure 6: Snapshots designed for physical servers do not adapt well to virtualization



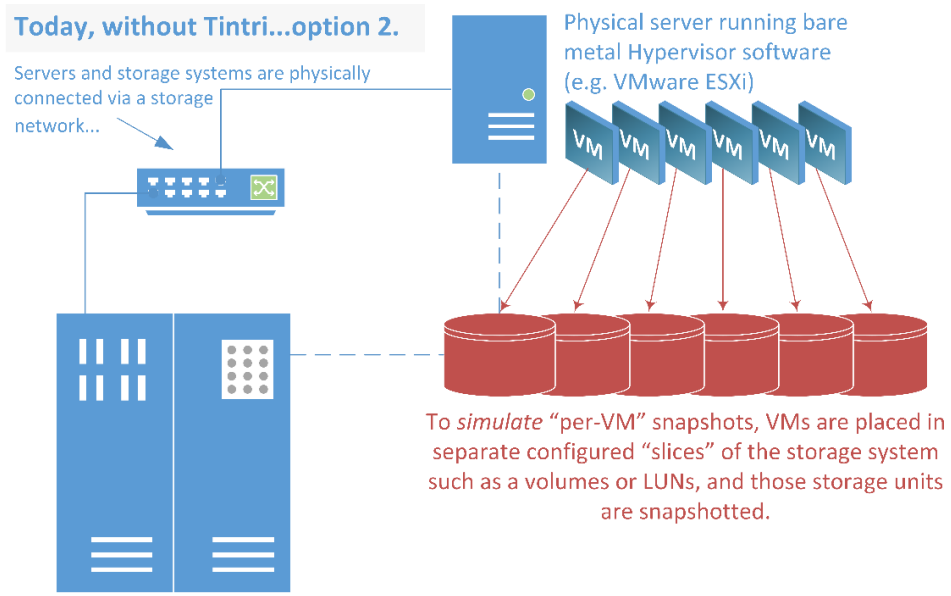
A snapshot of a LUN or volume captures all of its VMs, irrespective of which VM or VMs are of interest.

Snapshots are of the LUNs, volumes, or other storage system units, not of the individual VMs. Consequently, complex storage configurations are unavoidable when using traditionally structured, shared storage.

The paradigm shift from bare metal to virtualization has overstretched legacy storage technologies. Newly marketed flash-based products, which have incomprehensibly followed in the long treaded structural footsteps of traditional shared storage designs, embody these same limitations. Only Tintri VMstore is VM-aware.

Provision numerous datastores in order to target individual VMs for snapshots

Figure 7: Provisioning separate datastores for each VM to enable per-VM snapshots



There is no value in hindsight

To meet your criteria for creating snapshots of an individual VM when using traditionally structured storage, you must first configure the storage to match the way you want to manage the VMs. Then, forcibly locate the VMs accordingly. The unavoidable rigidity of traditionally structured storage conflicts with the software-defined datacenter and creates undue pressure on storage and virtualization managers by severely limiting the agility and resource utilization benefits intrinsic to virtualization.

The coupling of snapshot implementations to the internal structures of traditional storage designs creates a circular dependency, in which VM deployments must meet complex, vendor-specific storage placement requirements in order to have their own operational requirements satisfied. Since these storage architectures do not intrinsically recognize a VM, cannot interact with a VM, and are only capable of creating snapshots of the storage provisions where administrators locate their VMs, they are outdated and poorly adapted to the virtualization experience, forcing virtualization deployments into the bare metal methodologies of the last two decades.

Tintri VMstore has the advantage of being purpose built from the ground up for flash (SSD) and virtualization by pioneers in storage and virtualization. Each Tintri VMstore is a single datastore with individual, per-VM capabilities woven into its deepest depths and features. This innovative approach leapfrogs traditional storage vendors, which must develop, promote, and support bolt-on software and plugins in an effort to reduce the gaps between virtualization and their storage platforms and snapshots.

Tintri VMstore Overview

Traditionally structured shared storage systems, even newer products that contain flash storage, still suffer from the same complexity and storage centric management paradigm that has been around for over 20 years. Once installed in a rack, Tintri VMstore is ready for VMs in minutes. The VM centricity in the heart of Tintri VMstore elevates the VMs themselves to first class.

Tintri VMstore is distinctly different from other storage systems that cannot actionably distinguish between a VM and conventional documents and image files. Only Tintri VMstore sees VMs for what they are, and intrinsically "knows" vital details about each VM, and to which VM and vDisk every I/O operation belongs. True VM-awareness and the ability to exact snapshot, cloning and replication operations on a per-VM basis are unique to Tintri VMstore.

Tintri VMstore features such as SnapVM, CloneVM, and ReplicateVM promote the protection, management and replication of individual VMs from 1,000 VMs in 3U, to an unlimited number of VMs across multiple Tintri VMstore appliances. ReplicateVM enables individual *per-VM* replication over local, fast networks or wide area network (WAN) leased lines.

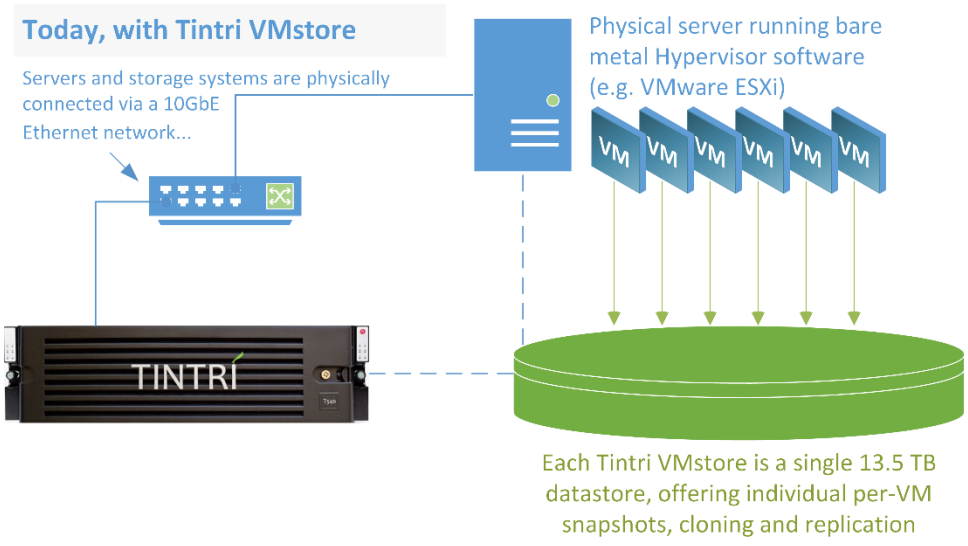
For an overview of Tintri VMstore, please see the following VMstore Overview document on Tintri's website: http://go.tintri.com/vmstore-whitepaper

Tintri VMstore Snapshots

The snapshot and cloning features built into Tintri VMstore apply directly to virtual machines, rather than to arbitrary units of storage (e.g. LUNs, volumes) where administrators must place their VMs.

