White Paper

Implementing EMC Symmetrix Virtual Provisioning with VMware vSphere

Applied Technology

Abstract

This white paper provides a detailed description of the technical aspects and benefits of deploying VMware vSphere version 4 on EMC® Symmetrix DMX™ or Symmetrix® VMAX™ devices using Virtual Provisioning™. This paper also includes an overview of the latest Symmetrix Virtual Provisioning features and best practice recommendations for deploying virtual machines onto Symmetrix thin devices.

October 2010

Part Number h6813.2

Table of Contents

Executive summary

One of the biggest challenges facing storage administrators is provisioning storage for new applications. Administrators typically allocate space based on anticipated future growth of applications. This is done to mitigate recurring operational functions, such as incrementally increasing storage allocations or adding discrete blocks of storage as existing space is consumed. Using this approach results in more physical storage being allocated to the application than needed for a significant amount of time and at a higher cost than necessary. The overprovisioning of physical storage also leads to increased power, cooling, and floor space requirements. Even with the most careful planning, it may be necessary to provision additional storage in the future, which could potentially require an application outage.

A second layer of storage overprovisioning happens when a server and application administrator overallocate storage for their environment. The operating system sees the space as completely allocated but internally only a fraction of the allocated space is used.

EMC Virtual Provisioning™ can address both of these issues. Virtual Provisioning, available starting with Enginuity™ release 5773, allows more storage to be presented to an application than is physically available. More importantly, Virtual Provisioning will allocate physical storage only when the storage is actually written to. This allows more flexibility in predicting future growth, reduces the initial costs of provisioning storage to an application, and can obviate the inherent waste in overallocation of space and administrative management of subsequent storage allocations.

EMC® Symmetrix DMX™ and Symmetrix® VMAX™ storage arrays continue to provide increased storage utilization and optimization, enhanced capabilities, and greater interoperability and security. The implementation of Virtual Provisioning, generally known in the industry as “thin provisioning,” for these arrays enables organizations to improve ease of use, enhance performance, and increase capacity utilization for certain applications and workloads.

VMware’s vSphere 4 virtualization suite now offers vStorage thin provisioning. Symmetrix Virtual Provisioning occurs at the array level, while VMware thin provisioning occurs on the virtual disk level. The host-level thin provisioning feature allows vSphere administrators to gain the ability to allocate virtual disk files as “thick” or “thin.” Thin provisioning of virtual disks allows virtual machines on VMware ESX hosts to provision the entire space required for the disk’s current and future activities, but at first commits only as much storage space as the disk needs for its initial operation. Thin provisioning enables organizations to provision heterogeneous storage pools, increase utilization, and reduce administration costs.

Introduction

This white paper will discuss how to best leverage Symmetrix Virtual Provisioning in a VMware vSphere environment. Symmetrix Virtual Provisioning is discussed as well as an overview of VMware vSphere 4. This paper primarily addresses the considerations and best practices for deploying VMware vSphere version 4 environments on thinly provisioned devices.

An understanding of the principles that are exposed here will allow the reader to deploy VMware vSphere environments with Virtual Provisioning in the most effective manner.

Audience

This white paper is intended for storage architects, and server and VMware administrators responsible for deploying VMware vSphere on Symmetrix DMX-3 and DMX-4 using the Enginuity 5773 operating environment, as well as on the Symmetrix VMAX using the Enginuity 5874 operating environment. Virtual Provisioning is now free for all supported Symmetrix platforms, new and existing, which includes all Symmetrix VMAX arrays plus DMX-3 and DMX-4 arrays running Enginuity 5773.

Terminology

Table 1. Basic Symmetrix array terms

Term	Description
Device	A logical unit of storage defined within a Symmetrix array. In this paper this style of device is also referred to as a traditional device and a non-thin device
Device Capacity	The actual storage capacity of a device
Device Extent	Specifies a quantum of logical contiguous block of storage
Host Accessible Device	A device that is exported for host use
Internal Device	A device used for internal function of the array
Metavolume	An aggregation of host accessible devices, seen from the host as a single device
Pool	A collection of internal devices for some specific purpose

Table 2. Virtual Provisioning terms

Term	Description
Thin Device	A host accessible device that has no storage directly associated with it
Data Device	An internal device that provides storage capacity to be used by thin devices
Thin Device Extent	The minimum quantum of storage that must be mapped at a time to a thin device
Thin Pool	A collection of data devices that provide storage capacity for thin devices. It should be noted that a thin pool can exist without any storage associated to it.
Thin Pool Capacity	The sum of the capacities of the member data devices
Bind	The process by which one or more thin devices are associated to a thin pool
Unbind	The process by which a thin device is diassociated from a given thin pool. When unbound, all previous allocated extents are freed and returned to the thin pool
Enabled Data Device	A data device belonging to a thin pool from which capacity is allocated for thin devices. This state is under user control
Disabled Data Device	A data device belonging to a thin pool from which capacity cannot be allocated for thin devices. This state is under user control
Activated Data Device	Reads and writes can be done from allocated and unallocated space on activated data devices.
Deactivated Data Device	Reads and writes can be done from already allocated space on deactivated data devices. No new allocations can be done on deactivated data devices.
Data Device Drain	The process of removing allocated extents off of one data device and moving them to another enabled data device in the same thin pool
Thin Pool Enabled Capacity	The sum of the capacities of enabled data devices belonging to a thin pool
Thin Pool Allocated Capacity	A subset of thin pool enabled capacity that has been allocated to thin devices bound to that thin pool
Thin Device Written Capacity	The capacity on a thin device that was written to by a host. In most implementations this will be a large percentage of the thin device allocated capacity
Thin Device Subscribed Capacity	The total amount of space that the thin device would use from the pool if the thin device were completely allocated
Oversubscribed Thin Pool	A thin pool whose thin pool enabled capacity is less than the sum of the reported sizes of the thin devices using the pool
Thin Device Allocation Limit	The capacity limit that a thin device is entitled to withdraw from a thin pool, which may be equal to or less than the thin device subscribed capacity

Term	Description
Thin Device Subscription Ratio	The ratio between the thin device subscribed capacity and the thin pool enabled capacity. This value is expressed as a percentage
Thin Device Allocation Ratio	The ratio between the thin device allocated capacity and the thin device subscribed capacity. This value is expressed as a percentage
Thin Device Utilization Ratio	The ratio between the thin device written capacity and thin device allocated capacity. This value is expressed as a percentage
Thin Pool Subscription Ratio	The ratio between the sum of the thin device subscribed capacity of all its bound thin devices and the associated thin pool enabled capacity. This value is expressed as a percentage
Thin Pool Allocation Ratio	The ratio between the thin pool allocated capacity and thin pool enabled capacity. This value is expressed as a percentage
Pre-allocating	Allocating a range of extents from the thin pool to the thin device at the time the thin device is bound. Sometimes used to reduce the operational impact of allocating extents or to eliminate the potential for a thin device to run out of available extents

EMC Symmetrix Virtual Provisioning

Symmetrix thin devices are logical devices that can be used in many of the same ways that Symmetrix devices have traditionally been used. Unlike traditional Symmetrix devices, thin devices do not need to have physical storage completely allocated at the time the device is created and presented to a host. A thin device is not usable until it has been bound to a shared storage pool known as a thin pool. Multiple thin devices may be bound to any given thin pool. The thin pool is comprised of devices called data devices that provide the actual physical storage to support the thin device allocations.

When a write is performed to a part of any thin device for which physical storage has not yet been allocated, the Symmetrix allocates physical storage from the thin pool for that portion of the thin device only. The Symmetrix operating environment, Enginuity, satisfies the requirement by providing a unit of physical storage from the thin pool called a thin device extent. This approach reduces the amount of storage that is actually consumed.

The thin device extent is the minimum amount of physical storage that can be reserved at a time for the dedicated use of a thin device. An entire thin device extent is physically allocated to the thin device at the time the thin storage allocation is made as a result of a host write operation. A round-robin mechanism is used to balance the allocation of data device extents across all of the data devices in the pool that are enabled and that have remaining unused capacity. The thin device extent size is 12 tracks (768 KB). Note that the initial bind of a thin device to a pool causes one thin device extent, or 12 tracks, to be allocated per thin device. If the thin device is a metavolume, then one thin device extent is allocated per meta member device. So a four-member thin meta would cause 48 tracks (3078 KB) to be allocated when the device is bound to a thin pool.

When a read is performed on a thin device, the data being read is retrieved from the appropriate data device in the thin pool. If a read is performed against an unallocated portion of the thin device, zeros are returned to the reading process.

When more physical data storage is required to service existing or future thin devices, for example, when a thin pool is running out of physical space, data devices can be added to existing thin pools dynamically without needing a system outage[1]. New thin devices can also be created and bound with existing thin pools.

When data devices are added to a thin pool they can be in an enabled or disabled state. In order for the data device to be used for thin extent allocation it needs to be enabled. For it to be removed from the thin pool,

[1] Enginuity operating environment 5874 SR1 on the Symmetrix VMAX and Solutions Enabler 7.1 allow users to rebalance extents over the thin pool when new data devices are added. This feature is discussed in detail in the next section.

it needs to be in a disabled state. Beginning with Enginuity 5874, active data devices can be disabled, which will cause any allocated extents to be drained to the other enabled devices in the pool. They can then be removed from the pool when the drain operation has completed.

The following figure depicts the relationships between thin devices and their associated thin pools. There are nine devices associated with thin Pool A and three thin devices associated with thin pool B.

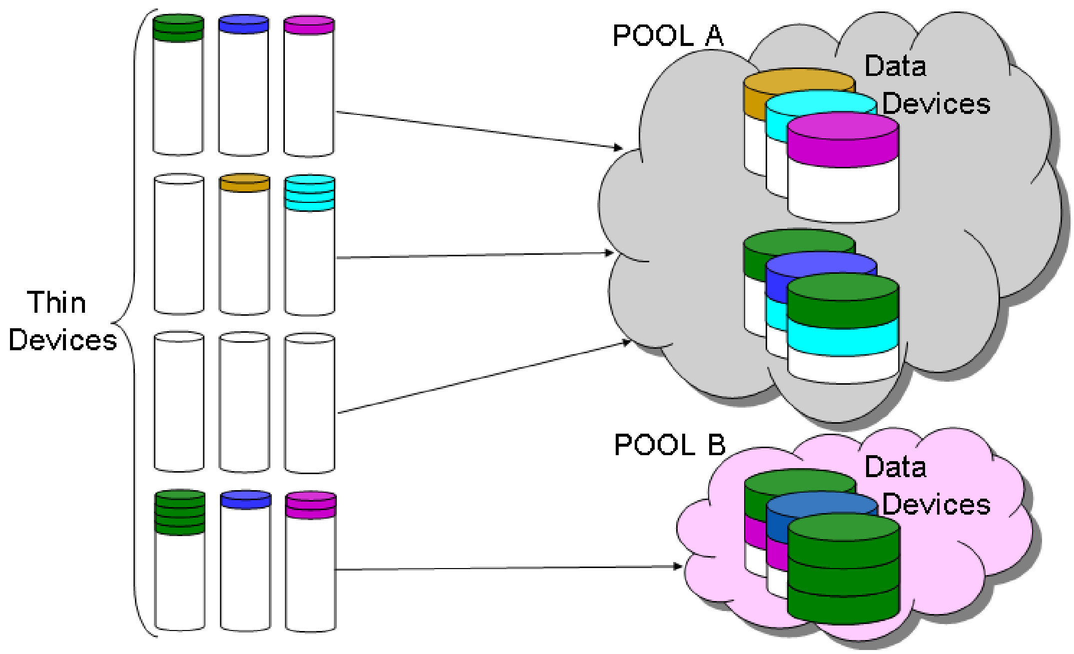


Figure 1. Thin devices and thin pools containing data devices

The way thin extents are allocated across the data devices results in a form of striping in the thin pool. The more data devices that are in the thin pool, the wider the striping is, and therefore the greater the number of devices that can participate in application I/O.

The maximum size of a thin device on a Symmetrix DMX is approximately 60 GB, which has been increased to 240 GB for the Symmetrix VMAX. If a larger size is needed, then a metavolume comprised of thin devices can be created. It is recommended that the metavolume be concatenated rather than striped since the data is already striped across the data devices in the thin pool, which may also be striped across physical disks depending on the RAID protection of the data devices. Concatenated metavolumes also support fast expansion capabilities, as new metavolume members can easily be added to the existing concatenated metavolume. This functionality is required when the provisioned thin device has been consumed by the host, and further storage allocations are required.

Concatenated metavolumes using regular volumes have the drawback of being limited by the performance of a single metamember. In the case of a concatenated metavolume comprised of thin devices, each member device is typically spread across the entire underlying thin pool, thus eliminating that drawback.

Concatenated metavolumes have two important operational advantages over striped metavolumes:

- Concatenated metavolumes can be expanded without destroying existing data by adding metavolume members to an existing metavolume.
- Non-metavolumes can be converted to concatenated metavolumes without destroying existing data.

In most cases, EMC recommends using concatenated rather than striped metavolumes with Virtual Provisioning. However, there may be certain situations where better performance may be achieved using striped metavolumes.

With Synchronous SRDF®, Enginuity allows one outstanding write per thin device per path. With concatenated metavolumes, this could cause a performance problem by limiting the concurrency of writes. This limit will not affect striped metavolumes in the same way because of the small size of the metavolume stripe (1 cylinder or 1920 blocks).

Enginuity allows eight read requests per path per thin device. This limits the number of read requests that can be passed through to the thin pool regardless of the number of data devices that may be in it. This can cause slower performance in environments with a high read miss rate.

Symmetrix Enginuity has a logical volume write pending limit to prevent one volume from monopolizing writeable cache. Because each meta member gets a small percentage of cache, a striped meta is likely to offer more writable cache to the meta volume.

Before configuring striped metavolumes, please consult with an EMC performance specialist.

Caution: Striped thin metavolumes cannot be expanded while preserving data.

A more detailed discussion of Symmetrix Virtual Provisioning is beyond the scope of this paper. For more information please refer to the technical note *Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning* available on Powerlink.

New features

Solutions Enabler 7.1 and Enginuity 5874 SR1 introduce two new features to Symmetrix Virtual Provisioning – thin pool write balancing and zero space reclamation. Thin pool write balancing provides the ability to automatically rebalance allocated extents on data devices over the entire pool when new data devices are added. Zero space reclamation allows users to reclaim space from tracks of data devices that are all zeros.

Thin pool write balancing

Whenever writes occur on a thin device, the actual data gets written to the data devices within the thin pool to which the thin device is bound. Enginuity intelligently spreads out the I/O equally on all the data devices within the thin pool, so at any given moment the used capacity of each data device is expected to be uniform across the pool.

Prior to the availability of automated thin pool write balancing, as soon as new data devices were added to the thin pool, the allocation spread immediately ceased to be uniform. If data devices were added in groups that were smaller in number than what were previously in the thin pool, performance decreased due to the fact that as soon as the original data devices became completely full, any new data written to the pool was striped over fewer data devices.

Therefore, the best practice was to add a large number of data devices, typically equal to or greater than the number of devices used when the pool was originally created to maintain expected performance levels. This method of expanding pools using larger numbers of devices led to an increase in assigned, but unused, storage as in most situations the thin pool did not need to be expanded by that much to fulfill the increased thin pool storage requirements.

Automated thin pool write balancing alleviates both of these problems. It is now possible to add any number of data devices and still have a pool that is balanced for capacity while preventing potential performance problems due to narrow striping that previously would be caused by adding too few data devices.

Thin pool write balancing works by normalizing the used capacity levels of data devices within a thin pool. A background optimization process scans the used capacity levels of the data devices within a pool and performs movements of groups from the most utilized devices to the least. The threshold for write balancing of 10% has two implications — that there can be up to a 10% difference between the most used, and least used data devices after a rebalance operation completes and that if the thin pool is within this range from the initiation of the balancing command, the balancing operation will not start.

The basic steps taken by thin pool write balancing are as follows:

- Initialization – the target resources are locked so that no other process can try to optimize them
- Scan – the pool is scanned for utilization statistics
- Move – the group movements are initiated between devices
- Wait – the task monitors the background data movements until the next period begins

The balancing procedure will start by calculating the minimum, maximum, and mean used capacity values of the data devices. Afterward, a source/target data device pair is automatically created by Enginuity to bring the source data device's utilization down and the target data device's utilization up toward the mean. Once the data movements have finished, the process restarts by scanning the pool again. Data devices that are currently a source or a target in the pool balance operation will be shown to have a state of "Balancing." While a data device is in this state, it cannot be disabled or drained. The "Balancing" state will be transient as Enginuity will pick different devices periodically to accomplish the pool balancing operation.

