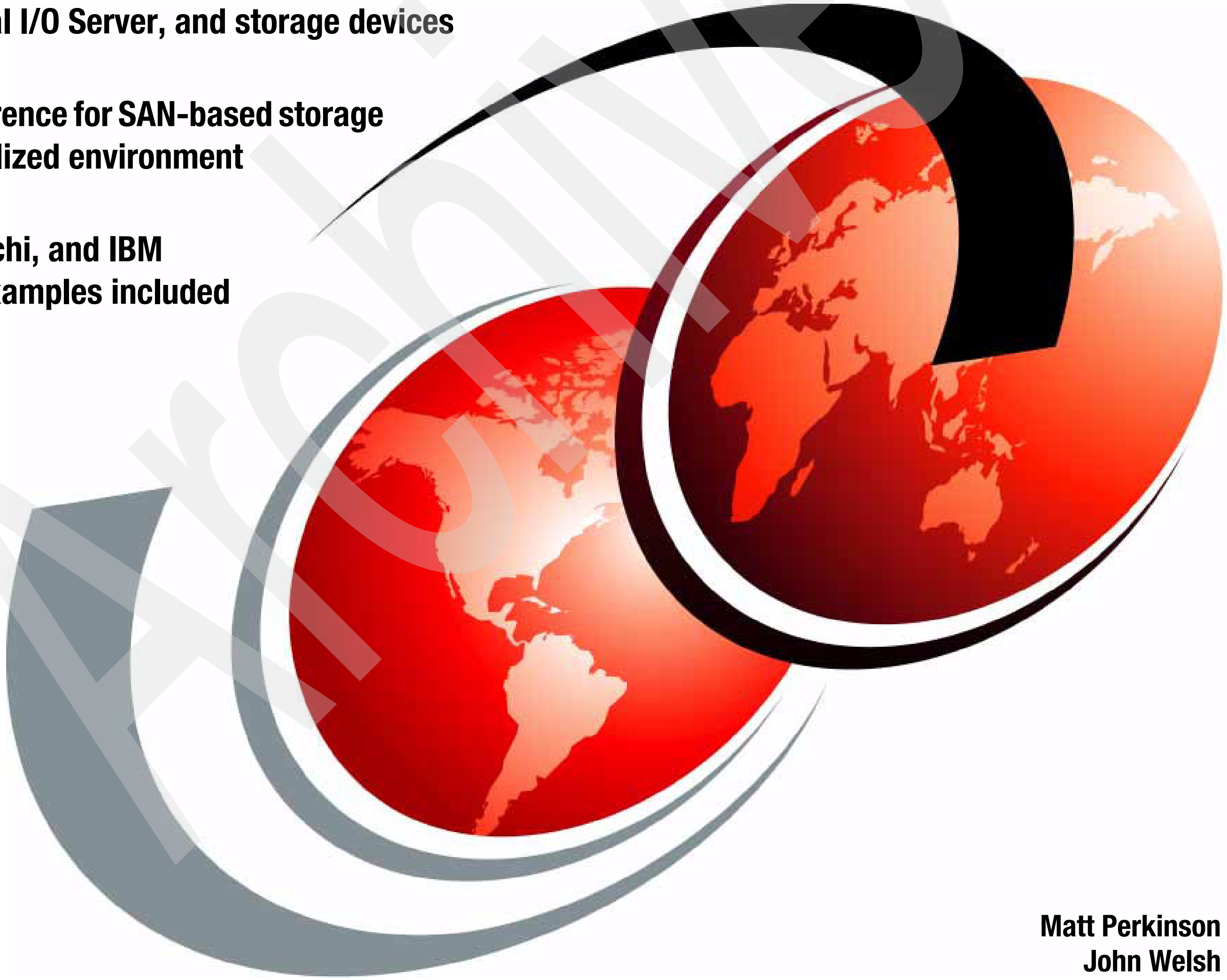
PowerVM and SAN Copy Services

- Real-world data center scenarios with procedures for AIX, Virtual I/O Server, and storage devices
- A key reference for SAN-based storage in a virtualized environment
- EMC, Hitachi, and IBM storage examples included



Matt Perkinson
John Welsh

Redpaper

ibm.com/redbooks

International Technical Support Organization

PowerVM and SAN Copy Services

January 2010

REDP-4610-00

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (January 2010)

This edition applies to Virtual I/O Server V2.1.2.0, AIX V6.1, and HMC V7.3.4-SP2.

This edition applies to EMC Solutions Enabler V7.0.0.0 and DMX-4-24 V5773.134.94.

This edition applies to Hitachi Command Control Interface V01-23-03/06 and USP V60-04-15-00/00.

This edition applies to IBM DSCLI V5.4.1.44 and DS8300 V5.4.1.44.

This edition applies to IBM SVC 4V.2.0.4.

This edition applies to IBM SAN32B V6.1.1d.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those 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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

The following terms are trademarks of other companies:

AIX®
FlashCopy®
IBM®
POWER6®
PowerVM™
POWER®
Redbooks®
Redpapers™
Redbooks (logo) ®
System Storage™

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redpapers™ publication is a guide to Copy Services as managed by AIX® in a IBM PowerVM™ virtualized environment. The goal of this paper is to provide step-by-step procedures about how storage subsystem-based Copy Services are accomplished in a virtualized environment and the approach for which they should be adopted.

Note: The term “Copy Services” in this publication is used as a generic term to describe the data replication and duplication functions that EMC TimeFinder, Hitachi ShadowImage, and IBM FlashCopy® provide.

This paper focuses on Copy Services technology from EMC, Hitachi, and IBM within the following storage subsystem models:

- EMC DMX-4
- Hitachi USP V
- IBM DS8300
- IBM SVC

The storage software products that will be used for the different Copy Services are as follows:

- EMC TimeFinder
- Hitachi ShadowImage
- IBM FlashCopy

While there are numerous storage subsystem vendors in the marketplace and industry, this paper has been limited to the vendors and the technologies described above.

The methodology and approach detailed within this document are consistent for all of the Copy Services functions for all of the storage subsystem replication tools. Where the methods differ is only in the syntax and storage software tool being used to complete the specific scenario.

We describe three scenarios in this paper that are considered the most common tasks for Copy Services that system and storage administrators would perform in their day-to-day responsibilities. Each scenario details the procedure required to be undertaken in order for the Copy Services to be completed per the scenario description.

These scenarios are designed to provide the resiliency of all active logical partitions within the PowerVM system. The objective is to prove that the storage subsystem-based replication tools are several abstraction layers from the Virtual I/O Server operating system and have no effect on the Virtual I/O Server operations and the virtualized client stability if the activities are performed as documented.

This paper is intended for use by the following audience:

- System/Storage Administrators

  These individuals may need to perform the activities documented is this paper for the purposes of backup, migration, or consolidation activities in day-to-day Virtual I/O Server or storage subsystem management.

- Architects/Solution Designers

  These individuals may need to understand the exacting details of block-level data replication and design an approach to complete the activities documented in this paper.

The following skill levels are encouraged in order to perform the tasks documented in this paper:

- Intermediate Virtual I/O Server management
- Intermediate Copy Services for EMC, Hitachi, and IBM storage subsystems
- Basic storage management and provisioning
- Intermediate AIX operating system administration
- Intermediate AIX Logical Volume Manager (LVM) administration

The team who wrote this paper

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Matt Perkinson is an Executive IT Consultant within the IBM Worldwide Systems Services Center of Excellence in the United States. He has over 14 years of experience in enterprise storage and systems architecture and administration. His areas of expertise include enterprise transformation solutions, high-availability clusters, storage management, SAN/NAS solutions, networking, hardware, client/server technology, disaster recovery solutions, network operating systems, and high-level programming languages.

John Welsh is an Enterprise Storage Architect in Australia. He has 12 years of experience in systems and storage. He started his career at Digital Equipment Corporation in 1997 and eventually worked for Hewlett-Packard for 10 years. He has worked at IBM for 2 years. His areas of expertise include disk and tape technologies, high-availability solutions, and design and architecture principles.

The project that produced this publication was managed by:
Scott Vetter, PMP

Thanks to the following people for their contributions to this project:

Garius Bias, David Bennin, Richard Conway, Denise Genty, Robert Haimowitz, Robert Kovacs, Kam Lee, Marcia Lehmann, Jim Pafumi, Vasu Vallabhaneni, James Younghaus
IBM U.S.A.

Kevin Richards
EMC

Chuck Coleman
Hitachi Data Systems

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks® publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Technical overview

This chapter provides details regarding the following topics:

- IBM PowerVM architecture overview
- Lab environment
- Copy Services scenarios

1.1 IBM PowerVM architecture overview

IBM PowerVM technology provides server virtualization on IBM POWER® processor-based systems. Each IBM PowerVM client is allocated a percentage of physical processor, memory, and I/O resources on which an AIX, IBM i, or Linux® operating system can be installed. PowerVM virtualization manages the resources and provides the ability to dynamically change the percentage of allocated resources depending on the disparate requirements of each client. When combined with the Virtual I/O Server, connected storage and shared network resources can also be virtualized.

Figure 1-1 provides an overview of the components within an IBM Power System and how each is allocated to a client.

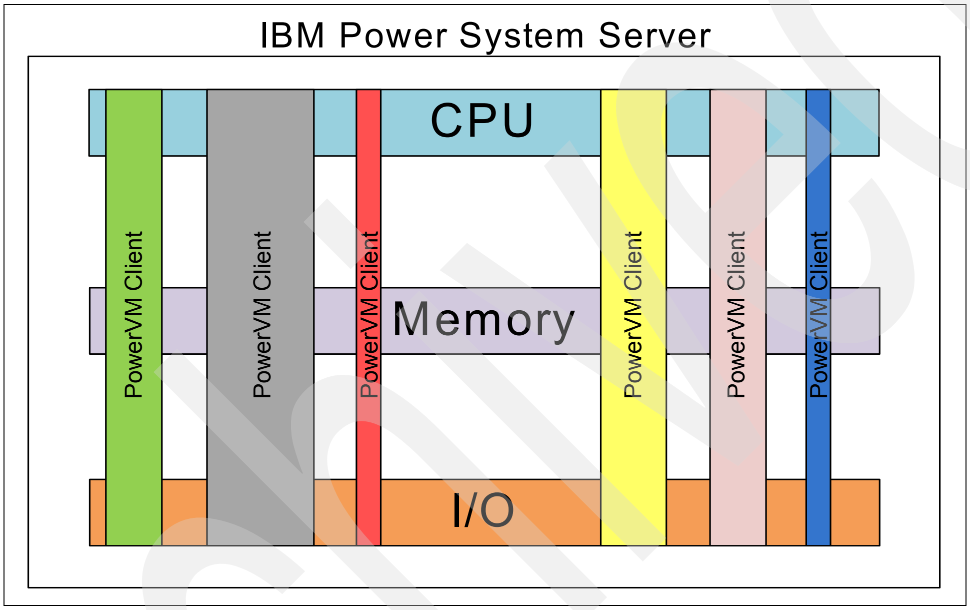


Figure 1-1 Virtual I/O Server client architecture

The Virtual I/O Server allows clients to share physical disk and network adapter resources. This capability allows an IBM Power System server to support more clients than the quantity of physical I/O slots that are available within the system. The clients can access the physical Fibre Channel or SCSI controllers managed by the Virtual I/O Server through the presentation of virtual SCSI adapters. For shared network resources, the clients can access the physical Ethernet adapters in the Virtual I/O Server through an defined virtual Ethernet adapter.

1.2 Lab environment

The following tables contain the technical details and specifications of the lab environment on which the various scenarios were performed.

Table 1-1 provides the configuration of the Virtual I/O Server and the supporting clients used in the scenarios.

Table 1-1 Virtual I/O Server specifications

# of CPUs	CPU type	CPU speed	Memory	HBA	Operating system
8	POWER6®	4.2 GHz	8 GB	4 x 4 Gbps	ioslevel 2.1.2.0
	Virtual I/O Server clients				
	Client name	**Operating system**	**# of CPUs**	**Memory**	
	client1	AIX V6.1	0.2	1 GB	
	client2	AIX V6.1	0.2	1 GB	
	client3	AIX V6.1	0.2	1 GB	

Table 1-2 provides the technical specifications regarding the various storage subsystem platforms that were used in the scenarios.

Table 1-2 Storage subsystem specifications

Vendor	Model	Cache	Host adapters	Microcode	# of disks
EMC	DMX-4 24	64 GB	56	5773.134.94	112 x 146 GB 15k Fibre Channel
Hitachi	USP V	16 GB	32	60-04-15-00/00	32 x 146 GB 15k Fibre Channel
IBM	DS8300	32 GB	32	5.4.1.44	32 x 146 GB 15k Fibre Channel
IBM	SAN Volume Controller (8G4)	16 GB	8	4.2.0.4	N/A

Table 1-3 lists the Copy Services software products that were used for the scenarios.

Table 1-3 Copy Services software

Vendor	Product	Version
EMC	Solutions Enabler - TimeFinder	7.0.0.0
Hitachi	CCI - ShadowImage	01-23-03/06
IBM (DS8300)	DSCLI - FlashCopy	5.4.1.44
IBM (SVC)	Native - FlashCopy	4.2.0.4

In the following list, some of the miscellaneous pieces of the infrastructure are documented. While not directly responsible in the execution of the scenarios, the items below are critical in supporting the overall environment. These items are:

- One dedicated SCSI Virtual I/O Server client to act as a main storage subsystem management server. It has the following attributes:
  - Operating system: AIX V6.1
  - Installed storage management software: EMC Solutions Enabler, Hitachi Command Control Interface, and IBM DSCLI
- One IBM 7310CR4 Hardware Management Console
  - Build Level: V7R3.4.0.2
- Two IBM SAN32B SAN switches
  - Firmware: 6.1.1d

1.3 Scenario overview

The three scenarios detailed in this section are typical of most functions that storage and system administrators would use for local storage subsystem based Copy Services.

Backup, migration, and server consolidation can be challenging processes to complete in today's complex environments.

Each of the three scenarios can be broken down to a seven step process as follows:

1. Identify the goals and requirements for the Copy Services to be performed (for example, backup, server migration, and server consolidation).
2. Identify the Virtual I/O Server and client configurations.
3. Complete the storage subsystem replication process.

   This step can be completed using either local or remote storage subsystem replication methods.

> **Note:** In this publication, only local Copy Services functionality is described.

4. Discover the replicated volume on the Virtual I/O Server.

   This step can be optional if you want to present the replicated volume to dedicated devices.
5. Map the replicated volume to the target client.

   This step can be optional if you want to present the replicated volume to dedicated devices.
6. Discover the new virtual SCSI device on the Virtual I/O Server client.

   Substitute a Virtual I/O Server client with a dedicated server as required.
7. Import the required devices and mount the file systems as appropriate.

1.3.1 Scenario #1: Copy of physical LUN and present to another client

In this scenario, there are two IBM PowerVM clients, each with one Volume Group (VG), which for the purposes of this demonstration contains the data to be replicated.

The objective is to create a block level copy of the VG using the three different Copy Services products: EMC TimeFinder, Hitachi ShadowImage, and IBM FlashCopy.

Once the copy is completed, the VG will be imported on a second client as a new VG with a new name and the file system built on the LV in the VG will be mounted.

This process is typically used for data migration, server consolidation, or point-in-time backups.

Figure 1-2 represents a logical overview of 1.3.1, “Scenario #1: Copy of physical LUN and present to another client” on page 5, demonstrating the replication of data from client1 to client2.

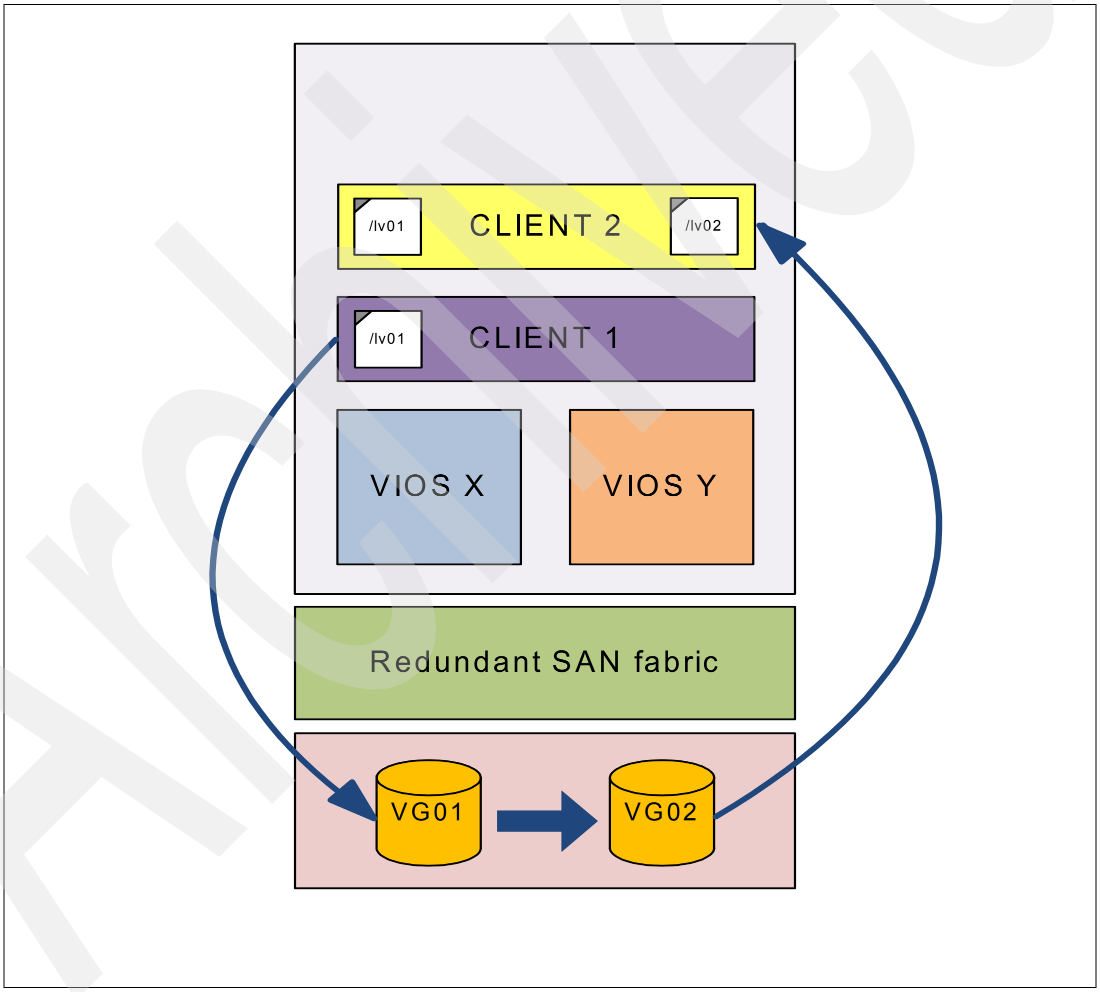


Figure 1-2 1.3.1, “Scenario #1: Copy of physical LUN and present to another client” on page 5

1.3.2 Scenario #2: Copy of physical LUN and present to same client

In this scenario, there is one IBM PowerVM client with one VG that, for the purposes of this demonstration, contains the data to be replicated.

The objective is to create a block level copy of the VG using the three different Copy Services products: EMC TimeFinder, Hitachi ShadowImage, and IBM FlashCopy.

Once the copy is completed, the VG will be imported on the same client as a new VG with a new name and the file system built on the LV in the VG will be mounted.

This process is typically used for local split-mirror backup processes or when a secondary copy of a local file system is needed for other purposes.

Figure 1-3 represents a logical overview of 1.3.2, "Scenario #2: Copy of physical LUN and present to same client" on page 6, demonstrating the replication of local data within client1.

