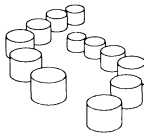




DG/UX Technical Brief

December 6, 1993

Information About AViiON® Systems from Data General's UNIX® Development Group



In This Issue:

Managing Virtual Disks with On-Line Storage Management (OSM)

Contents

How OSM Can Benefit You.....	2
On-Line Backups.....	2
On-Line Data Storage Failure Recovery	5
Flexible Data Storage Management..	8
Performance Improvements.....	10
Cache Integration	17
Other Advantages of OSM.....	22
Making the Transition to OSM From LDM.....	23
Summary of OSM Advantages	26
Questions and Answers	27
Related Documentation	30
OSM and LDM Terminology	31

Today's computing environments put a greater premium than ever on high availability and high performance of applications running on AViiON® platforms. To deliver the storage management functions normally found in mature system software, such as MVS, Data General introduces On-line Storage Management (OSM).

OSM improves on its Logical Disk Management (LDM) predecessor by providing two important innovations:

- A dynamic, on-line data storage management facility
- A flexible hierarchy to organize and manipulate data storage resources

The on-line data storage management facility allows you to manipulate your physical disk partitions without having to take your data off

line by unmounting file systems or making your database quiescent. This OSM feature is a marked enhancement to LDM, which first required that you take your data off line to change the disk configuration.

As your application and resource requirements change, you may want to reconfigure your system to achieve a different performance level or to increase capacity without affecting availability. OSM offers a flexible hierarchy that allows you to dynamically restructure configurations.

AViiON is a registered trademark and CLARiiON and DG/UX are trademarks of Data General Corporation.
 FrameMaker is a registered trademark of Frame Technology Corporation.
 Legato Networker is a trademark of Legato Systems, Inc.
 ONC is a trademark and NFS is a registered trademark of Sun Microsystems, Inc.
 UNIX is a registered trademark of UNIX System Laboratories, Inc.
 ©1993 Data General Corporation.
 All Rights Reserved.

Produced on a Data General AViiON AV4000 with FrameMaker® 3.1X.

012-004406-00

How OSM Can Benefit You

The on-line data storage management facility and hierarchical flexibility provide the following specific benefits:

- ❑ On-line backups
- ❑ On-line data storage failure recovery
- ❑ Dynamic resource configuration management
- ❑ Performance improvements
- ❑ Cache integration

As you encounter a new term while reading this brief, you may refer to the final section for a list of keywords and definitions.

On-line Backups

OSM makes it easier to perform backups of file systems on virtual disks without having to take the virtual disk's data off line.

OSM's hierarchical facility allows you to mirror an aggregation that contains critical data to be backed up on a regular basis. Figure 1 shows the critical data located in a mirror named **curly**. Aggregation **larry** and partition **moe** form the mirror images.

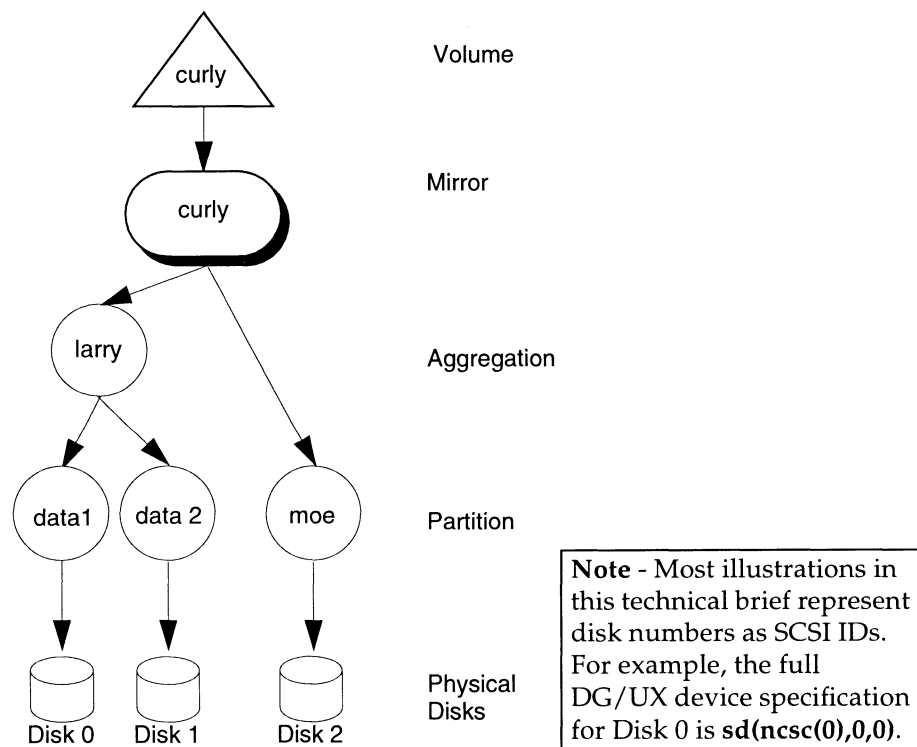


Figure 1 Mirror Containing Critical Data to be Backed Up

Note – In Figure 1, notice that partitioned virtual disks named **data1**, **data2**, and **moe** have the same function as LDM's unnamed logical disk pieces.

Assuming that the virtual disks to be used as mirror images pre-exist, you might create the mirror by using the following **sysadm** operations:

```
Device -> Disk -> Virtual -> Mirrors -> Mirror
```

```
Virtual Disk: curly
```

```
New Name for Mirror: curly
```

```
New Name for Image: larry
```

```
Another image? [yes]
```

```
Child Virtual Disk: moe
```

```
Begin Sync Immediately? [yes]
```

```
Another image? [no]
```

Next, you unlink one image from the mirror for backup while the other mirror image remains on line and in use. You can accomplish this task by using the following **sysadm** operation.

```
Device -> Disk -> Virtual -> Mirrors -> Unlink
```

```
Existing Mirror: curly
```

```
Image Virtual Disk(s) to Unlink: moe
```

Note – Before you unlink the image from the mirror, the application must flush its data to ensure that the data on the mirror is up-to-date.

The DG/UX system ensures that the virtual disk's file system is synchronized before it performs the unlink operation, making the mirror's images consistent and immediately mountable.

Already a volume, the unlinked image must be mounted before it can be dumped to tape. Figure 2 shows single image, **curly**, in use and the unlinked, duplicate image, **moe**, mounted as a separate volume.

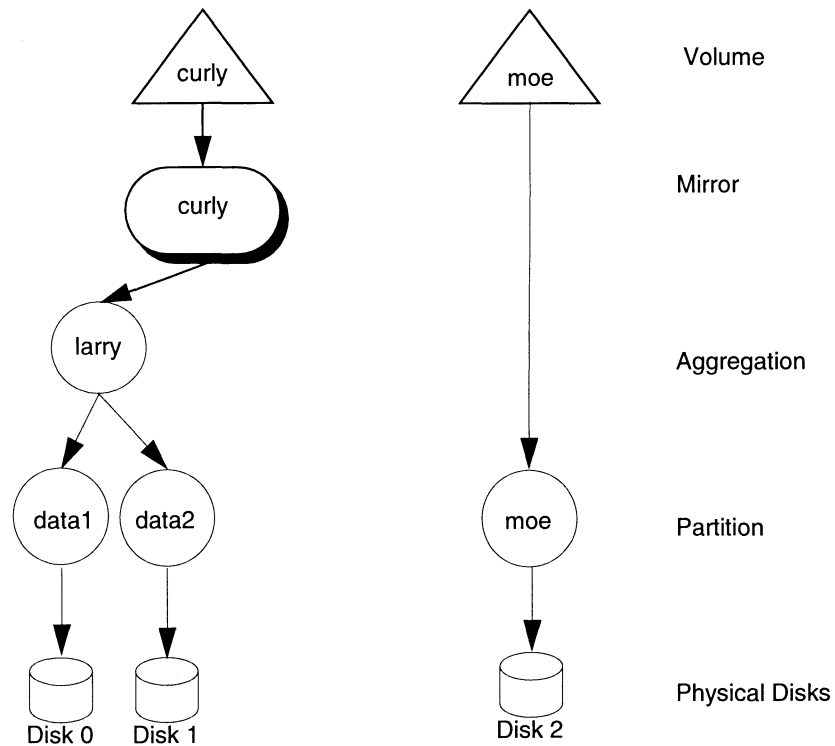


Figure 2 Mounting a Separate Volume

After you mount the partitioned virtual disk's file system, you then can use the desired tool, such as the Legato Networker™ product or the **tar**, **cpio**, or **dump2** commands, to back up the contents of the file system to tape. Following the backup, you should relink the partition to the critical data and synchronize the data, keeping the mirror image available for the next backup.

This technique provides you with a consistent copy of the duplicate data to be backed up while still having continuous read and write access to the data located on the first mirror image—with minimal performance loss. An application that is still accessing a virtual disk being synchronized will experience some performance degradation. You can reduce this impact through **sysadm**'s synchronization throttling operation `Device -> Disk -> Virtual -> Mirrors -> Throttle Sync`.