Tintri VMstore offers the best of both worlds. Speed, efficiency and power, with individual per-VM fidelity.

Figure 8: Tintri VMstore has high-fidelity per-VM snapshot, cloning and replication features while eliminating the deployment, configuration and management complexity necessitated by traditionally structured storage systems



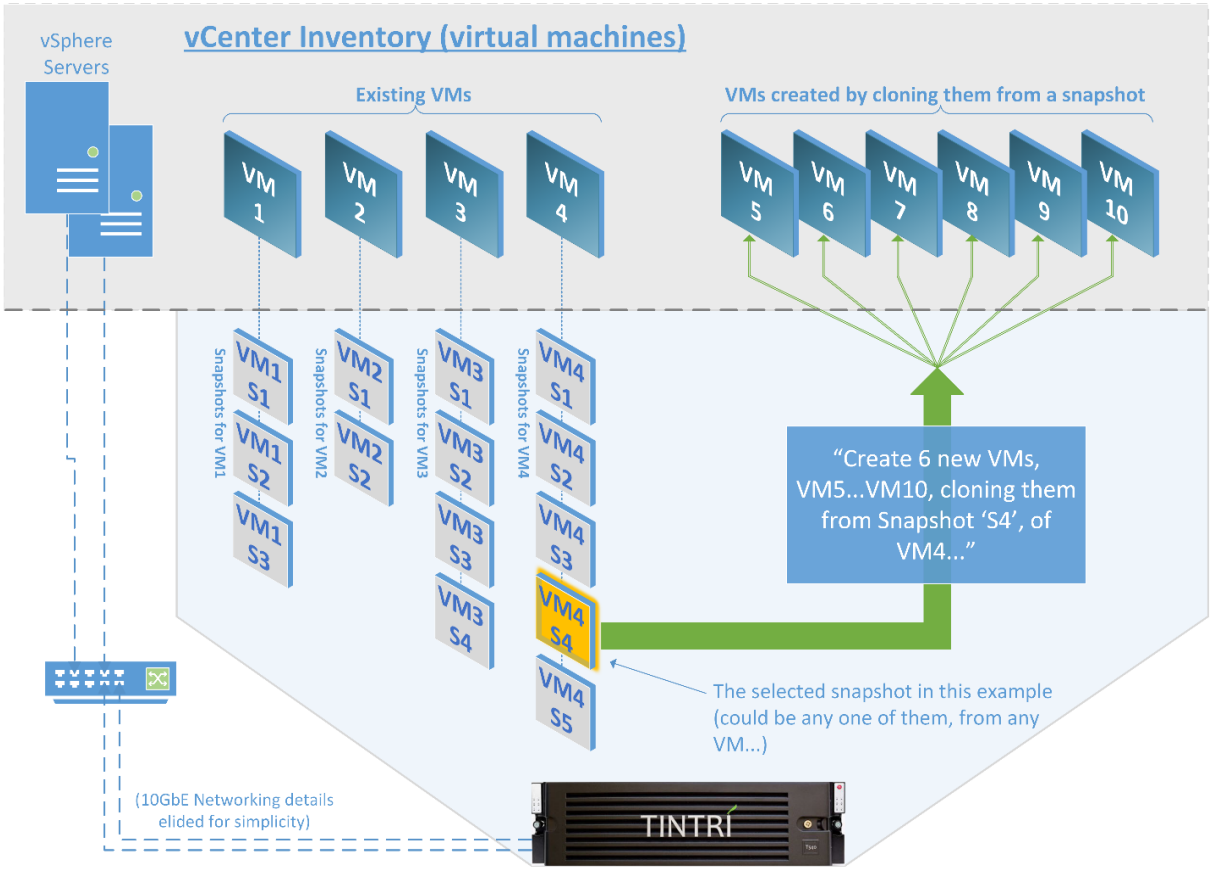
VM Cloning with Tintri VMstore

The term “cloning” suggests that you are making identical copies of VMs, but that is not the whole picture.

> **CLONING IS THE PROCESS BY WHICH YOU CREATE NEW VMS, USING THE VM CAPTURED IN THE SNAPSHOT AS THE BASIS OR STARTING POINT FOR THOSE VMS.**

Clones derived from a given snapshot share the same origin, but once created they operate as independent VMs with their own identities.[6]

Figure 9: Creating new VMs by cloning them from a selected snapshot



In the simplified example above, the cloning process from snapshot “S4”, of VM 4 results in six (top right) of the 10 VMs shown in inventory. The new VMs are customizable at first boot, either through manual or automated means, with their own individual names, IP addresses, software licenses, etc. Operations of this kind apply to both desktop VMs (i.e. VDI) and server VMs.

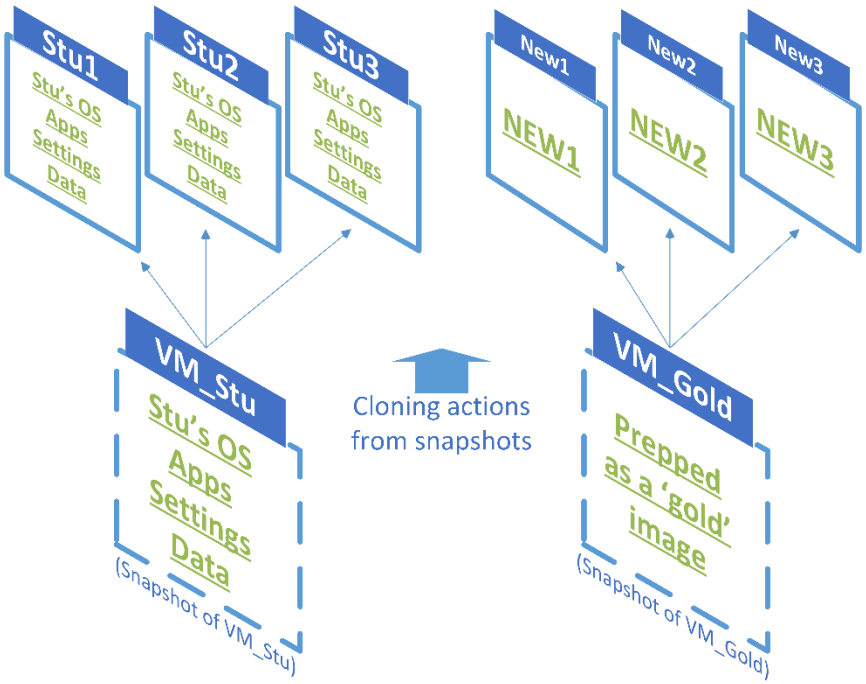
[6] Restore/recovery scenarios suggest that you are “restoring the same VM”, which is partly true. The following page provides additional information and clarification.

