Abstract
This white paper discusses the benefits of Virtual Provisioning™ on EMC® VNX™ storage systems. It provides an overview of this technology, and describes how Virtual Provisioning is implemented on the VNX.

December 2013
Copyright © 2013 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

Part Number h2222.4
# Table of Contents

**Executive summary** .................................................................................................................. 5

**Introduction** ................................................................................................................................. 5

**Audience** ........................................................................................................................................ 6

**Terminology** .................................................................................................................................... 6

**Business requirements** ................................................................................................................. 8

**Traditional storage provisioning and Virtual Provisioning** ......................................................... 9

**Pools** ............................................................................................................................................. 11

- Per-Tier RAID Configuration ........................................................................................................ 11
- Homogeneous pools ....................................................................................................................... 12
- Heterogeneous pools ..................................................................................................................... 12
- Pool Attributes ............................................................................................................................... 13
- Monitoring, adding, and deleting pool capacity ........................................................................... 15

**Pool LUNs** .................................................................................................................................. 17

- Thin LUNs ...................................................................................................................................... 17
- Thick LUNs ................................................................................................................................... 17
- LUN attributes ............................................................................................................................... 18
- Expanding Pools ............................................................................................................................ 19
- Expanding and shrinking pool LUNs ............................................................................................. 20

**Best practices** ............................................................................................................................. 20

**Using virtual provisioning for file** .............................................................................................. 21

- Creating a file system on pool LUNs ............................................................................................ 21
- Thin file systems .......................................................................................................................... 25
  - Technology considerations ........................................................................................................... 28
  - Monitoring File Systems ............................................................................................................. 29
  - Performance considerations ........................................................................................................ 30

**When to use traditional, thick, and thin LUNs** ............................................................................. 30

**Thin LUN space reclamation via Migration** .................................................................................. 31

**Data protection** ............................................................................................................................ 32

- RecoverPoint .................................................................................................................................. 32
- VNX Snapshots .............................................................................................................................. 33
- SnapView ...................................................................................................................................... 33
- SnapSure ....................................................................................................................................... 33
- MirrorView ..................................................................................................................................... 33
- VNX Replicator ............................................................................................................................. 34
- PowerPath Migration Enabler ......................................................................................................... 34
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using thin LUNs with applications</td>
<td>34</td>
</tr>
<tr>
<td>Host-based File Systems</td>
<td>35</td>
</tr>
<tr>
<td>Virtual environments</td>
<td>35</td>
</tr>
<tr>
<td>Veritas Storage Foundation</td>
<td>36</td>
</tr>
<tr>
<td>Conclusion</td>
<td>38</td>
</tr>
<tr>
<td>References</td>
<td>38</td>
</tr>
<tr>
<td>Appendix A: Virtual Provisioning limits</td>
<td>40</td>
</tr>
<tr>
<td>Appendix B: Guidelines for pools and pool LUNs</td>
<td>41</td>
</tr>
</tbody>
</table>
Executive summary

EMC® VNX™ Virtual Provisioning provides pool-based storage provisioning by implementing pool LUNs that can be either thin or thick. Thin LUNs provide on-demand storage that maximizes the utilization of your storage by allocating storage as it is needed. Thick LUNs provide high and predictable performance for your applications mainly because all of the user capacity is reserved and allocated upon creation. Both types of LUNs benefit from the ease-of-use features of pool-based provisioning. Pools and pool LUNs are also the building blocks for advanced data services such as Fully Automated Storage Tiering for Virtual Pools (FAST VP) and Compression.

VNX’s Virtual Provisioning technology supports features such as hot sparing, proactive sparing, and the ability to migrate data between thin LUNs, thick LUNs, or traditional LUNs without incurring application downtime. The ability to non-disruptively migrate data to different LUN and disk types provides the best solution for meeting your changing application and business requirements without incurring downtime. This flexibility separates VNX Virtual Provisioning from typical thin provisioning implementations.

Virtual Provisioning™ enables organizations to reduce storage costs by increasing capacity utilization, simplifying storage management, and reducing application downtime. Virtual Provisioning also helps companies to reduce power and cooling requirements and reduce capital expenditures.

Introduction

One of the biggest challenges facing storage administrators is balancing the storage requirements for various competing applications in their data centers. Administrators are typically forced to allocate space, initially, based on anticipated storage growth. They do this to reduce the management expense and application downtime incurred when they need to add more storage as their business grows. This generally results in the overprovisioning of storage capacity, which then leads to higher costs; increased power, cooling, and floor space requirements; and lower capacity utilization rates. Even with careful planning, it may be necessary to provision additional storage in the future. This may require application downtime depending on the operating systems involved.

VNX thin LUN technology addresses these concerns. Thin LUNs can present more storage to an application than is physically available. Storage managers are freed from the time-consuming administrative work of deciding how to allocate drive capacity. Instead, an array-based mapping service builds and maintains all of the storage structures based on a few high-level user inputs. Disk drives are grouped into pools that form the basis for provisioning actions. Physical storage is automatically allocated only when new data blocks are written.
**Thick** LUNs are also available in VNX. Thick and thin LUNs can share the same pool, allowing them to have the same ease-of-use benefits of pool-based provisioning. However, unlike a thin LUN, a thick LUN’s capacity is fully reserved on creation. A thick LUN provides better performance than a thin LUN. However, a thick LUN does not provide the flexibility of overprovisioning like a thin LUN does.

VNX provides a unified approach to virtual provisioning file and block data. File data served by VNX Data Movers can be hosted in pools. This allows files to have the benefits of pool-based provisioning as well as FAST VP.

Virtual Provisioning also simplifies storage management tasks by providing the ability to expand pool LUNs (thick and thin LUNs) with a few simple clicks. The underlying pools can also be expanded by adding drives non-disruptively when additional physical storage space is required. The only exception is that when a Pool LUN is associated with a file system, expansion is not allowed. Using Virtual Provisioning reduces the time and effort required to provision additional storage, and avoids provisioning storage that may not be needed.

You can manage pools and pool LUNs using the Unisphere™ software or Secure Command Line Interface (CLI) commands. VNX replication products, FAST Cache, Unisphere Analyzer, and Unisphere Quality of Service Manager work seamlessly across pool LUNs. EMC RecoverPoint also supports the replication of VNX pool LUNs.

Pools and pool LUNs are the building blocks for advanced data services such as FAST VP and Compression. FAST VP requires a license, while Compression does not require a license. For more detailed information on compression, refer to the *EMC VNX Deduplication and Compression* white paper. For more detailed information on FAST VP see the *EMC VNX FAST VP for Unified Storage Systems* white paper. Both papers are found on the EMC Online Support website.

The following are the key enhancements now available in the new EMC® VNX™ Operating Environment (OE) for block release 5.32, file version 7.1.

- **Per-Tier RAID Selection + Additional RAID configurations**
- **Pool Rebalancing upon expansion**

**Audience**

This white paper is intended for IT planners, storage architects, administrators, and others involved in evaluating, managing, operating, or designing VNX storage systems.