After the dump is complete, you can relink image **moe** to the mirror and synchronize the two images. The following **sysadm** operation performs these tasks.

```
Device -> Disk -> Virtual -> Mirrors -> Link
Existing Mirror: curly
Child Virtual Disk: moe
Begin Synchronization Immediately: [yes]
```

On-line Data Storage Failure Recovery

OSM enables you to recover from physical disk failure without having to take the data off line, allowing data processing activities to continue as usual.

Imagine a situation in which you notice an increase in soft errors on a physical device that contains critical data. The simple solution to the problem is to copy the data elsewhere. But, to do so under LDM, the entire logical disk on the failing device has to be taken off line. Taking the data off line halts all data processing activities for the duration of the recovery procedure.

Under LDM, after notifying affected users to temporarily stop their work, you then would have to perform a lengthy set of steps to move the logical disk contents of the failing disk to a known good disk.

With OSM, using far fewer steps and without having to take the data off line, you can fix the failing disk problem with the **sysadm** utility or the **admvdisk** commands (see Figure 3). You must first create the destination partitioned virtual disk to receive the data from the failing disk using the following **sysadm** menu selection sequence:

```
Device -> Disk -> Virtual -> Create
New Virtual Disk Name: extra
Striped? [no]
```

Note – The striping method you choose depends on your performance requirements. Shown here is only an example, no striping selected, which may not suit your needs.

```
Select Space by: Disk to partition and partition size
Disk to Partition: sd(incr(0),2,0)
Length of Piece in Blocks: 10000
Do you want to specify more pieces for this virtual disk? [no]
```

To execute the move operation, follow these steps:

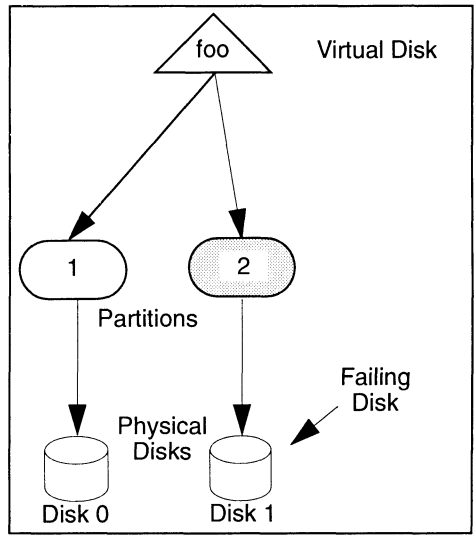
```
Device -> Disk -> Virtual -> Move
Source Virtual Disk: 2
```

Note – 2 is the name of the source partitioned virtual disk. See Figure 3.

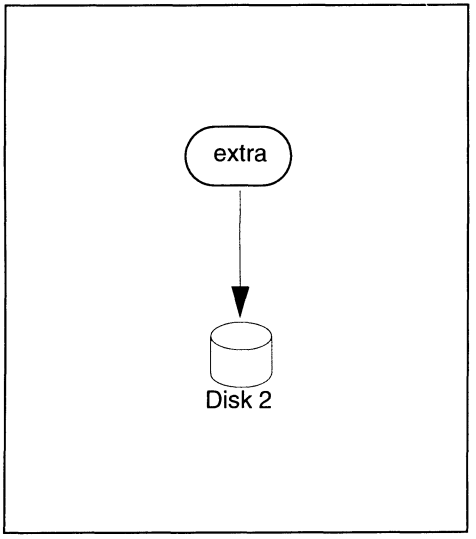
Destination Virtual Disk: **extra**

Implicit in OSM's move operation is the use of mirroring. The sequence of figures on the following two pages illustrates the actions associated with the OSM move operation. In the figures, the partition on the failing device is highlighted with shading.

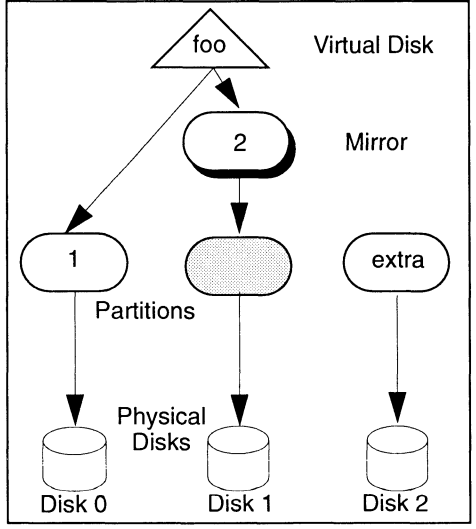
A) Data on the failing disk (Disk 1) is in jeopardy.



B) You must create explicitly a destination partitioned virtual disk (called "extra" in this example) on a known good disk.



C) The move operation places a mirror at the failing partition. The mirror assumes the name of partitioned virtual disk 2, leaving the partition nameless.



D) The move operation links the new image, extra, to the mirror.

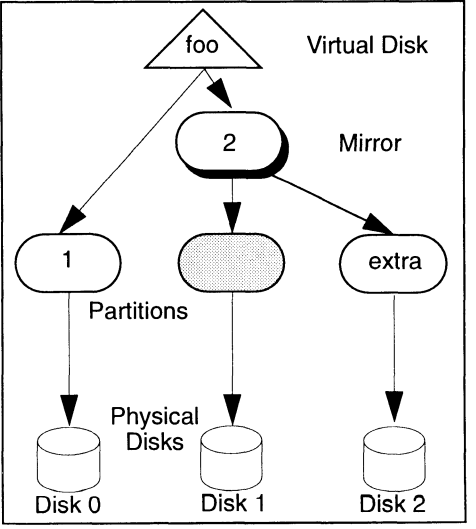
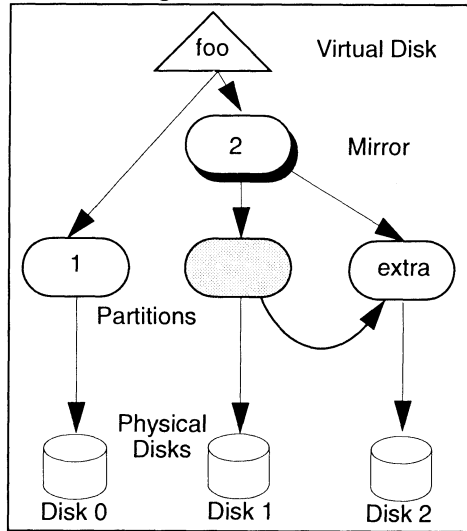
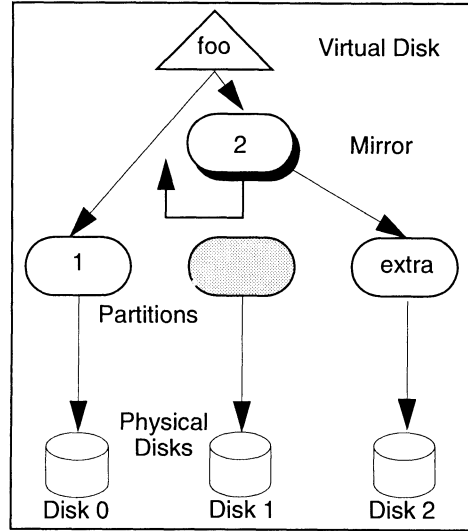


Figure 3 The OSM Move Operation (continued)

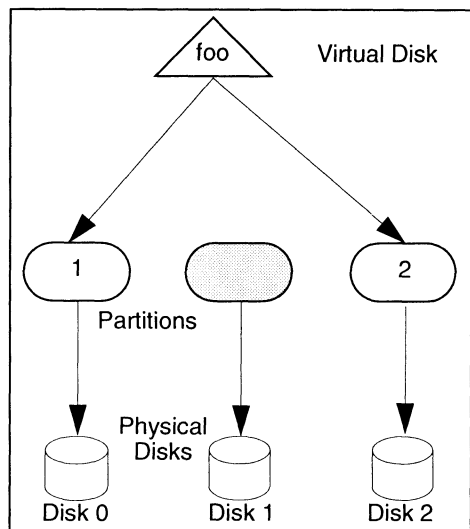
E) The move operation synchronizes the mirror images.



F) The move operation unlinks the failing partitioned virtual disk from the mirror.



G) The move operation extracts the mirror from the configuration.



H) The move operation explicitly removes the partitioned virtual disk on the failing disk.

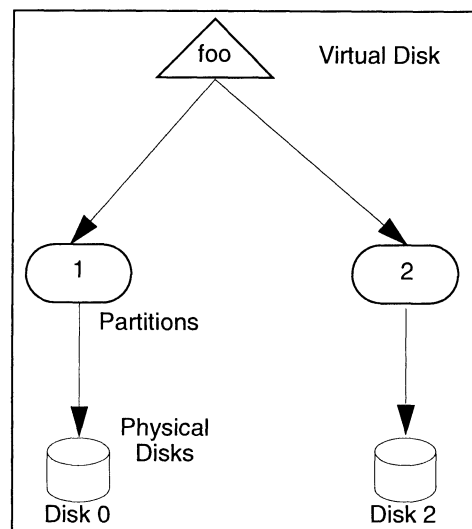


Figure 3 The OSM Move Operation

The result shown in step H is almost identical to that in step A; the exception is that Disk 2 replaces failing Disk 1.