While write balancing is going on, it is possible that the data can be moved from "deactivated" data devices but data will never be moved to "deactivated" data devices. Solutions Enabler will report the state "Balancing" in the pool state and once the task is completed, the pool state will revert to the default state of "Enabled". For more detailed information on how to perform thin pool write balancing, please refer to the technical note *Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning.*

Symmetrix thin pool write balancing using Solutions Enabler

In the following example a thin pool containing four data devices is completely full and requires more data devices to satisfy the storage needs of the bound thin devices. Figure 2 shows such a thin pool.

```
Symmetrix ID: 000194900080

Symmetrix ID                    : 000194900080
Pool Name                       : VP_Rebalance
Pool Type                       : Thin
Dev Emulation                   : FBA
Dev Configuration               : RAID-5(3+1)
Pool State                      : Enabled
# of Devices in Pool            :    4
# of Enabled Devices in Pool    :    4
Max. Subscription Percent       : None

Enabled Devices(4):
  {
   -----------------------------------------------------
   Sym         Total       Alloc        Free Full Device
   Dev           GBs         GBs         GBs  (%) State
   -----------------------------------------------------
   0313         10.0        10.0         0.0  100 Enabled
   0314         10.0        10.0         0.0  100 Enabled
   0315         10.0        10.0         0.0  100 Enabled
   0316         10.0        10.0         0.0  100 Enabled
          ---------- ----------- ----------- ----
   GBs          40.0        40.0         0.0  100
  }
```

Figure 2. Full Symmetrix thin pool before a write balance operation

To rectify this, as seen in Figure 3, two more data devices are added to the pool. Since the four previous data devices are completely full, any new writes to the thin pool will only be striped over the two recently added data devices, possibly reducing performance. Initiating the rebalance procedure shown in Figure 4 on page 10 will remediate this issue. In addition, the lower portion of Figure 4 shows the pool state change to "Balancing," which reports that the pool is now amidst being balanced.

```
Enabled Devices(6):
  {
   ------------------------------------------------------
   Sym         Total      Alloc         Free Full  Device
   Dev           GBs        GBs          GBs  (%)  State
   ------------------------------------------------------
   0313         10.0       10.0          0.0  100  Enabled
   0314         10.0       10.0          0.0  100  Enabled
   0315         10.0       10.0          0.0  100  Enabled
   0316         10.0       10.0          0.0  100  Enabled
   0317         10.0        0.0         10.0    0  Enabled
   0318         10.0        0.0         10.0    0  Enabled
          ---------- ---------- ---------- ----
   GBs          60.0       40.0         20.0   66
  }
```

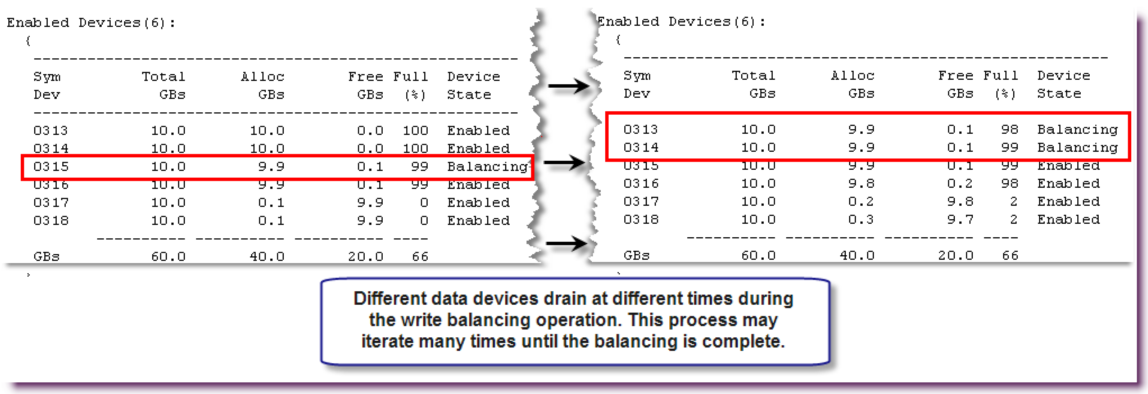
Figure 3. A Symmetrix thin pool with two new empty data devices before a rebalance operation

```
[root@licof016 root]# symconfigure -sid 80 -cmd "start balancing on pool VP_Reba
lance type=thin;" commit -nop -v

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    {
      START balancing on pool=VP_Rebalance type=thin;
    }

    Performing Access checks..................................Allowed.
    Checking Device Reservations..............................Allowed.
    Committing configuration changes..........................Reordering.
    Setting pool attributes .......
    Committing configuration change
    Terminating the configuration c

The configuration change session ha
```

```
Symmetrix ID: 000194900080

Symmetrix ID                      : 000194900080
Pool Name                         : VP_Rebalance
Pool Type                         : Thin
Dev Emulation                     : FBA
Dev Configuration                 : RAID-5(3+1)
Pool State                        : Balancing
# of Devices in Pool              :    6
# of Enabled Devices in Pool      :    6
Max. Subscription Percent         : None
```

Figure 4. Initiating a thin pool write balancing operation with Solutions Enabler

As pictured in Figure 5, while the balancing operation executes, varying data devices will go into the “Balancing” state temporarily as extents are drained off of them onto other less utilized data devices.

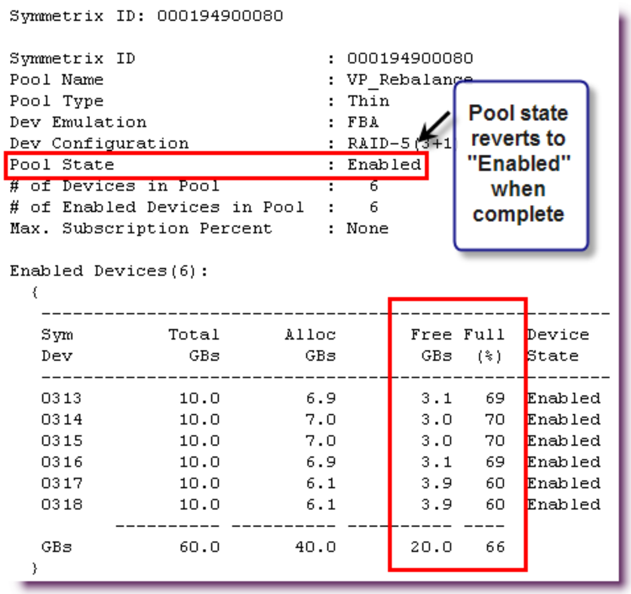


```
Enabled Devices(6):
  {
   ----------------------------------------------------
   Sym          Total      Alloc         Free Full  Device
   Dev            GBs        GBs          GBs  (%)  State
   ----------------------------------------------------
   0313          10.0       10.0          0.0  100  Enabled
   0314          10.0       10.0          0.0  100  Enabled
   0315          10.0        9.9          0.1   99  Balancing
   0316          10.0        9.9          0.1   99  Enabled
   0317          10.0        0.1          9.9    0  Enabled
   0318          10.0        0.1          9.9    0  Enabled
            ---------- ---------- ---------- ----
   GBs           60.0       40.0         20.0   66
  }
```

```
Enabled Devices(6):
  {
   ----------------------------------------------------
   Sym          Total      Alloc         Free Full  Device
   Dev            GBs        GBs          GBs  (%)  State
   ----------------------------------------------------
   0313          10.0        9.9          0.1   98  Balancing
   0314          10.0        9.9          0.1   99  Balancing
   0315          10.0        9.9          0.1   99  Enabled
   0316          10.0        9.8          0.2   98  Enabled
   0317          10.0        0.2          9.8    2  Enabled
   0318          10.0        0.3          9.7    2  Enabled
            ---------- ---------- ---------- ----
   GBs           60.0       40.0         20.0   66
  }
```


Figure 5. Thin pool write balancing and automatic data device draining

The thin pool write balancing procedure does not complete before the 10% differential threshold is met. The balanced pool can be seen in Figure 6.



```
Symmetrix ID: 000194900080

Symmetrix ID                        : 000194900080
Pool Name                           : VP_Rebalance
Pool Type                           : Thin
Dev Emulation                       : FBA
Dev Configuration                   : RAID-5(3+1)
Pool State                          : Enabled
# of Devices in Pool                :    6
# of Enabled Devices in Pool        :    6
Max. Subscription Percent           : None

Enabled Devices(6):
  {
   ------------------------------------------------------
   Sym          Total      Alloc       Free Full Device
   Dev            GBs        GBs        GBs  (%) State
   ------------------------------------------------------
   0313          10.0        6.9        3.1   69 Enabled
   0314          10.0        7.0        3.0   70 Enabled
   0315          10.0        7.0        3.0   70 Enabled
   0316          10.0        6.9        3.1   69 Enabled
   0317          10.0        6.1        3.9   60 Enabled
   0318          10.0        6.1        3.9   60 Enabled
            ---------- ---------- ---------- ----
   GBs           60.0       40.0       20.0   66
  }
```


Figure 6. Completed thin pool write balancing operation

Virtual Provisioning space reclamation

Virtual Provisioning space reclamation provides the ability to free, also referred to as "de-allocate," storage extents found to contain all zeros. This feature is an extension of the existing Virtual Provisioning space de-allocation mechanism. Previous versions of Enginuity and Solutions Enabler allowed for reclaiming allocated (reserved but unused) thin device space from a thin pool. Administrators now have the ability to reclaim both allocated/unwritten extents as well as extents filled with host-written zeros within a thin pool. The space reclamation process is nondisruptive and can be executed with the targeted thin device ready and read/write to operating systems and applications.

Starting the space reclamation process spawns a back-end disk director (DA) task that will examine the allocated thin device extents on specified thin devices. As discussed earlier, a thin device extent is 768 KB (or 12 tracks) in size and is the default unit of storage at which allocations occur. For each allocated extent, all 12 tracks will be brought into Symmetrix cache and examined to see if they contain all zero data. If the entire extent contains all zero data, the extent will be de-allocated and added back into the pool, making it available for a new extent allocation operation. An extent that contains any non-zero data is not reclaimed.

Symmetrix Virtual Provisioning space reclamation using Solutions Enabler

In the following example, the thin device in Figure 7 hosts one virtual machine with a 25 GB virtual disk. Only 1.5 GB of that virtual disk is actual data, while the remaining 23.5 GB is comprised completely of zeros.

```
Thin Devices(2):
  {
   ---------------------------------------------------------------------------
                        Pool            Pool              Pool
   Sym          Total  Subs         Allocated           Written
   Dev            GBs   (%)         GBs    (%)          GBs   (%)   Status

   0208          50.0    31        25.8     16         25.7    16   Bound
   0209          50.0    31         0.0      0          0.0     0   Bound
          ---------- ----- ---------- ----- ---------- -----
  GBs           100.0    63        25.8     16         25.7    16
  }
```

Figure 7. The allocation of a Symmetrix thin device before undergoing space reclamation

This situation is ideal for space reclamation as over 90% of the allocated and written space from this thin device can be reclaimed and returned to the thin pool as free space. In Figure 8, the space reclamation process is initiated through the Solutions Enabler command line interface. Enginuity is told to examine the entire thin device for any extent full of zeros and to then deallocate them.

```
[root@licof016 root]# symconfigure -sid 80 -cmd "free tdev 208 start_cyl=0, end_cyl=54613, type=reclaim;" -v commit -nop

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    Processing symmetrix 000194900080
    {
      free tdev 0208 start_cyl =0 cyl size =54614 cyl type =zero;
    }

    Performing Access checks..................................Allowed.
    Checking Device Reservations..............................Allowed.
    Locking devices...........................................Locked.
    Committing configuration changes..........................Started.
    Committing configuration changes..........................Committed.
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

Figure 8. Initiating space reclamation on a Symmetrix thin device

As seen in Figure 9, the space reclamation process deallocates 23.7 GB of zeros from the thin pool. After the space reclamation is done, thin device 0208 only consumes 1% of the thin pool instead of the 16% from before the reclamation process.

```
Thin Devices(2):
  {
   ----------------------------------------------------------------------
                        Pool          Pool              Pool
   Sym          Total  Subs       Allocated           Written
   Dev            GBs   (%)       GBs    (%)          GBs   (%)   Status

   0208          50.0    31      25.8     16         25.7    16   Reclaiming
   0209          50.0    31       0.0      0          0.0     0   Bound
          ---------- ----- ---------- ----- ---------- -----
  GBs           100.0    63      25.8     16         25.7    16
  }

Thin Devices(2):
  {
   ----------------------------------------------------------------------
                        Pool          Pool              Pool
   Sym          Total  Subs       Allocated           Written
   Dev            GBs   (%)       GBs    (%)          GBs   (%)   Status

