Tintri

TECHNICAL WHITE PAPER

Virtualize More with Application-Aware Flash Storage: A Tintri VMstore Overview

Flash and Virtualization: Storage, Interrupted

Enterprise storage is confronted by two revolutionizing technologies at once:

- Virtualization is the new normal. More than 75 percent of new workloads are now virtualized, and companies are beginning to make significant investments in virtual desktop infrastructure (VDI).
- Commodity flash storage is quickly becoming a significant part of both local and shared storage infrastructure.

Existing storage offerings are poorly adapted to both virtualization and flash:

- Virtualization benefits significantly from shared storage, but traditional general-purpose shared storage was designed 20 years before VMware popularized virtualization with a different set of workloads in mind.
- Flash storage — which is about 400 times faster than disk — must be treated differently than the rotating magnetic disks most storage systems were designed to use. As a result, most solutions use an expensive and complex bolt-on approach with flash-as-cache.

Traditionally structured flash storage arrays are designed to deliver high I/O rates, but they are not designed to run virtual workloads efficiently, so costly space and performance are wasted on idle data. It's the equivalent of using a commercial jet to commute 30 miles — it may be slightly faster than driving, but the fuel costs are prohibitively expensive.

Consequently, customers struggle with existing storage systems that are poorly adapted to both flash and virtualization, inhibiting the fundamental IT goals of lower cost and greater business agility. Systems purpose-built for both virtualization and flash can overcome these issues. As data centers move from about 30 percent virtualized to well over 60 percent virtualized, deploying storage specifically designed for these environments provides substantially more value. This paper will explore the challenges of designing storage systems using flash for virtualization, and describe the Tintri approach.

Flash: Speed has its Challenges

Flash storage can deliver 400-times greater raw performance than spinning disk, but introduces fundamental architectural changes. For comparison, the speed of sound — 768 mph at sea level — is "only" 250 times faster than the average speed of walking. To travel at supersonic speeds, engineers designed sophisticated aircraft systems specifically for high speeds. It may be possible to strap a rocket motor to one's back and attempt to travel at 768 mph, but the result would be less than ideal.

Flash poses similar challenges to existing storage systems. Multilevel cell (MLC) solid-state drives (SSDs) are the most cost-effective approach and provide excellent random IO performance, but have several idiosyncrasies which make MLC unsuitable as a simple drop-in replacement for rotating magnetic disks:

- **Cost-efficiency:** Although MLC is two to four times cheaper than its cousin SLC, it's still about 20 times more expensive than SATA disks. To use flash cost-efficiently, technologies like inline deduplication and compression are critical.
- **Latency spikes:** Flash drives are programmed at the page level (512B to 4KB), but can only be erased at the block level (512KB to 2MB) sizes much larger than average IO requests. This asymmetry in write vs. erase sizes leads to write amplification which, if not managed appropriately, creates latency spikes.
- **Durability:** MLC flash in particular can be vulnerable to durability and reliability problems in the underlying flash technology. Each MLC cell can be overwritten only 5,000 to 10,000 times before wearing out, so the file system must account for this and write evenly across cells.

Disk-based systems were created more than 20 years ago to cope with a decidedly different set of problems. Adapting these systems to use flash efficiently is comparable to attempting to adapt an old 8-bit single-threaded operating system to use today's multicore 64-bit architectures.

The Tintri Flash-Based Architecture

The Tintri VMstore appliance is designed from scratch to fully exploit flash technology for virtual environments. The custom Tintri OS is specifically designed to ensure robust data integrity, reliability and durability in flash, and operates at the virtualization layer (more on this later). MLC flash is a key technology that enables Tintri to meet the intense random IO required to aggregate hundreds or even thousands of VMs on a single appliance. The Tintri OS leverages flash in several ways:

- **Cost-efficiency:** By design, nearly all active data will live exclusively in flash. To maximize flash usage, Tintri combines fast inline dedupe and compression with file system intelligence that automatically moves only cold data to SATA. Inline dedupe and compression are also highly effective in virtualized environments where many VMs are deployed by cloning existing VMs, or have the same operating system and applications installed. Tintri VMstore flash is neither a pure read cache nor a separate pre-allocated storage tier. Instead, flash is intelligently utilized where its high performance will provide the most benefit.

- **Latency management:** Tintri employs sophisticated patent- pending technology to eliminate both the write amplification and latency spikes characteristic of MLC flash Technology (Figure 1). This approach delivers consistent sub-millisecond latency from costeffective MLC flash.

- **Flash durability:** Tintri uses an array of technologies including deduplication, compression, advanced transactional and garbage collection techniques, combined with SMART (Self- Monitoring, Analysis and Reporting Technology) monitoring of flash devices to intelligently maximize the durability of MLC flash. Tintri also employs RAID 6, eliminating the impact of potential latent manufacturing or internal software defects from this new class of storage devices.

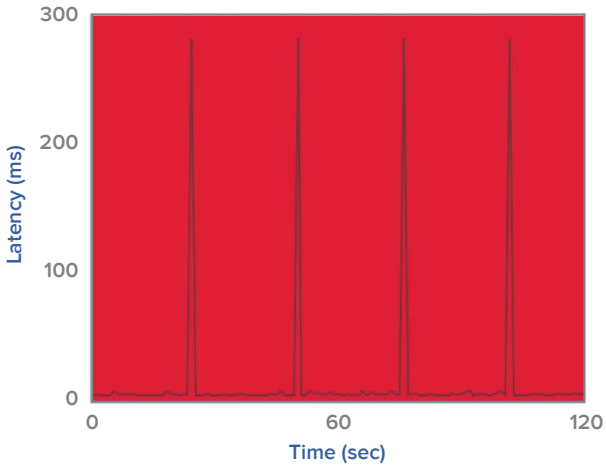


Figure 1: Tintri employs sophisticated, patent-pending technology to eliminate both write amplification and latency spikes.

Tintri VMstore leverages the strengths of MLC flash while negating its weaknesses, providing a highly reliable and durable storage system suitable for enterprise applications.