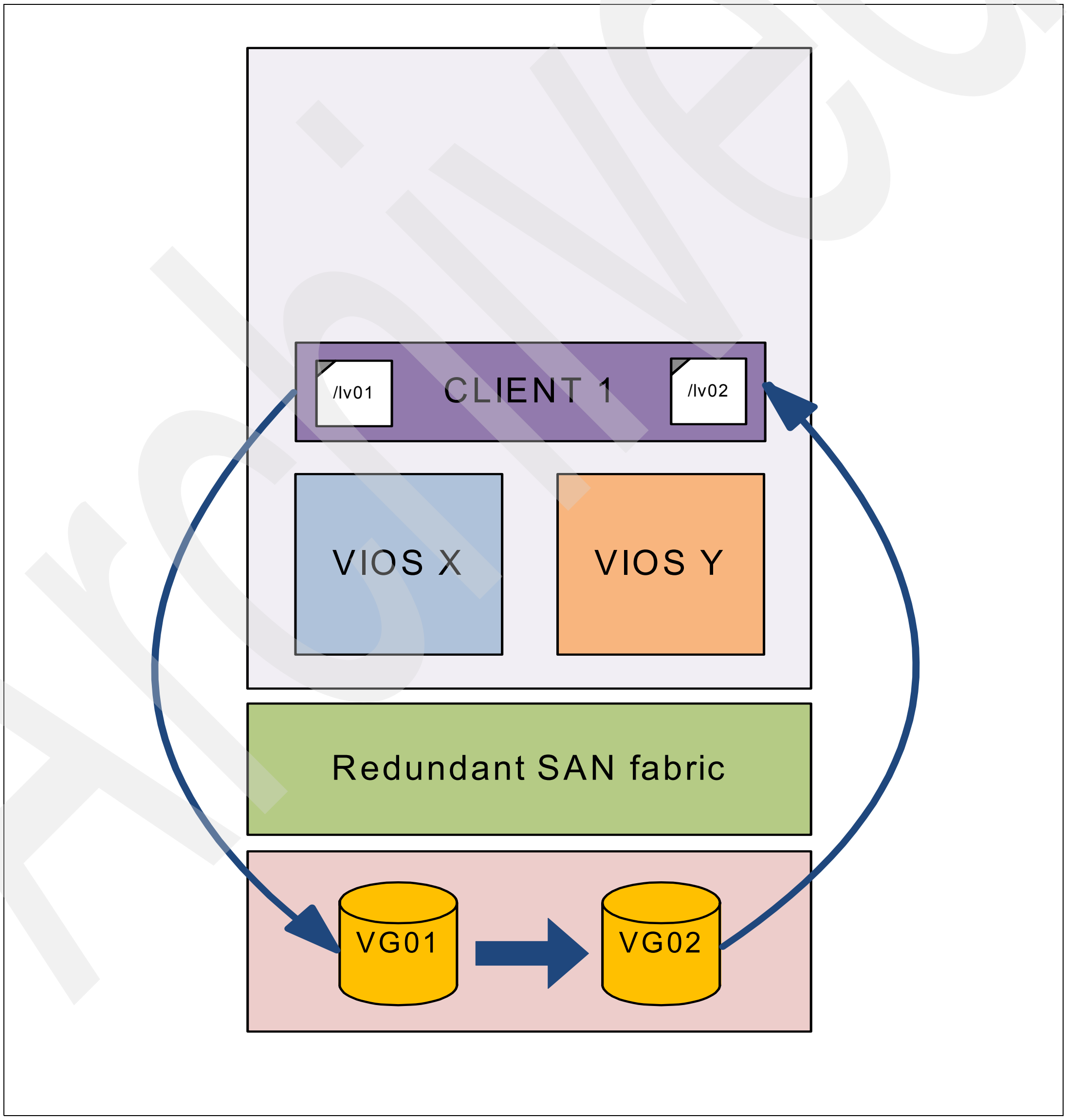


Figure 1-3 1.3.2, "Scenario #2: Copy of physical LUN and present to same client" on page 6

1.3.3 Scenario #3: Copy of physical LUN with multiple Logical Volumes and present to another client

In this scenario, there are two IBM PowerVM clients, each with one VG that (for the purposes of this demonstration) contains the data to be replicated.

In this scenario, each VG is a different Virtual I/O Server LV mapped as a virtual SCSI device to the clients. This differs from the previous two scenarios where the VG to be replicated was placed on its own dedicated LUN within the storage subsystem. This dedicated LUN was then mapped as a virtual SCSI device to the clients.

The objective is to create a block level copy of the VG using the three different Copy Services products: EMC TimeFinder, Hitachi ShadowImage, and IBM FlashCopy.

Once the copy is completed, the VG will imported on a third client as a new VG with a new name and the file system built on the LV in the VG will be mounted.

This process is typically used for regular split-mirror backups using a dedicated backup server or mount host.

Figure 1-4 represents a logical overview of 1.3.3, “Scenario #3: Copy of physical LUN with multiple Logical Volumes and present to another client” on page 7, demonstrating the replication of data from client1 & client2 to client3.

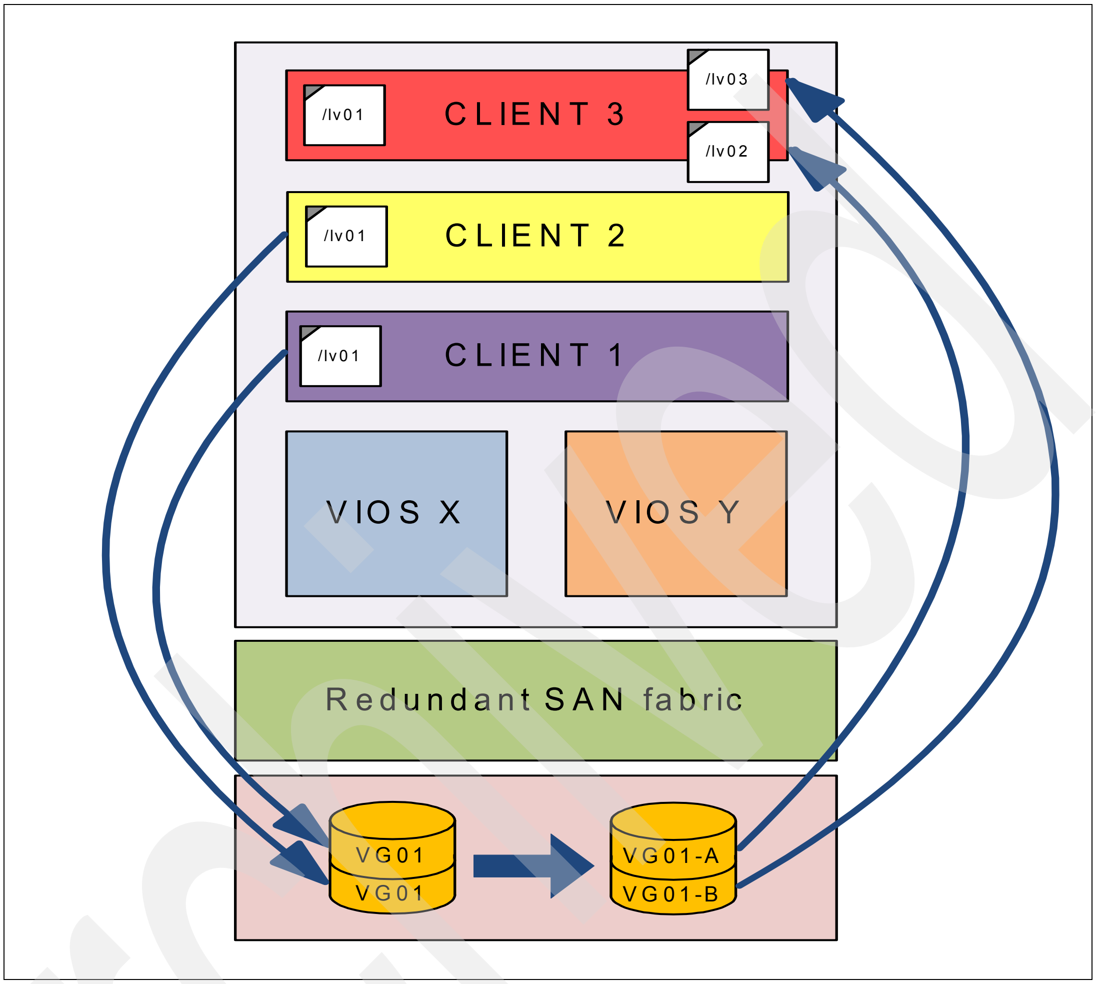


Figure 1-4 1.3.3, “Scenario #3: Copy of physical LUN with multiple Logical Volumes and present to another client” on page 7

2

Scenario #1: Copy of physical LUN and present to another client

In this scenario, there is one IBM PowerVM client, client1, with one Volume Group (VG), vg01, which contains the data to be replicated using Copy Services.

A copy will be made using the three Copy Services products:

- EMC TimeFinder
- Hitachi ShadowImage
- IBM FlashCopy

The copy will be mapped, discovered and imported onto a different client (client2) as a new VG. The Logical Volumes (LV), lv01 and its associated Logical Volume Manager (LVM) log volume, loglv01, will mount as a new file system.

This process is typically used for data migration or server consolidation.

2.1 EMC TimeFinder

In this section, the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using EMC TimeFinder are discussed.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the Physical Volume (PV) on the Virtual I/O Server where the VG and Logical Volume (LV) reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using EMC TimeFinder and mount it on client2:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk0 and has two LVs, lv01 and loglv01:

   ```
   # lspv
   hdisk43         00c7086c72bafc21                    rootvg          active
   hdisk0          00c7086ca22de0a3                    vg01            active

   # lsvg
   rootvg
   vg01

   # lsvg -l vg01
   vg01:
   LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   lv01                jfs2       130     130     1    open/syncd    /lv01
   loglv01             jfs2log    1       1       1    open/syncd    N/A
   ```

2. Now view the data that will be copied using the **ls** command:

   ```
   # ls -ltR /lv01
   /lv01:
   total 0
   drwx------    2 root     system          256 Oct 31 21:08 client1_dir
   drwxr-xr-x    2 root     system          256 Oct 29 16:32 lost+found

   /lv01/client1_dir:
   total 1024000
   -rw-------    1 root     system    524288000 Oct 31 21:08 file_client1

   /lv01/lost+found:
   total 0
   ```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. This unique ID is very important to know as it represents the physical LUN within the EMC storage subsystem.

   Physical disk hdisk0 will be our TimeFinder Clone source volume.

The **odmget** command is as follows:

```
# odmget -q "name=hdisk0 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk0"
        attribute = "unique_id"
        value = "321D1D0683033D09SYMMETRIX03EMCfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83) while the last four digits identify the Symmetrix Device ID of the LUN (033D).

Virtual I/O Server

Now that we know the unique ID hdisk0 within vg01 on client1, we now will determine which virtual device it represents on the Virtual I/O Server by performing the following step:

Using the **chkdev** command, you can determine that hdiskpower139 represents unique ID 1D0683033D09SYMMETRIX03EMCfcp within the Virtual I/O Server:

```
$ chkdev | grep -p 1D0683033D09SYMMETRIX03EMCfcp
NAME:                hdiskpower139
IDENTIFIER:          1D0683033D09SYMMETRIX03EMCfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

Note: Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicated by the bolded text within the following output:

```
321D1D0683033D09SYMMETRIX03EMCfcp05VDASD03AIXvscsi
```

Using the output that is highlighted in bold, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (033D)

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters, which allow for the direct connection to SAN fabrics for the allocation of a EMC gatekeeper devices for array management.

This server has EMC Solutions Enabler (SYMCLI) installed.

For this exercise, Symmetrix volume 033D will be the TimeFinder source volume and volume 0341 will be TimeFinder target volume.

To configure the storage subsystem management server, perform these steps:

1. Using the SYMCLI interface and the **symdg** and **symld** commands, you can create the Symmetrix Device Group that will be used to control the TimeFinder Clone operations against the source and target volumes.

   In this example, a device group client1_vg01_dg is created and the appropriate volumes are added to the device group:

   ```
   # symdg create client1_vg01_dg
   # symld -g client1_vg01_dg -sid 1983 add dev 033d
   # symld -g client1_vg01_dg -sid 1983 add dev 0341
   ```

2. Now display the configuration using the **symdg show** command:

   ```
   # symdg show client1_vg01_dg

   Group Name:  client1_vg01_dg

       Group Type                                   : REGULAR
       Device Group in GNS                          : No
       Valid                                        : Yes
       Symmetrix ID                                 : 000190101983
       Group Creation Time                          : Sat Oct 31 20:31:29 2009
       Vendor ID                                    : EMC Corp
       Application ID                               : SYMCLI

       Number of STD Devices in Group               :    2
       Number of Associated GK's                    :    0
       Number of Locally-associated BCV's           :    0
       Number of Locally-associated VDEV's          :    0
       Number of Locally-associated TGT's           :    0
       Number of Remotely-associated VDEV's(STD RDF):    0
       Number of Remotely-associated BCV's (STD RDF):    0
       Number of Remotely-associated TGT's(TGT RDF) :    0
       Number of Remotely-associated BCV's (BCV RDF):    0
       Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0
       Number of Remotely-assoc'd BCV's (Hop-2 BCV) :    0
       Number of Remotely-assoc'd VDEV's(Hop-2 VDEV):    0
       Number of Remotely-assoc'd TGT's (Hop-2 TGT) :    0

       Standard (STD) Devices (2):
           {
           ----------------------------------------------------------------
                                                     Sym                 Cap
           LdevName          PdevName                Dev  Att. Sts     (MB)
           ----------------------------------------------------------------
           DEV001            N/A                     033D (M)  RW     17258
           DEV002            N/A                     0341 (M)  RW     17258
           }
   ```

3. You are now ready to create a TimeFinder relationship and start copying data between the source and target volumes using the **symclone** command:

```
# symclone -g client1_vg01_dg create -precopy DEV001 sym ld DEV002

Execute 'Create' operation for device 'DEV001'
in device group 'client1_vg01_dg' (y/[n]) ? y

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'client1_vg01_dg'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'client1_vg01_dg' paired with target device 'DEV002'.

# symclone -g client1_vg01_dg query


Device Group (DG) Name: client1_vg01_dg
DG's Type             : REGULAR
DG's Symmetrix ID     : 000190101983


         Source Device                   Target Device            State
Copy
--------------------------------- ---------------------------- ------------
----
               Protected Modified                Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  Tracks   CGDP SRC <=> TGT  (%)
------------------------- ---------------------------- ------------ ----
DEV001    033D       0       0 DEV002    0341    0 XXX. Copied       100

Total           -------- --------                --------
  Track(s)             0        0                       0
  MB(s)              0.0      0.0                     0.0


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

4. Once the copy is complete, indicated by a status of Copied, use the **symclone** command to activate the target volume for host access:

```
# symclone -g client1_vg01_dg activate DEV001 sym ld DEV002

Execute 'Activate' operation for device 'DEV001'
in device group 'client1_vg01_dg' (y/[n]) ? y

'Activate' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'client1_vg01_dg'. Please wait...

'Activate' operation successfully executed for device 'DEV001'
in group 'client1_vg01_dg' paired with target device 'DEV002'.
```

Virtual I/O Server

When the copy operation successfully completes, you can present the new PV as a virtual device to client2 on the Virtual I/O Server by performing these steps:

1. Using the **chkdev** command, scan for the unique ID of the target volume, hdiskpower140, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 830341 | grep -p EMC
NAME:                hdiskpower140
IDENTIFIER:          1D0683034109SYMMETRIX03EMCfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (0341)

2. Now view the current virtual device configuration of client2 (vhost1) using the **lsmap** command:

```
$ lsmap -vadapter vhost1
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk760
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L1000000000000

VTD                   vtscsi69
Status                Available
LUN                   0x8a00000000000000
Backing device        hdiskpower69
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L239
```

3. Now that you know that the target volume is hdiskpower140 on the Virtual I/O Server, you can now assign it to client2 (vhost1) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdiskpower140 -vadapter vhost1
vtscsi2 Available

$ lsmap -vadapter vhost1
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi2
Status                Available
LUN                   0x8100000000000000
Backing device        hdiskpower140
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L862

VTD                   vtscsi3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk760
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L1000000000000

VTD                   vtscsi69
Status                Available
LUN                   0x8a00000000000000
Backing device        hdiskpower69
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L239
```

Client2

At this point, you have successfully created a copy of client1 vg01 using EMC TimeFinder and used the Virtual I/O Server to allocate the new virtual device to client2. Now it is time to scan for the new device and import the VG. Perform these steps:

1. Use the **cfgmgr** command to scan for the new device:

```
# cfgmgr

# lspv
hdisk0          00c7086c77cce272                    rootvg          active
hdisk1          00c7086c831f9c2a                    vg01            active
hdisk2          00c7086ca22de0a3                    None
```

2. Import the VG using the **importvg** command, making sure you rename it with a name that does not already exist on client2:

```
# importvg -y vg02 hdisk2
vg01
vg02

# lspv
hdisk0          00c7086c77cce272                    rootvg          active
hdisk1          00c7086c831f9c2a                    vg01            active
hdisk2          00c7086ca22de0a3                    vg02            active

# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP: vg02
PV IDENTIFIER:      00c7086ca22de0a3 VG IDENTIFIER
00c7086c00004c0000000124a22de0ce
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  2
TOTAL PPs:          134 (17152 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           3 (384 megabytes)        HOT SPARE:        no
USED PPs:           131 (16768 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  00..00..00..00..03
USED DISTRIBUTION:  27..27..26..27..24
MIRROR POOL:        None
```

3. In the output of the **importvg** command, the Logical Volumes on hdisk2, lv01, and loglv01, are already used on client2. The **importvg** command renamed them automatically to fslv00 and loglv02.

   Use the **lsvg** command to check the new names of the LVs:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv00              jfs2       130     130     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. Because the new VG name is vg02, loglv02 actually suits the new configuration quite well, but you might want to change the name of fslv00 using the **chlv** command:

```
# chlv -n lv02 fslv00

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       130     130     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

5. Now you can edit /etc/filesystems, create the mountpoints, and mount the new LVs by using the **vi** command:

```
# vi /etc/filesystems
(Append the following)
/lv02:
        dev             = /dev/lv02
        vfs             = jfs2
        log             = /dev/loglv02
        mount           = true
        account         = false

# mkdir /lv02

# mount /lv02
Replaying log for /dev/lv02.
```

6. Your last step is to look at your replicated data by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 31 21:08 client1_dir
drwxr-xr-x    2 root     system          256 Oct 29 16:32 lost+found

/lv02/client1_dir:
total 1024000
-rw-------    1 root     system    524288000 Oct 31 21:08 file_client1

/lv02/lost+found:
total 0
```

2.2 Hitachi ShadowImage

In this section, the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using Hitachi ShadowImage are discussed.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and LV reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using Hitachi ShadowImage and mount it on client2:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk0 and has two Logical Volumes (LV), lv01 and loglv01.

```
# lspv
hdisk43         00c7086c72bafc21                    rootvg          active
hdisk0          00c7086c8321623a                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. Now view the data that will be copied by using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Oct 29 14:08 Client1
drwxr-xr-x    2 root     system          256 Oct 28 16:32 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    524288000 Oct 29 14:08 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. It is important that you know this unique ID, as it represents the physical LUN within the Hitachi storage subsystem. Physical disk hdisk0 will be the ShadowImage source or P-VOL. Run the following odmget command:

```
# odmget -q "name=hdisk0 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk0"
        attribute = "unique_id"
        value = "3924240C50 0B0FA020006OPEN-V07HITACHIfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that the unique ID is known (hdisk0 within vg01 on client1), proceed to determine which virtual device it represents on the Virtual I/O Server by performing these steps:

1. Using the **chkdev** command, you can determine that unique hdisk759 represents ID 240C50 0B0FA020006OPEN-V07HITACHIfcp within the Virtual I/O Server:

```
$ chkdev | grep -p 0B0FA020006OPEN-V07HITACHIfcp
NAME:                hdisk759
IDENTIFIER:          240C50 0B0FA020006OPEN-V07HITACHIfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:
>
> `3924240C50 0B0FA020006OPEN-V07HITACHIfcp05VDASD03AIXvscsi`

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters, which allow for the direct connection to SAN fabrics for the allocation of a Hitachi command device.

The server has Hitachi storage system Command Control Interface (CCI) installed.

> **Note:** Hitachi ShadowImage software uses the in-band Fibre Channel (FC) protocol to send commands to the storage system in order to create and manage Hitachi Online RAID Configuration Manager (HORCM) associations.

Perform the following steps to configure the storage subsystem management server:

1. Using the **odmget CuAt** command, identify the PV of unique ID 240C50 0B0FA020006OPEN-V07HITACHIfcp, which will be our ShadowImage source or P-VOL:

```
# odmget CuAt | grep -p HITACHI
CuAt:
        name = "hdisk140"
        attribute = "unique_id"
        value = "240C50 0B0FA020006OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk141"
        attribute = "unique_id"
        value = "240C50 0B0FA020106OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk142"
        attribute = "unique_id"
        value = "240C50 0B0FA020206OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk143"
        attribute = "unique_id"
        value = "240C50 0B0FA020306OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk144"
        attribute = "unique_id"
        value = "270C50 0B0FA01A309OPEN-V-CM07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk145"
        attribute = "unique_id"
        value = "240C50 0B0FA030006OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk467"
        attribute = "unique_id"
        value = "240C50 0B0FA031006OPEN-V07HITACHIfcp"
        type = "R"
        generic = ""
        rep = "nl"
        nls_index = 79
```

This command is particularly useful, as you can also look to make sure a Hitachi ShadowImage command device is allocated by scanning for a unique ID that contains OPEN-V-CM07.

Physical disk hdisk144 is the ShadowImage command device.

Physical disk hdisk140 is the ShadowImage source or P-VOL.

hdisk142 (unique ID = 240C50 0B0FA020206OPEN-V07HITACHIfcp) is allocated to the storage subsystem management server, which is the ShadowImage target or S-VOL.

Note: In this case, you only need one command device for ShadowImage operations. For more complicated environments, you may need to assign multiple command devices. Do not forget that ShadowImage and TrueCopy should not share the same command device.

2. Using the Hitachi storage system Command Control Interface (CCI), you can create the source and target Hitachi Online RAID Configuration Manager (HORCM) configuration files using the **`mkconf`** command:

```
# echo hdisk140 | /HORCM/usr/bin/mkconf.sh -g Client1 -i 1 -s 54001
starting HORCM inst 1
HORCM inst 1 starts successfully.
HORCM Shutdown inst 1 !!!
A CONFIG file was successfully completed.
starting HORCM inst 1
HORCM inst 1 starts successfully.
DEVICE_FILE             Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk140                Client1  Client1_000 CL7-F     0    0 -    45306   512
HORCM Shutdown inst 1 !!!
Please check '/apps/horcm1.conf','/apps/log1/curlog/horcm_*.log', and modify
'ip_address & service'.