   0208          50.0    31       2.2      1          2.0     1   Bound
   0209          50.0    31       0.0      0          0.0     0   Bound
          ---------- ----- ---------- ----- ---------- -----
  GBs           100.0    63       2.2      1          2.0     1
  }
```

Figure 9. Space reclamation on a Symmetrix thin device

Host allocation considerations

Host file system and volume management routines behave in subtly different ways. These behaviors are usually invisible to end users. The introduction of thin devices can put these activities in a new perspective. For example, a UNIX file system might show as empty when a "`df -k`" command is executed, but many blocks may have been written to the thin device in support of file system metadata like inodes and others. From a Virtual Provisioning point of view this thin device could be using a substantial number of thin extents even though the space is showing as available to the operating system and logical volume manager. While this kind of preformatting activity diminishes the value of Virtual Provisioning in one area, it does not negate the other values that Virtual Provisioning brings.

In addition, in VMware vSphere environments, the guest operating system, file systems, and applications running in virtual machines each can have a different kind of behavior when laid down on a virtually provisioned device. An educated choice as to file system and logical volume manager options in the guest operating system can yield the maximum value from the Virtual Provisioning infrastructure.

Considerations for VMware vSphere on thin devices

There are certain considerations that VMware administrators must be aware of when deploying a VMware vSphere environment using Virtual Provisioning. How the various VMware vSphere features work in this kind of configuration depends on the feature, its usage, and the guest operating system running in the VMware vSphere environment. This section describes how VMware vSphere features interact with virtually provisioned devices.

Device visibility and access

Bound thin devices[2] appear like any other SCSI attached device to the VMware vSphere. A thin device can be used to create a VMware file system, or assigned exclusively to a virtual machine as a raw disk mapping (RDM).

One thing to remember is that a thin device when presented as RDM (either physical or virtual compatibility) to a virtual machine, the VMware kernel does not play a direct role in the I/O to the thin device generated by the guest operating system running in the virtual machine. In this configuration, the considerations for using thin devices are no different from the ones for physical servers running the same operating system and applications.

VMware file system on thin devices

VMware file system creation and formatting

The VMware file system (VMFS) has some interesting and valuable behavioral characteristics when viewed in a Virtual Provisioning landscape. First, a minimal number of thin extents are allocated from the thin pool when a VMware file system is created on virtually provisioned devices. The amount of storage required to store the file system metadata is a function of the size of the thin device.

The metadata for the VMware file system (CH_VP_VMax_DS) on the thin device `vmhba1:C0:T1:L5`, shown in Figure 10, consumes 561 MB. This is shown in Figure 11. Figure 12 shows the tracks allocated by the Enginuity operating system in response to the write activity generated by the formatting of the VMware file system. It can be seen from this figure that only 214 MB were actually allocated from the thin

[2] Thin devices not bound to a thin pool are presented as not-ready to the host. However, the SCSI INQ command to the device succeeds. The VMware ESX Server kernel, therefore, cannot determine the correct status of the unbound device. If the VMware ESX Server kernel performs a read on unbound thin devices zeros would be returned back. Similarly, any attempt to create a datastore on an unbound thin device will fail. Therefore, EMC strongly dissuades customers from presenting unbound thin devices to VMware ESX Servers.

pool. Therefore, the VMware file system does not write all of its metadata to disks on creation. The VMware file system formats and uses the reserved area for metadata as requirements arise.

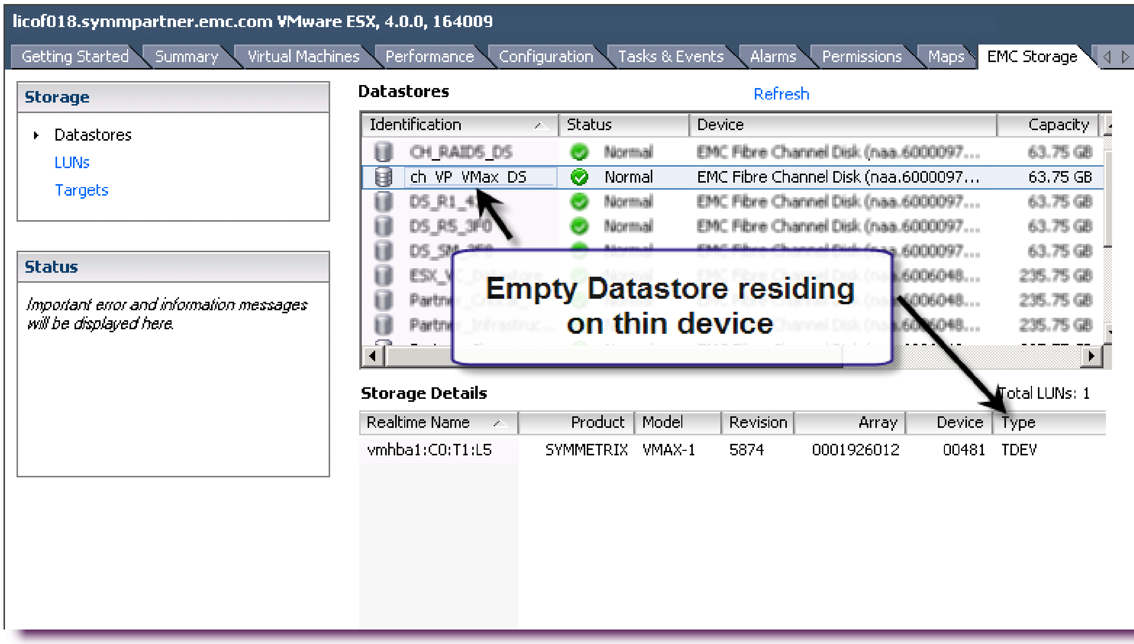


Figure 10. A VMFS datastore on a thin device viewed in Virtual Storage Integrator 3.0 for vSphere Client

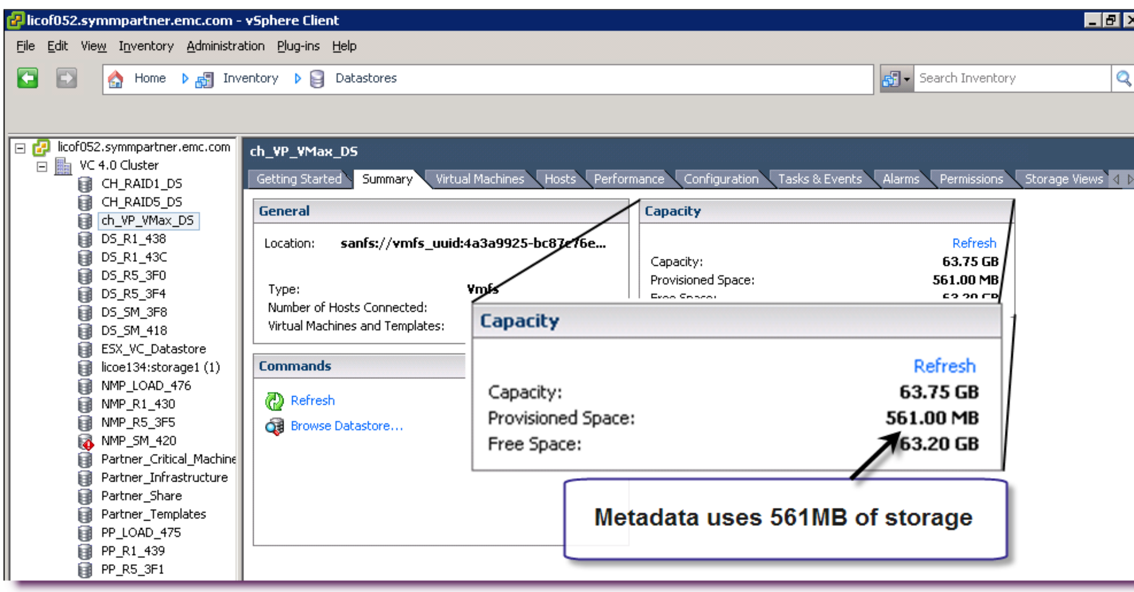


Figure 11. Metadata area reserved on the VMware file system

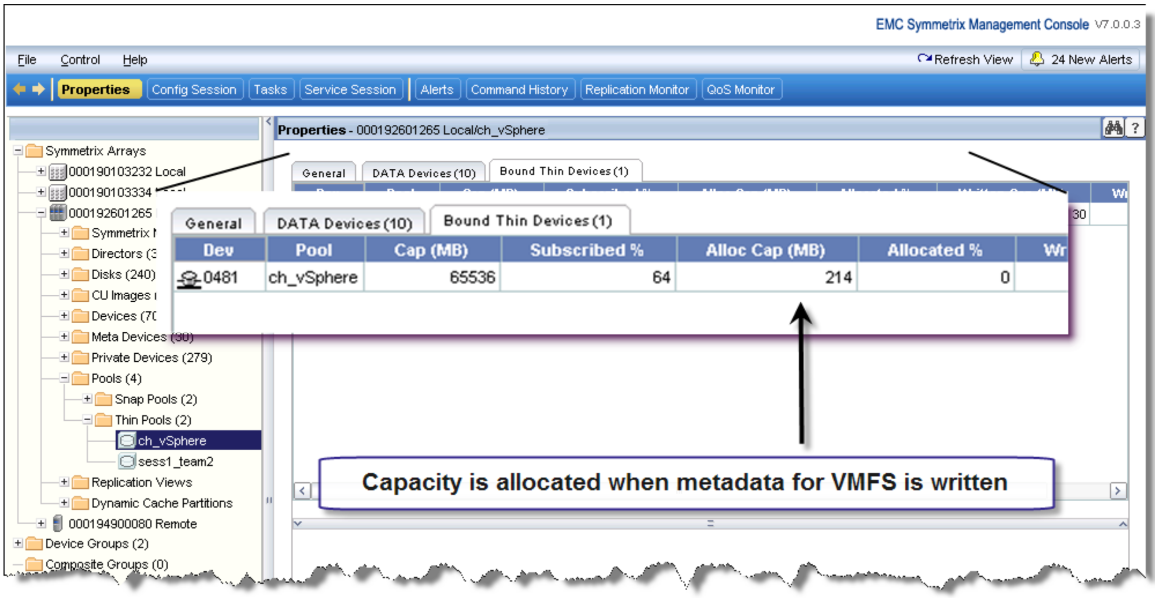


Figure 12. Allocation from a thin pool in response to VMware file system format activity

VMware file system block allocation and reuse algorithms

The current implementation of Virtual Provisioning does not have a mechanism to reclaim storage from the thin pool when an object is deleted by the application using the thin device[3]. If the file system created on a thin device does not reuse a previously allocated block, new tracks are continuously allocated from the thin pool in response to new write activity. This behavior reduces the benefit of using virtually provisioned devices.

Figure 13 shows the results from a series of tests conducted to determine the behavior of the VMware file system. The size of the thin device used in the test was 539.30 GB in size. The thin pool to which the thin device was bound had a total enabled capacity of 134.80 GB. The graph shows the percentage utilization of the VMware file system and the thin pool providing the storage for the thin device used in the experiment. The graph also shows the normalized utilization percentage of the VMware file system. This value, obtained by scaling the VMware file system utilization percentage by the ratio of thin device to thin pool size, enables effective evaluation of the VMware file system behavior.

It can be seen from the figure that the VMware file system utilization drops when the 30 GB virtual disk is deleted in file operation step No. 4. As expected, the percentage utilization of the thin pool does not change. However, from that point on, the VMware file system reuses the freed blocks as new files are created and deleted on the file system. The VMware file system, thus, exhibits a behavior that is beneficial when used with virtually provisioned devices.

[3] With the introduction of zero page reclamation in Solutions Enabler 7.1 for 5874 Enginuity, it is possible to reclaim this space if the previously used extents are zeroed out by the host operating system and space reclamation is then performed.

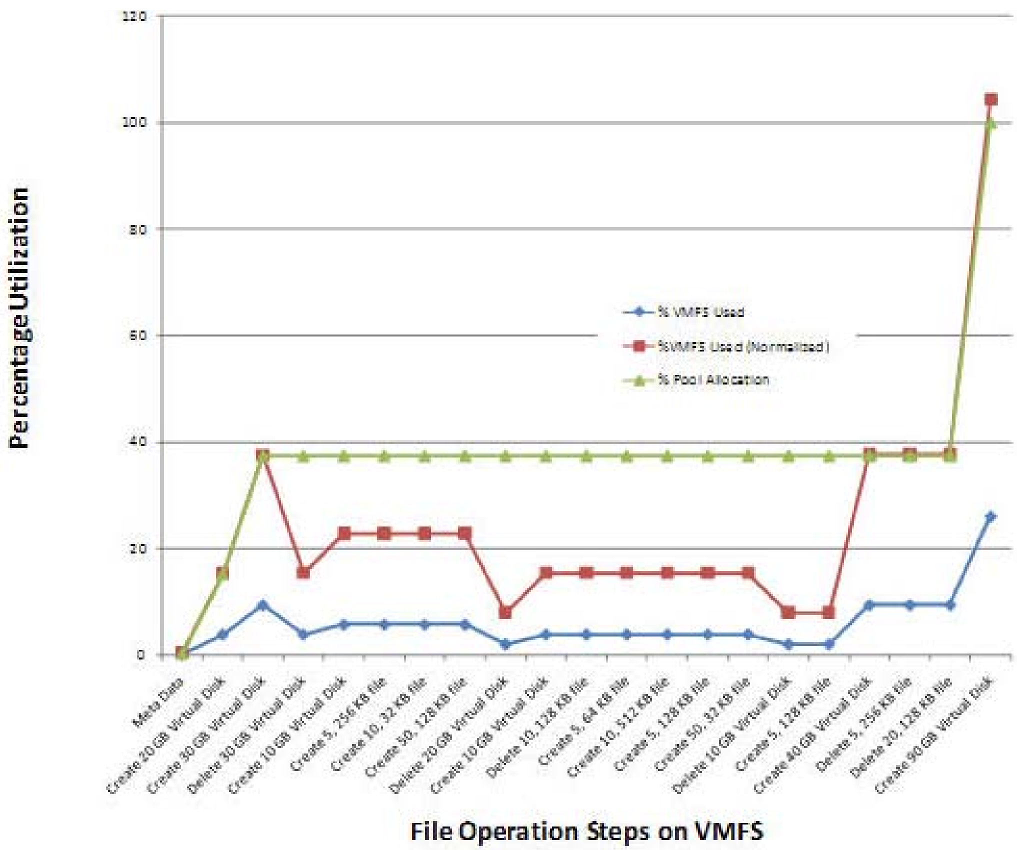


Figure 13. VMware file system allocation mechanisms

Impact of guest operating system activities on thin pool utilization

As discussed in the previous section, a small number of thin extents are allocated in response to the creation of a new virtual machine. However, the thin pool utilization rapidly grows as the user performs various activities in the virtual machine. For example, the installation of an operating system in the virtual machine triggers write I/Os to previously uninitialized blocks of data. These I/Os in turn result in allocation of new thin extents in the thin pool associated with the thin device.

The amount of storage used by the virtual machine depends on the behavior of the operating system, the logical volume manager, the file systems, and applications running inside the virtual machine. Poor allocation and reuse of blocks freed from deleted files can quickly result in thin pool allocation that is equal to the size of the virtual disks presented to the virtual machines. Nevertheless, the thin pool allocation in support of a virtual machine can never exceed the size of the virtual disks. Therefore, users should consider the guest operating system and application behavior when configuring virtual disks for new virtual machines.

A number of additional documents listed in the "References" section discuss the behavior of various operating systems, file systems, and applications in a Virtual Provisioning environment. Since the behavior does not change inside a virtual machine, readers should consult these documents to understand the anticipated behavior of the virtual machines in a VMware vSphere environment.

Virtual disk allocation mechanisms

VMware vSphere 4 offers multiple ways of formatting virtual disks and for the first time, has integrated these options into vCenter itself. For new virtual machine creation, only the formats eagerzeroedthick,

zeroedthick, and thin can be used. In order to create clones of a VMDK file with non-default allocation mechanisms (in other words, mechanisms other than thin, zeroedthick, or eagerzeroedthick) the CLI utility, `vmkfstools` must be used.

All of the allocation mechanisms that the VMware kernel provides for creating virtual disks are listed in Table 3.