Question: "If cloning operations create 'new' VMs, how also does cloning enable 'recovery' operations?"

Think of a VM as a logical object, generally consisting of a container with properties, and "contents." Similarly, shells contain shellfish. The "shell" in this example is the VMware VM and the "shellfish" is the *Guest OS (e.g. Windows, Linux)* and its applications and data.

The following is always true, when cloning a VM with Tintri VMstore: You are always creating a new *VM* that *contains* the same data as VM in the snapshot.

Figure 10: Creating VMs from snapshots (cloning)



In the example above, if something were to go horribly wrong with Stu's VM, he could delete his "old" failed VM, create a new VM from the snapshot of his VM (bottom left), give it the same VM name, and he is back in business. This is an example of a straightforward "restore", or "recovery" operation. Note: More than one VM may not seem useful for Stu.

Assume that you decidedly *prepare* "VM_Gold" to use as the starting point for creating numerous VMs with similar settings. Then, you create a snapshot to use as the basis for creating new VMs via cloning. This makes it tremendously easy to deploy additional VMs that have regular configurations and settings, rather than having to develop and produce each VM individually.

THE MOST NOTABLE POINTS TO CONCEPTUALIZE AT THIS STAGE:

- The state of a VM before it is snapshotted determines what is captured in the snapshot
- Cloning operations use the captured point-in-time state within VM snapshots as their base of inception
- Your achievement following cloning operations is determined by the VM snapshot, and what you do with the new VM once it has been created, added to vCenter inventory, and then powered on

Common VM Cloning Scenarios

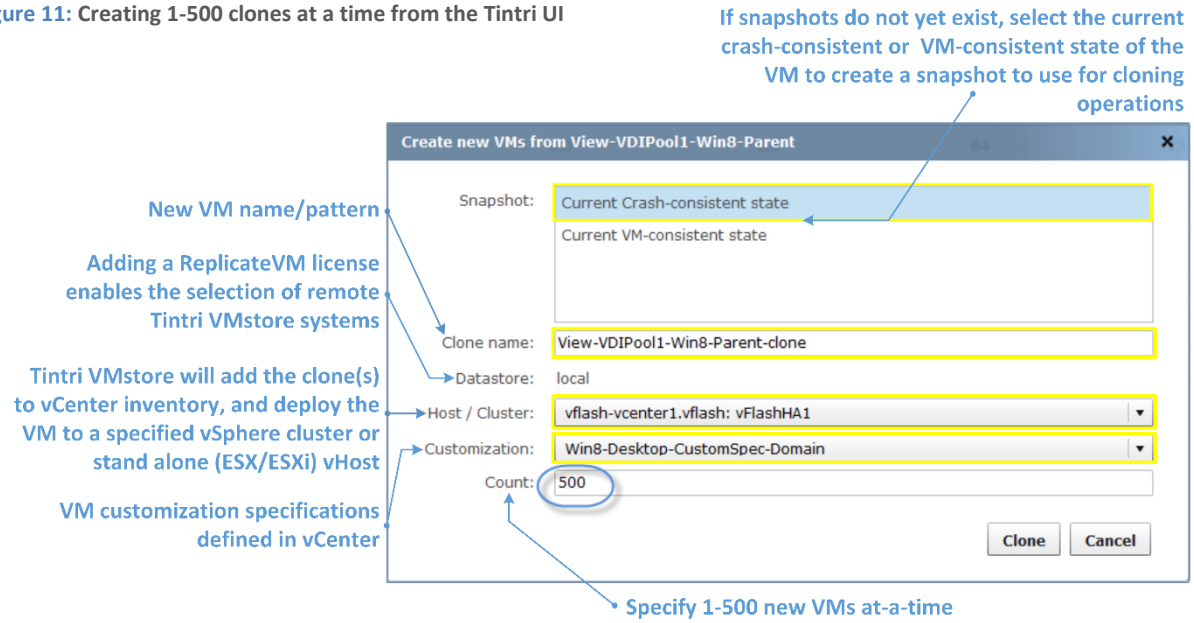
Tintri VMstore is capable of cloning VMs extraordinarily fast. Tintri VMstore excels in rapid server deployment scenarios such as spinning up hundreds of stateless (temporary) server VMs in response to high activity spikes, to deploying VDI desktop pools or catalogs of hundreds or thousands of VMs with VMware Horizon View or Citrix XenDesktop.

A partial listing of common applications where VM cloning operations are used:

- Creating VMware Horizon View Desktop Pools of VMs
- Creating XenDesktop Catalogs of VMs
- Dynamic environments, such as web sites, and software and database development and testing
- Recovering server VMs, such as an Exchange, SQL, Oracle, or SAP VMs from a snapshot
- Data restoration scenarios, where a VM's snapshots provide access to its *data* at the point-in-time a given snapshot was created, without disturbing its real time operational state
- Anywhere that you already use vCenter clones or are considering using vCenter clones

The highlighted "gold" VM snapshot ('S4') shown in Figure 9 could be the selected snapshot of any VM you want, for any situation where cloning is useful. The number of new VMs that Tintri VMstore creates "per-command", ranges from 1-500 VMs at a time.