# echo hdisk142 | /HORCM/usr/bin/mkconf.sh -g Client1 -i 4 -s 54004
starting HORCM inst 4
HORCM inst 4 starts successfully.
HORCM Shutdown inst 4 !!!
A CONFIG file was successfully completed.
starting HORCM inst 4
HORCM inst 4 starts successfully.
DEVICE_FILE             Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk142                Client1  Client1_000 CL7-F     0    3 -    45306   515
HORCM Shutdown inst 4 !!!
Please check '/apps/horcm4.conf','/apps/log4/curlog/horcm_*.log', and modify
'ip_address & service'.
```

A single server configuration will be implemented in this example, which establishes a relationship between the device of HORCM instance 1 and the device of HORCM instance 4 on the same server.

HORCM instance 1 is the P-VOL

HORCM instance 4 is the S-VOL.

3. Edit the HORCM configuration files and change the IP address of the HORCM_MON and HORCM_INST stanza’s to be that of the localhost. Also, change the service of the HORCM_INST stanza’s to point at each other (that is, horcm1 = 54001 and horcm4 = 54004):

```
# vi horcm1.conf

# vi horcm4.conf

# vi /etc/services
```

Note: The following is an example S-VOL HORCM configuration file that was used for this step:

```
# Created by mkconf.sh on Mon Oct 26 10:31:35 CDT 2009
HORCM_MON
#ip_address        service         poll(10ms)     timeout(10ms)
9.3.92.181         54004                  1000              3000
HORCM_CMD
#dev_name               dev_name                dev_name
#UnitID 0 (Serial# 45306)
/dev/rhdisk144
HORCM_DEV
#dev_group         dev_name        port#     TargetID     LU#     MU#
# hdisk142             SER =    45306  LDEV =  515 [ FIBRE FCTBL = 3 ]
Client1            Client1_000     CL7-F            0       3
HORCM_INST
#dev_group        ip_address      service
Client1          9.3.92.181      54001
```

4. Set up your environment variables and start the two HORCM instances:

```
# export HORCMINST=1

# export HORCC_MRCF=1

# horcmstart.sh
starting HORCM inst 1
HORCM inst 1 starts successfully.

# export HORCMINST=4

# horcmstart.sh
starting HORCM inst 4
HORCM inst 4 starts successfully.
```

Note: The environment variable set here, HORCC_MRCF=1, needs to be set in order to specify ShadowImage functions. If it is not set, TrueCopy functions will be performed instead.

5. Once the HORCM instances start successfully, you can now display the configuration using the **pairdisplay** command:

```
# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 SMPL     -     - - -
Client1 Client1_000 R   hdisk140       0  45306   200 SMPL     -     - - -
```

You are now ready to create a ShadowImage relationship between our P-VOL and S-VOL. This is a very destructive command, so you should follow a simple procedure so that you will not need to restore from tape: When creating a relationship, always use the HORCM instance of the S-VOL, which means that when you specify the relationship copy direction, you will always use the -vr switch (that is, -vr for remote or -vl for local).

In this example, a concurrent track copy count of 15 was used.

6. Using the **echo** command, verify the HORCM instance you currently have configured:

```
# echo $HORCMINST
4

# paircreate -g Client1 -c 15 –vr

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 S-VOL COPY     3 200 -
Client1 Client1_000 R   hdisk140       0  45306   200 P-VOL COPY     3 202 –

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 S-VOL PAIR   100 200 -
Client1 Client1_000 R   hdisk140       0  45306   200 P-VOL PAIR   100 202 –
```

7. Once the copy is complete, indicated by a status of PAIR, use the **pairsplit** command to remove the relationship:

```
# pairsplit -S -g Client1

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 SMPL     -     - - -
Client1 Client1_000 R   hdisk140       0  45306   200 SMPL     -     - - -
```

   The copy operation is now complete and ready to be presented to client2.

Virtual I/O Server

Once the copy operation successfully completes, you can present the new PV as a virtual device to client2 on the Virtual I/O Server.

The following steps can be performed to import the VG onto the Virtual I/O Server and present the VG as a virtual device to client2:

1. Using the **chkdev** command, scan for the unique ID of the S-VOL, 240C50 0B0FA020206OPEN-V07HITACHIfcp, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p "240C50 0B0FA020206OPEN-V07HITACHIfcp"
NAME:                hdisk761
IDENTIFIER:          240C50 0B0FA020206OPEN-V07HITACHIfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

2. View the current virtual device configuration of client2 (vhost1) using the **lsmap** command:

```
$ lsmap -vadapter vhost1
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk760
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L1000000000000

VTD                   vtscsi69
Status                Available
LUN                   0x8a00000000000000
Backing device        hdiskpower69
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L239
```

3. Knowing that the S-VOL is hdisk761 on the Virtual I/O Server, you can now assign it to client2 (vhost1) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk761 -vadapter vhost1
vtscsi4 Available

$ lsmap -vadapter vhost1
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi3
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk760
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L1000000000000

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk761
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L2000000000000

VTD                   vtscsi69
Status                Available
LUN                   0x8a00000000000000
Backing device        hdiskpower69
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L239
```

Client2

At this point, you have successfully created a copy of client1 vg01 using Hitachi ShadowImage and used the Virtual I/O Server to map the new virtual device to client2. Now it is time to scan for the new device and import the VG. Perform the following steps:

1. Use the **cfgmgr** command to scan for the new device:

```
# cfgmgr

# lspv
hdisk0          00c7086c77cce272                    rootvg          active
hdisk1          00c7086c831f9c2a                    vg01            active
hdisk2          00c7086c8321623a                    None
```

2. You can import the VG by using the **importvg** command, making sure to rename it with a name that does not already exist on client2:

```
# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A

# importvg -y vg02 hdisk2
0516-530 synclvodm: Logical volume name lv01 changed to fslv00.
0516-530 synclvodm: Logical volume name loglv01 changed to loglv02.
0516-712 synclvodm: The chlv succeeded, however chfs must now be
        run on every filesystem which references the old log name loglv01.
imfs: Warning: mount point /lv01 already exists in /etc/filesystems.
vg02

# lspv
hdisk0          00c7086c77cce272                    rootvg          active
hdisk1          00c7086c831f9c2a                    vg01            active
hdisk2          00c7086c8321623a                    vg02            active

# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     vg02
PV IDENTIFIER:      00c7086c8321623a VG IDENTIFIER
00c7086c00004c000000012483216266
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  2
TOTAL PPs:          159 (20352 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           8 (1024 megabytes)       HOT SPARE:        no
USED PPs:           151 (19328 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  00..00..00..00..08
USED DISTRIBUTION:  32..32..31..32..24
MIRROR POOL:        None
```

As shown in the output, the LVs on hdisk2, lv01 and loglv01, are already used on client2. The **importvg** command renamed them automatically to fslv00 and loglv02.

3. Use the **lsvg** command to check the new names of the LVs:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv00              jfs2       150     150     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. Because the new VG name is vg02, loglv02 actually suits the new configuration quite well, but you might want to change the name of fslv00 using the **chlv** command:

```
# chlv -n lv02 fslv00

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       150     150     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

5. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new Logical Volumes by using the **vi** command:

```
# vi /etc/filesystems
(Append the following)
/lv02:
    dev     = /dev/lv02
    vfs     = jfs2
    log     = /dev/loglv02
    mount       = true
    account     = false

# mkdir /lv02

# mount /lv02
Replaying log for /dev/lv02.
```

6. Look at your replicated data by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 29 14:08 Client1
drwxr-xr-x    2 root     system          256 Oct 28 16:32 lost+found

/lv02/Client1:
total 1024000
-rw-------    1 root     system    524288000 Oct 29 14:08 This is Client1

/lv02/lost+found:
total 0
```

2.3 IBM FlashCopy: IBM System Storage DS8300

In this section, the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using IBM FlashCopy are discussed.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and LV reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps create a block level copy of vg01 using IBM FlashCopy and mount it on client2:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk1 and has two Logical Volumes (LV), lv01 and loglv01:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe4111db6f92c                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. Now view the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Oct 30 14:19 Client1
drwxr-xr-x    2 root     system          256 Oct 30 10:46 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    524288000 Oct 30 14:19 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. This unique ID is very important to know, as it represents the physical LUN within the IBM storage subsystem. Physical disk hdisk1 will be the FlashCopy source volume. Run the following **odmget** command:

```
# odmget -q "name=hdisk1 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk1"
        attribute = "unique_id"
        value = "3520200B75BALB1100D07210790003IBMfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that you know the unique ID of hdisk1 within vg01 on client1, now you must determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can see that hdisk8 represents unique ID 3520200B75BALB1100D07210790003IBMfcp05VDASD03AIXvscsi within the Virtual I/O Server:

```
$ chkdev | grep -p 200B75BALB1100D07210790003IBMfcp
NAME:                 hdisk8
IDENTIFIER:           200B75BALB1100D07210790003IBMfcp
PHYS2VIRT_CAPABLE:    NA
VIRT2NPIV_CAPABLE:    YES
VIRT2PHYS_CAPABLE:    YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:
>
> 3520**200B75BALB1100D07210790003IBMfcp**05VDASD03AIXvscsi

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client.

The server has IBM DSCLI installed, which is used to manage storage subsystem functions.

Perform the following steps:

1. Using the **odmget CuAt** command, look for the unique ID 200B75BALB1100D07210790003IBMfcp, which will be our FlashCopy source volume:

```
# odmget CuAt | grep -p IBMfcp
CuAt:
        name = "hdisk6"
        attribute = "unique_id"
        value = "3E213600A0B8000291B0800009DCB0402FC540F1815
FAStT03IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79
```

```
CuAt:
        name = "hdisk7"
        attribute = "unique_id"
        value = "3E213600A0B8000291B0800009DCC0402FC6C0F1815
FAStT03IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk8"
        attribute = "unique_id"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk9"
        attribute = "unique_id"
        value = "200B75BALB1100E07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk10"
        attribute = "unique_id"
        value = "200B75BALB1110707210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk11"
        attribute = "unique_id"
        value = "200B75BALB1110807210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "vtscsi2"
        attribute = "udid_info"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0
```

Physical disk hdisk8 will be the FlashCopy source volume.

hdisk11 (unique ID = 200B75BALB1110807210790003IBMfcp) is allocated to the FlashCopy management server, which will be the FlashCopy target volume.

2. Using the DSCLI **lshostvol** command, determine the IBM System Storage™ DS8300 vpath ID of hdisk10 and hdisk11:

```
dscli> lshostvol
Date/Time: November 2, 2009 3:42:09 PM CST IBM DSCLI Version: 5.4.1.44 DS: -
Disk Name Volume Id             Vpath Name
==========================================
hdisk8    IBM.2107-75BALB1/100D ---
hdisk9    IBM.2107-75BALB1/100E ---
hdisk10   IBM.2107-75BALB1/1107 ---
hdisk11   IBM.2107-75BALB1/1108 ---
```

3. You are now ready to create the FlashCopy relationship between the source (vpath ID = 100D) and target (vpath ID = 1108) volumes using your FlashCopy management server and the **mkflash** command. Run the following command;

```
dscli> mkflash -dev IBM.2107-75BALB1 -persist -cp -seqnum 0001 100D:1108
Date/Time: November 2, 2009 3:42:28 PM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00137I mkflash: FlashCopy pair 100D:1108 successfully created.
```

4. Display the configuration using the **lsflash** command:

```
dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 3:42:58 PM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                   State AllowTgtSE
===============================================================================
===============================================================================
==================================================================
100D:1108 10     1           120     Enabled    Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        258760          Mon Nov 02
15:47:59 CST 2009 Mon Nov 02 15:47:59 CST 2009 Valid Disabled

dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 3:45:00 PM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                   State AllowTgtSE
===============================================================================
===============================================================================
==================================================================
100D:1108 10     1           120     Disabled   Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        0               Mon Nov 02
15:47:59 CST 2009 Mon Nov 02 15:47:59 CST 2009 Valid Disabled
```

5. Once the copy operation is complete, indicated by the 0 in 'OutOfSyncTracks', use the **rmflash** command to remove the relationship:

```
dscli> rmflash -dev IBM.2107-75BALB1 -seqnum 0001 100D:1108
Date/Time: November 2, 2009 3:45:18 PM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00144W rmflash: Are you sure you want to remove the FlashCopy pair
100D:1108:? [y/n]:y
CMUC00140I rmflash: FlashCopy pair 100D:1108 successfully removed.

dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 3:45:33 PM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00234I lsflash: No Flash Copy found.
```

Virtual I/O Server

Now that the copy operation is complete, you need to present the new PV as a virtual device to client2 by performing the following steps:

1. Using the **chkdev** command, scan for the unique ID of the FlashCopy target volume, 200B75BALB1110807210790003IBMfcp, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 200B75BALB1110807210790003IBMfcp
NAME:                hdisk11
IDENTIFIER:          200B75BALB1110807210790003IBMfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

2. View the current virtual device configuration of client2 (vhost0) using the **lsmap** command:

```
$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                     0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi5
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk9
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4010400E00000000
```

3. Because you know that the FlashCopy target volume is hdisk11 on the Virtual I/O Server, you can now assign it to client2 (vhost2) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk11 -vadapter vhost0
vtscsi3 Available

$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                     0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi3
Status                Available
LUN                   0x8300000000000000
Backing device        hdisk11
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4011400800000000

VTD                   vtscsi5
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk9
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4010400E00000000
```

Client2

At this point, you have successfully created a copy of client1, vg01, using IBM FlashCopy, and you used the Virtual I/O Server to map the new virtual device to client2. Now it is time to scan for the new device and import the VG.

Perform the following steps:

1. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411aeb7c10a                    vg01            active
hdisk2          000fe4111db6f92c                    None
```

2. You can now import the VG using the **importvg** command, making sure to rename it with a name that does not already exist on client2:

```
# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
```

```
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A

# importvg -y vg02 hdisk2
0516-530 synclvodm: Logical volume name lv01 changed to fslv00.
0516-530 synclvodm: Logical volume name loglv01 changed to loglv02.
0516-712 synclvodm: The chlv succeeded, however chfs must now be
        run on every filesystem which references the old log name loglv01.
imfs: Warning: mount point /lv01 already exists in /etc/filesystems.
vg02

# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411aeb7c10a                    vg01            active
hdisk2          000fe4111db6f92c                    vg02            active

# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     vg02
PV IDENTIFIER:      000fe4111db6f92c VG IDENTIFIER
000fe4110000d9000000016e1db6f963
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  2
TOTAL PPs:          159 (20352 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           8 (1024 megabytes)       HOT SPARE:        no
USED PPs:           151 (19328 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  00..00..00..00..08
USED DISTRIBUTION:  32..32..31..32..24
MIRROR POOL:        None
```

3. In the output of the **importvg** command, the Logical Volumes on hdisk2, lv01 and loglv01, are already used on client2. The **importvg** command renamed them automatically to fslv00 and loglv02.

   Using the **lsvg** command, check the new names of the LVs:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv00              jfs2       150     150     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. Because the new VG name is vg02, loglv02 actually suits the new configuration quite well, but you might want to change the name of fslv00 using the **chlv** command:

```
# chlv -n lv02 fslv00

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       150     150     1    closed/syncd  N/A
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

5. Now you can edit /etc/filesystems, create the mountpoints, and mount the new Logical Volumes using the vi command:

```
# vi /etc/filesystems
(Append the following)
/lv02:
    dev     = /dev/lv02
    vfs     = jfs2
    log     = /dev/loglv02
    mount       = true
    account     = false

# mkdir /lv02

# mount /lv02
Replaying log for /dev/lv02.
```

6. Your last step is to look at your replicated data using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 30 14:19 Client1
drwxr-xr-x    2 root     system          256 Oct 30 10:46 lost+found

/lv02/Client1:
total 1024000
-rw-------    1 root     system    524288000 Oct 30 14:19 This is Client1

/lv02/lost+found:
total 0
```

2.4 IBM FlashCopy: SAN Volume Controller (SVC)

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using IBM FlashCopy.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the Storage Subsystem Management Server
- Determine the PV on the Virtual I/O Server where the VG and LV reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps make a block level copy of vg01 using IBM FlashCopy and mount it on client2:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk1 and has two Logical Volumes (LV), lv01 and loglv01:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe41141804594                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. View the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Nov 02 08:51 Client1
drwxr-xr-x    2 root     system          256 Nov 02 07:46 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    524288000 Nov 02 08:51 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. It is important to know this unique ID, as it represents the physical LUN presented by way of the IBM SAN Volume Controller (SVC).

   Physical disk hdisk1 will be the FlashCopy source volume.

   Perform the following **odmget** command:

```
# odmget -q "name=hdisk1 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk1"
        attribute = "unique_id"
        value =
"48333321360050768018300FD50000000000000004214503IBMfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that you know the unique ID of hdisk1 within vg01 on client1, determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can see that hdisk12 represents unique ID 48333321360050768018300FD500000000000000004214503IBMfcp05VDASD03AIXvscsi within the Virtual I/O Server:

```
$ chkdev | grep -p 3321360050768018300FD500000000000000004214503IBMfcp
NAME:                hdisk12
IDENTIFIER:          3321360050768018300FD500000000000000004214503IBMfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

Note: Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:

4833**3321360050768018300FD500000000000000004214503IBMfcp**05VDASD03AIXvscsi

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client. The server has an SSH client installed that is used to manage IBM SAN Volume Controller (SVC) functions.

Perform these steps:

1. Using the **svcinfo lshostvdiskmap** command, look for the component of the unique ID that contains the SAN Volume Controller (SVC) vdisk_UID 60050768018300FD5000000000000000, which will be the FlashCopy source volume:

   ```
   IBM_2145:admin>svcinfo lshostvdiskmap
   id              name             SCSI_id        vdisk_id       vdisk_name
   wwpn             vdisk_UID
   0               host0             0             0              host0_1
   10000000C9738E84 60050768018300FD5000000000000000
   0               host0            1              1              host0_2
   10000000C9738E84 60050768018300FD5000000000000001
   0               host0            2              2              host0_3
   10000000C9738E84 60050768018300FD5000000000000002
   0               host0            3              3              host0_4
   10000000C9738E84 60050768018300FD5000000000000003
   0               host0            4              4              host0_5
   10000000C9738E84 60050768018300FD5000000000000004
   ```

   In this example, host0 is the Virtual I/O Server where all three clients are allocated.

   Physical disk vhost0_1 (unique ID = 60050768018300FD5000000000000000) will be the FlashCopy source volume.

   Physical disk vhost0_5 (unique ID = 60050768018300FD5000000000000004) will be the FlashCopy target volume.

