IBM

IBM PowerHA SystemMirror for AIX 7.1.3 Best Practices and Migration Guide

- Positions technically IBM PowerHA SystemMirror
- Includes best practices guidelines
- Describes migration scenarios

Dino Quintero
Shawn Bodily
Daniel J. Martin-Corben
Reshma Prathap
Kulwinder Singh
Ashraf Ali Thajudeen
William Nespoli Zanatta

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM PowerHA SystemMirror for AIX 7.1.3 Best Practices and Migration Guide

January 2015

SG24-8234-00

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

First Edition (January 2015)

This edition applies to PowerHA 7.1.3 SP1, AIX 7.1.3 SP1, Symantec Storage Foundation and High Availability 6.1.

© Copyright International Business Machines Corporation 2015. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices vii
Trademarks viii

IBM Redbooks promotions ix

Preface xi
Authors xi
Now you can become a published author, too! xii
Comments welcome xiii
Stay connected to IBM Redbooks xiii

Chapter 1. IBM PowerHA SystemMirror for AIX best practices 1
1.1 Introduction 2
1.2 Designing high availability 2
1.2.1 Risk analysis 3
1.3 Cluster components 4
1.3.1 Nodes 4
1.3.2 Networks 5
1.3.3 Adapters 6
1.3.4 Applications 7
1.4 Testing 10
1.5 Maintenance 10
1.5.1 Upgrading the cluster environment 11
1.6 Monitoring 13
1.7 PowerHA in a virtualized world 15
1.7.1 Maintenance of the VIOS partition: Applying updates 20
1.7.2 Workload partitions 21
1.8 Summary 24

Chapter 2. File system conversion and migration 27
2.1 Conversion versus migration 28
2.2 Volume and file system migration 28
2.2.1 Volume group conversion and migration 28
2.2.2 Limitations of migration 32
2.2.3 Migrating from VxVM to LVM 32

Chapter 3. Symantec Cluster Server powered by Veritas 35
3.1 Executive overview 36
3.2 Components of a Symantec cluster 36
3.3 Cluster resources 37
3.4 Cluster configurations 39
3.5 Cluster communication 39
3.6 Cluster installation and setup 40
3.7 Cluster administration facilities 40
3.8 PowerHA and Symantec Cluster Server compared 41
3.8.1 Components of a PowerHA cluster 41
3.8.2 Cluster resources 42
3.8.3 Cluster configurations 43
3.8.4 Cluster communications 44

© Copyright IBM Corp. 2015. All rights reserved.

3.8.5 Cluster installation and setup 44
3.8.6 Cluster administration facilities 45
3.8.7 PowerHA and Symantec Cluster Server feature comparison summary 45

Chapter 4. Migrating from Symantec Cluster Server powered by Veritas 47
4.1 Terminology 48
4.1.1 Cluster communication 48
4.1.2 Seeding 48
4.1.3 Coordination points 49
4.1.4 Fencing 49
4.1.5 Resources and groups 49
4.1.6 Storage foundation 50
4.2 Introduction 51
4.3 Daily administration 52
4.4 Cluster environment 52
4.5 Planning 52
4.5.1 Network considerations 53
4.5.2 Storage considerations 53
4.5.3 Application resources 53
4.6 Converting a Symantec Cluster Server to an IBM PowerHA cluster 54
4.7 Test environment overview 54
4.7.1 Environmental details 55
4.7.2 Symantec Cluster Server configuration 55
4.7.3 Collecting Cluster specifications 57
4.8 Creating the LVM volume group and the file systems 58
4.8.1 LVM creation 58
4.8.2 Add LVM to VCS 59
4.9 Installing the PowerHA software 61
4.9.1 Testing the cluster 65
4.10 Performing the migration 66
4.11 Roll-back and removal 72
4.12 Deleting the Symantec Cluster Server 72
4.13 Troubleshooting, and known issues 73
4.13.1 Volume Manager holds the disk 73
4.13.2 Volume group will not varyon with VCS after PowerHA test 74
4.14 Shared storage pools 75
4.15 Cluster migration 75
4.15.1 PowerHA and Storage Foundation 75
4.15.2 Configuring User-Defined Resources 76
4.15.3 Enabling Storage Foundation Resources in PowerHA 76
4.15.4 Mixed volume manager environment 80

Chapter 5. Converting a local PowerHA Standard Edition cluster to a PowerHA Enterprise Edition cluster 81
5.1 Test environment overview 82
5.2 Install PowerHA Enterprise Edition software 85
5.3 Delete existing cluster 87
5.4 Creating a three node two-site GLVM linked cluster 88
5.4.1 Define linked cluster with sites 89
5.4.2 Configure GLVM 93
5.4.3 Create GMVG 102
5.4.4 Create resource group 108
5.4.5 Add GMVG into resource group 109
5.5 Defining manual site split and merge policy 110

5.6 Testing manual split option 113
5.6.1 First node failure in site dallas. 114
5.6.2 Site split 115
5.6.3 Restart primary site nodes 117
5.6.4 Move resource group back to primary site 118
5.7 Testing manual merge option 120

Related publications 123
IBM Redbooks 123
Online resources 123
Help from IBM 123

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

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

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.

© Copyright IBM Corp. 2015. All rights reserved.

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:

AIX®
DB2®
developerWorks®
DS8000®
DYNIX/ptx®
Global Technology Services®
GPFS™
HACMP™
HyperSwap®
IBM®
POWER®
Power Systems™
POWER7®
POWER8™
PowerHA®
PowerVM®
Redbooks®
Redpapers™
Redbooks (logo)  ®
Storwize®
System p®
SystemMirror®
Tivoli®
XIV®

The following terms are trademarks of other companies:

Symantec is a registered trademark owned by Symantec Corporation or its affiliates in the U.S. and other countries.

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

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android

Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication positions high availability solutions for IBM Power Systems™ with IBM PowerHA® SystemMirror® Standard and Enterprise Editions (hardware, software, best practices, reference architectures, migration, and tools) with a well-defined and documented deployment model within an IBM Power Systems environment allowing customers a planned foundation for a dynamic high available infrastructure for their enterprise applications.

This Redbooks publication documents topics to leverage the strengths of IBM PowerHA SystemMirror Standard and Enterprise Editions 7.1.3 for IBM Power Systems to solve customers' application high availability challenges, and maximize systems' availability, and management.

This Redbooks publication focuses on providing the readers with technical information and references on the capabilities of each edition, functionalities, usability, and features that make IBM PowerHA SystemMirror a premier solution for high availability and disaster recovery for IBM Power Systems servers.

This Redbooks publication helps strengthen the position of the IBM PowerHA SystemMirror solution with a well-defined and documented best practices, usability, functionality, migration and deployment model within an IBM POWER® system virtualized environment allowing customers a planned foundation for business resilient infrastructure solutions.

This Redbooks publication is targeted toward technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) responsible for providing high availability solutions and support with the IBM PowerHA SystemMirror on IBM POWER.

Authors

This book was produced by a team of specialists from around the world working at the IBM International Technical Support Organization (ITSO), Poughkeepsie Center.

Dino Quintero is a Complex Solutions Project Leader and an IBM Senior Certified IT Specialist with the ITSO in Poughkeepsie, NY. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Shawn Bodily is a Senior IBM AIX® Consultant for Clear Technologies located in Dallas, Texas. He has 20 years of AIX experience and the last 17 years specializing in high availability and disaster recovery primarily focused around PowerHA. He is a double AIX advanced technical expert, IBM Power Systems and IBM Storage certified. He has written and presented extensively on high availability and storage via technical conferences, webinars, and onsite to clients. He is an IBM Redbooks platinum author co-authoring seven IBM Redbooks publications and two IBM Redpapers™.

© Copyright IBM Corp. 2015. All rights reserved.

Daniel J. Martin-Corben is a Technical Solutions Designer for IBM UK and has been working within UNIX for 20 years. He has held various roles in the sector but has finally returned to IBM. In the early days, he worked on Sequent IBM DYNIX/ptx® as a DBA, Upon joining IBM, he had his first introduction to IBM AIX and IBM HACMP™ (PowerHA) and the pSeries hardware, which has dominated his prolific career. IBM POWER8™ is his current focus, but he has extensive experience on various types of storage, which includes IBM V7000, IBM XIV®, and SAN Volume Controller. Not only does he have strong skills and knowledge with all IBM systems, but also Solaris, Symantec, HP-UX, VMware, and Windows. He has written extensively on his IBM developerWorks® blog “Power Me Up”.

Reshma Prathap is a Certified IT Specialist in Server Systems at IBM India. She is working for the India Software Lab Operations team where she is the technical lead for virtualization of IBM System p® and IBM System x servers. She has over six years of experience in Virtualization of System p and System x servers and four years of experience in implementing high availability solutions, especially PowerHA. She holds a Bachelor of Technology Degree in Electronics and Communication from Mahatma Gandhi University, India. Her areas of expertise include Linux, AIX, IBM POWER Virtualization, PowerHA SystemMirror, System Management, VMware, KVM, and IBM DB2® Database administration.

Kulwinder Singh is a Certified IT Specialist at IBM GTS-TSS. He has 16 years of information technology experience. He has been with IBM for the last seven years. His areas of expertise include AIX, IBM System p hardware, IBM storage, IBM GPFS™, PowerHA and IBM Tivoli® Storage Manager.

Ashraf Ali Thajudeen is an Infrastructure Architect in IBM Singapore GTS Services Delivery having more than eight years of experience in High Availability and Disaster Recovery Architectures in UNIX environments. He is an IBM Master Certified IT Specialist in Infrastructure and Systems Management and TOGAF 9 Certified in Enterprise Architecture. He has wide experience in designing, planning, and deploying PowerHA based solutions across ASEAN SO accounts. His areas of expertise include designing and implementing PowerHA and Tivoli automation solutions.

William Nespoli Zanatta is an IT Specialist from IBM Global Technology Services® Brazil. He has been with IBM for four years, supporting enterprise environments running AIX and Linux systems on POWER and System x. He has background experience with other UNIX versions and software development and his current areas of expertise include IBM PowerVM®, PowerHA, and GPFS.

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

Ella Buslovic
IBM International Technical Support Organization, Poughkeepsie Center

David Bennin, Noel Carroll, Mike Coffey, Richard Conway, Steven Finnes, Gary Lowther, Paul Moyer, Ravi Shankar, Scot Stansell, and Tom Weaver
IBM US

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in

length, and you can participate either in person or as a remote resident working from your home base.

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 books to be as helpful as possible. Send us your comments about this book 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 email 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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks
- Follow us on Twitter:

  https://twitter.com/ibmredbooks
- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806
- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm
- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

IBM PowerHA SystemMirror for AIX best practices

This chapter provides suggestions on how to prepare, design, install, test, maintain, monitor, and manage a PowerHA SystemMirror high-availability cluster.

This chapter contains the following topics:

- Introduction
- Designing high availability
- Cluster components
- Testing
- Maintenance
- Monitoring
- PowerHA in a virtualized world
- Summary

© Copyright IBM Corp. 2015. All rights reserved.

1.1 Introduction

IBM PowerHA SystemMirror for AIX (formerly IBM HACMP) was first available in 1991 and is now in its 24th release, with over 20,000 PowerHA clusters in production, worldwide. IBM PowerHA SystemMirror is recognized as a robust, mature high availability solution. PowerHA supports a wide variety of configurations, and offers a great deal of flexibility to the cluster administrator. With this flexibility comes the responsibility to make wise choices because many cluster configurations are available that work regarding the cluster passing verification and being brought online, but those configurations are not ideal in terms of providing availability.

This chapter publication[1] describes choices that the cluster designer can make, and suggests the alternatives that can achieve the highest level of availability.

This chapter discusses the following topics:

- Designing high availability
- Cluster components
- Testing
- Maintenance
- Monitoring
- PowerHA in a virtualized world
- Summary

1.2 Designing high availability

A fundamental goal of a successful cluster design is the elimination of single points of failure (SPOFs).

A high availability solution helps ensure that the failure of any component of the solution, whether it is hardware, software, or system management, does not cause the application and its data to be inaccessible to the user community. This solution is achieved through the elimination or masking of both planned and unplanned downtime. High availability solutions help eliminate single points of failure through appropriate design, planning, selection of hardware, configuration of software, and carefully controlled change management discipline.

To be highly available, a cluster must have no single point of failure. Although the principle of *no single point of failure* is accepted, it is sometimes inadvertently or deliberately violated. It is inadvertently violated when the cluster designer does not appreciate the consequences of the failure of a specific component. It is deliberately violated when the cluster designer chooses not to put redundant hardware in the cluster. The most common instance is when the cluster nodes that are chosen do not have enough I/O slots to support redundant adapters. This choice is often made to reduce the price of a cluster, and is generally a false economy; the resulting cluster is still more expensive than a single node, but has no better availability.

Plan a cluster carefully so that every cluster element has a backup (some say two of everything). A preferred practice is to use either paper or online planning worksheets to do this planning, and save them as part of the on-going documentation of the system. Table 1-1 on page 3 lists typical SPOFs within a cluster.

[1] This document applies to PowerHA 7.1.3 SP1 running under AIX 7.1.3 TL1.

Base the cluster design decisions on whether the cluster designs contribute to availability (that is, eliminate an SPOF) or detract from availability (gratuitously complex).

Table 1-1 Eliminating SPOFs

Cluster object	Eliminated as a single point of failure by these methods
Node	Use multiple nodes.
Power source	Use multiple circuits or uninterruptible power supplies (UPSs).
Network adapter	Use redundant network adapters.
Network	Use multiple networks to connect nodes.
TCP/IP subsystem	Use non-IP networks to connect adjoining nodes and clients.
Disk adapter	Use redundant disk adapter or multipath hardware.
Disk	Use multiple disks with mirroring or RAID.
Application	Add a node for takeover; configure application monitor.
Administrator	Add backup or very detailed operation guide.
Site	Add an additional site.

1.2.1 Risk analysis

Sometimes in reality, eliminating all SPOFs within a cluster is not feasible. Examples might include network and site:

- If the network as a SPOF must be eliminated, the cluster requires at least two networks. Unfortunately, this eliminates only the network that is directly connected to the cluster as a SPOF. It is not unusual for the users to be located some number of hops away from the cluster. Each of these hops involves routers, switches, and cabling, and each typically represents another SPOF. Truly eliminating the network as a SPOF can become a massive undertaking.
- Eliminating the site as a SPOF depends on distance and the corporate disaster recovery strategy. Generally, this involves using PowerHA SystemMirror Enterprise Edition. However, if the sites can be covered by a common storage area network, for example buildings within a 2 km radius, cross-site Logical Volume Manager (LVM) mirroring function as described in the *PowerHA Administration Guide* is most appropriate, providing the best performance at no additional expense. If the sites are within the range of Peer-to-Peer Remote Copy (PPRC) (roughly, 100 km) and compatible IBM ESS, DS8000®, SAN Volume Controller storage systems are used, then one of the PowerHA SystemMirror Enterprise Edition PPRC technologies is appropriate. Otherwise, consider PowerHA SystemMirror Global Logical Volume Manager (GLVM). For more information, see *IBM PowerHA Cookbook for AIX Updates*, SG24-7739.

Risk analysis techniques can be used to determine the SPOFs that simply must be handled and SPOFs that can be tolerated, as in this example:

- Study the current environment. Is the server room on a properly sized UPS, but no disk mirroring occurs today?
- Perform requirements analysis. How much availability is required and what is the acceptable likelihood of a long outage?
- Hypothesize all possible vulnerabilities. What might go wrong?

- Identify and quantify risks. What is the cost estimate of a failure versus the probability that it occurs?
- Evaluate counter-measures. What is required to reduce the risk or consequence to an acceptable level?
- Finally, make decisions, create a budget, and design the cluster.

1.3 Cluster components

The following section describes preferred practices for important cluster components.

1.3.1 Nodes

PowerHA v7.1 supports clusters of up to 16 nodes, with any combination of active and standby nodes. Although a possibility is to have all nodes in the cluster running applications (a configuration referred to as *mutual takeover*), the most reliable and available clusters have at least one standby node: one node that is normally not running any applications, but is available to take them over if a failure occurs on an active node.

Also, be sure to attend to environmental considerations. Nodes should not have a common power supply, which can happen if they are placed in a single rack. Similarly, building a cluster of nodes that are actually logical partitions (LPARs) with a single footprint is useful as a test cluster, but do not consider them for availability of production applications.

Choose nodes that have sufficient I/O slots to install redundant network and disk adapters (twice as many slots as is required for single node operation). This naturally suggests avoiding processors with small numbers of slots. For high availability best practices, do not consider or plan to use a node unless it has redundant adapters. Blades are an outstanding example. And, just as every cluster resource should have a backup, the root volume group in each node should be mirrored, or be on a RAID device. Furthermore, PowerHA v7.1 added the rootvg system event, which monitors rootvg and can help invoke a fallover in the event of rootvg loss.

Also, choose nodes so that, when the production applications are run at peak load, sufficient CPU cycles and I/O bandwidth still exist to allow PowerHA to operate. The production application should be carefully benchmarked (preferable) or modeled (if benchmarking is not feasible) and nodes chosen so that they do not exceed 85% busy, even under the heaviest expected load.

Note: Size the takeover node to accommodate all possible workloads: if a single standby is backing up multiple primaries, it must be capable of servicing multiple workloads.

On hardware that supports dynamic LPAR operations, PowerHA can be configured to allocate processors and memory to a takeover node before applications are started. However, these resources must be available, or acquirable through Capacity Upgrade on Demand (CUoD). Understand and plan for the worst case situation where, for example, all the applications are on a single node.

1.3.2 Networks

PowerHA is a network-centric application. PowerHA networks not only provide client access to the applications but are used to detect and diagnose node, network, and adapter failures. To do this, PowerHA uses these methods, which send heartbeats over *all* defined networks:

- Before PowerHA v7: Reliable Scalable Cluster Technology (RSCT)
- PowerHA v7 and later: Cluster Aware AIX (CAA)

By gathering heartbeat information about multiple nodes, PowerHA can determine what type of failure occurred and initiate the appropriate recovery action. Being able to distinguish between certain failures, for example the failure of a network and the failure of a node, requires a second network. Although this additional network can be “IP based,” it is possible that the entire IP subsystem can fail within a given node. Therefore, in addition there should be at least one, ideally two, non-IP networks. Failure to implement a non-IP network can potentially lead to a partitioned cluster, sometimes referred to as the *split brain syndrome*. This situation can occur if the IP network between nodes becomes severed or in some cases congested. Because each node is still alive, PowerHA concludes the other nodes are down and initiates a takeover. After takeover, one or more applications might be running simultaneously on both nodes. If the shared disks are also online to both nodes, the result can lead to data divergence (massive data corruption). This is a situation that must be avoided, at all costs.

Starting in PowerHA v7 with the use of CAA, the new cluster repository disk automatically provides a form of non-IP heartbeating. Another option is to use SAN heartbeat, which is commonly referred to as *sancomm* or by the device name it uses called *sfwcomm*. Using sancomm requires SAN adapters that support *target mode* and zoning the adapters together so they can communicate with each other.

Important network best practices for high availability are as follows:

- Failure detection is possible only if at least two physical adapters per node are in the same physical network or VLAN. Be extremely careful when you make subsequent changes to the networks, with regards to IP addresses, subnetmasks, intelligent switch port settings, and VLANs.
- The more unique types, both IP and non-IP, of networks the less likely of ever reporting a false-node-down failure.
- Where possible, use Etherchannel, Shared Ethernet Adapters (SEAs), or both, through the Virtual I/O Server (VIOS) with PowerHA to aid availability.

> **Note:** PowerHA sees Etherchannel configurations as single adapter networks. To aid problem determination, configure the `netmon.cf` file to allow ICMP echo requests to be sent to other interfaces outside of the cluster. See the PowerHA administration web page for further details:
>
> `http://www-01.ibm.com/support/knowledgecenter/SSPHQG_7.1.0/com.ibm.powerha.admngd/ha_admin_kickoff.htm`

- When you use multiple adapters per network, each adapter needs an IP address in a different subnet, using the same subnet mask.
- Currently, PowerHA supports IPv6 and Ethernet only.

- Ensure that you have in place the correct network configuration rules for the cluster with regards to Etherchannel, Virtual adapter support, service, and persistent addressing. For more information, see the PowerHA planning web page:

  http://www-01.ibm.com/support/knowledgecenter/SSPHQG_7.1.0/com.ibm.powerha.plangd/ha_plan.htm
- Name resolution is essential for PowerHA. External resolvers are deactivated under certain event processing conditions. Avoid problems by configuring `/etc/netsvc.conf` and NSORDER variable in `/etc/environment` to ensure that the host command checks the local `/etc/hosts` file first.
- Read the release notes that are stored in `/usr/es/sbin/cluster/release_notes`. Watch for new and enhanced features, such as collocation rules, persistent addressing, and fast failure detection.
- Configure persistent IP labels to each node. These IP addresses are available at AIX boot time and PowerHA strives to keep them highly available. They are useful for remote administration, monitoring, and secure node-to-node communications. Consider implementing a host-to-host IPSec tunnel between persistent labels between nodes. This can ensure that sensitive data, such as passwords, are not sent unencrypted across the network, for example when using the C-SPOC option to change a user password.
- If you have several virtual clusters split across frames, ensure boot subnet addresses are unique per cluster. This minimizes problems with `netmon` reporting the network is up when indeed the physical network outside the cluster might be down.

