Dynamic Storage Provisioning

Considerations and Best Practices for Using Virtual Disk Thin Provisioning

WHITE PAPER

Table of Contents

Introduction

VMware vSphere™ 4 introduces many new features that enable greater efficiency, control, and choice for storage resources. With the full support for thin provisioned virtual disks, VMware has provided a new management optimization feature that, if correctly used, can save significant storage resources. However, if not used correctly, could result in management challenges.

This paper examines the capabilities of thin provisioning, enhancements to VMware vCenter™ in vSphere, and addresses best practices to ensure thin provisioned virtual disks are a benefit and work to their full potential. This paper also explains how thin virtual disks fit into the overall storage stack, and how virtual disk thin provisioning operates with array-based thin provisioning and dynamic expansion of virtual disks, VMFS volumes, and LUNs within the array.

Background

Thin provisioning is a way to conserve storage resources by assigning only what is needed to back an allocation of space instead of allocating the full amount of space requested. Thin provisioning conserves space by providing the requestor only the amount of space needed at the time of the request, and then adding more as the demand for that allocated storage grows. But if not done correctly, this can lead to an out of space condition that could impact virtual machine availability.

As an example, airlines have been doing this for years. They became frustrated with last minute changes in customers' travel plans resulting in half empty planes. So, they started overbooking flights to increase the utilization of the seats on their flights. This concept has existed for many years in other industries as well, and has also been available in some file systems and storage arrays.

VMware has supported thin provisioned virtual disks in NFS datastores since they were supported in release 3 of the ESX server. VMware has also offered experimental support for thin provisioning in VMFS with that same release for block-based storage (FC and iSCSI). However, to create thin virtual disk on a VMFS volume datastore, one had to use the command line as it was not an option in the Virtual Center interface in VI3. With the release of vSphere, thin provisioning has been fully integrated into vCenter and is fully supported for both NFS and VMFS datastores.

The Problem

The basic dynamic which has led to the popularity of thin provisioning is that traditionally it has been very difficult to add additional storage to an existing allocation. Once a system is up and running in production, adding additional storage often requires application and system downtime. To overcome this challenge, the application users started requesting far more storage than actually needed. This lead to wasted storage space as the amount allocated was often much larger than what was really required. This leads to wasted storage being allocated but not used and in many cases never will be used.

Lead time and difficulty in getting additional storage also increased frustration. This resulted in application administrators inflating the size of their requests in hopes that they would not have to ask for more storage resource in the future. Application administrators often double the amount of space they requested to avoid having to ask for additional allocations in the future. This results in low utilization rates in many layers of the storage allocation process and reduces the overall efficiency of storage resources.

Thin Provisioning Terminology

When thin provisioning is discussed, it is important to define a few key terms to avoid confusion. These metrics are critical to managing an environment where thin provisioning is used. Similar to the airline scenario explained earlier, a virtual administrator needs to know the **capacity** (how many seats are on the aircraft), the total **committed** (how many reservations have been made for that flight) and the amount of **space used** (how many people have checked in for the flight).

Additionally, it is important to know that the percentage of allocations is relative to capacity. When the number is greater than 1.0, you have an overcommitted or "overbooked" situation. As that overcommitted number approaches a certain point, the chances of being caught short on space needed to accommodate the demand will increase. When space used approaches capacity, one needs to address a pending out of space situation and take steps to avoid being caught short.

TERM	DEFINITION
Capacity	Total amount of space in datastore
Committed	Total amount of space reserved for virtual machines in that datastore
Used	Total amount of space used by virtual machines
Overcommitted	Situation that occurs when committed/reserved space for virtual machines exceeds the capacity of the datastore
Percent Overcommitted	The degree to which committed allocations exceed capacity

Table 1. Terms and definitions for virtual disk thin provisioning

How Virtual Disk Thin Provisioning Works

When a virtual disk is created in a VMware ESX™ server environment, there is a file created within a datastore. The format of the virtual disk will be Zeroed Thick format by default. However two additional options exist: One is for that file to be eager Zeroed Thick format, and the other is for it to be in thin format. The difference between Eager Zeroed Thick and Zeroed Thick is related to the space allocated having blocks written with all zeros. The Eager Zeroed Thick format virtual disk will have this done for the entire space requested for that virtual disk at the time it is created. This process can take a considerable amount of time for really large VMDKs.

