

Survey Of Volume Managers

Nasser M. Abbasi

May 24, 2000

Contents

1	Advantages of Volume Managers	1
2	Terminology used in LVM software	1
3	Survey of Volume Managers	3
3.1	SGI XLV. First generation volume manager	3
3.2	SGI XVM Volume Manager. Second generation, enhanced XLV	3
3.3	Linux LVM (Logical Volume Manager)	13
3.4	Veritas Volume Manager. (Sometimes called VxVM)	13
3.5	The Vinum Volume Manager	15
3.6	Compaq SANworks Enterprise Volume manager	16
3.7	Sun StorEdge Volume Manager (SSVM). (repackaged version of Veritas Volume Manager.)	16
3.8	AIX Logical Volume Manager	17
3.9	Sequent Volume Manager (ptx/SVM)	17
3.10	HP Shared Logical Volume Manager (SLVM)	19
4	Other links to Volume Managers	19

1 Advantages of Volume Managers

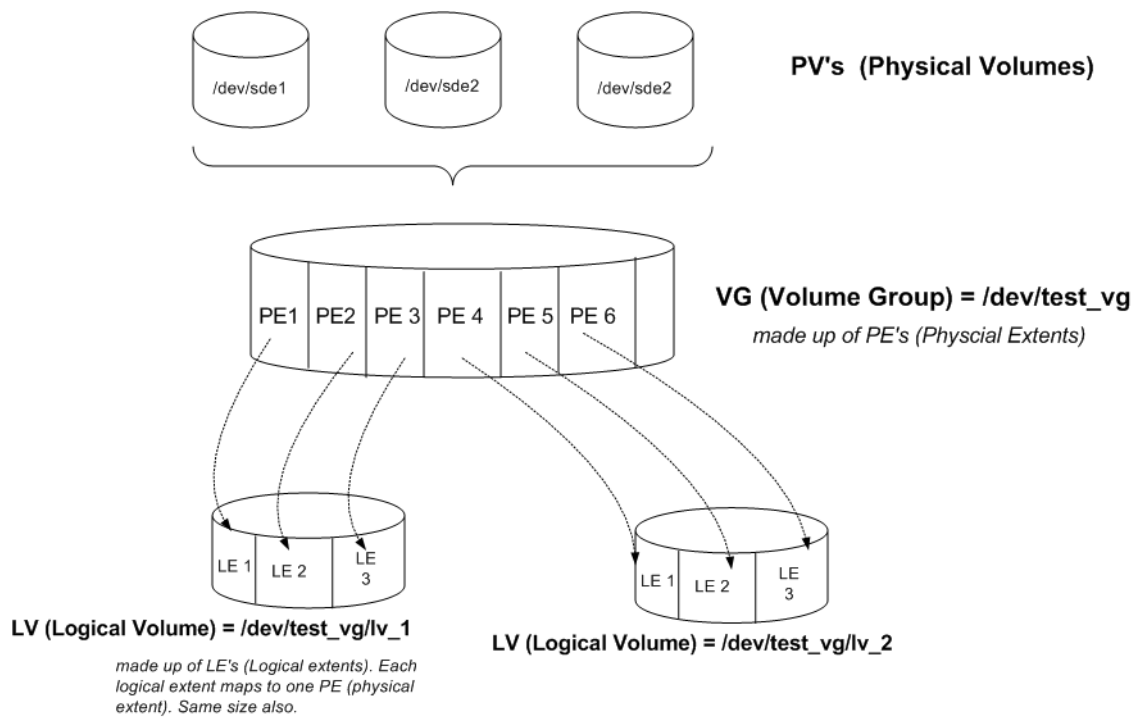
From <http://www.uwsg.iu.edu/usail/peripherals/disks/logical/>

Some of the advantages of using a logical volume manager:

1. They offer greater flexibility for disk partitioning.
2. The size of logical volumes can be modified according to need, while the operating system is running
3. Logical volumes can span multiple disks.
4. Disk mirroring is often supported, for greater data reliability

2 Terminology used in LVM software

Figure 2 on the following page Shows the main terms used in Logical Volume Manager software.