Figure 11: Creating 1-500 clones at a time from the Tintri UI



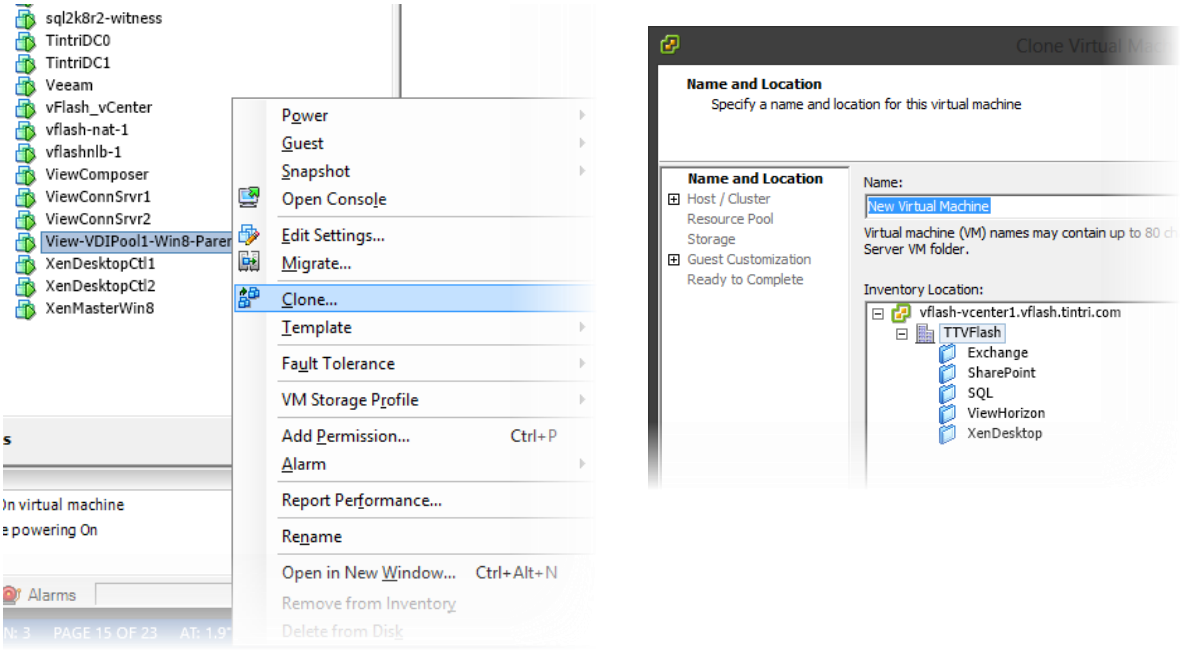
When cloning VMs from the Tintri VMstore UI, you can create clones from existing snapshots, or use the current state, where Tintri VMstore automatically creates a snapshot when you press the "clone" button.

You can use vCenter customization specifications (Tintri VMstore retrieves them from vCenter), and you can choose a specific vSphere cluster or vHost (ESX/ESXi server) to which you want to register and deploy the VM(s). Tintri VMstore automatically adds cloned VMs to your vCenter inventory for immediate use.

The Tintri VAAI Provider for vSphere

Installed on each vSphere server, the Tintri vStorage APIs for Array Integration (VAAI) plugin ensures that every VM in vCenter can leverage the fast and powerful, space-efficient cloning capability of Tintri VMstore.

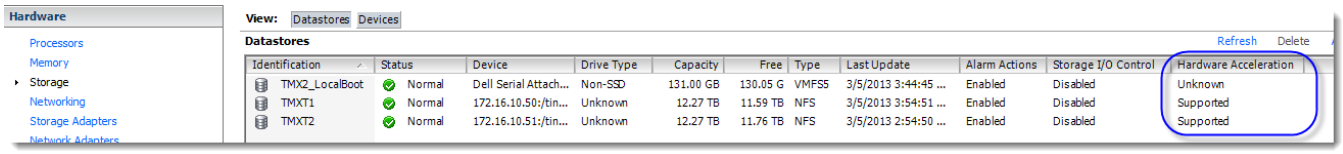
Using the vSphere client, administrators can right-click on a VM, and select “Clone VM” to start the Clone Virtual Machine wizard. The vSphere cloning operation leverages Tintri’s VAAI provider, which then delegates the cloning process to Tintri VMstore



How does VAAI work?

When vSphere receives a request to clone a VM, it inspects the VM’s datastore and checks to see if the datastore supports hardware acceleration. The Tintri VAAI provider plugin in each vSphere server serves as the intermediary between vSphere and Tintri VMstore. vSphere delegates the cloning operation to Tintri VMstore, and then communicates to vSphere through VAAI that the operation is complete.

Figure 12: Viewing datastores in the VMware vSphere Client



Using Windows PowerShell and VMware PowerCLI to automate the provisioning of VMs

Scripting VM cloning operations with PowerShell/PowerCLI or any other method that calls the vSphere API’s also leverages the Tintri VAAI provider, because vSphere interacts with datastores the same way regardless of whether or not the command arrives from the vSphere client, PowerCLI, via a vSphere API call, etc.

Tintri ReplicateVM

Introduced with TintriOS release 2.0, Tintri ReplicateVM extends the snapshot[7] and cloning capabilities of Tintri VMstore to appliances in a single data center or across multiple data centers.

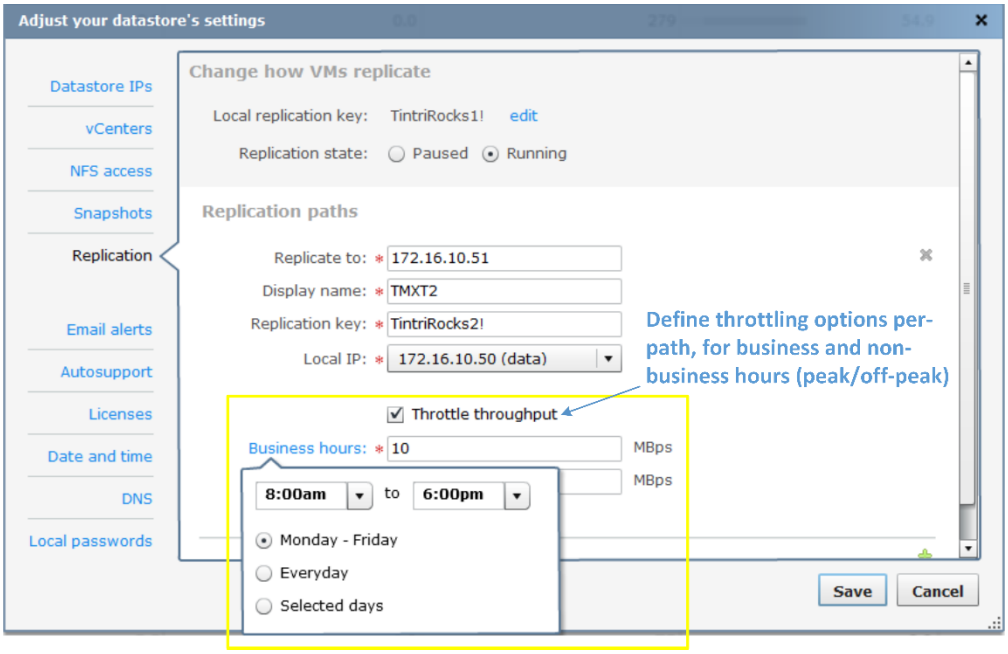
Like cloning, ReplicateVM also uses the point-in-time snapshots of a given VM as the measure of replication. For example, a snapshot schedule that creates snapshots every 15 minutes determines "what is replicated", every 15 minutes. Therefore, for a VM protected with replication, the snapshot schedule and the replication schedule are one in the same.