**Terminology**

The following terminology appears in this white paper:

**Allocated capacity** — For a pool, this is the space currently used by all LUNs in the pool. For a thin LUN, this is the physical space used by the LUN. For a thick LUN, this is the host-visible capacity used by the LUN. Allocated capacity is slightly larger than the capacity used by the host because metadata exists at the pool LUN level.
**Available capacity** — The amount of actual physical pool space that is currently not allocated for pool LUNs.

**Automatic Volume Management (AVM)** — Feature of VNX that creates and manages volumes automatically. AVM organizes volumes into storage pools for file that can be allocated to file systems.

**Total Allocation** — See “Allocated capacity”.

**High water mark** — Trigger point at which VNX performs one or more actions, such as sending a warning message, extending a volume, or updating a file system, as directed by the related feature’s software/parameter settings.

**LUN migration** — A VNX feature that migrates data to another traditional LUN, pool LUN, or metaLUN without disrupting running applications.

**Mapped pool** — A storage pool for file that is created during the normal storage discovery (diskmark) process for use on VNX for File. It is a one-to-one mapping with a VNX storage pool for block.

A mapped pool can contain a mix of different types of LUNs that use any combination of data services (thin, thick, auto-tiered, mirrored, and compressed). However, for best file system performance, mapped pools should contain LUNs that use the same data services. For example, a mapped pool should not contain thin and thick LUNs, or mirrored and non-mirrored LUNs, or compressed and non-compressed LUNs, and so forth.

**Oversubscribed capacity** — The amount of user capacity configured for pool LUNs that exceeds the physical capacity in a pool. Oversubscribing capacity is supported via thin LUNs.

**Pool LUN** — A logical unit of storage created in a pool. A pool LUN can be either a thin LUN or a thick LUN.

**Sdelete** — A Windows command line utility that allows you to delete one or more files or directories, or to clean the free space on a logical disk.

**Slice** — A 1 GB unit of capacity, which represents the minimum amount of capacity that can be physically allocated to a pool LUN. Pool LUNs are comprised of slices.

**Storage Pool for block** — A group of disk drives for configuring pool LUNs (thick and thin). There may be zero or more pools in a storage system. Disks can only be a member of one pool, and they cannot also be in a separate user-defined RAID group.

**Storage Pool for file** — Groups of available disk volumes organized by AVM that are used to allocate available storage to file systems. They can be created automatically when using AVM or manually by the user.

**Subscribed capacity** — The total amount of capacity configured for LUNs in the pool. This number can be greater than the available user capacity. The available user capacity can be expanded by adding drives to the pool.

**Thick LUN** — A type of pool LUN in which allocated physical space is equal to the user capacity seen by the host server.
**Thin-friendly** – A term that is frequently used for file systems and applications that do not pre-allocate all of the storage space during initialization. This term is also used for file systems that reuse deleted space before consuming additional storage. Both of these features improve capacity utilization in thin provisioning.

**Thin file system** – A thin file system that lets you *allocate* storage based on long-term projections, while you consumed only the file system resources that you currently need. NFS or CIFS clients and applications see the virtual maximum size of the file system of which only a portion is physically allocated.

**Thin LUN** — A type of pool LUN where physical space allocated can be less than the user capacity seen by the host server.

**Threshold alert** — An alert issued when the % Full Threshold has been exceeded.

**Traditional LUN** — A logical unit of storage created on a user-defined RAID group. The amount of physical space allocated is the same as the user capacity seen by the host server. Traditional LUNs cannot be created on a pool; they are always created on a RAID group.

**Total Capacity** — The total amount of physical storage capacity in the pool that is available for pool LUNs. This is also referred to as “usable capacity.” It is measured as raw disk capacity minus overhead (RAID overhead and mapping overhead). For a pool LUN, this is the size of the LUN as it appears to the host. For pool LUNs, this is sometimes called *host visible capacity*.

**VMware® Distributed Resource Scheduler (DRS)** – A utility in VMware that balances the computing workload (the CPU, memory, and network) with available resources in a virtualized environment.

**Volume** — On a VNX, a virtual disk into which a file system places data.

**% Full** — The percentage of pool capacity that is currently consumed. It is calculated using this formula:

\[
\text{% Full} = \frac{\text{Consumed capacity}}{\text{User capacity}}
\]

**% Full Threshold** — A parameter that is set by the user. The system generates an alert when this threshold is exceeded.

**Business requirements**

Organizations, both large and small, need to reduce the cost of managing their storage infrastructure while meeting rigorous service level requirements and accommodating explosive storage capacity growth.
Virtual Provisioning addresses several business objectives that have drawn increasing focus:

- **Reducing capital expenditures and ongoing costs**
  Virtual Provisioning reduces capital costs by delivering storage capacity on *actual* demand instead of allocating storage capacity on *anticipated* demand. Ongoing costs are reduced because fewer disks consume less power and cooling, and less floor space.

- **Maximizing the utilization of storage assets**
  Organizations need to accommodate growth by drawing more value from the same or fewer storage resources. Operational efficiency remains an ongoing challenge, as organizations often overallocate storage to applications to reduce the risk of outage and the need to reprovision later on.

- **Reducing the cost of storage administration**
  “Ease-of-use” initiatives span multiple aspects of storage processes, including staff training, initial storage provisioning, the addition of new storage, and the management and monitoring of storage systems. Virtual Provisioning simplifies the process of adding storage.

**Traditional storage provisioning and Virtual Provisioning**

Storage provisioning is the process of assigning storage resources to meet the capacity, availability, and performance needs of applications.

With traditional block storage provisioning, you create a RAID group with a particular RAID protection level and a certain number of drives. RAID groups are restricted to a
single drive type and to a maximum of 16 drives. When LUNs are bound on the RAID group, the host reported capacity of the LUNs is equal to the amount of physical storage capacity allocated. The entire amount of physical storage capacity must be present on Day One, resulting in low levels of utilization. Recovering underutilized space remains a challenge. Figure 2 and Figure 3 show the differences between traditional provisioning and Virtual Provisioning.

![Diagram of traditional storage provisioning](image)

**Figure 2. Traditional storage provisioning**

With traditional provisioning, the storage administrator needs to carefully carve out the storage for an application based on the amount forecasted by the application administrator. There is a tendency for these forecasts to be inflated. In some companies, an application administrator may monitor storage space and ask the storage administrator to provision additional storage. The storage administrator must rely on timely and accurate communications from various applications people to effectively manage storage space utilization. If the application requires more storage than forecasted, then storage can be expanded using metaLUN technology. Creating and maintaining metaLUNs requires additional planning and effort.

Pools can be homogeneous (having a single drive type) or heterogeneous (having different drive types). Heterogeneous pools are the building blocks for data efficiency services like FAST VP, which automatically places the most frequently used data on faster performing drives and places less actively used data on lower performing drives. For more information on FAST VP, refer to the *EMC VNX FAST VP – A Detailed Review* white paper on EMC Online Support.

With Virtual Provisioning, you can select the RAID protection by tier during Pool creation. You can create thin LUNs or thick LUNs in the pools. With thin LUNs, the *user capacity* (storage perceived by the application) can be larger than the *consumed capacity* on the storage system. This simplifies the creation and allocation of storage capacity. Thin LUNs can be sized to accommodate growth without regard for currently available assets. Physical storage is assigned to the server in a capacity-on-demand fashion from a shared pool. With thick LUNs, all the storage is reserved and pre-allocated up front at the time of creation.