Table 3. Allocation policies when creating new virtual disks on a VMware datastore

Allocation mechanism (Virtual Disk format)	**VMware kernel behavior**
zeroedthick	All space is allocated at creation but is not initialized with zeros. However, the allocated space is wiped clean of any previous contents of the physical media. This is the default policy when creating new virtual disks.
eagerzeroedthick	This allocation mechanism allocates all of the space and initializes all of the blocks with zeros. This allocation mechanism performs a write to every block of the virtual disk, and hence results in equivalent storage use in the thin pool.
thin	This allocation mechanism does not reserve any space on the VMware file system on creation of the virtual disk. The space is allocated and zeroed on demand.
rdm	The virtual disk created in this mechanism is a mapping file that contains the pointers to the blocks of SCSI disk it is mapping. However, the SCSI INQ information of the physical media is virtualized. This format is commonly known as the “Virtual compatibility mode of raw disk mapping”.
rdmp	This format is similar to the rdm format discussed above. In this format, the SCSI INQ information of the physical media is not virtualized. This format is commonly known as the “Pass-through raw disk mapping”.
raw	This mechanism can be used to address all SCSI devices supported by the kernel except for SCSI disks.
2gbsparse	The virtual disk created using this format is broken into multiple sparsely allocated extents (if needed), with each extent no more than 2 GB in size. However, a sparse disk cannot be powered on, on an ESX/ESXi host, unless you first re-import the disk in a compatible format, such as zeroedthick, eagerzeroedthick, or thin, with `vmkfstools`.
monosparse	A monolithic sparse disk. A disk in this format can be used with other VMware products. This format is only applicable as a destination format in a clone operation, and not usable for disk creation.
monoflat	A monolithic flat disk. A disk in this format can be used with other VMware products. This format is only applicable as a destination format in a clone operation, and not usable for disk creation.

“Zeroedthick” allocation format

When creating, cloning, or converting virtual disks, the default option is “thick.” The “thick” selection, as it is referred to within the vSphere Client, is actually the “zeroedthick” format. This is the recommended disk format for use with Symmetrix Virtual Provisioning. In this allocation scheme, the storage required for the virtual disks is reserved in the datastore but the VMware kernel does not initialize all the blocks. The blocks are initialized by the guest operating system as write activities to previously uninitialized blocks are performed. The VMware file system will return zeros to the guest operating system if it attempts to read blocks of data that it has not previously written to. This is true even in cases where information from previous allocation is available—the VMware file system will not present stale data to the guest operating system when the virtual disk is created using the “zeroedthick” format. Since the VMFS volume will report the virtual disk as fully allocated, the risk of oversubscribing and running out of space in the thin pool is reduced. This is shown in Figure 14, which displays a single 10 GB virtual disk (with only 1400 MB of actual data on it) that uses the “zeroedthick” allocation method. The datastore browser reports the virtual disk as consuming the full 10 GB on the volume.

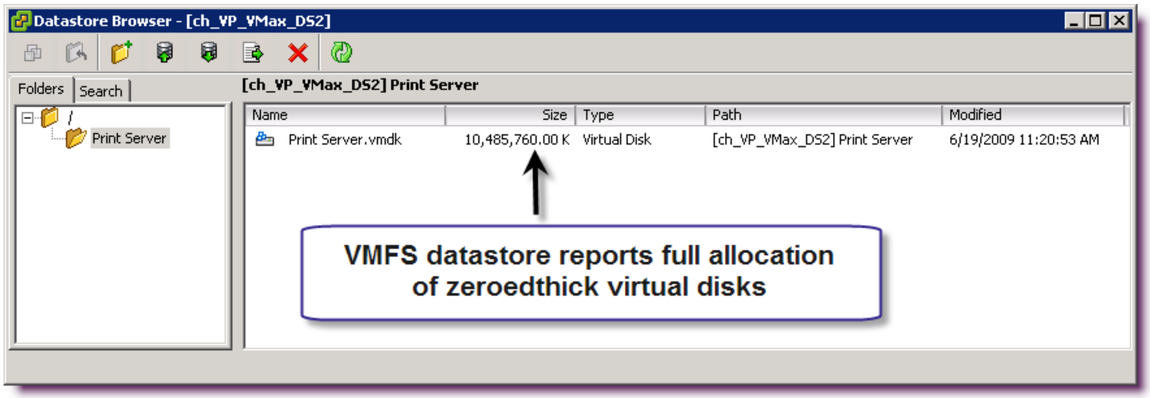


Figure 14. Zeroedthick virtual disk allocation size as seen in a VMFS datastore browser

However, since the VMware kernel does not actually initialize unused blocks, the full 10 GB is not consumed on the thin device. This virtual disk resides on thin device 0482 and, as seen in Figure 15, only consumes 1600 MB of space on it (1400 MB for the data and 200 for the VMFS metadata).

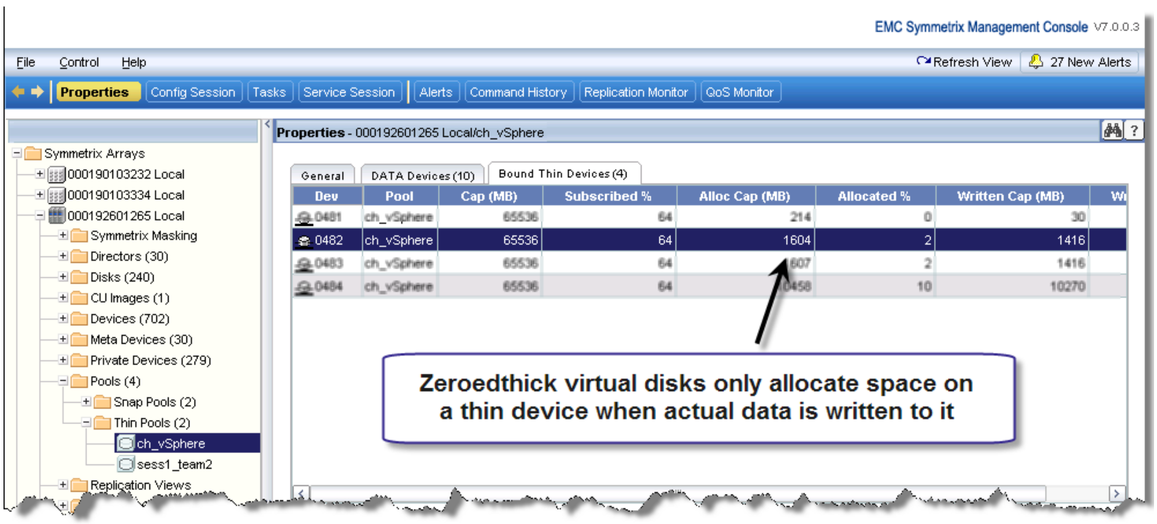


Figure 15. Zeroedthick virtual disk allocation on Symmetrix thin devices

"Thin" allocation format

Like zeroedthick, the "thin" allocation mechanism is also Virtual Provisioning friendly, but, as will be explained in this section, should only be used with caution in conjunction with Symmetrix Virtual Provisioning. "Thin" virtual disks increase the efficiency of storage utilization for virtualization environments by using only the amount of underlying storage resources needed for that virtual disk exactly like zeroedthick. But, unlike zeroedthick, thin devices do not reserve space on the VMFS volume. Upon the initial allocation of the virtual disk, the VMFS storage requirements will be given 1 MB[4] of space in the datastore. As that space is filled, additional 1 MB chunks of storage will be allocated for the virtual disk so that the underlying storage demand will grow as its size increases.

[4] It is important to note that this allocation size depends on the block size of the target VMware file system. The default block size is 1 MB but this can be altered to 2, 4, or 8 MB. Therefore, if the block size is 8 MB, then the initial allocation of a new "thin" virtual disk will be 8 MB and will be subsequently increased in size by 8 MB chunks.

A single 10 GB virtual disk (with only 1.4 GB of actual data on it) that uses the thin allocation method is shown in Figure 16. The datastore browser reports the virtual disk as consuming only 1.4 GB on the volume.

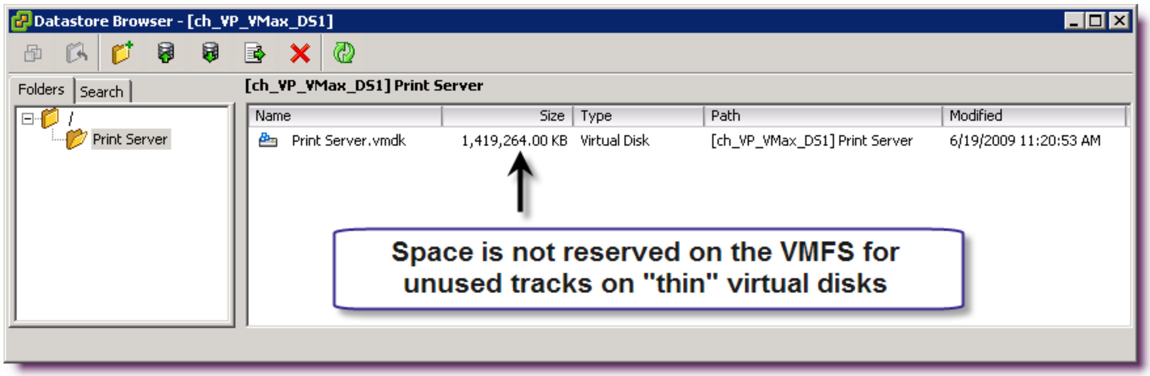


Figure 16. Thin virtual disk allocation size as seen in VMFS datastore browser

As is the case with zeroedthick allocation format, the VMware kernel does not actually initialize unused blocks, so the full 10 GB is neither reserved on the VMFS nor consumed on the thin device. This virtual disk resides on thin device 0483 and, as seen in Figure 17, only consumes 1600 MB of space on it (1400 MB for the data and 200 MB for the VMFS metadata).

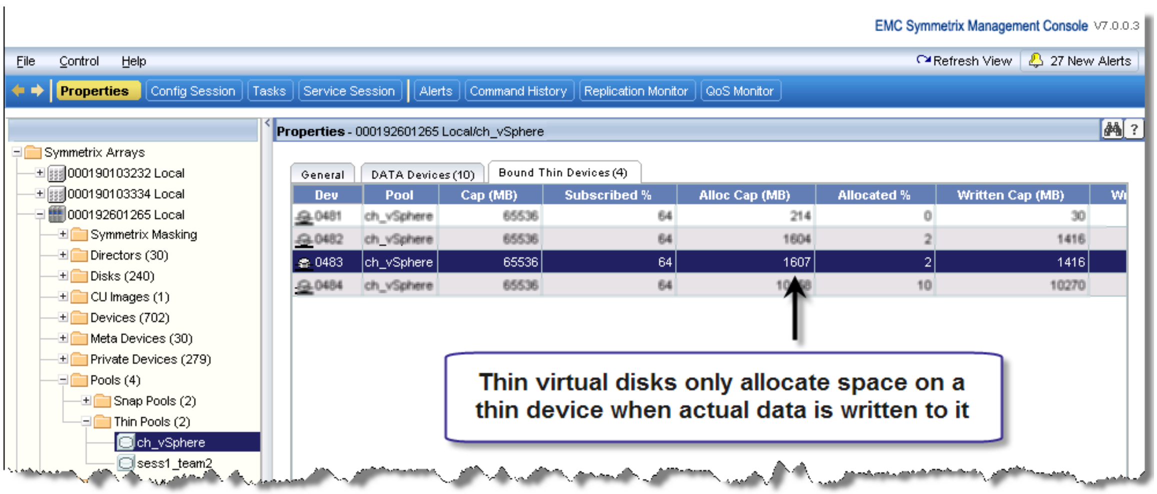


Figure 17. Thin virtual disk allocation on Symmetrix thin devices

The risk of exceeding the Symmetrix Virtual Provisioning thin pool capacity is much higher when virtual disks are allocated using this policy since the oversubscription to physical storage occurs at two independent layers that currently do not communicate with each other. Therefore, EMC does not recommend the use of "thin" virtual disks in conjunction with Symmetrix Virtual Provisioning without very careful monitoring on the VMware layer as well as the Symmetrix thin pool. The use of "thin" virtual disks is acceptable with regular Symmetrix devices as long as capacity levels are diligently monitored. To help with this, vCenter has been updated with new management screens and capabilities such as raising alerts, alarms, and improved datastore utilization reports to enable management of overprovisioned datastores.

"Eagerzeroedthick" allocation format

With the "eagerzeroedthick" allocation mechanism, space required for the virtual disk is completely allocated and written to at creation time. This leads to a full reservation of space on the VMFS datastore and on the underlying Symmetrix device. Accordingly, it takes longer to create disks in this format than to create other types of disks. Because of this behavior, “eagerzeroedthick” format is not ideal for use with virtually provisioned devices. It should be noted that any virtual machine using vSphere’s Fault Tolerance feature uses this allocation format for all of its virtual disks. As soon as Fault Tolerance is enabled on a virtual machine, all associated virtual disks will be converted to an “eagerzeroedthick” format (the virtual machine must be powered down first).

A single 10 GB virtual disk (with only 1.4 GB of actual data on it) that uses the “eagerzeroedthick” allocation method is shown in Figure 18. The datastore browser reports the virtual disk as consuming the entire 10 GB on the volume.

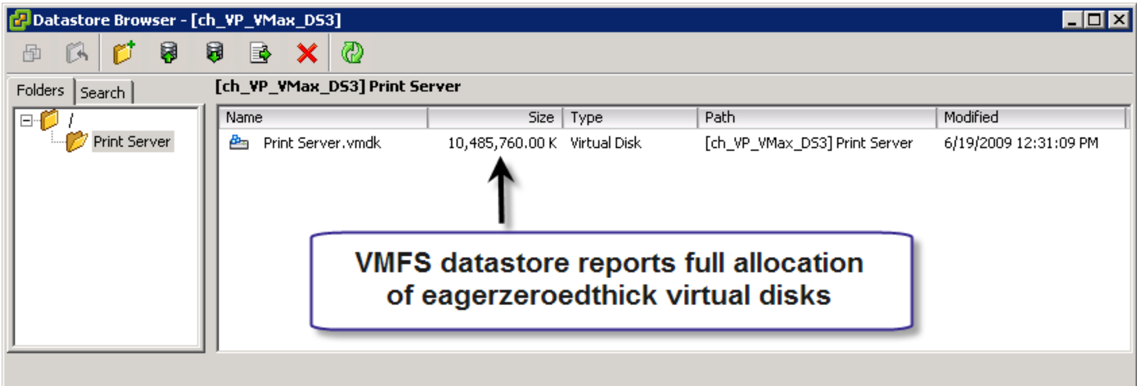


Figure 18. Eagerzeroedthick virtual disk allocation size as seen in a VMFS datastore browser

Unlike with “zeroedthick” and “thin”, the VMware kernel does initialize all the unused blocks, so the full 10 GB is reserved on the VMFS and consumed on the thin device. The “eagerzeroedthick” virtual disk resides on thin device 0484 and, as seen in Figure 19, consumes 10.2 GB of space on it (10 GB for the data and 200 MB for the VMFS metadata).

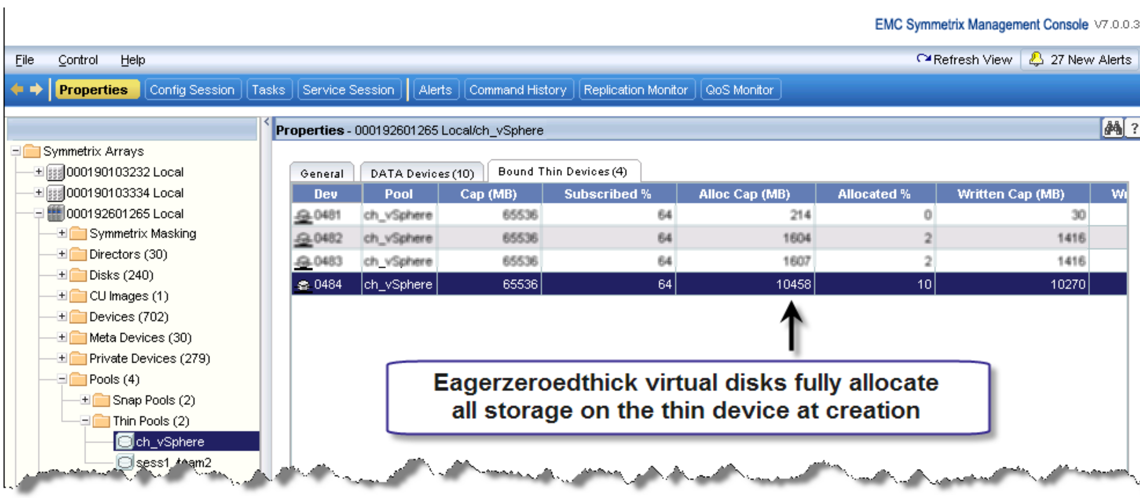


Figure 19. Eagerzeroedthick virtual disk allocation on Symmetrix thin devices

Migrating virtual machines