Flexible Data Storage Management

OSM's hierarchical facility gives you the ability to solve interesting design or performance problems that LDM does not.

A flexible hierarchy not only lets you create complex virtual disk layouts, but it also lets you create simple hierarchical layouts using aggregations. An aggregated virtual disk results from collecting the desired physical disk resources into one large virtual disk, giving the appearance of a continuous resource stream.

You then carve out the desired space from the aggregation, irrespective of physical disk origin, to use as mountable volumes. You may choose to use an aggregation if you do not want to keep track of free space spanning multiple physical disks. Should you desire fast data on an aggregation, you need to stripe it only once rather than striping each partition you carve from it.

Note – When striping, each partition forming the aggregation must be the same size. In addition, you cannot expand or shrink striped aggregations.

Figure 4 shows how you might aggregate free space into an aggregated virtual disk.

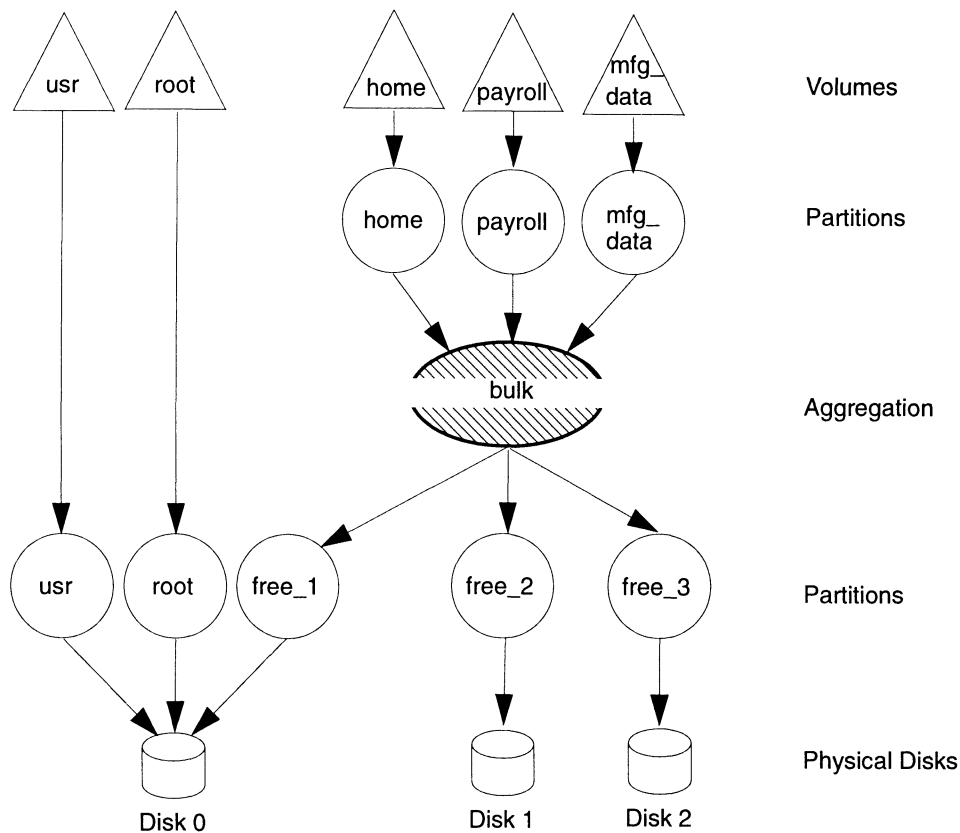


Figure 4 Aggregation of All Free Space into Bulk Virtual Disk

You use the following **sysadm** operations to aggregate free space:

```
Device -> Disk -> Virtual -> Create
New Virtual Disk Name: bulk
Striped? [no] yes
Stripe Size (in blocks): [16]
Select Space by: Creating a new partition
```

Note – The striping method you choose depends on your performance requirements. In this example, the default stripe size is selected.

```
Disk to Partition: sd(incr(0),0,0)
Length of Piece in Blocks: 150000
Do you want to specify more pieces for this virtualdisk? [no] yes
Select Space by: Name of an existing virtual disk
Child virtual disk: free_1
Do you want to specify more pieces for this virtual disk? [no] yes
```

```

Select space by: Name of an existing virtual disk
Child virtual disk: free_2
Do you want to specify more pieces for this virtual disk? [no] yes
Select Space by: Name of an existing virtual disk
Child virtual disk: free_3
Do you want to specify more pieces for this virtual disk? [no]

```

These operations form an aggregated virtual disk named **bulk** from three pre-existing partitions **free_1**, **free_2**, and **free_3**.

From this striped free space, you could carve out the virtual disk space necessary to accommodate your software needs, such as **home**, **payroll**, and **mfg_data**. The following **sysadm** operations illustrate how you might create a partition from an existing aggregation.

```

Device -> Disk -> Virtual -> Create
New Virtual Disk Name: payroll
Select Space by: Disk to partition and partition size
Disk to Partition From: bulk
Length of Piece in Blocks: 40000
Do you want to specify more pieces for this virtual disk? [no]

```

The partitioning of striped aggregations provides you with the immediate benefit of striping. LDM's limited hierarchy prevents such a structuring of striped free space.

Performance Improvements

While the virtual disk is on line and in use, OSM allows you to move virtual disks that are in contention for limited physical disks to less heavily loaded physical disks. With the **sar -d** command and the new **nsar** command, you can monitor your system's contention for a particular physical disk, specifically listing the average number of requests outstanding and being served at present. Knowing that a physical disk is approaching full utilization and that the average queue size (or length) is between 10-15 requests (or higher), you might shift some of the physical disk's burden elsewhere.

Suppose you're building a program using a compiler, an assembler, and a linker, and you observe that the build is taking far longer than usual. One file system you know to be heavily used is **/tmp**. So, after checking your physical disk layouts to determine where virtual disk **tmp** resides, you decide to move it to a less heavily used physical disk. With OSM, you can move the virtual disk while the file system remains on line and in use.

An OSM Performance Case Study

Let's compare an LDM configuration to an OSM configuration. Start with a 10-Gbyte CLARiiON™ disk array containing 20 disk drives to house your database. The goal for each configuration is to mirror the data and to unlink a mirror image for subsequent dumping to tape. After the dump completes, you then will want to relink the image to the mirror. The time required to resynchronize the data presents a problem. LDM's rigid logical disk hierarchy permits only one solution for grouping disk resources, which costs time. OSM's flexible hierarchy, on the other hand, gives you options for grouping your disk resources, which can save time. The remaining section outlines the procedures required for each configuration.

Note – When building this configuration with OSM, do not create virtual disks with the DG/UX file system structure. Many database products use raw disks (virtual disks without file systems) to bypass the overhead that is associated with file system maintenance. Refer to your database documentation for exact requirements.

In an LDM configuration, you create a two-image software mirror, with each image containing a ten-piece logical disk. Figure 5 shows this arrangement.

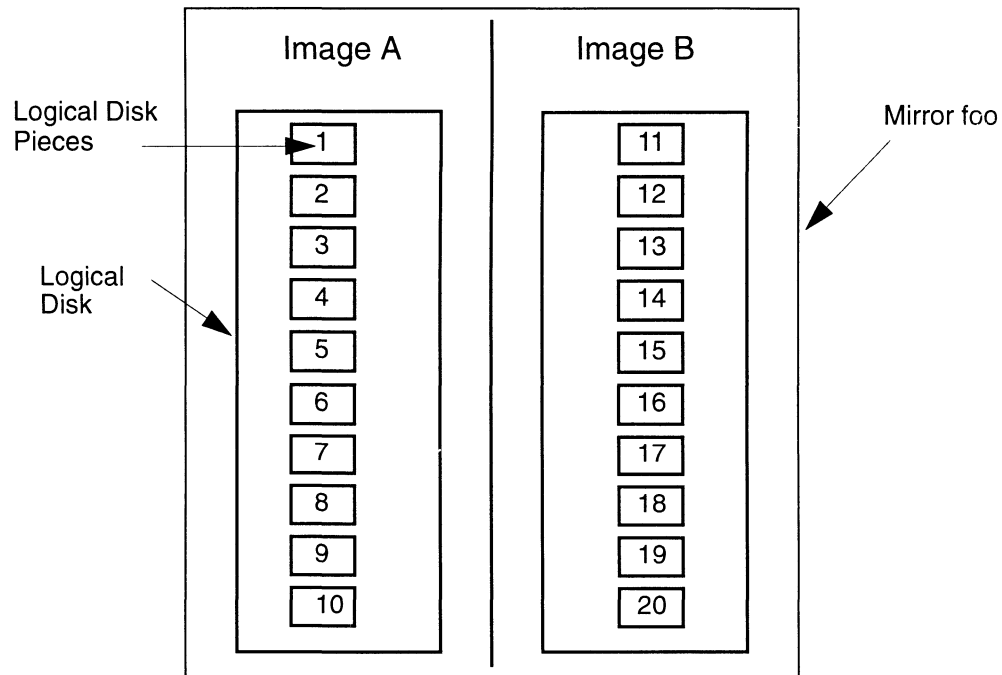


Figure 5 LDM Ten-Logical Disk Piece Two-Image Mirror

Later, deciding to back up image **B** to tape, you unlink image **B** from the mirror, access it as a separate logical disk, and dump the image to tape. After you complete the dump, you relink image **B** with mirror **foo** and resynchronize the data from image **A** to image **B**. Because the system recognizes each mirror image as a single logical unit, image **B** is synchronized as one sequential operation as shown in Figure 6.