Overview of terminology
used in Logical Volume
Manager software.

Nasser Abbasi 5/24/00.
vlm_overview.vsd

Figure 1: Terms used in LVM software.

3 Survey of Volume Managers

This a list of volume managers that I collected searching the web:

3.1 SGI XLV. First generation volume manager

XLV is SGI volume manager that is integrated in XFS (SGI file system).
see <http://www.sgi.com/Technology/xfs-whitepaper.html>

From above URL:

3.5.1 Basics

The xlv volume manager (XLV) is an integral part of the XFS filesystem(1). The volume manager provides an operational interface to the system's disks and isolates the higher layers of the filesystem and applications from the details of the hardware. Essentially, higher-level software "sees" the logical volumes created by XLV exactly like disks.

Yet, a logical volume is a faster, more reliable "disk" made from many physical disks providing important features such as the following (discussed in detail later):

- concatenating volumes for a larger disk
- striping volumes for a larger disk with more bandwidth
- plexing (mirroring) volumes for a more reliable disk

The use of volumes enables XFS to create filesystems or raw devices that span more than one disk partition. These volumes behave like regular disk partitions and appear as block and character devices in the /dev directory. Filesystems, databases, and other applications access the volumes rather than the partitions. Each volume can be used as a single filesystem or as a raw partition. A logical volume might include partitions from several physical disks and, thus, be larger than any of the physical disks.

Filesystems built on these volumes can be created, mounted, and used in the normal way.

The volume manager stores all configuration data in the disk's labels. These labels are stored on each disk and will be replicated so that a logical volume can be assembled even if some pieces are missing. There is a negligible performance penalty for using XLV when compared to accessing the disk directly; although plexing (mirroring data) will mildly degrade write performance.

3.2 SGI XVM Volume Manager. Second generation, enhanced XLV

URL <http://www.sgi.com/developers/devnews/2000/xvm.html>

From above URL

Working with the XVM Volume Manager

By Steven Levine, Senior Technical Writer

The XVM Volume Manager is the next-generation SGI logical volume improvements over the XLV logical volume manager. It is avail

environment in the IRIX 6.5.7 release and is planned to be available in a future IRIX release.

In addition to the features that XLV logical volumes support, the following capabilities are available:

Support for a cluster environment

In conjunction with CXFS filesystem, the XVM Volume Manager provides a cluster environment, providing an image of the XVM devices across the cluster. Administration of XVM devices from any cell in the cluster is supported. Devices are assigned dynamically to the entire cluster or to individual nodes.

Flexible volume layering and configuration

The elements that make up an XVM logical volume can be layered. For example, using the XVM Volume Manager, an administrator can create a logical volume configuration or use stripe-on-stripe layering to create a logical volume in situations where this results in better volume performance.

Mirroring of root and swap partitions

The XVM Volume Manager allows you to specify an XVM disk as a mirror for root and swap partitions.

Large number of slices

The layout of a disk under XVM is independent of the underlying hardware. The XVM Volume Manager determines how the disk is sliced. Because of this, you can divide a disk into an arbitrary number of slices.

Large number of volumes

The XVM Volume Manager supports thousands of volumes on a single disk. The expansion of the label file is as needed. Under XVM, there are no limits on the number of volumes.

Improved mirror performance

The XVM Volume Manager allows you to specify the read policy allowing you to read from the mirror in a sequential or round-robin fashion based on the needs of your configuration. You can also specify whether round-robin is preferred for reading.

The XVM Volume Manager also allows you to specify when a mirror should be synchronized at creation and to specify that a particular filesystem, will never need to be synchronized.

This article provides information on the following topics:

- General administrative differences between XLV and XVM
- XVM layered logical volume configuration example
- Using XVM in a clustered environment
- Mirroring root and swap partitions (support planned for IR)

The procedures described in this article are meant to provide information on the XVM Volume Manager and how to administer XVM logical volumes with it. More information about the XVM Volume Manager and its operation is provided in the XVM Administrator's Guide, available on the SGI Technical Publications website at techpubs.sgi.com/library/.