1.3.3 Adapters

As stated previously, each network that is defined to PowerHA should have at least two adapters per node. Although it is possible to build a cluster with fewer, the reaction to adapter failures is more severe; the resource group must be moved to another node. AIX provides support for both Etherchannel and Shared Ethernet Adapters. This often allows the cluster node to logically have defined one adapter interface per network. This reduces the number of IP addresses required, allows the boot IP address and service IP to be on the same subnet, and can result in not needing to define persistent addresses.

Many IBM Power Systems servers contain built-in virtual Ethernet adapters. These historically have been known as Integrated Virtual Ethernet (IVE) or Host Ethernet Adapters (HEAs). Some newer systems now contain Single Root I/O Virtualization (SRIOV) adapters. Most of these adapters provide multiple ports. One port on such an adapter should not be used to back up another port on that adapter because the adapter card is a common point of failure. The same is often true of the built-in Ethernet adapters; in most IBM Power Systems servers, ports have a common adapter. When the built-in Ethernet adapter can be used, a preferred practice is to provide an extra adapter in the node, with the two backing up each other. However, be aware that, when using these specific types of adapters, in many cases, Live Partition Mobility might be unable to be used.

Also be aware of network detection settings for the cluster and consider tuning these values. These values apply to *all* networks. However, be careful when you use custom settings because setting these values too low can lead to undesirable results, like false takeovers. These settings can be viewed and modified by using either the **`clmgr`** command or **`smitty sysmirror`**.

1.3.4 Applications

The most important part of making an application run well in a PowerHA cluster is understanding the application's requirements. This is particularly important when designing the *resource group* policy behavior and dependencies. For high availability to be achieved, the application must be able to stop and start cleanly and not explicitly prompt for interactive input. Some applications tend to bond to a particular operating system characteristic such as a uname, serial number, or IP address. In most situations, these problems can be overcome. A vast majority of commercial software products that run under AIX are suited to be clustered with PowerHA.

Application data location

Where should application binaries and configuration data reside? There are many arguments to this discussion. Generally, keep all the application binaries and data where possible on the shared disk because forgetting to update it on all cluster nodes when it changes is easy. This can prevent the application from starting or working correctly when it is run on a backup node. However, the correct answer is not firm. Although many application vendors have suggestions for how to set up the applications in a cluster, these are recommendations. Just when it seems to be clear cut as to how to implement an application, someone thinks of a new set of circumstances. Here are several guidelines:

- If the application is packaged in LPP format, it is usually installed on the local file systems in rootvg. This behavior can be overcome by storing the installation packages to disk by using the **`bffcreate`** command and then restoring them with the preview option. This action shows the installation paths, and then symbolic links can be created before installation, which points to the shared storage area.
- If the application is to be used on multiple nodes with different data or configurations, the application and configuration data are probably on local disks and the data sets on shared disk with application scripts, altering the configuration files during fallover.
- Also, remember the PowerHA file collections facility can be used to keep the relevant configuration files in sync across the cluster. This is useful for applications that are installed locally.

Start and stop scripts

Application *start* scripts should not assume the status of the environment. Intelligent programming must correct any irregular conditions that might occur. The cluster manager spawns theses scripts in a separate job in the background and carries on processing. Some tasks that a start script should perform are as follows:

1. Check that the application is not currently running. This is important because resource groups can be placed into an unmanaged state (forced-down action, in previous versions). Using the default startup options, PowerHA will rerun the application start script, which might cause problems if the application is running. A simple and effective solution is to check the state of the application on startup. If the application is found to be running, end the start script with `exit 0`.
2. Verify the environment. Are all the disks, file systems, and IP labels available?
3. If different commands are to be run on different nodes, store the executing HOSTNAME to a variable.
4. Check the state of the data. Does it require recovery? Always assume that the data is in an unknown state since the conditions that occurred to cause the takeover cannot be assumed.
5. Do prerequisite services exist that must be running? Is it feasible to start all prerequisite services from within the start script? Is there an inter-resource group dependency or

resource group sequencing that can guarantee that the previous resource group started correctly? PowerHA can implement checks on resource group dependencies including collocation rules.

6. When the environment looks correct, start the application. If the environment is not correct and error recovery procedures cannot fix the problem, ensure that adequate alerts (email, SMS, SMTP traps, and others) are sent over the network to the appropriate support administrators.

The *stop* scripts differ from start scripts in that most applications have a documented start-up routine and not necessarily a stop routine. The assumption is that once the application is started, why stop it? Relying on a failure of a node to stop an application will be effective, but to use some of the more advanced features of PowerHA, the requirement exists to stop an application cleanly. Avoid the following issues, among others:

- Be sure to terminate any child or spawned processes that might be using disk resources. Consider implementing child resource groups.
- Verify that the application is stopped to the point that the file system is free to be unmounted. The **`fuser`** command can verify that the file system is free.
- In some cases, it might be necessary to double-check that the application vendor's stop script stopped all the processes; sometimes it might be necessary to terminate some processes by force. Clearly the goal is to return the machine to the state it was in before the application start script was run.
- Failure to exit the stop script with a zero return code will stop cluster processing.

> **Note:** This is not the case with start scripts when using the background startup option.

Remember, most vendor stop/starts scripts are not designed to be cluster proof. A useful tip is to have stop and start script verbosely output using the same format to the `/tmp/hacmp.out` file. This can be achieved by including the following line in the header of the script:

```
set -x && PS4="${0##*/}"'[$LINENO] '
```

Application monitoring

With PowerHA, you can monitor the state of an application. Although optional, implementation is highly suggested. This mechanism provides for self-healing clusters. To ensure that event processing does not hang because of failures in the user-supplied script and to prevent hold-up during event processing, PowerHA has always started the application in the background. This approach has disadvantages:

- There is no wait or error checking.
- In a multitiered environment, there is no easy way to ensure that applications of higher tiers have been started.

Application monitoring can either check for process death or run a user-supplied custom monitor method during the start-up or continued running of the application. The latter is particularly useful when the application provides some form of transaction processing: a monitor can run a null transaction to ensure that the application is functional. The preferred practice for applications is to have both process death and user-supplied application monitors in place.

More information about application monitoring is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

Do not forget to test the monitoring, and start, restart, and stop methods carefully. Poor start, stop, and monitor scripts can cause cluster problems, not only in maintaining application availability but avoiding data corruption.

Behavior: By having monitoring scripts exit with nonzero return codes when the application has not failed, in conjunction with poor start/stop scripts, can result in undesirable behavior (for example, data corruption). The application is down and is also in need of emergency repair that might involve restoring data from backup.

In addition, PowerHA also supplies a number of tools and utilities to help in customization efforts like pre-event and post-event scripts and user-defined resources. Be careful to use only those for which PowerHA also supplies a man page (`lslpp -f cluster.man.en_US.es.data`) because those are the only ones for which upward compatibility is guaranteed. A good example for this use is application provisioning.

Application provisioning

PowerHA can drive Dynamic LPAR and some Capacity on Demand (CoD) operations to ensure that adequate processing and memory are available for one or more applications upon start-up. This is shown in Figure 1-1. Also see the following web page for information about supported Capacity Upgrade on Demand (CUoD) types:

http://www-01.ibm.com/support/knowledgecenter/SSPHQG_7.1.0/com.ibm.powerha.admngd/ha_admin_types_cuod_licenses.htm

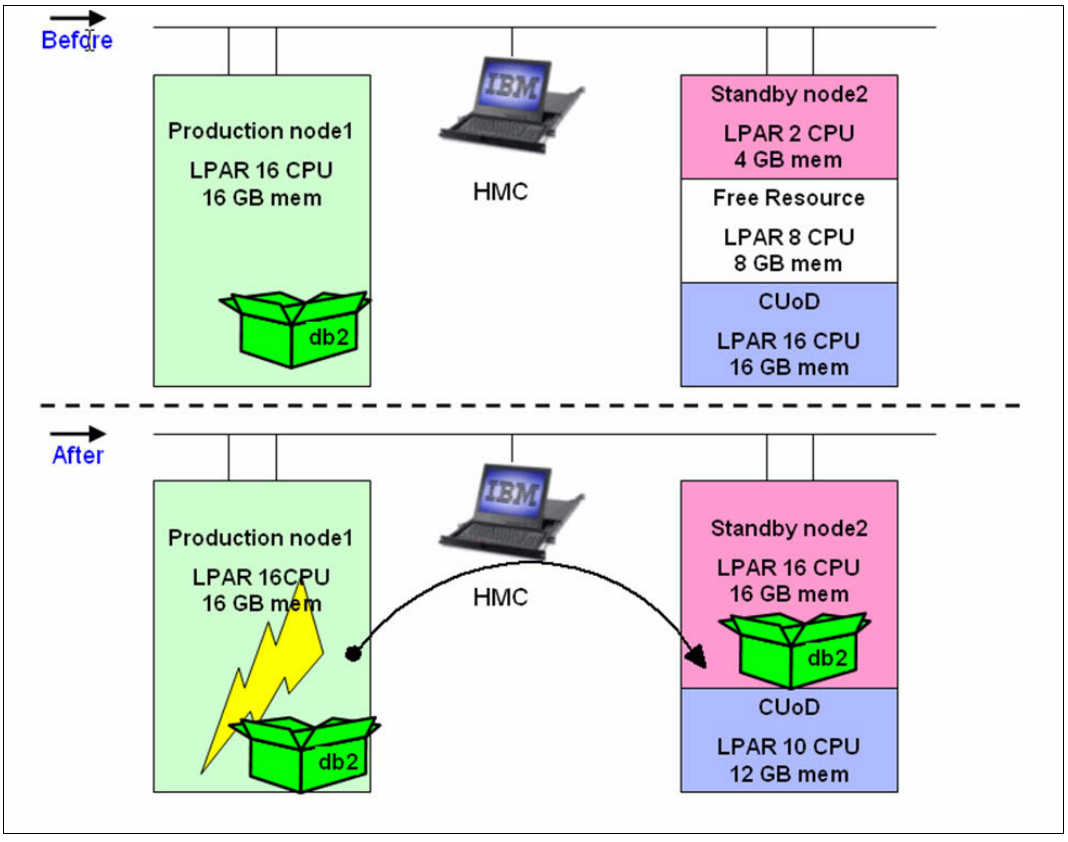

Figure 1-1 Application provisioning example

This process can be driven by using PowerHA SMIT panels. However, consider the following information about this approach:

- The CoD activation key must be entered manually *before* any PowerHA dynamic logical partitioning (DLPAR) event.
- The LPAR name must be the AIX operating system host name, which must be the PowerHA node name.
- Large memory moves are actioned in one operation. This can be time-consuming and delay event processing.
- The LPAR host name must be resolvable at the Hardware Management Console (HMC).
- If the acquisition or release fails, the operation is not repeated on another HMC if defined.

More detail about using and configuring this option is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

1.4 Testing

Although this statement sounds simplistic, the most important step in testing is to actually do it.

A cluster should be thoroughly tested before initial production (and after **clverify** command runs without errors or warnings). This means that every cluster node and every interface that PowerHA uses must be stopped and started again to validate that PowerHA responds as expected. Be sure to perform the same level of testing after each change to the cluster. PowerHA offers a cluster test tool that can be run on a cluster before it is put into production. This will verify that the applications are brought back online after node, network, and adapter failures. Run the test tool as part of any comprehensive cluster test effort.

More information about the cluster test tool is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

In addition, plan for regular testing. A common safety recommendation is to test home smoke detectors twice a year; the switch to and from Daylight Saving Time are commonly when to test. Similarly, if the enterprise can afford to schedule it, node fallover and fallback tests should be scheduled bi-annually. These tests at least indicate whether any problems have crept in, and allow for correction before the cluster fails in production.

On a more regular basis, run the **clverify** command. Seriously consider errors and warning messages, and correct those problems at the first opportunity. The **clverify** command runs automatically daily at 00:00 hours. Part of the practice for administrators should be to routinely check the logs daily, and react to any warnings or errors.

1.5 Maintenance

Even the most carefully planned and configured cluster might experience problems if it is not well maintained. A large part of best practice for a PowerHA cluster is associated with maintaining the initial working state of the cluster through hardware and software changes.

Before any change to a cluster node, take a PowerHA snapshot. If the change involves installing a PowerHA, AIX, or other software fix, also take a **mksysb** backup, use **multibos**, or

alt_disk_install (alt_disk_copy, alt_clone, alt_disk_mksysb are useful options). Also *apply* fixes and updates instead of *commit*. In this way, removing fixes, if necessary, is easier.

On successful completion of the change, use SMIT to display the cluster configuration, print, and save the smit.log file. The **clmgr** facility can also be used to generate an HTML report of the cluster configuration in PowerHA v7.1.3.

All mission-critical high-availability cluster enterprises should, as a preferred practice, maintain a test cluster identical to the production ones. Thoroughly test all changes to applications, cluster configuration, or software on the test cluster before putting them on the production clusters. To at least partially automate this effort, use the PowerHA cluster test tool.

Change control is important in a PowerHA cluster. In some organizations, databases, networks, and clusters are administered by separate individuals or groups. When any group plans maintenance on a cluster node, it should be planned and coordinated among all parties. All should be aware of the changes being made to avoid introducing problems. Organizational policy must preclude “unilateral” changes to a cluster node. In addition, change control in a PowerHA cluster must include a goal of having all cluster nodes at the same level. Upgrading only the node running the application is insufficient (and not recommended). Develop a process that encompasses the following set of questions:

- Is the change necessary?
- How urgent is the change?
- How important is the change? (This is not the same as urgent.)
- What impact does the change have on other aspects of the cluster?
- What is the impact if the change is not allowed to occur?
- Are all steps needed to implement the change clearly understood and documented?
- How will the change be tested?
- What is the plan for backing out the change if necessary?
- Is the appropriate expertise available if problems develop?
- When is the change scheduled?
- Were the users notified?
- Does the maintenance period include sufficient time for a full set of backups before the change and sufficient time for a full restore afterwards should the change fail testing?

This process should include an electronic form, which requires appropriate signoffs before the change can go ahead. Every change, even the minor ones, must follow the process. The notion that a change, even a small change, might be permitted (or sneaked through) without following the process must not be permitted.

To this end, the preferred practice is to use the PowerHA C-SPOC facility, or C-SPOC command-line equivalent, where possible for any change. Especially with regards to shared volume groups. If the installation uses AIX password control on the cluster nodes (as opposed to NIS or LDAP), C-SPOC should also be used for any changes to users and groups. PowerHA will then ensure that the change is properly reflected to all cluster nodes.

More information about cluster maintenance and administration is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

1.5.1 Upgrading the cluster environment

OK, so you want to upgrade? Start by reading the upgrade chapter in the PowerHA installation documentation and make a detailed plan:

http://ibm.co/1qkduDw

Taking the time to review and plan thoroughly will save many “I forgot to do that” problems during and after the migration or upgrade process. Remember to check all version compatibilities between the levels of software and firmware, and, most important, the application software certification against the level of AIX and PowerHA. If you are not sure, check with IBM support or use the Fix Level Recommendation Tool (FLRT):

http://www14.software.ibm.com/webapp/set2/flrt/home

Do not attempt to upgrade AIX or PowerHA without first taking a backup and checking that it is restorable. In all cases, completing the process in a test environment before actually doing it for real is extremely useful. AIX facilities, such as **alt_disk_copy** and **multibos** for creating an alternative rootvg, which can be activated by rebooting, and are useful tools worth exploring and using.

Before attempting the upgrade, complete the following steps:

1. Check that cluster and application are stable and that the cluster can synchronize cleanly.
2. Take a cluster snapshot and save it to a temporary non-cluster directory (export SNAPSHOTPATH=<some other directory>).
3. Save event script customization files and user-supplied scripts to a temporary non-cluster directory. If you are unsure that any custom scripts are included, check by using **odmget HACMPcustom** command.
4. Check that the same level of cluster software (including program temporary fixes (PTFs)) are on all nodes before beginning a migration.
5. Ensure that the cluster software is committed (and not just applied).

Where possible, use the rolling migration method to ensure maximum availability. Effectively, cluster services are stopped one node at a time by using the takeover option (now referred to as *move resource groups*). The node or system is updated accordingly and cluster services are restarted. This operation is completed one node at a time until all nodes are at the same level and are operational.

Note: Although PowerHA will work with mixed levels of AIX or PowerHA in the cluster, the goal is to have all nodes at exactly the same levels of AIX, PowerHA, and application software. In addition, PowerHA prevents changes to the cluster configuration when mixed levels of PowerHA are present.

PowerHA service packs can now be applied using a *nondisruptive update* method. The process is identical to the rolling migration; however, resource groups are placed into an *unmanaged state* to ensure that they remain available and performed one at a time from start to finish.

Note: During this state, the application (or applications) is not under the control of PowerHA (for example, not highly available). Using the default startup options, PowerHA relies on an application monitor to determine the application state and hence appropriate actions to take.

Alternatively, the entire cluster and applications can be gracefully shut down to update the cluster using either the *snapshot* or *offline* conversion methods. Historically, upgrading the cluster this way has resulted in fewer errors but requires a period of downtime.

Tip: Demos are available for performing migrations; see the following web address:

http://www.youtube.com/PowerHAguy

More information about the migration process is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

1.6 Monitoring

In a clustered environment, gaining timely and accurate status information regarding the cluster topology and application resources is critical. Also critical is for application monitors to be configured for each application that is to be made highly available in the cluster[2]. Without application monitors, PowerHA has no mechanism to know whether your applications are available and performing as you expect.

PowerHA provides commands such as **cldump** and **clstat** for monitoring the status of the cluster.

The SNMP protocol is the crux to obtaining the status of the cluster. The SNMP protocol is used by network management software and systems for monitoring network applications and devices for conditions that warrant administrative attention. SNMP protocol consists of a database, and a set of data objects. The set of data objects forms a Management Information Base (MIB). The standard SNMP agent is the **snmpd** daemon. An SNMP Multiplexing protocol (SMUX) subagent allows vendors to add more MIB information that is product-specific. The **clstrmgr** daemon in PowerHA acts as a SMUX subagent. The SMUX peer function, contained in the **clstrmgrES** daemon, maintains cluster status information for the PowerHA MIB. When the clstrmgrES starts, it registers with the SNMP daemon, **snmpd**, and continually updates the MIB with cluster status information in real time. PowerHA implements a private MIB branch maintained through a SMUX peer subagent to SNMP contained in **clstrmgrES**, as shown in Figure 1-2.

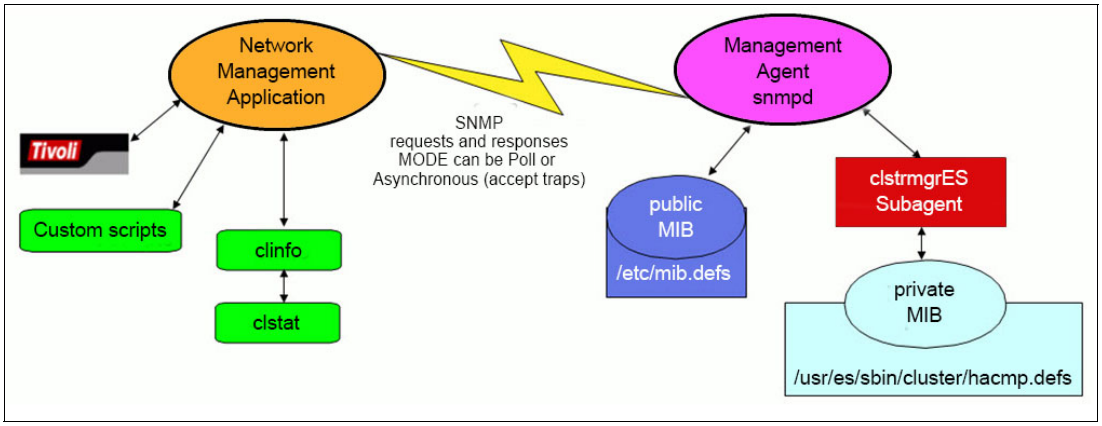

Figure 1-2 PowerHA private Managed Information Base (MIB)

The **clinfo** daemon status facility has a few considerations and many users or administrators of PowerHA clusters implement custom monitoring scripts. This might seem complex but it is remarkably straight forward. The cluster SNMP MIB data can be pulled simply over a secure session by using the following command:

```
ssh $NODE snmpinfo -v -m dump -o /usr/es/sbin/cluster/hacmp.defs risc6000clsmuxpd
> $OUTFILE
```

PowerHA participates under the IBM Enterprise SNMP MIB (Figure 1-3 on page 14):

[2] Application Monitoring is a feature of PowerHA and aides the cluster in determining whether the application is alive and well. Application Monitoring is beyond the scope of this chapter.

ISO (1) → Identified Organization (3) → Department of Defense (6) → Internet (1) → Private (4) → Enterprise (1) → IBM (2) → IBM Agents (3) → AIX (1) → aixRISC6000 (2) → risc6000agents (1) → risc6000clsmuxpd (5)

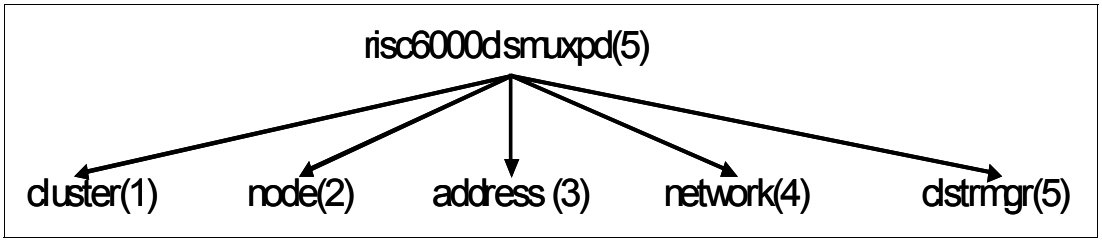

Figure 1-3 PowerHA cluster MIB structure

The resultant MIB for PowerHA **cluster** would be 1.3.6.1.4.1.2.3.1.2.1.5.1. The data held within this MIB can be pulled using the **snmpinfo** command as shown in Example 1-1.

