Quick Reference: AIX Logical Volume Manager and Veritas Volume Manager October 2000

November 30, 2000

Johnny Shieh

Quick Reference: AIX Logical Volume Manager and Veritas Volume Manager

October 2000

Quick Reference: AIX Logical Volume Manager and Veritas Volume Manager

October 2000

Table of Contents

Quick Reference: AIX Logical Volume Manager and Veritas Volume Manager

In the world of UNIX storage management, there are two primary leaders: IBM and Veritas. Both companies offer products that help UNIX system administrators manage storage in very flexible methods in comparison to older UNIX implementations. IBM offers the Logical Volume Manager (LVM) as part of its Advanced Interactive Executive (AIX) operating system. The LVM is built into the base operating system and is provided as part of the base AIX installation. Veritas offers the Veritas Volume Manager (VxVM) which is either packaged as a standalone add-on or part of a larger package such as the Veritas On-Line Storage Manager. VxVM is designed to be an additional software package added to a UNIX operating system, most notably the Solaris operating system by Sun Microsystems, Inc.

For detailed information about the AIX operating system, refer to the following Web address: http://www.ibm.com/servers/aix/library/.

AIX library information is listed under *Technical Publications*.

Introduction

This document will help system administrators co-manage AIX systems from the LVM point of view. A table of limitations that affect both VxVM and LVM operations is included. Where values differ greatly, an explanation is given from the LVM point of view as to why LVM differs from VxVM. It is not the purpose of this paper to explain VxVM limitations. Finally, although limits and actions between VxVM and LVM may be close or identical, execution may not be the same.

Terminology

LVM and VxVM use the following terms to specify their various components:

AIX Logical Volume Manager	Veritas Volume Manager
Volume Groups	Disk Groups
Physical Volumes	Disk Access/Disk Media
Physical Partitions	Subdisks
Logical Volume	Volume
Logical Partition	Plex
Logical Volume Mirrors	Logical Partition copies

Limitations

This section discusses limitations in the following areas:

- Volume groups
- Physical volumes in volume groups
- Physical partitions per volume group
- Logical volumes per volume group
- Logical partitions per logical volume

- Mirrored copies of a logical volume

Volume Groups

The limit of 256 possible volume groups has been a part of AIX since LVM's inception in AIX 3.1. With the advent of a 64-bit operating system, the volume group limit rises to 4,096 entities. Because of constant increases in disk capacity and the use of multiple physical *and* logical volumes within the volume group, the number of volume groups is not a hindrance to customers.

Parameter	**LVM**	**VxVM**
Equivalent	Volume Groups	Disk Groups
Limits	32-bit: 256 64-bit: 4,096	None

Physical Volumes in Volume Groups

From AIX 3.1 to AIX 4.2, the limit of 32 disks per volume group was fixed with a “standard” volume group. Starting with AIX 4.3, the user can create a “big” volume group that has the expanded disk count. Volumes created at an older level of AIX are upwardly compatible; that is, a volume group created on AIX 3.1 or 3.2 can be introduced into a system running on AIX 4.3. However, downward compatibility of volume groups is *not* supported. Again, because of the rapid growth of commercial disk volumes and the ability to specify physical partition sizes within a volume group, the current limit of 128 disks per volume group is not a problem.

Parameter	**LVM**	**VxVM**
Equivalent	Physical Volumes in Volume Group	Disk Access & Disk Media
Limits	Standard: 32 *or* Big: 128	None

Physical Partitions per Volume Group

The following table reviews the limitations in the physical partitions per volume group.

Parameter	**LVM**	**VxVM**
Equivalent	Physical Partitions	Subdisks
Limits	1,016 per disk *or* 1016*128 = 130,048 per volume group	None

Logical Volumes per Volume Group

The following table reviews the limitations in the logical volumes per volume group.

Parameter	LVM	VxVM
Equivalent	Logical Volumes	Volume
Limits	Standard: 256 *or* Big: 512	None

Logical Partitions per Logical Volume

The following table reviews the limitations in the logical partitions per logical volume.