VMware Storage VMotion is a component of VMware vSphere that provides a graphical interface for live migration of virtual machine disk files within or across storage arrays with no downtime or disruption in service. Storage VMotion relocates virtual machine disk files from one shared storage location to another shared storage location with continuous availability. Storage VMotion enables organizations to perform proactive storage migrations, simplify array migrations, improve virtual machine storage performance, and free up valuable storage capacity. Storage VMotion is fully integrated with VMware vCenter Server to provide easy migration and monitoring.

VMware Storage VMotion works by first moving the home directory of the virtual machine to the new specified location. This directory contains the metadata about the virtual machine such as the configuration .vmx file, the swap, and any log files. After transferring the home directory, Storage VMotion copies the data of the virtual machine VMDKs to the destination storage host, using what VMware calls “changed block tracking” to maintain the data’s integrity during the Storage VMotion process. Subsequently, the block tracking module is read to determine what portions of the disk were written to during the first copy iteration. After the first iteration is complete, a second iteration is executed, where any portions that were changed since the start of the first iteration are copied over. This process may continue multiple times until completed. After the copy has finished, the virtual machine is rapidly suspended and resumed so that it can begin using the new home directory and disk file on the newly specified VMFS datastore. Lastly, VMware ESX does not allow the virtual machine to start running again until the final changed tracks of the source VMDK are copied over to the destination and the original files are deleted. This approach ensures total data reliability and is swift enough to be imperceptible to any end users.

Storage VMotion can be used to perform many different functions, including:

- Migrating a VM between datastores on the same array
- Migrating a VM between datastores on different arrays
- Changing the virtual disk format

For the purposes of explanation, the following section will provide a typical example to extol the uses of Storage VMotion. Consider a virtual machine that has a new requirement to reside on the VMFS datastore residing on a RAID 5 Symmetrix device instead of its current VMFS datastore that resides on a RAID 1 device. Using Storage VMotion, the virtual machine will be migrated, while powered on, between the two datastores.

In Figure 20, the storage configuration of the source RAID 1 datastore, CH_RAID1_DS, and the target datastore, CH_RAID5_DS, are shown within the vSphere Client.

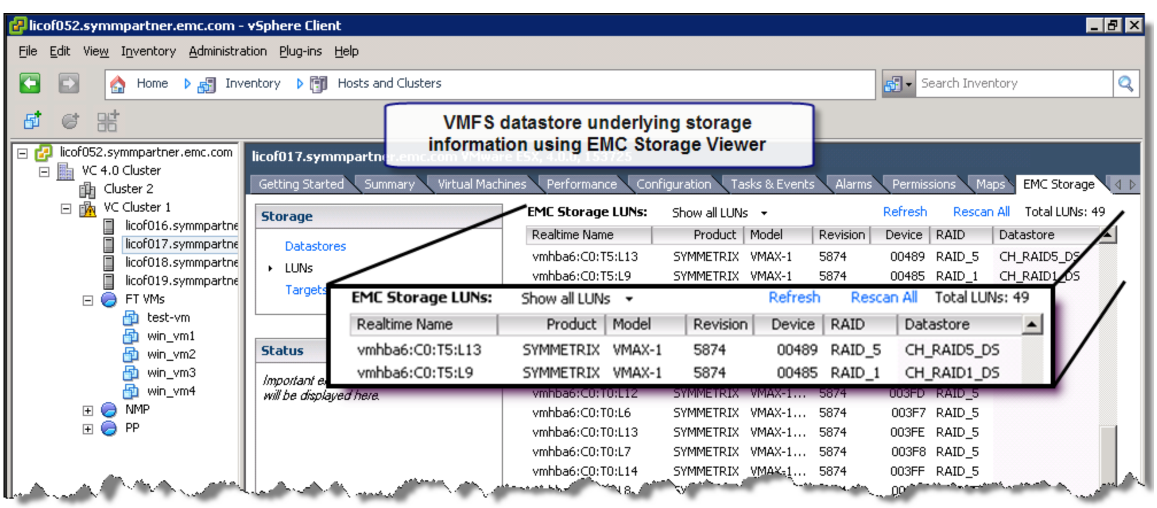


Figure 20. Viewing Symmetrix device configuration using Virtual Storage Integrator 3.0 for vSphere Client

Virtual Machine test-vm currently resides on a RAID 1 datastore and will be moved to a RAID 5 datastore without interrupting the virtual machine’s applications. The Migrate Virtual Machine wizard offers three choices: **Change host**, **Change datastore**, and **Change both host and datastore**. The first option initiates

VMotion, and the second option initiates Storage VMotion (Figure 21). The third option (changing both the host and datastore at once) is only available when the virtual machine is powered off.

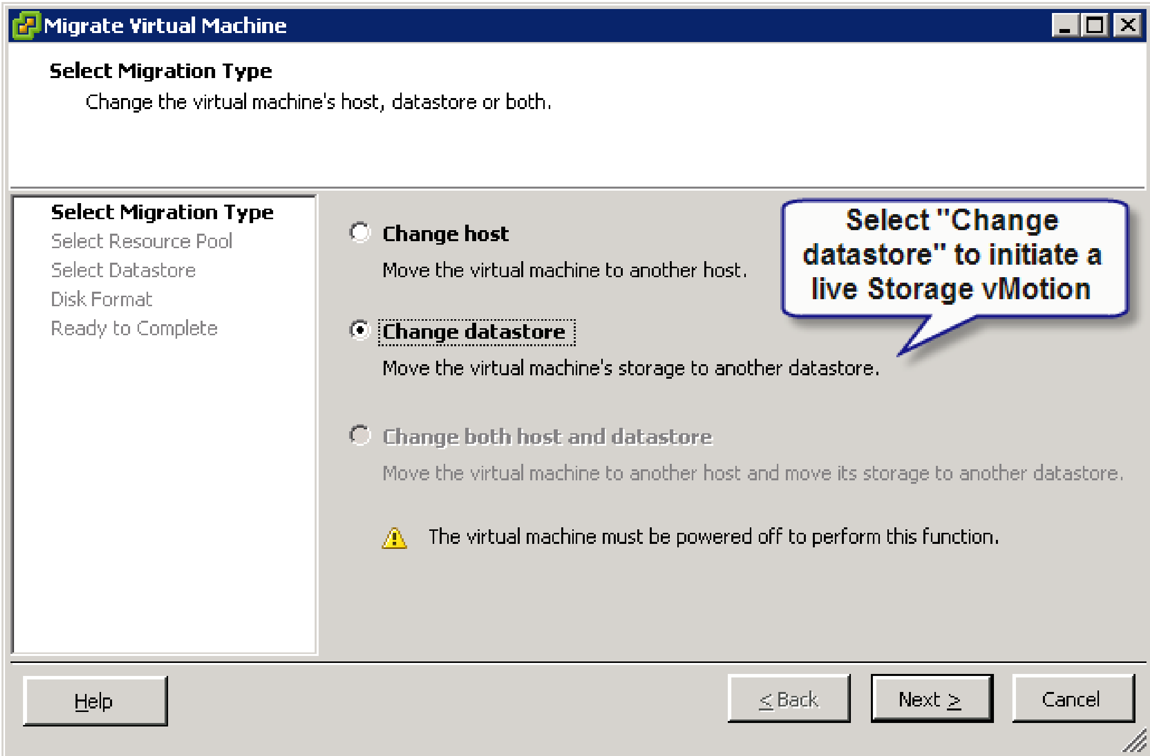


Figure 21. Migrate Virtual Machine wizard in vSphere Client

After the **Change datastore** radio button is selected, the wizard displays the available datastores to the virtual machine's current ESX server. In this case CH_RAID5_DS will be chosen to move the virtual machine onto a RAID 5 device (Figure 22).

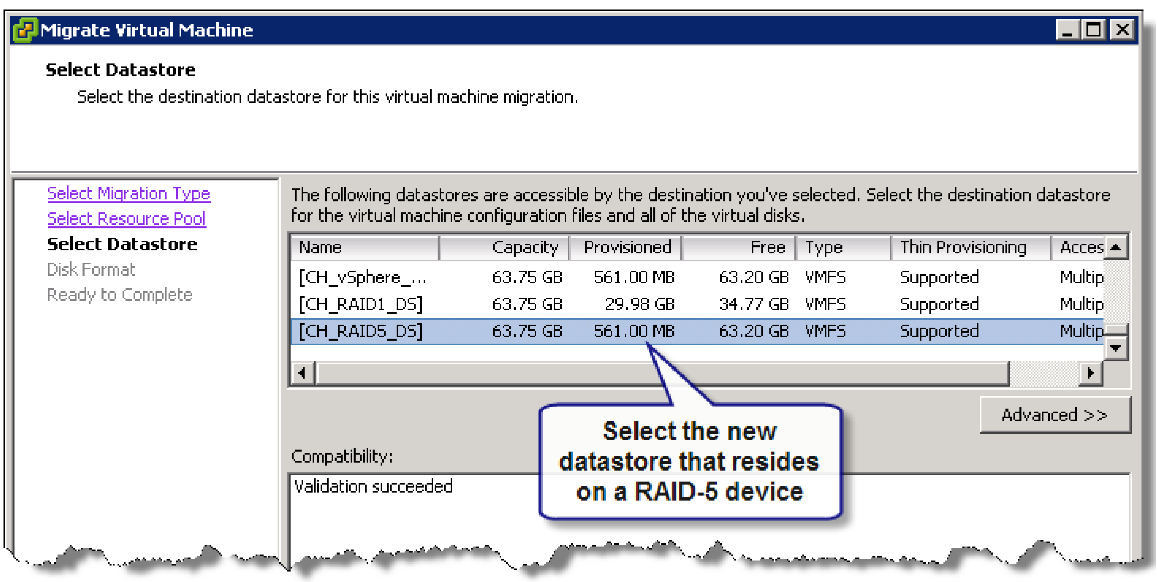


Figure 22. Migrating a VM with Storage VMotion to a new datastore

The Migrate Virtual Machine wizard also gives the option to change the format of the virtual disk during migration (Figure 23). As previously discussed, EMC strongly recommends the use of the “thick” format when using Symmetrix Virtual Provisioning due to the dangers of surpassing subscription limits when using the “thin” format on a virtually provisioned Symmetrix device. It is important to note that the “thick’ option is the zeroedthick format, not eagerzeroedthick, which will therefore remain Symmetrix Virtual Provisioning friendly. The “thin” format may be used on regular Symmetrix devices, but care needs to be taken to monitor free space as the “thin” disks grow.

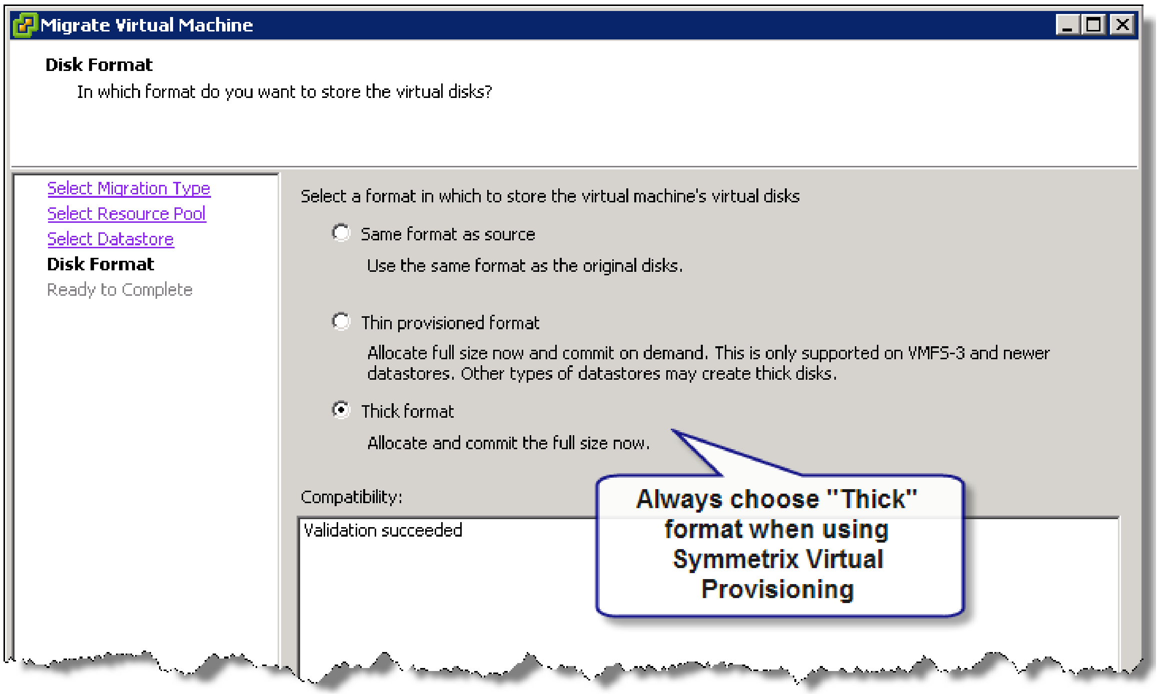


Figure 23. Selecting virtual disk format in Migrate Virtual Machine wizard

Once the wizard is finished, the Storage VMotion session begins and will move the virtual machine’s files to the new location without disrupting the virtual machines’ availability (Figure 24).

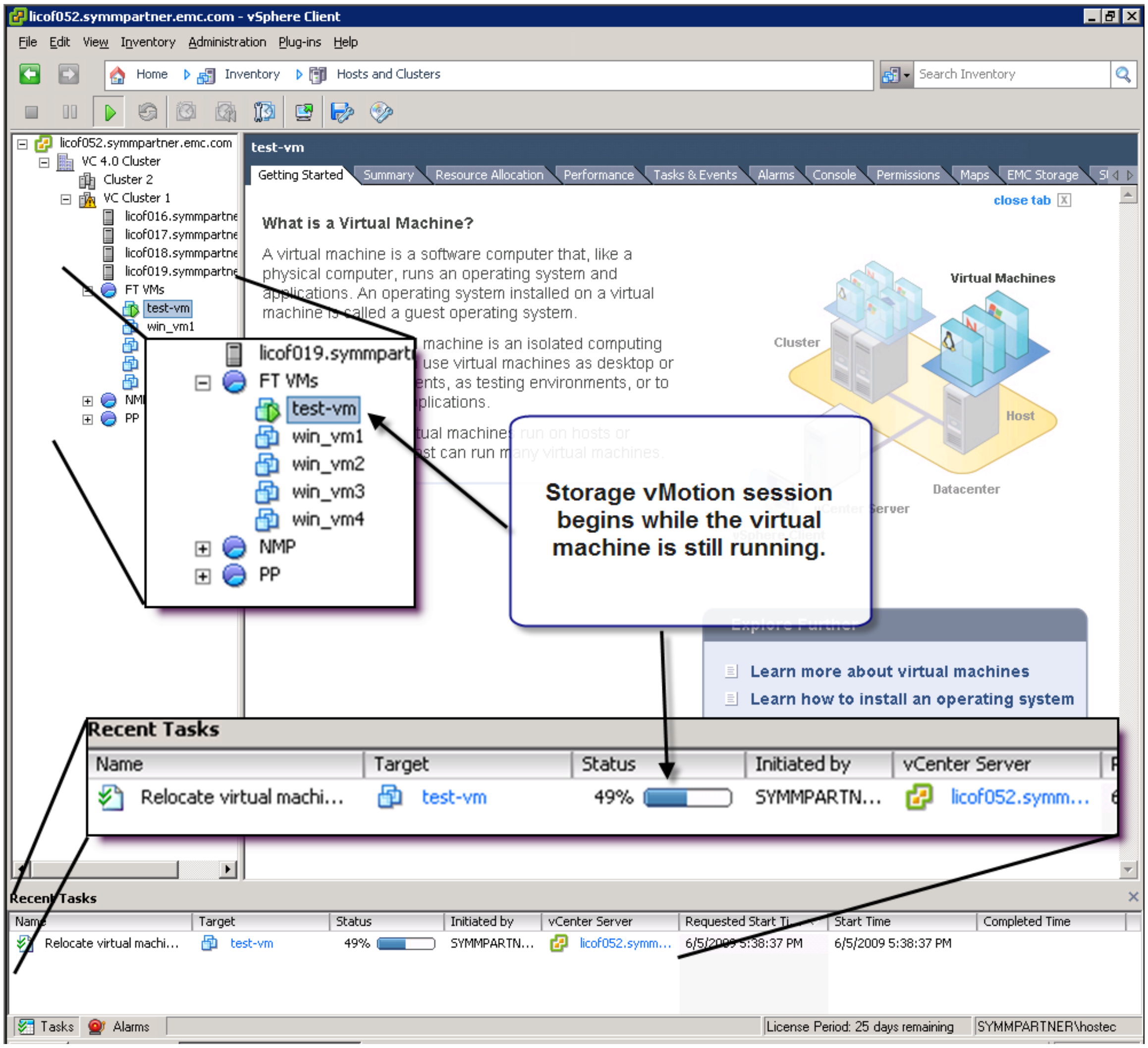


Figure 24. Storage VMotion session in progress

Changing virtual disk formats using Storage VMotion

Many times it is necessary to modify the allocation format of existing virtual disks to satisfy new storage requirements. In the case of migrating from between thin and thick storage environments, the ability to change disks formats is especially important. The built-in mechanism to nondisruptively convert virtual disks formats is Storage VMotion using the Migrate Virtual Machine wizard. As discussed in the previous section, Storage VMotion is typically used to move virtual machines from one datastore to another and, if desired, allows for the virtual disks to be re-created with a different allocation mechanism. In the current release of vSphere 4.0, Storage VMotion does not allow a virtual machine to have the same target VMware file system as the source. Consequently, in order for the virtual disk format to be changed, the location must be changed as well. This requires an administrator to Storage VMotion the virtual machine from the source volume to a different target volume and then Storage VMotion it back to the original datastore using the desired virtual disk allocation format.

Inflating virtual disks

In certain cases, a change of the allocation mechanism from “thin” to “eagerzeroedthick” is necessitated for various virtual machines due to shifting requirements. The typical way to alter the allocation format of virtual disks is through the use of the Storage VMotion wizard, but it does not offer the option to convert disks to “eagerzeroedthick”. Instead, vSphere offers the ability to convert virtual disks from the “thin” allocation mechanism to “eagerzeroedthick” using a process referred to as “inflating” by VMware.

Virtual machines that are candidates to be protected by VMware Fault Tolerance would be prime examples of a situation that requires this operation, since Fault Tolerant-protected VMs must use “eagerzeroedthick” for all of their virtual disks. Accordingly, during the initialization of Fault Tolerance on a virtual machine, all associated virtual disks are converted automatically if they are not currently “eagerzeroedthick.”

If it is desired to perform this conversion before either activating Fault Tolerance or other reasons entirely, the inflation of virtual disks can be performed through the vSphere Client “Browse Datastore” function (Figure 25). It is important to note that disk inflating is an offline operation. Consequently, when executing this operation on a virtual disk, all associated virtual machines must first be shut down.

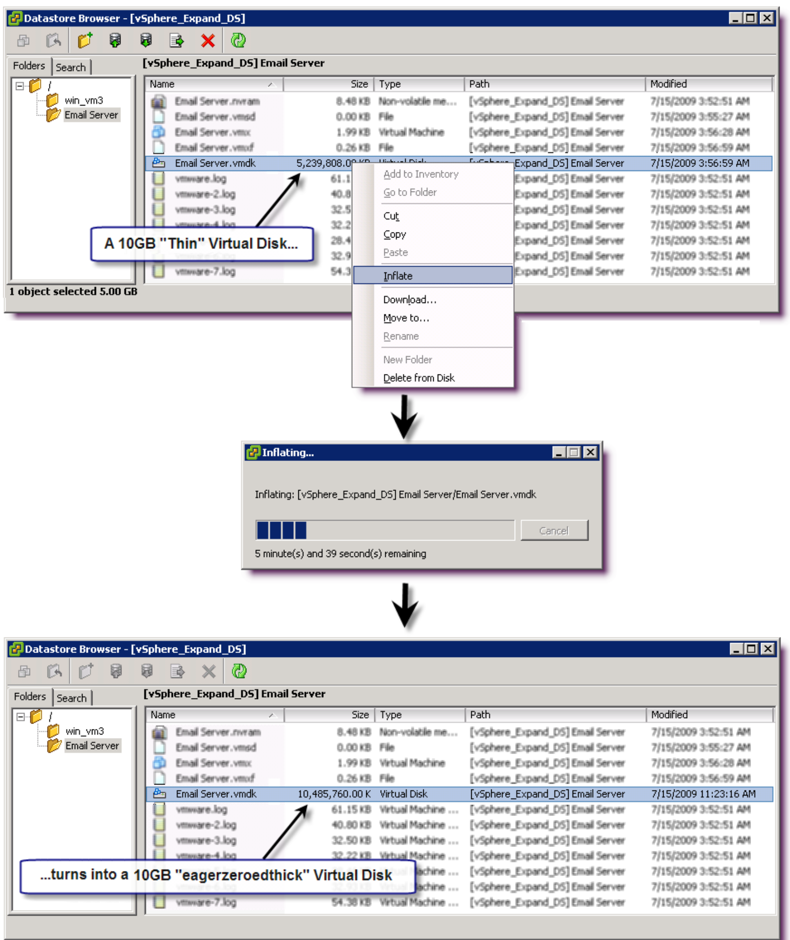


Figure 25. Inflating a thin virtual disk to the eagerzeroed thick format

Creating new virtual disks

The Create a Disk wizard provided by vSphere Client is not impacted by the use of virtually provisioned devices. This is due to the fact that a VMFS datastore on thin devices appear as any other datastore to the wizard. This is true for when a virtual disk is created during the process of making a new virtual machine and for when a new virtual disk is made for an existing one. The Add Hardware – Create a Disk wizard offers three options for choosing the allocation mechanism of a new virtual disk (Figure 26).

The three options are:

Default (no selection). This will create the virtual disk using the “zeroedthick” mechanism and is the only method recommended for use with Symmetrix Virtual Provisioning

Allocate and commit space on demand (Thin Provisioning). This will create the virtual disk using the “thin” mechanism. EMC highly discourages the use of this format for Symmetrix Virtual Provisioning.

Support clustering features such as Fault Tolerance. This will create the virtual disk using the “eagerzeroedthick” mechanism and it is not recommended to be used with Symmetrix Virtual Provisioning as it is not Virtual Provisioning “friendly”.

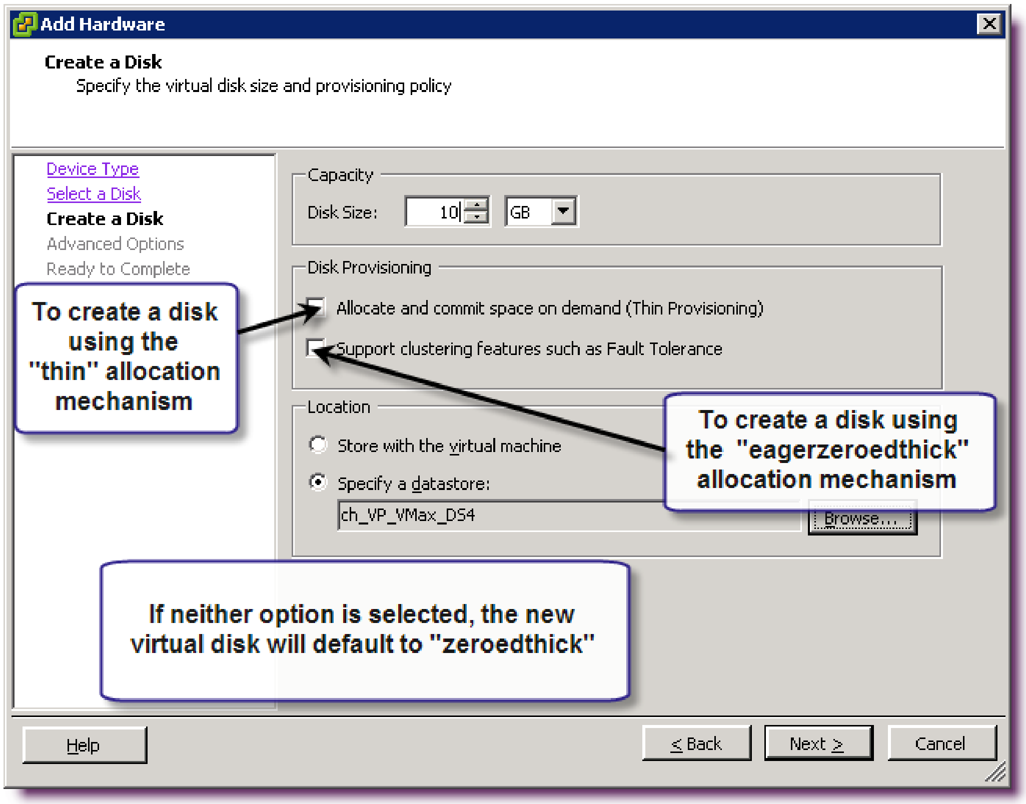


Figure 26. Disk format options when creating a new virtual disk