2. Now you can create the FlashCopy relationship between the source and target volumes using your FlashCopy management server and the **mkfcconsistgrp** and **mkfcmap** commands:

```
IBM_2145:admin>svctask mkfcconsistgrp -name host0
FlashCopy Consistency Group, id [1], successfully created

IBM_2145:admin>svctask mkfcmap -source host0_1 -target host0_5 -name host0
-consistgrp host0
FlashCopy Mapping, id [0], successfully created

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

3. Once the FlashCopy relationship is created, prepare the consistency group with the **prestartfcconsistgrp** command:

```
IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status prepared
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

4. Now start the FlashCopy using the **startfcconsistgrp** command:

```
IBM_2145:admin>svctask startfcconsistgrp host0

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status copying
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 100
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

5. Once the copy operation is complete, indicated by a progress of“100 and a status of idle_or_copied, use the **rmfcmap** command to remove the relationship:

```
IBM_2145:admin>svctask rmfcmap host0

IBM_2145:admin>svcinfo lsfcmap host0
CMMVC5754E The object specified does not exist, or the name supplied does not
meet the naming rules.
```

Virtual I/O Server

Now that the copy operation is complete, you need to present the new PV as a virtual device to client2 by performing these steps:

1. Using the **chkdev** command, scan for the unique ID of the FlashCopy target volume, 60050768018300FD5000000000000004, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 60050768018300FD5000000000000004
NAME:                hdisk16
IDENTIFIER:          332136005076801830OFD5000000000000004042145O3IBMfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
```

```
VIRT2PHYS_CAPABLE:   NA
```

2. Now view the current virtual device configuration of client2 (vhost0) using the **lsmap** command:

```
$ lsmap -vadapter vhost0
SVSA            Physloc                                       Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                      0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi3
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk13
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L1000000000000
```

3. Now that you know that the FlashCopy target volume is hdisk16 on the Virtual I/O Server, you can now assign it to client2 (vhost0) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk16 -vadapter vhost0
vtscsi6 Available

$ lsmap -vadapter vhost0
SVSA            Physloc                                       Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                      0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi3
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk13
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L1000000000000

VTD                   vtscsi6
Status                Available
LUN                   0x8300000000000000
Backing device        hdisk16
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L4000000000000
```

Client2

At this point, you have successfully created a copy of client1, vg01, using IBM FlashCopy and used the Virtual I/O Server to map the new virtual device to client2. Now it is time to scan for the new device and import the VG.

Perform these steps:

1. Using the **cfgmgr** command, we scan for the new device:

```
# cfgmgr

# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411c1eb093d                    vg01            active
hdisk2          000fe41141804594                    None
```

2. You can now import the VG using the **importvg** command, making sure to rename it with a name that does not already exist on client2.

```
# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A

# importvg -y vg02 hdisk2
0516-530 synclvodm: Logical volume name lv01 changed to fslv00.
0516-530 synclvodm: Logical volume name loglv01 changed to loglv00.
0516-712 synclvodm: The chlv succeeded, however chfs must now be
        run on every filesystem which references the old log name loglv01.
imfs: Warning: mount point /lv01 already exists in /etc/filesystems.
vg02

# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411c1eb093d                    vg01            active
hdisk2          000fe41141804594                    vg02            active

# lspv hdisk2
PHYSICAL VOLUME:    hdisk2                   VOLUME GROUP:     vg02
PV IDENTIFIER:      000fe41141804594 VG IDENTIFIER
000fe4110000d9000000016e41808f94
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            128 megabyte(s)          LOGICAL VOLUMES:  2
TOTAL PPs:          159 (20352 megabytes)    VG DESCRIPTORS:   2
FREE PPs:           8 (1024 megabytes)       HOT SPARE:        no
USED PPs:           151 (19328 megabytes)    MAX REQUEST:      256 kilobytes
FREE DISTRIBUTION:  00..00..00..00..08
USED DISTRIBUTION:  32..32..31..32..24
MIRROR POOL:        None
```

3. In the output of the **importvg** command, the Logical Volumes on hdisk2, lv01 and loglv01, were already used on client2. The **importvg** command renamed them automatically to fslv00 and loglv02.

   Using the **lsvg** command, check the new names of the LVs:

   ```
   # lsvg -l vg02
   vg02:
   LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   fslv00              jfs2       150     150     1    closed/syncd  N/A
   loglv02             jfs2log    1       1       1    closed/syncd  N/A
   ```

4. Because the new VG name is vg02, loglv02 actually suits the new configuration quite well, but you might want to change the name of fslv00 using the **chlv** command:

   ```
   # chlv -n lv02 fslv00

   # lsvg -l vg02
   vg02:
   LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   lv02                jfs2       150     150     1    closed/syncd  N/A
   loglv02             jfs2log    1       1       1    closed/syncd  N/A
   ```

5. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new Logical Volumes using the **vi** command:

   ```
   # vi /etc/filesystems
   (Append the following)
   /lv02:
       dev     = /dev/lv02
       vfs     = jfs2
       log     = /dev/loglv02
       mount       = true
       account     = false

   # mkdir /lv02

   # mount /lv02
   Replaying log for /dev/lv02.
   ```

6. Your last step is to look at your replicated data using the **ls** command:

   ```
   # ls -ltR /lv02
   /lv02:
   total 0
   drwx------    2 root     system          256 Nov 02 08:51 Client1
   drwxr-xr-x    2 root     system          256 Nov 02 07:46 lost+found

   /lv02/Client1:
   total 1024000
   -rw-------    1 root     system    524288000 Nov 02 08:51 This is Client1

   /lv02/lost+found:
   total 0
   ```

Scenario #2: Copy of physical LUN and present to same client

In this scenario, there are two IBM PowerVM clients, client1 and client2, each with one Volume Group (VG), vg01, which contains the data to be replicated using Copy Services.

A copy will be made using the three Copy Services products:

- EMC TimeFinder
- Hitachi ShadowImage
- IBM FlashCopy

The copy will be mapped, discovered, and imported on the same client (client1) as a new VG. The Logical Volume (LV) lv01 and its associated Logical Volume Manager (LVM) log volume, loglv01, will mount as a new file system.

This process is typically used for local split-mirror backup processes where the primary disk resources are minimally interrupted during the initial commencement of the Copy Services operation and are relieved of any additional backup I/O load.

3.1 EMC TimeFinder

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using EMC TimeFinder.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and LV reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which, for the purposes of this demonstration, contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using EMC TimeFinder Clone and mount the volume back to client1:

1. When you use the **lspv** command, you can see that vg01 resides on hdisk0:

```
# lspv
hdisk43       00c7086c72bafc21                    rootvg          active
hdisk0        00c7086ca22de0a3                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME        TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01           jfs2       130     130     1    open/syncd    /lv01
loglv01        jfs2log    1       1       1    open/syncd    N/A
```

2. View the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Nov 01 02:36 client1
drwxr-xr-x    2 root     system          256 Oct 29 16:32 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    524288000 Nov 01 02:26 file_client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. It is important to know this unique ID, as it represents the physical LUN within the EMC storage subsystem.

Run the following command:

```
# odmget -q "name=hdisk0 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk0"
        attribute = "unique_id"
        value = "321D1D0683033D09SYMMETRIX03EMCfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Physical disk hdisk0 will be our TimeFinder Clone source volume.

Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (033D)

Virtual I/O Server

Now that you know the unique ID of hdisk0 within vg01 on client1, determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can determine that hdiskpower139 represents unique ID 1D0683033D09SYMMETRIX03EMCfcp in the Virtual I/O Server:

```
$ chkdev | grep -p "1D0683033D09SYMMETRIX03EMCfcp"
NAME:                hdiskpower139
IDENTIFIER:          1D0683033D09SYMMETRIX03EMCfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following:
>
> 32**1D1D0683033D09SYMMETRIX03EMCfcp**05VDASD03AIXvscsi

Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (033D)

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters that allow for the direct connection to SAN fabrics for the allocation of a EMC gatekeeper devices for array management.

The server has EMC Solutions Enabler (SYMCLI) installed.

For this exercise, Symmetrix volume 033D will be the TimeFinder source volume and volume 0341 will be TimeFinder target volume.

Perform these steps:

1. Using the SYMCLI interface and the **symdg** and **symld** commands, you can create the Symmetrix Device Group to be used to control the TimeFinder Clone operations against the source and target volumes:

   ```
   # symdg create client1_vg01_dg
   # symld -g client1_vg01_dg -sid 1983 add dev 033d
   # symld -g client1_vg01_dg -sid 1983 add dev 0341
   ```

   In this example, a device group client1_vg01_dg will be created and the appropriate volumes will be added to the device group.

2. Now display the configuration using the **symdg show** command:

   ```
   # symdg show client1_vg01_dg

   Group Name:  client1_vg01_dg

       Group Type                                   : REGULAR
       Device Group in GNS                          : No
       Valid                                        : Yes
       Symmetrix ID                                 : 000190101983
       Group Creation Time                          : Sat Oct 31 20:31:29 2009
       Vendor ID                                    : EMC Corp
       Application ID                               : SYMCLI

       Number of STD Devices in Group               :    2
       Number of Associated GK's                    :    0
       Number of Locally-associated BCV's           :    0
       Number of Locally-associated VDEV's          :    0
       Number of Locally-associated TGT's           :    0
       Number of Remotely-associated VDEV's(STD RDF):    0
       Number of Remotely-associated BCV's (STD RDF):    0
       Number of Remotely-associated TGT's(TGT RDF) :    0
       Number of Remotely-associated BCV's (BCV RDF):    0
       Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0
       Number of Remotely-assoc'd BCV's (Hop-2 BCV) :    0
       Number of Remotely-assoc'd VDEV's(Hop-2 VDEV):    0
       Number of Remotely-assoc'd TGT's (Hop-2 TGT) :    0

       Standard (STD) Devices (2):
           {
           ----------------------------------------------------------------
                                                    Sym               Cap
           LdevName         PdevName                Dev  Att. Sts     (MB)
           ----------------------------------------------------------------
           DEV001           N/A                     033D (M)  RW     17258
           DEV002           N/A                     0341 (M)  RW     17258
           }
   ```

3. Now you can create a TimeFinder relationship between the P-VOL and S-VOL using the **symclone** command:

```
# symclone -g client1_vg01_dg create -precopy DEV001 sym ld DEV002

Execute 'Create' operation for device 'DEV001'
in device group 'client1_vg01_dg' (y/[n]) ? y

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'client1_vg01_dg'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'client1_vg01_dg' paired with target device 'DEV002'.

# symclone -g client1_vg01_dg query


Device Group (DG) Name: client1_vg01_dg
DG's Type             : REGULAR
DG's Symmetrix ID     : 000190101983


         Source Device                   Target Device            State
Copy
--------------------------------- ---------------------------- ------------
----
               Protected Modified                Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  Tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------
----
DEV001    033D         0        0 DEV002    0341        0 XXX. Copied       100

Total           -------- --------                --------
  Track(s)             0        0                       0
  MB(s)              0.0      0.0                     0.0


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

4. Once the copy is complete, indicated by a status of Copied, use the **symclone** command to activate the target volume for host access:

```
# symclone -g client1_vg01_dg activate DEV001 sym ld DEV002

Execute 'Activate' operation for device 'DEV001'
in device group 'client1_vg01_dg' (y/[n]) ? y

'Activate' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'client1_vg01_dg'. Please wait...

'Activate' operation successfully executed for device 'DEV001'
in group 'client1_vg01_dg' paired with target device 'DEV002'.
```

Virtual I/O Server

With the successful completion of the copy operation, you are now able to present the new PV as a virtual device to client1 on the Virtual I/O Server by performing these steps:

1. Using the **chkdev** command, scan for the unique ID of the S-VOL, hdiskpower58, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 830341 | grep -p EMC
NAME:                hdiskpower58
IDENTIFIER:          1D06830341O9SYMMETRIX03EMCfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

   Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (0341).

2. View the current virtual device configuration of client1 (vhost0) using the **lsmap** command:

```
$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi1
Status                Available
LUN                   0x8100000000000000
Backing device        hdiskpower139
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L858

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdiskpower140
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L862
```

3. Now that you know that the S-VOL is hdiskpower58 on the Virtual I/O Server, you can now assign it to client1 (vhost0) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdiskpower58 -vadapter vhost0
vtscsi58 Available

$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi1
Status                Available
LUN                   0x8100000000000000
Backing device        hdiskpower139
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L858

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdiskpower140
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L862

VTD                   vtscsi58
Status                Available
LUN                   0xbb00000000000000
Backing device        hdiskpower58
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L180
```

Client1

At this point, you have successfully created a copy of client1, vg01, using EMC TimeFinder Clone and used the Virtual I/O Server to map the new virtual device to client1. Now it is time to scan for the new device and import the VG.

Perform these steps:

1. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk43       00c7086c72bafc21                    rootvg          active
hdisk0        00c7086ca22de0a3                    vg01            active
hdisk1        00c7086ca22de0a3                    vg01            active
```

2. In the output of the **lspv** command above, the Logical Volume Manager (LVM) has recognized the PV_ID. You need to tread carefully in the next few steps to clear the PV_ID using the **chdev** command:

   ```
   # lsvg
   rootvg
   vg01

   # chdev -l hdisk1 -a pv=clear
   hdisk1 changed

   # lspv
   hdisk43        00c7086c72bafc21                     rootvg         active
   hdisk0         00c7086ca22de0a3                     vg01           active
   hdisk1         none                                 None
   ```

   You have now cleared the PV_ID.

3. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on client1.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, although if f the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that a limit of 15 characters applies.

   Run the following command:

   ```
   # recreatevg -y vg02 -Y CL1 hdisk1
   vg02

   # lsvg
   rootvg
   vg01
   vg02

   # lsvg -l vg02
   vg02:
   LV NAME          TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   CL1lv01          jfs2       130     130     1    closed/syncd  N/A
   CL1loglv01       jfs2log    1       1       1    closed/syncd  N/A
   ```

4. Now you can edit /etc/filesystems, create the mountpoints, and mount the new Logical Volumes by using the **vi** command:

   ```
   # vi /etc/filesystems
   (Append the following)
   /CL1lv01:
           dev             = /dev/CL1lv01
           vfs             = jfs2
           log             = /dev/CL1loglv01
           mount           = true
           check           = false

   # mkdir /CL1lv01

   # mount /CL1lv01
   Replaying log for /dev/CL1lv01.
   ```

5. Your last step is to look at your replicated data using the **ls** command:

```
# ls -Rtl /CL1lv01
/CL1lv01:
total 0
drwx------    2 root     system          256 Nov 01 02:36 client1
drwxr-xr-x    2 root     system          256 Oct 29 16:32 lost+found

/CL1lv01/Client1:
total 1024000
-rw-------    1 root     system    524288000 Nov 01 02:26 file_client1

/CL1lv01/lost+found:
total 0
```

3.2 Hitachi ShadowImage

In this section, we discuss the steps that are required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using Hitachi ShadowImage.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and Logical Volume (LV) reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which, for the purposes of this demonstration, contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using Hitachi ShadowImage and mount it back to client1:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk0 and has two LVs, lv01 and loglv01:

```
# lspv
hdisk43         00c7086c72bafc21                    rootvg          active
hdisk0          00c7086c8321623a                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. View the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Nov 04 09:21 Client1
drwxr-xr-x    2 root     system          256 Nov 03 16:53 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 04 09:21 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. Knowing this unique ID is very important, as it represents the physical LUN within the Hitachi storage subsystem.

   Physical disk hdisk0 is the ShadowImage source or P-VOL.

   Run the following command:

```
# odmget -q "name=hdisk0 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk0"
        attribute = "unique_id"
        value = "3924240C50 0B0FA020006OPEN-V07HITACHIfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that you know the unique ID of hdisk0 within vg01 on client1, determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can determine that hdisk759 represents unique ID 240C50 0B0FA020006OPEN-V07HITACHIfcp within the Virtual I/O Server:

```
$ chkdev | grep -p 0B0FA020006OPEN-V07HITACHIfcp
NAME:                hdisk759
IDENTIFIER:          240C50 0B0FA020006OPEN-V07HITACHIfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:
>
> 3924240C50 **0B0FA020006OPEN-V07HITACHIfcp**05VDASD03AIXvscsi

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters that allow for a direct connection to SAN fabrics for the allocation of a Hitachi command device.

The server has Hitachi storage system Command Control Interface (CCI) installed.

> **Note:** Hitachi ShadowImage software uses the in-band Fibre Channel (FC) protocol to send commands to the storage system in order to create and manage HORCM associations.

Perform these steps:

1. Using the **`odmget CuAt`** command, identify the PV of unique ID 240C50 0B0FA020006OPEN-V07HITACHIfcp, which is the ShadowImage source or P-VOL.

   This command is particularly useful, as you can look and make sure that a Hitachi ShadowImage command device is allocated by scanning for a unique ID that contains OPEN-V-CM07.

   Physical disk hdisk144 is the ShadowImage command device.

   Physical disk hdisk140 is the ShadowImage source or P-VOL.

   hdisk142 (unique ID = 240C50 0B0FA020206OPEN-V07HITACHIfcp) is allocated to the ShadowImage management server, which is the ShadowImage target or S-VOL.

   Run the following command:

```
# odmget CuAt | grep -p HITACHI
CuAt:
        name = "hdisk140"
        attribute = "unique_id"
        value = "240C50 0B0FA020006OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk141"
        attribute = "unique_id"
        value = "240C50 0B0FA020106OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk142"
        attribute = "unique_id"
        value = "240C50 0B0FA020206OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk143"
        attribute = "unique_id"
        value = "240C50 0B0FA020306OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk144"
        attribute = "unique_id"
        value = "270C50 0B0FA01A309OPEN-V-CM07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk145"
        attribute = "unique_id"
        value = "240C50 0B0FA030006OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk467"
        attribute = "unique_id"
        value = "240C50 0B0FA031006OPEN-V07HITACHIfcp"
        type = "R"
        generic = ""
        rep = "nl"
        nls_index = 79
```

> **Note:** In this case, there is only one command device for ShadowImage operations. For more complicated environments, you may need to assign multiple command devices. Do not forget that ShadowImage and TrueCopy should not share the same command device.

2. Using the ShadowImage Command Control Interface (CCI), you can create the source and target Hitachi Online RAID Configuration Manager (HORCM) configuration files using the **mkconf** command.

   A single server configuration will be implemented in this example, which establishes a relationship between the device of HORCM instance 1 and the device of HORCM instance 4 on the same server.

   HORCM instance 1 is the P-VOL.

   HORCM instance 4 is the S-VOL.

   Run the following command:

```
# echo hdisk140 | /HORCM/usr/bin/mkconf.sh -g Client1 -i 1 -s 5001
starting HORCM inst 1
HORCM inst 1 starts successfully.
HORCM Shutdown inst 1 !!!
A CONFIG file was successfully completed.
starting HORCM inst 1
HORCM inst 1 starts successfully.
DEVICE_FILE            Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk140               Client1  Client1_000 CL7-F     0    0 -    45306   512
HORCM Shutdown inst 1 !!!
```

```
Please check '/apps/horcm1.conf','/apps/log1/curlog/horcm_*.log', and modify
'ip_address & service'.