Parameter	LVM	VxVM
Equivalent	Logical Partitions in a Logical Volume	Subdisks per plex
Limits	32,512	4,096

Mirrored Copies of a Logical Volume

At the present time, the capability to have more than three mirrors is not seen as a necessity. Along with the increased disk requirement, the more copies a logical has, the longer "writes" will take because all "writes" must return from their respective disk platters before a "write" returns to the file system. Experience shows that customers tend to use only two of the three possible mirrored copies.

Parameter	LVM	VxVM
Equivalent	Mirrors per Logical Partition	Copies per plex
Limits	3	32

Functional Differences

Software RAID Levels

Redundant Arrays of Independent Disks (RAID) is formally defined as a method to store data on any type of disk medium. There are five formal levels, plus the level RAID-0+1, which was created at the grass roots level. In both VxVM and LVM, RAID levels 2, 3, and 4 are viewed as inappropriate RAID levels to pursue in a volume management product. However, VxVM offers RAID 5, whereas LVM does not. The RAID-5 strategy is a method of spreading data onto multiple disks and recovering from disk failures by doing parity calculations. During LVM's development, hardware I/O controllers were designed to handle the RAID-5 strategy specifically. The parity calculations required for "reads" or "writes" or recovery are considered a detriment to performance in LVM, rather than an improvement. Although VxVM offers RAID-5, their documentation also points out that RAID-5 is more efficiently handled by a hardware solution rather than software.

RAID Level	LVM	VxVM
RAID 0 (striping)	Yes	Yes
RAID 1 (mirroring)	Yes	Yes
RAID 0+1 (mirroring stripes)	Yes	Yes

RAID Level	LVM	VxVM
RAID 5	No	Yes

Mirrored Read Policy

In the case of mirrored “reads,” the volume management program is given great flexibility because there are supposed to be multiple copies of data. And because the write algorithm is guaranteed to perform consistent “writes,” any copy of one piece of data is as good as the others when it comes to “reads.” The *round robin* method tracks the last copy used for a read request and sends the read request to an alternate copy (which implies another physical disk). The algorithm follows a list of disks and mirrors until the list has been traversed, and then loops back to the beginning of the disk mirror list.

The *closest head to data* algorithm is one which the read algorithm analyzes the physical address of a read request (for all mirrors) and then looks for the *end head* location (where a head will be after it completes a read or write). Then the algorithm will send the read request to the disk that will have to traverse the least amount of disk territory in order to reach the location for the read. This was the primary LVM mirror read algorithm until AIX 4.1, when it was completely scrapped, because the complex math and head location tracking code required for this algorithm did not give the read request any significant advantage over a straightforward algorithm of *least busy*.

In the least busy algorithm, when a read request arrives to LVM, the code determines which disk (that has a copy of the required data) has the fewest outstanding I/O requests asked of it. It then assigns the read request to that particular disk drive. This is a much simpler and easier-to-maintain algorithm. LVM then introduced round robin in AIX 4.3 to allow users the option to spread the disk usage evenly among disk drives.

Method	LVM	VxVM
Primary Method	Least busy disk	Round Robin
Alternative Method	Round Robin	Closest head to data

Hot Spot Management

Hot spot management occurs when the user wants to mirror a section of data that is being used frequently for “reads” or “writes.” Logically, if it is being written too often, then the user must think that data is important and wants a mirrored copy in case of a disk failure. For read purposes, mirroring data will speed up the data access on read requests. VxVM has a method in which running a command will analyze the *hot spot* and mirror that portion of disk with another disk. The LVM option for this is not as complex. A command option will display the read/write load for all logical volumes. Then it will be up to the user to manually mirror the logical volumes he thinks are important. The LVM tool provides only the information, not the automatic mirroring.

Read/Write Resynchronization

Two major methods exist to recover from a system crash when it may have occurred during mirrored “writes.” One of the reasons this is a concern is that writes to mirrors are done in parallel. Because a user cannot be sure of the status of the parallel “writes” (such as being incomplete when a system crashed), the mirrors that are supposed to be identical will not be identical. Worst of all, there is no way to determine which copies could be out of sequence with each other.

One method of preventing this problem is for the code that controls the mirrored "writes" to add the write entry into a log. Thus, if a crash occurs during the recovery of the system crash, the mirror code goes to the small log and looks up the "writes" that may have been in "flight" during the crash. Then the volume manager resynchronizes these data portions, even though there may be no differences in data. The downside to this method is that sometimes for every write (to the data on the disk), the user must perform a "pre-write" to the disk. Thus, one write request takes two write actions to accomplish. The upside to this concept is that if a crash were to occur, the recovery work and time to resynchronize mirrors is very minimal.