XLV and XVM

There are some basic conceptual differences between the XVM layered logical volume manager. The XVM logical volume manager requires a physical disk, which is then called an XVM physvol. This allows XVM to be limited in the number of partitions that you can create on the disk's partitions, but instead you create slices on the disk's building blocks.

The following procedures show how you would create the same layered logical volume and as an XVM logical volume. The commands provided in the procedures create a volume named stripedvol in which the data is to be striped across the disk's building blocks.

assume that the disks have already been partitioned as IRIX or Solaris with a one-to-one correspondence in the numbered procedures in the text.

To create the XLV logical volume, perform the following procedure:

1. Create the logical volume:

```
xlvmake> vol stripedvol
```

2. Create the data subvolume:

```
xlvmake> data
```

3. Create the plex:

```
xlvmake> plex
```

4. Create the striped volume element to be striped across the disks:

```
xlvmake> ve -stripe dks0d2s7 dks0d3s7 dks5d4s7
```

5. Exit the XLV Volume Manager. Since the `xlvmake -A` option is used, the `xlvmake` command is automatically called to configure the volume into the running system.

```
xlvmake> end
```

```
xlvmake> exit
```

6. Execute the `mkfs` command on the filesystem:

```
# mkfs /dev/xlv/stripedvol
```

7. Mount the filesystem:

```
# mount /dev/xlv/stripedvol /mnt
```

To create the same volume as an XVM logical volume, perform the following procedure:

1.Label the disks as XVM physical volumes:

```
xvm:cluster> label -name disk0 dks2d70vol
xvm:cluster> label -name disk1 dks2d71vol
xvm:cluster> label -name disk2 dks2d72vol
```

2.Create XVM slices out of each disk:

```
xvm:cluster> slice -all disk*
```

3.Create the striped volume element and attach it to the vol
subvolume in the process):

```
xvm: cluster> stripe -volname stripedvol

slice/disk0s0 slice/disk1s0 \

slice/disk2s0
```

4.Exit the XVM Volume Manager:

```
xvm:cluster> quit
```

5.Execute the mkfs command on the filesystem:

```
# mkfs /dev/cxvm/stripedvol
```

6.Mount the filesystem. For a filesystem in a CXFS cluster,
CXFS GUI or CXFS CLI. On a single-cell system, you can use

Migrating from existing XLV logical volumes to XVM logical vo
You delete the XLV volumes and unmount the XLV filesystems, g
then create slices on the XVM disks to match the partitions u
XVM volumes to match the XLV volumes and mount the filesystem

detail in the XVM Volume Manager Administrator's Guide.

XVM Configuration Example: Striping Mirrors

The following XVM configuration example should provide a feel for using the XVM Volume Manager to create a layered logical volume. This example creates a logical volume with striping and mirroring. In this example the logical volume contains a striped slice with mirroring a slice that contains all of the usable blocks of a

1. Assign four disks to XVM to manage:

```
xvm:cluster> label -name disk0 dks2d70
disk0
xvm:cluster> label -name disk1 dks2d71
disk1
xvm:cluster> label -name disk2 dks2d72
disk2
xvm:cluster> label -name disk3 dks2d73i
disk3
```

2. Create a slice out of all of the usable blocks on each XVM disk:

```
xvm:cluster> slice -all disk*
</dev/cxvm/disk0s0> slice/disk0s0
</dev/cxvm/disk1s0> slice/disk1s0
</dev/cxvm/disk2s0> slice/disk2s0
</dev/cxvm/disk3s0> slice/disk3s0
```

3. Create two mirrors, each consisting of two of the slices you just created. The `-clean` option on the `mirror` command creates new mirrors that will be written to before they are read, and the `-no-sync` option indicates that the mirrors do not need to be synchronized.