Pools and pool LUNs can be expanded on the fly any time additional storage is required. It is a completely non-disruptive procedure which can be achieved with a few simple clicks. Additional steps may be required within the host operating system to recognize the added capacity. You should not be expanding LUNs that are assigned to the File side.
Virtual Provisioning provides the flexibility of choosing between thick or thin LUNs based on the application requirements.

Pools

A pool is somewhat analogous to a RAID group, which is a physical collection of disks on which logical units (LUNs) are created. Pools are dedicated for use by pool (thin and thick) LUNs. Pools can contain a few disks or hundreds of disks, whereas RAID groups are limited to 16 disks. Because of the large number of disks supported in a pool, pool-based provisioning spreads workloads over many resources requiring minimal planning and management effort.

Per-Tier RAID Configuration

A new enhancement in the EMC® VNX™ Operating Environment (OE) for block release 5.32, file version 7.1 is the ability to select RAID protection according to tier, during pool creation you can choose the appropriate RAID type according to drive type. Keep in mind that each tier has a single RAID type and once the RAID configuration is set for that tier in the pool, you cannot change it.

Another RAID enhancement coming in this release is the option for more efficient RAID configurations, users will have the following options in pools (new options noted with an asterisk):

Table 1. RAID Configuration Options

<table>
<thead>
<tr>
<th>RAID Type</th>
<th>Preferred Drive Count Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAID 1/0</td>
<td>4+4</td>
</tr>
<tr>
<td>RAID 5</td>
<td>4+1, 8+1*</td>
</tr>
<tr>
<td>RAID 6</td>
<td>6+2, 14+2*</td>
</tr>
</tbody>
</table>

RAID 5 (8+1) and RAID 6 (14+2) provide 50% savings over the current options, because of the higher data:parity ratio. It should be noted that the tradeoff for higher data:parity ratio translates into larger fault domains and potentially longer rebuild times. This is especially true for RAID 5, with only a single parity drive. Users are advised to choose carefully between (4+1) and (8+1) – according to whether robustness or efficiency is a higher priority. For RAID 6, with 2 parity drives, robustness is less likely to be an issue.
For best practice recommendations, refer to the EMC Unified Storage Best Practices for Performance and Availability – Common Platform and Block Storage white paper on EMC Online Support.

Pools can be homogeneous or heterogeneous. Homogeneous pools have a single drive type (Flash, SAS, or NL-SAS). Heterogeneous pools contain different drive types.

**Homogeneous pools**

Homogeneous pools are recommended for applications with similar and expected performance requirements, and only one drive type (Flash, SAS, or NL-SAS) is available for selection during pool creation.

**Heterogeneous pools**

Heterogeneous can consist of different types of drives. VNX supports Flash, SAS, and NL-SAS drives in one pool. Heterogeneous pools, like all pools, let you select a RAID type by tier.

Heterogeneous pools provide the infrastructure for FAST VP, which facilitates automatic data movement to appropriate drive tiers depending on the I/O activity for that data. The most frequently accessed data is moved to the highest tier (Flash drives) in your pool for faster access, medium activity data is moved to SAS drives, and low activity data is moved to the lowest tier (NL-SAS drives). EMC recommends that you do not deploy heterogeneous pools without FAST VP.

As shown in Figure 5, there can be a maximum of three tiers in a heterogeneous pool, based on the three drive types available on each of the two platforms. Heterogeneous
pools can also have two drive types and still leverage FAST VP. FAST VP does not differentiate tiers by the drive speeds, so users should ensure that they choose drives with common rotation speeds when building their pools.

**Figure 5. Heterogeneous VNX storage pools**

**Pool Attributes**

Pools are simple to create because they require few user inputs:

- Pool Name: For example, “Application Pool 2”
- Disks: Number and type
- Protection level

Figure 6 shows an example of how to create a heterogeneous storage pool with Pool 1 as its name and RAID 5 (4+1) protection for the Flash drives, RAID 5 (4+1) protection for the SAS drives, and RAID 6 (6+2) for the NL-SAS drives.
A VNX can contain one or many pools (up to the limits listed in “Appendix A: Virtual Provisioning limits”). The smallest pool size is 3 drives for RAID 5, 4 drives for RAID 6, and 2 drives for RAID 1/0. However, the recommended RAID Configuration is different for each drive type to ensure performance and availability.

Pools can be as large as the maximum number of drives (except vault drives and hot spares) allowed per system type. For example, a VNX7500 can contain 996 drives in a single pool or between all pools. Vault drives (the first four drives in a storage system) cannot be part of a pool, so Unisphere dialog boxes and wizards do not allow you to select these drives. Large pools must be created by using multiple operations. Depending on the system type, pools can be created by using the maximum allowed drive increment and then expanded until you reach the desired number of drives in a pool. Once the pool is fully initialized, you can create LUNs on it. For example, to create an 80-drive pool on a VNX5300, you need to create a pool with 40 drives and then expand the pool with another 40 drives. You can also expand pools at a later time if more storage is needed. The maximum allowed drive increments for different system types are shown in Table 2.

Users need to be conscious of fault domains. A fault domain refers to data availability. A Virtual provisioning pool is made up of one or more private RAID groups. A pool’s fault domain is a single pool private RAID group. That is, the availability of a pool is the availability of any single private RAID group. Unless RAID 6 is the pool’s level of protection, avoid creating pools with very large number of RAID groups. For more information regarding the benefits of smaller pools refer to the EMC

**Table 2. Drive increments for VNX models**

<table>
<thead>
<tr>
<th>VNX model</th>
<th>Maximum allowed drive increments</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNX5100</td>
<td>20</td>
</tr>
<tr>
<td>VNX5300</td>
<td>40</td>
</tr>
<tr>
<td>VNX5500</td>
<td>80</td>
</tr>
<tr>
<td>VNX5700</td>
<td>120</td>
</tr>
<tr>
<td>VNX7500</td>
<td>180</td>
</tr>
</tbody>
</table>

More detailed information on the limits for different releases, such as disks per pool and pools per system, is provided in “Appendix A: Virtual Provisioning limits.” Also refer to the *EMC Virtual Provisioning Release Notes* for current information.

Users are advised to make every effort to ensure that pools are created with common drive capacities, although this is not enforced in the UI. When creating a pool, different types of disks (Flash, SAS, and NL-SAS) can be different sizes. To maximize space utilization, however, all disks of a particular type should be the same size in each pool because it is possible that the larger disks of the same type will be truncated. If it becomes absolutely necessary to use different capacity drives to create a pool, it is best to create the pool in stages. For example, if you have ten 600 GB SAS drives and five 300 GB SAS drives, first create the pool by selecting only the ten 600 GB drives, and then expand the pool by adding the other five 300 GB drives.

**Monitoring, adding, and deleting pool capacity**

As shown in Figure 7, *total capacity* is the total physical capacity available to all LUNs in the pool. *Total Allocation* is the total physical capacity currently assigned to all pool LUNs. *Subscribed capacity* is the total host reported capacity supported by the pool. *Over-subscribed capacity* is the amount of user capacity configured for LUNs that exceeds the physical capacity in a pool. Thin LUNs with larger user capacity than the physical capacity can result in an oversubscribed pool.