# echo hdisk142 | /HORCM/usr/bin/mkconf.sh -g Client1 -i 4 -s 54004
starting HORCM inst 4
HORCM inst 4 starts successfully.
HORCM Shutdown inst 4 !!!
A CONFIG file was successfully completed.
starting HORCM inst 4
HORCM inst 4 starts successfully.
DEVICE_FILE             Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk142                Client1  Client1_000 CL7-F     0    3 -    45306   515
HORCM Shutdown inst 4 !!!
Please check '/apps/horcm4.conf','/apps/log4/curlog/horcm_*.log', and modify
'ip_address & service'.
```

3. Edit the HORCM configuration files and change the IP address of the HORCM_MON and HORCM_INST stanzas to address of the localhost. Change the service of the HORCM_INST stanzas to point at each other (that is, horcm1 = 54001 and horcm4 = 54004):

```
# vi horcm1.conf
# vi horcm4.conf
# vi /etc/services
```

Note: The following is an example S-VOL HORCM configuration file that was used for this step of the procedure:

```
# Created by mkconf.sh on Mon Oct 26 10:31:35 CDT 2009
HORCM_MON
#ip_address        service         poll(10ms)     timeout(10ms)
9.3.92.181         54004                  1000              3000
HORCM_CMD
#dev_name               dev_name                dev_name
#UnitID 0 (Serial# 45306)
/dev/rhdisk144
HORCM_DEV
#dev_group         dev_name        port#     TargetID     LU#     MU#
# hdisk142             SER =    45306  LDEV =  515 [ FIBRE FCTBL = 3 ]
Client1            Client1_000     CL7-F            0       3
HORCM_INST
#dev_group        ip_address       service
Client1           9.3.92.181       54001
```

4. Set up your environment variables and start the two HORCM instances:

```
# export HORCMINST=1

# export HORCC_MRCF=1

# horcmstart.sh
starting HORCM inst 1
HORCM inst 1 starts successfully.

# export HORCMINST=4

# horcmstart.sh
starting HORCM inst 4
HORCM inst 4 starts successfully.
```

> **Note:** The environment variable set here, HORCC_MRCF=1, needs to be set in order to specify ShadowImage functions. If it is not set, TrueCopy functions will be performed instead.

5. Once the HORCM instances start successfully, you can display the configuration using the **pairdisplay** command:

```
# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 SMPL     -     - - -
Client1 Client1_000 R   hdisk140       0  45306   200 SMPL     -     - - -
```

   You are now ready to create a ShadowImage relationship between the P-VOL and S-VOL. This is a very destructive command, so follow this simple procedure, which will obviate the need to restore from tape: When creating a relationship, always use the HORCM instance of the S-VOL, which means that when you specify the relationship copy direction, you will always use the -vr switch (that is, -vr for remote or -vl for local).

   In this example, a concurrent track copy count of 15 was used.

6. Using the **echo** command, verify the HORCM instance you currently have configured:

```
# echo $HORCMINST
4

# paircreate -g Client1 -c 15 –vr

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 S-VOL COPY     3 200 -
Client1 Client1_000 R   hdisk140       0  45306   200 P-VOL COPY     3 202 –

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 S-VOL PAIR   100 200 -
Client1 Client1_000 R   hdisk140       0  45306   200 P-VOL PAIR   100 202 –
```

7. Once the copy is complete (indicated by a status of PAIR), use the **pairsplit** command to remove the relationship:

```
# pairsplit -S -g Client1

# pairdisplay -fxcd -CLI -g Client1
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client1 Client1_000 L   hdisk142       0  45306   202 SMPL     -     - - -
Client1 Client1_000 R   hdisk140       0  45306   200 SMPL     -     - - -
```

Virtual I/O Server

With the successful completion of the copy operation, you are now able to present the new PV as a virtual device to client1 on the Virtual I/O Server.

The following steps import the VG into the Virtual I/O Server and present the VG as a virtual device to client1:

1. Using the **chkdev** command, scan for the unique ID of the S-VOL 240C50 0B0FA020206OPEN-V07HITACHIfcp, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p "240C50 0B0FA020206OPEN-V07HITACHIfcp"
NAME:                hdisk761
IDENTIFIER:          240C50 0B0FA020206OPEN-V07HITACHIfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

2. View the current virtual device configuration of client1 (vhost0) using the **lsmap** command:

```
$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi2
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk759
Physloc               U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L0

VTD                   vtscsi58
Status                Available
LUN                   0xbb00000000000000
Backing device        hdiskpower58
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L180
```

3. Because you know that the S-VOL is hdisk761 on the Virtual I/O Server, you can now assign it to client1 (vhost0) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk761 -vadapter vhost0
vtscsi4 Available

$ lsmap -vadapter vhost0
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi2
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk759
Physloc               U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L0

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk761
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L2000000000000

VTD                   vtscsi58
Status                Available
LUN                   0xbb00000000000000
Backing device        hdiskpower58
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L180
```

Client1

At this point, you have successfully created a copy of client1, vg01, using Hitiachi ShadowImage and used the Virtual I/O Server to map the new virtual device to client1. Now it is time to scan for the new device and import the VG.

Perform these steps:

4. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk43         00c7086c72bafc21                    rootvg          active
hdisk0          00c7086c8321623a                    vg01            active
hdisk1          00c7086c8321623a                    vg01            active
```

5. In the output of the **lspv** command above, the Logical Volume Manager (LVM) has recognized the PV_ID. You need to tread carefully when using the next few steps to clear the PV_ID using the **chdev** command:

```
# lsvg
rootvg
vg01

# chdev -l hdisk1 -a pv=clear
hdisk1 changed

# lspv
hdisk43         00c7086c72bafc21                    rootvg          active
hdisk0          00c7086c8321623a                    vg01            active
hdisk1          none                                None
```

   You have now cleared the PV_ID.

6. Using the **recreatevg** command, import the Volume Group, making sure to rename it with a name that does not already exist on client1.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, make sure you remember a limit of 15 characters applies.

   Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk1
vg02

# lsvg
rootvg
vg01
vg02

# lsvg -l vg02
vg02:
LV NAME         TYPE       LPs      PPs      PVs  LV STATE      MOUNT POINT
CL1lv01            jfs2       150      150      1    open/syncd    N/A
CL1loglv01         jfs2log    1        1        1    open/syncd    N/A
```

7. Now you can edit /etc/filesystems, create the mountpoints, and mount the new Logical Volumes by using the **vi** command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv01:
    dev     = /dev/CL1lv01
    vfs     = jfs2
    log     = /dev/CL1loglv01
    mount       = true
    check       = false

# mkdir /CL1lv01

# mount /CL1lv01
Replaying log for /dev/CL1lv01.
```

8. Your last step is to look at your replicated data using the **ls** command:

```
# ls -ltR /CL1lv01
/CL1lv01:
total 0
drwx------    2 root     system          256 Nov 04 09:21 Client1
drwxr-xr-x    2 root     system          256 Nov 03 16:53 lost+found

/CL1lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 04 09:21 This is Client1

/CL1lv01/lost+found:
total 0
```

3.3 IBM FlashCopy: IBM System Storage DS8300

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using IBM FlashCopy.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and Logical Volume (LV) reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using IBM FlashCopy and mount it back to client1.

1. Using the **lspv** and **lsvg** commands, you determine that vg01 resides on hdisk0 and has two LVs, lv01 and loglv01:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe4111db6f92c                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. View the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Nov 02 09:32 Client1
drwxr-xr-x    2 root     system          256 Nov 01 16:59 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 02 09:32 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. Knowing this unique ID is very important, as it represents the physical LUN within the IBM storage subsystem.

   Physical disk hdisk1 is the FlashCopy source volume.

   Run the following command:

```
# odmget -q "name=hdisk1 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk1"
        attribute = "unique_id"
        value = "3520200B75BALB1100D07210790003IBMfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that you know the unique ID of hdisk1 within vg01 on client1, determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can determine that hdisk8 represents unique ID 3520200B75BALB1100D07210790003IBMfcp05VDASD03AIXvscsi within the Virtual I/O Server:

```
$ chkdev | grep -p 200B75BALB1100D07210790003IBMfcp
NAME:                hdisk8
IDENTIFIER:          200B75BALB1100D07210790003IBMfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:
>
> 3520**200B75BALB1100D07210790003IBMfcp**05VDASD03AIXvscsi

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client. The server has IBM DSCLI installed, which is used to manage storage subsystem functions.

Perform these steps:

1. Using the **odmget CuAt** command, look for the unique ID 200B75BALB1100D07210790003IBMfcp, which is the FlashCopy source volume.

```
# odmget CuAt | grep -p IBMfcp
CuAt:
        name = "hdisk6"
        attribute = "unique_id"
        value = "3E213600A0B8000291B0800009DCB0402FC540F1815
FAStT03IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk7"
        attribute = "unique_id"
        value = "3E213600A0B8000291B0800009DCC0402FC6C0F1815
FAStT03IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk8"
        attribute = "unique_id"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk9"
        attribute = "unique_id"
        value = "200B75BALB1100E07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk10"
        attribute = "unique_id"
        value = "200B75BALB1110707210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk11"
        attribute = "unique_id"
        value = "200B75BALB1110807210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "vtscsi2"
        attribute = "udid_info"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0
```

Physical disk hdisk8 is the FlashCopy source volume.

hdisk11 (unique ID = 200B75BALB1110807210790003IBMfcp) is allocated to the FlashCopy management server, which is the FlashCopy target volume.

2. Using the DSCLI **lshostvol** command, determine the IBM DS8300 vpath ID of hdisk10 and hdisk11:

```
dscli> lshostvol
Date/Time: November 2, 2009 12:09:47 AM CST IBM DSCLI Version: 5.4.1.44 DS: -
Disk Name Volume Id             Vpath Name
==========================================
hdisk8    IBM.2107-75BALB1/100D ---
hdisk9    IBM.2107-75BALB1/100E ---
hdisk10   IBM.2107-75BALB1/1107 ---
hdisk11   IBM.2107-75BALB1/1108 ---
```

3. You are now ready to create the FlashCopy relationship between the source (vpath ID = 100D) and target (vpath ID = 1108) volumes using your FlashCopy management server and the **mkflash** command:

```
dscli> mkflash -dev IBM.2107-75BALB1 -persist -cp -seqnum 0001 100D:1108
Date/Time: November 2, 2009 12:10:09 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00137I mkflash: FlashCopy pair 100D:1108 successfully created.
```

4. Display the configuration using the **lsflash** command:

```
dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 12:10:51 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                    State AllowTgtSE
===============================================================================
===============================================================================
=================================================================
100D:1108 10     1           120     Enabled    Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        305238          Mon Nov 02
10:28:00 CST 2009 Mon Nov 02 10:28:00 CST 2009 Valid Disabled

dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 12:13:42 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                    State AllowTgtSE
===============================================================================
===============================================================================
=================================================================
100D:1108 10     1           120     Disabled   Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        0               Mon Nov 02
10:28:00 CST 2009 Mon Nov 02 10:28:00 CST 2009 Valid Disabled
```

5. Once the copy operation is complete, indicated by the 0 in OutOfSyncTracks, use the **rmflash** command to remove the relationship:

```
dscli> rmflash -dev IBM.2107-75BALB1 -seqnum 0001 100D:1108
Date/Time: November 2, 2009 12:24:32 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00144W rmflash: Are you sure you want to remove the FlashCopy pair
100D:1108:? [y/n]:y
CMUC00140I rmflash: FlashCopy pair 100D:1108 successfully removed.

dscli> lsflash -dev IBM.2107-75BALB1 -l 100D:1108
Date/Time: November 2, 2009 12:25:15 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00234I lsflash: No Flash Copy found
```

Virtual I/O Server

Now that the copy operation is complete, you need to present the new PV as a virtual device to client1 by performing these steps:

1. Using the **chkdev** command, scan for the unique ID of the FlashCopy target volume, 200B75BALB1110807210790003IBMfcp, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 200B75BALB1110807210790003IBMfcp
NAME:                hdisk11
IDENTIFIER:          200B75BALB1110807210790003IBMfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

2. View the current virtual device configuration of client1 (vhost2) using the **lsmap** command:

```
$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk8
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4010400D00000000
```

3. Now that you know that the FlashCopy target volume is hdisk11 on the Virtual I/O Server, you can now assign it to client1 (vhost2) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk11 -vadapter vhost2
vtscsi3 Available

$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk8
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4010400D00000000

VTD                   vtscsi3
Status                Available
LUN                   0x8300000000000000
Backing device        hdisk11
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4011400800000000
```

Client1

At this point, you have successfully created a copy of client1, vg01, using IBM FlashCopy and used the Virtual I/O Server to map the new virtual device to client1. Now it is time to scan for the new device and import the Volume Group.

Perform these steps:

4. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe4111db6f92c                    vg01            active
hdisk2          000fe4111db6f92c                    vg01            active
```

5. In the output of the **lspv** command above, the Logical Volume Manager (LVM) has recognized the PV_ID. You need to tread carefully when performing the next few steps to clear the PV_ID using the **chdev** command:

```
# lsvg
rootvg
vg01

# chdev -l hdisk2 -a pv=clear
hdisk2 changed

# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe4111db6f92c                    vg01            active
hdisk2          none                    None
```

   You have now cleared the PV_ID.

6. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on client1.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk2
vg02

# lsvg
rootvg
vg01
vg02

# lsvg -l vg02
vg02:
LV NAME        TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL1lv01           jfs2       150     150     1    open/syncd    N/A
CL1loglv01        jfs2log    1       1       1    open/syncd    N/A
```

7. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new Logical Volumes by using the **`vi`** command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv01:
    dev     = /dev/CL1lv01
    vfs     = jfs2
    log     = /dev/CL1loglv01
    mount       = true
    check       = false

# mkdir /CL1lv01

# mount /CL1lv01
Replaying log for /dev/CL1lv01.
```

8. Your last step is to look at your replicated data using the **`ls`** command:

```
# ls -ltR /CL1lv01
/CL1lv01:
total 0
drwx------    2 root     system          256 Nov 02 09:32 Client1
drwxr-xr-x    2 root     system          256 Nov 01 16:59 lost+found

/CL1lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 02 09:32 This is Client1

/CL1lv01/lost+found:
total 0
```

3.4 IBM FlashCopy: SAN Volume Controller (SVC)

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Physical Volume (PV) using IBM FlashCopy.

The major steps include:

- Determine the client PV of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and Logical Volume (LV) reside

Client1

Client1 has a secondary Volume Group (VG), vg01, which, for the purposes of this demonstration, contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg01 using IBM FlashCopy and mount it on client2:

1. Using the **lspv** and **lsvg** commands, you can determine that vg01 resides on hdisk1 and has two LVs, lv01 and loglv01:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe41141804594                    vg01            active

# lsvg
rootvg
vg01

# lsvg -l vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs2       150     150     1    open/syncd    /lv01
loglv01             jfs2log    1       1       1    open/syncd    N/A
```

2. View the data that will be copied using the **ls** command:

```
# ls -ltR /lv01
/lv01:
total 0
drwx------    2 root     system          256 Nov 05 09:39 Client1
drwxr-xr-x    2 root     system          256 Nov 04 10:38 lost+found

/lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 02 09:39 This is Client1

/lv01/lost+found:
total 0
```

3. Using the installed ODM package and the **odmget** command, you can determine the unique ID of the PV. It is very important to know this unique ID, as it represents the physical LUN presented by way of the IBM SAN Volume Controller (SVC).

   Physical disk hdisk1 will be the FlashCopy source volume.

   Run the following command:

```
# odmget -q "name=hdisk1 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk1"
        attribute = "unique_id"
        value =
"48333321360050768018300FD500000000000000004214503IBMfcp05VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Virtual I/O Server

Now that you know that the unique ID is hdisk1 within vg01 on client1, determine which virtual device it represents on the Virtual I/O Server.

Using the **chkdev** command, you can see that hdisk12 represents unique ID 4833332136005076801830OFD5000000000000000004214503IBMfcp05VDASD03AIXvscsi within the Virtual I/O Server:

```
$ chkdev | grep -p 33213600507680183000FD5000000000000000004214503IBMfcp
NAME:                hdisk12
IDENTIFIER:          33213600507680183000FD5000000000000000004214503IBMfcp
PHYS2VIRT_CAPABLE:   NA
VIRT2NPIV_CAPABLE:   YES
VIRT2PHYS_CAPABLE:   YES
```

> **Note:** Only part of the unique ID provided in the output of the **odmget** command is required to be captured in order to use the **chkdev** command on the Virtual I/O Server, as indicted by the bolded text within the following output:
>
> 4833**33213600507680183000FD5000000000000000004214503IBMfcp**05VDASD03AIXvscsi

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client. The server has an SSH client installed that is used to manage IBM SAN Volume Controller (SVC) functions.

Perform these steps:

1. Using the **svcinfo lshostvdiskmap** command, look for the component of the unique ID that contains the SAN Volume Controller (SVC) vdisk_UID 6005076801830OFD5000000000000000, which is the FlashCopy source volume:

   ```
   IBM_2145:admin>svcinfo lshostvdiskmap
   id              name             SCSI_id        vdisk_id       vdisk_name
   wwpn             vdisk_UID
   0               host0             0              0              host0_1
   10000000C9738E84 60050768018300FD5000000000000000
   0               host0             1              1              host0_2
   10000000C9738E84 60050768018300FD5000000000000001
   0               host0             2              2              host0_3
   10000000C9738E84 60050768018300FD5000000000000002
   0               host0             3              3              host0_4
   10000000C9738E84 60050768018300FD5000000000000003
   0               host0             4              4              host0_5
   10000000C9738E84 60050768018300FD5000000000000004
   ```

   In this example, host0 is the Virtual I/O Server where all three clients are allocated.

   Physical disk host0_1 (unique ID = 60050768018300FD5000000000000000) is the FlashCopy source volume.

   Physical disk host0_5 (unique ID = 60050768018300FD5000000000000004) is the FlashCopy target volume.

2. You are now ready to create the FlashCopy relationship between the source and target volumes using your FlashCopy management server and the **mkfcconsistgrp** and **mkfcmap** commands. Run the following commands:

```
IBM_2145:admin>svctask mkfcconsistgrp -name host0
FlashCopy Consistency Group, id [1], successfully created

IBM_2145:admin>svctask mkfcmap -source host0_1 -target host0_5 -name host0
-consistgrp host0
FlashCopy Mapping, id [0], successfully created

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

3. Once the FlashCopy relationship is created, prepare the consistency group using the **prestartfcconsistgrp** command:

```
IBM_2145:admin>svctask prestartfcconsistgrp host0

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status prepared
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

4. Now start the FlashCopy using the **startfcconsistgrp** command:

```
IBM_2145:admin>svctask startfcconsistgrp host0

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
```

```
target_vdisk_name host0_5
group_id 1
group_name host0
status copying
progress 0
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_1
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 100
copy_rate 50
start_time 091106173350
dependent_mappings 0
autodelete off
```

5. Once the copy operation is complete, which is indicated by a progress of 100, and a status of idle_or_copied, use the **rmfcmap** command to remove the relationship:

```
IBM_2145:admin>svctask rmfcmap host0

IBM_2145:admin>svcinfo lsfcmap host0
CMMVC5754E The object specified does not exist, or the name supplied does not
meet the naming rules.
```

Virtual I/O Server

With the copy operation now complete, you need to present the new PV as a virtual device to client1 by performing the following steps:

1. Using the **chkdev** command, scan for the unique ID of the FlashCopy target volume, 60050768018300FD5000000000000004, so you can present the physical disk as a virtual device:

```
$ chkdev | grep -p 60050768018300FD5000000000000004
NAME:                hdisk16
IDENTIFIER:          3321360050768018300FD500000000000000404214503IBMfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

2. Now view the current virtual device configuration of client1 (vhost2) using the **lsmap** command:

```
$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk12
Physloc               U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L0
```

3. Now that you know that the FlashCopy target volume is hdisk16 on the Virtual I/O Server, you can now assign it to client1 (vhost2) as a virtual device using the **mkvdev** command:

```
$ mkvdev -vdev hdisk16 -vadapter vhost2
vtscsi6 Available

$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi2
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk12
Physloc               U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L0

VTD                   vtscsi6
Status                Available
LUN                   0x8300000000000000
Backing device        hdisk16
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L4000000000000
```

Client1

At this point, you have successfully created a copy of client1, vg01, using your IBM FlashCopy and used the Virtual I/O Server to map the new virtual device to client1. Now it is time to scan for the new device and import the Volume Group.

Perform these steps:

1. Using the **cfgmgr** command, scan for the new device:

   ```
   # cfgmgr

   # lspv
   hdisk0          000fe411206febdd                    rootvg          active
   hdisk1          000fe41141804594                    vg01            active
   hdisk2          000fe41141804594                    vg01            active
   ```

2. In the output of the **lspv** command above, the Logical Volume Manager (LVM) has recognized the PV_ID. You need to tread carefully when performing the next few steps to clear the PV_ID using the **chdev** command:

   ```
   # lsvg
   rootvg
   vg01

   # chdev -l hdisk2 -a pv=clear
   hdisk2 changed

   # lspv
   hdisk0          000fe411206febdd                    rootvg          active
   hdisk1          000fe41141804594                    vg01            active
   hdisk2          none                                None
   ```

   You have now cleared the PV_ID.

3. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on client1.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that a limit of 15 characters applies.