Configuring Replication Paths

Tintri VMstore currently supports 16 paths per system. Setting up replication paths between Tintri VMstore systems is straightforward. On each Tintri VMstore:

- Specify the network path to your destination Tintri VMstore (Figure 13)
- Right-click on each VM and select the "Protect" (snapshot scheduling) options for a VM (Figure 15)
- Specify up to 3 snapshot schedules, for each individual VM (Figure 17)
- Check the "Protect by replicating snapshots" checkbox (Figure 16, Figure 17)

Figure 13: Specifying a path and options for ReplicateVM between Tintri VMstore appliances



In addition to specifying the host name and IP address for a path, Tintri VMstore requires administratively defined *replication keys* when setting up paths. You can create and use more "cryptic" looking keys, but entering keys that are easier to remember and type are easier. Each administrator will decide what works best for their organization.

[7] Tip: A basic understanding of snapshots makes understanding cloning and the replication topic in this section easier. The "VM Snapshots 101" section in this paper provides helpful insights into how snapshots work.

Per-path Throttling

In addition to specifying the local and destination network paths between Tintri VMstore devices, ReplicateVM path specifications include options to throttle, or limit the maximum amount of snapshot data transferred in megabytes-per-second (MBps), *over each defined path.*

Throttling throughput allows administrators to regulate and predictably manage the bandwidth utilization used by replication during business and non-business hours. This is particularly beneficial when replicating data over fixed wide area network (WAN) leased lines between data centers in different locations, and in different time zones.

TIP: Each path consists of a source and destination IP address, (replication) key, and optional throttle settings.

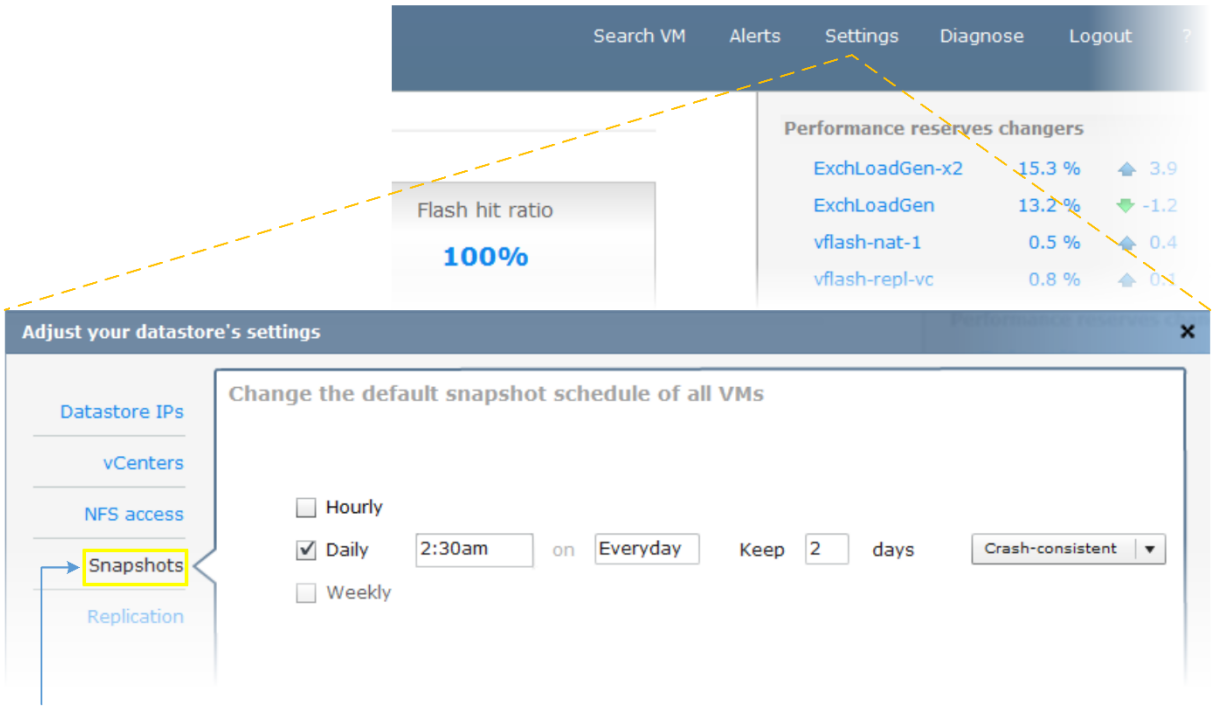
Replication Starts with Snapshots

The system default snapshot schedule does not determine per-VM replication with ReplicateVM. However, when you select a VM for replication you can "inherit" or use the system default snapshot schedule (Figure 16) for replication, or create an individual schedule for each VM (Figure 17).

How to define a system default snapshot schedule

To configure a default schedule for all VMs on the system, visit the "Settings" menu on Tintri VMstore, and then select the "Snapshots" tab (lower left, below):

Figure 14: Tintri VMstore system-wide VM snapshot default schedule settings

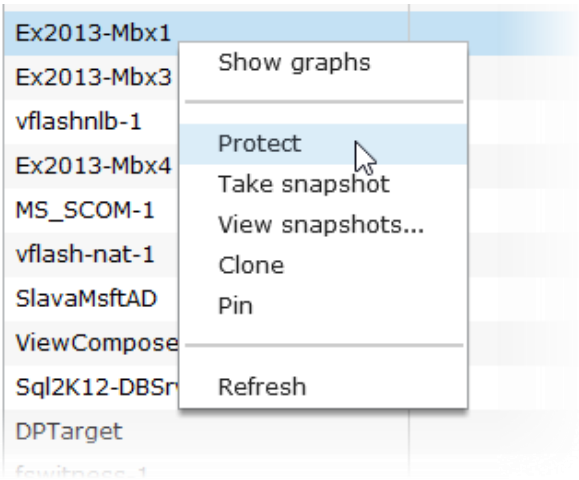


The default "Snapshots" tab of the Tintri VMstore main settings dialog (Hourly, Daily, Weekly)