The Zeroed Thick will zero out only a single VMFS block at a time. And as the virtual disk space is filled out, it will incrementally zero out a block at a time as it grows. Both thick virtual disk formats will have a continuous chunk of a given datastore as the space is allocated and reserved at the time the VMDK was created.

The thin format virtual disk gets a reservation for space, but all that space is not dedicated to the virtual disk until the space is needed to store additional data. This growth of the thick format virtual disk is incremental to that of the VMFS volume block size. The default value is 1MB and is set at the time the VMFS volume is created. Once set, this value cannot be subsequently changed. See table 2 for details on these steps.

VMDK FORMAT	SPACE DEDICATED	ZEROED OUT BLOCKS	INCREMENTAL GROWTH
Thin	Reservation only	As it grows	1 VMFS block at a time
Zeroed Thick	Full amount	As it grows	No
Eager Zeroed Thick	Zeroed Thick	At creation time	No

Table 2. Preparing datastore space for use by a virtual disk

As there can be a slight delay caused by the time it takes to zero out a new block of a virtual disk, the new VMware® Fault Tolerance capability requires use of Eager Zeroed Thick format disks. So use of that feature is not compatible with thin format virtual disks. In other words, you can not use Fault Tolerance on thin format virtual disks. However you could use Fault Tolerance on array-based thin provisioned disks.

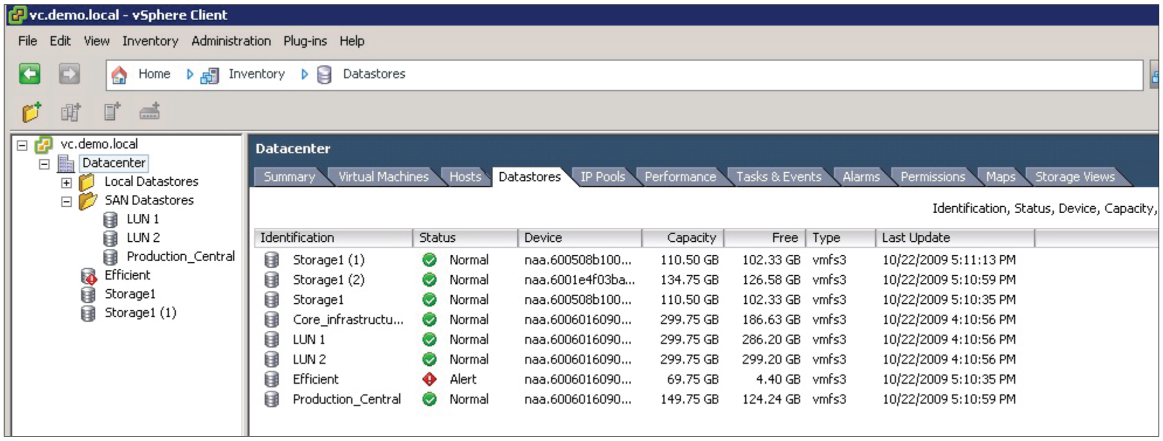
Managing Virtual Disk Thin Provisioning

When thin provisioning is used for virtual disks, it is important to keep track of your free space in your storage resource pool (i.e. datastore) to ensure you are not running out of space. If you have a pending out-of-space issue, best practice is to take steps to either free up space in the existing resource pool or increase the size of that pool before application disruption or downtime occur. To avoid disruption, it is best practice to track space usage over time and set alerts or alarms that will call attention to a pending out of space issue.

VMware vCenter™ has several new capabilities that help manage virtual disk thin provisioning. The ability to create thin virtual disks when creating, cloning, deploying, and migrating virtual machines are only part of the management interface. The tracking of datastore usage, the raising alerts if nearing a full status, as well as providing some new means to alleviate a pending out of space issues are also key features for management.

Datastore Views

In vSphere, datastores are now managed as an object in vCenter. One can now look at a listing of all datastores in a virtualization environment managed by vCenter. This listing of datastores can also be grouped into folders and arranged hierarchically. Sorting datastores by departments, protocols, or different levels of service quality is also now an option.