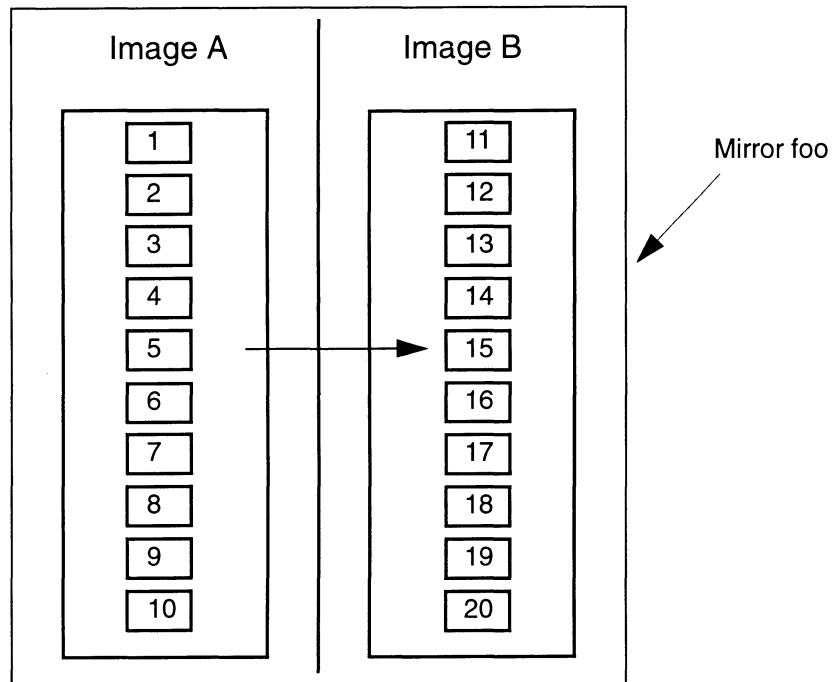


Figure 6 Sequential Synchronization of LDM Ten-Logical Disk Piece Ten-Image Mirror

The system reads logical disk piece 1 and writes to logical disk piece 11, then it reads piece 2 and writes to piece 12, and so on through pieces 10 and 20, which is a time-consuming operation. On a typical 10-Gbyte CLARiiON configuration, resynchronizing in this fashion can take up to ten hours.

OSM allows an alternate configuration that eliminates the bottleneck of a single, sequential synchronization operation. Synchronization speed depends on factors such as the number of disks, mirrors, and simultaneous mirror synchronizations that are ongoing in a configuration.

Note – Remember, when you build this configuration with OSM, do not create virtual disks with the DG/UX file system structure.

With OSM, you might create ten two-image mirrors, and then group all ten mirrors into an aggregation, as shown in Figure 7.

Note – Make sure that you build mirror images on separate SCSI buses.

With LDM you also could create ten two-image mirrors, but OSM lets you aggregate mirrors, which is an operation that was not allowed in LDM.

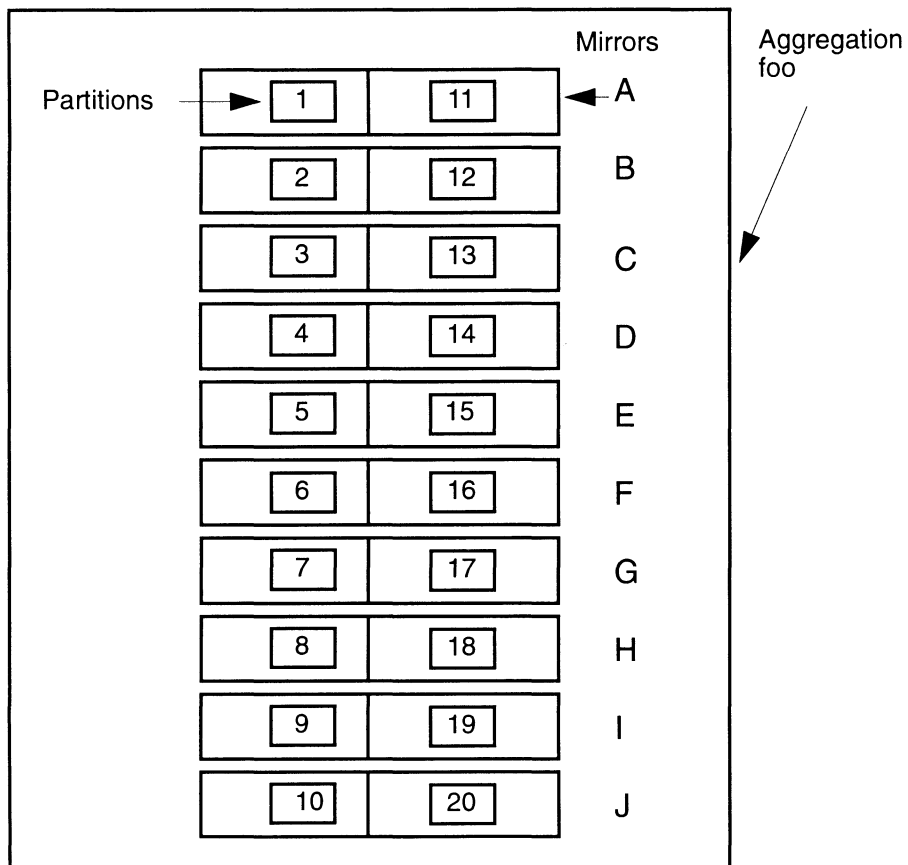


Figure 7 OSM Ten Two-Image Mirrors

To back up the mirrors' images to tape, you would temporarily unlink images 11 through 20 from their mirrors. To dump the contents of these aggregations as a single unit, using OSM, you can temporarily combine images 11 through 20 into an aggregation as shown in Figure 8. OSM's ability to aggregate mirrors makes this operation simple.

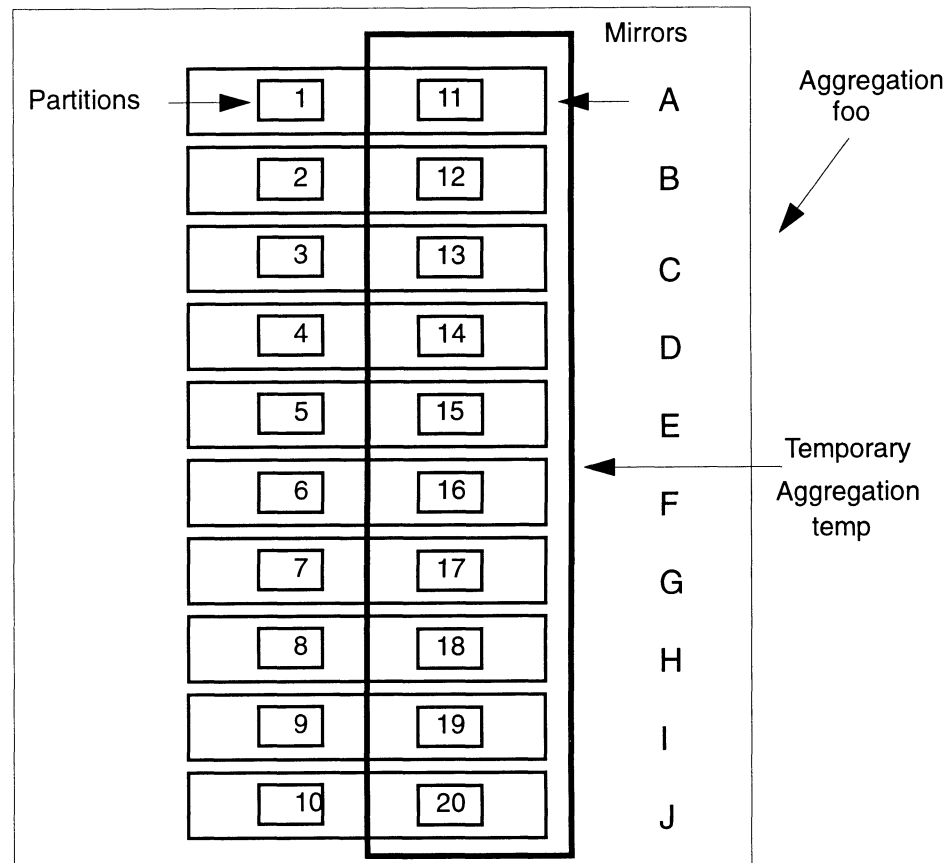


Figure 8 OSM Temporary Aggregation of Mirror Images

After the dump, delete aggregation **temp** since it is no longer needed, leaving images 11-20 intact as part of mirrors **A** through **J**. Then, relink these images to their respective mirror images. Finally, resynchronize partition image 11 with 1, 12 with 2, and so on through 20 with 10. Because all mirrors can be synchronized simultaneously, OSM synchronization potentially takes 1/10 the time required by LDM. While the system reads image 1 and

writes to image 11 on mirror **A**, it also can read image 2 and write to image 12 on mirror **B**, and so on through image 10 and 20 on mirror **J**, as shown in Figure 9.