If you do not specify the `-clean` option, executing this command synchronizes the data on the slices. A message indicating the synchronization is written to the SYSLOG, and another message would be written when the command completes.

You will not need to define a persistent name for the volume.

```
xvm:cluster> mirror -tempname -clean slice/disk0s0 slice/disk1s0
</dev/cxvm/vol2> mirror/mirror1
xvm:cluster> mirror -tempname -clean slice/disk2s0 slice/disk3s0
</dev/cxvm/vol3> mirror/mirror2
```

4. Create a stripe that consists of the two mirrors you have generated to contain the stripe. This command attaches the stripe to the volume.

```
xvm:cluster> stripe -volname mirvol mirror/mirror1 mirror/mirror2
</dev/cxvm/mirvol> stripe/stripes2
```

5. Display the XVM logical volume.

```
xvm:cluster> show -top mirvol
vol/mirvol          0 online
  subvol/mirvol/data 71085312 online
    stripe/stripes2 71085312 online,tempname
      mirror/mirror1 35542780 online,tempname
        slice/disk0s0 35542780 online
        slice/disk1s0 35542780 online
      mirror/mirror2 35542780 online,tempname
        slice/disk2s0 35542780 online
        slice/disk3s0 35542780 online
```

6. Exit the xvm tool by typing exit (or quit or bye). You can then format the volume.

```
xvm:cluster> exit
3# mkfs /dev/cxvm/mirvol
meta-data=/dev/cxvm/mirvol      isize=256      agcount=17, a
data      =                      bsize=4096    blocks=444435
          =                      sunit=16       swidth=32 blk
```

```
naming    =version 1                bsize=4096
log       =internal log            bsize=4096  blocks=1168
realtime  =none                    extsz=65536 blocks=0, rte
```

7. Mount the filesystem. For a filesystem in a CXFS cluster, GUI or the CXFS CLI.

XVM in a Clustered Environment

An XVM physical volume can be controlled by a single node or as defined by the CXFS Cluster Manager. If an XVM physical volume has a local domain and the controlling node is the volume's owner, the volume is controlled by any node in a CXFS cluster, the volume has a cluster owner, and the cluster is the volume's owner. When a volume's owner is a cluster, an XVM physical volume is controlled by the cluster. When a volume's owner is a node, the node is the volume's owner. When a volume's owner is a cluster, an XVM physical volume is controlled by the cluster.

Only the owner for an XVM physical volume can modify the configuration. Other nodes may be able to see the disk itself, but the XVM Volume Manager treats the disk as a foreign disk, even if the disk has been labeled as an XVM physical volume. The XVM configuration.

When you bring up the XVM Volume Manager with the `xvm` command enabled, the `xvm:cluster>` prompt appears by default, indicating that all XVM physical volumes you create in this XVM session are in the cluster domain. When the `xvm:local>` prompt appears, indicating that all XVM physical volumes in this session are in the local domain.

When you are running the XVM Volume Manager in the cluster domain, all XVM physical volumes that are also in the cluster domain, even if they are owned by a local node, are in the cluster domain. To see and modify local disks, use the `set domain` command, or you use the `local:` prefix when specifying a physical volume when you are running the XVM Volume Manager in the local domain. To see and modify cluster disks, use the `cluster:` prefix when specifying the physical volume.

XVM System Disks: Mirroring Root and Swap (IRIX 6.5.9)

Support for mirroring the root and swap partitions is planned. Once this feature is fully supported, you will be able to label a disk as a system disk. You will also have the option of labeling the disk's partitions on an existing system disk.

The XVM label command provides a `-type` option that you can use to specify the type of disk that you are creating. There are three types of XVM disks:

option

An XVM physvol of type `option` is the default XVM disk type. It includes partitions for the volume and volume header.

root

An XVM physvol of type `root` includes a root and a swap partition for the volume header and the whole volume. When you label a disk as a root disk, a logical volume for the root partition and a logical volume for the swap partition are created. The root volume is named `physvol_root` and the swap volume is named `physvol_swap`, where `physvol` is the name of the XVM root physical volume.

usrroot

An XVM physvol of type `usrroot` includes a root and a swap partition and a space on the disk that can be allocated for a `usr` filesystem. The `usr` filesystem can be any XVM logical volume configuration. You can use the `fstab` file to mount `/usr`.