![Figure 7. Pool % full threshold](image-url)
In Figure 8 the **Storage Pool Properties** dialog box shows parameters such as free capacity, percent full, total allocation, total subscription, percent subscribed, and oversubscribed capacity.

![Storage Pool Properties dialog box VNX OE for Block 32](image)

**Figure 8. Storage Pool Properties dialog box VNX OE for Block 32**

Pools are monitored using the **% Full Threshold Alert** setting and storage capacity reports. This alert is only active if there are one or more thin LUNs in a pool, since thin LUNs are the only way to oversubscribe a pool. If the pool only contains thick LUNs, the alert is not active as there is no risk of running out of space due to oversubscription. You can specify the value for **% Full Threshold** (Total Allocation/Total capacity) when a pool is created.

Alerts can be monitored by using the **Alerts** tab in Unisphere. In Unisphere’s Event Monitor wizard, you can also select the option of receiving alerts through email, a paging service, or an SNMP trap. Table 3 displays information about thresholds and their settings.

**Table 3. Threshold alerts VNX OE Block 32**

<table>
<thead>
<tr>
<th>Threshold Type</th>
<th>Threshold Range</th>
<th>Threshold Default</th>
<th>Alert Severity</th>
<th>Alert Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>User settable</td>
<td>1%-84%</td>
<td>70%</td>
<td>Warning</td>
<td>None</td>
</tr>
<tr>
<td>Built-in</td>
<td>N/A</td>
<td>85%</td>
<td>Critical</td>
<td>Clears user settable alert</td>
</tr>
</tbody>
</table>
Allowing total allocation to exceed 90% of total capacity puts you at the risk of running out of space and impacting all applications using LUNs in the pool.

**Pool LUNs**

A VNX pool LUN is similar to a traditional LUN in many ways. Many of the same Unisphere operations and Secure CLI commands can be used on pool LUNs and traditional LUNs. Most user-oriented functions work the same way, including underlying data integrity features, LUN migration, local and remote protection, and LUN properties information. Pool LUNs are available as choices in Unisphere Tasks, Unisphere Analyzer, and Unisphere Quality of Service Manager. Features such as hot sparing and proactive sparing also work with pool LUNs. It is also possible to migrate a traditional LUN (or metaLUN) to a pool LUN and vice versa.

**Thin LUNs**

The primary difference between a thin LUN when compared to traditional LUNs and thick LUNs is that thin LUNs present more storage to an application than is physically allocated. Presenting storage that is not physically available avoids underutilizing the storage system’s capacity. Thin LUNs consume storage as needed from the underlying pool.

Thin LUNs typically have lower performance than thick LUNs because all of the user capacity is reserved upon creation of a thick LUN so the mapping requirements for a thick LUN are much less compared to a thin LUN. However, thin LUNs result in higher capacity utilization.

**Thick LUNs**

With the introduction of the new EMC® VNX™ Operating Environment (OE) for block release 5.32, file version 7.1 there is no change in behavior for Thin LUNs; but Thick LUNs will be fully allocated upon creation. This means that all 1 GB slices will be allocated when the LUN is first created. The writes will still be organized according to Logical Block Address range, as was originally the case.

Thick LUNs and thin LUNs can share the same pool. When a thick LUN is created, all of its capacity is reserved and allocated in the pool for use by that LUN. Therefore, a thick LUN will never run out of capacity. Any new writes are located and distributed in all the pre-allocated area, so the host reported capacity is roughly equal to the consumed capacity. Thick LUNs are higher performing than thin LUNs because of the direct addressing, and should be used for applications where performance is more important than space savings.

This enhancement has beneficial implications for FAST VP, in that by fully allocating the Thick Pool LUN upon creation, users can better control which tier the slices are written to. As pools are initially being created, and there is still sufficient space in the highest tier, users can be assured that when they create a LUN with either “Highest Available Tier” or “Start High, then Auto-Tier,” data will be written to the highest tier – because the LUN is allocated immediately.
**LUN attributes**

The maximum size for a pool LUN is 16 TB. The minimum user capacity size is 1 MB. However, the consumed capacity for the smallest LUN is 3 GB because of overhead. Pool LUNs are simple to create, with four inputs:

- Pool selection
- Amount of host visible user capacity
- Pool LUN name (optional)
- Thin or not

A thin LUN can be created by selecting the **Thin** checkbox, as shown in Figure 9.

![Create LUN dialog box for VNX OE for Block 32](image)

**Figure 9.** Create LUN dialog box for VNX OE for Block 32

There are only a few restrictions with pool LUNs and they are enforced through Unisphere:

- A pool LUN cannot be used as a component of a metaLUN.
- A thin LUN cannot be used in the reserved LUN pool (which reserves space for SnapView snapshots), but thick LUNs can be used.
Pools and subsequently pool LUNs are covered with default cache options that aren’t tunable like traditional LUNs. The only cache option for a pool is to enable or disable FAST Cache when available.

When you create a thin LUN, there is a small initial capacity allocation to the LUN. Figure 10 shows the 3 GB initial consumed capacity to the LUN. Some of the initial allocation is used for LUN metadata, and the remaining space is available for incoming writes.

![Figure 10. Create Thin LUN dialog box.](image)

As new writes come into a thin LUN, more physical space is allocated in 1 GB slices. This storage-on-demand algorithm ensures that at least a 1 GB slice is available at all times. LUN metadata is also embedded within these additional 1 GB slices. Data is written to thin LUNs in 8 KB chunks and optimally placed within the 1 GB slices.

**Expanding Pools**

You can expand a pool by adding drives to it. It is a completely non-disruptive operation, and the increased capacity can be used by LUNs in that pool. Total allocation is reclaimed by the pool when LUNs are deleted. Oversubscribed pools can run out of space. Therefore, it is a best practice to ensure that a monitoring strategy is in place and you have the appropriate resources to expand the pool.

When a storage pool is expanded, the sudden introduction of new empty disks combined with relatively full existing disks causes a data imbalance. This imbalance is resolved by automating a one-time data relocation, referred to as rebalancing. This
rebalance relocates slices within the tier of storage that has been expanded, to achieve best performance. Rebalancing occurs both with and without the FAST VP enabler installed.

**Expanding and shrinking pool LUNs**

VNX has the capability to expand pool LUNs, with a few simple clicks, making the expanded capacity immediately available.

For a thick LUN, the pool must have enough storage for the expansion to succeed, whereas for a thin LUN the storage does not need to be available. It is important to note that you cannot expand a pool LUN if it is part of a data protection or LUN-migration operation. VNX for File offers mechanisms for expanding the capacity of file systems. This is discussed in more detail in the “Using virtual provisioning for file” section.

Also available in VNX is the capability of *shrinking* pool LUNs. This capability is only available for LUNs that are served by Windows Server 2008, because it is a server-controlled operation, and Windows Server 2008 is currently the only server OS that has this feature built in. The shrinking process involves two steps:

1. Shrink the file system from Windows Disk Management.
2. Shrink the pool LUN using a command window and the DISKRAID utility. The utility is available through VDS Provider, which is part of the EMC Solutions Enabler package.