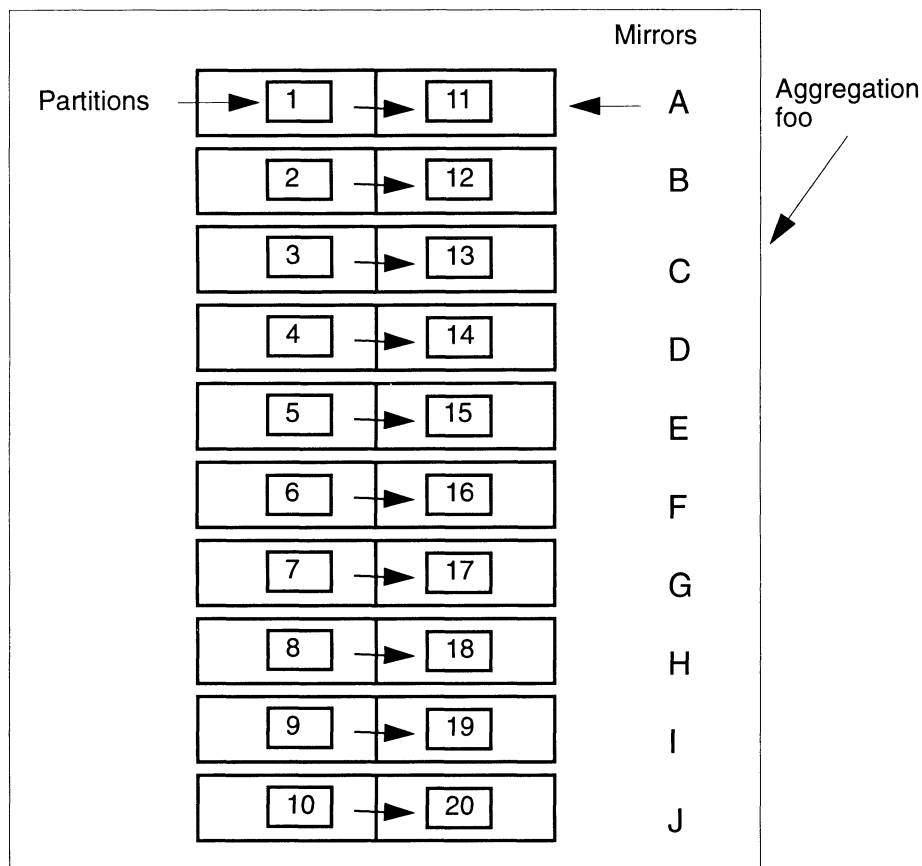


Figure 9 Simultaneous Synchronization of OSM Ten Two-Image Mirrors

OSM's hierarchical flexibility offers a potential time benefit, which is the reduction of mirror synchronization time in this example. You do not see a performance benefit in a single mirror's synchronization time. However, there are performance benefits for multiple mirror synchronizations because separate threads of control perform the mirror synchronization simultaneously. While there are numerous operations required for this benefit, you write the Bourne shell script once and execute it repeatedly. A script is preferable to the execution of sequential `sysadm` operations for the case study above.

The example script in Figure 10 assumes the existence of the basic configuration: an aggregation formed by ten two-image mirrors. Read the script's comments.

```

#!/bin/sh

# This script would be run to break apart the virtual disk
# hierarchy in the example, in order to back up one image to tape.
#
# This script presupposes the following:
# 20 partition virtual disks named disk_1 through disk_20, that
# correspond to the physical units in the CLARiiON.
# 10 mirror virtual disks named A through J, as shown in Figure 8.
# A has disk_1 and disk_11 as images;
# B has disk_2 and disk_12 as images; etc.
# An aggregation virtual disk named foo that has as its children
# virtual disks A through J, as shown in Figure 8.

# Unlink the second image from each mirror.
#
admvdisk -o unlink -c disk_11 A
admvdisk -o unlink -c disk_12 B
admvdisk -o unlink -c disk_13 C
admvdisk -o unlink -c disk_14 D
admvdisk -o unlink -c disk_15 E
admvdisk -o unlink -c disk_16 F
admvdisk -o unlink -c disk_17 G
admvdisk -o unlink -c disk_18 H
admvdisk -o unlink -c disk_19 I
admvdisk -o unlink -c disk_20 J
#
# Create a temporary aggregation of all the second images
admvdisk -o create -t aggregation \
-c disk_11 \
-c disk_12 \
-c disk_13 \
-c disk_14 \
-c disk_15 \
-c disk_16 \
-c disk_17 \
-c disk_18 \
-c disk_19 \
-c disk_20 \
temp

# Now we're ready to back up /dev/dsk/temp, using the backup
# facility of our choice.
# Insert the appropriate backup commands here.
# Remove the temporary aggregation.
admvdisk -o remove -q temp

# Relink the secondary images to their respective mirrors
admvdisk -o link -c disk_11 A
admvdisk -o link -c disk_12 B
admvdisk -o link -c disk_13 C
admvdisk -o link -c disk_14 D
admvdisk -o link -c disk_15 E
admvdisk -o link -c disk_16 F
admvdisk -o link -c disk_17 G
admvdisk -o link -c disk_18 H
admvdisk -o link -c disk_19 I
admvdisk -o link -c disk_20 J

# Initiate synchronization operations on all the mirrors
admvdisk -o sync A B C D E F G H I J

```

Figure 10 Script to Automate sysadm Operations

Cache Integration

OSM and LDM have the same performance goal when caching: to use a fast front-end device to buffer quickly read and write operations to a slower back-end device without risking data integrity. Let's review the basic idea behind caching before we talk about how OSM optimizes it.

How Caching Works

The primary caching configuration uses a nonvolatile random access memory (NVRAM) as the fast and stable front-end device and a physical disk functioning as the slow back-end device. Even though the RAM card has limited storage capacity, its fast I/O performance greatly improves the overall performance of applications whose network-driven I/O activities are characterized by synchronous transmissions that occur in bursts. Some database management systems are tuned for such I/O transmission rates. A disk, serving as the back-end device, has more storage capacity than does the front-end.

In response to an I/O request, the system attempts to write the data to the fast front-end device. If the front end is already full of data, it will flush one or more of its buffers to the back end to make room for the incoming data. You get optimal performance when the majority of the data being accessed can be kept on the front-end device.

As mentioned earlier, the caching principle is the same for LDM and OSM. So, why is OSM's implementation better? OSM's caching gives you continuous data access. Also, OSM supports the use of both a NVRAM card and a disk as front-end devices. LDM does not support the use of a disk as a front-end device.

NVRAM Front-End Device Sharing

The purpose of caching is to accelerate I/O performance. Assuming you have sufficient physical resources, you may choose to set up a caching configuration initially. But, in most instances, you'd probably prefer to demonstrate a need for caching first. For example, you'd want to decide how to best set up caching, and then run the application with and without caching and compare the results before committing to a caching configuration.

Suppose you have two file systems on a local AViiON server, which you later decide to export over the network. Figure 11 shows a simple configuration in which two partitions, **notes** and **home**, are mounted as exportable volumes.

With OSM, you can arrange for the file system to be cached, without taking that file's data off line. With LDM, on the other hand, you would have to take the file's data off line to set up the cache.

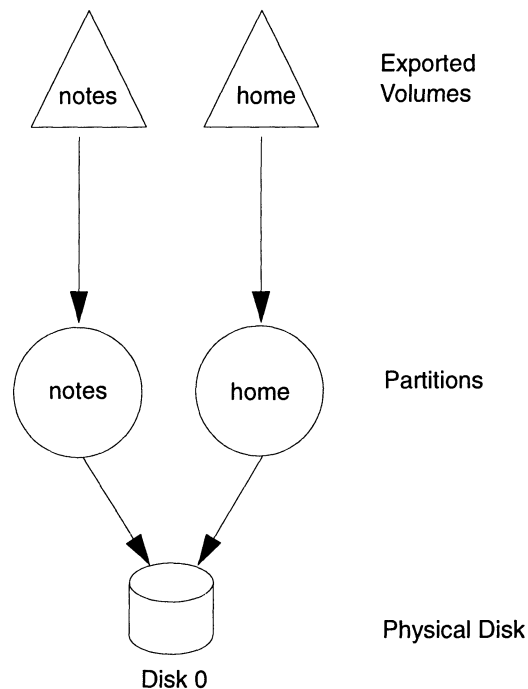


Figure 11 Two Partitions Mounted as Exportable Volumes

The drawback of exported file system I/O is its slow speed. One of the principal causes of slowness of remote-mounted file systems is delayed data write confirmations. To maintain data integrity, ONC™/NFS® does not acknowledge a successful write operation until the user data, inodes, and index blocks are actually written to the remote physical device.