Example 1-1 snmpinfo command

```
# snmpinfo -v -m dump -o /usr/es/sbin/cluster/hacmp.defs cluster
clusterId.0 = 1120652512
clusterName.0 = "sapdemo71_cluster"
clusterConfiguration.0 = ""
clusterState.0 = 2
clusterPrimary.0 = 1
clusterLastChange.0 = 1386133818
clusterGmtOffset.0 = 21600
clusterSubState.0 = 32
clusterNodeName.0 = "mhoracle1"
clusterPrimaryNodeName.0 = "mhoracle1"
clusterNumNodes.0 = 2
clusterNodeId.0 = 1
clusterNumSites.0 = 0
```

Individual elements, for example the cluster state and cluster sub state, can be pulled as shown in Example 1-2.

Example 1-2 Showing the cluster state

```
# snmpinfo -v -o /usr/es/sbin/cluster/hacmp.defs ClusterState.0
clusterState.0 = 2

# snmpinfo -v -o /usr/es/sbin/cluster/hacmp.defs ClusterSubState.0
clusterSubState.0 = 32

Note: the -v translates the numbered MIB branch path to reable variable name.

# snmpinfo -o /usr/es/sbin/cluster/hacmp.defs ClusterState.0
1.3.6.1.4.1.2.3.1.2.1.5.1.4.0 = 2
```

Example 1-2 shows that the cluster has a state of 2 and a substate of 32. To determine the meaning of these values, see the `/usr/es/sbin/cluster/hacmp.my` file, which contains a description of each HACMP MIB variable (Example 1-3 on page 15).

Example 1-3 Snapshot of the HACMP MIB definition file

```
clusterState    OBJECT-TYPE
                SYNTAX  INTEGER { up(2), down(4),
                                  unknown(8), notconfigured(256) }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The cluster status"

clusterSubState OBJECT-TYPE
                SYNTAX  INTEGER { unstable(16), error(64),
                                  stable(32), unknown(8), reconfig(128),
                                  notconfigured(256), notsynced(512) }
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The cluster substate"
```

We can conclude from Example 1-3 that the cluster status is *up* and *stable*. This is the mechanism that **clinfo**/**clstat** uses to display the cluster status.

The **clstat** utility uses clinfo library routines (through **clinfo** daemon) to display all node, interface, and resource group information for a selected cluster. The **cldump** does likewise as a one-off command by interrogating the private MIB directly within the cluster node. Both are solely reliant on the SNMP protocol and rely on the mechanism described previously.

Graphical monitoring can also be performed from IBM Systems Director by using the PowerHA SystemMirror plug-in.

More information and options about cluster monitoring are in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

1.7 PowerHA in a virtualized world

PowerHA works with virtual devices, however some restrictions apply when using virtual Ethernet or virtual disk access. Creating a cluster in a virtualized environment will add new SPOFs, which must be considered. PowerHA nodes inside the same physical footprint (frame) must be avoided if high availability is to be achieved; consider this configuration only for test environments. To eliminate the additional SPOFs in a virtual cluster, implement the use of a second VIOS in each frame with the Virtual I/O Client (VIOC) LPARs that are within different frames, ideally some distance apart.

Redundancy for disk access can be achieved through LVM mirroring, RAID, and Multi-Path I/O (MPIO). LVM mirroring is most suited to eliminate the VIOC rootvg as a SPOF, as shown in Figure 1-4 on page 16. The root volume group can be mirrored using standard AIX practices. In the event of VIOS failure, the LPAR will see stale partitions and the volume group must be resynchronized by using **syncvg**. This procedure can also use logical volumes as backing storage to maximize flexibility. For test environments, whereby each VIOC is in the same frame, LVM mirroring can be used for datavgs also.

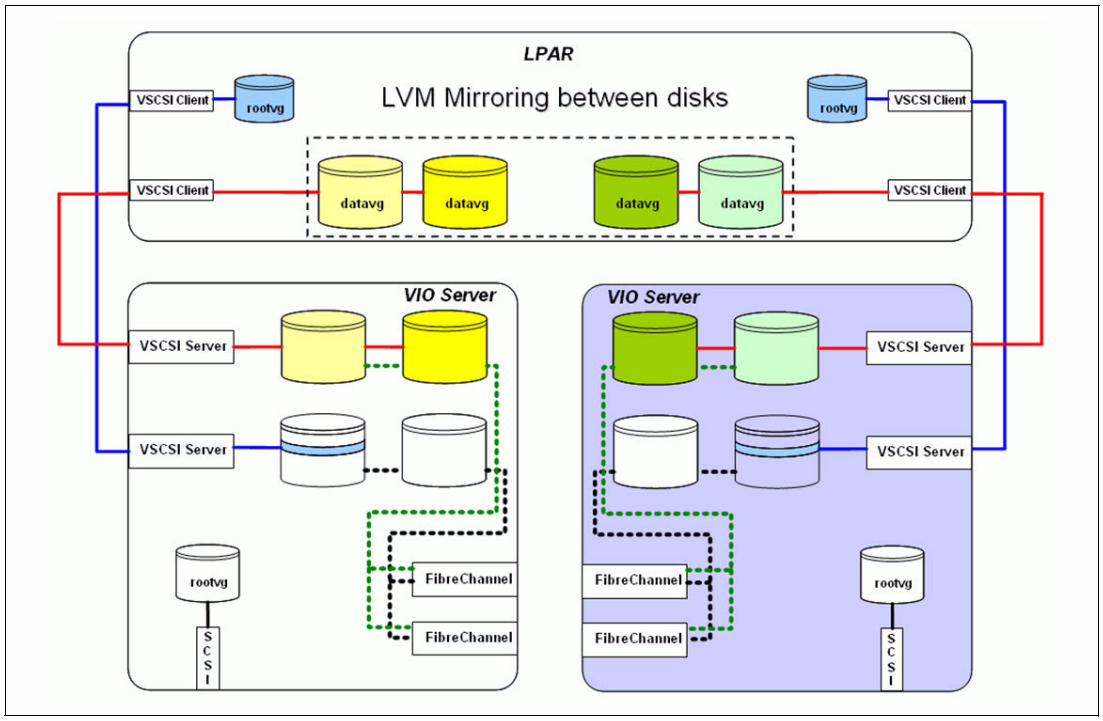

Figure 1-4 Redundancy using LVM Mirroring

For shared data volume groups, deploy the MPIO method (Figure 1-5 on page 17). A LUN is mapped to both VIOS in the SAN. From both Virtual I/O Servers, the LUN is mapped again to the same VIOC. The VIOC LPAR will correctly identify the disk as an MPIO-capable device and create one hdisk device with two paths. The configuration is then duplicated on the backup frame or node. Currently, the virtual storage devices work only in failover mode, other modes are not yet supported. All devices accessed through a VIOS must support a `no_reserve` attribute. If the device driver is not able to “ignore” the reservation, the device cannot be mapped to a second VIOS. Currently, the reservation held by a VIOS cannot be broken by PowerHA, hence only devices that will not be reserved on open are supported. Therefore, PowerHA requires the use of enhanced concurrent mode volume groups (ECVGs). The use of ECVGs is a general requirement starting with PowerHA v7.1.0.

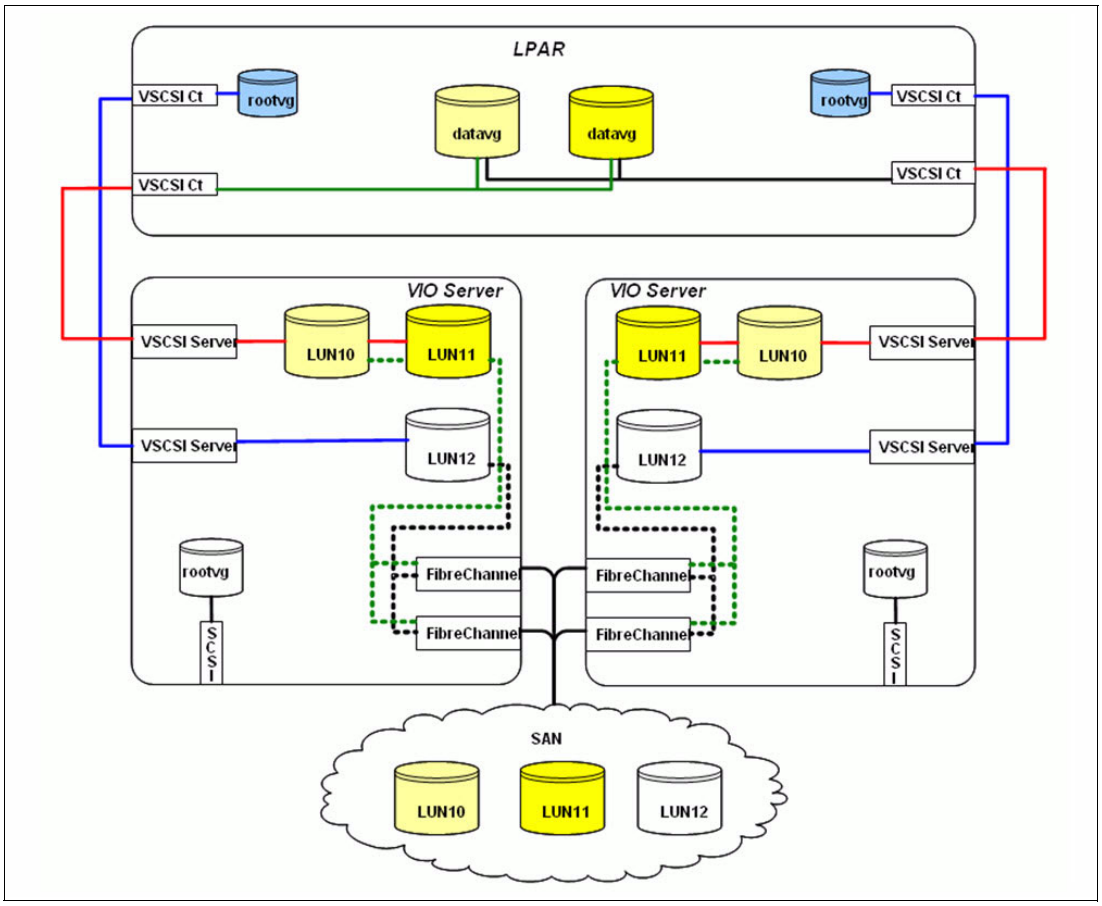

Figure 1-5 Redundancy using MPIO

In a virtualized networking environment, a VIOS is needed for access to the outside world via a layer-2 based Ethernet bridge, which is referred to as a Shared Ethernet Adapter (SEA). Now, the physical network devices along with the SEA are the new SPOFs. How are these SPOFs eliminated? Again by using a second VIOS. Etherchannel technology from within the VIOS can be used to eliminate both the network adapters and switch as a SPOF. To eliminate the VIOS as a SPOF, two choices are available:

- Etherchannel (configured in backup mode *only, no aggregation*) in the VIOC. See Figure 1-6 on page 18.
- SEA failover through the hypervisor. See Figure 1-7 on page 19.

Both methods have advantages and disadvantages. However, SEA failover is generally considered the preferred practice because it provides the use of Virtual LAN ID (VID) tags and keeps the client configuration cleaner.

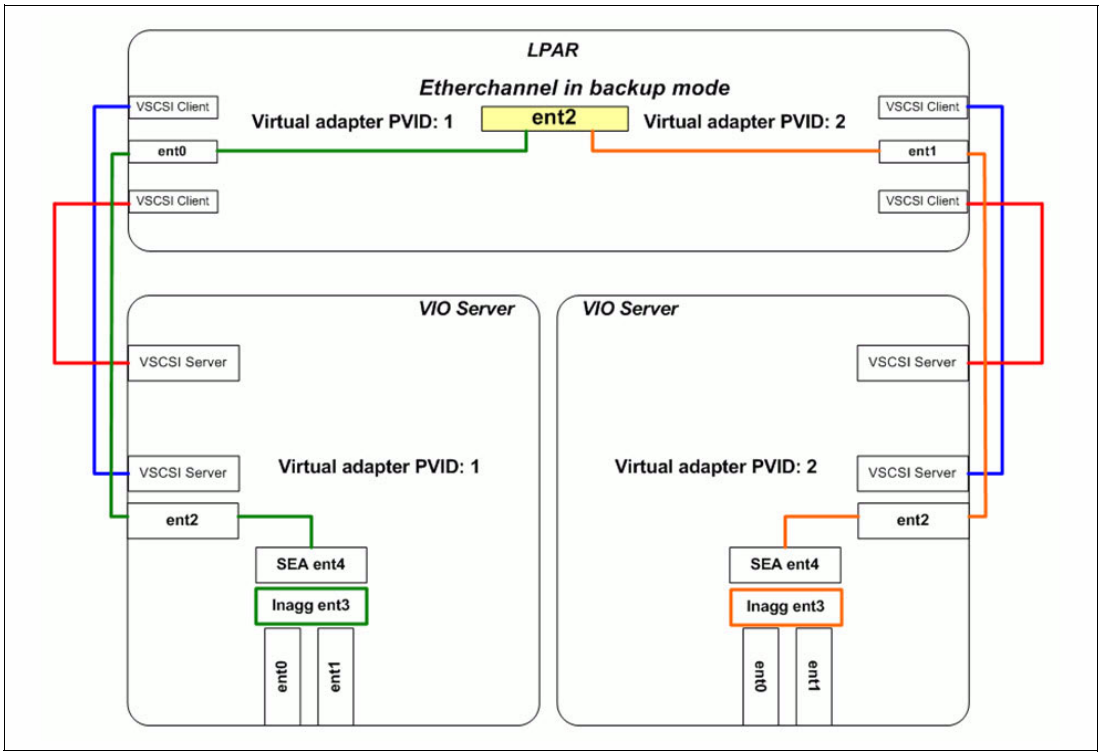

Figure 1-6 Etherchannel in backup mode

From the client perspective only a single virtual adapter is required. However, having a second virtual adapter will not eliminate a SPOF because the adapter is not real. The SPOF is the hypervisor. Generally, single interface networks are not a preferred practice because this limits the error detection capabilities of PowerHA. In this case, it cannot be avoided so to help with further analysis, add external IP addresses to the `netmon.cf` file. In addition, at least two physical adapters per SEA should be used in the VIOS in an Etherchannel configuration. Adapters in this channel can also form an aggregate, but remember that most vendors require adapters that form an aggregate to share the same backplane (a SPOF, so do not forget to define a backup adapter). An exception to this rule is Nortel's Split Multi-Link Trunking. Depending on your environment, this technology might be worth investigating.

Figure 1-7 on page 19 shows a SEA failover.

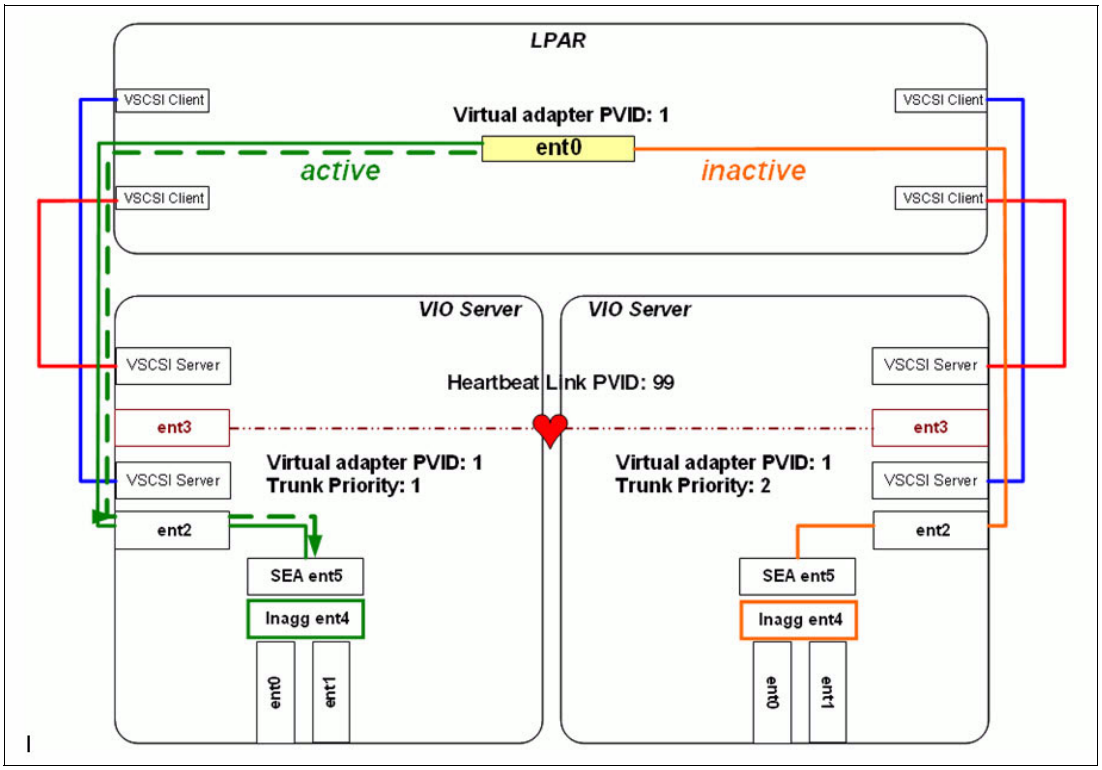

Figure 1-7 SEA failover

Finally, you see a view of the big picture. Be methodical in your planning. As Figure 1-8 on page 20 shows, even a simple cluster design can soon become rather complex.

More information about implementing PowerHA in a virtualized environment, including the use of NPIV, is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

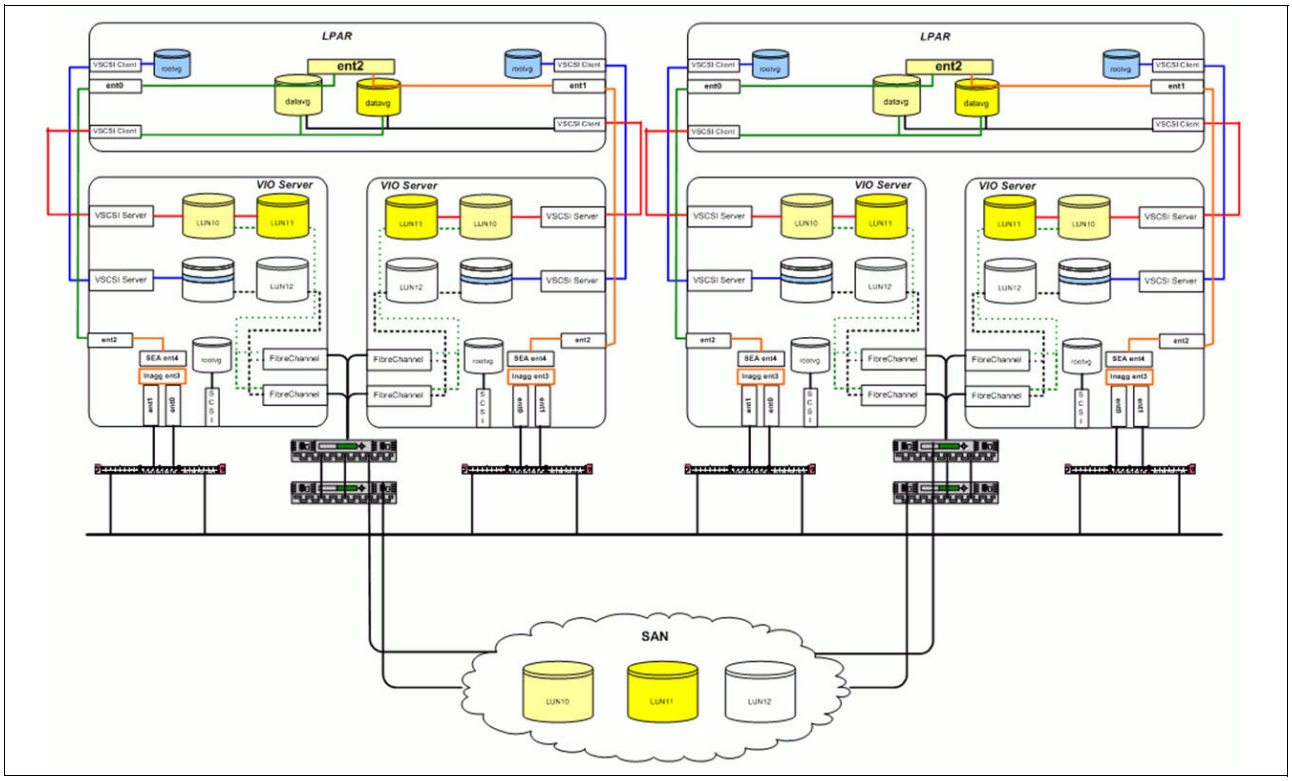

Figure 1-8 A PowerHA cluster in a virtualized world

1.7.1 Maintenance of the VIOS partition: Applying updates

The VIOS must be updated in isolation, that is, with no client access. A simple way of achieving this is to start by creating a new profile for the VIOS by copying the existing one. Then, delete all virtual devices from the profile and reactivate the VIOS using the new profile. This ensures that no client partition can access any devices and that the VIOS is ready for maintenance.