Enabling Replication for a VM

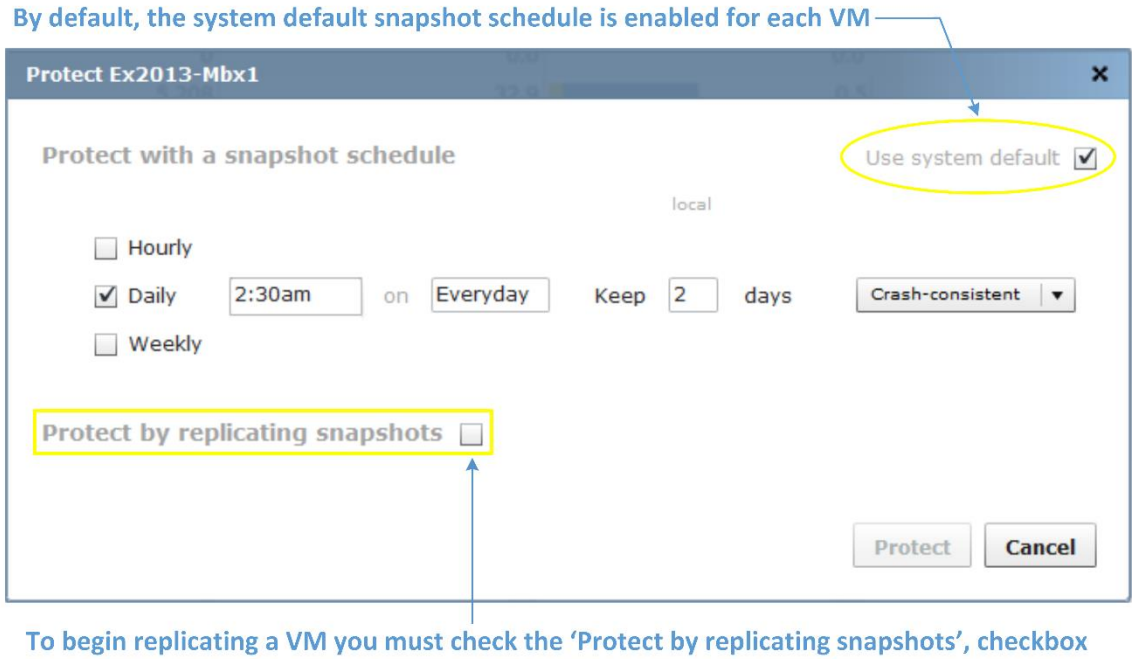
Configuring per-VM replication is a matter of selecting a VM and instructing Tintri VMstore to "protect" it by creating snapshots of the VM according to the desired (hourly, daily, weekly) snapshot schedules, and then replicating those snapshots to a destination Tintri VMstore.

Figure 15: Accessing the 'Protect' menu item of a VM, from the Tintri VMstore UI



The "Protect" menu opens the protection settings for the selected VM. Note in Figure 16 that the "Use system default" checkbox is initially "checked", and the schedule matches the system default schedule (Figure 14).

Figure 16: You selectively enable replication on a per-VM basis

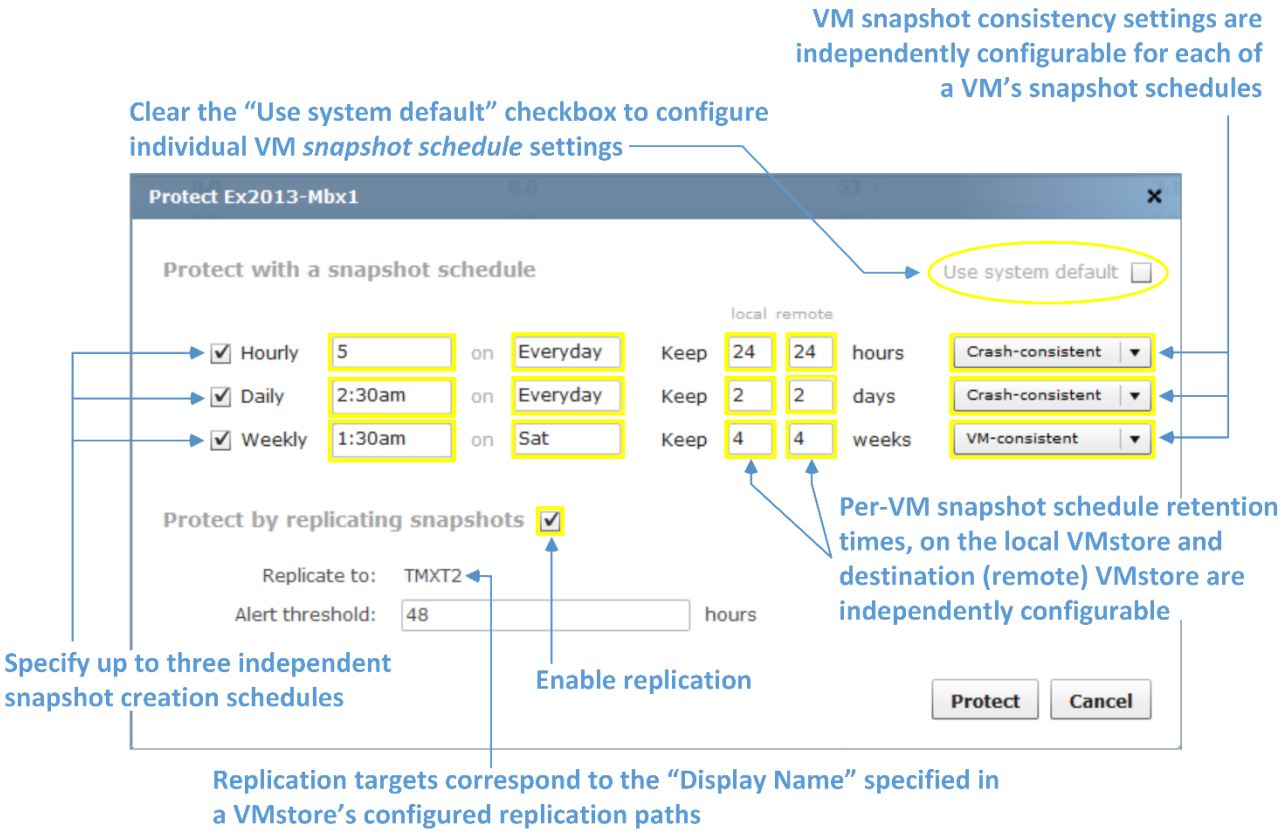


Once a given VM is "protected", and the "Protect by replicating snapshots" option checked (see above), Tintri VMstore will begin transporting a VM's individual deduplicated and compressed snapshots (Figure 20).