The new LUN size is shown as soon as the shrink process is completed. A background task is triggered to reclaim the deleted or shrunk space and return it back to the pool. Once the task is completed, the reclaimed space can be used by any other LUN in that pool.

**Best practices**

The following are the recommended best practices for managing pools:

1. **EMC recommends that you do not** change the default owner of the pool LUN once it is provisioned. This can adversely affect performance. It changes the underlying private structures, which provide storage for the pool LUN, which are still controlled by the original SP.

2. **It is a best practice to let the system balance the pool LUNs on both SPs when you create the pool LUNs,** which it does by default. If you must manually change the setting, the option to choose SP ownership is under the **Advanced** tab in the **Create LUN** dialog box in the Unisphere software. If you manually adjust the setting, you must also manually ensure balance between the SPs.

If you must change the SP ownership after the LUN is created, LUN migration can be leveraged to migrate the LUN to a new LUN with the desired SP owner. Then you must perform a trespass operation for the LUN from its previous owner to the new owner.
3. You should consider the metadata overhead before provisioning pool thin LUNs for your applications. You can estimate the metadata overhead by adding the fixed overhead to the variable overhead. As noted above, the fixed overhead is 3 GB for thin LUNs. The variable overhead is equal to 3% of the LUN's user capacity. (This is a conservative approximation for thin LUNs 250 GB and larger, while smaller LUNs may have slightly higher variable overhead.) Please note that EMC personnel and authorized partners can provide exact capacity utilization based on your specific configuration.

This formula approximates the storage consumed from the pool if you write to the entire LUN. For example, if you have a 250 GB thin LUN that gets fully consumed over a period of time, based on the above formula, it uses approximately 260 GB from the pool.

**Using virtual provisioning for file**

When using Virtual Provisioning for File, EMC recommends that you create file systems using thick LUNs as the underlying storage. Although you can create file systems using thin LUNs as the underlying storage, EMC does not recommend this because thin optimization is achieved using the thin attribute at the file system level.

**Creating a file system on pool LUNs**

To create file systems on pool LUNs:

1. Provision LUNs (thick LUNs recommended) from a pool.
2. Assign the LUNs to the protected File Storage Group.
3. Initiate a rescan of the storage systems (under the System tab in Unisphere). This will:
   a. Start a diskmark that makes the LUNs available to VNX file components for use as file storage space.
   b. Create a pool for file that is the same name as the corresponding pool for block.
   c. Create a disk volume in a 1:1 mapping for each LUN that was added to the File Storage Group.
4. Create a file system for file on the disk volumes. The pool LUNs presented to file will operate as they do for any other pool LUN in the system.

Figure 11 shows how file systems are built from LUNs in a storage pool.
Figure 11. Virtual Provisioning for VNX File OE for File 7.0

Best Practices when creating a file system on pool LUNs are:

- Allocate the entire pool to file systems.
- The entire pool should use thick LUNs only.
- Recommendations for thick LUNs:
  - 1 Thick LUN per physical disk in a pool.
  - Pool LUN count should be divisible by five to facilitate striping.
  - Balance SP ownership.
- All LUNs in the Pool should have the same tiering policy. This is needed to support slice volumes.
- Prior to VNX OE for File V7.0.3x, the use of AVM was not optimal due to the fact that AVM concatenated volumes as opposed to striping and slicing. Wherever possible, upgrade to V7.0.3x prior to implementation. If this is not possible, Manual Volume Management can provide reasonable performance configurations by creating stripe volumes manually and building user defined pools.
- We recommend that you not use the Block-side thin provisioning or compression on VNX LUNs used for file system allocations. Instead, you should use the File-side thin provisioning and File-side deduplication (which includes compression).
• VNX file configurations, for both VNX and VNX gateways with Symmetrix, will not expressly forbid mixing LUNs with different data service attributes. Users are warned that it is not recommended to mix due to the impact of spreading a file system across.

Note: VNX for File configurations will not allow mixing of mirrored and non-mirrored types in a pool. If you attempt to do this, the disk mark will fail.

• Where the highest levels of performance are required, use RAID Group LUNs for File.

• When additional capacity is required, present additional pool LUNs to VNX for File. Pool LUN expansion is not yet supported by VNX for File.

Creating file systems using pool-based LUNs is supported in Unisphere and the CLI. All relevant Unisphere configuration wizards support a pool configuration, except for the **VNX Provisioning Wizard for File**. Pool properties are found in the properties pages of pools for file (see Figure 12), and property pages for volumes and file systems (see Figure 13). You can only modify pool properties in the block pool or LUN areas of Unisphere. On the File System Properties pages in the Advanced Data Services section, FAST VP tiering policy, thin, compression, or mirrored is listed if it is enabled on the Block side.

![Figure 12: File System Properties window for VNX OE for File 7.0](image)
Figure 13. File System Properties window for VNX OE for File 7.0

The Volume Manager allows you to create volumes from file storage pools. The file storage pool is a collection of the disk devices presented to the file side from the block side of the storage system, as shown in Figure 14.

![Diagram of file systems and block levels]

**Figure 14. Provisioning at the volume management level**
Files created on pool LUNs are configured differently from files created on RAID groups. In VNX, AVM adheres to the following when creating files on underlying pool LUNs:

- Given a set of available LUNs from a mapped pool, AVM first divides the set into thick and thin LUNs and then attempts to find the space among the thick LUNs. (Please note that best practices suggest to NOT use thin LUNs, although AVM does not reject their use.)
- AVM tries to stripe 5 LUNs together, with the same size, same data services, and in an SP-balanced manner. If 5 LUNs cannot be found, AVM tries 4, 3, and then 2.
- AVM prefers selecting SP-balanced LUNs over homogeneous data services.
- If thick LUNs cannot be found to satisfy the above, the same search is implemented for thin LUNs.
- If AVM cannot find like-sized LUNs to stripe, it then concatenates LUNs to meet the size requirement.
- If AVM cannot find thick LUNs to concatenate, it tries to find thin LUNs to concatenate.
- If AVM cannot find thin LUNs to concatenate, it then tries to concatenate thick and thin together to meet the size requirement. If this still fails, the file system creation or extension fails.

*Managing Volumes and File Systems with VNX AVM* product documentation provides further information on using AVM with mapped pools.

**Thin file systems**

Thin file systems can be created on pool-based LUNs and RAID Group based LUNs. In VNX, you use AVM to create a thin file system the same way that you create a thick file system. Thin and thick file systems have different disk usage characteristics. Like a thick file system, a thin file system is configured with an initial size that is fully allocated. However, the thin file system has defined attributes that determine when it should be extended and, optionally, what its maximum size should be. The maximum file system size is what the end user sees as the available capacity in the file system.
Figure 15. No thin on file systems

In Figure 15 each user home directory area is 100 GB in size; this results in a large amount of allocated data that is unused.

Figure 16. Thin file systems

In Figure 16, each user home directory size is presented as 100 GB to the client; however the space that is actually allocated is much smaller. More space is automatically allocated as required.