Before restarting the VIOS, manual failover from the client must be performed so all disk access and networking goes through the alternate VIOS. The steps to accomplish this are as follows:

1. For MPIO storage, disable the activate path by using the following command:

   ```
   chpath -l hdiskX -p vscsiX -s disable
   ```

2. For LVM mirrored disks, set the virtual SCSI target devices to a defined state in the VIOS partition.
3. Initiate the SEA failover from the active VIOS by using the following command:

   ```
   chdev -dev entX -attr ha_mode=auto
   ```

4. For Etherchannel in the VIOC, initiate a force failover by using the following command:

   ```
   smitty etherchannel
   ```

After the update is applied, reboot the VIOS. The client is then redirected to the newly updated VIOS and the same procedure is followed on the alternative VIOS. An important factor is to be sure that each VIOS used has the same code level.

1.7.2 Workload partitions

Workload partitions (WPARs) are software-created virtualized operating system environments within a single instance of the AIX operating system. WPARs secure and isolate the environment for the processes and signals that are used by enterprise applications.

WPARs types are application WPARs or system WPARs. System WPARs are autonomous virtual system environments with their own private file systems, users and groups, login, network space, and administrative domain.

By default, a system WPAR shares the two file systems named `/usr` and `/opt` from the global environment by using read-only namefs mounts. You can configure WPARs to have a non-shared, writable `/usr` file system and `/opt` file system. The WPARs are also called private.

For more information about IBM AIX WPARs, see *Exploiting IBM AIX Workload Partitions*, SG24-7955.

In AIX Version 7, administrators now can create WPARs that can run AIX 5.2 or AIX 5.3 in an AIX 7 operating system instance. Both are supported on the IBM POWER7® server platform and PowerHA. For details about PowerHA support, go to the following web page:

`http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10782`

Important:

- Versioned WPARs can be non-shared system WPARs only.
- The WPAR offering is supported by IBM PowerHA SystemMirror since version 5.4.1. However, particularly in the planning phase, be careful because the combination of WPARs and PowerHA in an environment can potentially introduce new single points of failure (SPOFs).
- PowerHA does not manage or monitor the WPAR. It manages and monitors only the applications that run within the WPAR.

When deploying WPAR environments, carefully consider that you must ensure maximum availability. Potentially, the following new SPOFs can be introduced into the environment:

- The network between the WPAR host and the NFS server
- The NFS server
- The WPARs
- The operating system hosting the WPARs
- The WPAR applications

The current support of WPAR in PowerHA is oriented toward the basic WPARs:

- Currently, support is available for local (`namefs` file systems) and NFS WPARs only. WPARs can be shared or private. Versioned WPARs are also supported.
- When a WPAR-enabled resource group (RG) is brought online, all its associated resources are activated within the corresponding WPAR. The WPAR-enabled RG is associated with a WPAR based on their common name. If a resource group called `wpar_rg` is WPAR-enabled, it is associated with a WPAR with the name `wpar_rg`.
- When an RG is WPAR-enabled, all user scripts, such as application start and stop scripts must be accessible within the WPAR, at the paths that are specified in the PowerHA configuration. The user is responsible for verifying that these scripts are executable and return `0`.

- A WPAR-enabled RG can consist of some nodes that are not WPAR-capable so you do not need to upgrade all nodes of the RG to the latest AIX operating system version. And when a WPAR-enabled RG comes online on a WPAR-incapable node, it behaves as though the WPAR property for the RG is not set. However, you must ensure that all user-defined scripts are accessible at the same path as previously specified in the PowerHA configuration.
- A WPAR-enabled RG supports these resources: service label, application servers, and file systems. The service address is mandatory. The service address is allocated to the WPAR when PowerHA starts the RG.
- When a WPAR-enabled RG is deleted, the corresponding WPAR on the nodes of the RG are unaffected (that is, the corresponding WPAR is not deleted).
- All the supported resource types that are supported for a WPAR-enabled RG can be DARE-added and removed from a WPAR-enabled RG. If the WPAR property of an RG is changed through DARE (when the RG is online), the effect takes place when the RG is brought online the next time.
- PowerHA configuration verification checks that all WPAR-capable nodes of a WPAR-enabled RG have a WPAR that is configured for the RG (that is, a WPAR with the same name as the RG). If the PowerHA configuration verification is run with corrective action enabled, you are prompted to fix the WPAR-related verification errors through PowerHA corrective action. It might mean the creation of a local WPAR on all nodes that are specified in the RG modification menu.
- When a WPAR-enabled RG is brought online on a WPAR-capable node, PowerHA (which runs in the global WPAR) automatically sets up **`rsh`** access to the corresponding WPAR to manage various resources that are associated with the RG.

Important: PowerHA automatically assigns and unassigns resources to and from a WPAR as the corresponding WPAR-enabled resources come online (or go offline). You must not assign any PowerHA resources to a WPAR.

Considerations

Consider the following important information:

- PowerHA Smart Assist scripts are not supported for a WPAR-enabled RG. Therefore, any application server or application monitoring script that uses the PowerHA Smart Assist scripts cannot be configured as a part of a WPAR-enabled RG.
- Process application monitoring is not supported for WPAR-enabled RGs.
- For every WPAR-capable node that is a part of a WPAR-enabled RG and contains a WPAR for a WPAR-enabled RG, at least one of the service labels (of the WPAR-enabled RG) must be accessible from the corresponding global WPAR.

Important: Only the global instance can run PowerHA. A WPAR can be considered an RG of the type WPAR-enabled RG only.

Figure 1-9 on page 23 shows a highly available WPAR environment with both resilience for the NFS server and the WPAR hosting partitions. The WPAR zion is under the control of PowerHA and shares both file systems from the local host and the NFS server.

Note: The movement of WPAR RG will checkpoint all running applications that will automatically resume from the checkpoint state on the backup node (no application start up is required, but a small period of downtime is experienced).

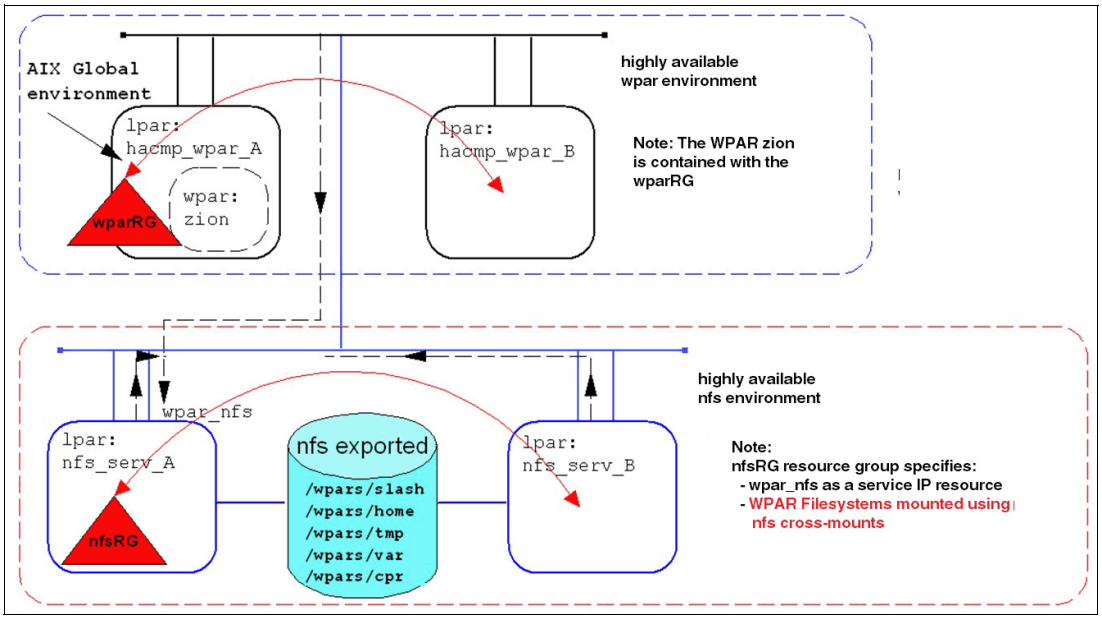

Figure 1-9 Highly available WPAR sample environment

Using this integration method makes WPAR support with PowerHA independent. For example, the same implementation steps can be done with any supported version of PowerHA.

```
wpar: zion

# lswpar -M zion

MountPoint        Device        Vfs     Nodename  Options
-------------------------------------------------------------
/wpars/zion       /wpars/slash  nfs     wpar_nfs  bg,intr
/wpars/zion/home  /wpars/home   nfs     wpar_nfs  bg,intr
/wpars/zion/tmp   /wpars/tmp    nfs     wpar_nfs  bg,intr
/wpars/zion/var   /wpars/var    nfs     wpar_nfs  bg,intr
/wpars/zion/cpr   /wpars/cpr    nfs     wpar_nfs  bg,intr
/wpars/zion/opt   /opt          namefs            ro
/wpars/zion/proc  /proc         namefs            rw
/wpars/zion/usr   /usr          namefs            ro

# netstat -i
Name  Mtu   Network     Address            ZoneID    Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      ea.48.f0.0.60.3              134297     0   196617     0     0
en0   1500  10.47       zion                         134297     0   196617     0     0
```

Figure 1-10 Example layout for WPAR: zion

More information about using WPARs with PowerHA is in *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

1.8 Summary

Spend considerable time in the planning stage. This is where the bulk of the documentation will be produced and will lay the foundation for a successful production environment. Start by building a detailed requirements document. Focus on ensuring the cluster does what the users want it to do and that the cluster behaves how you intend it to behave. Next, build a technical detailed design document. Details should include a thorough description of the storage, network, application, or cluster environment (hardware and software configuration) and the Cluster Behavior (RG policies, location dependencies, and more). Finally, make certain the cluster undergoes comprehensive and thorough testing before “going live” and further at regular intervals.

After the cluster is in production, all changes must be made in accordance with a documented Change Management procedure, and the specific changes must follow the Operational Procedures using (where possible) cluster-aware tools.

Following those steps from the initial start phase can greatly reduce the likelihood of problems and change after the cluster is put into production.

The following lists describe what to do, what not to do, and other information about PowerHA.

What to do:

- Must use IP address takeover (IPAT) through aliasing style networking and enhanced concurrent volume groups (VGs).
- Ensure that the hardware and software environment has a reasonable degree of currency. Take regular cluster snapshots and system backups.
- Configure application monitors to enhance availability and aid self-healing.
- Implement a test environment to ensure that changes are adequately tested.
- Implement a reliable heartbeat mechanism and include at least one non-IP network.
- Ensure that mechanisms are in place to send alerts via SNMP, SMS, or email when failures are encountered within the cluster.
- Implement verification and validation scripts that capture common problems (or problems that are discovered in the environment) for example, volume group settings, NFS mount and export settings, and application changes. In addition, ensure that these mechanisms are kept current.
- Make use of available PowerHA features, such as remote notification, extended cluster verification methods, “automated” cluster testing (in test only), and file collections.

What *not* to do:

- Do not introduce changes to one side of the cluster while not keeping the other nodes in sync. Always ensure that changes are synchronized immediately. If some nodes are up and others down, ensure that the change is made and synchronized from an active node.
- Do not attempt changes outside the control of PowerHA by using custom mechanisms. Where possible, use C-SPOC.
- Do not configure applications to bind in any way to node-specific attributes, such as IP addresses, host names, CPU IDs, and more. The preferred approach is to move the applications from node-to-node manually before putting them in resource groups under the control of PowerHA.

- Do not make the architecture too complex or implement a configuration that is difficult to test.
- Do not deploy basic application start and stop scripts that do not include prerequisite checking and error recovery routines. Always ensure these scripts verbosely log to stdout and stderr.
- Do not implement nested file systems that create dependencies or waits and other steps that elongate failovers.
- Do not provide root access to untrained and cluster-unaware administrators.
- Do not change the cluster failure detection rate without careful thought and consideration.
- Do not action operations such as the following example when stopping an application. This might also result in killing the PowerHA application monitor too.

  ```
  # kill `ps -ef | grep appname | awk '{print $2}'`
  ```

- Do not rely on standard AIX volume groups (VGs) if databases use raw logical volumes. Consider instead implementing big or scalable VGs. This way, user, group, and permission information can be stored in the VGDA header and reduce the likelihood of problems during failover.
- Do not rely on any form of manual effort or intervention that might be involved in keeping the applications highly available.

Consider the following additional information:

- A written cluster requirements document allows you to carry out a coherent and focused discussion with the users about what they want done. It also allows you to refer to these requirements while you design the cluster and while you develop the cluster test plan.
- A written cluster design document describes, from a technical perspective, exactly how you intend to configure the cluster environment.
- A written test plan allows you to test the cluster against the requirements (which describes what you were supposed to build) and against the cluster design document (which describes what you intended to build). Format the test plan in a way that allows you to record the pass or failure of each test, which can help you more easily know what failed, allowing you to eventually demonstrate that the cluster actually does what the users wanted it to do and what you intended it to do.
- Do not make the mistake of assuming that you have time to write the operational documentation after the cluster is in production.
- Create a cluster HTML report by using the **clmgr** command.

2

File system conversion and migration

This chapter provides information regarding file system conversion and migration between Volume Manager (VxVM) in Symantec Storage Foundation powered by Veritas to AIX Logical Volume Manager (LVM).

This chapter contains the following topics:

- Conversion versus migration
- Volume and file system migration

© Copyright IBM Corp. 2015. All rights reserved.

2.1 Conversion versus migration

First, it is important to introduce the difference between conversion and migration.

For the purpose of this book, the term *conversion* should be applied to data that can be transformed in a way that it is presented in different form. Migration, however, applies when data is copied, moved, or restored between different media.

Although there are utilities to perform conversion, migration should be the most common and reliable way to make changes in computer data.

2.2 Volume and file system migration

During the planning phase of the cluster migration, one of the most important steps is to plan for the data migration, since it may include backup and restore activities that can span throughout several hours or even days.

In the following sections, we present some information about volume group and file system data migration. We discuss briefly how Veritas Volume Manager performs LVM migration and demonstrates how to achieve the reverse results.

2.2.1 Volume group conversion and migration

In this section, we provide a short overview of the LVM to VxVM migration in order for you to understand the overall process.

LVM to VxVM migration

Symantec Storage Foundation supports either the online migration of LVM logical volumes to VxVM volumes and the LVM and JFS/JFS2 data conversion offline.

Both of the operations are supported by the Symantec Storage Foundation suite with some limitations.

Offline conversion

The offline method implies that all JFS/JFS2 file systems and LVM logical volumes are not mounted during the process. The conversion of data is performed by the **vxconvert** tool.

The conversion is accomplished by overwriting LVM and JFS/JFS2 metadata with VxVM and VxFS metadata. During the process of file system conversion ***vxconvert*** saves a copy of the original JFS/JFS2 metadata, then it scans the inode tree and builds new VXFS metadata, which is finally written over the old JFS/JFS2 data within the logical volume. During the operation, data blocks are not touched, being therefore fully preserved.

This method of conversion is quick, mature, and proven to be reliable over many years. The time for the conversion will be directly impacted by the number of file systems and entries in the structure.

Note: Before attempting to convert or migrate data, always ensure that you have current backups.

Online migration

The online migration process requires that a diskgroup and volumes structure is created in the Veritas Volume Manager in a way that it reflects the existing LVM structure.

Note: In our scenario, it was required that at least two disks were assigned to the VxVM diskgroup to perform the migration. This is because VxVM creates mirrored volumes to store data.

After creating a structure on VxVM matching the current LVM layout, the **vxmigadm** command can be used to **analyze**, **start**, **abort**, and **commit** the migration.

During the analyze process, Veritas verifies whether all requirements for the migration are met. If not, error messages will be displayed. The following list shows some errors that we encountered during our trial:

- The VxVM diskgroup name must match the LVM volume group name.
- The VxVM diskgroup must not be in CDS format.
- The VxVM volumes names must match the name and size of the LVM logical volumes.
- The permissions of volume devices and raw devices must match between VxVM and LVM devices.

Note: We recommend to perform the *analyze* operation before attempting to perform the migration to avoid undesired results. The analyze operation can be started with the following command:

```
# vxmigadm analyze -g <volume-group>
```

Upon a successful analysis of the environment, the *start* operation can be performed. The **vxmigadm** utility creates a whole new structure and starts the process. This is a very interesting part of the migration. We have reproduced the steps manually, which are shown through the next examples with explanation for each step.

In the following examples, we migrate the LVM volume group *bogusvg* containing a single logical volume *boguslv03* to the VxVM structure.

When the migration is started, **vxmigadm** performs changes to the VxVM diskgroups. Initially the volumes are stopped, putting them in the *DISABLED state.* The former LVM logical volumes and their respective entries under the directory */dev* are renamed through a call to **chlv** as shown in Example 2-1.

Example 2-1 chlv is called to rename the LVM logical volume

```
[peter101:root] / # chlv -n boguslv03_vxlv boguslv03
[peter101:root] / # lsvg -l bogusvg
bogusvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
boguslv03_vxlv      jfs2       8       8       1    closed/syncd  /will/bogusfs03
```

Next, a call to **vxddladm** is performed and a foreign disk device is added to VxVM having the LVM logical volume as its backend device, as shown in Example 2-2.

Example 2-2 vxddladm creates a foreign disk device with the LVM logical volume as the backend

```
[peter101:root] / # /usr/sbin/vxddladm addforeign blockpath=/dev/boguslv03_vxlv
charpath=/dev/rboguslv03_vxlv
```

```
[peter101:root] / # /usr/sbin/vxddladm listforeign

The Paths included are
----------------------

Based on Directory names:
-------------------------

Based on Full Path:
-------------------
/dev/boguslv03_vxlv      block  /dev/rboguslv03_vxlv     char  Suppress auto
```

Now the LVM logical volume becomes a disk device in the VxVM database. New entries with the original names are then created with new major and minor numbers associating the devices to the VxVM diskgroup. The disk initially appears as having the type set to *simple* (second column) but additional commands readds it as *nopriv* as shown in Example 2-3.

Example 2-3 vxdisk list command

```
[peter101:root] / # vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
boguslv03_vxlv nopriv          -            -           online
disk_0       auto:LVM        -            -            LVM
disk_1       auto:cdsdisk    disk         bogusvg      online
disk_2       auto:LVM        -            -            LVM
disk_3       auto:cdsdisk    disk_3       bogusvg      online
ds3400-0_0   auto:cdsdisk    -            -            online
ds3400-0_1   auto:cdsdisk    -            -            online
ds3400-0_2   auto:cdsdisk    corben_dg101  corben_dg1   online
ds3400-0_3   auto:cdsdisk    -            -            online
ds3400-0_4   auto:cdsdisk    corben_dg102  corben_dg1   online
```

New calls to **vxmake** create a new pair of subdisk and plex, which are then associated by *vxsd*. At this time, **vxmigadm** create new entries in */dev* using the original logical volume name, but binding the major and minor numbers to the VxVM diskgroup.

With the plex ready, **vxmigadm** replaces the original plex with the one associated to the LVM logical volume. Example 2-4 illustrates the removal of the plex. Notice that there are no plexes associated to the volume *boguslv03*.

Example 2-4 Removal of the plex

```
[peter101:root] / # /usr/sbin/vxplex -g bogusvg -f dis boguslv03-01
[peter101:root] / # vxprint -htv -g bogusvg
V  NAME         RVG/VSET/CO  KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE
SC NAME         PLEX         CACHE    DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
DC NAME         PARENTVOL    LOGVOL
SP NAME         SNAPVOL      DCO
EX NAME         ASSOC        VC                       PERMS    MODE     STATE

v  boguslv03    -            DISABLED CLEAN    2097152  SELECT    -        fsgen
```

Example 2-5 shows the step in which the new plex *boguslv03-lvm_plex* is attached to the diskgroup.

Example 2-5 The plex with the LVM volume as backend is attached to the diskgroup

```
[peter101:root] / # /usr/sbin/vxplex -g bogusvg att boguslv03 boguslv03-lvm_plex
[peter101:root] / # vxprint -htv -g bogusvg
V  NAME         RVG/VSET/CO  KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE
SC NAME         PLEX         CACHE    DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
DC NAME         PARENTVOL    LOGVOL
SP NAME         SNAPVOL      DCO
EX NAME         ASSOC        VC                       PERMS    MODE     STATE

v  boguslv03    -            DISABLED EMPTY    2097152  SELECT    -        fsgen
pl boguslv03-lvm_plex boguslv03 DISABLED EMPTY 2097152  CONCAT    -        RW
sd boguslv03_vxlv-01 boguslv03-lvm_plex boguslv03_vxlv 0 2097152 0 boguslv03_vxlv
ENA
```

At this point, the initial migration setup is ready and the volume is set to be *enabled*. Notice that during the entire setup, the VxVM *boguslv03* was *disabled*. Right after enabling the volume, **vxmigadm** calls **vxsnap** to create a snapshot of the volumes.

Now the file system can be made available to the application using the VxVM devices from the */dev/vx/dsk* directory. The */etc/filesystems* entries are then changed as shown in Example 2-6.

Example 2-6 /etc/filesystems changed device

```
/will/bogusfs03:
        dev             = /dev/vx/dsk/boguslv03
        vfs             = jfs2
        log             = INLINE
        mount           = false
        options         = rw
        account         = false
```

After the VxVM mirror copies achieve a consistent synchronized state with the LVM copies, two can commit the migration. Upon a commit, the old LVM volumes will be dissociated from the VxVM configuration.

As you may have noticed, during the entire volume group migration process, the file system itself was never touched remaining a JFS2 file system. The actual migration from JFS2 to VXFS cannot be performed online and requires an extended outage.

This process provides some flexibility to decide the best moment to perform the commit the migration or even to attempt an uninstallation.

While a utility is provided to automate all the steps of LVM to VxVM conversion, the opposite migration can be performed by following the uninstallation process manually, for each volume group.