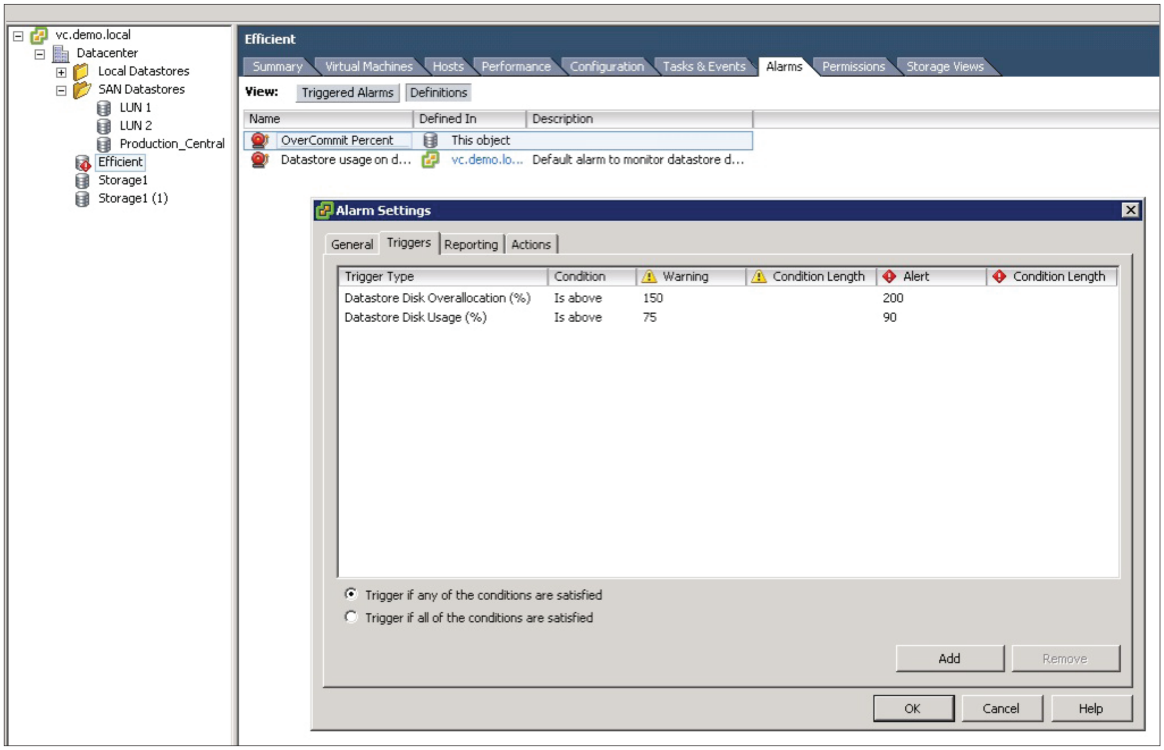


Identification	Status	Device	Capacity	Free	Type	Last Update
Storage1 (1)	Normal	naa.600508b100...	110.50 GB	102.33 GB	vmfs3	10/22/2009 5:11:13 PM
Storage1 (2)	Normal	naa.6001e4f03ba...	134.75 GB	126.58 GB	vmfs3	10/22/2009 5:10:59 PM
Storage1	Normal	naa.600508b100...	110.50 GB	102.33 GB	vmfs3	10/22/2009 5:10:35 PM
Core_infrastructu...	Normal	naa.6006016090...	299.75 GB	186.63 GB	vmfs3	10/22/2009 4:10:56 PM
LUN 1	Normal	naa.6006016090...	299.75 GB	286.20 GB	vmfs3	10/22/2009 4:10:56 PM
LUN 2	Normal	naa.6006016090...	299.75 GB	299.20 GB	vmfs3	10/22/2009 4:10:56 PM
Efficient	Alert	naa.6006016090...	69.75 GB	4.40 GB	vmfs3	10/22/2009 5:10:35 PM
Production_Central	Normal	naa.6006016090...	149.75 GB	124.24 GB	vmfs3	10/22/2009 5:10:59 PM

For each datastore, there are a set of detail screens that reflect the topology, events, performance, and summary information. This enables the VI admin to see a listing of which ESX servers or virtual machines are accessing a given datastore.