Figure 17 shows the File System Creation tab in the New File System window. Use this window to create thin file systems. This is where you enter the Storage Capacity value as you would for any file system. You select Thin Enabled, which automatically...
enables auto-extension. You must enter the high water mark for auto-extension and the Maximum Capacity, which specifies what the file system capacity is visible to the client in the field. Clicking **OK** creates a file system with the initial capacity of 200 MB, which is the actual space allocated to the file system at this time.

![Figure 17. Unisphere New File System screen](image)

After creating the file system shown in Figure 17, 10 TB would be visible to the user, although VNX actually only allocated 200 MB of storage. This is illustrated in Figure 18.
Implementing file-based thin has many benefits, including the fact that it prevents overprovisioning, which improves your TCO. However, there are things you need to consider before incorporating file-based thin into your environment. For example, you need to consider the possible impact on VNX SnapSure™, VNX Replicator, deduplication, and the suitability of certain application types.

When using thin file systems, consider the following:

- Enabling thin with automatic file system extension does not automatically reserve the space from the storage pool for that file system.

  Administrators must ensure that adequate storage space exists so that the automatic extension operation can succeed. If the available storage is less than the amount of storage required to extend the file system, automatic extension fails. In this case, the user receives an error message when the file system runs out of space. Note that the user may get this message even though it “appears” that there is free space in the file system.

- VNX Replicator is an asynchronous remote replication mechanism. It provides a read-only point-in-time copy of a source file system. It then periodically updates this copy to make it consistent with the source object. With a thin file system, only data that is allocated on the source object is copied to the target object.

  Therefore, like VNX SnapSure, the destination file system is also thin.
- Enable automatic file system extension and thin only on the source file system in a replication scenario. The destination file system synchronizes with the extension of the source file system and extends automatically.

- In most cases, AVM is suitable for your environment. When performance is your top priority and AVM does not provide what is required, then you must manually create and manage VNX volumes and file systems. Automatic file system extension is not supported when manually managing VNX volumes. This means that thin cannot be used, which is something to consider before starting to manually provision VNX volumes in your environment.

**Monitoring File Systems**

There are several methods to proactively monitor the utilization of thin file systems and the storage pools on which they were created. There are also *trending* and *prediction* graphs for the utilization of thin file system and storage pools.

Figure 19 shows the information that is provided on the utilization of a thin file system.

![Figure 19. Using Unisphere to determine the utilization of a thin file system](image)

Use Unisphere to configure proactive alerts when a thin file system or file storage pool is close to being oversubscribed. You can customize these alert notifications according to the file system and storage pool utilization; predicted time-to-fill; and overprovisioning. The alert notifications include logging the event log file, sending an email, or generating a Simple Network Management Protocol (SNMP) trap.
You can configure two types of storage used notifications:

- Current size – how much of the currently allocated file system/storage pool capacity is used
- Maximum size – how much of the configured maximum file system/storage pool capacity is used (when the file system/storage pool will be fully extended)

When using thin file systems, it is essential to track the utilization of the storage with these monitoring capabilities. That allows you to provision more storage when you need it, and avoid shortages that could impact users and applications.

*Configuring Events and Notifications on VNX for File* provides further information on setting event notifications in VNX.

**Performance considerations**

In general, some applications, I/O workloads, and storage deployment scenarios see performance improvements from using thin file systems. However, it is important to note that these improvements may change over time as the thin file system expands and as the data is used, deleted, or modified.

In a thin file system (as opposed to a file system on a thin LUN), performance improves mostly with random and mixed read I/O. Because the thin file system initially occupies less space on disk than a fully provisioned file system occupies, there are smaller disk seeks required for random reads. Disk seeks impact I/O latency, so minimizing seeks can improve performance. With sequential read I/O, disk seeks are already infrequent, and therefore a performance improvement would not be noticed. Write I/O will also not see much performance improvement as disk seeks are usually not necessary, or only minimal (except for random overwriting), and, for a large part, will be cached anyway. It should be emphasized that this performance improvement may decrease over time as the file system is further used and extended and fragmented, thus increasing the size of disk seeks and the corresponding latency.

**When to use traditional, thick, and thin LUNs**

It is important to understand your application requirements and select the approach that meets your needs. If conditions change, you can use VNX LUN migration to migrate among thin, thick, and traditional LUNs.

Use RAID groups and traditional LUNs:

- For applications that require extreme performance (for example, when milliseconds of performance are critical).
- For the most predictable performance.
- For precise data placement on physical drives and logical data objects.
- For physical separation of data.
Use thick LUNs:
• For applications that require good performance.
• For taking advantage of data efficiency services like FAST VP.
• When using storage pools with VNX for File.
• For easy setup and management.

Use thin LUNs:
• For applications with moderate performance requirements.
• For taking advantage of data efficiency services like FAST VP and compression.
• For easy setup and management.
• For best space efficiency.
• For energy and capital savings.
• For applications where space consumption is difficult to forecast.

**Thin LUN space reclamation via Migration**

Space reclamation allows the system to regain allocated storage that is not used. This feature works with LUN migration and SAN Copy™ when migrating existing volumes to thin LUNs. Unused storage is often locked by applications that are either no longer important to organizations or do not need the storage that was originally allocated to them. With the space reclamation capability, you can non-disruptively migrate existing volumes of your applications to thin LUNs and reclaim the unused space. Since thin LUNs only consume storage to which data is written, all the allocated but unused storage is returned to the pool so that it can be used by other LUNs in the pool. The process is completely transparent and allows you to move your applications without requiring any downtime.

Space reclamation occurs automatically when performing a SAN Copy pull operation on source volumes from older CLARiiON® systems, Symmetrix® storage systems, and supported third-party systems to a thin destination LUN on VNX. Space reclamation also occurs when you perform a LUN migration to move an existing traditional LUN or thick LUN to a thin LUN within the same array. The software detects zeros at 8 KB chunk granularity. For example, it will only migrate 8 KB chunks with data in them. If the 8 KB chunk is filled with zeros, it is not migrated to a thin LUN.

With the sdelete utility for NTFS you get this benefit with LUN migration as well as compression. The sdelete utility offers an easy way to condition the file system prior to the array operation. Using the -c option replaces deleted files with zeros thereby allowing the space reclamation processes to work effectively. Run and complete the utility prior to performing the array-based migration or compression operation.

“Deleted” files in NTFS aren't removed from disk but are overwritten when the space is required. Since the migration software processes data at the bit level, it has no
Therefore, the data of deleted files is processed the same way as the data of active, non-deleted files. This can significantly reduce the benefits received from space reclamation.

Data protection

VNX Virtual Provisioning supports local and remote replication. When replicating a thin LUN, the *host visible* capacity, and not the *consumed* capacity, is used to determine which LUNs are eligible for replication. For example, assume you plan to replicate a thin LUN to another thin LUN. The thin LUN has a *host visible capacity* of 100 GB and *consumed capacity* of 20 GB. The destination thin LUN has to be at least 100 GB to be eligible for the replication operation. Thin-to-thin replication provides optimal space efficiency for your replicas because only the used or consumed space is replicated. To provide even more flexibility, the VNX also supports replication between thin and thick LUNs, and traditional LUNs.