Note: Detailed information about LVM to VxVM conversion is available in the Storage Foundation documentation at the Symantec website:

https://sort.symantec.com/documents/doc_details/sfha/6.0/Linux/ProductGuides

2.2.2 Limitations of migration

Although the **vxmigadm** command is convenient, it does have specific limitations that must be considered when planning for the migration.

In environments in which there are a high number of disks, it is often required to use LVM volume groups types of *Big* or *Scalable* types. At the time this documentation was written, the available version of Storage Foundation did not support the migration of Big or Scalable volume groups. Also, if advanced LVM functions like snapshots are used, the volume group cannot be migrated.

Note: For detailed information about migration limitations, refer to the Storage Foundation and high availability documentation at the Symantec website:

https://sort.symantec.com/documents/doc_details/sfha/6.0/Linux/ProductGuides

2.2.3 Migrating from VxVM to LVM

The management of VxVM volumes are not supported through the PowerHA management interface, thus if you are planning to migrate to PowerHA, it is a good idea to think about moving data from VxVM to LVM.

The conversion from VxVM to LVM is not supported the same way as the inverse. However, Symantec offers an uninstallation method that allows the administrators to migrate the logical volumes from VxVM back to LVM.

The procedure is quite simple and well explained in the Storage Foundations Installation Guide at the following site:

https://sort.symantec.com/documents/doc_details/sfha/6.0/Linux/ProductGuides

Important: Be aware that while JFS/JFS2 file systems can be converted to VxFS, the opposite direction is not supported by either IBM or Symantec. Also, always have valid backups before ever performing a migration.

Though the above mentioned lack of support statement may sway your decision on performing a storage migration, it does not impact a migration of the clustering software as PowerHA does support the VxFS file system.

Migrating a logical volume from VxVM to LVM

The migration process requires available disks to create a new LVM structure matching the VxVM structure to be migrated.

The following examples demonstrate how the migration of a volume from VxVM to LVM was possible by using the procedures described on the Symantec uninstall documentation. Example 2-7 on page 33 shows the list of available volumes in the disk group.

Example 2-7 List of available volumes in diskgroup

```
[peter101:root] / # vxprint -htv -g will_dg01
V  NAME         RVG/VSET/CO  KSTATE   STATE    LENGTH   READPOL   PREFPLEX UTYPE
PL NAME         VOLUME       KSTATE   STATE    LENGTH   LAYOUT    NCOL/WID MODE
SD NAME         PLEX         DISK     DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
SV NAME         PLEX         VOLNAME  NVOLLAYR LENGTH   [COL/]OFF AM/NM    MODE
SC NAME         PLEX         CACHE    DISKOFFS LENGTH   [COL/]OFF DEVICE   MODE
DC NAME         PARENTVOL    LOGVOL
SP NAME         SNAPVOL      DCO
EX NAME         ASSOC        VC                         PERMS    MODE     STATE

v  will_lv01    -            ENABLED  ACTIVE   2097152  SELECT    -        fsgen
pl will_lv01-01 will_lv01    ENABLED  ACTIVE   2097152  CONCAT    -        RW
sd vscsi-0_0-01 will_lv01-01 vscsi-0_0 0      2097152  0         disk_1   ENA
```

The next examples illustrate the steps taken to create the LVM volume group, the logical volume, and the actual migration step with a simple **dd** from the old volume into the new volume.

First, in Example 2-8, a new volume group is created using hdisk7 and hdisk8 and 128 Mb for the physical partition (PP) size. Next, a new 2 GB logical volume is created by using 16 PPs.

Example 2-8 Migrating a logical volume from VxVM to LVM

```
[peter101:root] / # mkvg -y bogusvg -s 128 hdisk7 hdisk8
bogusvg

[peter101:root] / # mklv -t vxfs -y boguslv02 bogusvg 16
boguslv01
```

The actual migration is shown in Example 2-9 by executing a **dd** from the old VxVM volume to the new LVM logical volume. Next, an **fsck** is run against the new logical volume and the file system is then mounted.

Example 2-9 Issuing the dd command

```
[peter101:root] / # dd if=/dev/vx/dsk/will_dg01/will_lv01 of=/dev/boguslv02
bs=4024k
260+1 records in.
260+1 records out.
[peter101:root] / # fsck -V vxfs /will/bogusfs

file system is clean - log replay is not required
[peter101:root] / # mount /will/bogusfs
[peter101:root] / # df -m /will/bogusfs
Filesystem    MB blocks      Free %Used    Iused %Iused Mounted on
/dev/boguslv02   1024.00    941.85    9%        6     1% /will/bogusfs
```

Notice that the file system was mounted and has 1 Gb in size. However, the logical volume created before the migration was 2 Gb. This happens because the actual file system migrated had 1 Gb in size before the migration. This difference was intentional to illustrate that the migration can be performed even if the sizes of the old and new logical volumes do not match.

In order to adjust the sizes, the file system can be increased to match the size of the logical volume using the VxVM utility **fsadm** as shown in Example 2-10 on page 34.

Example 2-10 Adjusting the file system size

```
[peter101:root] / # fsadm -b 2G /will/bogusfs
UX:vxfs fsadm: INFO: V-3-25942: /dev/rboguslv02 size increased from 2097152
sectors to 4194304 sectors

[peter101:root] / # df -m /will/bogusfs
Filesystem    MB blocks      Free %Used    Iused %Iused Mounted on
/dev/boguslv02   2048.00   1901.62    8%        7     1% /will/bogusfs
```

Notice that after running **fsadm**, our file system was increased to 2 Gb. Though this is a simple migration example, it can be applied to more complex scenarios.

Note: The actual uninstalling is a supported procedure by Symantec, however we suggest checking with Symantec if it is supported in a migration scenario.

3

Symantec Cluster Server powered by Veritas

This chapter introduces Symantec Cluster Server, previously known as Veritas Cluster Server (VCS) for AIX on IBM Power. It is a high availability software package that is designed to reduce both planned and unplanned downtime in a business critical environment.

Note: While previous versions of VCS work with Linux on IBM Power, at the time of writing Symantec does not have a version of this product for Linux on Power. The last version to work with Linux on Power was version 5, which is also no longer supported by Symantec.

The following topics are discussed:

- Executive overview
- Components of a Symantec cluster
- Cluster resources
- Cluster configurations
- Cluster communication
- Cluster installation and setup
- Cluster administration facilities
- PowerHA and Symantec Cluster Server compared

© Copyright IBM Corp. 2015. All rights reserved.

3.1 Executive overview

Symantec Cluster Server is a clustering solution available on Oracle Solaris, HP-UX, AIX, Linux, VMWare, and Windows. It is scalable up to 64 nodes in an AIX cluster, and supports the management of multiple VCS clusters (Windows or UNIX) from a single web or Java based graphical user interface (GUI). However, individual clusters must be composed of systems running the same operating system.

Symantec Cluster Server has similar base functionality as IBM PowerHA SystemMirror for AIX (PowerHA) product, eliminating single points of failure through the provision of redundant components, automatic detection of application, adapter, network, and node failures, and managing failover to a remote server with limited outage to the end user.

The VCS GUI-based cluster management console provides a common administrative interface in a cross platform environment. There is also integration with other Symantec products, such as the Symantec Replicator Option and Symantec Cluster Server's Global Cluster Option.

3.2 Components of a Symantec cluster

A Symantec cluster is composed of nodes, external shared disks, networks, applications, and clients. Specifically, a cluster is defined as all servers with the same cluster ID connected via a set of redundant heartbeat paths:

- Nodes: Nodes in a Symantec cluster are called *cluster servers*. There can be up to 64 cluster servers in an AIX Symantec cluster. A node runs an application or multiple applications, and can be added to or removed from a cluster dynamically.
- Shared external disk devices: Symantec Cluster Server supports a number of third-party storage vendors, and works in small computer system interface (SCSI), network-attached storage (NAS), and storage area network (SAN) environments. In addition, Symantec offers a Cluster Server Storage Certification Suite (SCS) for OEM disk vendors to certify their disks for use with VCS. Contact Symantec directly for more information about SCS.
- Networks and disk channels: These channels, in VCS cluster networks, are required for both heartbeat communication to determine the status of resources in the cluster, and also for client traffic. VCS uses its own protocol, Low Latency Transport (LLT), for cluster heartbeat communication. A second protocol, Group Membership Services/Atomic Broadcast (GAB), is used for communicating cluster configuration and state information between servers in the cluster. The LLT and GAB protocols are used instead of a TCP/IP based communication mechanism. VCS requires a minimum of two dedicated private heartbeat connections, or high-priority network links, for cluster communication. To enable active takeover of resources, should one of these heartbeat paths fail, a third dedicated heartbeat connection is required.

Client traffic is sent and received over public networks. This public network can also be defined as a low-priority network, so should there be a failure of the dedicated high-priority networks, heartbeats can be sent at a slower rate over this secondary network. A further means of supporting heartbeat traffic is disk fencing via vxfen. The disks that act as coordination points are called *coordinator disks*. Coordinator disks are three standard disks or LUNs set aside for I/O fencing during cluster reconfiguration. Coordinator disks do not serve any other storage purpose in the VCS configuration. You can configure coordinator disks to use Symantec Dynamic Multi-Pathing. Dynamic Multi-pathing (DMP) allows coordinator disks to take advantage of the path failover and the dynamic adding and removal capabilities of DMP. So, you can configure I/O fencing to use either DMP devices or the

underlying raw character devices. I/O fencing uses SCSI-3 disk policy that is either raw or dmp based on the disk device that you use. The disk policy is dmp by default. Previous versions of VCS used GABdisk, which is no longer supported from version 5.1.

Ethernet is the only supported IP network type for VCS.

3.3 Cluster resources

Resources to be made highly available include network adapters, shared storage, IP addresses, applications, and processes. Resources have a type associated with them and you can have multiple instances of a resource type. Control of each resource type involves bringing the resource online, taking it offline, and monitoring its health.

- Agents: For each resource type, VCS has a cluster agent that controls the resource. Types of VCS agents include:
  - *Bundled agents* are standard agents that come bundled with the VCS software for basic resource types, such as disk, IP, and mount. Examples of actual agents are *Application*, *IP*, *DiskGroup*, and *Mount*. For more information, see the *Symantec Bundled Agents Reference Guide*.
  - *Enterprise agents* are for applications, and are purchased separately from VCS. Enterprise agents exist for products such as DB2, Oracle, and Symantec Netbackup.
  - *Storage agents* also exist to provide access and control over storage components, such as the Symantec ServPoint (NAS) appliance.
  - *Custom agents* can be created using the Symantec developer agent for additional resource types, including applications for which there is no enterprise agent. See the *Symantec Cluster Server Agents Developers Guide* for information about creating new cluster agents.

  Symantec cluster agents are multithreaded, so they support the monitoring of multiple instances of a resource type.
- Resource categories: A resource also has a category associated with it that determines how VCS handles the resource. Resources categories include:

  On-Off VCS starts and stops the resource as required (most resources are On-Off).

  On-Only Brought online by VCS, but is not stopped when the related service group is taken offline. An example of this kind of resource would be starting a daemon.

  Persistent VCS cannot take the resource online or offline, but needs to use it, so it monitors its availability. An example would be the network card that an IP address is configured upon.
- Service group: A set of resources that are logically grouped to provide a service. Individual resource dependencies must be explicitly defined when the service group is created to determine the order resources are brought online and taken offline. When Symantec cluster server is started, the cluster server engine examines resource dependencies and starts all the required agents. A cluster server can support multiple service groups.

  Operations are performed on resources and also on service groups. All resources that comprise a service group will move if any resource in the service group needs to move in response to a failure. However, where there are multiple service groups running on a cluster server, only the affected service group is moved.

The service group type defines takeover relationships, which are either:

– Failover: The service group runs only one cluster server at a time and supports failover of resources between cluster server nodes. Failover can be both unplanned (unexpected resource outage) and planned, for example, for maintenance purposes. Although the nodes, which can take over a service group, will be defined, there are three methods by which the destination failover node is decided:

  • Priority: The *SystemList* attribute is used to set the priority for a cluster server. The server with the lowest defined priority that is in the running state becomes the target system. Priority is determined by the order the servers are defined in the SystemList with the first server in the list being the lowest priority server. This is the default method of determining the target node at failover, although priority can also be set explicitly.

  • Round: The system running the smallest number of service groups becomes the target.

  • Load: The cluster server with the most available capacity becomes the target node. To determine available capacity, each service group is assigned a capacity. This value is used in the calculation to determine the failover node, which is based on the service groups active on the node.

– Parallel: Service groups are active on all cluster nodes that run resources simultaneously. Applications must be able to run on multiple servers simultaneously with no data corruption. This type of service group is sometimes also described as *concurrent*. A parallel resource group is used for things like web hosting.

The web VCS interface is typically defined as a service group and kept highly available. It should be noted, however, that although actions can be initiated from the browser, it is not possible to add or remove elements from the configuration via the browser. The Java VCS console should be used for making configuration changes.

In addition, service group dependencies can be defined. Service group dependencies apply when a resource is brought online, when a resource faults, and when the service group is taken offline. Service group dependencies are defined in terms of a parent and child, and a service group can be both a child and parent. Service group dependencies are defined by three parameters:

– Category

– Location

– Type

Values for these parameters are:

– online/offline

– local/global/remote

– soft/hard

As an example, take two service groups with a dependency of online, remote, and soft. The category online means that the parent service group must wait for the child service group to be brought on online before it is started. Use of the remote location parameter requires that the parent and child must necessarily be on different servers. Finally, the type soft has implications for service group behavior should a resource fault. See the *Symantec Cluster Server User Guide* for detailed descriptions of each option. Configuring service group dependencies adds complexity, so must be carefully planned.

► Attributes: All VCS components have attributes associated with them that are used to define their configuration. Each attribute has a *data type* and *dimension*. Definitions for data types and dimensions are detailed in the *Symantec Cluster Server User Guide*. An example of a resource attribute is the IP address associated with a network interface card.

- System zones: VCS supports system zones, which are a subset of systems for a service group to use at initial failover. The service group chooses a host within its system zone before choosing any other host.

3.4 Cluster configurations

The Symantec terminology used to describe supported cluster configurations are:

Asymmetric There is a defined primary and a dedicated backup server. Only the primary server is running a production workload.

Symmetric There is a two node cluster where each cluster server is configured to provide a highly available service and acts as a backup to the other.

N-to-1 There are N production cluster servers and a single backup server. This setup relies on the concept that failure of multiple servers at any one time is relatively unlikely. In addition, the number of slots in a server limits the total number of nodes capable of being connected in this cluster configuration.

N+1 An extra cluster server is included as a spare. Should any of the N production servers fail, its service groups move to the spare cluster server. When the failed server is recovered, it simply joins as a spare so there is no further interruption to service to failback the service group.

N-to-N There are multiple service groups running on multiple servers, which can be failed to potentially different servers.

3.5 Cluster communication

Cross cluster communication is required to achieve automated failure detection and recovery in a high availability environment. Essentially all cluster servers in a Symantec cluster must run the following:

High availability daemon (HAD)

This is the primary process and is sometimes referred to as the *cluster server engine*. A further process, *hashadow*, monitors HAD and can restart it if required. VCS agents monitor the state of resources and pass information to their local HAD. The HAD then communicates information about cluster status to the other HAD processes using the GAB and LLT protocols.

Group membership services/atomic broadcast (GAB)

GAB operates in the kernel space, monitors cluster membership, tracks cluster status (resources and service groups), and distributes this information among cluster nodes using the low latency transport layer.

Low latency transport (LLT)

LLT operates in kernel space, supporting communication between servers in a cluster, and handles heartbeat communication. LLT runs directly on top of the DLPI layer in UNIX. LLT load balances cluster communication over the private network links.

A critical question related to cluster communication is, “What happens when communication is lost between cluster servers?” VCS uses heartbeats to determine the health of its peers and requires a minimum of two heartbeat paths, either private, public, or disk based. With only a single heartbeat path, VCS is unable to determine the difference between a network failure and a system failure. The process of handling loss of communication on a single network as opposed to a multiple network is called *jeopardy*. So, if there is a failure on all communication channels, the action taken depends on what channels have been lost and the state of the channels before the failure. Essentially, VCS will take action such that only one node has a service group at any one time; in some instances, disabling failover to avoid possible corruption of data. A full discussion is included in “Network partitions and split-brain” in Chapter 22, “Troubleshooting and Recovery”, in the *Symantec Cluster Server 6.1 Administrator’s Guide - Linux* (http://tinyurl.com/kb7pxrw).

3.6 Cluster installation and setup

Installation of VCS on AIX can be done via **`installp`** or SMIT. It should be noted, however, that if **`installp`** is used, LLT, GAB, and the `main.cf` file must be configured manually. Alternatively, we recommend that the */installer* script bundled with the Symantec packages should be used to handle the installation of the required software and initial cluster configuration.

Note: All installations of the cluster during the creation of this IBM Redbooks publication were done via the */installer* script bundled with the required Symantec package.

After the VCS software has been installed, configuration is typically done via the VCS Java GUI interface. The first step is to carry out careful planning of the wanted high availability environment. There are no specific tools in VCS to help with this process. When this has been done, service groups are created and resources are added to them, including resource dependencies. Resources are chosen from the bundled agents and enterprise agents, or if there are no existing agents for a particular resource, a custom agent can be built. After the service groups have been defined, the cluster definition is automatically synchronized to all cluster servers.

Under VCS, the cluster configuration is stored in ASCII files. The two main files are the *main.cf* and *types.cf*:

- main.cf: Defines the entire cluster
- types.cf: Defines the resources

These files are user readable and can be edited in a text editor. A new cluster can be created based on these files as templates.

3.7 Cluster administration facilities

Administration in a Symantec cluster is generally carried out via the cluster manager Java GUI interface. The cluster manager provides a graphical view of cluster status for resources, service groups, and heartbeat communication among others:

- Security: A VCS administrator can have one of five user categories. These include *Cluster Administrator*, *Cluster Operator*, *Group Administrator*, *Group Operator*, and *Cluster Guest*. Functions within these categories overlap. The Cluster Administrator has full

privileges and the ClusterGuest has read only function. User categories are set implicitly for the cluster by default, but can also be set explicitly for individual service groups.

- Logging: VCS generates both error messages and log entries for activity in the cluster from both the cluster engine and each of the agents. Log files related to the cluster engine can be found in the `/var/VRTSsvc/log` directory, and agent log files in the `$VCS_HOME/log` directory. Each VCS message has a tag, which is used to indicate the type of the message. Tags are of the form TAG_A-E, where TAG_A is an error message and TAG_D indicates that an action has occurred in the VCS cluster. Log files are ASCII text and user readable. However, the cluster management interface is typically used to view logs.
- Monitoring and diagnostic tools: VCS can monitor both system events and applications. Event triggers allow the system administrator to define actions to be performed when a service group or resource hits a particular trigger. Triggers can also be used to carry out an action before the service group comes online or goes offline. The action is typically a script, which can be edited by the user. The event triggers themselves are predefined. Some can be enabled by administrators, where others are enabled by default. In addition, VCS provides simple network management protocol (SNMP), management interface base (MIB), and simple mail transfer protocol (SMTP) notification. The severity level of a notification is configurable. Event notification is implemented in VCS using *triggers*.
- Emulation tools: The VCS Java Cluster Manager GUI offers a feature called the HA Fire Drill. This feature runs checks against resources fixing and checking for specific errors. These checks verify the resources defined in the VCS configuration file (`main.cf`) have the required infrastructure to fail over on another node. This could involve checking for existence of mount directories and more. These checks can only be done when the service group is online, and it verifies that the specified node is a viable failover target capable of hosting that service group. More information can be found about this and how to do the HA Fire Drill in the *Symantec Cluster Server Administrator's Guide.*

3.8 PowerHA and Symantec Cluster Server compared

The following section describes PowerHA and highlights where terminology and operation differ between PowerHA and Symantec Cluster Server (VCS). PowerHA and VCS have fairly comparable function, but differ in some areas. VCS has support for cross-platform management, is integrated with other Symantec products, and uses a GUI interface as its primary management interface. PowerHA is optimized for AIX and IBM POWER servers, and is tightly integrated with the AIX operating system. PowerHA can readily utilize availability functions in the operating system to extend its capabilities to monitoring and managing of non-cluster events.

3.8.1 Components of a PowerHA cluster

A PowerHA cluster is similarly comprised nodes, external shared disks, networks, applications, and clients:

Nodes Nodes in a PowerHA cluster are called cluster nodes, compared with VCS cluster server. There can be up to 16 nodes in a PowerHA/ES cluster, including in a concurrent access configuration. A node will run an application or multiple applications, and can be added to or removed from a cluster dynamically.

Shared disks PowerHA has built-in support for a wide variety of disk attachments, including Fibre Channel and several varieties of SCSI. PowerHA provides an interface for OEM disk vendors to provide additional attachments for NAS, SAN, and other disks.

Networks IP networks in a PowerHA cluster are used for both heartbeat/message communication to determine the status of the resources in the cluster, and also for client traffic. PowerHA uses an optimized heartbeat protocol over IP. Supported IP networks, up to PowerHA 6.1, include Ethernet, FDDI, token-ring, SP-Switch, and ATM. PowerHA 7.1 and above only supports Ethernet IP networks. Non-IP networks are also supported to prevent the Internet Protocol network from becoming a single point of failure in a cluster.

Networks based on SNA are also supported as cluster resources. Cluster configuration information is propagated over the public Internet Protocol networks in a PowerHA cluster. However, heartbeats and messages, including cluster status information, is communicated over all PowerHA networks.

3.8.2 Cluster resources

Resources to be made highly available include network adapters, shared storage, IP addresses, applications, and processes. Resources have a type, and you can have multiple instances of a resource type.