The second method that volume managers use to enforce mirror consistency is to treat a mirror as totally out of sync. Instead of a log file to track writes, no tracking is done before a write goes to the disk platter. Instead, in the event of a system restart, the volume mirroring code assumes that *all* the mirrors are out of sync. Then the volume manager starts two types of recovery. First, every read request from the user causes the mirroring code to pick one of the mirrors' data and return it to the user. But at the same time, the mirroring code uses this same data and writes it over the other mirrors. Thus, further requests to that same data or its mirrors will result in the same consistent data. Second, the mirroring code initiates a background resynchronization which goes slowly through the entire mirror, taking one copy and using it as a master copy for the other copies. This is done in case there are no users who ever request data from other portions of the mirror. The positive portion of this algorithm is that writes that occurred before the system crash only take one write to complete because there is no log to update. The downside to this algorithm is that after a system restart, the mirrored database will perform sluggishly until all resynchronizations are complete.

LVM and VxVM offer both these type of resynchronization algorithms. However, each uses a different default recovery process.

Resync Methods	**LVM**	**VxVM**
Default Method of Resync	Resync only those partitions marked dirty in a log	Forced resync of all partitions
Alternative Method of Resync	Forced resync of all partitions	Resync only those partitions marked dirty in a log

Real-time File System Resizing

Both VxVM and LVM enable file systems to be expanded while the file system is mounted and in use. Both volume managers are tied closely to specific file systems. VxVM only works with the Veritas File System, and LVM only works with the JFS or JFS2 (enhanced JFS) file system found on AIX. The one difference between VxVM and LVM is the philosophy on the concept of file system shrinkage. LVM/JFS/JFS2 does not allow the user the ability to shrink the file system. The reasoning behind this is that sometimes portions of files are spread out over the file system for performance reasons. And when a file system needs to be shrunk, the file system must go out and gather up the "pieces" of a file and relocate them to a portion of the file system that will not be eliminated. This method is time-consuming and not very efficient for performance. Thus on AIX, file system reduction is not permitted on JFS and JFS2.

VxVM takes a different approach to this. It does allow the reduction of a file system size, but it is very simplistic and has drawbacks. They literally “chop” off the end of the logical volume and file system up to the point where the user wants the file system reduced. Warnings are included in the VxVM product regarding this practice; if the user has data at the end of the file system, the data may be corrupted or lost. Thus, for the purposes of this paper, the ability of VxVM to reduce file system size is considered limited.

Common Features

The following is a list of the common features in the LVM and VxVM methods of storage management:

- Online backup of file systems
- Snapshot backup of file systems
- Quick resynchronization of changed partitions after online or snapshot backup
- Migration of data with active volumes
- Transparent data stream switch after mirrored disks fail
- Ability and limits to operate in a multinode concurrent configuration
- Commands to replace dead or failing drives
- Hot spare, standby disks

Online Backup of File Systems

A mirrored file system can be split off and a copy of the file system can be archived to backup media such as tape or optical store. While this is occurring, the other mirrors that make up the file system still allow users to read and write to the mirrors. This backup method is primarily designed as a performance enhancement. One set of disks (the mirrors not performing backups) can still be used to satisfy user requests without getting bogged down by the request of the backup device.

Snapshot Backup of File Systems

A subset of the online backup option, snapshot backup allows an equivalent of a snapshot of the system so that at a given time, a frozen picture and status of the file system can be recorded. Typically, when a backup is performed on an active system, a few “blurred” files may exist. Blurred files are those being altered by an application and archived at the same time.

Quick Resynchronization of Changed Partitions After Backup

When a mirror is “broken off” into a standalone entity for backup, there is a high probability that a change occurred to the still-running file system that is not being archived. After the backup archive has completed, the detached mirror must be reinserted into the mirroring group with the lowest disruption of service and performance. This function will resynchronize the returning mirror in only the areas of the mirrors that changed during the time period where the mirror was either online or snapshot backup.