The Virtualization and Storage Mismatch

Virtualization introduces an element of simplicity and agility the physical world lacks, with a single view of resources under hypervisor control (CPU, memory, and networking resources). However, there is a language barrier. Virtualization owes its success in transforming data centers with the powerful virtual machine (VM) abstraction. An application in a virtual infrastructure is for the first time, a truly logical object. Virtual applications can be copied, reconfigured, redeployed, analyzed, and managed in ways that are difficult for physical machines. Virtualization provides not just the benefits of server and desktop consolidation but also simplifies data-center management, deployment, and maintenance.

However, most existing IT infrastructure and tools—including storage—don't "speak" virtualization as their native language. This obscures the relationship between the virtualized application and the underlying infrastructure.

The industry has had to rethink traditional functions like monitoring and troubleshooting to account for virtualization, but not every element of the infrastructure has adapted.

Virtualization has improved the cost and efficiency of managing servers—but has significantly increased the time and complexity of diagnosing and fixing performance problems with storage. Designed before the widespread adoption of virtualization, legacy shared storage systems provide little help resolving performance problems with individual VMs. The result is a suboptimal infrastructure dominated by ever-escalating storage costs due to over-provisioning. According to VMware's own estimates in 2010, storage accounted for up to 60 percent of virtualization deployment costs.

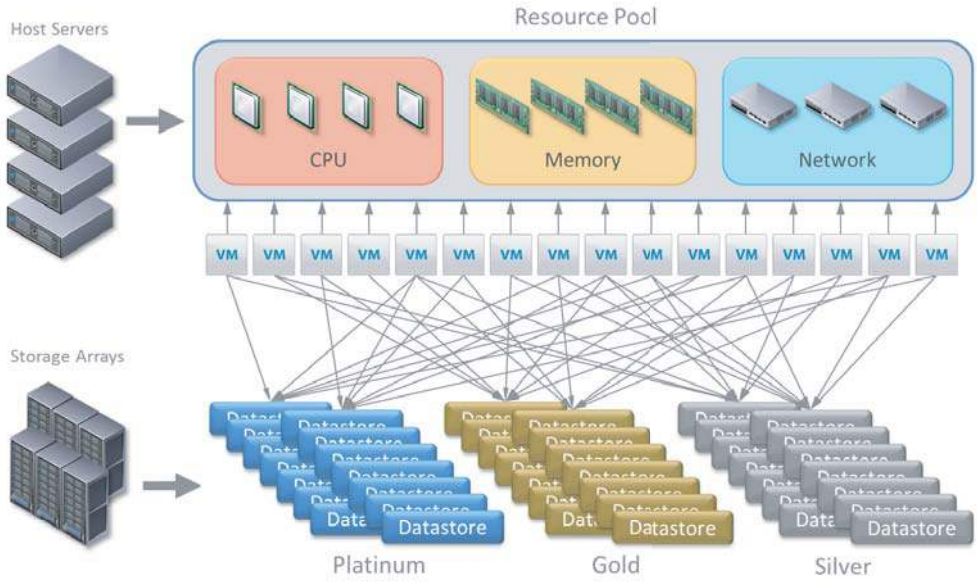


Figure 2: Legacy shared storage maps VMs to LUNs or volumes, rather than managing at the VM and vDisk level.

In fact, traditional shared storage amplifies troubleshooting issues, via multiple opaque layers hidden from the VM administrator. Existing storage systems carve out LUNs or volumes, which are mismatched with virtual resources—VMs and vDisks.

Because of its compatibility with virtualization, adoption of shared storage — both SAN (Fibre Channel or iSCSI) and NAS (NFS) — has accelerated. However, traditional shared storage products present barriers to virtualization: They manage objects such as LUNs, volumes, or tiers, which have no intrinsic meaning for VMs (Figure 2). Legacy storage cannot monitor, snapshot, set policies or replicate individual VMs.

This mismatch increases cost and complexity. Each new VM instance must be assigned a specific storage LUN or volume. When IO requirements and VM behavior are not well understood, this becomes a painful trial-and-error process. Storage and VM administrators must coordinate to ensure each application has not only the space it needs, but also sufficient IO performance for the expected load.

Usually, multiple VMs occupy the same volume or LUN to reduce mapping complexity and space overhead; however, this complicates IO performance problems. A storage-centric view of performance data means administrators must work backward to determine which VMs are affected and which VMs are generating load.

Even technologies such as auto-tiering, which seek to reduce storage management overhead, operate at the wrong level. Without the ability to report behavior on a per-VM or per-virtual disk level, "advanced" storage technology increases complexity and risk. Instead of the unvarnished VM model provided by hypervisors, legacy storage responds with a blizzard of options and interfaces.

In these situations, the complexity of configuring, managing and tuning traditional storage for VMs is costly and ultimately limits the adoption of virtualization. In fact, many applications cannot be cost-effectively virtualized with legacy shared storage.

Overcoming the Limits of Traditional Storage with Application-Aware Storage

Purpose-built for VMs and focused specifically on the problems of VM storage, Tintri VMstore™ provides management at the same level as the rest of the virtual infrastructure (Figure 3).

Legacy Shared Storage		Tintri
LUN	RAID Level	VM
Volume	WW Names	vDisk
Disk RPM	Cache Size	
Block Size	Dedupe	
Stripe Width	Snapshot Frequency	
Sequential / Random	Inner / Outer Tracks	
Pathing	Degraded Performance	

Figure 3: Tintri eliminates unnecessary abstractions.

Tintri incorporates advances in flash technology, file system architecture, and user interface design to make storage for virtual applications uncomplicated and efficient. Tintri VMstore is designed from the ground up exclusively for VMs by experts in both virtualization and storage.

Tintri VMstore is managed in terms of VMs and virtual disks, not LUNs or volumes. The Tintri OS is built from scratch to meet the demands of a VM environment, and to provide features relevant to VMs. It is designed to use flash efficiently and reliably while leveraging key technologies like deduplication, compression and automatic data placement to deliver 99 percent of IO from flash.

These innovations shift the focus from managing storage as a separately configured component to managing VMs as a whole. This overcomes the performance, management and cost obstacles that prevent virtualization of more of the computing infrastructure. Our sharp focus on creating a better storage system for VMs enables us to build a fundamentally new type of product.