When you label a disk as an XVM `usrroot` disk, a logical volume for the root partition and a logical volume for the swap partition are also created. The root volume is named `physvol_root` and the swap volume is named `physvol_swap`, where `physvol` is the name of the XVM `usrroot` physical volume.

When you create an XVM system disk, you must create the XVM disks. XVM system disks are local to the node which boots from them. When you label an XVM disk, the layout for partition 0 and partition 1 remain the same.

The following example creates a system disk with combined root and swap partitions.

The system disk physvol this example creates is named fred.

```
xvm:local> label -type root -name fred dks0d3
```

After you execute this command, a physvol named fred is created. The physvol, one for root and one for swap.

In addition to the system disk, two logical volumes are created: logical volume fred_root for the root filesystem and logical volume fred_swap for the swap.

To mirror a system disk, you label a disk or disks to use as a mirror for the system disk. You use the -mirror option of the label command to specify the disks you want to mirror. The new disks you label become mirrors of the system disk. Executing this command does not add new logical volumes to the existing root and swap logical volumes.

Alternately, you can create an additional XVM root or usrroot logical volume. You can then insert a mirror into an existing root or usrroot logical volume that were created on the new XVM physvol.

The following example creates a system disk that mirrors the system disk. The physvol named fred. No new volumes are created; instead, a mirror is created for the existing root and swap logical volume.

```
xvm:local> label -type root -name wilma -mirror fred
```

After you execute this command, a physvol named wilma is created. The slices that are created on physvol wilma are inserted into the existing root and swap logical volumes.

Summary

Starting with IRIX 6.5.7, the XVM Volume Manager is available in the IRIX environment. Future IRIX releases will support this product in the IRIX environment.

3.3 Linux LVM (Logical Volume Manager)

URL <http://linux.msede.com/lvm/> by Heinz Mauelshagen. Above URL contains good presentations and more documentation on LVM. This is based originally on the OSF LVM.

From above URL:

The Logical Volume Manager (LVM) is a subsystem for on-line disk storage management which has become a de-facto standard for storage management across UNIX implementations.

The Logical Volume Manager adds an additional layer between the physical devices and the block I/O interface in the kernel to allow a logical view on storage. Unlike current partition schemes where disks are divided into fixed-sized continuous partitions, LVM allows the user to consider disks, also known as physical volumes (PV), as a pool (or volume) of data storage, consisting of equal-sized extents.

A LVM system consists of arbitrary groups of physical volumes, organized into volume groups (VG). A volume group can consist of one or more physical volumes. There can be more than one volume group in the system. Once created, the volume group, and not the disk, is the basic unit of data storage (think of it as a virtual disk consisting of one or more physical disks).

The pool of disk space that is represented by a volume group can be apportioned into virtual partitions, called logical volumes (LV) of various sizes. A logical volume can span a number of physical volumes or represent only a portion of one physical volume.

The size of a logical volume is determined by its number of extents. Once created, logical volumes can be used like regular disk partitions - to create a file system or as a swap device.

LVM was initially developed by IBM and subsequently adopted by the OSF (now OpenGroup) for their OSF/1 operating system. The OSF version was then used as a base for the HP-UX and Digital UNIX operating system LVM implementations. Another LVM implementation is available from Veritas which works differently. The Linux implementation is similar to the HP-UX LVM implementation.

3.4 Veritas Volume Manager. (Sometimes called VxVM)

URL <http://www.sun.com/storage/software/volumemgr.html>.

From the above URL

VERITAS Volume Manager software is an advanced, system-level disk and storage array solution that alleviates downtime during system maintenance by enabling easy, online disk administration and configuration. The product also helps ensure data integrity and high availability by offering fast failure recovery and fault tolerant features. VERITAS Volume Manager software provides easy-to-use, online storage management for enterprise computing and emerging Storage Area Network (SAN) environments.