When the wizard shown in Figure 26 is completed, a 10 GB “zeroedthick” virtual disk is created on the VMFS datastore ch_VP_VMax_DS4 hosted on thin device 02B0. As noted in Figure 27, the new virtual disk only allocates about 4 MB of space on the thin device (the other 134 MB is for VMFS metadata allocated at the creation of the VMFS volume).

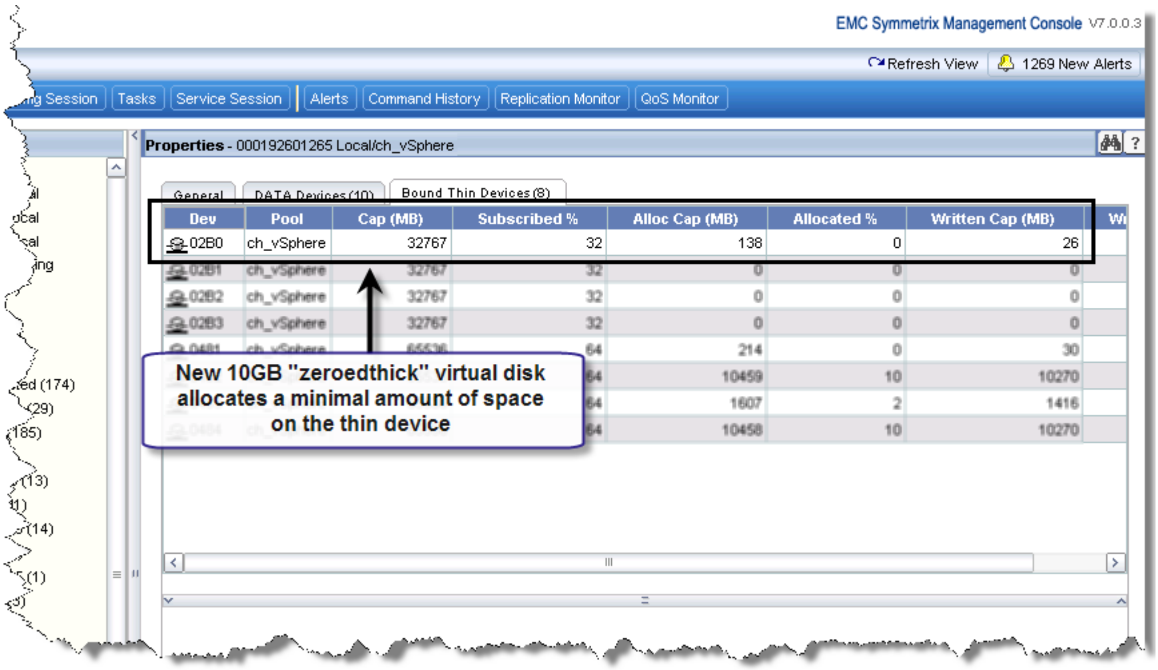


Figure 27. New "zeroedthick" virtual disk thin device allocation in Symmetrix Management Console

Figure 27 shows the storage utilization of the VMware file system and the thin pool supporting the datastore after the New Virtual Machine wizard has completed. The figure clearly shows that a small number of tracks are initialized when a new virtual disk is created. However, as shown in Figure 28, the VMware kernel reserves the storage requirement for the virtual machine on the VMware file system for future use.

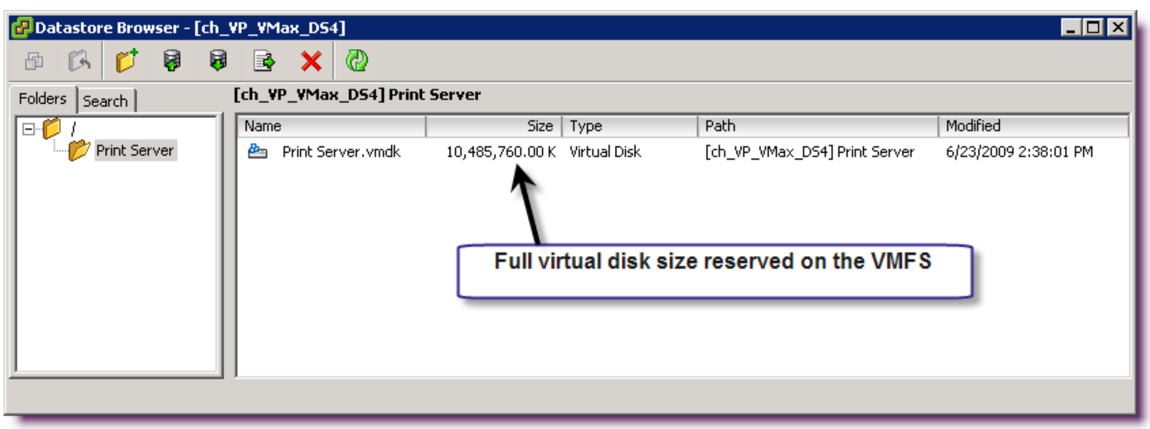


Figure 28. VMware datastore utilization on creation of a new virtual disk

Duplicating virtual machines

In a large VMware vSphere environment it is impractical to manually install and configure the guest operating system on every new virtual machine. To address this, VMware provides multiple mechanisms to simplify the process to create preconfigured new virtual machines:

- Cloning: This wizard-driven method allows users to select a virtual machine and clone it to a new virtual machine. The wizard allows users to change some characteristics of the cloned virtual machine.

- Deploy from Template: The users can use a virtual machine to create a template. Once the template has been created, new virtual machines can be deployed from the template and customized to meet specific requirements. The VMware vSphere Client provides wizards for both activities.

Detailed discussion of these options is beyond the scope of this paper. Readers should consult VMware documentation for details.

Cloning virtual machines

In VMware Virtual Infrastructure 3 and earlier, the cloning process, unfortunately, would convert the virtual disk on the source virtual machines to the "eagerzeroedthick" format on the cloned virtual machine. No other options were allowed in the GUI. Furthermore, since Virtual Infrastructure does not provide a mechanism to change the allocation policy for the virtual disk on the cloned virtual machine, the cloning process was inherently detrimental to the use of virtually provisioned devices. For that reason, EMC did not recommend using the cloning wizard to provision new virtual machines in these environments.

In VMware vSphere 4.0 this issue has been resolved. The Clone Virtual Machine wizard offers users a number of customizations including:

The VMware ESX Server cluster on which the cloned virtual machine will be hosted

The resource pool to associate the virtual machine with

The VMFS datastore on which to deploy the cloned virtual machine

The ability to choose a virtual disk format. This is the only option that impacts Virtually Provisioned Symmetrix devices

The Clone Virtual Machine wizard offers three options for the target disk format— the same format as source, Thin Provisioned format, and Thick format (Figure 29).

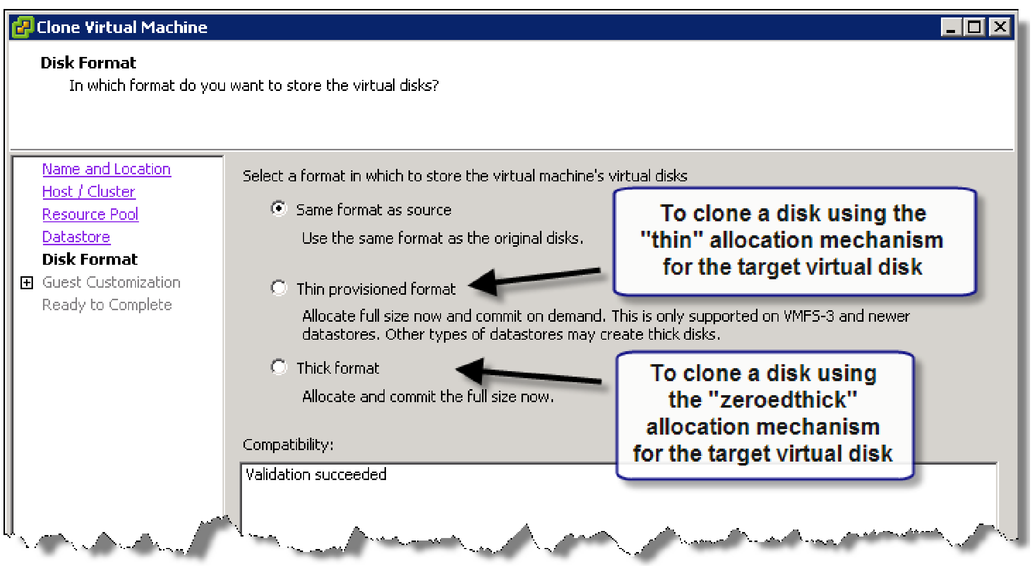


Figure 29. Disk Format options in the Clone Virtual Machine wizard

The "thin" format is only to be used when the underlying datastore resides on a regular (non-thin) Symmetrix device. As previously discussed, a "thin" virtual disk on a datastore located on Symmetrix thin pool greatly increases the chance of exceeding storage limits and for that reason is discouraged. If the datastore is located on a thin pool the "Thick format" option should be used. This option allocates the disk using the zeroedthick method, which is Symmetrix Virtual Provisioning friendly.

It should be noted that in the Clone Virtual Machine wizard, there is no way to clone a virtual disk using the allocation mechanism "eagerzeroedthick" for the target virtual disk unless the source disk is “eagerzeroedthick” in the first place. In order to achieve this, the disk must first be made “thin” and later be inflated.

Virtual machine templates

Invocation of virtual machine templates makes creating and expanding virtual environments simple and efficient, which is highly desirable in large infrastructures. A template is a golden image created from an already configured virtual machine. A template usually includes a specified operating system, a base set of applications, and configuration of virtual hardware. Virtual machines can be turned into or deployed from a template with great ease. Both of these processes have implications when it comes to Symmetrix Virtual Provisioning.

Creating templates from existing virtual machines

As discussed earlier, VMware vSphere provides three mechanisms to simplify the provisioning of new virtual machines. The first option, cloning, was discussed in the previous section. The second alternative, Deploy from Template, requires a predefined template describing the virtual machine configuration. The template can be created by either cloning an existing virtual machine or by converting an existing virtual machine in place. The process to clone an existing virtual machine to a new template involves copying the virtual disks associated with the source virtual machine.

The problem described in the previous section was also an issue during the process to clone a virtual machine to a template in Virtual Infrastructure 3.5 and earlier. This has also been resolved in vSphere 4.0.

In the vSphere 4.0 Client, the wizard to clone a virtual machine to a template offers the user three options on how to format the disk (Figure 30).

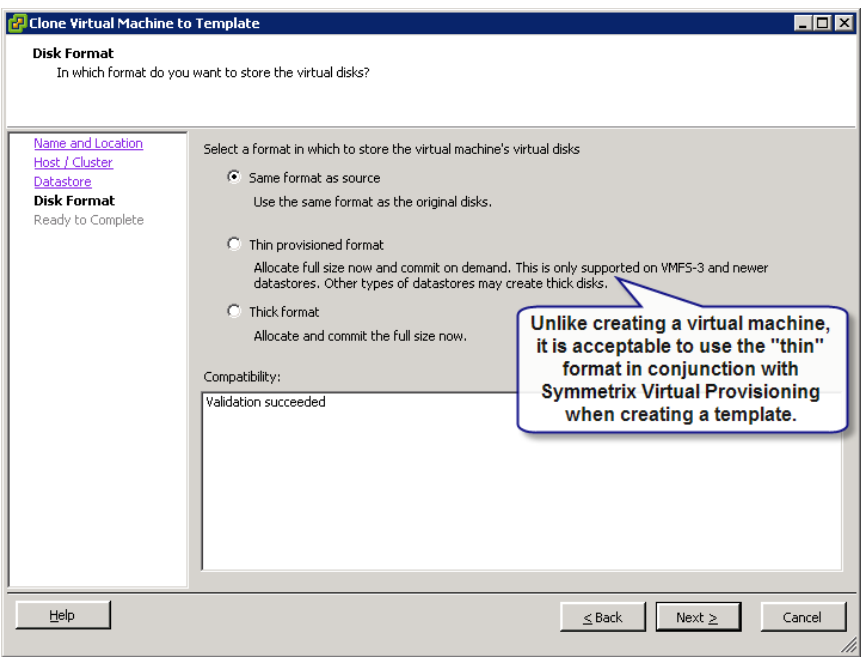


Figure 30. Creating templates from existing virtual machines

As previously discussed, when using Symmetrix Virtual Provisioning the use of the “thin” format is discouraged when creating a new virtual machine. However, when creating/cloning to a template, the use of the “thin” format is acceptable. This is due to the fact that templates (contrasted with virtual machines) will never grow in size. If this option is selected, it is recommended that a separate datastore is dedicated for templates as the contents of the datastore will tend to be relatively static. Consequently, in the worst-case scenario, if the capacity of the thin pool supporting the datastore holding the templates is exceeded, the only impact would be to the process of creating new templates. Existing templates can still be used to provision new virtual machines. This behavior, although inconvenient, has a minimal impact on the production environment.

Using the “thick” format (zeroedthick) for templates will just squander space on VMFS volumes with unused, but allocated, tracks. This is especially important when virtual machine templates have large VMDKs.

However, care must be taken if the template is reverted back into a virtual machine. If the template’s virtual disks were thin, they will remain that way. So, if they reside on a Symmetrix thin pool dangerous oversubscription scenarios might ensure. For that reason, if a “thin” template is to be turned into a virtual machine, it is highly recommended to alter the virtual disks allocation mechanism from “thin” to “thick” using Storage VMotion.

An alternative to the Clone to Template process is the in-place conversion of an existing virtual machine into a template. This process does not modify the virtual disks associated with the virtual machine. The only modification that occurs is to the configuration file. Therefore, an in-place conversion to a template does not unnecessarily increase allocations from the thin pool.

Based on these observations, EMC recommends the use of virtually provisioned devices for templates. A dedicated template datastores on a thin device should be used in conjunction with the “thin” option of the Clone to Template wizard. Alternatively, the in-place conversion of virtual machines on thin devices can be used.

Deploying new virtual machines from templates

Prior to vSphere, when a new virtual machine was deployed from a template, the virtual disk on the new virtual machine was always created using the “eagerzeroedthick” format. Therefore, a significant benefit afforded by the virtually provisioned devices on the Symmetrix storage array was lost. Starting with vSphere, the wizard for deploying new virtual machines from a template provides the ability to choose the virtual disk allocation mechanism.

The options for disk allocations are **Same format as source**, **Thin Provisioned format** and **Thick format**. Since it is highly likely that the template is using “thin” virtual disks, do not use the **Same format as source** option if you are deploying the new virtual machine to a VMFS on Symmetrix virtually provisioned device. The **Thin** format is only to be used when the underlying datastore resides on a regular (non-thin) device. If the datastore is located on a thin pool the **Thick format** (zeroedthick) should be used.