Caching is one answer to reducing write confirmation delays. With normal operating system circumstances (e.g., sufficient CPU), the ability to dynamically insert a front-end device encourages you to experiment with your configuration without inconvenience due to down time. Through experimentation, you decide whether to cache both partitions or just one of them. With OSM, you can easily test all three possibilities to decide which one gives the best performance.

An OSM caching advantage is that multiple caches with different back-end devices can dynamically share the same front-end device. Device sharing eliminates potential front-end device idle time. An example of resource sharing is shown as follows:

To create the front-end virtual disk, **front**, perform the **sysadm** operation:

```
Device -> Disk -> Virtual -> Create
```

and identify the NVRAM card, **nvr(0)**, as the disk to partition.

To build a cache from pre-existing partitioned virtual disks, **notes** and **front**, select:

```
Device -> Disk -> Virtual -> Caches -> Cache
```

Sysadm then prompts for names of the cache, the back-end device, and the front-end device. By convention, the cache itself normally inherits the name of the virtual disk being cached (the virtual disk being cached is the back-end device). The cache's assumption of the back-end device's name leaves the back end nameless. You either may assign a new name to the back end or leave it nameless. Finally, you must name the front end. In this example, you assign the cache the back-end device's name, **notes**; the back end a new name, **notes_be**; and the front end, **notes_fe**. Only an excerpt of the essential **sysadm** dialog is provided below:

```
Virtual Disk: notes
Name for Cache: [notes]
New Name for Back End: notes_be
Specify a front end? [yes]
Front End Virtual Disk: front
Specify Another Front End? [yes] no
```

Repeat this **sysadm** operation to create another cache, this time specifying **home** and **front** as the back- and front-end devices.

Figure 12 on the next page shows the resulting front-end sharing.

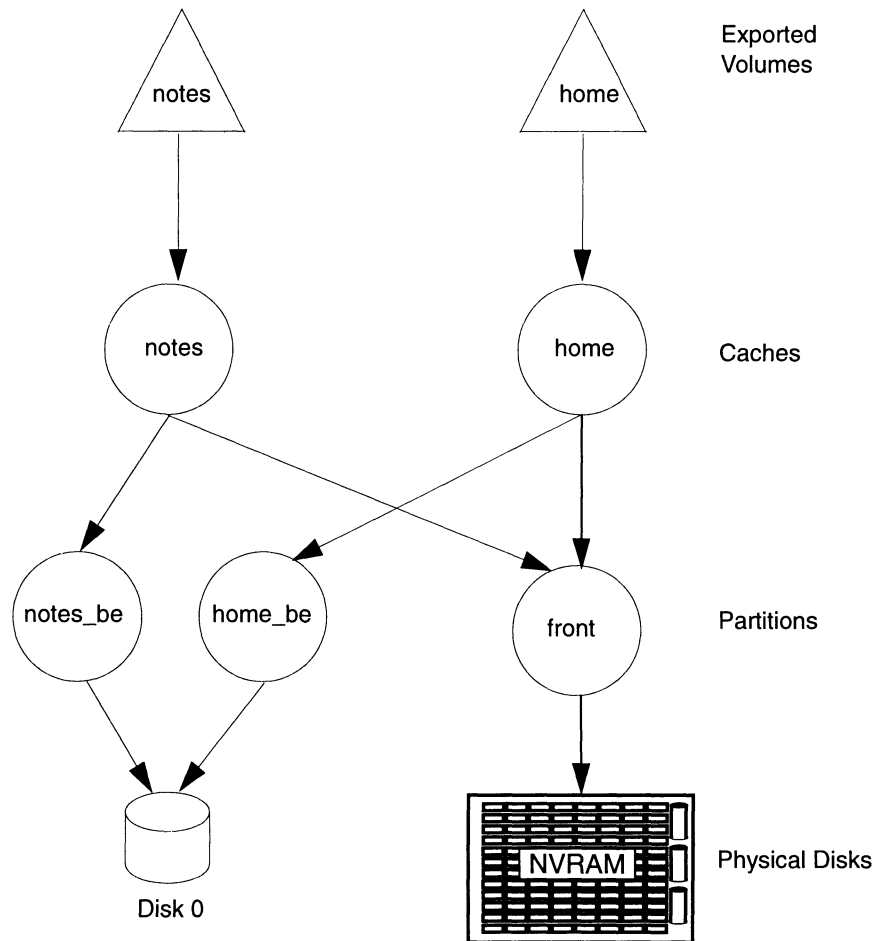


Figure 12 A Configuration with Caching

A NVRAM card that functions as a front-end device connected to a slower back-end disk (containing **notes_be** and **home_be**) eliminates most of the delay associated with writing. With the inode on the NVRAM card, you eliminate the need to keep writing to disk. The fast NVRAM front end allows ONC/NFS to return okay responses to remote disk write operations almost instantaneously. The cache writes the data from the NVRAM card to the slower back-end device later.

With OSM, you can tune a cached disk to optimize cache efficiency by using **sysadm**'s read and write weight parameters. For example, suppose that on a particular day, users on your system will perform a lot of product compilations in their home directories at their workstations. On that day, you may choose to cache only the **home** file system or you may want to increase the read and write weight of the cache front end toward the **home**

file system. This adjustment causes the blocks in the cache that automatically are allocated to the **home** file system to be retained longer than those allocated to the **notes** file system. Increasing read and write weight for a heavily accessed file system may improve compilation time.

Fast Front-End Disk Device

A fast disk, as well as a NVRAM card, can act as front-end devices for a slower, larger storage device. Figure 13 shows such a configuration.

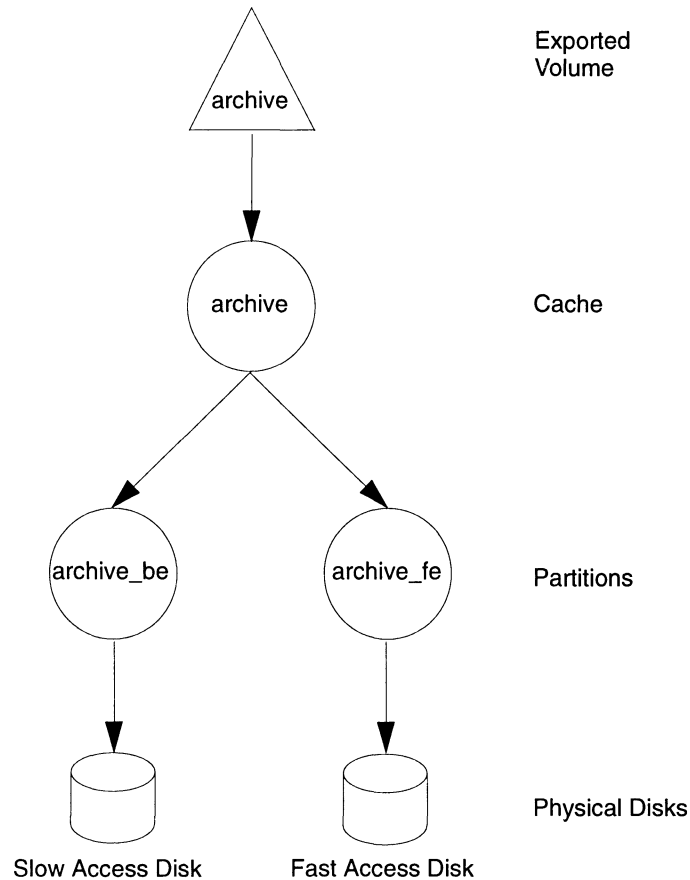


Figure 13 Using a Fast Disk as a Front-End Cache Device

In this example, a slow access device (such as a magneto-optical device) is used for archival storage. By creating a cache front-end partition, **archive_fe**, from the fast disk and associating it with the slow back end, **archive_be**, writes to the archival disk will occur at the rate of the fast disk. The speed at which data is read from the slow disk also increases if there are multiple reads of the same data with close proximity. Once the data has

been read from the slow device and is in the cache's buffer, subsequent reads and writes of the same data occur at the speed of the fast front-end device.

Again, only with OSM can you decide at any time to reconfigure disk resources for performance gains at no availability cost. Furthermore, such a configuration is impossible with LDM because you cannot use disks as front-end devices.

Other Advantages of OSM

OSM also introduces these enhancements to the DG/UX operating system environment:

- ❑ New **rename** and **move** virtual disk operations and improved **copy** and **expand** virtual disk operations.
- ❑ The ability to boot from an aggregation virtual disk.

Rename operation

Under LDM, you improvise a rename operation by executing a lengthy set of commands in off-line mode. With OSM, while the target virtual disk is on line, you perform the virtual disk rename operation with one simple command.

Note – The new virtual disk name is not transmitted automatically to the `/etc/fstab` file. You must specify a different mount point in the `fstab` file using the `sysadm` operation `File System -> Local Filesys -> Modify`.