Through the support of RAID redundancy techniques, VERITAS Volume Manager software helps protect against disk and hardware failures, while providing the flexibility to extend the capabilities of existing hardware. By providing a logical volume management layer, VERITAS Volume Manager overcomes the physical restriction imposed by hardware disk devices.

Highlights:

Data redundancy (RAID 0, 1, 0+1, 1+0, 5).

Dynamic multipathing (DMP) support.

Intuitive Java technology-based platform-independent graphical user interface.

Enables easy movement of data between nodes in a SAN environment.

Features	Benefits
HIGH AVAILABILITY -----	
On-line storage reconfiguration and database table space growth	Uninterrupted data access during planned system maintenance
Mirroring of storage (RAID-1)	Higher availability with fault tolerance and faster data access
Data redundancy (RAID-5, RAID-1+0, RAID-0+1)	Reduces storage costs for data redundancy or by maintaining access to data even when multiple disks fail
Hot relocation of failed redundant storage	Automatic restoration of data redundancy when disks fail
DRL (Dirty Region Logging)	Provides fast recovery after a system failure
SCALABILITY/PERFORMANCE -----	
On-line performance monitoring and tuning tools	Tools to assist in identification and minimization of I/O bottlenecks
Striping (RAID-0) and selective disk mirroring	Increases throughput and bandwidth while providing scalable performance and balancing of application data loads
Dynamic Multipathing (DMP) support	Increases performance to multi-controller disk arrays by spreading I/O between the multiple paths into the array. Easier access to data and automatic path recovery across redundant fibre channel loops
Spanning of multiple disks	Eliminates physical storage limitations
On-line relayout	On-line tuning ability enables reconfiguration of storage layout
Task monitor	Ability to throttle volume recovery speed and monitor VM I/O tasks

CENTRALIZED MANAGEMENT

Intuitive Java-based platform-independent graphical user interface

Displays logical view of storage devices, providing easy monitoring of disk I/O configuration and management that improves productivity

Device driver, file system, and database independence

Easy integration with disk subsystems and arrays including Hardware RAID systems

Free space pool management for automatic or directed allocations

Simplifies administration and provides flexible use of available hardware

Logical diskgroups

Enables easy movement of data between nodes in a SAN environment

Maximum volume size button

Automatically calculates the maximum free space available for building volumes on chosen disks

Unique disk ids

Eliminates disk misidentification allowing disks to be added and removed and configured automatically and on-line

HETEROGENEOUS SUPPORT

Ability to move diskgroups between different versions of

Easier migration with reduced downtime volume Manager

Available on numerous Sun hardware platforms

Reduces training costs

Supports multiple Sun disk arrays

Provides maximum flexibility by allowing businesses to select the storage hardware solution that best meets their needs

INTEGRATION

Data Snapshot

Provides mirrored data snapshot which enables on-line backup of data via backup application such as VERITAS NetBackup

More on Veritas volume manager, from URL <http://www.veritas.com/us/products/volumemanager/> which contains number of white papers and data sheet on the product.

3.5 The Vinum Volume Manager

URL <http://www.lemis.com/vinum.html> FreeBSD volume manager. From above URL:

The Vinum Volume Manager is a block device driver which implements virtual disk drives. It isolates disk hardware from the block device interface and maps data in ways which result in an increase in flexibility, performance and reliability compared to the traditional slice view of disk storage. Vinum implements the RAID-0, RAID-1 and RAID-5 models, both individually and in combination.

Also see <http://www.shub-internet.org/brad/FreeBSD/vinum.html>

3.6 Compaq SANworks Enterprise Volume manager

URL: <http://www.compaq.com/products/storageworks/Storage-Management-Software/evmindex.html> From above URL:

Product Description

Key Features

Cloning and Snapshot - SANworks Enterprise Volume Manager use can be optimized by selecting either cloning or snapshot, depending on the application. Snapshots are virtual copies and clones are physical copies. Snapshot is ideal for quick recovery. Both are ideal for non-disruptive backup.