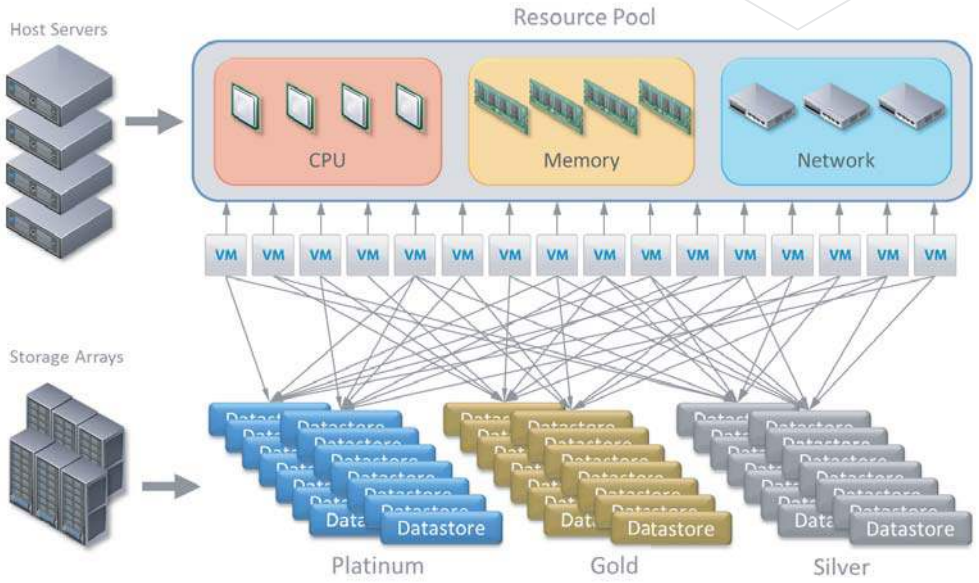


Figure 4: Tintri VMstore maps directly to the VM and vDisk abstractions.

Building a VM-focused management interface relies on far more than just an attractive GUI. The underlying storage system natively understands and supports storage management operations such as performance and capacity monitoring, snapshots, quality of service (QoS) management, and replication at the VM level.

Focusing exclusively on VMs enables Tintri to eliminate unnecessary levels of mapping and complexity required by general purpose storage systems (Figure 4). Decision-making is delegated to lower levels of the system and achieves much higher levels of automation and optimization than possible for general-purpose storage systems. The result is an agile architecture with much simpler abstractions and interfaces, which in turn facilitates further automation and optimization.

The way Tintri focuses on VMs is most apparent in the VMstore management interface, which presents VMs as the basic units of management, rather than LUNs, volumes, or files. Every object in the interface is familiar to VM administrators (Figure 5). The interface is straightforward enough for VM administrators to manage storage directly, yet sophisticated enough for storage administrators to leverage their expertise in managing storage for large numbers of VMs.

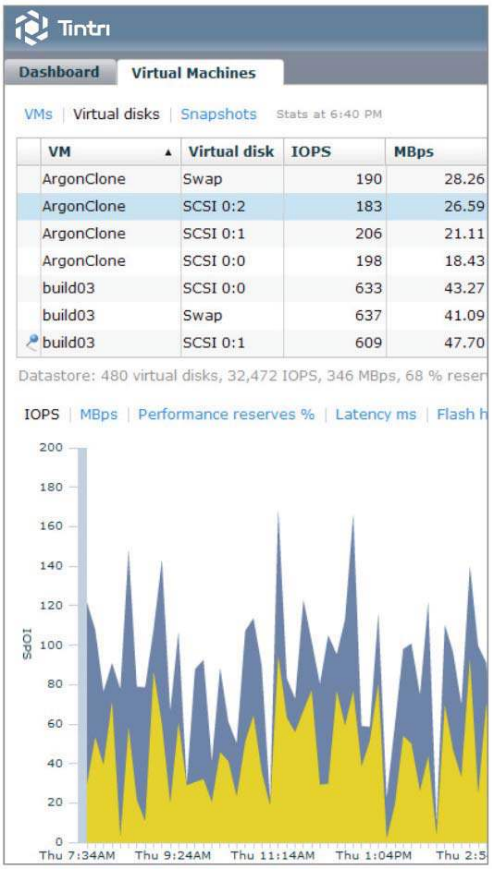


Figure 5: The Tintri user interface displays VM and virtual disk level statistics.

Instant Bottleneck Visualization

Administrators dread troubleshooting storage performance problems. A VM complaint may be due to a problem with the storage, but how do you verify this when the VM is sharing a LUN with a dozen other VMs, and the LUN is a slice of a RAID array that contains many other LUNs? Unfortunately, the legacy array provides no statistics on a per-VM basis. The problem could have roots in the ESX host or the storage network, or even the user's application.

Identifying performance bottlenecks is a time consuming, frustrating and sometimes inconclusive process that requires iteratively gathering data, analyzing the data to form a hypothesis, and testing. In large enterprises, this process often involves coordination between several people and departments and can span many days or even weeks.