Automated Alarms

One of the most critical new additions for managing environments that leverage thin provisioning is the ability to set alerts and alarms at the datastore and virtual machine levels. These alarms can be customized to trigger alerts upon certain thresholds so that a VI Admin can take action required to prevent an out-of-space issue. Alarms can be set on a datastore for a percentage full—as well as a percent overcommitted.



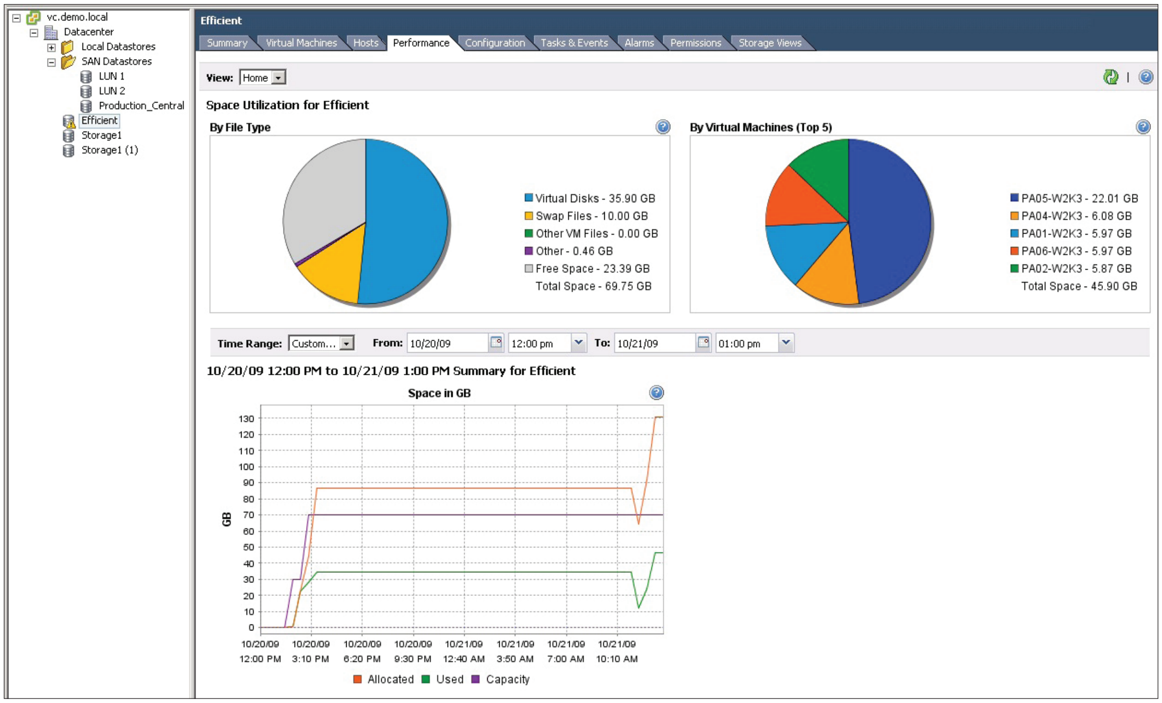
Name	Defined In	Description
OverCommit Percent	This object	
Datastore usage on d...	vc.demo.lo...	Default alarm to monitor datastore d...

Trigger Type	Condition	Warning	Condition Length	Alert	Condition Length
Datastore Disk Overallocation (%)	Is above	150		200	
Datastore Disk Usage (%)	Is above	75		90	

● Trigger if any of the conditions are satisfied
○ Trigger if all of the conditions are satisfied

The automated alarms will trigger a warning alert when used space is 75% of capacity for a given datastore and a red alert when used space is 85% of capacity by default. Another alert which should be defined when using thin provisioned virtual disks in a datastore is the percent-over-commit alarm. It is not an alarm that is on a datastore by default. It is recommended that it be added and customized to trigger a warning alarm at 150%, and a red alert at 200%. However, these numbers can be adjusted to help the admin decide how much overcommitment they are comfortable with on each individual datastore. In the example screenshot, both these alarms have been triggered.

Permissions can also be set on a datastore to provide control over which users can and can not provision space in a given datastore. These enhancements in vCenter are a key capability for VMware being able to offer full support for thin provisioned virtual disks in vSphere.