Move operation

A true **move** mechanism doesn't exist in LDM. You improvise a **move** operation by executing several commands, which produce a destination logical disk with the correct content but with a different logical disk name. To correct the logical disk's name, you then execute a lengthy set of commands necessary to rename the logical disk. Again, these steps require the logical disk to be taken off line.

With OSM, with the source virtual disk on line, in one step, you can move the contents of the source disk to the destination disk. The destination disk takes the source disk's name and role. Following a successful move, the source virtual disk is deleted. See Figure 3 for a summary of the move operation.

Copy and expand disk operations

The copy and expand disk operations of OSM and LDM are basically the same. However, the difference is that with LDM, the source logical disk must be unmounted, which takes the data off line. By contrast, with OSM you can expand full file systems while they are still mounted and in use.

Note – You cannot enlarge or shrink striped virtual disks.

Booting from aggregated virtual disk

With LDM, you are restricted to booting from single-pieced logical disks. This restriction is a particular hardship when, during upgrades, expansion of the **root** or **usr** logical disks is necessary. If adjacent space on the physical disk is unavailable for expansion, you must locate sufficient space elsewhere on the physical disk and copy the contents of the logical disk there. With the logical disk at its new location, you then can expand the logical disk as necessary.

By contrast, OSM allows you to boot from its multi-pieced logical disk equivalent—the aggregated virtual disk. All partitions forming the aggregation, however, must be located on the same physical disk.

Note – Do not install the DG/UX operating system (**root**, **usr**, or **swap**) on a bulk aggregation and attempt to boot from it.

Making the Transition to OSM from LDM

When upgrading to DG/UX 5.4R3.00, with your permission, all physical disks in your configuration that can be read, written, and are in LDM format will have their metadata formats changed automatically to virtual disk format. Metadata describes the layout of the physical disk and how it is partitioned. The process of converting a physical disk's metadata from logical to virtual disk format takes less than a second per disk and does not touch the data contained in the logical disk or file system. All you have to do is press the Enter key to start the metadata conversion process.

For the easiest transition from LDM to OSM, you are strongly encouraged to convert all logical disks during the upgrade procedure. Your configurations, including mirrors and striped logical disks, will be preserved during the conversion.

Note – If you have a cache, delete it before you shut down your system to perform the upgrade. The data being cached will remain intact. By deleting the cache, you are dissolving the relationship between the front and back ends. Immediately following the upgrade, you may rebuild the cache quite easily.

After you complete the upgrade and as you gain familiarity with the new OSM terminology and functions, you may experiment with your virtual disk arrangements while on line and with confidence that you will not destroy or alter permanently your disk configurations.

Why Did the Metadata Format Change?

The metadata format changed for increased flexibility. A new format was developed to accommodate the requirements of virtual disks that the old format couldn't handle. In addition, the new metadata format is sufficiently flexible to support future needs.

Upgrading to DG/UX 5.4R3.00

OSM provides two options for making the transition to the OSM technology:

- ❑ One-time operation to convert all physical disks from a logical disk format to a virtual disk format
- ❑ Continued use of logical disks in a virtual disk environment (compatibility mode)

In most cases, you'll want to convert all logical disks to virtual disks. Physical disks containing logical disks **root**, **usr**, and **swap** *must* be converted to virtual disk format; they cannot be used in compatibility mode.

After a device is registered in compatibility mode, you can read and write its data, but you cannot alter the logical disk in any way with operations that expand, delete, or re-create the disk's metadata. Logical disks on CD-ROM devices cannot be converted to virtual disk format, so they must be used in compatibility mode. Operating in compatibility mode neither affects your access to those files or data nor reduces the performance of those file systems.

Since the virtual disk metadata format is slightly smaller than that of the logical disk format, there is no risk of format conversion failure because of limited disk space.

The conversion occurs during the “Prepare Physical Disks” upgrade step. To convert all physical disks, accept the default responses to the following prompts:

```
Prepare physical disks

Run this step now? [yes]

Convert these logical disk formatted physical drives to virtual
disk format:

sd(isc(0),0,0)
sd(isc(0),3,0)
sd(isc(0),5,0)
```

```
Physical disks(s) to convert: [all]
```

Accepting the default converts all physical disks. To convert only a subset of all physical disks, you must specify them explicitly. Recall, however, that you must convert physical disks that contain the operating system—**root**, **usr**, and **swap** logical disks.

The conversion utility acts on all physical disks at once because it must locate all logical disk pieces across physical disks and convert them to their virtual disk equivalents. This single conversion process ensures that the logical disks are mounted in the correct file system location when the system first boots.

The conversion utility converts only the disk metadata format; it does not touch the user data. The utility converts the metadata only after it ensures that it is safe to do so—without risk of data loss or corruption. Even if your system crashes during a conversion, the data is still safe. After you reboot the system, you can restart the conversion.

If, after operating in compatibility mode for a while, you decide to convert selected physical disks to virtual disk format, you may do so using the **sysadm** operation `Device -> Disk -> Physical -> Convert`.

Caution – After you convert from LDM to OSM, do not boot a pre-DG/UX 5.4R3.00 kernel or stand-alone **diskman**. If you accidentally boot one of them, the pre-DG/UX 5.4R3.00 kernel will attempt to repair what it believes to be a damaged LDM-formatted physical disk. An attempted repair operation destroys the primary virtual disk information table (VDIT), which restricts the ways in which you can access data. For example, you can continue to read and write virtual disks but you can neither create new ones nor change the attributes of existing ones.

To recover, shut down the system and boot a DG/UX 5.4R3.00 kernel from tape or disk. Execute the **sysadm** operation `Device -> Disk -> Physical -> Repair`, which copies the secondary VDIT from the rear of

the physical disk to the front, restoring the corrupted primary VDIT. After the VDIT is restored, you may register the disk (Device -> Disk -> Physical -> Register) and access it, as desired.

Can I Revert to LDM Format?

You are strongly encouraged to convert all physical disks to virtual disk format. However, any physical disk that you converted from LDM to OSM can be reverted to LDM format. Original virtual disk-formatted physical disks cannot be converted to LDM format because LDM format takes more space than does OSM format.

Some OSM hierarchical constructs may have no LDM equivalent. For example, conversion of an aggregated virtual disk composed of other aggregations to the logical disk equivalent does not work. If a backward conversion fails, the conversion utility leaves your virtual disks intact. You may continue to use virtual disks in an LDM environment only if you rearrange virtual disks in the simple hierarchy supported in LDM. For example, you may continue to use simple partitions, aggregations, and mirrors; but, you cannot use higher order configurations such as aggregated aggregations or aggregated mirrors.

To revert to an earlier DG/UX release, you must first convert all physical disks in virtual disk format to logical disk format. You can perform the conversion using the **sysadm** operation Device -> Disk -> Physical -> Convert.

If these devices were originally formatted in virtual disk format, you must reformat them as logical disks using an earlier release of the DG/UX system. You must reload the **root**, **usr**, and other optional logical disks, such as **usr_opt_X11**, at a minimum because they contain the release specific files.

Back up your data before formatting your physical disks. The data on the physical disks will be lost when reformatting. After you reformat the physical disks, copy the backed up data onto the reformatted logical disks.

Summary of OSM Advantages

In sum, the following are the advantages that OSM offers you:

- On-line backups
- On-line data storage failure recovery
- Flexible data storage management
- Performance improvements
- Cache integration
- New **rename** and **move** operations and improved **copy** and **expand** operations
- Ability to boot from an aggregated virtual disk
- Extensible metadata format to enhance disk capabilities

Questions and Answers

Q: What is stand-alone **sysadm**? How does it relate to **diskman**? Does it break any existing functionality?

A: Stand-alone **sysadm** replaces stand-alone **diskman**. It provides all the **diskman** capabilities, but through the **sysadm** interface. Stand-alone **sysadm** is the same as stand-among **sysadm** except that it contains a subset of the key commands to support system installation. The new stand-alone **sysadm** installation sequence resembles the CD-ROM installation rather than the former stand-alone **diskman**-based tape installation.

Conceivably, the substitution of **sysadm** for **diskman** could require modification of your installation scripts. In general, however, we do not expect you to experience “broken” functionality. We believe that using a consistent menu-driven interface for both stand-alone and stand-among **sysadm** operations will make DG/UX system administration easier.

Q: Why did the LDM **admmdisk** and **admdisk** commands go away? What were they replaced with? How does this affect me?

A: Because OSM entirely replaces its predecessor, LDM, Data General replaced the LDM **admmdisk** and **admdisk** commands with a new OSM **admvdisk** command. You must edit scripts and substitute the **admdisk** and **admmdisk** commands and arguments with the appropriate **admvdisk** command and argument equivalents. Additionally, the **admpdisk** command has changed to reflect OSM.

Q: How does the new disk format affect CD-ROM devices containing either the DG/UX system or an add-on package?

A: Starting with DG/UX 5.4R3.00, the DG/UX system distribution CD-ROM will be in virtual disk format suitable for booting during installation.