For file replication, EMC recommends using VNX Replicator or EMC RecoverPoint. VNX Replicator provides granularity to the file system level, while RecoverPoint provides single consistency group replication for cabinet-level failover.

RecoverPoint

RecoverPoint Continuous Data Protection (CDP) and Continuous Remote Replication (CRR) support replication for thin LUNs, thick LUNs, and traditional LUNs. When using RecoverPoint to replicate to a thin LUN, only data is copied; unused space is ignored so the target LUN is thin after the replication. This can provide significant space savings when replicating from a non-thin volume to a thin volume. When using
RecoverPoint, we recommend that you not use journal and repository volumes on thin LUNs.

In VNX OE for File 7, when RecoverPoint is used for file, a single-consistency group replication and failover is available. For replication at the individual file-system level, you can use Replicator. Only CRR is supported in VNX OE for File 7. For more information on using RecoverPoint, see the Using RecoverPoint/SE with VNX for File for Disaster Recovery white paper on EMC Online Support.

**VNX Snapshots**

VNX Snapshots is a new VNX software feature that creates point-in-time data copies. Unlike SnapView snapshots and clones, VNX snapshots do not require a special Reserved LUN Pool. Instead, VNX Snaps share capacity in the pool, which users can easily manage with the controls provided with VNX Snaps.

VNX Snapshots use *redirect on write* (ROW) technology. ROW allows redirecting new writes destined for the primary LUN to a new location in the storage pool. Such implementation is different from *copy on first write* (COFW) used in SnapView feature where the writes to the Primary LUN were held until the original data was copied to the Reserved LUN pool to preserve a snapshot.

VNX Snapshots and SnapView can coexist with each other. Furthermore, SnapView clones, which use a different technology from SnapView snapshots, will work with VNX Snapshots. VNX Snapshots can be used for data backups, software development and testing, repurposing, data validation, and local rapid restores.

**SnapView**

For local replication, SnapView™ snapshots and clones are supported on thin and thick LUNs. SnapView clones support replication between thick, thin, and traditional LUNs. When cloning from a thin LUN to a traditional LUN or thick LUN, the physical space of the traditional/thick LUN must equal the host visible capacity of the thin LUN. This results in a fully allocated thin LUN if the traditional LUN/thick LUN is failover. Cloning from a traditional or thick LUN to a thin LUN results in a fully allocated thin LUN. The initial synchronization forces the initialization of all the subscribed capacity.

**SnapSure**

With SnapSure, you can create and manage checkpoints on thin and thick file systems. Checkpoints are point-in-time, logical images of a file system. You can create checkpoints on file systems that use pool LUNs or traditional LUNs.

**MirrorView**

When mirroring a thin LUN to another thin LUN, only consumed capacity is replicated between the storage systems. This is the most beneficial for initial synchronizations. Steady state replication is similar, since only new writes are written from the primary storage system to the secondary system.
When mirroring from a thin LUN to a traditional or thick LUN, the thin LUN’s host visible capacity must be equal to the traditional LUN’s capacity or the thick LUN’s user capacity. Any failback scenario that requires a full synchronization from the secondary to the thin primary image causes the thin LUN to become fully allocated. When mirroring from a thick LUN or traditional LUN to a thin LUN, the secondary thin LUN is fully allocated.

With MirrorView, if the secondary image LUN is added with the no initial sync option, the secondary image retains its thin attributes. However, any subsequent full synchronization from the traditional LUN or thick LUN to the thin LUN, as a result of a recovery operation, causes the thin LUN to become fully allocated. For more information on using pool LUNs with MirrorView, see the MirrorView Knowledgebook white paper on EMC Online Support.

**VNX Replicator**

VNX Replicator provides replication and failover for individual file systems. Replicator is used to replicate thin and thick file systems. Replicator can be used with file systems that use pool LUNs or traditional LUNs. With a thin file system, only data that is allocated on the source object is copied to the target object. Therefore the destination file system is also thin.

When replicating, you must enable automatic file system extension and thin only on the source file system. The destination file system synchronizes with the extension of the source file system and extends automatically.

**PowerPath Migration Enabler**

EMC PowerPath® Migration Enabler (PPME) is a host-based migration tool that enables non-disruptive or minimally disruptive data migration between storage systems or between logical units within a single storage system. The Host Copy technology in PPME works with the host operating system to migrate data from the source logical unit to the target. With PPME 5.3, the Host Copy technology supports migrating virtually provisioned devices. When migrating to a thin target, the target’s thin-device capability is maintained. This can be very useful when migrating data from older systems with non-thin volumes to new thinly provisioned storage systems. Host Copy migrations for thin LUNs are supported on Windows, Linux, AIX, and Solaris hosts connected to VNX.

**Using thin LUNs with applications**

Due to the storage-on-demand feature of thin LUNs, not all application environments are well suited to thin LUNs. In general, it is a best practice to use thin-friendly applications that do not preallocate all of the storage space during initialization. Thin-friendly applications also reuse deleted space before consuming additional storage. The following are guidelines for using thin LUNs with applications most commonly used with VNX.
Host-based File Systems

When creating a file system on a thin LUN, you need to consider how much metadata is written to the thin LUN. An inefficient file system, which writes a lot of metadata to the LUN, causes thin devices to become fully allocated more quickly than with an efficient file system. For this reason, an efficient file system is thin-friendly.

Another thin-friendly feature is the ability to effectively reuse deleted space. If you delete a file, the file system knows it can use those blocks to store new data. The underlying VNX thin LUN, however, is not aware of this. When you create a new file on the host file system, depending on the file system, it may or may not use the space freed up by a deleted file. If it writes the new file in the previously reclaimed area, then the thin LUN does not consume more space from the pool. However, if it writes in a previously unwritten area, then the thin LUN consumes more space from the pool.

With NTFS, some operations are thin-friendly. For example, an NTFS format does not preallocate the physical space. Instead, it creates file system metadata that only consumes a few gigabytes of metadata on the thin LUN. NTFS writes new data on the LUN and updates its own metadata accordingly. However, NTFS is not very effective when it comes to reusing deleted space.

Other file systems like Linux ext2 and ext3, Solaris ZFS and UFS, and Symantec VxFS do not preallocate all the storage and also effectively reuse deleted space before allocating new space and thus work nicely with thin LUNs. For the latest list of file systems supported with VNX Virtual Provisioning, check the EMC Support Matrix at https://elabnavigator.emc.com/ on EMC Online Support.

Virtual environments

For VMware environments, the Virtual Machine File System (VMFS) has many characteristics that are thin-friendly. First, a minimal number of thin extents are allocated from the pool when a VMware file system is created on thin LUNs. Also, a VMFS datastore reuses previously allocated blocks that are beneficial to thin LUNs. When using RDM volumes, the file system or device created on the guest OS dictates whether the RDM volume is thin-friendly.