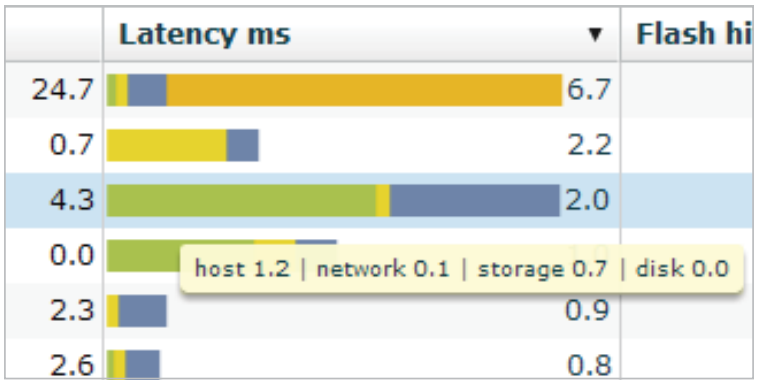


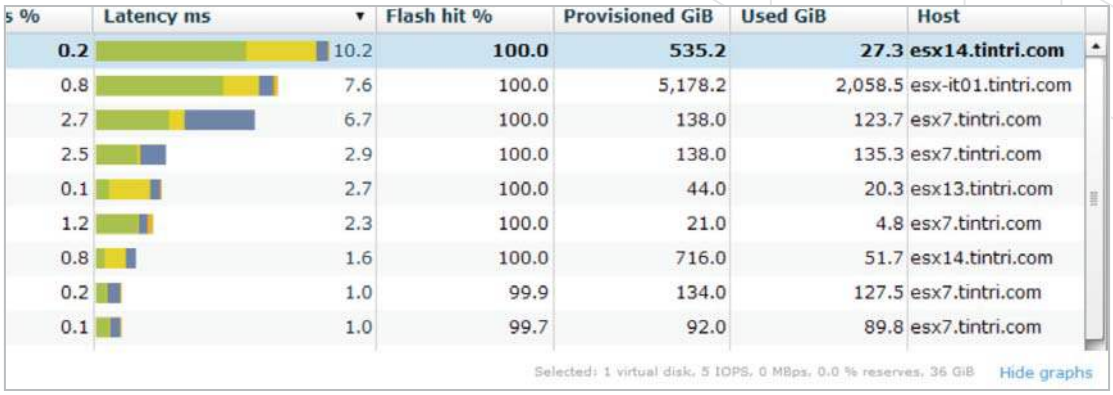
Figure 6: Tintri provides instant insight in to VM latency across the infrastructure.

Tintri Bottleneck Visualization

The troubleshooting process described above is fully automated using Tintri instant bottleneck visualization. For each VM and vDisk stored on the system, Tintri displays a breakdown of the end-to-end latency all the way from the guest OS down to the disks within the Tintri appliance. For any VM or vDisk, you can see at a glance how much of the latency was spent in the ESX host, the network, the Tintri file system, or accessing the disk (Figure 6). Moreover, a history of this information is automatically stored and can be displayed as a graph, so you can see the bottleneck for each VM at any given point over the last seven days.

This visualization is generated by automatically collecting per-VM hypervisor latency stats and correlating them with per-VM storage stats Tintri VMstore collects for each VM (see diagram, below). Hypervisor latencies are obtained using standard vCenter APIs while the network, file system and disk latencies are provided by Tintri VMstore, which knows, for each IO request, the identity of the corresponding VM.

Tintri provides latency statistics in an intuitive format (Figure 7). In an instant, you can see the bottleneck rather than trying to deduce where it is, based on indirect measurements and time-consuming detective work.



s %	Latency ms	Flash hit %	Provisioned GiB	Used GiB	Host
0.2	**10.2**	**100.0**	**535.2**	**27.3**	**esx14.tintri.com**
0.8	7.6	100.0	5,178.2	2,058.5	esx-it01.tintri.com
2.7	6.7	100.0	138.0	123.7	esx7.tintri.com
2.5	2.9	100.0	138.0	135.3	esx7.tintri.com
0.1	2.7	100.0	44.0	20.3	esx13.tintri.com
1.2	2.3	100.0	21.0	4.8	esx7.tintri.com
0.8	1.6	100.0	716.0	51.7	esx14.tintri.com
0.2	1.0	99.9	134.0	127.5	esx7.tintri.com
0.1	1.0	99.7	92.0	89.8	esx7.tintri.com

Selected: 1 virtual disk, 5 IOPS, 0 MBps, 0.0 % reserves, 36 GiB Hide graphs

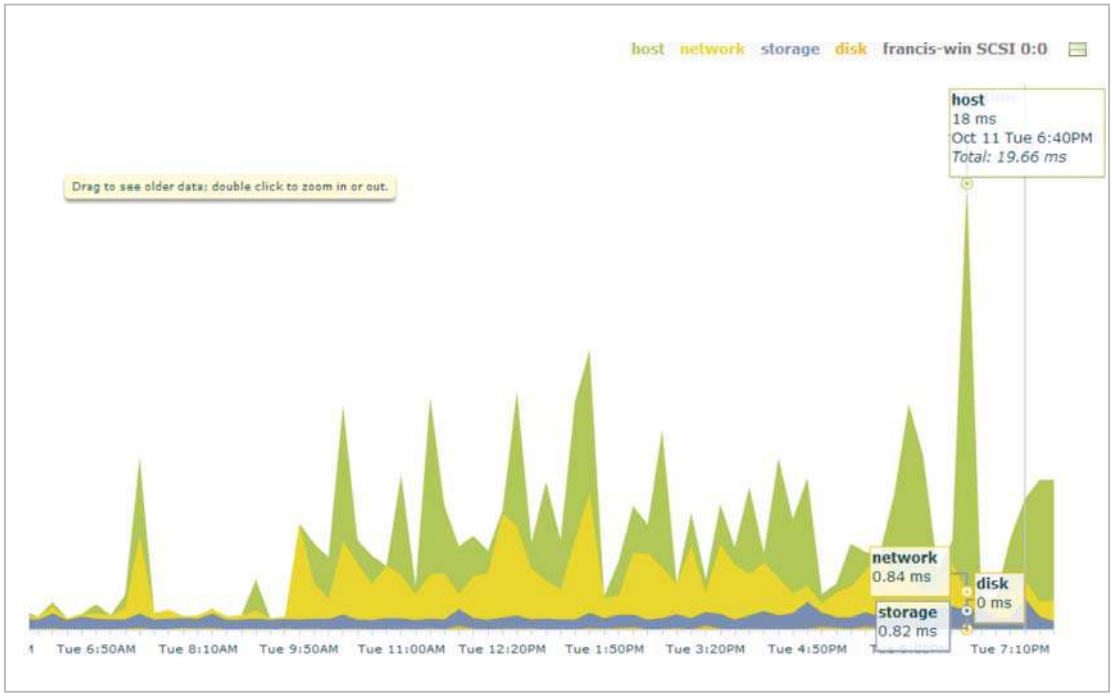


Figure 7: Tintri provides insight in to the historical latency and statistics at the VM level.

VM Alignment

VM alignment is the daunting to-do item. It is a problem that poses real challenges as virtualization spreads into more mainstream workloads. Misaligned VMs magnify IO requests, consuming extra IOPS on the storage array. At a small scale, the impact is small. However, the impact snowballs as the environment grows with a single array supporting hundreds of VMs. At this size, performance impact estimates range from 10 percent to more than 30 percent.

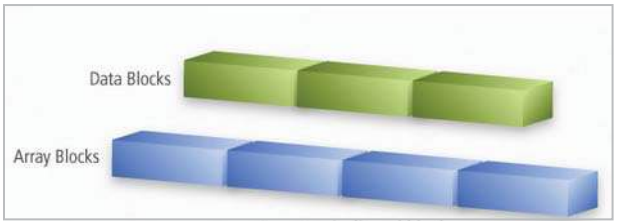


Figure 8: Misaligned blocks.