It should be noted that in the Deploy Virtual Machine from Template wizard, there is no way to deploy a new virtual machine disk using the allocation mechanism “eagerzeroedthick” for the target virtual disk unless the source disk is “eagerzeroedthick” in the first place. In order to achieve this, the disk must first be made “thin” and later be inflated.

Virtual machines on VMware file systems of different block sizes

During the creation of a new VMware file system volume, VMware offers block sizes of 1, 2, 4, or 8 MB. The larger block sizes change the maximum size of the virtual disks that can be created on those volumes. With vSphere 4.0 update 1 and earlier, moving virtual machines between VMware file systems that have different block sizes will completely zero out the virtual disks of the virtual machine if the “thick/zeroedthick” option is selected.

Important: This behavior is changed in vSphere 4.0 update 2. When the “thick” format is selected in vSphere 4.0 update 2, the wizard will use the Virtual Provisioning-friendly allocation mechanism “zeroedthick” and not “eagerzeroedthick”. EMC recommends upgrading to ESX 4.0 update 2 for this reason.

In essence, this full allocation results in an “eagerzeroedthick” virtual disk, which is not ideal to Symmetrix Virtual Provisioning. This behavior will be observed during all of the following operations:

- Migrating a virtual machine between VMware file systems of differing block sizes using Storage VMotion

Cloning a virtual machine from a VMware file system of different block size than the target VMware file system

Deploying a virtual machine from a template that is stored on a VMware file system with a different block size than that of the target VMware file system.

As shown in Figure 31, a VMware file system contains one virtual machine with a 20 GB virtual disk using the “zeroedthick” allocation mechanism. It reserves all 20 GB on the file system but only consumes 5.5 GB of space on the thin pool as expected.

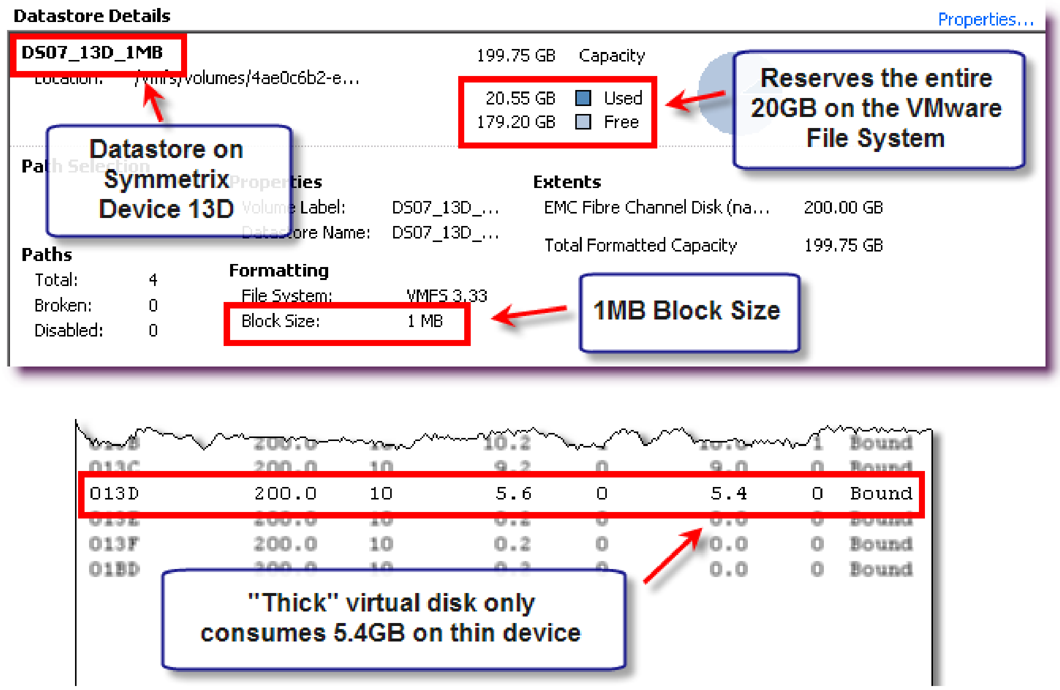


Figure 31. Virtual machine allocations before a migration across different block sizes

In Figure 32, a Storage VMotion procedure is initiated on a virtual machine residing on a VMware file system with a 1 MB block size on a thin device to move it to another thin device with a VMware file system using a block size of 8 MB. The target allocation mechanism is “zeroedthick.” The expected resulting allocation should be a full reservation (20 GB) on the VMware file system and only an allocation of 5.4 GB on the thin device. The unavailability of zero optimization when straddling VMware file systems with different block sizes results in the full allocation of 20 GB on the thin pool. This effect can be seen in Figure 33.

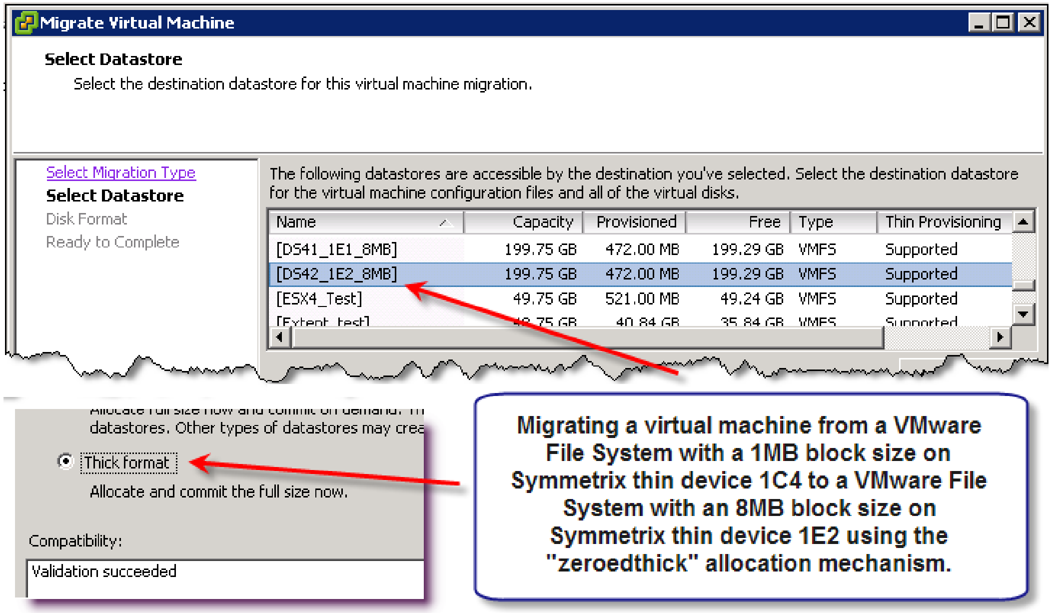


Figure 32. Performing a Storage VMotion between VMware file systems of different block sizes

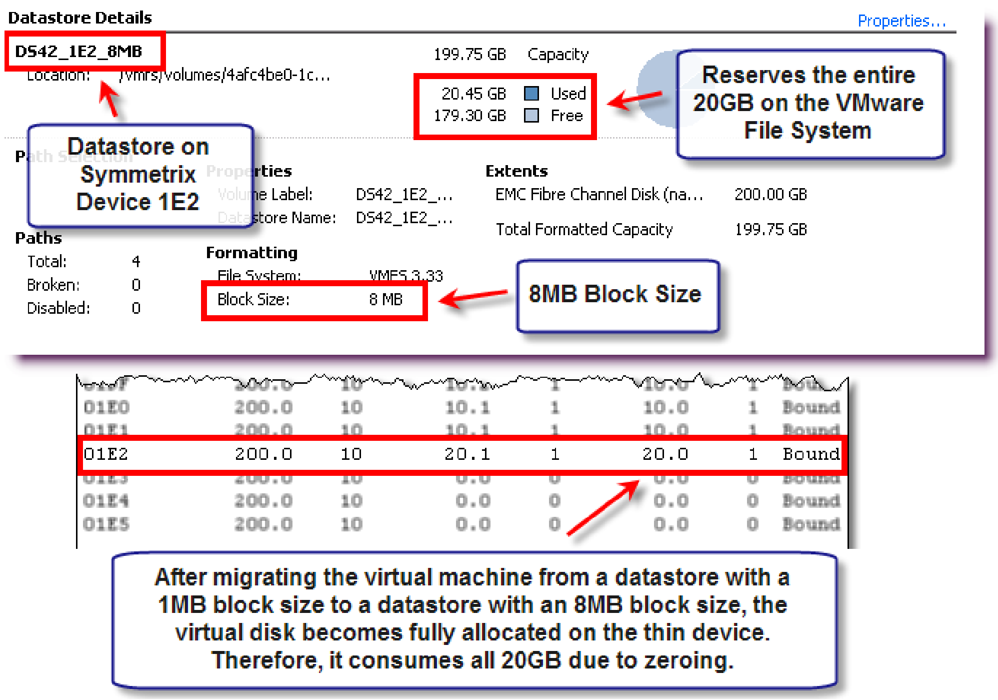


Figure 33. Virtual machine allocations after a migration across different block sizes

Therefore, it is important to use the “thin” allocation mechanism when moving, cloning, or deploying across VMware file systems that have different block sizes. When the “thin” option is selected the target virtual disks will not be zeroed out on the target thin pool.

After the migration/cloning/deployment procedure is complete, the virtual disk format should be changed back to “zeroedthick” using Storage VMotion. As mentioned earlier, currently, it is not possible to convert the virtual disks of a virtual machine between the “thick” and “thin” allocation mechanisms without also changing the datastore. Therefore, when moving virtual machines between different block sizes only, to avoid unnecessary allocation on the thin pool, it is recommended to follow these steps when performing one of these operations[5]:

1. When deploying a virtual machine from a template, cloning a virtual machine, or migrating a virtual machine between different block sizes, choose the “thin” option.
2. Choose a temporary target datastore (preferably a regular Symmetrix thick device to avoid wasted allocations on the thin pool) that matches the block size of the final desired target datastore.
3. Initiate the migrate/deploy/clone operation
4. Once the initial operation is complete, migrate the virtual machine from the temporary datastore to the desired target thin device using Storage VMotion while using the “thick” option

VMware DRS and Virtual Provisioning

The EMC Symmetrix thin devices behave like any other SCSI disk attached to the VMware ESX Server kernel. If a thin device is presented to all nodes of a VMware DRS cluster group, Virtual Center allows live migration of viable virtual machines on the thin device from one node of the cluster to another using VMware vMotion.

Automatic migrations of virtual machines occur if the “automatic” policy is enabled for the DRS cluster group. Virtual Center will suggest migration recommendations for virtual machines if the VMware DRS cluster group automation level is set to Partially Automated or Manual. These behaviors are not different from any other VMware DRS cluster using shared storage on non-thin devices.

Nondisruptive expansion of VMFS datastores on thin devices

Thin devices presented to a VMware vSphere environment should be configured to address the storage requirement for the lifetime of the infrastructure. This approach is viable since thin devices do not require equivalent physical storage initially. Physical capacity can be added to the thin pool as the environment grows.

Therefore, a properly configured and sized thin device should not have a need for expansion. Nevertheless, it is possible to expand a thin device nondisruptively while maintaining existing data on the devices. To exploit this functionality, the thin devices need to be presented as a concatenated (*not* striped) metavolume to the ESX cluster. New in vSphere, VMFS Volume Grow allows administrators to take advantage of expanded thin metavolumes without resorting to using multiple extents.

VMFS Volume Grow offers a new way to increase the size of a datastore that resides on a VMFS volume. It complements the dynamic LUN expansion capability that exists in Symmetrix DMX and Symmetrix VMAX storage arrays. If a LUN is increased in size, then VMFS Volume Grow enables the VMFS volume extent to dynamically increase in size as well. Often, virtual disks threaten to outgrow their current VMFS datastores and they either need to be moved to a new, larger datastore or the current datastore needs to be increased in size to satisfy the growing storage requirement. Prior to vSphere, the option for increasing the size of an existing VMFS volume was to extend via a process called spanning. Even if the newly available space was located on the LUN on which the original VMFS volume resided the only option was to add an extent. A separate disk partition could be created on that additional space and then the new partition could be added to the VMFS volume as an extent. With VMFS Volume Grow the existing partition table is changed and extended to include the additional capacity of the underlying LUN instead of creating a second partition. The process of increasing the size of the VMFS volume is integrated through the vCenter GUI. Provided that additional capacity on the existing extent is there, or has been recently been increased in capacity, the VMFS volume can now be expanded dynamically up to the approximately 2 TB[6] limit per

[5] This entire process to fix the incorrect allocation of zeros can be avoided by using zero page reclamation available on the Symmetrix VMAX with Enginuity 5874 and Solutions Enabler 7.1. More information on doing so can be found in the section, “Virtual Provisioning space reclamation in VMware vSphere .”
[6] The actual limit is 2 terabytes minus 512 bytes.

LUN. For VMFS volumes that might already span multiple extents, the VMFS Volume Grow can be used to grow each of those extents up to approximately 2 TB as well. Avoiding using multiple extents removes unnecessary complexity to managing storage back-end information of a given VMFS datastore.

Thin metavolumes work similarly to regular metavolumes and as a result use the same commands (form meta, dissolve meta, convert meta) to be configured and altered. A few notes should be mentioned when thin metavolumes are used.

Only thin devices can be used to create a thin metavolume. Mixing of thin and non-thin devices to create a metavolume is not allowed.

Only unbound thin devices can be formed into or added to a metavolume.

For a striped metavolume, all members should have the same device size.

Figure 34 shows the process to expand the concatenated metavolume by adding additional thin devices. The existing data is preserved during the expansion operation.