- PowerHA event scripts: Both PowerHA and VCS support built-in processing of common cluster events. PowerHA provides a set of predefined event scripts that handle bringing resources online, taking them offline, and moving them if required. VCS uses bundled agents. PowerHA provides an event customization process and VCS provides a means to develop agents.

  Application server This is the PowerHA term used to describe how applications are controlled in a PowerHA environment. Each application server is composed of a start and stop script, which can be customized on a per node basis. Sample start and stop scripts are available for download for common applications at no cost.

  Application monitor Both PowerHA and VCS have support for application monitoring, providing for retry/restart recovery, relocation of the application, and for different processing requirements, based on the node where the application is being run.

  The function of an application server coupled with an application monitor is similar to a VCS enterprise agent.

- Resource group: This is equivalent to a VCS service group, and is the term used to define a set of resources that comprise a service. The resource group defines startup, fallover and failback behavior. The startup options include:

  Online On Home Node

  A list of participating nodes is defined for a resource group, with the order of nodes indicating the node priority for the resource group. Resources are owned by the highest priority node available. If there is a failure, the next active node with the highest priority will take over.

  Online On First Available Node

  A list of participating nodes is defined for a resource group. However, the resource group will come online to the node that activates first in the cluster. This could result in multiple resource groups starting up on only one node.

Online Using Node Distribution Policy

Similar to the previous option but it will activate only one resource group at a time during node startup. If there are more resource groups than nodes, it will spread the resource groups across all the nodes. This prevents all resource groups from starting on just one node.

Online On All Available Nodes

Resource group activates on all nodes as each node joins the cluster. Typically, this also infers concurrent shared data access across the nodes. An example of an application that uses this option Oracle Real Application Cluster. This option only supports the use of raw logical volumes or GPFS. Other options that usually do not require shared data may include some application servers.

The fallover options include:

Fallover To Next Priority Node In The List

The resource group that is online on only one node at a time follows the default node priority order specified in the resource group's nodelist. It will move to the highest one available.

Fallover Using Dynamic Node Priority (DNP)

It is also possible to set a dynamic node priority (DNP) policy, which can be used at failover time to determine the best takeover node. Each potential takeover node is queried regarding the DNP policy, which might be something like most free CPU. This option is only utilized if there are more than two nodes in a cluster. There are both predefined and custom user-defined options available in PowerHA.

The fallback options include:

Never fallback

A resource group will not automatically fallback to the highest priority node when it rejoins the cluster.

Fallback To Higher Priority Node In The List

A resource group will automatically fallback to the highest priority node when it rejoins the cluster. This will incur a small outage time. This option can also be utilized with a feature called *fallback timer*. This allows one to specify a date and time for the resource group to move back.

By default, resource groups are brought online in parallel to minimize the total time required to bring resources online. It is possible, however, to define a temporal order if resource groups need to be brought online sequentially. Also, it is possible to define resource group dependencies to achieve the wanted results.

3.8.3 Cluster configurations

PowerHA and VCS are reasonably comparable in terms of supported cluster configurations, although the terminology differs. PowerHA cluster configurations include:

Standby configurations

Support a traditional hardware configuration where there is redundant equipment available as a hot standby. Though historically this implies a one-to-one, it could also be many-to-one.

Mutual Takeover configurations

All cluster nodes do useful work and act as a backup to each other.

Concurrent All cluster nodes are active and have simultaneous access to the same shared resources.

3.8.4 Cluster communications

Cross cluster communication is a part of all high availability software, and in PowerHA this task is carried out by the following components:

- Cluster manager daemon (clstrmgrES): This can be considered similar to the VCS cluster engine and must be running on all active nodes in a PowerHA cluster. In the classic feature of PowerHA, the clstrmgrES is responsible for monitoring nodes and networks for possible failure, and keeping track of the cluster peers. Beginning with PowerHA v7 some of the functions carried out by RSCT, specifically topsvcs, moved to Cluster Aware AIX (CAA).3.8.3, “Cluster configurations” on page 43. The clstrmgr executes scripts in response to changes in the cluster (events) to maintain availability in the clustered environment.
- Cluster communications daemon (clcomd): This provides cluster-based communications.
- Reliable Scalable Cluster Technology (RSCT): This is used extensively in PowerHA for messaging, monitoring cluster status, and event monitoring. RSCT is part of the AIX base operating system and is composed of:
  - Group services: Coordinates distributed messaging and synchronization tasks.
  - Event management: Monitors system resources and generates events when resource status changes.

PowerHA and VCS both have a defined method to determine whether a remote system is alive, and a defined response to the situation where communication has been lost between all cluster nodes. These methods essentially achieve the same result, which is to avoid multiple nodes trying to grab the same resources.

3.8.5 Cluster installation and setup

Installation of PowerHA for AIX software is via the standard AIX installation process using **`installp`**, from the command line or via SMIT. Installation of PowerHA automatically updates a number of AIX files, such as `/etc/services` and `/etc/inittab`. No further system-related configuration is required following the installation of the PowerHA software.

The main SMIT PowerHA configuration menu (fast path **`smitty sysmirror`**) outlines the steps that are required to configure a cluster. The cluster topology is defined first and synchronized via the network to all nodes in the cluster and then the resource groups are set up. Resource groups can be created on a single PowerHA node and the definitions propagated to all other nodes in the cluster. The resources, which comprise the resource group, have implicit dependencies that are captured in the PowerHA software logic.

PowerHA configuration information is held in the object data manager (ODM) database, providing a secure but easily shareable means of managing the configuration. A *cluster snapshot* function is also available, which captures the current cluster configuration in two ASCII user readable files. The output from the snapshot can then be used to clone an existing PowerHA cluster or to reapply an earlier configuration. In addition, the snapshot can be easily modified to capture additional user-defined configuration information as part of the PowerHA snapshot. VCS does not have a snapshot function *per se*, but allows for the current

configuration to be dumped to file. The resulting VCS configuration files can be used to clone cluster configurations. There is no VCS equivalent to applying a cluster snapshot.

3.8.6 Cluster administration facilities

Cluster management is typically via the System Management Interface Tool (SMIT). The PowerHA menus are tightly integrated with SMIT and are easy to use. There is also close integration with the AIX operating system:

- Security: PowerHA employs AIX user management to control access to cluster management function. By default, the user must have root privilege to make any changes. AIX *roles* can be defined if wanted to provide a more granular level of user control. Achieving high availability requires good change management, and this includes restricting access to users who can modify the configuration.
- Logging: PowerHA log files are simple ASCII text files. There are separate logs for messages from the cluster daemons and for cluster events. The primary log file for cluster events is the `PowerHA.out` file, which is by default in `/tmp`. The system administrator can define a non-default directory for individual PowerHA log files. The contents of the log files can be viewed via SMIT or a web browser. In addition, RSCT logs are also maintained for PowerHA/ES.
- Monitoring and diagnostic tools: PowerHA has extensive event monitoring capability and it is possible to define a custom PowerHA event to run in response to the outcome of event monitoring. In addition, multiple pre-events and post-events can be scripted for all cluster events to tailor them for local conditions. PowerHA and VCS both support flexible notification methods, SNMP, SMTP, and email notification. PowerHA uses the AIX error notification facility and can be configured to react to any error reported to AIX. VCS is based on event triggers and reacts to information from agents. PowerHA also supports pager notification.
- Emulation tools: PowerHA can emulate error log entries to validate any customization to error notification. The VCS Java Cluster Manager GUI offers a feature called the HA Fire Drill. This feature runs checks against resources fixing and checking for specific errors.
- Both PowerHA and VCS provide tools to enable maintenance and change in a cluster without downtime. PowerHA has the cluster single point of control (CSPOC) and dynamic reconfiguration capability (DARE). CSPOC allows a cluster change to be made on a single node in the cluster and for the change to be applied to all nodes. Dynamic reconfiguration uses the **`cldare`** command to change configuration, status, and location of resource groups dynamically. It is possible to add nodes, remove nodes, and support rolling operating systems or other software upgrades. VCS has the same capabilities and cluster changes are automatically propagated to other cluster servers. However, PowerHA has the unique ability to emulate migrations for testing purposes.

3.8.7 PowerHA and Symantec Cluster Server feature comparison summary

Table 3-1 shows the PowerHA and Symantec Cluster Server environment support.

Table 3-1 PowerHA and Symantec Cluster Server environment support

Environment	PowerHA	VCS for AIX
Operating system	AIX 6.1.6 and above AIX 7.1.0 and above	AIX 6.1.6 and above AIX 7.1.0 and above
Network connectivity	Ethernet	Ethernet
Disk connectivity	iSCSI, Fibre Channel	iSCSI, Fibre Channel

Environment	PowerHA	VCS for AIX
Maximum servers in a cluster	16	64*
Concurrent disk access	Yes - Raw logical volumes only	N/A
LPAR support	Yes	Yes
Integrated DLPAR Fallover Support	Yes	No, can be customized via scripts
Storage subsystems	All supported by VIOS	All supported by VIOS

* VCS is capable of supporting clusters with up to 64 nodes. Symantec has tested and qualified configurations of up to 32 nodes at the time of the 6.1 release.

Table 3-2 shows the PowerHA and Veritas disaster recovery support.

Table 3-2 PowerHA and Veritas disaster recovery support

Replication option supported	PowerHA Enterprise Edition	VCS for AIX
GLVM	Yes	No
XIV Sync/Async	Yes	Yes
DS8000 Metro/Global	Yes	Yes
SVC Metro/Global	Yes	Yes
IBM Storwize® v7000 Metro/Global	Yes	Yes
EMC SRDF Sync/Async	Yes	Yes
EMC SRDF/Star	No	Yes
EMC RecoverPoint	No	No
Hitachi True Copy/Universal Replicator	Yes	Yes
HP Continuous Access	Yes	Yes
HP 3PAR Remote Copy	No	Yes
Netapp SnapMirror	No	Yes

4

Migrating from Symantec Cluster Server powered by Veritas

This chapter presents a migration scenario followed by a cluster administration scenario.

The following topics are discussed in this chapter:

- Terminology
- Introduction
- Daily administration
- Cluster environment
- Planning
- Converting a Symantec Cluster Server to an IBM PowerHA cluster
- Test environment overview
- Creating the LVM volume group and the file systems
- Installing the PowerHA software
- Performing the migration
- Roll-back and removal
- Deleting the Symantec Cluster Server
- Troubleshooting, and known issues
- Shared storage pools
- Cluster migration

© Copyright IBM Corp. 2015. All rights reserved.

4.1 Terminology

The following subsections contain terminology and basic concepts to establish a link between the Symantec Cluster Server (VCS) and PowerHA for those who are not familiar with VCS terminology.

4.1.1 Cluster communication

Starting in IBM PowerHA 7.1, the Cluster Aware AIX (CAA) has been one of the key requirements to deploy a new cluster. CAA is an AIX operating system component that established common cluster capabilities for applications such as RSCT and PowerHA. PowerHA uses CAA for base cluster communication of events, heartbeating, and cluster-wide configuration management.

With CAA, the cluster communication for events and configuration is no longer limited to network availability. The use of shared disks across all nodes enhances failure detection and event notification among the nodes.

Note: CAA has been available since AIX 6.1 TL6 and 7.1 and is well described in *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030, published in October 2012.

Unlike PowerHA, Symantec Cluster Server (VCS) is restricted to network communication over Ethernet interfaces for most of its cluster management and event notification, which is indeed a way to keep a common communication layer for multiple platforms.

Although this communication method may appear antiquated when compared to the CAA capabilities, it has components to make it reliable. First, VCS requires redundant private network for cluster-wide and cluster-to-nodes communication. Further, the Group Membership Services and Atomic Broadcasting (GAB) and Low Latency Transport (LLT) features provide membership control and high-performance communication across the private network to ensure resiliency.

LLT is a proprietary replacement for the IP stack, which provides low-latency high-performance communication. LLT features traffic distribution of communication packets across the available channels and is responsible for the cluster heartbeat. GAB stays on top of LLT using its communication features to control the node membership across the cluster.

Note: Detailed information about Symantec Cluster Server communication can be found in the documentation section of the *Symantec Operations Readiness Tools (SORT)* website:

`http://sort.symantec.com`

4.1.2 Seeding

Seeding is a protection feature of GAB to verify that all nodes configured in the cluster are available and joined the cluster properly. A node is required to be seeded before it has VCS services activated and be enabled to act as a member node.

4.1.3 Coordination points

In a Symantec Cluster Server, the coordination points are components that provide additional references to the cluster nodes when a problem occurs. The coordination points play an important role during fencing; however, although it is desired that a cluster has coordination points they are actually not required.

Coordination points can be deployed either with disks (*Coordination Disks*) or with servers (*Coordination Point Servers* or *CP Servers*). In the first method, VCS requires at least three disks in an exclusive *diskgroup*, for fencing. For a *CP Server*, Symantec suggests an exclusive server with VCS installed.

Note: Since Coordination Point Servers require an installation of the Symantec Cluster Server, additional licensing costs may imply in this configuration.

The role of the *coordination points* in VCS is similar to what is achieved when *netmon.cf* is configured in PowerHA.

4.1.4 Fencing

Fencing isolates a failed node from the cluster preventing it from acquiring resources and causing problems.

In PowerHA, fencing capabilities are provided by the CAA layer which has embedded network and disk communication capabilities.

The Symantec Cluster Server provides fencing in two different ways: disk-based and server-based. Disk-based and server-based fencing are achieved by using coordination points, which can be *Coordinator Disks* or *Coordination Point Servers (CP Servers),* respectively.

4.1.5 Resources and groups

In Symantec Cluster Server (VCS), the resources are configured through the **`hares`** command. There are different classes of resources known as *resource types,* each having a different set of attributes.

By default, *resources* in VCS have to be *enabled* to work in the cluster. In the event of problems after the configuration of a resource, it may be necessary to perform a *probe* operation. Probing a resource will test it for errors and activate it or point errors. The command **`hares`** can be used to probe the resources.

Once the *resources* are configured, they must be bound to a *group* with the ***hagrp*** command. Groups are collection of resources to be managed across the nodes of the cluster. Each group in VCS has a set of global attributes and another set of attributes tied to each node, which can indicate the status of that *group* in each of the nodes.

Resources dependency

In VCS, the dependency relationship between the resources is defined as *parent* and *child* in a way that the parent resource depends on the child resource. Child resources are started before parent resources.

Let us pick a *DiskGroup* and *Mount* resources called *res_dg01* and *res_mnt01* as an example. From a logical volume perspective, the diskgroup can be seen as the *parent* resource while the mount point would be a *child* resource. In this relationship, the child depends on the parent.

In VCS however, from a resource dependency perspective, that relationship is seen in the opposite way. The *parent* depends on the *child* resources. Therefore, *res_mnt01* is considered the parent resource and depends on *res_dg01,* the child resource, to be made available first.

4.1.6 Storage foundation

The volume manager in Storage Foundation, formerly known as Veritas Volume Manager (VxVM) is basically built of physical and virtual objects.

Physical objects are mainly represented by the disks available to the operating system. Virtual objects are represented by VxVM disks, subdisks, plexes, volumes, and diskgroups, which we briefly describe below.

VxVM disks

In order to use a physical object or a disk in Storage Foundation, the disks must be captured or encapsulated as an VxVM Disk, which is virtual object. These objects now can be used to compose VxVM diskgroups and volumes.

In most cases VxVM disks will be a representation of an operating system disk, but Storage Foundation allows for other types of storage entities to be encapsulated as an VxVM Disk. For example, an LVM logical volume (LV) can be encapsulated as a VxVM Disk to be used in VxVM Volumes. This scenario can be seen in the online migration of LVM to VxVM.

Diskgroups

Diskgroups in Storage Foundation are composed of *VxVM Disks* and have a number of configurations and attributes that describe the configuration of its volumes, plexes, and subdisks. For comparison, *diskgroups* are similar to *volume groups* in LVM.

Volumes

The concept of VxVM *volumes* is similar to the logical volumes in LVM. However, while in LVM, LVs are composed of a set of physical partitions, VxVM Volumes are built of *plexes,* which are an entity that does not exist in LVM.

Plexes

Plexes are an additional layer between the *volumes* and *subdisks* in which the layout of the *diskgroup* is defined. Plexes can be arranged in different ways (stripped, mirrored, concatenated, and so on) and also provide a software layer for RAID layouts.

The *plex* layer plays an important role in the VxVM structure by adding flexibility to the product. Plexes allows layout changes. For example, a *VxVM Volume* initially created with a RAID 0+1 layout can be changed to RAID 1+0 without difficulties.

LVM does not have any kind of entity that corresponds to a *VxVM Plex*.

Subdisks

Subdisks are logical disk devices bound to a VxVM Disk. In most cases, a *subdisk* will be bound to a disk. There are specific situations where the subdisks may have different devices

as their back-end devices. On “LVM to VxVM migration” on page 28, we describe the online migration from LVM to VxVM in which a logical volume is configured as a VxVM Disk and attached to a subdisk to configure a VxVM volume.

4.2 Introduction

In the following sections, we illustrate a process to migrate resources running under Symantec Cluster Server to PowerHA.

We have chosen to perform an in-place cluster migration on the same nodes running the current Symantec Cluster Server. We have decided to perform this type of migration to understand and demonstrate how much of the process can be accomplished without required downtime.

The current state of the virtualization technologies makes the creation of additional resources fairly easy. However, there are a number of other factors that will influence decisions when preparing a cluster migration.

Strict service management policies may represent a constraint on schedules when they define processes to add and retire servers in the IT environment. In such cases, utilizing available tools for migration can save a considerable number of labor hours.

Many different factors affect the time required to perform a full cluster migration. A significant one is if a large data migration is also required. We have split the migration procedure into different processes, so that the migration can be performed in different phases or maintenance windows:

- **Cluster configuration migration only**

  This step is to migrate only the cluster configuration from Symantec Cluster Server to PowerHA, preserving the Volume Manager storage resources (diskgroups and file systems).

  User-defined resources will be used in PowerHA to manage the non-LVM diskgroups and file systems. This generally would not be required because PowerHA does have integrated support for VxVM. However, at the time of writing it did not support the latest levels (6.1) we used during for our migration.

  Note: The configuration step itself can be performed with the Symantec Cluster Server still active, without any downtime. However, the cluster must be fully stopped before any resource groups are started in PowerHA.

- **Cluster configuration plus VxVM to LVM migration**

  On this step, we migrate the volume group data, migrating data from Volume Manager volumes to new LVM logical volumes.

  During this process, the file systems type and data are fully preserved. This means that at the end of the process, LVM will contain logical volumes in the *vxfs* format.

  At this time, we are able to allow PowerHA to manage the LVM volume groups. The *vxfs* file systems will still be managed as user-defined resources.

- **Cluster configuration and file system data migration**

  This is the final step, where all data is migrated from one file system to another either by copy, or backup and restore processes.

Note: Unless data can be migrated by using snapshots, all applications are required to be down and file systems should not be updated during the copy of data.

4.3 Daily administration

Administration of both products are similar and require the same level of attention and care.

Administration of PowerHA can be performed through **clmgr** command-line interface (CLI), SMIT, or IBM Systems Director PowerHA plug-in. All task should be available through all three methods.

Symantec Cluster Server mostly utilizes a graphical interface that is freely available. Further VCS can be also managed by the Veritas Operation Manager (VOM) if available on the environment. Also administration can be performed through a command-line interface.

For PowerHA users preparing for a migration, it will be fairly easy to use VCS commands. A number of the commands used in VCS are illustrated throughout this chapter.

4.4 Cluster environment

Both the Symantec and IBM products offer a wide range of implementation options. Since our tests are intended to be a proof of concept, we have decided to keep the cluster as simple as possible.

The chosen configuration for the migration is a Symantec Cluster Server managing one Volume Manager disk group and one service IP address within the same service group. The cluster infrastructure is made of two private networks for the heartbeating and three VFC-attached disks for fencing.

This book will not cover the steps of creating a cluster since the procedure is well explained in other publications. Instead, we cover specific steps that are specific to this migration scenario.

Note: For detailed information about creating PowerHA clusters, refer to *PowerHA SystemMirror for AIX Cookbook Update, SG24-7739-01*.

4.5 Planning

Before starting the cluster migration, ensure that all resources required for the migration are available within your environment.

Most of the PowerHA cluster setup can be performed offline. This allows you to configure the cluster in advance and ultimately reduce the required outage to perform the transition from one product to another.

The migration of data is probably the most important and sensitive part of the transition. Especially when there is not a tool to perform the migration from VxFS back to JFS/JFS2.

Depending on the type of data to be migrated, different approaches may apply. For example, file systems, which hold mostly static files, can be replicated in advance while still online by

using snapshots or tools like rsync and just resynchronized at the time of the transition. Database files, however, which are kept open for updates most of the time, should be migrated with the applications down. Raw devices can be migrated from VxVM to LVM, but also require that applications are taken down before the migration.

As mentioned in 4.2, “Introduction” on page 51, Symantec diskgroups and file systems can be managed by PowerHA by using user-defined resources. Although customized scripts are required to manage such resources, they play an important role during the migration of the cluster.

4.5.1 Network considerations

The existing network infrastructure used with Symantec Cluster Server can be used by PowerHA if an in-place migration is being performed. A similar network infrastructure will be required if PowerHA is being configured in different boxes.

In complex networks, obtaining new IP addresses for an entire new setup may not be wanted. This makes utilizing the existing one an easier choice.

However, if the new cluster is being deployed in a new infrastructure, make sure that all VLANs are properly configured and that any firewall rules required for the applications are properly configured.