   ```
   # recreatevg -y vg02 -Y CL1 hdisk2
   vg02

   # lsvg
   rootvg
   vg01
   vg02

   # lsvg -l vg02
   vg02:
   LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   CL1lv01             jfs2       150     150     1    closed/syncd  N/A
   CL1loglv01          jfs2log    1       1       1    closed/syncd  N/A
   ```

4. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new Logical Volumes with the vi command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv01:
    dev     = /dev/CL1lv01
    vfs     = jfs2
    log     = /dev/CL1loglv01
    mount       = true
    check       = false

# mkdir /CL1lv01

# mount /CL1lv01
Replaying log for /dev/CL1lv01.
```

5. Your last step is to look at your replicated data using the **`ls`** command:

```
# ls -ltR /CL1lv01
/CL1lv01:
total 0
drwx------    2 root     system          256 Nov 05 09:39 Client1
drwxr-xr-x    2 root     system          256 Nov 04 10:38 lost+found

/CL1lv01/Client1:
total 1024000
-rw-------    1 root     system    536870912 Nov 02 09:39 This is Client1

/CL1lv01/lost+found:
total 0
```

4

Scenario #3: Copy of physical LUN with multiple Logical Volumes and present to another client

In this scenario, there are three IBM PowerVM clients: client1, client2, and client3. Client1 and client2 each have one Volume Group (VG), vg01, which contains the data to be replicated using Copy Services.

Each VG is a different Virtual I/O Server Logical Volume (LV) mapped as a virtual SCSI device to the clients.

A copy will be made of both clients data using the three Copy Services products:

- EMC TimeFinder
- Hitachi ShadowImage
- IBM FlashCopy

The copies will be mapped, discovered, and imported on a different client (client3) as new VGs. The LVs, lv01 of each client, and their associated Logical Volume Manager (LVM) log volumes, loglv01 of each client, will mount as new file systems.

This process is typically used for regular split-mirror backups using a dedicated backup server or mount host. This function allows for complete separation of CPU, memory, disk, and tape resources during backup processes, with a minimal disruption to primary disk I/O during the initial Copy Services operation.

4.1 EMC TimeFinder

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Logical Volume (LV) using EMC TimeFinder.

This example uses a LV virtual SCSI device presented from the Virtual I/O Server to clients client1 and client2.

The major steps include:

- Determine the client Physical Volume (PV) of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the Volume Group (VG) and LV reside
- Determine the LV on the Virtual I/O Server that is presented to the client where a VG and LV reside

Client1

Client1 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using EMC TimeFinder and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk0:

```
# lspv
hdisk43       00c7086c72bafc21                    rootvg          active
hdisk0        00c7086c8321623a                    vg01            active
hdisk1        00c7086cb1d4b327                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk1 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk1"
        attribute = "unique_id"
        value = "372200c7086c00004c0000000124b1c9e9ea.105VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk0 is the VG ID of the volume group from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 33722**00c7086c00004c0000000124b1c9e9ea.1**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       60      60      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -Rtl /lv02
/lv02:
total 0
drwx------    2 root     system          256 Nov 02 18:27 client1
drwxr-xr-x    2 root     system          256 Nov 01 16:25 lost+found

/lv02/client1:
total 1024000
-rw-------    1 root     system    524288000 Nov 02 18:27 file_client1

/lv02/lost+found:
total 0
```

Client2

Client2 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using EMC TimeFinder and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk1:

```
# lspv
hdisk0         00c7086c77cce272                    rootvg          active
hdisk1         00c7086c831f9c2a                    vg01            active
hdisk2         00c7086cb1dd5f95                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "372200c7086c00004c0000000124b1c9e9ea.205VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk1 is the VG ID of the volume group from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 33722**00c7086c00004c0000000124b1c9e9ea.2**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       60      60      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -Rtl /lv02
/lv02:
total 0
drwx------    2 root     system          256 Nov 02 18:29 client2
drwxr-xr-x    2 root     system          256 Nov 01 16:33 lost+found

/lv02/client2:
total 1024000
-rw-------    1 root     system    524288000 Nov 02 18:30 file_client2

/lv02/lost+found:
total 0
```

Virtual I/O Server

Now that you know the unique IDs of hdisk2 on client1 (vhost2) and hdisk2 on client2 (vhost0), map to the virtual devices they represent on the Virtual I/O Server by performing these steps:

1. Using the **lsmap** command, determine that the mapped virtual devices are LVs:

```
$ lsmap -vadapter vhost0 -type lv
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi1
Status                Available
LUN                   0x8100000000000000
Backing device        lv01
Physloc

$ lsmap -vadapter vhost1 -type lv
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi2
Status                Available
LUN                   0x8100000000000000
Backing device        lv02
Physloc
```

2. You now know the names of the mapped LVs and can use the **lspv** and **lsvg** commands to determine what VG they are members of and the physical devices in which they reside:

```
$ lsvg
rootvg
vg01

$ lspv | grep vg01
hdiskpower139    00c7086ca22de0a3                    vg01            active

$ lsvg -lv vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs        537     537     1    open/syncd    N/A
lv02                jfs        537     537     1    open/syncd    N/A
```

3. Using the **chkdev** command, determine that hdiskpower139 represents unique ID `1D0683033D09SYMMETRIX03EMCfcp` within the Virtual I/O Server. Physical disk hdiskpower139 is the TimeFinder source volume. Run the following command:

```
$ chkdev | grep -p hdiskpower139
NAME:                hdiskpower139
IDENTIFIER:          1D0683033D09SYMMETRIX03EMCfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

   Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (033D).

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters that allow for a direct connection to SAN fabrics for the allocation of EMC gatekeeper devices for array management. This server has EMC Solutions Enabler (SYMCLI) installed.

For this exercise, Symmetrix volume 033D will be the TimeFinder source volume and volume 0341 will be TimeFinder target volume.

Perform these steps:

1. Using the SYMCLI interface and the **symdg** and **symld** commands, create the Symmetrix Device Group to be used to control the TimeFinder Clone operations against the source and target volumes.

   In this example, a device group sharedvg_c1c2_dg will be created and the appropriate volumes will be added to the device group.

   Run the following command:

```
# symdg create sharedvg_c1c2_dg
# symld -g sharedvg_c1c2_dg -sid 1983 add dev 033d
# symld -g sharedvg_c1c2_dg -sid 1983 add dev 0341
```

2. Display the configuration by using the **symdg show** command:

```
# symdg show sharedvg_c1c2_dg

Group Name: sharedvg_c1c2_dg

    Group Type                                   : REGULAR
    Device Group in GNS                          : No
    Valid                                        : Yes
    Symmetrix ID                                 : 000190101983
    Group Creation Time                          : Sat Oct 31 20:31:29 2009
    Vendor ID                                    : EMC Corp
    Application ID                               : SYMCLI

    Number of STD Devices in Group               :    2
    Number of Associated GK's                    :    0
    Number of Locally-associated BCV's           :    0
    Number of Locally-associated VDEV's          :    0
    Number of Locally-associated TGT's           :    0
    Number of Remotely-associated VDEV's(STD RDF):    0
    Number of Remotely-associated BCV's (STD RDF):    0
    Number of Remotely-associated TGT's(TGT RDF) :    0
    Number of Remotely-associated BCV's (BCV RDF):    0
    Number of Remotely-assoc'd RBCV's (RBCV RDF) :    0
    Number of Remotely-assoc'd BCV's (Hop-2 BCV) :    0
    Number of Remotely-assoc'd VDEV's(Hop-2 VDEV):    0
    Number of Remotely-assoc'd TGT's (Hop-2 TGT) :    0

    Standard (STD) Devices (2):
        {
        ----------------------------------------------------------------
                                                 Sym               Cap
        LdevName         PdevName                Dev  Att. Sts     (MB)
        ----------------------------------------------------------------
        DEV001           N/A                     033D (M)  RW     17258
        DEV002           N/A                     0341 (M)  RW     17258
        }
```

3. You are now ready to create a TimeFinder relationship between the source volume and target volume by using the **symclone** command:

```
# symclone -g sharedvg_c1c2_dg create -precopy DEV001 sym ld DEV002

Execute 'Create' operation for device 'DEV001'
in device group 'sharedvg_c1c2_dg' (y/[n]) ? y

'Create' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'sharedvg_c1c2_dg'. Please wait...

'Create' operation successfully executed for device 'DEV001'
in group 'sharedvg_c1c2_dg' paired with target device 'DEV002'.

# symclone -g sharedvg_c1c2_dg query


Device Group (DG) Name: sharedvg_c1c2_dg
```

```
DG's Type             : REGULAR
DG's Symmetrix ID     : 000190101983


         Source Device                  Target Device           State
Copy
--------------------------------- ---------------------------- ------------
----
               Protected Modified                Modified
Logical   Sym  Tracks    Tracks   Logical   Sym  Tracks   CGDP SRC <=> TGT  (%)
--------------------------------- ---------------------------- ------------
----
DEV001    033D         0        0 DEV002    0341        0 XXX. Copied       100

Total           -------- --------                --------
  Track(s)             0        0                       0
  MB(s)              0.0      0.0                     0.0


Legend:

(C): X = The background copy setting is active for this pair.
     . = The background copy setting is not active for this pair.
(G): X = The Target device is associated with this group.
     . = The Target device is not associated with this group.
(D): X = The Clone session is a differential copy session.
     . = The Clone session is not a differential copy session.
(P): X = The pre-copy operation has completed one cycle
     . = The pre-copy operation has not completed one cycle
```

4. Once the copy is complete, indicated by a status of Copied, use the **symclone** command to activate the target volume for host access:

```
# symclone -g sharedvg_c1c2_dg activate DEV001 sym ld DEV002

Execute 'Activate' operation for device 'DEV001'
in device group 'sharedvg_c1c2_dg' (y/[n]) ? y

'Activate' operation execution is in progress for device 'DEV001'
paired with target device 'DEV002' in
device group 'sharedvg_c1c2_dg'. Please wait...

'Activate' operation successfully executed for device 'DEV001'
in group 'sharedvg_c1c2_dg' paired with target device 'DEV002'.
```

Virtual I/O Server

When the copy operation successfully completes, you can present the new PV back to the Virtual I/O Server.

The following steps can be performed to import the VG onto the Virtual I/O Server and present the VG as a virtual device to client3:

1. When logged into the Virtual I/O Server as the padmin user, you are operating in a restricted shell. If you want to perform a **cfgmgr** command to scan for the new device, you need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # cfgmgr

   # exit
   ```

2. Using the **chkdev** command, scan for the unique ID of the target volume, hdiskpower140, so that you can determine the new physical device name and import the VG:

   ```
   $ chkdev | grep -p 830341
   NAME:                hdiskpower140
   IDENTIFIER:          1D068303410 9SYMMETRIX03EMCfcp
   PHYS2VIRT_CAPABLE:   YES
   VIRT2NPIV_CAPABLE:   NA
   VIRT2PHYS_CAPABLE:   NA

   $ lspv | grep hdiskpower140
   hdiskpower140    00c7086caf002b48                     vg01            active
   ```

   Using the output that is highlighted in bold above, you can use this information to determine the EMC storage subsystem and LUN within that storage system for the PV. The first two digits represent the last two digits of the EMC storage subsystem serial number (83), while the last four digits identify the Symmetrix Device ID of the LUN (0341).

3. In the output of the **lspv** command, you can determine that LVM has recognized the PV_ID. You need to tread carefully in the next step when you clear the PV_ID using the **chdev** command.

   When logged into the Virtual I/O Server using the padmin user, you are operating in a restricted shell. In order to perform a **chdev** command with the appropriate switches to clear the PV_ID, you will need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # chdev -l hdiskpower140 -a pv=clear
   hdiskpower140 changed

   # lspv | grep hdiskpower140
   hdiskpower140   none                                None
   ```

   You have now cleared the PV_ID.

4. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on the Virtual I/O Server.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

Run the following command:

```
# recreatevg -y vg02 hdiskpower140
vg02
```

5. You can now return back to the Virtual I/O Server restricted shell and continue with the final steps. Run the following command;

```
# exit

$ lsvg
rootvg
vg01
vg02

$ lsvg -lv vg02
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv01              jfs        537     537     1    closed/syncd  N/A
fslv02              jfs        537     537     1    closed/syncd  N/A
```

6. View the current virtual device configuration of client3 (vhost2) by using the **lsmap** command:

```
$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U9117.MMA.107086C-V1-C22                     0x0000003f

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        hdiskpower68
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L70

VTD                   vtscsi3
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk762
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L3000000000000
```

7. Using the **mkvdev** command, map the new LVs on VG02 to client3 (vhost2):

```
$ mkvdev -vdev fslv01 -vadapter vhost2
vtscsi4 Available

$ mkvdev -vdev fslv02 -vadapter vhost2
vtscsi5 Available
```

Client3

At the point, you have successfully created a copy of the vg02 VG of client1 and the vg02 VG of client2 using EMC TimeFinder. You have also successfully mapped the virtual devices to the VG on the Virtual I/O Server. Now it is time to scan for the new device and import the VG.

Perform these steps:

1. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk0          00c7086c7e15d10f                    rootvg          active
hdisk1          00c7086cb1d4b327                    vg01
hdisk4          00c7086c97f77df2                    None
hdisk5          00c7086c97f7f9f7                    None

# lsvg
rootvg
vg01
```

2. Using the **recreatevg** command, import the VGs (VG), making sure you rename them with new names that do not already exist on client3.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the Virtual I/O Server, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk1
vg02

# recreatevg -y vg03 -Y CL2 hdisk2
vg03

# lsvg
rootvg
vg01
vg02
vg03

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL1lv02             jfs2       60      60      1    closed/syncd  N/A
CL1loglv02          jfs2log    1       1       1    closed/syncd  N/A

# lsvg -l vg03
vg03:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL2lv02             jfs2       60      60      1    closed/syncd  N/A
CL2loglv02          jfs2log    1       1       1    closed/syncd  N/A
```

3. Now you can edit /etc/filesystems, create the mountpoints, and mount the new LVs by using the **vi** command;

```
# vi /etc/filesystems
(Append the following)
/Client1/lv02:
        dev             = /dev/CL1lv02
        vfs             = jfs2
        log             = /dev/CL1loglv02
        mount           = true
        account         = false

/Cleint2/lv02:
        dev             = /dev/CL2lv02
        vfs             = jfs2
        log             = /dev/CL2loglv02
        mount           = true
        account         = false

# mkdir /Client1/lv02

# mkdir /Client2/lv02

# mount /CL1lv02
Replaying log for /dev/CL1lv02.

# mount /CL2lv02
Replaying log for /dev/CL2lv02.
```

4. Your last step is to look at your replicated data by using the **ls** command:

```
# ls -Rtl /Client1
total 0
drwxr-xr-x    4 root     system          256 Nov 02 18:27 lv02
/Client1/lv02:
total 0
drwx------    2 root     system          256 Nov 02 18:27 client1
drwxr-xr-x    2 root     system          256 Nov 01 16:25 lost+found

/Client1/lv02/client1:
total 1024000
-rw-------    1 root     system    524288000 Nov 02 18:27 file_client1

/Client1/lv02/lost+found:
total 0

# ls -Rtl /Client2
total 0
drwxr-xr-x    4 root     system          256 Nov 02 18:29 lv02
/Client2/lv02:
total 0
drwx------    2 root     system          256 Nov 02 18:29 client2
drwxr-xr-x    2 root     system          256 Nov 01 16:33 lost+found

/Client2/lv02/client2:
total 1024000
-rw-------    1 root     system    524288000 Nov 02 18:30 file_client2
```

```
/Client2/lv02/lost+found:
total 0
```

4.2 Hitachi ShadowImage

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Logical Volume (LV) using Hitachi ShadowImage.

This example uses a LV virtual SCSI device presented from the Virtual I/O Server to the clients client1 and client2.

The major steps include:

- Determine the client Physical Volume (PV) of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the Volume Group (VG) and LV reside
- Determine the LV on the Virtual I/O Server that is presented to the client where a VG and LV reside

Client1

Client1 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using Hitachi ShadowImage and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk43         00c7086c72bafc21                    rootvg          active
hdisk0          00c7086c8321623a                    vg01            active
hdisk2          00c7086c97f77df2                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "372200c7086c00004c0000000124976f6ffd.105VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk2 is the VG ID of the volume group from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 3722**00c7086c00004c0000000124976f6ffd.1**05VDASD03AIXvscsi

3. Using the `lsvg` command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the `ls` command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 28 14:25 Client1
drwxr-xr-x    2 root     system          256 Oct 27 16:52 lost+found

/lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 14:25 This is Client1

/lv02/lost+found:
total 0
```

Client2

Client2 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services. Make a block level copy of vg02 using Hitachi ShadowImage and mount it on client3 by performing these steps:

1. Using the `lspv` command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk0          00c7086c77cce272                    rootvg          active
hdisk1          00c7086c831f9c2a                    vg01            active
hdisk3          00c7086c97f7f9f7                    vg02            active
```

2. Using the installed ODM package and the `odmget` command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk3 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk3"
        attribute = "unique_id"
        value = "372200c7086c00004c0000000124976f6ffd.205VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk3 is the VG ID of the volume group from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 3722**00c7086c00004c0000000124976f6ffd.2**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 28 15:31 Client2
drwxr-xr-x    2 root     system          256 Oct 27 11:03 lost+found

/lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 15:31 This is Client2

/lv02/lost+found:
total 0
```

Virtual I/O Server

Now that you know the unique IDs of hdisk2 on client1 (vhost0) and hdisk3 on client2 (vhost1), map them to the virtual devices they represent on the Virtual I/O Server by performing these steps:

1. Using the **lsmap** command, determine whether the mapped virtual devices are LVs:

```
$ lsmap -all -type lv
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U9117.MMA.107086C-V1-C12                     0x00000002

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        lv01
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U9117.MMA.107086C-V1-C16                     0x0000000b

VTD                   vtscsi5
Status                Available
LUN                   0x8200000000000000
Backing device        lv02
Physloc
```

2. You now know the names of the mapped LVs and can use the **lspv** and **lsvg** commands to determine what VG they are a member of and the physical devices in which they are resident:

```
$ lsvg
rootvg
vg01

$ lspv | grep vg01
hdisk761        00c7086c8321623a                    vg01            active

$ lsvg -lv vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs        512     512     1    open/syncd    N/A
lv02                jfs        512     512     1    open/syncd    N/A

$ lsvg vg01
VOLUME GROUP:       vg01                     VG IDENTIFIER:
00c7086c00004c0000000124976f6ffd
VG STATE:           active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1275 (20400
megabytes)
MAX LVs:            1024                     FREE PPs:       251 (4016
megabytes)
LVs:                2                        USED PPs:       1024 (16384
megabytes)
OPEN LVs:           2                        QUORUM:         2 (Enabled)
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     1048576                  MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

3. Using the **chkdev** command, determine that hdisk761 represents unique ID 240C50 0B0FA020206OPEN-V07HITACHIfcp within the Virtual I/O Server. Physical disk hdisk761 will be the ShadowImage source or P-VOL. Run the following command:

```
$ chkdev | grep -p hdisk761
NAME:                hdisk761
IDENTIFIER:          240C50 0B0FA020206OPEN-V07HITACHIfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client with dedicated Fibre Channel (FC) SCSI adapters that allow a direct connection to SAN fabrics for the allocation of a Hitachi command device.

The server has Hitachi storage system Command Control Interface (CCI) installed.

> **Note:** Hitachi ShadowImage software uses the in-band Fibre Channel (FC) protocol to send commands to the storage system in order to create and manage Hitachi Online RAID Configuration Manager (HORCM) associations.

Perform these steps:

4. Using the **odmget CuAt** command, identify the PV of the unique ID 240C50 0B0FA020206OPEN-V07HITACHIfcp, which is the ShadowImage source or P-VOL.

   This command is particularly useful because you can also look and make sure that there is a Hitachi ShadowImage command device allocated by looking for a unique ID that contains OPEN-V-CM07.

   Physical disk hdisk144 is the ShadowImage command device.

   Physical disk hdisk142 is the ShadowImage source or P-VOL.

   hdisk467 (unique ID = 240C50 0B0FA031006OPEN-V07HITACHIfcp) is allocated to the ShadowImage management server, which is the ShadowImage target or S-VOL.

   Run the following command;

```
# odmget CuAt | grep -p HITACHI
CuAt:
        name = "hdisk140"
        attribute = "unique_id"
        value = "240C50 0B0FA020006OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk141"
        attribute = "unique_id"
        value = "240C50 0B0FA020106OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk142"
        attribute = "unique_id"
        value = "240C50 0B0FA020206OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk143"
        attribute = "unique_id"
        value = "240C50 0B0FA020306OPEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk144"
        attribute = "unique_id"
        value = "270C50 0B0FA01A309OPEN-V-CM07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk145"
        attribute = "unique_id"
        value = "240C50 0B0FA0300060PEN-V07HITACHIfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk467"
        attribute = "unique_id"
        value = "240C50 0B0FA0310060PEN-V07HITACHIfcp"
        type = "R"
        generic = ""
        rep = "nl"
        nls_index = 79
```

> **Note:** In this case, there is only one command device for ShadowImage operations. For more complicated environments, you may need to assign multiple command devices. Do not forget that ShadowImage and TrueCopy should not share the same command device.

5. Using the ShadowImage Command Control Interface (CCI), you can create the source and target Hitachi Online RAID Configuration Manager (HORCM) configuration files by using the **mkconf** command.

   A single server configuration will be implemented in this example, which establishes a relationship between the device of HORCM instance 1 and the device of HORCM instance 4 on the same server.

   HORCM instance 1 is the P-VOL.

   HORCM instance 4 is the S-VOL.

   Run the following command:

```
# echo hdisk142 | /HORCM/usr/bin/mkconf.sh -g Client0 -i 1 -s 54001
starting HORCM inst 1
HORCM inst 1 starts successfully.
HORCM Shutdown inst 1 !!!
A CONFIG file was successfully completed.
starting HORCM inst 1
HORCM inst 1 starts successfully.
DEVICE_FILE             Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk142                Client0  Client0_000 CL7-F     0    2 -    45306   514
HORCM Shutdown inst 1 !!!
Please check '/etc/horcm1.conf','/etc/log0/curlog/horcm_*.log', and modify
'ip_address & service'.