```
[root@LicoF016 ~]# symconfigure -sid 1265 -cmd "add dev 0451:0452 to meta 044D;" -v commit -nop

A Configuration Change operation is in progress. Please wait...

    Establishing a configuration change session...............Established.
    Processing symmetrix 000192601265
    {
      add dev 0451:0452 to meta 044D;
    }

    Performing Access checks..................................Allowed.
    Checking Device Reservations..............................Allowed.
    Locking devices...........................................Locked.
    Initiating COMMIT of configuration changes................Queued.
    COMMIT requesting required resources......................Obtained.
    Local:  COMMIT............................................Done.
    Terminating the configuration change session..............Done.

The configuration change session has successfully completed.
```

Figure 34. Expanding a concatenated metavolume using Solutions Enabler 7.0

After the expansion is complete (and a rescan is performed), VMware vSphere recognizes the additional storage available on the newly expanded thin device. Using VMFS Volume Grow, the VMware file system can then be expanded to occupy the additional storage without the need for using multiple extents.

In vCenter, as seen in Figure 35, the datastore vSphere_Expand_DS is full. It can also be seen in the figure that the underlying device currently had no extra storage for VMFS expansion. The datastore consumes approximately 256 GB on a thin metavolume that is 256 GB in size. The process to expand the datastore on an enlarged thin metavolume is exemplified in the next paragraph.

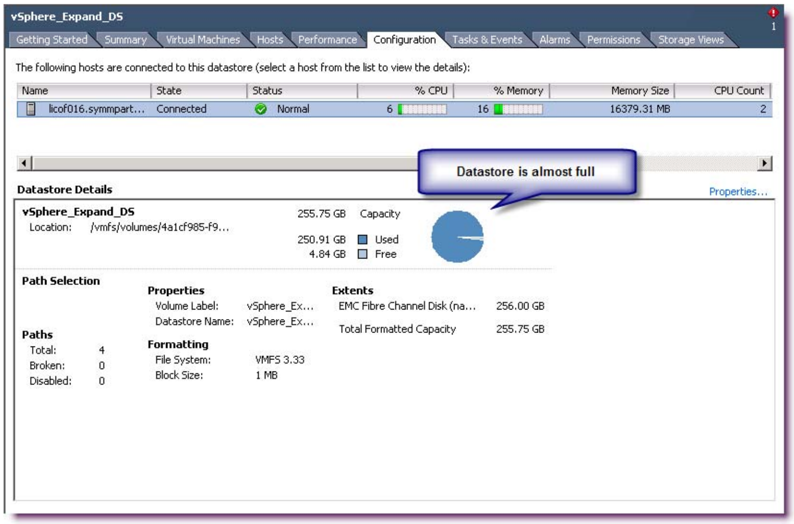


Figure 35. VMFS datastore before being grown using VMFS Volume Grow

After expanding the thin metavolume using storage array utilities (SYMCLI or SMC), the SCSI bus on one of the ESX Servers accessing the datastore should be rescanned. Once the rescan has been executed, the VMFS datastore can be grown. To do so, in vSphere Client, navigate to the datastores listing on any ESX server and select the datastore that needs to be expanded. As shown in Figure 36, clicking the **Properties** link results in a new window shown in Figure 37.

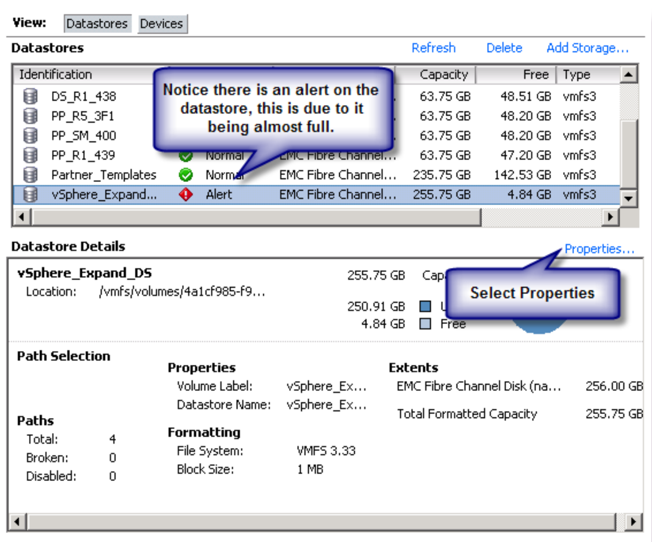


Figure 36. VMFS properties in vSphere Client

In the Datastore Properties pop-up window, click the **Increase** button (Figure 37) and on the subsequent window select the same device on which the datastore currently resides.

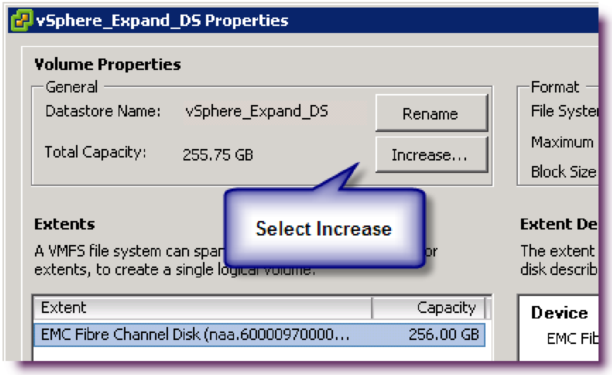


Figure 37. Initiating VMFS Volume Grow in vSphere Client

The following window (Figure 38) summarizes the disk layout of the VMFS volume. The figure shows that the datastore currently uses 256 GB and has an additional 128 GB of free space.

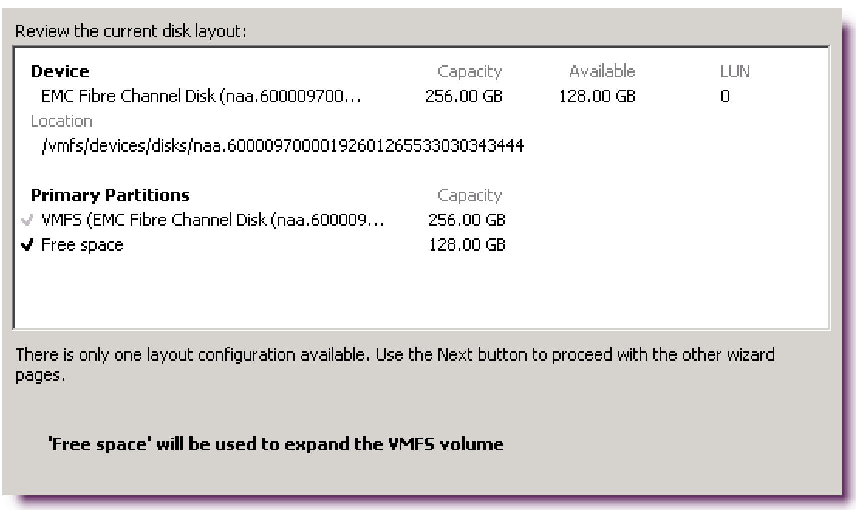


Figure 38. Confirming available free space in the VMFS Volume Grow wizard

The option to use only part of the newly available space is offered when the next step of the wizard is selected. For this example, all of the additional space will be merged into the VMFS volume. This is achieved by selecting the **Maximize capacity** checkbox as shown in Figure 39.

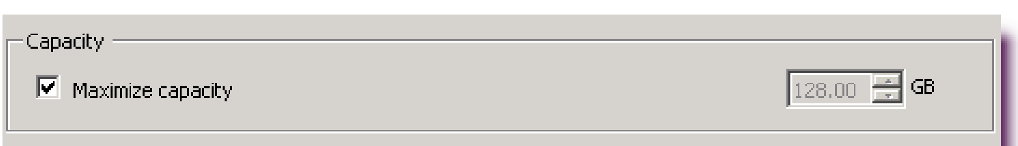


Figure 39. Choosing how much additional capacity for the VMFS Volume Grow operation

The final result is shown in Figure 40. As it can be seen, the datastore vSphere_Expand_DS now occupies all 384 GB of the metavolume.

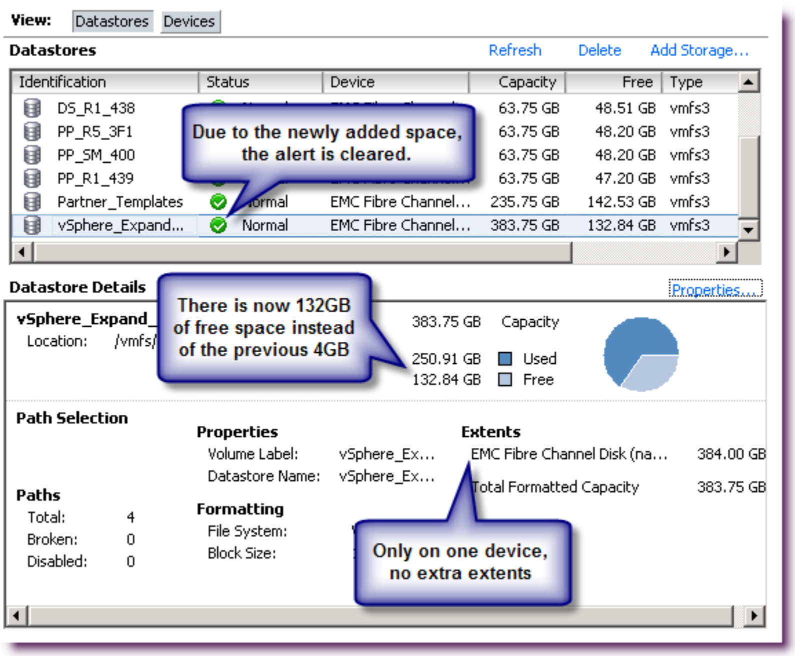


Figure 40. Newly expanded VMFS volume

Use of Enginuity 5874 SR1 thin pool space reclamation in VMware environments

In a VMware environment, the zeroing out of virtual disks can lead to unnecessary physical storage allocation on Symmetrix thin pools, a situation that is inherently detrimental to the space-saving benefit of Symmetrix Virtual Provisioning. Introduced in the Enginuity 5874 SR1 for the Symmetrix VMAX is the ability to perform space reclamation on thin devices hosting, for example, zeroed-out virtual disks. In VMware vSphere 4.0, there are situations where performing space reclamation on thin devices will be helpful. In the next few subsections, Symmetrix Virtual Provisioning space reclamation and the considerations for it in a vSphere environment are discussed in detail.

Virtual Provisioning space reclamation in VMware vSphere environments

In VMware vSphere 4.0, the ability to choose virtual disk allocation options allows administrators to deploy virtual machines using the thin, zeroedthick, or eagerzeroedthick mechanisms. Since newly provisioned virtual machines cloned from existing virtual machines or templates are not forced into the eagerzeroedthick format as in Virtual Infrastructure 3, virtual machines created in vSphere are typically Symmetrix Virtual Provisioning "friendly."

As shown in Figure 41, if a new virtual machine is created with **Support clustering features such as Fault Tolerance** selected the virtual disks for the virtual machine are created using the "eagerzeroedthick" allocation mechanism, and if placed on a thin device, results in a complete allocation. Space reclamation can be used in this instance to reclaim the consumed space on the thin pool. However, if the virtual machine is targeted to be used with VMware Fault Tolerance, the zeros should not be reclaimed. VMware fully allocates virtual disks of Fault Tolerant machines purposefully and the zeros are not meant to be removed and could lead to issues with the use of the Fault Tolerant virtual machine.

As discussed in the previous section, "Virtual machines on VMware file systems of different block sizes," zero page reclamation can be used as an alternative to using multiple Storage VMotion operations. With zero page reclamation, users can deploy/clone/migrate the virtual machines across block sizes using the "thick" option. This will result in a full allocation on the thin pool, but through the use of zero page reclamation, these zeros can be reclaimed resulting in optimal allocation.

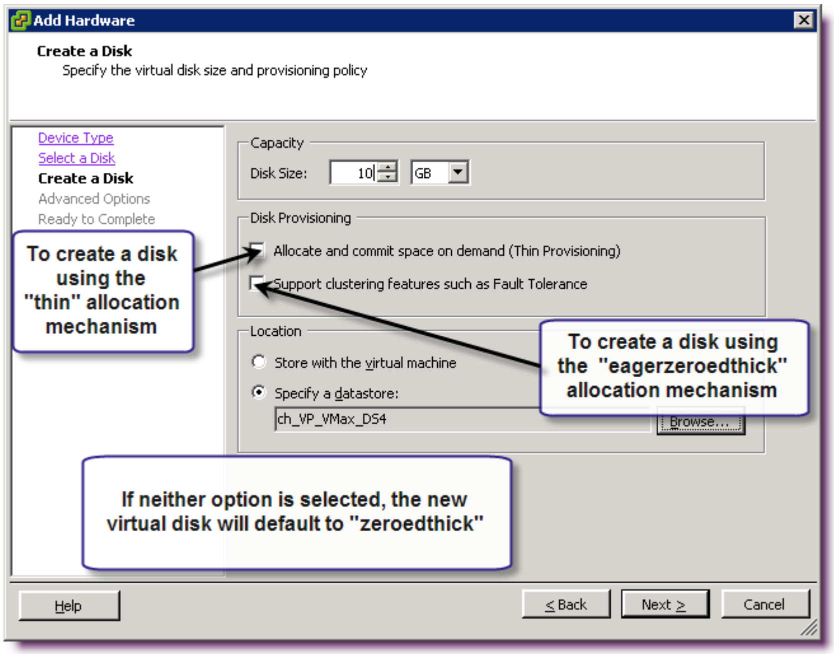


Figure 41. Virtual disk allocation mechanism options in vSphere 4.0

Migrating from Virtual Infrastructure 3.x to vSphere 4.0

The migration of a VMware environment from Virtual Infrastructure 3 to vSphere 4 is a prime candidate for Symmetrix space reclamation. Templates, virtual machines deployed from templates, and virtual machines produced from the cloning process in Virtual Infrastructure 3 result in fully allocated virtual disks. In vSphere environments the zeroed blocks can be reclaimed by using the integrated Storage VMotion feature. But this requires two VMotion tasks: one to convert the virtual machine to "thin" to reclaim the zeros, and a second VMotion it to "zeroedthick" to follow best practices when storing virtual machines on Symmetrix thin devices. In the cases of large virtual machine farms it is laborious and inefficient to manually convert each virtual machine from eagerzeroedthick to "thin" to the optimal "zeroedthick" format for Symmetrix Virtual Provisioning. Instead, space reclamation can be used on each Symmetrix thin device to remove the zeroed out allocations while simultaneously offloading the host CPU cycles and memory requirements consumed by Storage VMotion.

Considerations for space reclamation in VMware environments

For vSphere 4.0, there are important considerations that should be evaluated before performing reclaims.

If a thin device stores multiple virtual machines, a space reclamation function cannot be performed on just one of the virtual disks. It is currently not possible to determine the specific blocks each virtual disk consumes on the thin device and therefore it is not possible to limit a reclaim to a single virtual disk out of many on a given VMware file system[7]. Consequently, a reclaim should only be performed on a thin device if all virtual machines hosted by the VMware file system volume on that thin device can and should be reclaimed. In particular, in vSphere environments, virtual machines that are set up in a Fault Tolerant configuration should not share Symmetrix thin devices (or the VMware file systems that are hosted on

[7] EMC is actively working with VMware on the vStorage initiative that may allow in the future for space reclamation to be performed on a single virtual disk.

them) with other virtual machines. This will reduce the risk of reclaiming the zeros from a Fault Tolerant virtual machine.

Additionally, zeroing out by guest operating systems must be taken into account before performing space reclamation. If the virtual machines' operating system or its hosted applications zero out files for a particular reason, any implications of removing those zeros should be considered before reclaiming them.

The use of Virtual Storage Integrator 3.0 (available for Virtual Infrastructure 3.5 and vSphere 4.0) is highly recommended in order to easily map virtual machines and their corresponding VMware file system volumes to the correct underlying Symmetrix thin device. Double-checking storage mapping information with the Virtual Storage Integrator 3.0 will eliminate the possibility of performing space reclamation on an incorrect thin device.

Local and remote replication between thin devices

Organizations will be able to perform "thin to thin" replication with Symmetrix thin devices by using standard TimeFinder®, SRDF®, and Open Replicator operations. This includes TimeFinder/Snap, TimeFinder/Clone, SRDF/Synchronous, SRDF/Asynchronous, and SRDF/Data Mobility.

In addition, thin devices can be used as control devices for hot and cold pull and cold push Open Replicator copy operations. If a push operation is done using a thin device as the source, zeroes will be sent for any regions of the thin device that have not been allocated, or that have been allocated but have not been written to.

Open Replicator can also be used to copy data from a regular device to a thin device. If a pull or push operation is initiated from a regular device that targets a thin device, then a portion of the target thin device, equal in size to the reported size of the source volume, will become allocated.

Local replication between thin and regular devices

TimeFinder/Clone supports the creation of clone sessions between regular devices and thin devices on Symmetrix VMAX arrays as of the Enginuity 5874 Q4 2009 service release.

When a clone copy is made between a regular source device and a thin target device, thin device extents that have never been written to by a host will not be copied to the target volume. Following the clone copy, any thin device extents that were allocated on the clone target that contain all zeros can be reclaimed and added back to the target thin device's pool. This is accomplished using Virtual Provisioning space reclamation.

Performance considerations

The architecture of Virtual Provisioning creates a naturally striped environment where the thin extents are allocated across all volumes in the assigned storage pool. The larger the storage pool for the allocations is, the greater the number of devices that can be leveraged for VMware vSphere I/O.

One of the possible consequences of using a large pool of data devices and potentially sharing these devices with other applications is variability in performance. In other words, possible contention with other applications for physical disk resources may cause inconsistent performance levels. If this variability is not desirable for a particular application, that application could be dedicated to its own thin pool. Symmetrix arrays support up to 512 thin pools. This includes Virtual Provisioning thin pools, SRDF/A Delta Set Extension (DSE) pools, or TimeFinder/Snap pools.

When a new data extent is required from the thin pool there is an additional latency introduced to the write I/O while the thin extent location is assigned and formatted. This latency is approximately 1 millisecond. Thus, for a sequential write stream, a new extent allocation will occur on the first new allocation, and again when the current stripe has been fully written and the writes are moved to a new extent. If the application cannot tolerate the additional latency it is recommended to preallocate storage to the thin device when the thin device is bound to the thin pool.

Thin pool management

When storage is provisioned from a thin pool to support multiple thin devices there is usually more "virtual" storage provisioned to hosts than is supported by the underlying data devices. This is one of the main reasons for using Virtual Provisioning. However, there is a possibility that applications using a thin pool may grow rapidly and request more storage capacity from the thin pool than is actually there. This condition is known as oversubscription. This is an undesirable situation. The next section discusses the steps necessary to avoid oversubscription.

Thin pool monitoring

Along with Virtual Provisioning come several methodologies to monitor the capacity consumption of the thin pools. The Solutions Enabler `symcfg monitor` command can be used to monitor pool utilization, as well as display the current allocations through the `symcfg show pool` command. There are also event thresholds that can be monitored through the SYMAPI event daemon, thresholds that can be set with the Symmetrix Management Console and SNMP traps that can be sent for monitoring by EMC Ionix™ ControlCenter® or any data center management product.

System administrators and storage administrators must put processes in effect to monitor the capacity for thin pools to make sure that they do not get filled. The pools can be dynamically expanded to include more data devices without application impact. These devices should be added in groups long before the thin pool approaches a full condition. If devices are added individually, hot spots on the disks can be created when much of the write activity is suddenly directed to the newly added data devices because other data devices in the pool are full. With the introduction of thin pool write balancing in Enginuity 5874 SR1 the data devices can be added individually without introducing hot spots on the pool. This functionality is not on the Symmetrix DMX or included in the initial release of 5874 code on the Symmetrix VMAX.

Exhaustion of oversubscribed pools

If a thin pool is oversubscribed and has no available space for new extent allocation, different behaviors can be observed depending on the activity that caused the thin pool capacity to be exceeded. If the capacity was exceeded when a new virtual machine was being deployed using the Virtual Center wizard, the error message shown in Figure 42 is posted. An I/O error message is displayed when the thin pool capacity is exceeded while using command line utilities on the service console.

The behavior of virtual machines when the thin pool is oversubscribed depends on a number of factors including the guest operating system, the application running in the virtual machine, and also the format that was utilized to create the virtual disks. The virtual machines behave no differently than the same configuration running in a non-virtual environment. In general the following comments can be made for VMware vSphere environments:

- Virtual machines configured with virtual disks using the "eagerzeroedthick" format continue to operate without any disruption.
- Virtual machines that do not require additional storage allocations continue to operate normally.
- If any virtual machine in the environment is impacted due to lack of additional storage, other virtual machines continue to operate normally as long as those machines do not require additional storage.
- Some of the virtual machine may need to be restarted after additional storage is added to the thin pool. In this case, if the virtual machine hosts an ACID-compliant application (such as relational databases), the application performs a recovery process to achieve a transactionally consistent point in time.
- The VMkernel continues to be responsive as long as the VMware ESX server is installed on a device with sufficient free storage for critical processes.

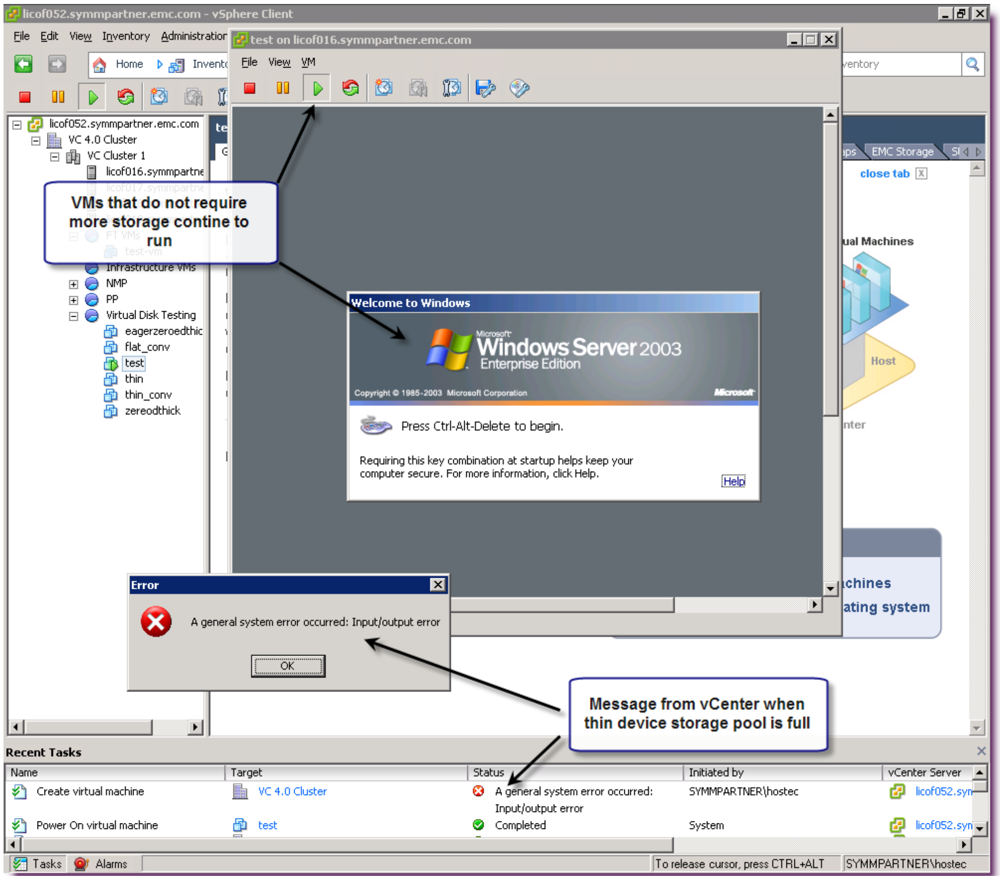


Figure 42. Error message displayed by vSphere Client on thin pool full conditions

Conclusion

EMC Symmetrix Virtual Provisioning provides a simple, noninvasive, and economical way to provide storage for VMware vSphere 4 environments. The VMware file system behavior dovetails very nicely into this type of configuration, while others may not allow VMware administrators to realize the full benefits of the product.

The following scenarios are optimal for use of Virtual Provisioning with VMware vSphere 4:

- Provisioned virtual machines that use the “zeroedthick” (referred to as “thick”) allocation mechanism
- Creation of virtual machine templates on virtually provisioned devices using the “thin” allocation mechanism with the Clone to Template wizard
- Systems where overallocation of storage is typical
- Systems where rapid growth is expected over time but downtime is limited

The following are situations where Virtual Provisioning may not be optimal:

- Provisioned virtual machines that use the “eagerzeroedthick” allocation mechanism
- VMware vSphere environments that cannot tolerate an occasional response time increase of approximately 1 millisecond due to writes to uninitialized blocks
- Virtual machines in which data is deleted and re-created rapidly and the underlying data management structure does not allow for reuse of the deleted space
- Provisioning new virtual machines with the VMware converter directly onto Symmetrix thin pools

References

White papers

New Features in EMC Enginuity 5773 for Symmetrix Open Systems Environments

New Features in EMC Enginuity 5874 for Symmetrix Open Systems Environments

Implementing Virtual Provisioning on EMC Symmetrix with VMware Infrastructure 3

Implementing Virtual Provisioning on EMC Symmetrix VMAX with Oracle 10g and 11g

Implementing Virtual Provisioning on EMC Symmetrix with Microsoft Exchange Server

Implementing Virtual Provisioning on EMC Symmetrix with Microsoft SQL Server

Implementing Virtual Provisioning on EMC Symmetrix with DB2 LUW Databases

Implementing Virtual Provisioning on EMC Symmetrix DMX with Sybase ASE

Technical note

Best Practices for Fast, Simple Capacity Allocation with EMC Symmetrix Virtual Provisioning

Feature sheet

Symmetrix Virtual Provisioning Feature Specification

Product documentation

EMC Solutions Enabler Symmetrix Array Management CLI Product Guide

EMC Solutions Enabler Symmetrix Array Controls CLI Product Guide

EMC Solutions Enabler Symmetrix CLI Command Reference