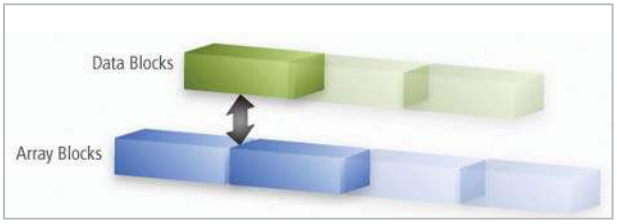


Figure 9: Overhead due to misaligned IO.

Every guest OS writes data to disk in logical chunks. Storage arrays also represent data in logical blocks. When a VM is created, the block boundaries on the guest OS and storage don't always align automatically.

If the blocks are not aligned, guest requests span two storage blocks, requiring additional IO (see Figure 8 and Figure 9).

A VM runs a guest OS that includes one or more virtual disks to store state. The guest OS typically defines the layout of each virtual disk with a common partition layout, such as a master boot record (MBR). The MBR stores information about how each virtual disk is partitioned into smaller regions, with its size and location. Except for Windows Server 2008 and Windows 7, blocks defined by the guest OS file system (NTFS, EXT3, etc.) do not typically align with the underlying datastore block layout.

Tintri VM Auto-alignment

So why are VMs misaligned? Certainly administrators attempt to address the issue by using a variety of utilities to manually align VMs and reduce performance demand. Numerous blogs, white papers, and knowledgebase articles describe why VMs should be aligned and provide step-by-step instructions. Unfortunately, as administrators know, realigning a VM is a manual process. Worse — it generally requires downtime.

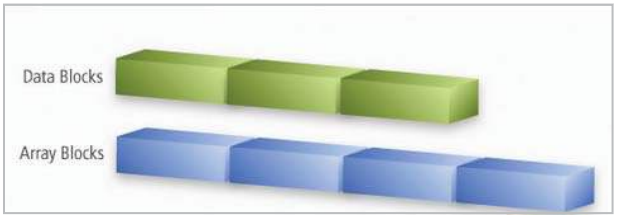


Figure 10: Aligned blocks.

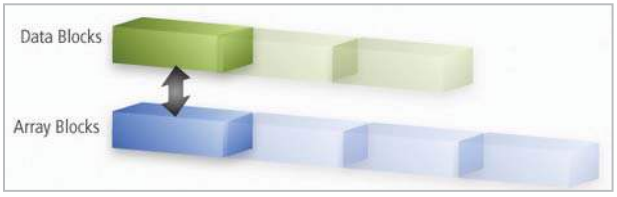


Figure 11: No overhead due to aligned IO.

Our application-aware file system intrinsically "understands" each virtual disk. Building on this foundation, Tintri VMstore offers VM auto-alignment. Rather than the conventional disruptive approach of realigning each guest, Tintri VMstore dynamically adapts to the guest layout (Figure 10 and Figure 11). Nothing changes from the guest OS point of view. Tintri VMstore automatically aligns all VMs as they are migrated, deployed, cloned or created—with zero downtime. A VM administrator can now eliminate this arcane task and enjoy performance gains from 10 percent to more than 30 percent with no VM downtime, and zero user interaction.

Granular and Scalable Snapshots

Legacy shared storage architectures provide snapshots of storage objects, such as LUNs and volumes, rather than VMs. These snapshot technologies lead to inefficient storage utilization as hundreds of VMs with varying change rates are often snapshotted at once. Snapshot schedules can only be set at a LUN, or a volume level, leading to such best practices recommendations as creating one LUN per VM as a work around for the need to create individualized snapshot schedules at per VM level.

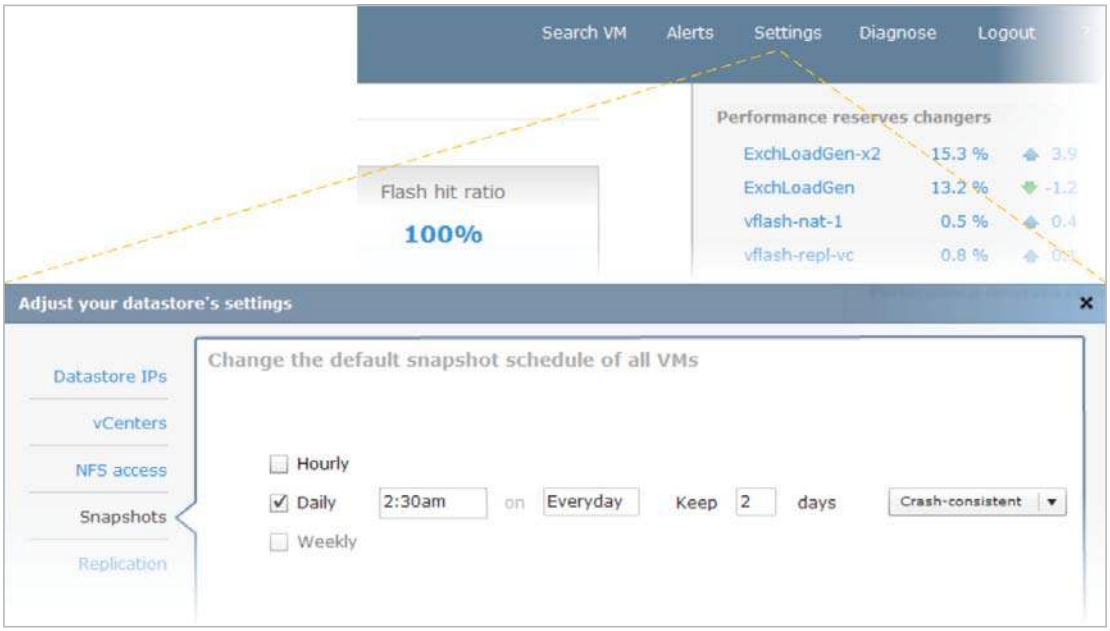


Figure 12: Administrators set a system default schedule for VM snapshots

Unique space-efficient and granular per-VM snapshots allow administrators to create snapshots of individual VMs and quickly recover data or entire VMs from snapshots. Tintri OS supports 128 snapshots per VM for scalable data protection. Data protection management is also simplified with default snapshot schedules that protect every VM automatically while custom schedules on a per-VM basis (Figure 12) can be used to tailor data protection needs for specific VMs.