Migration of Data with Active Volumes

Data can be moved from disk-to-disk while the data residing on those platters are in use. This is done through the process of mirroring data (which causes the data to move from one disk to another) and then unmirroring data from the source disk. The end result is that it seams that stored file systems or data moves from one disk to another without having to unmount a file system or shutdown a system.

Transparent Data Stream Switch After Mirrored Disks Fail

When a disk error occurs on a mirrored system, users are not aware of this occurrence unless they check the error logs for the system. After a mirror has failed, the mirroring code does *not* attempt any future I/O to the failed mirror unless explicitly ordered to do so. This method can save on future performance problems that are associated with trying to visit a disk that will never respond, because it no longer functions.

Operation in a Multinode Concurrent Configuration

When multiple computer systems are attached to the same common set of disk drives, a large set of disks can be partitioned off or shared by CPUs that are physically separated. This is very important to high-availability systems that need data access continuously for extended periods of time.

Commands to Replace Dead or Failing Drives

Appropriate commands are designed specifically to be used by system engineers to replace disks and move information from a failing disk to a new, healthy disk.

Hot Spare, Standby Disks

Hot spare, standby disks can be used to migrate the contents of data from one disk to a spare disk when the system believes that a disk is starting to fail. This is done with the temporary mirroring of data.

Commands

The following table gives a list of general actions used in the system management of LVM and the equivalent command set used to administer VxVM volumes.

Tasks	**AIX Logical Volume Manager**	**Veritas Volume Manager**
Create a Volume Group	mkvg	vxdg init
Create a Logical Volume	mklv	vxassist make
Add a Physical Disk to a Volume Group	extendvg	vxdiskadd
Add Physical Partitions to Logical Volume	extendlv	vxassist growto
Change characteristics of Logical Volume	chlv	vxedit set
Remove Physical Disk from Volume Group	reducevg	vxdiskadm
Remove Logical Volume from Volume Group	rmlv	vxedit rm
Remove definition of Volume Group from operating system	exportvg	?
Display information about physical volumes, volume groups or logical volumes	• lspv • lsvg • lslv	vxstat
Move Logical Volume to from one Physical Volume to another Physical Volume	migratepv	vxassist move

Tasks	**AIX Logical Volume Manager**	**Veritas Volume Manager**
Administer disks	extendvg *or* reducevg	vxdiskadm
Set up disks	extendvg	vxdisksetup
Create configuration records for storage structures	mkvg *or* mklv	vxmake
Manage plexes or Volume Groups	mkvg *or* mklv	vxplex
Display Volume Group information	lsvg	vxprint
Change size of Logical Volume	extendlv *or* chlv	vxresize
Manage subdisk or physical volume	chpv	vxsd
Manage volume	• chlv • mklv • rmlv	vxvol
Set up sysboot information on VM disk	bosboot	vxbootsetup
Manage VM disks	N/A	vxdisk
Back up operating system	mksysb *or* mkcd	Solstice Backup: nwadmin
Restore operating system	mksysb *or* mkcd	• nwadmin • nwrecover

Special Notices

This document was produced in the United States. IBM may not offer the products, programs, services or features discussed herein in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products, programs, services, and features available in your area. Any reference to an IBM product, program, service or feature is not intended to state or imply that only IBM's product, program, service or feature may be used. Any functionally equivalent product, program, service or feature that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, service or feature.

Information in this document concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. Sources for non-IBM list prices and performance numbers are taken from publicly available information including D.H. Brown, vendor announcements, vendor WWW Home Pages, SPEC Home Page, GPC (Graphics Processing Council) Home Page and TPC (Transaction Processing Performance Council) Home Page. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed ″AS IS″. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. The use of this information or the implementation of any techniques described herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Customers attempting to adapt these techniques to their own environments do so at their own risk.

IBM is not responsible for printing errors in this publication that result in pricing or information inaccuracies.

The information contained in this document represents the current views of IBM on the issues discussed as of the date of publication. IBM cannot guarantee the accuracy of any information presented after the date of publication.

Any performance data contained in this document was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally available systems. Some measurements quoted in this document may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries: AIX. A full list of U.S. trademarks owned by IBM can be found at http://iplswww.nas.ibm.com/wpts/trademarks/trademar.htm.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

Other company, product and service names may be trademarks or service marks of others.

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.