With ESX® 3.x, when creating a VMware virtual disk, the default type is “zeroedthick.” This is the recommended virtual disk type for thin LUNs. Using zeroedthick, the storage required for the virtual disk is reserved in the datastore, but the VMware kernel does not initialize all the blocks. In addition, features such as VMware DRS, Converter, and VMotion® are thin-friendly. VM Clones, Storage VMotion, Cold Migration, and Templates are not thin-friendly. VM Cloning fully allocates all blocks. There is currently no workaround for this. VMware Templates also allocate all blocks. The workaround is to shrink VMDKs before creating a template and use the “Compact” option. Customers are advised to use these features on thick LUNs. For detailed information about implementing Virtual Provisioning with VMware, refer to the TechBook Using EMC VNX Storage with VMware vSphere available on EMC Online Support.
With vSphere 4.0, VNX thin LUNs can be configured as thick (zeroedthick) or thin. With either option, the VMware administrator must monitor the consumed capacity on the VMFS datastore. VMware vSphere™ provides an alert when the thresholds are reached. Also, with ESX 4.0, features such as VM Clones, Storage VMotion, Cold Migration, and Templates become thin-friendly. This is because, unlike ESX 3.x, the zeroedthick or thin format remains intact on the destination datastore. This means that the consumed capacity of the source virtual disk is preserved on the destination virtual disk and is not fully allocated.

vSphere 5 adds the ability to perform thin LUN space reclamation at the storage system level. VMFS-5 uses the SCSI UNMAP command to return space to the storage pool when created on thin LUNs. SCSI UNMAP is used any time VMFS-5 deletes a file, such as Storage vMotion, Delete VM, Delete Snapshot, etc. Earlier versions of VMFS would only return the capacity at the file system level. An array specific feature, like LUN Migration, would have to be used to physically reclaim the space. vSphere 5 greatly simplifies the process by conducting space reclaim automatically.

The VMware kernel provides other mechanisms for creating virtual disks that are not thin-friendly. The “eagerzeroedthick” format is not recommended for thin LUNs, because at creation it performs a write to every block of the virtual disk and results in equivalent storage use in the thin pool.

In Hyper-V environments, EMC recommends that you select the “dynamically expanding” option for virtual hard disk on the NTFS file system with VNX thin LUNs; this preserves disk resources. When using pass-through, the file system or guest OS dictates whether the pass-through device will be thin-friendly. For more information on using Hyper-V Server, see the Using EMC CLARiiON with Microsoft Hyper-V Server white paper on EMC Online Support.

**Veritas Storage Foundation**

Veritas Storage Foundation, by Symantec, enhances the value of thin-provisioned storage environments by optimizing migrations and enabling storage reclamation. VNX thin LUNs can take advantage of the SmartMove and Thin Reclamation capabilities in Storage Foundation to help customers capitalize on their existing storage investment and minimize costs.

**SmartMove**

Veritas SmartMove enables efficient online migrations from thick storage to thin storage, and in the process, reclams unused space. It supports UNIX and Linux platforms with Storage Foundation 5.0 MP3, and Windows platforms with Storage Foundations 5.1. Tight integration between Veritas Volume Manager and Veritas File System allows SmartMove to identify the blocks deleted by an application. SmartMove uses this information to ensure that only the necessary blocks of data are moved during online migration.
Thin Reclamation API

Thin Reclamation API enables online reclamation of unused storage over time, so that thin-provisioned storage can remain thin. When data is deleted from a host file system, the physical storage stays allocated to a thin LUN. To reclaim this unused but allocated storage, Thin Reclamation API sends the SCSI WRITE_SAME command (with the UNMAP bit turned on) to inform the storage array that all physical storage for a certain block range can be reclaimed. When the storage system receives SCSI WRITE_SAME commands from the Thin Reclamation API, it zeros out the specified block range in the thin LUN. Over time, a background process returns the zeroed space back to the pool where they can be used by other applications.
Conclusion

When implemented appropriately, Virtual Provisioning can be a powerful complement to an organization’s processes and technologies for improving ease of use and utilizing storage capacity more efficiently. VNX Virtual Provisioning integrates well with existing management and business continuity technologies, and is an important advancement in capabilities for VNX customers.

VNX Virtual Provisioning:

- Saves time:
  - Easy to create pools and pool LUNs
  - Easy to expand pools and pool LUNs
  - Easy to monitor and manage
- Reduces provisioning uncertainty:
  - Decisions are easy to modify
  - No impact on host servers
- Reduces upfront investment and saves energy:
  - Highly space-efficient
  - Multiple applications share resources
  - Physical storage can be added as required
- Builds on existing VNX features:
  - Migration is supported among all types of LUNs
  - VNX Snaps, SnapView snapshots and clones, SnapSure checkpoints, Replicator, MirrorView/S, MirrorView/A, SAN Copy
  - Unisphere Quality of Service Manager and Unisphere Analyzer
  - Unisphere Reports

References

- EMC VNX FAST VP — A Detailed Review
- EMC Virtual Provisioning Release Notes
- EMC VNX Deduplication and Compression
- EMC Unified Storage Best Practices for Performance and Availability — Common Platform and Block Storage — Applied Best Practices
- MirrorView Knowledgebook:Releases 30- 32
- EMC CLARiiON SAN Copy — A Detailed Review
• EMC RecoverPoint/SE for the CLARiiON CX4 — Applied Technology
• EMC CLARiiON Integration with VMware ESX — Applied Technology
• Managing Volumes and File Systems with VNX AVM
Appendix A: Virtual Provisioning limits

Table 4. Limits with VNX Operating Environment for Block 32

<table>
<thead>
<tr>
<th>System</th>
<th>Max pools</th>
<th>Max disks per pool</th>
<th>Total disks in all pools</th>
<th>Max thin LUNs per pool</th>
<th>Max thin LUNs in all pools</th>
</tr>
</thead>
<tbody>
<tr>
<td>VNX5100</td>
<td>10</td>
<td>71</td>
<td>71</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>VNX5300</td>
<td>20</td>
<td>121</td>
<td>121</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>VNX5500</td>
<td>40</td>
<td>246</td>
<td>246</td>
<td>1024</td>
<td>1024</td>
</tr>
<tr>
<td>VNX5700</td>
<td>40</td>
<td>496</td>
<td>496</td>
<td>2048</td>
<td>2048</td>
</tr>
<tr>
<td>VNX7500</td>
<td>60</td>
<td>996</td>
<td>996</td>
<td>2048</td>
<td>2048</td>
</tr>
</tbody>
</table>
Appendix B: Guidelines for pools and pool LUNs

Guidelines for pools
- Expand pools by approximately the same number of disks as originally present in the pool.
- Avoid running pools at near maximum capacity.
- Ensure that a monitoring strategy is in place for all pools.
- Fault domain considerations: Unless RAID 6 is the pool's level of protection, avoid creating pools with very large number of RAID groups. Keep in mind that smaller pools have its benefits.

Guidelines for pool LUNs
Operating systems and applications are thin LUN-friendly when:
- The file systems efficiently reuse deleted space.
- Applications reserve space but do not initialize all storage.
- Consider metadata overhead and performance impact before provisioning thin LUNs.
- Balance your LUNs on both SPs and do not change the default owner once provisioned.

---

1 For continuous updates to best practices concerning Virtual Provisioning please use the VNX white paper *EMC Unified Storage Best Practices for Performance and Availability — Command and Block — Applied Best Practices*