View: Home

Space Utilization for Efficient

By File Type

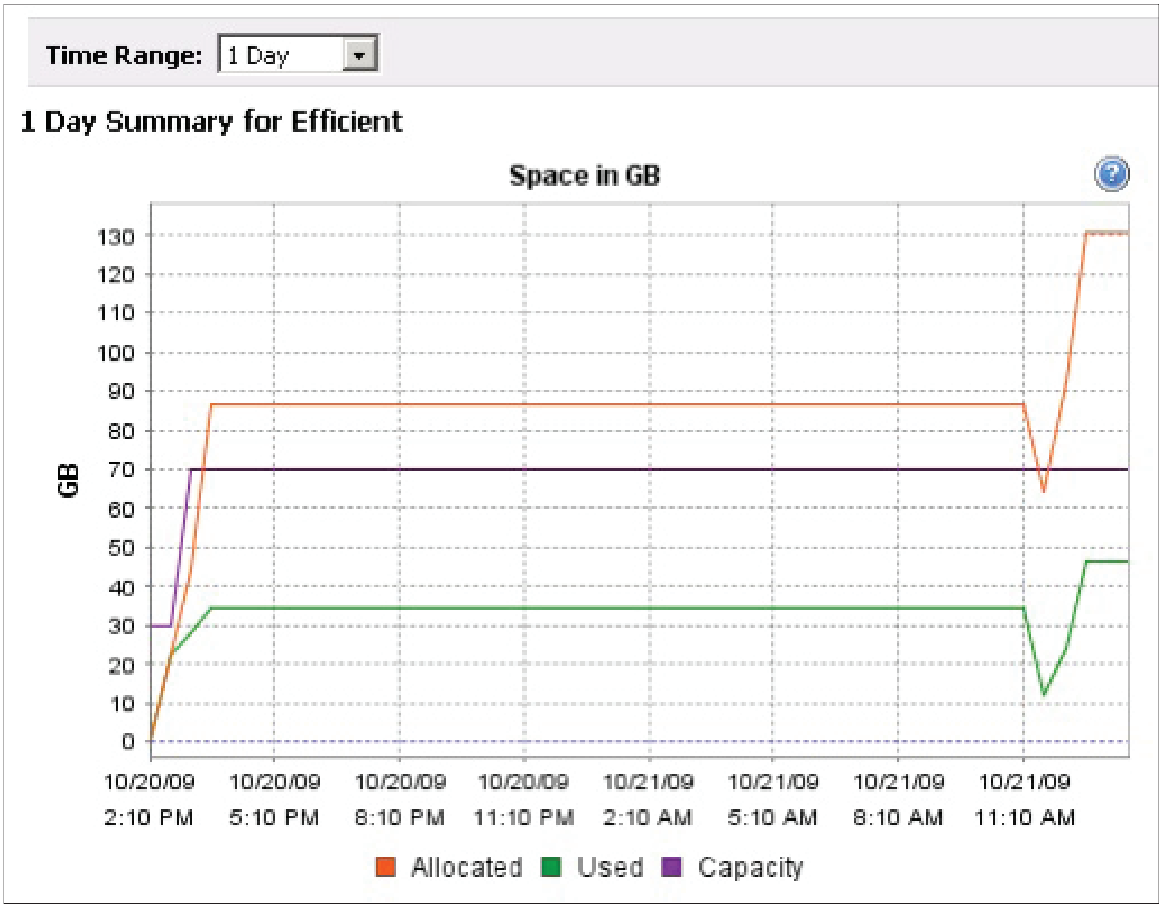
By Virtual Machines (Top 5)

Time Range: Custom... From: 10/20/09 12:00 pm To: 10/21/09 01:00 pm

10/20/09 12:00 PM to 10/21/09 1:00 PM Summary for Efficient

Space in GB

A closer look at the previous screenshot shows a new view that provides a single picture of how full – as well as overcommitted – a specific datastore is over time. The view reflects the capacity of a datastore in purple, the amount of that space that is used in green, and what has been allocated in orange. When the orange line is below the purple you have no overcommitment in the datastore. When the space used begins to approach the capacity, it is time to consider handling a pending out-of-space condition.