System Default Snapshot Schedules

Note that a system-wide default snapshot schedule does not create interdependencies between each VM's respective snapshots. Each VM still "owns" its own individual snapshots and the system-default schedule can be overridden on a per-VM basis as shown in Figure 13.

Per-VM snapshot management schedules

To apply VM-specific snapshot settings other than the system default schedule is straightforward and painless.

Select "Protect" from the per-VM context menu in the Tintri VMstore UI to spawn the Protect settings dialog.

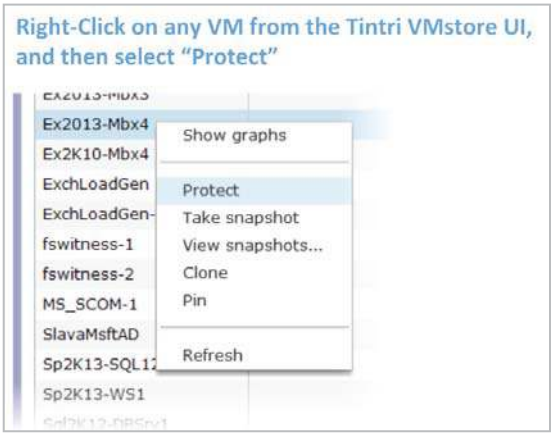


Figure 13: Setting per-VM snapshot protection and retention schedules

Figure 14 (below) shows the settings for a VM named “Ex2013-Mbx4.” Note that the “Use system default” checkbox is cleared so that the selected VM’s snapshot schedules can be set according to its individual requirements, rather than using the system-wide defaults.

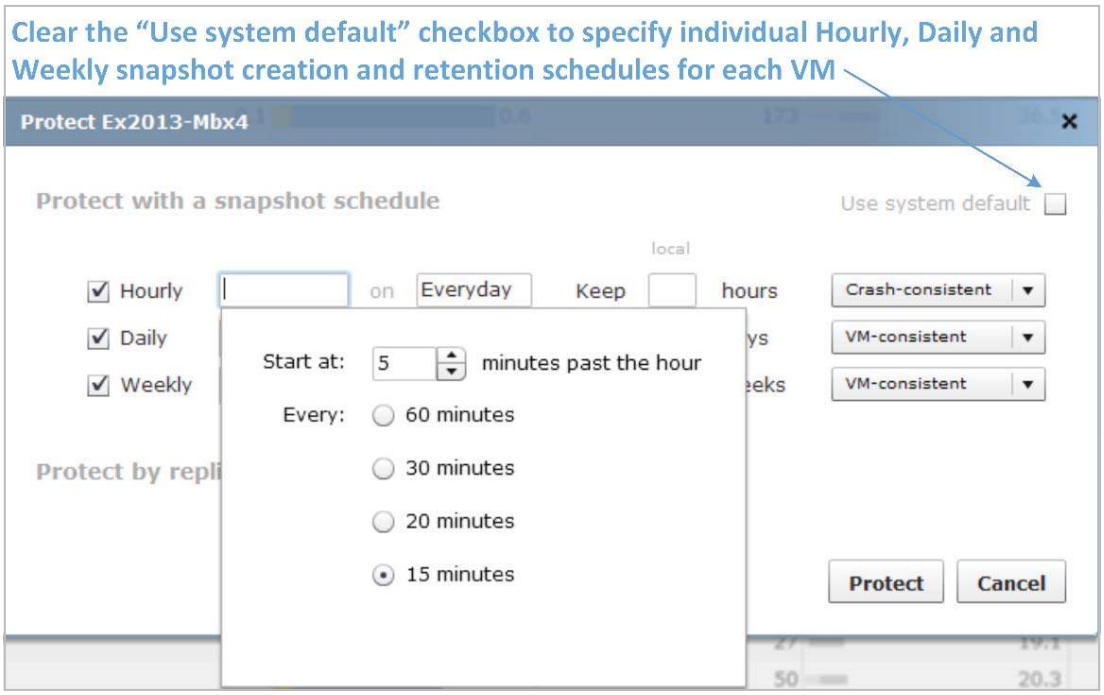


Figure 14: Specifying up to three, Hourly, Daily, and Weekly, per-VM snapshot protection schedules

Tintri OS provides crash-consistent and hypervisor coordinated VM-consistent snapshots. Crash consistent snapshots do not take extra measures with the hypervisor or guest VM to coordinate snapshots. Thanks to integration with native hypervisor management tools, such as VMware vCenter integration, Tintri OS provides VM consistent snapshots for simpler application recovery. With VM consistent snapshots, hypervisor management APIs are invoked to quiesce the application in a VM for a VM consistent snapshot.

Unlike storage-centric snapshot technologies in legacy shared storage systems, Tintri per-VM snapshots make recovery workflows remarkably easy. Files from individual VMs can be recovered without additional management overhead, dramatically reducing the time to recovery.

Advanced Per-VM Cloning

Tintri OS leverages per-VM snapshots to allow users to create new VMs through cloning operations, where the state captured in a given VM snapshot serves as the “parent”, and the new “cloned” VMs can be thought of as “children.” Like actual children, new VMs created via cloning operations exist and function independently from the parent VM(s) from which they are created. Behind the scenes, the new VMs share common vDisk references with their parent VM snapshots to maximize space and performance efficiencies.

The unique way—in fact, patented method—Tintri uses flash assures that clones are 100% performance efficient. They get the same level of performance as any other VM stored on a Tintri VMstore system.

Initially, new VMs created via cloning do not consume any significant space since they are virtually identical to their respective parent VMs. The extent to which they individually grow and diverge from the data they share with their respective parents defines their incremental storage space requirements.

Legacy shared storage systems that provide cloning capabilities at the LUN or file system volume level vastly complicate VM deployment, cloning and management operations. Per-VM, space efficient cloning operates at the per-VM level, and eliminates the limitations of legacy storage architectures that necessitate complex storage provisioning and management procedures.