4.5.2 Storage considerations

The PowerHA setup requires at least one additional disk drive shared between the nodes for the Cluster Aware AIX (CAA) repository disk and volume group.

For our test, we created a new 1 Gb Shared Storage Pool LUN and mapped to both nodes of the cluster.

The storage requirements for the migration will depend on the plans for data migration. While migrating only the cluster configuration does not add many storage requirements, copying or restoring data to fresh volume groups and file systems may require duplication of the current available disks.

Note: At the time of writing, the HMC interface to manage Shared Storage Pool resources does not allow mapping a LUN to multiple nodes. The mapping must be performed through the CLI in the VIOS.

4.5.3 Application resources

Application resources are used to manage the application availability on the cluster. They usually perform tasks like start and stop of applications, but can also perform other tasks like application monitoring.

In Symantec Cluster Server, a resource of the *Application* type is used to start, stop, and monitor applications. In PowerHA, the start and stop tasks are configured in *Application Resources* and the monitors are configured in the *Application Monitors Resources.*

Theoretically, the scripts used to manage application resources in the Symantec cluster can be migrated to PowerHA if they use the same convention for return codes and are not bound to the Storage Foundation suite.

4.6 Converting a Symantec Cluster Server to an IBM PowerHA cluster

In this section, we cover how to convert Symantec Cluster Server (VCS) to a PowerHA cluster. Our intention is to document it so that anyone who has not worked with VCS before will be able to perform these steps. It is *always* recommended to perform these steps in a test or pre-production environment before ever performing in production. Also, valid backups should also exist before beginning.

In our example, we make a copy of the Symantec Storage Foundation disk pool on LVM, so that we can take a copy of the data on VxVM and then migrate the system to PowerHA. This LVM disk will be first controlled by VCS until all data copies and migrations are completed or until you are ready to swap clusters. At which point, VCS can then be stopped and PowerHA started, taking ownership of the LVM disk and the related applications.

Overview of the following tasks:

1. Test environment overview
2. Create LVM disk pool
    - Define linked cluster with sites
    - Configure volume groups
    - Create logical volumes
    - Create file systems
    - Add new LVM into VCS Service Groups
    - Test Failover
3. Install PowerHA
4. Add LVM into PowerHA
5. Test PowerHA
6. Migrate to PowerHA from VCS
7. Roll back and removal instructions
8. Remove VCS

4.7 Test environment overview

In our example, we have a local two-node cluster using a single IP network. Each systems' *rootvg* is mapped over via *shared storage pools (SSPs)*. The application data is utilizing NPIV-attached IBM DS3400 shared storage as shown in Figure 4-1 on page 55.

What is not shown is that our test environment utilizes dual VIO Servers within each of the IBM Power P750 machines. The boot volumes are presented up to the virtual machines (LPARs) through VIOS via SSP. There is not a technical requirement for this; however, we chose to do so because at the time of writing there was other existing documentation of such examples. The shared disk resources between the systems are mapped to each via NPIV. This is due to limitations with SCSI-3 reserves that the Volume Manager requires.

Note: Because the IBM AIX operating system does not support full SCSI-3 persistent reserve capabilities, SSP implements additional options. Details can be found in the *IBM PowerVM Enhancements What is New in 2013*, SG24-8198.

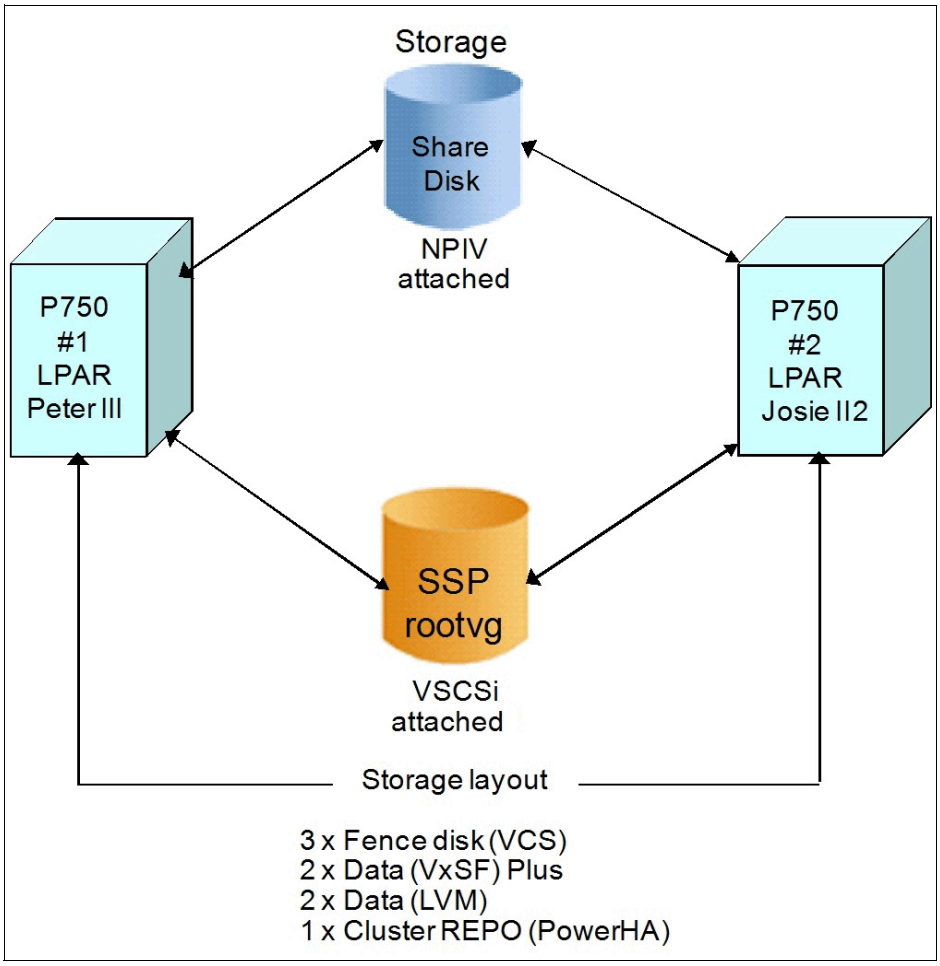

Figure 4-1 Test environment configuration

4.7.1 Environmental details

The VIOS and virtual machines software details are as follows:

- IBM AIX 7.1 TL3 SP3
- IBM PowerHA 7.1.3.1
- Symantec Cluster Server (VCS) 6.1
- Symantec Storage Foundation (VxSF) 6.1
- IBM VIO Server 2.2.3.2

4.7.2 Symantec Cluster Server configuration

This section describes the Symantec Cluster Server configuration.

Topology

The details of the main cluster topology configuration are shown in Figure 4-2 on page 56.

```
[josie112:root] / # lltstat -n
LLT node information:
    Node                 State    Links
     0 peter111          OPEN        2
   * 1 josie112          OPEN        2
[josie112:root] / # hares -display webip |grep ArgListValues
webip        ArgListValues         josie112   Device    1       en0     Address
1       192.168.100.110 NetMask 1       255.255.252.0   Options 1       ""
RouteOptions    1       ""      PrefixLen       1       0
webip        ArgListValues         peter111   Device    1       en0     Address
1       192.168.100.110 NetMask 1       255.255.252.0   Options 1       ""
RouteOptions    1       ""      PrefixLen       1       0
```

Figure 4-2 Test environment topology

The cluster is set up as a two-node (*Peter111* and *Josie112)* cluster. There is one service IP (*corben110)* address.

The cluster resources and resource group defined are shown in Figure 4-3.

```
[josie112:root] / # hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  josie112             RUNNING              0
A  peter111             RUNNING              0

-- GROUP STATE
-- Group           System               Probed     AutoDisabled    State

B  ClusterService  josie112             Y          N               OFFLINE
B  ClusterService  peter111             Y          N               ONLINE
B  corbenSG01      josie112             Y          N               ONLINE
B  corbenSG01      peter111             Y          N               OFFLINE
B  corbenSG02      josie112             Y          N               ONLINE
B  corbenSG02      peter111             Y          N               OFFLINE
B  vxfen           josie112             Y          N               ONLINE
B  vxfen           peter111             Y          N               ONLINE
[josie112:root] / # hares -display |grep ONLINE
RES_phantom_vxfen State                josie112   ONLINE
RES_phantom_vxfen State                peter111   ONLINE
coordpoint        State                josie112   ONLINE
coordpoint        State                peter111   ONLINE
corbenDG01        State                josie112   ONLINE
corbenSG01_mnt1   State                josie112   ONLINE
corbenSG01_mnt2   State                josie112   ONLINE
corbenSG01_mnt3   State                josie112   ONLINE
csgnic            State                josie112   ONLINE
csgnic            State                peter111   ONLINE
nfsd              State                josie112   ONLINE
webip             State                peter111   ONLINE
```

Figure 4-3 Beginning cluster resource group configuration

The service group (*corbenSG01)* contains three VxFS striped mounts called */share/data<1/2/3>*, then the second service group *(corbenSG02)* is an example of a simple test application that starts, stops, and monitors the NFS daemon.

4.7.3 Collecting Cluster specifications

Now that we have the basics of the Symantec Cluster Server, we need to obtain the related information in order to perform our migration. Now we need to get the information about the type of resources that we have, as shown in Example 4-1.

Example 4-1 Collecting the cluster resource specifications

```
[peter111:root] / # hares -display -group corbenSG01 -attribute Type
#Resource         Attribute             System     Value
corbenDG01        Type                  global     DiskGroup
corbenSG01_mnt1   Type                  global     Mount
corbenSG01_mnt2   Type                  global     Mount
corbenSG01_mnt3   Type                  global     Mount
[peter111:root] / # hares -display -group corbenSG02 -attribute Type
#Resource         Attribute             System     Value
nfsd              Type                  global     Application
```

From the resource list shown in Example 4-1, we can see that our two service groups *corbenSG01* and *corbenSG02* are type DiskGroup and Application, so we can customize our queries as appropriate. First the DiskGroup, which we select from the information as shown in Example 4-2.

Example 4-2 DiskGroup information

```
[peter111:root] / # hares -display -group corbenSG01 -attribute BlockDevice
MountPoint FSType FsckOpt State
#Resource         Attribute             System     Value
corbenDG01        State                 josie112   ONLINE
corbenDG01        State                 peter111   OFFLINE

corbenSG01_mnt1   State                 josie112   ONLINE
corbenSG01_mnt1   State                 peter111   OFFLINE
corbenSG01_mnt1   BlockDevice           global     /dev/vx/dsk/corbendg01/stripe01
corbenSG01_mnt1   FSType                global     vxfs
corbenSG01_mnt1   FsckOpt               global     %-n
corbenSG01_mnt1   MountPoint            global     /share/data1

corbenSG01_mnt2   State                 josie112   ONLINE
corbenSG01_mnt2   State                 peter111   OFFLINE
corbenSG01_mnt2   BlockDevice           global     /dev/vx/dsk/corbendg01/stripe02
corbenSG01_mnt2   FSType                global     vxfs
corbenSG01_mnt2   FsckOpt               global     %-n
corbenSG01_mnt2   MountPoint            global     /share/data2

corbenSG01_mnt3   State                 josie112   ONLINE
corbenSG01_mnt3   State                 peter111   OFFLINE
corbenSG01_mnt3   BlockDevice           global     /dev/vx/dsk/corbendg01/stripe03
corbenSG01_mnt3   FSType                global     vxfs
corbenSG01_mnt3   FsckOpt               global     %-n
corbenSG01_mnt3   MountPoint            global     /share/data3
```

Example 4-2 on page 57 shows the information that we need to create a VxFS and add it into the cluster. But also some of this is what we need to migrate to PowerHA LVM. However, the application information is slightly different as shown in Example 4-3.

Example 4-3 Application information details

```
[peter111:root] / # hares -display -group corbenSG02 -attribute MonitorProgram
StartProgram StopProgram
#Resource     Attribute         System     Value
nfsd          MonitorProgram    global     /home/veritas/monitor/nfsd.monitor.sh
nfsd          StartProgram      global     /usr/bin/startsrc -s nfsd
nfsd          StopProgram       global     /usr/bin/stopsrc -s nfsd
```

Now that we have all the required information, we can advance to the next stage of migration.

4.8 Creating the LVM volume group and the file systems

In this section, we create the definitions needed in VCS and for our LVM volume group and journal file systems.

The steps we will be taking are as follows:

1. Create the volume group in LVM
2. Create the logical volumes
3. Create the file systems
4. Import the volume group across the cluster

Then, within the Symantec Cluster Server:

5. Define the new Service Group
6. Add the LVM volume group
7. Add the file system mounts
8. Link the resources together
9. Bring the file systems online
10. Test Failover

4.8.1 LVM creation

In this example, we cover the commands needed to create the LVM structure that we tested. This example gives a simple framework to start from:

```
[josie112:root] / # mkvg -V 52 -f -y corbenvg01 -s 256 -n -C hdisk7 hdisk8
corbenvg01
mkvg: This concurrent capable volume group must be varied on manually.
[josie112:root] / # varyonvg corbenvg01
[josie112:root] / # mklv -y lvstripe01 -t jfs2 -x 4096 -S1M corbenvg01 12 hdisk7
hdisk8
lvstripe01
```

Repeating the steps for *lvstripe02* and *lvstripe03* shows:

```
[josie112:root] / # crfs -v jfs2 -d lvstripe01 -m /data1_new -A no -a
logname=INLINE -a logsize=10
File system created successfully.
3135188 kilobytes total disk space.
New File System size is 6291456
```

Repeat again for /data2_new.

Once completed, varyoff the VG and import it into the other nodes in your cluster as follows:

```
[josie112:root] / # varyoffvg corbenvg01
[peter111:root] / # importvg -V 52 -y corbenvg01 hdisk7
corbenvg01
0516-783 importvg: This imported volume group is concurrent capable.
        Therefore, the volume group must be varied on manually.
[peter111:root] / #
```

The previous step imports all the mount points for you and gets them ready and fresh for the file systems. Remember that at the end, you need to leave the VG varied off and ready for the cluster to bring it online.

4.8.2 Add LVM to VCS

In this section, we cover all the commands that we need to create an LVM volume group within the Symantec Cluster Server. Though this can also be accomplished via the “Cluster Manager GUI”, not everyone may have access to it. So we show how to perform these tasks via the CLI.

1. First, we need to set the cluster to read/write:

   ```
   # haconf -makerw
   ```

2. Then, add the Service Group, specifying the wanted name. In our case, it is *corbenSG3lvm* with our two nodes *peter111* and *josie112*:

   ```
   # hagrp -add  corbenSG3lvm
   VCS NOTICE V-16-1-10136 Group added; populating SystemList and setting the
   Parallel attribute recommended before adding resources
   # hagrp -modify corbenSG3lvm SystemList  peter111 0 josie112 1
   # hagrp -modify corbenSG3lvm Parallel 0
   ```

3. Add the volume group. The resource is called *corbenSG3vg,* which is to be part of the service group *corbenSG3lvm.*

   Note: In order to create this resource, you need to have the volume group major number. Without it, you will not be able to bring the service group online. The *Enabled 1* parameter means this resource is enabled and ready to use, 0(Zero) is disabled, which cannot be bought online without enabling first.

   ```
   # hares -add  corbenSG3vg  LVMVG  corbenSG3lvm
   VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before
   agent monitors
   # hares -modify corbenSG3vg MajorNumber  52
   # hares -modify corbenSG3vg ImportvgOpt  n
   # hares -modify corbenSG3vg VolumeGroup  corbenvg01
   # hares -modify corbenSG3vg Enabled 1
   ```

4. Then, we add the related mounts; you will need to ensure that you have the correct parameters, those being the mount point, device path, file system type, mount options, and FSCK option. Note the percent(%) is required before the parameter so that the minus(-) is added to the parameter and not passed as part of the whole command. The speech marks(“) are also required to ensure the ‘jfs2’ parameter is passed with the mount options:

   ```
   # hares -add  corbenSG3lvmMNT1 Mount  corbenSG3lvm
   ```

```
VCS NOTICE V-16-1-10242 Resource added. Enabled attribute must be set before
agent monitors
# hares -modify corbenSG3lvmMNT1 MountPoint  /data1_new
# hares -modify corbenSG3lvmMNT1 BlockDevice  /dev/lvstripe01
# hares -modify corbenSG3lvmMNT1 FSType  jfs
# hares -modify corbenSG3lvmMNT1 MountOpt  "%-V jfs2"
# hares -modify corbenSG3lvmMNT1 FsckOpt %-p
# hares -modify corbenSG3lvmMNT1 Enabled 1
```

In this example, we have three file systems. Therefore, these commands are repeated twice more for the mounts and devices.

5. Then link the resources together. This is the parent resource linking to the child resource (see “Resources dependency” on page 49). In our example, the mount is linked to the volume group:

   ```
   # hares -link corbenSG3lvmMNT1 corbenSG3vg
   ```

6. Save the configuration and set the cluster back to read-only:

   ```
   # haconf -dump -makero
   ```

7. Finally, bring the resources online on a node of your choice:

   ```
   # hagrp -online corbenSG3lvm -sys peter111
   ```

Figure 4-4 shows an example of the new service group online.

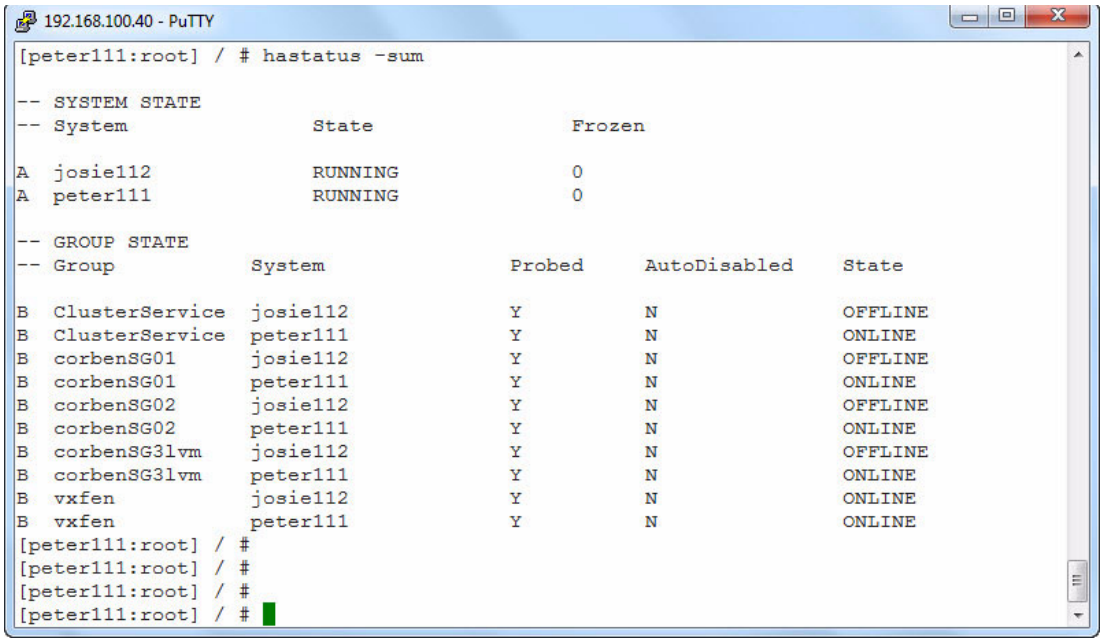

Figure 4-4 New Service Group Online

We also need to test that all is working correctly with the resource group on our cluster. A good simple test is to move the resources to another node in the cluster:

```
# hagrp -switch corbenSG3lvm -to josie112
```

We can see an example of the move in Figure 4-5 on page 61.

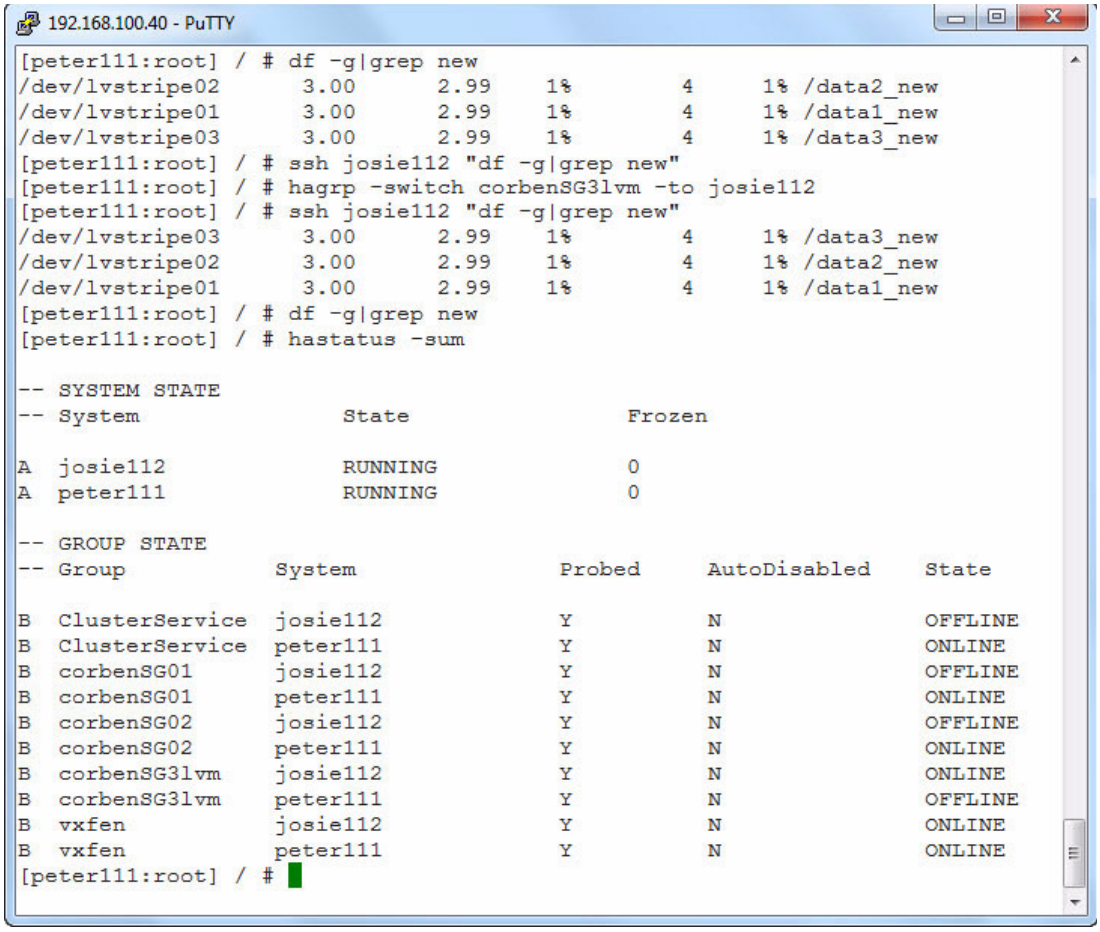

Figure 4-5 VCS LVM file system move

We recommend to perform full failover testing of the cluster now that we have added this new resource. This will enable us to determine if there are any resource conflicts. Details about how to back out of your changes are detailed in 4.11, “Roll-back and removal” on page 72, if needed.