The DG/UX 5.4R3.00 CD-ROM release media does not contain alternative releases of the operating system as do former CD-ROM release media.

Add-on software on CD-ROM media supplied by Data General is in logical disk format. Both the **sysadm** and **admpdisk** command support disk registration in compatibility mode, which lets you read data but not alter the metadata on the device. To register a disk in this fashion, you invoke compatibility mode through the **sysadm** or **admpdisk** command.

Vendors who use a DG/UX 5.4R3.00 system will develop add-on software in virtual disk format. Products to support pre-R3.00 DG/UX releases should be developed on pre-R3.00 DG/UX systems.

Q: When I upgrade to DG/UX 5.4R3.00, do I retain the same logical disk names; e.g., **root**, **usr**, and **swap**?

A: Yes. The physical disk's metadata keeps the disk's layout in virtual disk format. When your AViiON system boots, it automatically will register all virtual disk formatted media and then read the virtual disk layout from the physical disks. The virtual disk manager will read this information to create the correct volumes needed by the **mount** command to mount the virtual disks' file systems.

During the upgrade, the installation utility will find and offer to convert all logical disks to virtual disks, retaining the exact logical disk structure, but in virtual disk format. This allows your system to boot and automatically mount the correct file systems after upgrading from DG/UX 5.4.2, DG/UX 5.4R2.01, or DG/UX 5.4R2.10 system to DG/UX 5.4R3.00.

Q: Why not support both LDM and OSM? What do I get with one that I cannot get with the other?

A: Data General is not supporting both technologies because OSM adds sophistication, efficiency, and flexibility to what is already offered by LDM. With OSM, you can manipulate virtual disks without having to take your data off line. The on-line feature offers high data availability at no risk of data loss. With OSM's flexible hierarchical structure, you can dynamically modify disk allocations to improve your system's performance. OSM enables the system administrator to react to changing space needs without having to shut down the system, take data off line, or map out disk space allocations in advance of need. Data General considers OSM to be a cornerstone in the high-availability arena and a strong basis for future product innovation.

Q: If I have applications that use an LDM **ioctl** interface for doing raw I/O on a device, what happens when I switch to OSM?

You can perform raw I/O operations on devices using the kernel's virtual disk manager. However, you may need to modify applications that use kernel **ioctl** calls because the programming interface at this level has changed.

Fortunately, most applications perform raw I/O with the raw physical devices. In most cases, applications perform I/O to the **/dev/dsk** and **/dev/pdsk** directories and will not require any modifications.

If your application references a virtual disk that requires a long virtual disk name rather than a short one, however, you may need to modify the application. All virtual disks have a long name (such as **usr_opt_X11,0CDE6D03,DEF4236A,0**). As a convenience, however, at disk creation time, a virtual disk has a shortened name (such as **usr_opt_X11**) as well as a long name entered in **/dev/dsk** and **/dev/pdsk**. During disk registration of identically named devices, a race condition occurs in which the first one registered will get the shortened name. Both devices, of course, have unique long names.

Q: How do I administer OSM without **diskman** and the methods I know in **sysadm**?

A: You can perform OSM operations by using stand-alone **sysadm** (the replacement for **diskman**), stand-among **sysadm** (invoked from the shell), or the **admpdisk** and **admvdisk** commands. The **admfilesystem** and **admpdisk** commands are still present but have been enhanced. The advantage of the new stand-alone **sysadm** and its shell commands is that now you can use **sysadm** to both install and manage the system. The **admvdisk** command combines the separate functions of **admldisk** and **admmdisk** into a single administrative command while supporting the new capabilities of OSM.

Q: Does the Minimum Equipment Configuration (MEC) for AViiON systems change as a result of moving to OSM?

A: The MEC will increase with DG/UX 5.4R3.00, but not because of OSM. The new MEC is 16 Mbytes of main memory, which is needed to support the features of stand-alone **sysadm**. DG/UX R3.00 supports these installation scenarios: install from tape, CD-ROM, remote tape, and remote file system. Default sizes for the operating system packages will remain the same as in previous releases. However, this release includes an additional package, **X.desktop**, as a replacement for the package Looking Glass, which was discontinued starting at DG/UX 5.4R2.10. Looking Glass required 20,000 blocks; **X.desktop** requires 50,000 blocks.

Q: Does OSM have an effect on operator-initiated or machine-initiated failover?

A: Yes. Each system that is set up in a failover configuration must be running DG/UX 5.4R3.00.

Related Documentation

Refer to the following customer documents for complete information on creating and using virtual disks and monitoring performance.

Installing the DG/UX™ System

Customizing the DG/UX™ System

Managing the DG/UX™ System

The CLARiiON™ Disk-Array Storage System with the DG/UX™ Operating System

Using the AV SysScope™ Performance Monitor

OSM and LDM Terminology

AV SysScope™—Set of performance measurement tools for AViiON computers running the DG/UX operating system. The tools include **sscope**, **sscope-ps**, **sscope-log**, and **sscope-collect**. Information on this tool is located in *Using the AV SysScope™ Performance Monitor*.

Aggregation—A type of virtual disk. OSM equivalent of multi-pieced logical disk under LDM. Combination of virtual disks of any type to form a higher-level virtual disk whose total size is the sum of the sizes of its constituents. For example, an aggregation is formed by grouping multiple partitions, aggregations, mirrors, or any combination of same virtual disk types. An aggregation can comprise up to 120 partitions on one or more physical disks. An aggregation can be striped as long as each child component is the same size.

Back-end device—Used in a caching configuration in which a virtual disk containing data located on a large, slow device (typically a slow disk) is associated with a virtual disk on a small, fast device (typically a NVRAM card or a fast disk).

Cache—A type of virtual disk. Association of two or more data storage devices, typically a small, fast one with a large, slow one, so that an application uses the fast device for read and write operations while the operating system duplicates these operations on the larger device. Caching offers the benefit of speed from the fast device and capacity from the large device.

Compatibility mode—Way in which logical disks on physical disks remain in operation in a DG/UX 5.4R3.00 configuration. Such LDM-formatted physical devices are not converted to OSM format. After a device is registered in compatibility mode, its data can be read and written but not altered with operations that expand, delete, or re-create the disk's metadata. Logical disks on CD-ROM devices cannot be converted to OSM format, so they must be used in compatibility mode.

File system—Software-formatted portion of disk space located on a logical disk or virtual disk. The file system contains the internal data structures that the operating system requires to keep track of files and directories. A virtual disk, which may be a volume, must contain a file system before it can be mounted. File systems, which are listed in `/etc/fstab`, are mounted at system boot time.

Front-end device—Used in a caching configuration in which a virtual disk located on a small, fast device (typically a NVRAM card or a fast disk) is associated with another virtual disk on a large, slow back-end device (typically a slow disk).

LDM—Logical disk manager. The predecessor to OSM, LDM allows access and manipulation of logical disks built on physical disks. LDM takes logical disks off line when changing disk storage options, rendering the data unusable for the duration of the operation.

Logical Disk—In LDM, a software abstraction, enabling the DG/UX system to manage an area of a specified size on physical disks. A logical disk can comprise multiple pieces spanning up to 32 noncontiguous areas on one or more physical disks.

Mirror—A type of virtual disk. A collection of identically-sized virtual disks (in OSM) or logical disks (in LDM) that are duplicates (copies) of each other. In OSM, multi-image mirrors (three maximum) can be created from partitions, aggregations, other mirrors, caches, or any combination of the same virtual disk types. In LDM, only logical disks can be mirrored.

Mount Point—The directory location of a logical or virtual disk, which has been overlaid with a file system structure. A mounted file system allows it to be user-accessible. At boot time, the system mounts directories in `/etc/fstab`.

OSM—On-line Storage Management. A software technology allowing the access and manipulation of software virtual disks on physical disks while the data is on line and in use. Also, it offers a flexible hierarchical structure wherein existing resource configurations can be regrouped into higher-level configurations at no performance or availability cost.

Partition—A type of virtual disk. OSM equivalent of the logical disk piece, it is the specific, contiguous space reserved on a physical disk for later data storage. You can create a partition from a physical disk or partition from an existing virtual disk. The difference between a partition and a logical disk piece is that a partition can be mounted; a logical disk piece cannot.

Physical disk—Hardware medium on which logical disks (or virtual disks) are mapped and accessed.

Striping—Storage of sequential file elements on alternating physical disks. This storage attribute enables the interleaving of sequential disk accesses, taking advantage of the disk hardware's read-ahead feature to speed disk I/O. Striping requires that each virtual disk be the same size. A striped virtual disk cannot be expanded or shrunk. Striping is available only with aggregated virtual disks.

Virtual Disk—OSM counterpart to the LDM logical disk, which is an amount of space that you reserve on a physical disk onto which data is stored. A virtual disk is further characterized by a type: partition, aggregation, mirror, cache, or combination of same virtual disk types.

Volume—A virtual disk for which entries are created in `/dev/dsk` or `/dev/rdisk`. These entries are needed to mount or read and write the virtual disk. Any virtual disk can be declared a volume and therefore, be directly accessible. By contrast, LDM logical disk pieces cannot be independently mounted.