# echo hdisk467 | /HORCM/usr/bin/mkconf.sh -g Client0 -i 4 -s 54004
starting HORCM inst 4
HORCM inst 4 starts successfully.
HORCM Shutdown inst 4 !!!
A CONFIG file was successfully completed.
starting HORCM inst 4
HORCM inst 4 starts successfully.
DEVICE_FILE             Group    PairVol     PORT   TARG  LUN M   SERIAL  LDEV
hdisk467                Client0  Client0_000 CL7-F     0    6 -    45306   784
HORCM Shutdown inst 4 !!!
Please check '/etc/horcm4.conf','/etc/log4/curlog/horcm_*.log', and modify
'ip_address & service'.
```

6. Edit the HORCM configuration files and change the IP address of the `HORCM_MON` and `HORCM_INST` stanzas to match the localhost. Change the service of the `HORCM_INST` stanzas to point at each other (that is, `horcm1 = 54001` and `horcm4 = 54004`).

   Run the following commands:

   ```
   # vi horcm0.conf

   # vi horcm4.conf

   # vi /etc/services
   ```

   Note: The following is an example S-VOL HORCM configuration file that was used for this step of the procedure:

   ```
   # Created by mkconf.sh on Mon Oct 26 10:31:35 CDT 2009
   HORCM_MON
   #ip_address        service         poll(10ms)     timeout(10ms)
   9.3.92.181         54004                  1000             3000
   HORCM_CMD
   #dev_name               dev_name                dev_name
   #UnitID 0 (Serial# 45306)
   /dev/rhdisk144
   HORCM_DEV
   #dev_group         dev_name        port#     TargetID     LU#     MU#
   # hdisk467             SER =    45306  LDEV =  515 [ FIBRE FCTBL = 3 ]
   Client1            Client1_000     CL7-F            0       3
   HORCM_INST
   #dev_group        ip_address      service
   Client1           9.3.92.181      54001
   ```

7. Set up your environment variables and start the two HORCM instances by running the following command:

   ```
   # export HORCMINST=1

   # export HORCC_MRCF=1

   # horcmstart.sh
   starting HORCM inst 1
   HORCM inst 1 starts successfully.

   # export HORCMINST=4

   # horcmstart.sh
   starting HORCM inst 4
   ```

```
HORCM inst 4 starts successfully.
```

Note: The environment variable set here, `HORCC_MRCF=1`, needs to be set in order to specify ShadowImage functions. If it is not set, TrueCopy functions will be performed instead.

8. Once the HORCM instances start successfully, you can display the configuration by using the **`pairdisplay`** command:

```
# pairdisplay -fxcd -CLI -g Client0
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client0 Client0_000 L   hdisk467       0  45306   310 SMPL     -     - - -
Client0 Client0_000 R   hdisk142       0  45306   202 SMPL     -     - - -
```

You are now ready to create a ShadowImage relationship between the P-VOL and S-VOL. This is a very destructive command, so follow this simple procedure to avoid needing to restore from tape.

When creating a relationship, always use the HORCM instance of the S-VOL, which means that when you specify the relationship copy direction you will always use the -vr switch (that is, -vr for remote or -vl for local).

In this example, a concurrent track copy count of 15 was used.

Perform these steps:

1. Using the **`echo`** command, verify the HORCM instance you currently have configured:

```
# echo $HORCMINST
4

# paircreate -g Client0 -c 15 -vr

# pairdisplay -fxcd -CLI -g Client0
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client0 Client0_000 L   hdisk467       0  45306   310 S-VOL COPY     5 202 -
Client0 Client0_000 R   hdisk142       0  45306   202 P-VOL COPY     5 310 -

# pairdisplay -fxcd -CLI -g Client0
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client0 Client0_000 L   hdisk467       0  45306   310 S-VOL PAIR   100 202 -
Client0 Client0_000 R   hdisk142       0  45306   202 P-VOL PAIR   100 310 –
```

2. Once the copy is complete, indicated by a status of PAIR, use the **`pairsplit`** command to remove the relationship:

```
# pairsplit -S -g Client0

# pairdisplay -fxcd -CLI -g Client0
Group   PairVol L/R  Device_File    M   Seq# LDEV# P/S Status    %  P-LDEV# M
Client0 Client0_000 L   hdisk467       0  45306   310 SMPL     -     - - -
Client0 Client0_000 R   hdisk142       0  45306   202 SMPL     -     - - -
```

Virtual I/O Server

When the copy operation successfully completes, you can present the new PV back to the Virtual I/O Server.

The following steps can be performed to import the VG onto the Virtual I/O Server and present the VG as a virtual device to client3:

1. When logged into the Virtual I/O Server as the padmin user, you operate in a restricted shell. In order to run a **cfgmgr command** to scan for the new device, you need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   # cfgmgr

   # exit
   ```

2. Using the **chkdev** command, scan for the unique ID of the S-VOL, 240C50 0B0FA031006OPEN-V07HITACHIfcp, so you can determine the new physical device name and import the VG:

   ```
   $ chkdev | grep -p "240C50 0B0FA031006OPEN-V07HITACHIfcp"
   NAME:                hdisk761
   IDENTIFIER:          240C50 0B0FA031006OPEN-V07HITACHIfcp
   PHYS2VIRT_CAPABLE:   YES
   VIRT2NPIV_CAPABLE:   NA
   VIRT2PHYS_CAPABLE:   NA

   $ lspv | grep hdisk761
   hdisk761         00c7086c8321623a                     vg01             active
   ```

3. In the output of the **lspv** command, you determine that LVM has recognized the PV_ID. You need to tread carefully in the next few steps in order to clear the PV_ID by using the **chdev** command.

   When logged into the Virtual I/O Server as the padmin user, you operate in a restricted shell. In order to perform a **chdev** command with the appropriate switches to clear the PV_ID, you need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # chdev -l hdisk761 -a pv=clear
   hdisk761 changed

   # lspv | grep hdisk761
   hdisk761        none                                None
   ```

   You have now cleared the PV_ID.

4. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on the Virtual I/O Server.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command;

   ```
   # recreatevg -y vg02 hdisk761
   vg02
   ```

5. You can now return to the Virtual I/O Server restricted shell and continue with the final steps. Run the following command:

```
# exit

$ lsvg
rootvg
vg01
vg02

$ lsvg -lv vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv01              jfs        512     512     1    closed/syncd  N/A
fslv02              jfs        512     512     1    closed/syncd  N/A
```

6. View the current virtual device configuration of client3 (vhost2) by using the **lsmap** command:

```
$ lsmap -vadapter vhost2
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U9117.MMA.107086C-V1-C22                     0x0000003f

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        hdiskpower68
Physloc               U789D.001.DQDMLMP-P1-C1-T2-L70

VTD                   vtscsi3
Status                Available
LUN                   0x8200000000000000
Backing device        hdisk762
Physloc
U789D.001.DQDMLMP-P1-C1-T2-W50060E8005B0FA65-L3000000000000
```

7. Using the **mkvdev** command, map the new LVs on vg02 to client3 (vhost2):

```
$ mkvdev -vdev fslv01 -vadapter vhost2
vtscsi6 Available

$ mkvdev -vdev fslv02 -vadapter vhost2
vtscsi7 Available
```

Client3

At this point, you have successfully created a copy of the VG, vg02, of client1 and the VG,vg02, of client2 using Hitachi ShadowImage. You have also successfully mapped the virtual devices to the VG on the Virtual I/O Server. Now it is time to scan for the new device and import the VG.

Perform the following steps:

1. Using the **cfgmgr** command scan for the new device.

```
# cfgmgr

# lspv
hdisk0          00c7086c7e15d10f                    rootvg          active
hdisk1          00c7086c82e6c683                    vg01            active
hdisk4          00c7086c97f77df2                    None
hdisk5          00c7086c97f7f9f7                    None

# lsvg
rootvg
vg01
```

2. Using the **recreatevg** command, import the VGs, making sure you rename them with new names that do not already exist on client3.

   it is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the Virtual I/O Server, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk4
vg02

# recreatevg -y vg03 -Y CL2 hdisk5
vg03

# lsvg
rootvg
vg01
vg02
vg03

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL1lv02             jfs2       66      66      1    closed/syncd  N/A
CL1loglv02          jfs2log    1       1       1    closed/syncd  N/A

# lsvg -l vg03
vg03:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL2lv02             jfs2       66      66      1    closed/syncd  N/A
CL2loglv02          jfs2log    1       1       1    closed/syncd  N/A
```

3. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new LVs by running the **`vi`** command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv02:
    dev     = /dev/CL1lv02
    vfs     = jfs2
    log     = /dev/CL1loglv02
    mount       = true
    check       = false

/CL2lv02:
    dev     = /dev/CL2lv02
    vfs     = jfs2
    log     = /dev/CL2loglv02
    mount       = true
    check       = false

# mkdir /CL1lv02

# mkdir /CL2lv02

# mount /CL1lv02
Replaying log for /dev/CL1lv02.

# mount /CL2lv02
Replaying log for /dev/CL2lv02.
```

4. Your last step is to look at your replicated data by using the **`ls`** command:

```
# ls -ltR /CL1lv02
/CL1lv02:
total 0
drwx------    2 root     system          256 Oct 28 14:25 Client1
drwxr-xr-x    2 root     system          256 Oct 27 16:52 lost+found

/CL1lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 14:25 This is Client1

/CL1lv02/lost+found:
total 0

# ls -ltR /CL2lv02
/CL2lv02:
total 0
drwx------    2 root     system          256 Oct 28 15:31 Client2
drwxr-xr-x    2 root     system          256 Oct 27 11:03 lost+found

/CL2lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 15:31 This is Client2

/CL2lv02/lost+found:
total 0
```

4.3 IBM FlashCopy: IBM System Storage DS8300

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Logical Volume (LV) using IBM FlashCopy.

This example uses a LV virtual SCSI device presented from the Virtual I/O Server to the clients client1 and client2.

The major steps include:

- Determine the client Physical Volume (PV) of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the Volume Group (VG) and LV reside
- Determine the LV on the Virtual I/O Server which is presented to the client where a VG and LV reside

Client1

Client1 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using IBM FlashCopy and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe4111db6f92c                    vg01            active
hdisk2          000fe41132064400                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "3722000fe4110000d90000000124ba90f3e0.105VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

Note: The unique ID value of hdisk2 is the VG ID of the VG from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:

3722**000fe4110000d90000000124ba90f3e0.1**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 28 16:12 Client1
drwxr-xr-x    2 root     system          256 Oct 27 18:09 lost+found

/lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 16:12 This is Client1

/lv02/lost+found:
total 0
```

Client2

Client2 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using IBM FlashCopy and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411aeb7c10a                    vg01            active
hdisk2          000fe411b271042a                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "3722000fe4110000d90000000124ba90f3e0.205VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk2 is the VG ID of the VG from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 3722**000fe4110000d90000000124ba90f3e0.2**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 28 16:31 Client2
drwxr-xr-x    2 root     system          256 Oct 27 18:23 lost+found

/lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 16:31 This is Client2

/lv02/lost+found:
total 0
```

Virtual I/O Server

Now that you know the unique ID of hdisk2 on client1 (vhost2) and the unique ID of hdisk2 on client2 (vhost0), map to the virtual devices they represent on the Virtual I/O Server by performing these steps:

1. Using the **lsmap** command, determine that the mapped virtual devices are LVs:

```
$ lsmap -all -type lv
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                     0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi7
Status                Available
LUN                   0x8300000000000000
Backing device        lv02
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U8204.E8A.10FE411-V2-C12                     0x00000004

VTD                   vtscsi1
Status                Available
LUN                   0x8100000000000000
Backing device        p2rootvg
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
```

```
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi3
Status                Available
LUN                   0x8300000000000000
Backing device        lv01
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost3          U8204.E8A.10FE411-V2-C14                     0x00000006

VTD                   vp4rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p4rootvg
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost7          U8204.E8A.10FE411-V2-C19                     0x00000007

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        p5rootvg
Physloc
```

2. You now know the names of the mapped LVs and can use the **lspv** and **lsvg** commands to determine what VG they are a member of and the physical devices in which they are resident:

```
$ lsvg
rootvg
vg01

$ lspv | grep vg01
hdisk11          000fe4111db6f92c                     vg01             active

$ lsvg -lv vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs        512     512     1    open/syncd    N/A
lv02                jfs        512     512     1    open/syncd    N/A

$ lsvg vg01
VOLUME GROUP:       vg01                     VG IDENTIFIER:
000fe4110000d90000000124ba90f3e0
VG STATE:           active                   PP SIZE:        128 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      159 (20352
megabytes)
```

```
MAX LVs:            512                      FREE PPs:       19 (2432
megabytes)
LVs:                2                        USED PPs:       140 (17920
megabytes)
OPEN LVs:           2                        QUORUM:         2 (Enabled)
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     130048
MAX PPs per PV:     1016                     MAX PVs:        128
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

3. Using the **chkdev** command, determine that hdisk11 represents unique ID 200B75BALB1110807210790003IBMfcp within the Virtual I/O Server.

   Physical disk hdisk11 will be the FlashCopy source volume.

   Run the following command:

```
$ chkdev | grep -p hdisk11
NAME:                 hdisk11
IDENTIFIER:           200B75BALB1110807210790003IBMfcp
PHYS2VIRT_CAPABLE:    YES
VIRT2NPIV_CAPABLE:    NA
VIRT2PHYS_CAPABLE:    NA
```

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client. The server has IBM DSCLI installed, which is used to manage storage subsystem functions.

Perform the following steps:

1. Using the **odmget CuAt** command, identify the PV of the unique ID 200B75BALB1110807210790003IBMfcp, which is the FlashCopy source volume.

   Physical disk hdisk11 is the FlashCopy source volume.

   hdisk10 (unique ID = 200B75BALB1110707210790003IBMfcp) is allocated to the storage subsystem management server, which is the FlashCopy target volume.

   Run the following command;

```
# odmget CuAt | grep -p IBMfcp
CuAt:
        name = "hdisk8"
        attribute = "unique_id"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk9"
        attribute = "unique_id"
        value = "200B75BALB1100E07210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk10"
        attribute = "unique_id"
        value = "200B75BALB1110707210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "hdisk11"
        attribute = "unique_id"
        value = "200B75BALB1110807210790003IBMfcp"
        type = "R"
        generic = "D"
        rep = "nl"
        nls_index = 79

CuAt:
        name = "vtscsi2"
        attribute = "udid_info"
        value = "200B75BALB1100D07210790003IBMfcp"
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0

CuAt:
        name = "vtscsi5"
        attribute = "udid_info"
        value = "200B75BALB1100E07210790003IBMfcp"
        type = "R"
        generic = ""
        rep = "s"
        nls_index = 0
```

2. Using the DSCLI **lshostvol** command, determine the IBM DS8300 vpath ID of hdisk10 and hdisk11:

```
dscli> lshostvol
Date/Time: November 3, 2009 9:41:51 AM CST IBM DSCLI Version: 5.4.1.44 DS: -
Disk Name Volume Id             Vpath Name
==========================================
hdisk8    IBM.2107-75BALB1/100D ---
hdisk9    IBM.2107-75BALB1/100E ---
hdisk10   IBM.2107-75BALB1/1107 ---
hdisk11   IBM.2107-75BALB1/1108 ---
```

3. You are now ready to create the FlashCopy relationship between the source (vpath ID = 1108) and target (vpath ID = 1107) volumes using your FlashCopy management server and the **mkflash** command:

```
dscli> mkflash -dev IBM.2107-75BALB1 -persist -cp -seqnum 0001 1108:1107
Date/Time: November 3, 2009 9:42:16 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00137I mkflash: FlashCopy pair 1108:1107 successfully created.
```

4. Display the configuration by using the **lsflash** command:

```
dscli> lsflash -dev IBM.2107-75BALB1 -l 1108:1107
Date/Time: November 3, 2009 9:42:47 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                   State AllowTgtSE
===============================================================================
===============================================================================
==========================================================
1108:1107 11     1           120     Enabled    Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        291369          Tue Nov 03
09:46:10 CST 2009 Tue Nov 03 09:46:10 CST 2009 Valid Disabled

dscli> lsflash -dev IBM.2107-75BALB1 -l 1108:1107
Date/Time: November 3, 2009 9:47:04 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
ID        SrcLSS SequenceNum Timeout ActiveCopy Recording Persistent Revertible
SourceWriteEnabled TargetWriteEnabled BackgroundCopy OutOfSyncTracks
DateCreated                  DateSynced                   State AllowTgtSE
===============================================================================
===============================================================================
==========================================================
1108:1107 11     1           120     Disabled   Disabled  Enabled    Disabled
Enabled            Enabled            Enabled        0               Tue Nov 03
09:46:10 CST 2009 Tue Nov 03 09:46:10 CST 2009 Valid Disabled
```

5. Once the copy operation is complete, indicated by the 0 in OutOfSyncTracks, use the **rmflash** command to remove the relationship:

```
dscli> rmflash -dev IBM.2107-75BALB1 -seqnum 0001 1108:1107
Date/Time: November 3, 2009 9:49:03 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00144W rmflash: Are you sure you want to remove the FlashCopy pair
1108:1107:? [y/n]:y
CMUC00140I rmflash: FlashCopy pair 1108:1107 successfully removed.

dscli> lsflash -dev IBM.2107-75BALB1 -l 1108:1107
Date/Time: November 3, 2009 9:49:18 AM CST IBM DSCLI Version: 5.4.1.44 DS:
IBM.2107-75BALB1
CMUC00234I lsflash: No Flash Copy found.
```

Virtual I/O Server

When the copy operation successfully completes, you can present the new PV to the Virtual I/O Server.

The following steps can be performed to import the VG onto the Virtual I/O Server and present the VG as a virtual device to client3:

1. When logged into the Virtual I/O Server as the padmin user, you are operating in a restricted shell. In order to perform a **cfgmgr** command to scan for the new device, you need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # cfgmgr

   # exit
   ```

2. Using the **chkdev** command, scan for the unique ID of the target volume, 200B75BALB1110707210790003IBMfcp, so you can determine the new physical device name and import the VG:

   ```
   $ chkdev | grep -p 200B75BALB1110707210790003IBMfcp
   NAME:                hdisk10
   IDENTIFIER:          200B75BALB1110707210790003IBMfcp
   PHYS2VIRT_CAPABLE:   YES
   VIRT2NPIV_CAPABLE:   NA
   VIRT2PHYS_CAPABLE:   NA

   $ lspv | grep hdisk10
   hdisk10         000fe4111db6f92c                    vg01            active
   ```

3. In the output of the **lspv** command, you can determine that LVM has recognized the PV_ID. You need to tread carefully when performing the next step to clear the PV_ID using the **chdev** command.

   When logged into the Virtual I/O Server as the padmin user, you are operating in a restricted shell. In order to perform a **chdev** command with the appropriate switches to clear the PV_ID, you will need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # chdev -l hdisk10 -a pv=clear
   hdisk10 changed

   # lspv | grep hdisk10
   hdisk10         none                                None
   ```

   You have now cleared the PV_ID.

4. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on the Virtual I/O Server.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that you have a limit of 15 characters.