In the time range shown in the screenshot above, one can see the reflection of:

- Creating a 30GB datastore
- Moving a thick provisioned virtual machine into the datastore with VMware Storage VMotion™
- Increasing the size of the datastore to 70GB with VMFS Volume Grow
- Moving another virtual machine with SVM that was thin provisioned
- Using SVM to move the thin provisioned virtual machine out of the datastore
- Adding a few more thin provisioned virtual machines onto this datastore

Without such visibility into space allocations and the ability to raise alerts, thin provisioned virtual disks would be much more difficult to manage. The risk of running out of space in a datastore would be much greater without such insight and automated alarms. With these new datastore level views, usage metrics, and alerts, it is now much safer to use thin provisioning to overcommit space in a given datastore.

Options for Handling Overcommitted Datastores

If a datastore is overcommitted and the used space is approaching the capacity of that datastore, it is best to have a few options available to avoid being caught short and also have enough time to execute one of those options before the datastore fills up. Two basic options exist to address this pending out-of-space issue in vSphere:

1) Move some virtual machines out of the datastore to make more space

2) Increase the size of the existing datastore

Freeing Up Space in a Datastore

To make more space in an existing datastore, one might delete virtual machines or files that are no longer needed. One could also consolidate VMFS snapshots and merge multiple point-in time copies into the parent image to free up some space. Deleting un-needed virtual machines might also free up space in the datastore. One can also cold migrate virtual machines that are not currently up and running to another datastore. Or one might choose to use VMotion to move an up and running virtual machine to another datastore to free up space.

VMotion has been enhanced in vSphere to include integration with vCenter, and is now fully supported for migration from and to datastores of multiple protocols. This means that one can now migrate from and to FC, iSCSI, and NFS datastores and in doing so move between different storage vendors. In addition, one can change a virtual disk from thick to thin or from thin to thick format when executing VMotion.

Increasing Capacity of a Datastore

There are two ways to grow the capacity of an existing VMFS datastore:

1) Add another extent to the VMFS Volume and have the volume span multiple extents

2) Expand the existing VMFS Volume on additional space using the new feature in vSphere called VMFS Volume Grow

Adding another extent to an existing VMFS Volume is a capability introduced in release 3 of the ESX Server. This is also referred to as spanning a VMFS Volume across more than one LUN. This process makes use of additional capacity by way of VMFS acting as a volume manager. The additional extent (LUN) is then concatenated to the first LUN, or LUNs if the VMFS Volume already spanned more than one LUN.

VMFS Volume Grow was introduced in vSphere and provides a new means of increasing an existing VMFS Volume onto either a recently increased capacity of an existing LUN, or space not used when the VMFS Volume was originally created. The latter would not be considered best practices as it would be more common to have a VMFS volume/datastore occupy the full capacity of a LUN upon which it resides.

Dynamic LUN Expansion is becoming a common feature of most storage arrays. And VMFS Volume Grow complements the underlying disks array dynamic expansion capability.

Performance Considerations

When considering use of thin provisioned virtual disks there are often concerns about its impact on overall virtual machine performance. These concerns often stem from the perception that fragmentation may reduce performance when the entire disk is scanned as well as more frequent updates to the metadata updates may cause increased contention. The next portion will address why both those concerns should not prevent one from considering use of thin virtual disks.

Fragmentation

The concern of fragmentation stems from additional allocations of space being assigned to the growing thin provisioned virtual disk not being as localized as with a thick disk. As a thin provisioned virtual disk grows, an incremental addition of one block of space is assigned. The size of that additional block will be equal to what the block size is for the VMFS volume it resides in. The default setting is 1MB but can be configured at the time the VMFS Volume is created to be up to 8MB. As there may be many thin provisioned virtual disks all growing within the same datastore, the collection of blocks that make up a single VMDK may be scattered about in that underlying datastore.

The concern is that as fragmentation increases, the access time to gather all the blocks that make up a VMDK will not be as fast as it would be when those blocks are found in a continuous section of the datastore storage. However as the allocation unit of 1MB is fairly large, the impact of that scattering does not cause much degradation of performance. So although this is a factor to be considered, it is relatively small. Currently there is no tool to measure the degree of fragmentation that exists in vSphere. And the only utility to defragment a VMDK file is VMotion – to move the VMhome to another datastore and then SVM it back to the original datastore.