Customizing the ReplicateVM Settings for a VM

Each VM is configurable with its own snapshot and replication settings. Figure 17 below describes the additional options available for each individual VM when you clear the "Use system default" checkbox.

Figure 17: Further customizing a VM's snapshot protection with hourly, daily, and weekly snapshot schedules, with replication enabled

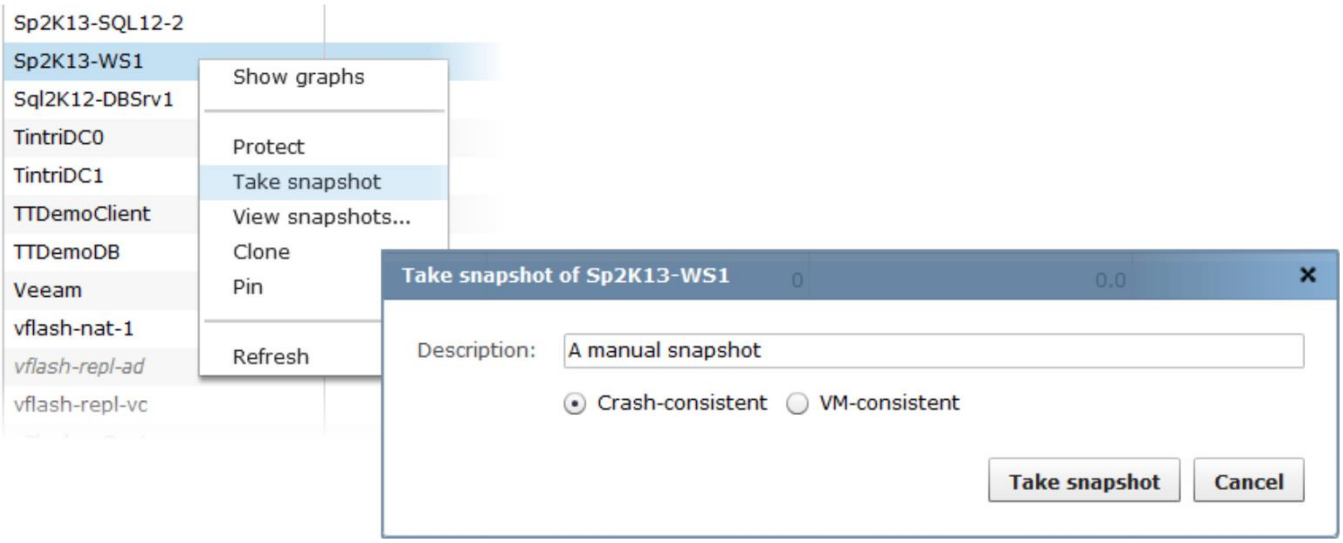


The options in Figure 17 are applicable to any VM on Tintri VMstore. Notably absent references to arcane and complex storage related tasks are not omissions. Tintri VMstore keeps the focus on the VMs and their applications and services.

ReplicateVM per-VM Updates

New snapshots trigger ReplicateVM updates. Therefore, a snapshot created via one of Tintri VMstore's per-VM schedules, or because of an administrative action (Figure 18), will initiate a ReplicateVM update to a VM's destination VMstore.

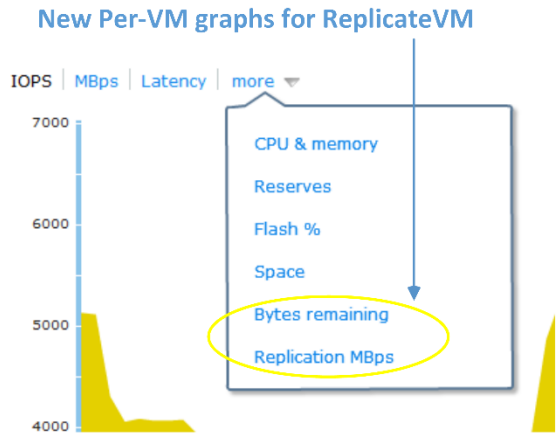
Figure 18: Creating a manual snapshot



Replication Speed and Efficiency

For a VM protected by snapshots and ReplicateVM, each replication update is deduplicated and compressed to provide optimal performance and fast VMstore-to-VMstore transfer times with maximum network efficiency.

Known for its intuitive UI and integrated per-VM graphing capabilities, Tintri VMstore adds new monitoring and graphs for ReplicateVM that allow administrators to monitor replication performance on a per-VM basis.



A VM's ReplicateVM graphs can accessed easily from the Tintri UI by selecting the "more" drop down list for the VM (Figure 19).

ReplicateVM graphs are viewable on both the originating (source) and remote (destination) Tintri VMstore systems for each VM.

The progress and throughput of a VM, and its overall replication status is clearly visible and easy to see at-a-glance.

Figure 19: Accessing per-VM ReplicateVM graphs

Logical vs. network throughput

In Figure 20 below, the logical throughput represents the "fully hydrated" size of the data, if it were not deduplicated and compressed. The network throughout is the actual amount of data (measured in MBps) being actively transferred. Deduplication and compression are natural attributes of ReplicateVM.

Figure 20: The bytes remaining (progress) and the logical ("deduped size") and network (actual) replication throughput of a VM viewed from the Tintri VMstore UI

Tip: Remember from Figure 13 that for each path configured between your Tintri VMstore appliances, configurable *throttle* values determine the upper limit or replication throughput "ceiling" (measured in MBps) for a given path during peak and non-peak business hours.