Web-based application - SANworks Enterprise Volume Manager is accessible from any system that has a web browser.

Multi-platform support - SANworks Enterprise Volume Manager operates on the RAID Array 8000 (RA8000) and the Enterprise Storage Array 12000 (ESA12000) using HSG80 controllers in switch or hub configurations. SANworks Enterprise Volume Manager supports Windows NT V4.0, Windows 2000, Sun Solaris V2.6, 7 and Tru64 Unix V4.0F/G. Other platform support is to follow. SANworks Enterprise Volume Manager provides consistent storage management regardless of the platform.

Plug and Play with existing applications - Microsoft Exchange, StorageWorks Enterprise Backup Solution, VERITAS NetBackup, VERITAS Backup Exec, Legato NetWorker, CA ARCServeIT, Oracle, and Microsoft SQL and with plans to support other applications in the future.

Supports LAN-less backup - Backup data is isolated from the general purpose LAN, so there is no network performance degradation during backup. All volume movement is on the SAN.

FC-AL or Switched Fibre Channel topologies supported - Customers with either technology can take advantage of the features of EVM. Snapshots are available in switch configurations only.

3.7 Sun StorEdge Volume Manager (SSVM). (repackaged version of Veritas Volume Manager.)

URL's:

<http://www.sun.co.uk/services/educational/catalog/courses/UK-ES-310.html>

<http://www.carumba.com/talk/veritas/volumemanager.shtml>

3.8 AIX Logical Volume Manager

<http://www.redbooks.ibm.com/abstracts/sg245432.html>

3.9 Sequent Volume Manager (ptx/SVM)

<http://www.numaq.com/products/software/layered/svm.html> From above URL:

Sequent Volume Manager (ptx/SVM[tm]) provides enhanced disk management facilities and increased system availability for Sequent's NUMA-Q[tm] servers running DYNIX/ptx, Sequent's enhanced implementation of the UNIX operating system. Based on an emerging standard, VxVM[tm] from Veritas[tm] Software Corporation, ptx/SVM offers significant enhancements over traditional mirroring products. These include features that significantly improve data integrity, system availability, and system performance.

In addition to its sophisticated mirroring capabilities, ptx/SVM also provides disk striping, disk concatenation, hot sparing, and on-line disk management. With on-line disk management, a system administrator can optimize disk performance by moving data between disks while the system is running.

Drives under ptx/SVM control can be dynamically resynchronized with one or more mirrored partners, independent of the disk controller, without taking the system off-line. ptx/SVM can also control the resynchronization rate, which can be set to minimize impact on performance or to minimize the time required to perform a resync operation.

ptx/SVM offers users the advantages of open systems with its access to powerful and dynamic volume management tools.

ptx/SVM Highlights

Disk Mirroring

Data availability and integrity are increased with the continuous maintenance of up to 32 copies of critical data. ptx/SVM automatically uses these data mirrors in the event of a disk failure.

ptx/SVM adds greatly to system availability by allowing system administrators to dynamically create, remove, and allocate mirrors, as well as perform on-line resynchronization and snap-shot backups with a minimum impact on users.

If mirroring is used, depending on the layout, ptx/SVM may automatically divide the read load among all the mirrors, creating multiple read paths which can enhance system performance.

Disk Concatenation

Disk concatenation allows a user to create logical volumes that can span multiple disks. Two or more physical disks or disk segments can be viewed as a single entity.

Disk Striping

Striping allows portions of multiple disks to be viewed as a single logical entity. Striping improves performance by distributing the data of a heavily-used partition over several disks, thus increasing the number of heads available to read and write data.

Hot Sparing

ptx/SVM allows the designation of dedicated disks as hot spares that are used to replicate mirrored data from disks that have failed, thus increasing the availability of mirrored data.