Run the following command:

```
# recreatevg -y vg02 hdisk10
vg02
```

5. You can now return to the Virtual I/O Server restricted shell and continue with the final steps:

```
# exit

$ lsvg
rootvg
vg01
vg02

$ lsvg -lv vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
fslv01              jfs        512     512     1    open/syncd    N/A
fslv02              jfs        512     512     1    open/syncd    N/A
```

6. View the current virtual device configuration of client3 (vhost7) by using the **lsmap** command:

```
$ lsmap -vadapter vhost7
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost7          U8204.E8A.10FE411-V2-C19                     0x00000007

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        p5rootvg
Physloc

VTD                   vtscsi6
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk12
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W500507630400812C-L4011400700000000
```

7. Using the **mkvdev** command, map the new LVs on vg02 to client3 (vhost2):

```
$ mkvdev -vdev fslv01 -vadapter vhost7
vtscsi6 Available

$ mkvdev -vdev fslv02 -vadapter vhost7
vtscsi8 Available
```

Client3

At this point, you have successfully created a copy of the VG vg02 of client1 and the VG vg02 of client2 using IBM FlashCopy. You have also successfully mapped the virtual devices to the VG on the Virtual I/O Server. Now it is time to scan for the new device and import the VG by performing the following steps:

1. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk0          000fe411b66af634                    rootvg          active
hdisk1          000fe411b6bb57f9                    vg01            active
hdisk2          000fe41132064400                    None
hdisk3          000fe411b271042a                    None

# lsvg
rootvg
vg01
```

2. Using the **recreatevg** command, import the VGs, making sure you rename them with new names that do not already exist on client3.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the Virtual I/O Server, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk2
vg02

# recreatevg -y vg03 -Y CL2 hdisk3
vg03

# lsvg
rootvg
vg01
vg02
vg03

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL1lv02             jfs2       66      66      1    closed/syncd  N/A
CL1loglv02          jfs2log    1       1       1    closed/syncd  N/A

# lsvg -l vg03
vg03:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL2lv02             jfs2       66      66      1    closed/syncd  N/A
CL2loglv02          jfs2log    1       1       1    closed/syncd  N/A
```

3. Now you can edit /etc/filesystems, create the mountpoints, and mount the new LVs by using the **vi** command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv02:
    dev     = /dev/CL1lv02
    vfs     = jfs2
    log     = /dev/CL1loglv02
    mount       = true
    check       = false

/CL2lv02:
    dev     = /dev/CL2lv02
    vfs     = jfs2
    log     = /dev/CL2loglv02
    mount       = true
    check       = false

# mkdir /CL1lv02

# mkdir /Cl2lv02

# mount /CL1lv02
Replaying log for /dev/CL1lv02.

# mount /CL2lv02
Replaying log for /dev/CL2lv02.
```

4. Your last step is to look at your replicated data by using the **ls** command:

```
# ls -ltR /CL1lv02
/CL1lv02:
total 0
drwx------    2 root     system          256 Oct 28 16:12 Client1
drwxr-xr-x    2 root     system          256 Oct 27 18:09 lost+found

/CL1lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 16:12 This is Client1

/CL1lv02/lost+found:
total 0

# ls -ltR /CL2lv02
/CL2lv02:
total 0
drwx------    2 root     system          256 Oct 28 16:31 Client2
drwxr-xr-x    2 root     system          256 Oct 27 18:23 lost+found

/CL2lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 28 16:31 This is Client2

/CL2lv02/lost+found:
total 0
```

4.4 IBM FlashCopy: IBM SAN Volume Controller (SVC)

In this section, we discuss the steps required to complete a block level copy of a Virtual I/O Server client Logical Volume (LV) using IBM FlashCopy.

This example uses a LV virtual SCSI device presented from the Virtual I/O Server to the clients client1 and client2.

The major steps include:

- Determine the client Physical Volume (PV) of which a copy will be made
- Determine the relationship of the client PV to the storage subsystem management server
- Determine the PV on the Virtual I/O Server where the VG and LV reside
- Determine the LV on the Virtual I/O Server which is presented to the client where the VG and LV reside

Client1

Client1 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using IBM FlashCopy and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk0          000fe411206febdd                    rootvg          active
hdisk1          000fe41141804594                    vg01            active
hdisk2          000fe4114283c48a                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "3722000fe4110000d90000000124cb0e6a0b.105VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk2 is the VG ID of the VG from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 3722**000fe4110000d90000000124cb0e6a0b.1**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 30 08:52 Client1
drwxr-xr-x    2 root     system          256 Oct 29 12:13 lost+found

/lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 30 08:52 This is Client1

/lv02/lost+found:
total 0
```

Client2

Client2 has a tertiary VG, vg02, which for the purposes of this demonstration contains the data to be replicated using Copy Services.

The following steps can be performed to make a block level copy of vg02 using IBM FlashCopy and mount it on client3:

1. Using the **lspv** command, determine that vg02 resides on hdisk2:

```
# lspv
hdisk0          000fe411201305c3                    rootvg          active
hdisk1          000fe411c1eb093d                    vg01            active
hdisk2          000fe411c2edb457                    vg02            active
```

2. Using the installed ODM package and the **odmget** command, determine the unique ID of the PV. The unique ID in this configuration represents the VG ID on the Virtual I/O Server. Run the following command:

```
# odmget -q "name=hdisk2 and attribute=unique_id" CuAt

CuAt:
        name = "hdisk2"
        attribute = "unique_id"
        value = "3722000fe4110000d90000000124cb0e6a0b.205VDASD03AIXvscsi"
        type = "R"
        generic = ""
        rep = "n"
        nls_index = 0
```

> **Note:** The unique ID value of hdisk2 is the VG ID of the VG from which it is created on the Virtual I/O Server, indicated by the bolded text within the following output:
>
> 3722**000fe4110000d90000000124cb0e6a0b.2**05VDASD03AIXvscsi

3. Using the **lsvg** command, determine the file system for which the copy will be made:

```
# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv02                jfs2       66      66      1    closed/syncd  /lv02
loglv02             jfs2log    1       1       1    closed/syncd  N/A
```

4. View the data that will be copied by using the **ls** command:

```
# ls -ltR /lv02
/lv02:
total 0
drwx------    2 root     system          256 Oct 30 09:49 Client2
drwxr-xr-x    2 root     system          256 Oct 29 13:26 lost+found

/lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 30 09:49 This is Client2

/lv02/lost+found:
total 0
```

Virtual I/O Server

Now that you know the unique IDs of hdisk2 on client1 (vhost2) and hdisk2 on client2 (vhost0), map to the virtual devices they represent on the Virtual I/O Server by performing these steps:

1. Using the **lsmap** command, determine that the mapped virtual devices are LVs:

```
$ lsmap -all -type lv
SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost0          U8204.E8A.10FE411-V2-C11                     0x00000003

VTD                   vtscsi0
Status                Available
LUN                   0x8100000000000000
Backing device        p1rootvg
Physloc

VTD                   vtscsi7
Status                Available
LUN                   0x8300000000000000
Backing device        lv02
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost1          U8204.E8A.10FE411-V2-C12                     0x00000000

VTD                   vtscsi1
Status                Available
LUN                   0x8100000000000000
Backing device        p2rootvg
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost2          U8204.E8A.10FE411-V2-C13                     0x00000005

VTD                   vp3rootvg
```

```
Status                Available
LUN                   0x8100000000000000
Backing device        p3rootvg
Physloc

VTD                   vtscsi6
Status                Available
LUN                   0x8300000000000000
Backing device        lv01
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost3          U8204.E8A.10FE411-V2-C14                     0x00000006

VTD                   vp4rootvg
Status                Available
LUN                   0x8100000000000000
Backing device        p4rootvg
Physloc

SVSA            Physloc                                      Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost7          U8204.E8A.10FE411-V2-C19                     0x00000007

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        p5rootvg
Physloc
```

2. You now know the names of the mapped LVs and can use the **lspv** and **lsvg** commands to determine what VG they are a member of and the physical devices in which they are resident:

```
$ lsvg
rootvg
vg01

$ lspv | grep vg01
hdisk15          000fe411cb0e699b                     vg01            active

$ lsvg -lv vg01
vg01:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
lv01                jfs        512     512     1    open/syncd    N/A
lv02                jfs        512     512     1    open/syncd    N/A

$ lsvg vg01
VOLUME GROUP:       vg01                     VG IDENTIFIER:
000fe4110000d90000000124cb0e6a0b
VG STATE:           active                   PP SIZE:        16 megabyte(s)
VG PERMISSION:      read/write               TOTAL PPs:      1275 (20400
megabytes)
```

```
MAX LVs:            1024                     FREE PPs:       251 (4016
megabytes)
LVs:                2                        USED PPs:       1024 (16384
megabytes)
OPEN LVs:           2                        QUORUM:         2 (Enabled)
TOTAL PVs:          1                        VG DESCRIPTORS: 2
STALE PVs:          0                        STALE PPs:      0
ACTIVE PVs:         1                        AUTO ON:        yes
MAX PPs per VG:     1048576                  MAX PVs:        1024
LTG size (Dynamic): 256 kilobyte(s)          AUTO SYNC:      no
HOT SPARE:          no                       BB POLICY:      relocatable
```

3. Using the **chkdev** command, determine that hdisk15 represents unique ID 332136005076801830OFD5000000000000000304214503IBMfcp within the Virtual I/O Server:

```
Physical disk hdisk15 will be the FlashCopy source volume.

$ chkdev | grep -p hdisk15
NAME:                hdisk15
IDENTIFIER:          332136005076801830OFD5000000000000000304214503IBMfcp
PHYS2VIRT_CAPABLE:   YES
VIRT2NPIV_CAPABLE:   NA
VIRT2PHYS_CAPABLE:   NA
```

Storage subsystem management server

The storage subsystem management server used in this demonstration is a Virtual I/O Server client. The server has an SSH client installed that is used to manage IBM SAN Volume Controller (SVC) functions.

Perform these steps:

1. Using the **svcinfo lshostvdiskmap** command, look for the component of the unique ID that contains the SAN VOlume Controller (SVC) vdisk_UID 60050768018300FD5000000000000000, which will be the FlashCopy source volume.

   In this example, host0 is the Virtual I/O Server where all three clients are allocated.

   Physical disk host0_4 (unique ID = 60050768018300FD5000000000000003) is the FlashCopy source volume.

   Physical disk host0_5 (unique ID = 60050768018300FD5000000000000004) is the FlashCopy target volume.

   Run the following command:

```
IBM_2145:admin>svcinfo lshostvdiskmap
id              name             SCSI_id       vdisk_id       vdisk_name
wwpn             vdisk_UID
0               host0             0             0              host0_1
10000000C9738E84 60050768018300FD5000000000000000
0               host0             1             1              host0_2
10000000C9738E84 60050768018300FD5000000000000001
0               host0             2             2              host0_3
10000000C9738E84 60050768018300FD5000000000000002
0               host0             3             3              host0_4
10000000C9738E84 60050768018300FD5000000000000003
0               host0             4             4              host0_5
10000000C9738E84 60050768018300FD5000000000000004
```

2. You are now ready to create the FlashCopy relationship between the source and target volumes using your FlashCopy management server and the **mkfcconsistgrp** and **mkfcmap** commands:

```
IBM_2145:admin>svctask mkfcconsistgrp -name host0
FlashCopy Consistency Group, id [1], successfully created

IBM_2145:admin>svctask mkfcmap -source host0_4 -target host0_5 -name host0
-consistgrp host0
FlashCopy Mapping, id [0], successfully created

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 4
source_vdisk_name host0_4
target_vdisk_id 3
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 0
copy_rate 50
start_time 091106201633
dependent_mappings 0
autodelete off
```

3. Once the FlashCopy relationship is created, prepare the consistency group by using the **prestartfcconsistgrp** command:

```
IBM_2145:admin>svctask prestartfcconsistgrp host0

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 4
source_vdisk_name host0_4
target_vdisk_id 3
target_vdisk_name host0_5
group_id 1
group_name host0
status prepared
progress 0
copy_rate 50
start_time 091106201633
dependent_mappings 0
autodelete off
```

4. Now start the FlashCopy by using the **startfcconsistgrp** command:

```
IBM_2145:admin>svctask startfcconsistgrp host0

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 4
source_vdisk_name host0_4
target_vdisk_id 3
```

```
target_vdisk_name host0_5
group_id 1
group_name host0
status copying
progress 0
copy_rate 50
start_time 091106201633
dependent_mappings 0
autodelete off

IBM_2145:admin>svcinfo lsfcmap host0
id 0
name host0
source_vdisk_id 0
source_vdisk_name host0_4
target_vdisk_id 4
target_vdisk_name host0_5
group_id 1
group_name host0
status idle_or_copied
progress 100
copy_rate 50
start_time 091106201633
dependent_mappings 0
autodelete off
```

5. Once the copy operation is complete, indicated by a progress of 100 and a status of idle_or_copied, use the **rmfcmap** command to remove the relationship:

```
IBM_2145:admin>svctask rmfcmap host0

IBM_2145:admin>svcinfo lsfcmap host0
CMMVC5754E The object specified does not exist, or the name supplied does not
meet the naming rules.
```

   The copy operation is now complete and ready to be presented to client3.

Virtual I/O Server

With the successful completion of the copy operation, you are now able to present the new PV back to the Virtual I/O Server.

The following steps can be performed to import the VG onto the Virtual I/O Server and present the VG as a virtual device to client3:

1. When logged into the Virtual I/O Server as the padmin user, you operate in a restricted shell. In order to perform a **cfgmgr** command to scan for the new device, you need to use the **oem_setup_env** command to switch to a privileged shell:

```
$ oem_setup_env

# cfgmgr

# exit
```

2. Using the **chkdev** command, scan for the unique ID of the target volume, 60050768018300FD5000000000000004, so you can determine the name of the new physical device and import the VG:

   ```
   $ chkdev | grep -p 60050768018300FD5000000000000004
   NAME:                hdisk16
   IDENTIFIER:          3321360050768018300FD500000000000000404214503IBMfcp
   PHYS2VIRT_CAPABLE:   YES
   VIRT2NPIV_CAPABLE:   NA
   VIRT2PHYS_CAPABLE:   NA

   $ lspv | grep hdisk16
   hdisk16          000fe411cb4e34bd                    vg01            active
   ```

3. In the output of the **lspv** command, you can determine that LVM has recognized the PV_ID. You need to tread carefully when performing the next step to clear the PV_ID using the **chdev** command.

   When logged into the Virtual I/O Server as the padmin user, you operate in a restricted shell. In order to perform a **chdev** command with the appropriate switches to clear the PV_ID, you need to use the **oem_setup_env** command to switch to a privileged shell:

   ```
   $ oem_setup_env

   # chdev -l hdisk16 -a pv=clear
   hdisk16 changed

   # lspv | grep hdisk16
   hdisk16         none                                None
   ```

   You have now cleared the PV_ID.

4. Using the **recreatevg** command, import the VG, making sure to rename it with a name that does not already exist on the Virtual I/O Server.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the client, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

   Run the following command;

   ```
   # recreatevg -y vg02 hdisk16
   vg02
   ```

5. You can now return to the Virtual I/O Server restricted shell and continue with the final steps by running the following command:

   ```
   # exit

   $ lsvg
   rootvg
   vg01
   vg02

   $ lsvg -lv vg02
   vg02:
   LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
   fslv01              jfs        512     512     1    open/syncd    N/A
   fslv02              jfs        512     512     1    open/syncd    N/A
   ```

6. View the current virtual device configuration of client3 (vhost7) by using the **lsmap** command:

```
$ lsmap -vadapter vhost7
SVSA           Physloc                                       Client Partition
ID
--------------- -------------------------------------------- ------------------
vhost7         U8204.E8A.10FE411-V2-C19                      0x00000007

VTD                   vtscsi4
Status                Available
LUN                   0x8200000000000000
Backing device        p5rootvg
Physloc

VTD                   vtscsi5
Status                Available
LUN                   0x8100000000000000
Backing device        hdisk14
Physloc
U78A0.001.DNWGCV7-P1-C4-T1-W5005076801401FAA-L2000000000000
```

7. Using the **mkvdev** command, map the new LVs on vg02 to client3 (vhost7):

```
$ mkvdev -vdev fslv01 -vadapter vhost7
vtscsi7 Available

$ mkvdev -vdev fslv02 -vadapter vhost7
vtscsi9 Available
```

Client3

At this point, you have successfully created a copy of the VG vg02 of client1 and the VG vg02 of client2 using IBM FlashCopy. You have also successfully mapped the virtual devices to the VG on the Virtual I/O Server. Now it is time to scan for the new device and import the VG.

Perform these steps:

1. Using the **cfgmgr** command, scan for the new device:

```
# cfgmgr

# lspv
hdisk0         000fe411b66af634                    rootvg          active
hdisk1         000fe411ca12edec                    vg01            active
hdisk2         000fe41142d5918e                    None
hdisk3         000fe411c33ec562                    None

# lsvg
rootvg
vg01
```

2. Using the **recreatevg** command, import the VGs, making sure you rename them with new names that do not already exist on client3.

   It is handy to have the LV names of the new VG available when using the **recreatevg** command, because if the LV name already exists on the Virtual I/O Server, it will be renamed automatically.

   When specifying the values for the -Y switch, remember that there is a limit of 15 characters.

Run the following command:

```
# recreatevg -y vg02 -Y CL1 hdisk2
vg02

# recreatevg -y vg03 -Y CL2 hdisk3
vg03

# lsvg
rootvg
vg01
vg02
vg03

# lsvg -l vg02
vg02:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL1lv02             jfs2       66      66      1    closed/syncd  N/A
CL1loglv02          jfs2log    1       1       1    closed/syncd  N/A

# lsvg -l vg03
vg03:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
CL2lv02             jfs2       66      66      1    closed/syncd  N/A
CL2loglv02          jfs2log    1       1       1    closed/syncd  N/A
```

3. Now you can edit `/etc/filesystems`, create the mountpoints, and mount the new LVs by using the **`vi`** command:

```
# vi /etc/filesystems
(Append the following)
/CL1lv02:
    dev     = /dev/CL1lv02
    vfs     = jfs2
    log     = /dev/CL1loglv02
    mount       = true
    check       = false

/CL2lv02:
    dev     = /dev/CL2lv02
    vfs     = jfs2
    log     = /dev/CL2loglv02
    mount       = true
    check       = false

# mkdir /CL1lv02

# mkdir /Cl2lv02

# mount /CL1lv02
Replaying log for /dev/CL1lv02.

# mount /CL2lv02
Replaying log for /dev/CL2lv02.
```

4. Your last step is to look at your replicated data by using the **ls** command:

```
# ls -ltR /CL1lv02
/CL1lv02:
total 0
drwx------    2 root     system          256 Oct 30 08:52 Client1
drwxr-xr-x    2 root     system          256 Oct 29 12:13 lost+found

/CL1lv02/Client1:
total 1024000
-rw-------    1 root     system    536870912 Oct 30 08:52 This is Client1

/CL1lv02/lost+found:
total 0

# ls -ltR /CL2lv02
/CL2lv02:
total 0
drwx------    2 root     system          256 Oct 30 09:49 Client2
drwxr-xr-x    2 root     system          256 Oct 29 13:26 lost+found

/CL2lv02/Client2:
total 1024000
-rw-------    1 root     system    536870912 Oct 30 09:49 This is Client2

/CL2lv02/lost+found:
total 0
```

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks publications

For information about ordering these publications, see “How to get Redbooks” on page 121. Note that some of the documents referenced here may be available in softcopy only.

- *IBM PowerVM Live Partition Mobility*, SG24-7460
- *IBM System p Advanced POWER Virtualization (PowerVM) Best Practices*, REDP-4194
- *IBM System Storage DS8000: Copy Services in Open Environments*, SG24-6788
- *PowerVM Migration from Physical to Virtual Storage*, SG24-7825
- *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940
- *SAN Volume Controller V4.3.0 Advanced Copy Services*, SG24-7574

Online resources

These Web sites are also relevant as further information sources:

- EMC U.S.A. Web site:

  `http://www.emc.com/`

- Hitachi Data Systems U.S.A Web site:

  `http://www.hds.com/`

- IBM Storage U.S.A Web site:

  `http://www-03.ibm.com/systems/storage/`

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

PowerVM and SAN Copy Services

Redpaper™

Real-world data center scenarios with procedures for AIX, Virtual I/O Server, and storage devices

A key reference for SAN-based storage in a virtualized environment

EMC, Hitachi, and IBM storage examples included

This IBM Redpapers publication is a guide to Copy Services as managed by AIX in a IBM PowerVM virtualized environment. The goal of this paper is to provide step-by-step procedures about how storage subsystem-based Copy Services are accomplished in a virtualized environment and the approach for which they should be adopted.

This paper focuses on Copy Services technology from EMC, Hitachi, and IBM within the following storage subsystem models:

- EMC DMX-4
- Hitachi USP V
- IBM DS8300
- IBM SVC

The storage software products that will be used for the different Copy Services are as follows:

- EMC TimeFinder
- Hitachi ShadowImage
- IBM FlashCopy

While there are numerous storage subsystem vendors in the marketplace and industry, this paper has been limited to the vendors and the technologies described above.

REDP-4610-00