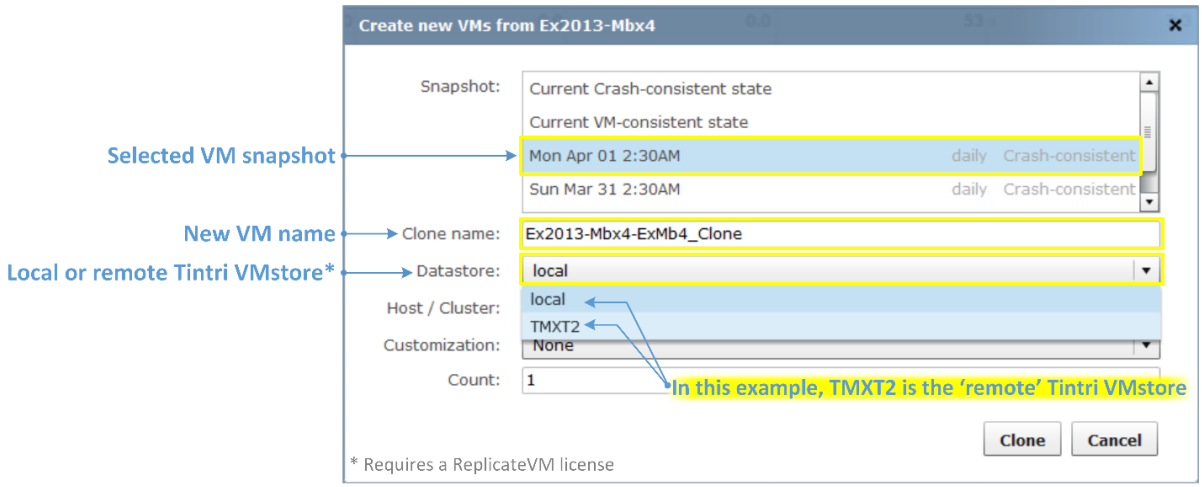
Take a moment to review the ease with which you can replicate a VM from one data center to another, and in the case of a primary data center outage, restore its services almost instantaneously at your remote site…

Remote Cloning with ReplicateVM

In addition to the cloning options mentioned in the "VM Cloning with Tintri VMstore" section of this paper, ReplicateVM extends the cloning power of Tintri VMstore and provides an array of new and flexible options:

- You can replicate VMs from Tintri VMstore to VMstore, in one-to-one, and many-to-one topologies, bi-directionally (in both directions)
- Cloning for restore or deployment operations, is supported "locally" (on the originating end of VM's replication path), and *remotely*, or on the destination VMstore to which a VM's snapshots are being replicated (i.e. via remote cloning)

Figure 21: Cloning a VM, specifying a local or remote Tintri VMstore



For VMs protected by ReplicateVM, remote cloning allows an administrator to create new VMs from snapshots on the *remote or destination end of a ReplicateVM configuration.* On the remote VMstore, the virtualization administrator simply browses the Tintri VMstore using the datastore browser in the vSphere client for example, and then adds the new VM to vCenter inventory. Remote cloning is a powerful remote management feature with many possible applications.

Applications

ReplicateVM can be incredibly useful in many applications. The list below is a sample of the applications used by Tintri customers, including applications tested and supported by Tintri VMstore with ReplicateVM.

- VMware Horizon View (VDI)
- Citrix XenDesktop (VDI)
- Microsoft SQL Server 2005 (including database mirroring for HA)
- Microsoft SQL Server 2008/R2 (including database mirroring for HA)
- Microsoft SQL Server 2012 (including Always On Availability Groups for HA)
- Microsoft Exchange Server 2010 (including Database Availability Groups for HA)
- Microsoft Exchange Server 2013 (including Database Availability Groups for HA)
- SAP on Unix and Windows
- Engineering and Geo-physical applications
- Test, Development and QA
- Transactional Financial Applications
- Private Cloud and Hosting Provider implementations
- Business Intelligence and Reporting Applications

ReplicateVM is particularly powerful when it comes to replicating important, mission critical data sets and assets essential to the operations of an organization. This includes but is not limited to protecting the core VM and application images used in server and desktop virtualization environments, and replicating the snapshots of those applications and images across multiple systems in geographically dispersed data centers.

Contact Tintri for specific information on how Tintri VMstore can accelerate the data protection objectives you are facing for all of your virtualization applications and assets.

Summary

Point-in-time snapshots support data protection, recovery, and cloning operations. Traditionally structured storage systems are poorly adapted for VMs, exceedingly complex, and conflict with the software-defined datacenter. Tintri VMstore provides system-wide default snapshot schedules as well as individual per-VM schedules for Hourly, Daily and Weekly VM snapshots. Tintri VMstore supports creating both crash-consistent and vm-consistent snapshots. Each snapshot schedule, including the system default schedules, and for each individual VM, supports its own snapshot consistency setting. Snapshots and clones share data blocks, resulting in significant space and performance efficiencies. Cloning operations use snapshots as their base of inception, resulting in new VMs. The state of a given VM captured in a VM snapshot provides the starting point for new VMs created via cloning operations. A VM created via the cloning method can "replace" its respective VM in a recovery operation, or spring to life as a new VM starting with the same settings as the VM in its parent snapshot. Tintri VMstore supports multiple vCenter instances. Through its vCenter integration, Tintri VMstore discovers the vSphere clusters and stand-alone ESX/ESXi hosts associated with the vCenter instance. When cloning, administrators can select the target vSphere cluster or stand-alone ESX/ESXi server to which the newly created cloned VM(s) are to be registered. Tintri VMstore can create new VM clones with extraordinary speed and in very large numbers. The Tintri VMstore UI supports cloning operations from 1-500 VMs at-a-time, and then meters the vCenter registrations to ease the load on vCenter.

Tintri VMstore release 2.0 includes new advancements that extend Tintri's unique and powerful per-VM capabilities through Tintri ReplicateVM. ReplicateVM yields an unprecedented level of data protection for VMs, free of the complexities associated with traditionally structured storage systems. ReplicateVM updates occur in accordance with the creation of new snapshots for VMs protected by ReplicateVM. Establishing replication paths between Tintri VMstore appliances in the same data center or in remote data centers is straightforward. ReplicateVM paths between Tintri VMstore systems include one-to-one, many-to-one, and bi-directional configurations, all of which function simultaneously. Deduplication and compression ensures efficient data transfers between Tintri VMstore systems, over fast, local networks and wide area networks. Each path configuration is tunable by applying throttle values on each path to limit throughput. Remote cloning supports creating clones of replicated VMs on the destination VMstore in a ReplicateVM cloning configuration. ReplicateVM is broadly applicable to a large number of virtualization environments and virtualized enterprise applications.

PTS
DATA CENTER SOLUTIONS
For More Information:
(866) 787-3271
Sales@PTSdcs.com

2570 W. El Camino Real, Mountain View, California 94040 | (650) 209-3900 | info@tintri.com | www.tintri.com

Tintri and Tintri VMstore are trademarks or registered trademarks of Tintri, Inc. All other trademarks or service marks are the property of their respective holders and are hereby acknowledged. © 2013 Tintri. All rights reserved. v1.1