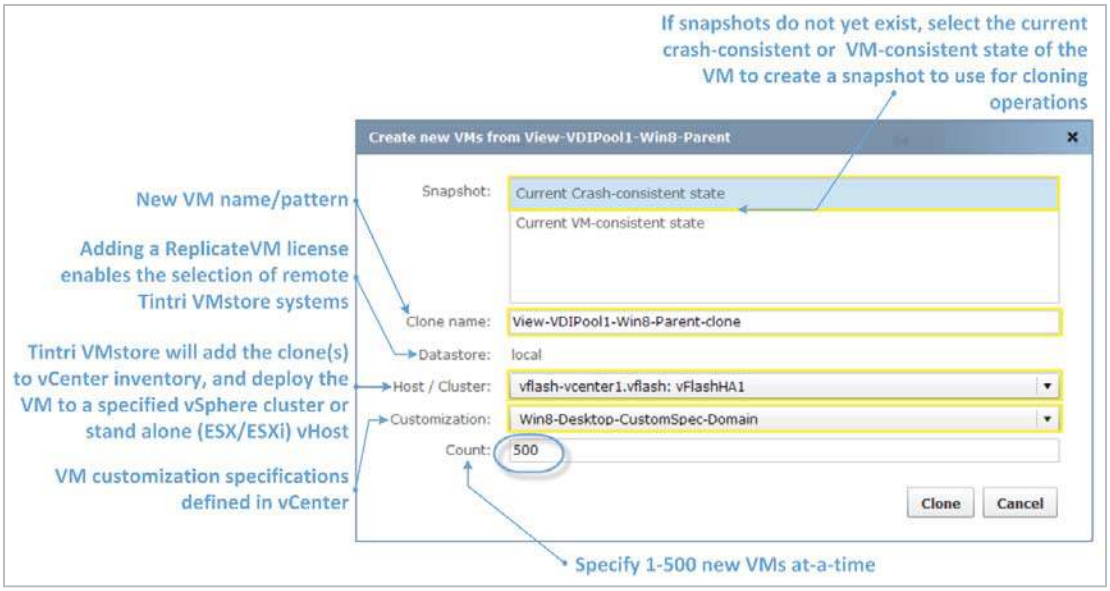


Figure 15: Per-VM cloning using the Tintri VMstore UI

Using the Tintri UI, hundreds of clone VMs can be created at a time (Figure 15). Users can select an existing snapshot of a VM or the live running state to create clone VMs. The clone VMs are automatically registered and visible to hypervisor for immediate use. Administrators can also select customization specifications defined in vCenter for preparing the newly created clone VMs. Further, clones can also be created from template VMs for use cases such as provisioning, test & development, and virtual desktop infrastructure.

Tintri ReplicateVM™

Unique to Tintri VMstore, ReplicateVM enables administrators to apply protection policies to individual VMs, rather than to arbitrary units of storage such as volumes or LUNs. ReplicateVM efficiently replicates the deduplicated and compressed snapshots of VMs from one Tintri VMstore to another. Replication can be dedicated to specific network interfaces, and optionally throttled to limit the rate of replication when replicating snapshots between Tintri VMstore appliances located in datacenters connected over wide-area networks.

The power of ReplicateVM comes clearly into view when administrators realize the power they can wield by rightclicking on a VM and then quickly and easily establishing a protective snapshot and replication policy on each VM or VMs as-needed. Protection policies are applied to database server VMs running applications like Microsoft SQL Server, Oracle, SAP, Microsoft Exchange. Distributing “Gold” (master/parent) VM images used to create Desktop Pools for VMware Horizon View or VM Catalogs for XenDesktop VDI deployments enables multi-site HA for VDI.

Accessing protected VM snapshots on the source, or destination Tintri VMstore system is painless due to seamless vSphere integration. As discussed previously, Tintri VMstore always adds newly cloned VMs to the vCenter inventory you specify[1], so that they are ready to be powered and into service immediately.

1. Tintri VMstore supports multiple vCenter instances, each of which is selectable when creating clones. Multiple vCenter support is not tied to a ReplicateVM license; it is a standard option that was added in a prior release (Tintri OS 1.4, Spring 2012).

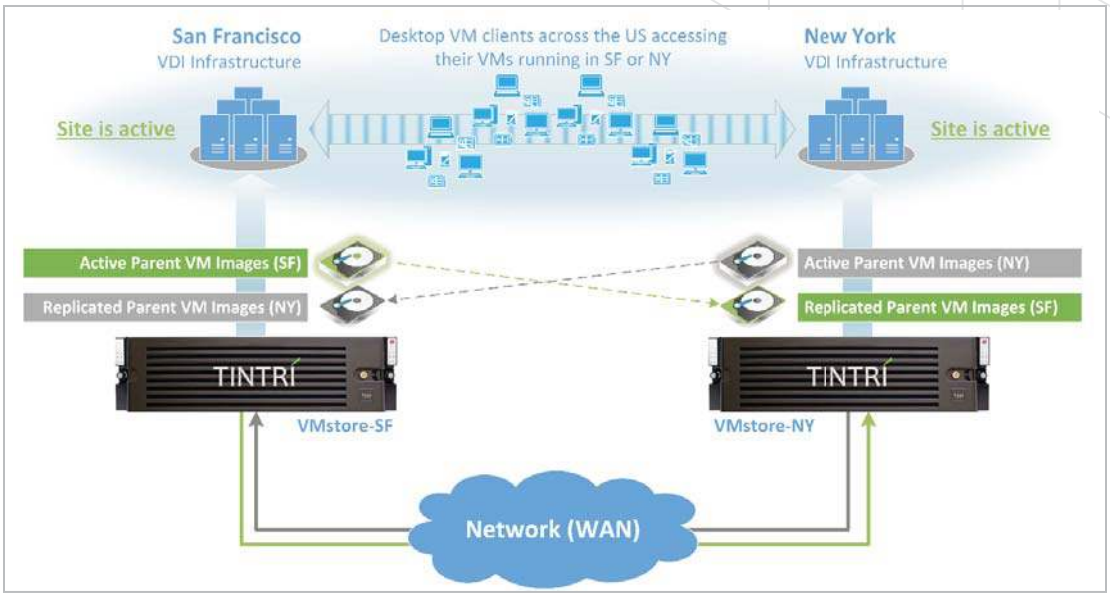


Figure 16: VDI high availability example, spanning two data center locations

Exceptionally strong, flexible and easy to use capabilities like these, with per-VM selectivity, put Tintri VMstore in a class by itself.