Disk groups

ptx/SVM allows for the segregation of disks into logical groups called disk groups. Disk groups improve access to data objects by maintaining separate databases of data objects and allows for the ability to create up to 100,000 data objects providing the ability for a system to scale to very large disk farms. Disk groups can also be "exported" from one system and "imported" to another system simplifying test and staging environments.

On-Line Volume Management

ptx/SVM provides an easy-to-use administrative interface that allows data to be moved among disks while the system is running. I/O performance can be optimized on-line by reorganizing the data volumes, and general administrative tasks can be performed on-line.

ptx/SVM also supports "rolling upgrades" that allows administrators to upgrade ptx/SVM in clustered systems with minimal system downtime.

Java Based, Menu-Oriented, or Command Line User Interfaces

ptx/SVM provides support for a Java based graphical user interface, a menu-driven interface and a conventional command line interface for system administrators.

Command Point SVM, Sequent's port of Veritas' new GUI Volume Manager Storage Administrator, performs three primary roles:

It enables an administrator to perform ptx/SVM administrative actions (e.g., creating, modifying, and removing ptx/SVM objects) using GUI dialogs instead of the command line.

It provides top-down and detailed views of the ptx/SVM configuration.

It reports many ptx/SVM error conditions, such as I/O errors and failure of the volume configuration daemon.

Dirty Region Logging (DRL)

DRL is a fast resynchronization mechanism for private storage. If a mirrored volume needs to be resynchronized due to a system crash, only the addresses with outstanding writes stored in the log need to be resynchronized.

Sequent Support

Sequent offers full product support for ptx/SVM, including a manual for system administrators and training classes specifically for ptx/SVM. Consulting services are also available to assist with particular system configurations and implementations.

3.10 HP Shared Logical Volume Manager (SLVM)

<http://www.software.hp.com/STK/partner/relnotes/10.20/chp3.html#LVM>

From above URL:

For 10.10, LVM has been modified to support volume groups that are connected to multiple systems. This feature is called Shared LVM (SLVM) and is supported only on S800 machines.

SLVM is a mechanism that permits multiple systems in an MC/LockManager cluster to share (read/write) disk resources in the form of volume groups. The objective is a highly available system by providing direct access to disks from multiple nodes and by supporting mirrored disks, thereby eliminating single points of failure.

SLVM permits a two system cluster to have read/write access to a volume group by activating the volume group in shared mode.

SLVM is designed to be used only by specialized distributed applications (such as Oracle Parallel Server) that use raw access to disks, rather than going through a file system. The applications must provide their own concurrency control for their data, as well as transaction logging and recovery facilities, as appropriate. Applications that are not network aware, such as file systems, will not be supported on volume groups activated in shared mode.

SLVM requires services provided by MC/LockManager and thus only clusters that have MC/LockManager will be able to use shared activation.

4 Other links to Volume Managers

- Logical Volume Manager HOWTO.
<http://www.ds9a.nl/lvm-howto/HOWTO//cvs/lvm-howto/lvm-howto.html>
- AIX logical volume manager Overview.
http://nscp.upenn.edu/aix4.3html/aixbman/baseadm/lvm_overview.htm
- news group for Veritas volume manager: veritas.volume.manager.english.
See http://support.veritas.com/nntprules_ddProduct_VOLUMEMAN_crumb_on.htm to join.

- veritas volume manager white papaers.
<http://www2.veritas.com/us/products/volumemanagernt/whitepapers.shtml>
- Heinz Mauelshagen's LVM (Logical Volume Manager) howto.21/02/2000
<http://linux.msede.com/lvm/doc/LVM-HOWTO>
- SGI XFS file system.
<http://www.sgi.com/Technology/xfs-whitepaper.html>
- IBM paper. Open systems, storage, and performance for the mainframe mind.
<http://www.storage.ibm.com/hardsoft/diskdrls/technolo/open.htm>
- A Walkthrough of the Logical Volume Manager for Linux
<http://www.gweep.net/sfoskett/linux/lvmlinux.html>