Increased Metadata Updates

The issue of contention results from locking done to ensure that each new allocation for a new block does not compromise the integrity of the clustered files system free space. Each time VMFS allocates a new block from free space to a virtual machine, it needs to make sure that another ESX server for another virtual machine is not also allocating to that block. The mechanism that VMFS uses to ensure this integrity of free space allocation is to issue a SCSI reservation on the LUN that the datastore resides on while the VMFS metadata is updated by a single ESX server to reflect the allocation of a specific free space to a give VMDK. That process will be executed more frequently when many thin provisioned virtual disks are growing rapidly across a cluster of many ESX servers. To offset the additional metadata updates which thin provisioning can introduce, there have been several improvements delivered in vSphere that reduce the overall need to take a SCSI reservation as well as the time those reservations take to complete. VMware has added additional efficiencies to make the increase in the need to manage allocations from the shared storage free space pool less of a bottleneck for performance.

Although VMware does not have any specific benchmark numbers or performance studies that quantify the effects of either fragmentation or more frequent updates to manage the free space pool when using thin provisioned virtual disks, it is believed that although there might be a slight overhead associated with thin provisioning, it will not be significant.

Where to Thin Provision

Thin provisioning can be leveraged within the storage array as well as at the virtual disk level. Some common questions are: Where is it best to use thin provisioning and can it be used at both levels at the same time? The answers are that the ESX server does know if the underlying LUN is thin provisioned or not. So if it is managed correctly at both levels it can be complementary to have thin provisioning at both the array and virtualization levels.

If you consider that the purpose of thin provisioning is to reduce the amount of wasted space, then Storage Administrators can reduce wasted storage by using thin provisioning at the array level while at the same time VI Administrators can use thin provisioning at the virtual disk level. These are often two separate administration teams using separate management interfaces to maintain the use of thin provisioning at their respective levels.

There are some cases where people feel it would be simpler to use one or the other but not both at the same time. This often is a result of them fearing that free space might be tough to measure. And they may worry about it not being as efficient as it might otherwise be in terms of knowing what free space exists as well as having to keep track of overcommitment in more than one location. However, as there are normally two separate administration teams managing their respective resource pools this confusion should be minimal.

If all of the storage in a single array is used for virtualization environments then one might argue that the use of both thin provisioning at the array and the virtual disk levels might not yield as much space savings beyond what would result from using just the array-based thin provisioning. However, one should consider that wasted space in the guest OS use of storage is what is saved by over committing a datastore. And that datastore will grow over time on the underlying storage allocation. So use of both array-based and virtual disk thin provisioning is complimentary.

One thing which virtual disk thin provisioning does not support currently is the ability to reclaim free space when the GOS deletes a large file. Once written, there is no real way to mark that space as free space so that the virtual disk could compress and free up a block it no longer needs. That would require an interface with the GOS that is not currently supported as an industry standard.

But the ability to dynamically grow a virtual disk while the virtual machine is up and running is supported with vSphere. The hot virtual disk extend feature enables expanding the virtual disk through the "edit settings" option for that virtual machine. And as long as the GOS can support dynamic LUN expansion, hot virtual disk expand can be done while the virtual machine is up and running. This enables dynamic growth of storage needs for the virtual machine and with that there is less need to oversize a virtual disk to ask for more space.

Conclusion

With vSphere, one has many new choices for virtual machine layout and the opportunity to increase the efficiency of the storage resources. Thin provisioning is an option that is fully supported in vSphere. And it provides new ways to dynamically manage storage resources for your virtualization environment.

This paper has placed the use of thin provisioned disks in context of how the ESX server and more specifically how VMFS manages storage for virtualization environments. It has explained the concepts of allocation, preparation, and incrementally doing those steps can increase storage utilization and efficiency. It has added context around considerations and trade offs that should be factored when considering use of thin provisioned disks.

Thin provisioning won't be an ideal solution for every environment. But when used and managed correctly, it does offer significant benefits to providing efficient and dynamic space allocation that improves the overall operations for many virtual administrators—helping them solve their storage management challenges with new levels of efficiency.

About the Author

Paul Manning is senior technical marketing manager at VMware, where he specializes in virtual storage management. Previously, he worked at EMC and Oracle, where he had more than ten years' experience designing and developing storage infrastructure and deployment best practices. He has also developed and delivered training courses on best practices for highly available storage infrastructure to a variety of customers and partners in the United States and abroad. He is the author of numerous publications and presented many talks on the topic of best practices for storage deployments and performance optimization.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com