ReplicateVM updates between Tintri VMstore systems are de-duped and compressed. Figure 17 below shows the logical size (without dedupe and compression) of a ReplicateVM update vs. the actual MBps being transferred over the network. Time-of-day throttling options allow administrators to regulate throughput during peak and non-peak hours.

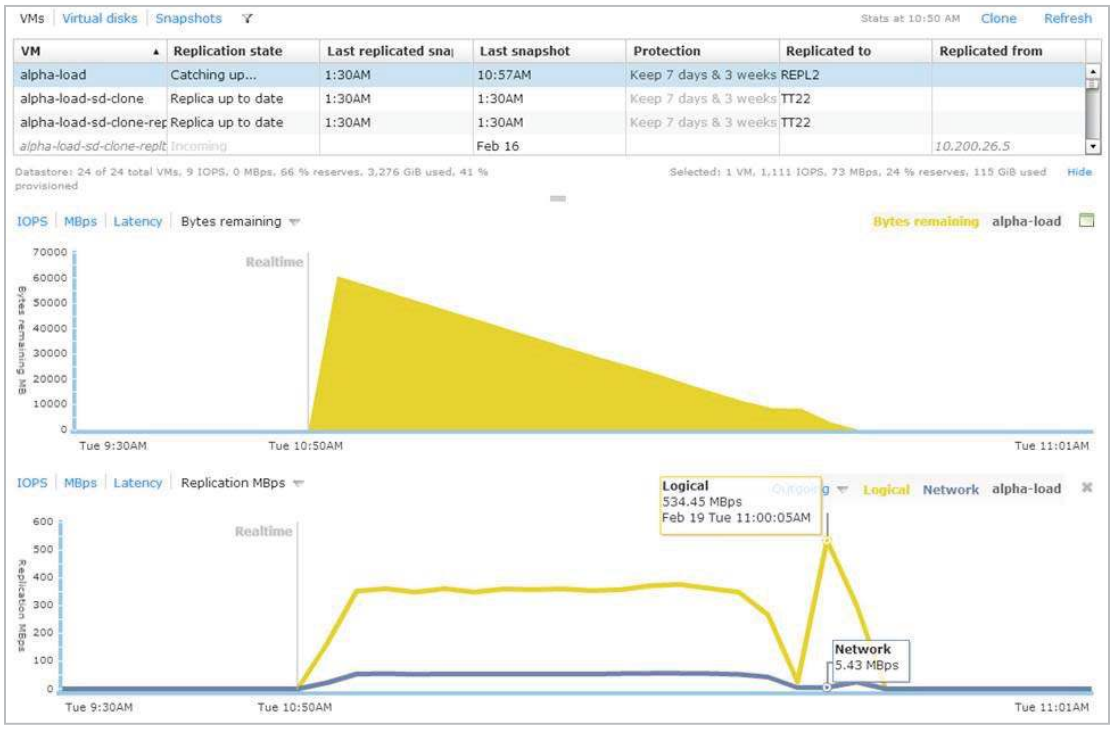


Figure 17: The bytes remaining (progress) and the logical ("deduped size") and network (actual) replication throughput of a VM viewed from the Tintri VMstore UI

Tintri Cloning with ReplicateVM

In addition to the cloning options mentioned in the Advanced Per-VM Cloning 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).

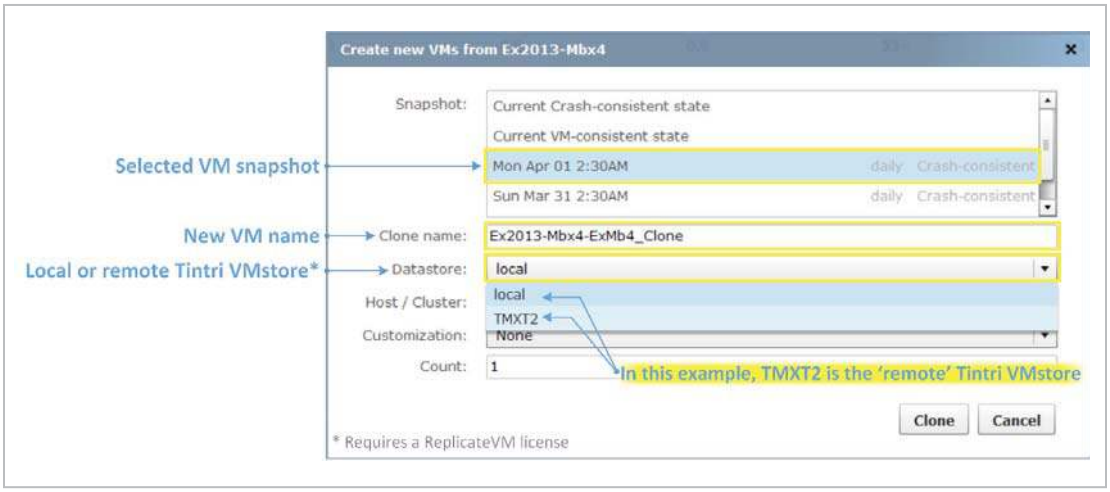


Figure 18: Note the ease with which you can replicate a VM from one data center to another, and in the case of a primary data center outage, restore its services almost instantaneously at your remote site.

Summary

Storage remains the primary obstacle to accelerating virtualization growth. Tintri VMstore allows you to overcome the complexity, performance, and cost obstacles that prevent virtualization of more of your computing infrastructure. Tintri VMstore, with its innovative Application-aware file system, leverages cost-effective MLC flash to deliver high performance in a remarkably small footprint that can scale incrementally to meet growing needs. The Tintri file system uses deduplication, compression, snapshots, clones and thin provisioning to provide the unparalleled VM density required for deploying virtual infrastructures. Instant bottleneck visualization and VM autoalignment are a direct outgrowth of our custom Application-aware file system. ReplicateVM software dramatically simplifies the snapshot protection and replication of VMs and dovetails seamlessly into vSphere virtual infrastructures for maximum flexibility across a wide range of applications. Tintri application-aware storage eliminates complicated issues in virtualized environments and leverages flash to provide sub-millisecond latency for hundreds of VMs on a single device.

Tintri

201 Ravendale Dr.,
Mt. View CA 94043
650.810.8200
info@tintri.com | **www.tintri.com**