4.9 Installing the PowerHA software

In our scenario, PowerHA Standard Edition is to be installed on the two local nodes.

When installing it, you can choose to install all, or only those filesets specific to your environment. The IBM PowerHA Standard Edition requires the installation and acceptance of license agreements for the Standard Edition `cluster.license` fileset.

This cluster was configured utilizing standard SMIT sysmirror menus. There is also the option of utilizing the PowerHA Systems Director plug-in to configure, monitor, and manage the cluster. More information about the Systems Director plug-in option can be found in the *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106.

To start creating the cluster, enter **`smitty sysmirror`** → **`Cluster Nodes and Networks`** → **`Standard Cluster Deployment`** → **`Setup a Cluster, Nodes and Networks`**.

Complete the options as wanted and press Enter. Upon execution, it will perform a discovery to gather both IP and shared disk information to be used in the cluster configuration.

The next step is to define a cluster repository disk and multicast address. We can use fast paths in SMIT to bypass additional menus. Execute **smitty cm_setup_menu → Define Repository Disk and Cluster IP Address**.

For the repository disk field, press F4 and get a pick list to choose the wanted disk. Our repository disk was made available between the nodes in the cluster with Shared Storage Pools, something that is now fully supported with PowerHA. This data is gathered during the discovery in the first step of creating the cluster. The available disks list is created by finding all shared disks, with PVIDs, not currently in a volume group. See the troubleshooting section in 4.13, “Troubleshooting, and known issues” on page 73 if you cannot find any disk or get errors.

Since this is going to be a PowerHA 7.1.3 cluster, we utilize *unicast* instead of the previously required *multicast* as the primary heartbeat mechanism. The Cluster IP address is not required to be entered. If you want one, you can either manually enter it or let PowerHA create one. It does this by taking the last three octets of the host name IP address from the node in which the cluster is being created on and replacing the first octet with 228.

After performing the previous two steps, it is recommended to synchronize the cluster. Execute **smitty sysmirror → Cluster Nodes and Networks → Verify and Synchronize Cluster Configuration** and press Enter twice. The main reason being is that the first time the cluster is synced, the CAA cluster is created automatically. This way if a problem is encountered, it can be addressed before adding all the additional cluster components. Figure 4-6 shows the disk and CAA volume group information after the synchronization and CAA cluster was created successfully.

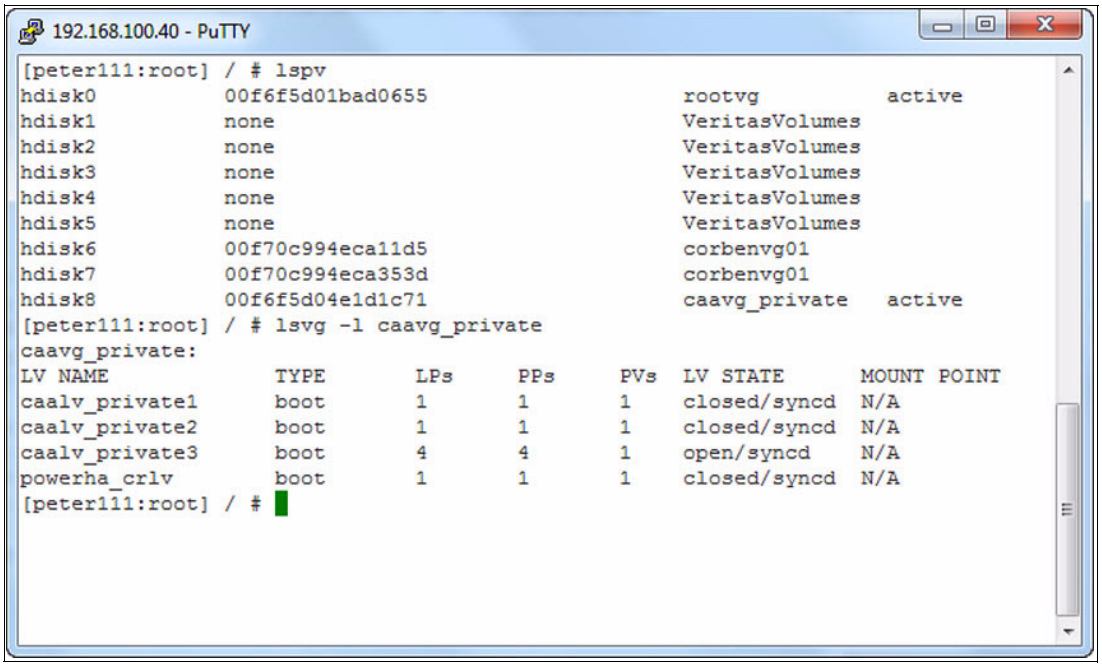

Figure 4-6 CAA cluster created successfully

Though optional, and not used in this test configuration, it is considered a best practice to also configure SAN-based communications. This requires setting the appropriated FC adapter attributes on VIO servers, and adding virtual Ethernet adapters using VLAN 3358.

We now need to create our resources. This will consist of an application controller, service address, and shared volume group. They will then be added into a resource group.

In our scenario, we have no real application to utilize. So we created a dummy application controller by simply having it execute a banner command. We can add it by executing **smitty sysmirror → Cluster Applications and Resources → Resources → Configure User Applications (Scripts and Monitors) → Application Controller Scripts**.

In our case, we will be using the information we collected from 4.7.3, “Collecting Cluster specifications” on page 57.

Optionally, you can also add a monitor for your application, though it is not required for the cluster to start. This differs from VCS, which has a need for the monitor to configure the application resource.

Now we need to add a service IP address. To do so, execute the fast path of **smitty cm_resources_menu → Configure Service IP Labels/Addresses → Add a Service IP Label/Address** (choose “Network Name” from pop-up). In our cluster, we created it on *Peter111* with host name address of 192.168.100.110, as this is the cluster address defined for VCS.

After adding the service IP, we can see it has been added to the cluster topology as shown from the *cllsif* output below in Figure 4-7.

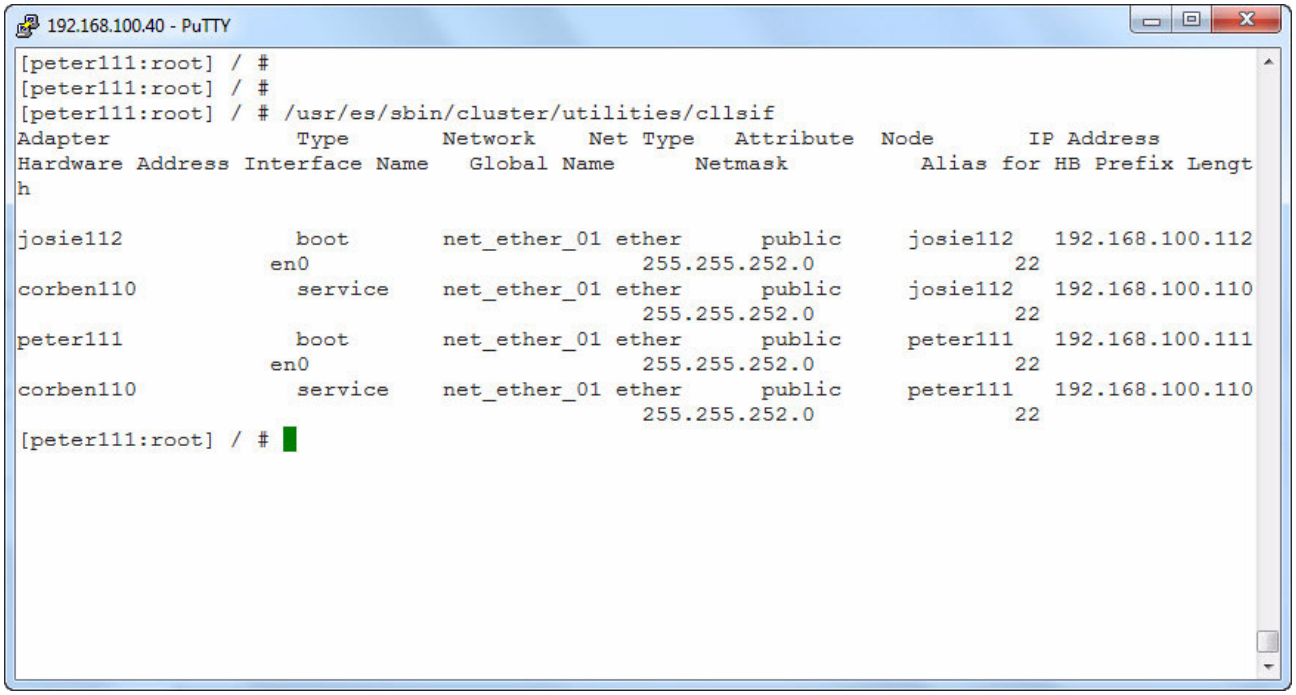

Figure 4-7 cllsif output

Next, we create a resource group and add the resources to it. To create a new resource group, execute the fast path **smitty cm_add_resource_group**.

To add the resources to the resource group, execute the same fast path of **smitty cm_resource_groups → Change/Show Resources and Attributes for a Resource Group** and choose the previously created resource group. Then, for the fields *of Service IP Labels/Addresses, Application Controllers,* and *Volume Groups*, press F4 and a pop-up window appears with the ones previous created. Choose them, and press Enter.

If you need to add any additional storage, this can be accomplished by using the Cluster Single Point of Control facility (C-SPOC). Enter **smitty cspoc → Storage →** ***Volume Groups → Create a Volume Group*** (choose both nodes).

For your environment, repeat the previous steps as needed. Once completed, just to be sure, go ahead and synchronize the cluster.

We now have a two-node “hot-standby” cluster created consisting of the following:

- Two nodes, *peter111* and *josie112*)
- One IP-network, *net_ether_01*)
- One repository disk, *hdisk8*
- One resource group, *corben110rg*, peter111 is primary, josie112 is backup.
- One application server, *nfsd*
- One service address, *corben110*, with IP of 192.168.100.110 via IP aliasing
- One shared VG, *corbenvg01*

The cluster configuration details can be seen in Figure 4-8.

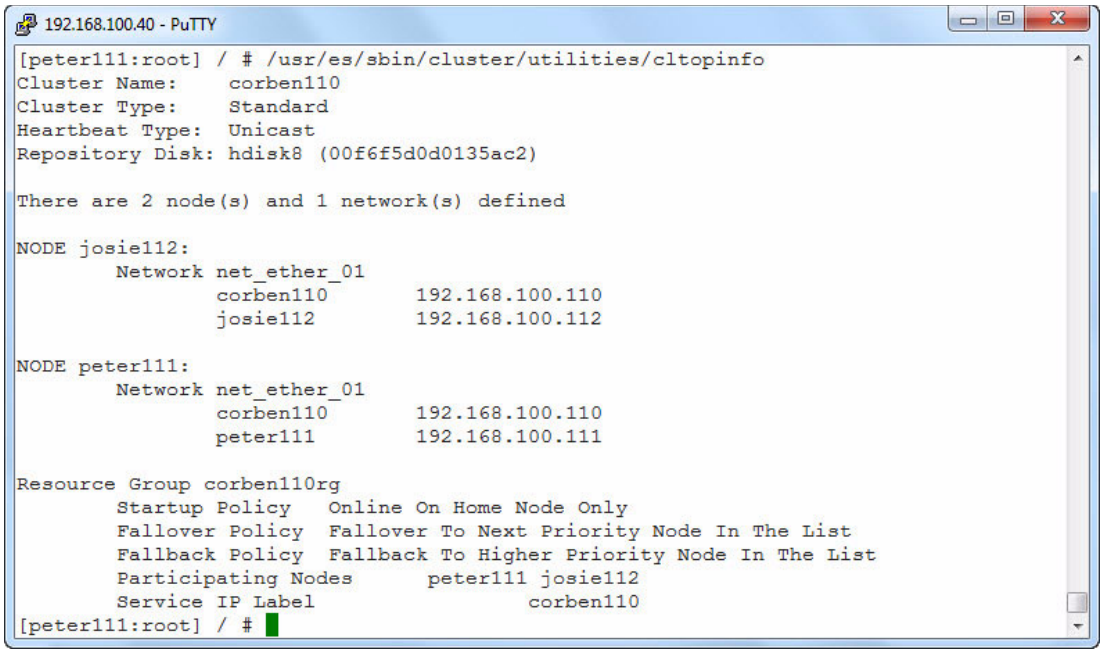

Figure 4-8 cltopinfo output

We leave the cluster down ready for testing as PowerHA and VCS are now sharing resources. We have successfully had both instances working together ready to take control but this is not a supported or recommended configuration.

4.9.1 Testing the cluster

To execute the cluster test tool enter **smitty hacmp_testtool_menu**, then choose “Execute Automated Test Procedure” as shown in Figure 4-9.

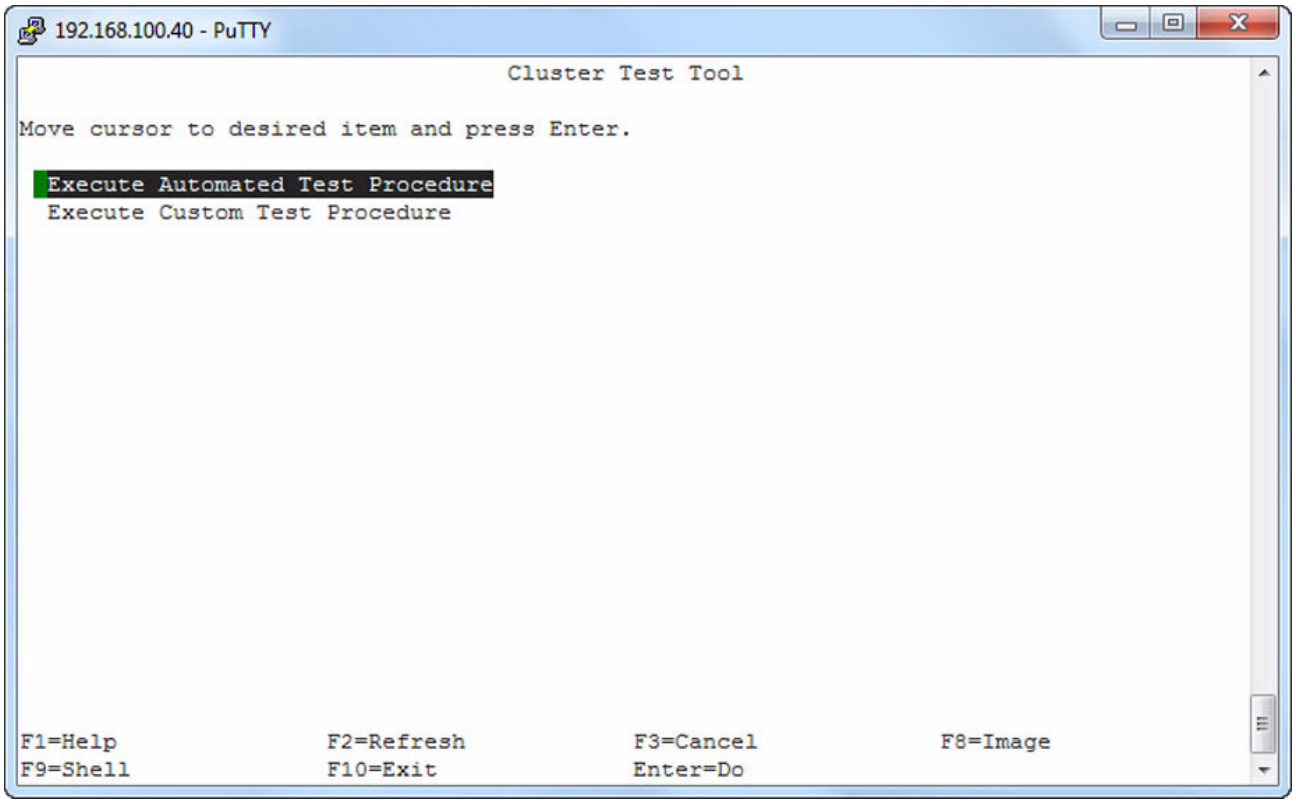

Figure 4-9 Cluster test tool menu

When you press Enter, the final menu is displayed. The detailed results of each test are displayed in the SMIT window during execution (Example 4-4), and are also saved in */var/hacmp/log/cl_testtool.log*.

Example 4-4 Output while testing the cluster

```
[peter111:root] / # cat /var/hacmp/log/cl_testtool.log|egrep "Complete|Completion"
|| Test 1 Complete - NODE_UP: Start cluster services on all available nodes
25/06/2014_07:17:49: || Test Completion Status: PASSED
|| Test 2 Complete - NODE_DOWN_GRACEFUL: Stop cluster services gracefully on a
node
25/06/2014_07:18:35: || Test Completion Status: PASSED
|| Test 3 Complete - NODE_UP: Restart cluster services on the node that was
stopped
25/06/2014_07:19:31: || Test Completion Status: PASSED
|| Test 4 Complete - NODE_DOWN_TAKEOVER: Stop cluster services with takeover on a
node
25/06/2014_07:21:17: || Test Completion Status: PASSED
|| Test 5 Complete - NODE_UP: Restart cluster services on the node that was
stopped
25/06/2014_07:23:44: || Test Completion Status: PASSED
|| Test 6 Complete - NODE_DOWN_FORCED: Stop cluster services forced on a node
25/06/2014_07:24:18: || Test Completion Status: PASSED
```

```
|| Test 7 Complete - NODE_UP: Restart cluster services on the node that was
stopped
25/06/2014_07:25:55: || Test Completion Status: PASSED
## Cluster Testing Complete: Exit Code 0
|| Test 1 Complete - VG_DOWN: Bring down volume group
25/06/2014_07:27:28: || Test Completion Status: PASSED
## Cluster Testing Complete: Exit Code 0
|| Test 1 Complete - CLSTRMGR_KILL: Kill the cluster manager on a node
25/06/2014_07:29:50: || Test Completion Status: PASSED
## Cluster Testing Complete: Exit Code 0
```

The overall test time was 14 minutes and each of the following events were executed successfully:

1. NODE_UP - Each node one at a time.
2. NODE_DOWN_GRACEFUL – Same as above.
3. NODE_UP – Same as above.
4. NODE_DOWN_TAKEOVER – Graceful down and moves resource group from peter111 to josie112.
5. NODE_UP – Restart services on previously down node (peter111).
6. NODE_DOWN_FORCED – On peter111.
7. NODE_UP – Restart services on previously down node (peter111).
8. VG_DOWN – Simulates volume group loss (rg_move runs from josie112 to peter111).
9. CLSTRMGR_KILL – Creates hard fallover via halt on peter111.

While this testing does cover the core basic functionality of the cluster, additional granular level testing via the Custom Test Procedure is often wanted to include such common events as:

- FAIL_LABEL – (Both IP and Non-IP)
- NETWORK_DOWN_LOCAL – (Both IP and Non-IP)
- JOIN_LABEL – (Both IP and Non-IP)
- NETWORK_UP_LOCAL – (Both IP and Non-IP)
- SERVER_DOWN – (nice test when application monitoring is being used)

There are several specific events related to additional configuration options within PowerHA. This includes, but is not limited to, sites and global networks. Manually creating failures for testing is also encouraged, for example, disabling ports, pull cables, and so on.

4.10 Performing the migration

The migration of the servers is relatively simple once all the devices and cluster have been configured. When you can schedule downtime, you should be able to migrate the cluster services from VCS to PowerHA.

Note: We mentioned previously that this work can be done with the cluster live, but it is *not* supported. Without testing and knowing the application/environment involved, it is impossible for us and the related support teams to guarantee anything. All actions and commands performed are taken from well documented and supported processes. We have consolidated the actions together in order to perform our migrations. We can confirm in the scenarios mentioned in this IBM Redbooks publication that we were able to migrate a live cluster from VCS to PowerHA without any downtime or outages, but we cannot make this same guarantee in any other situation.

1. First, ensure all data has been synchronized over to the new file systems.
2. Stop VCS processes and disable the applications from auto starting on restart:

```
# haconf -makerw
# hagrp -disable corbenSG3lvm
# hagrp -disable corbenSG02
# hagrp -disable corbenSG01
# hagrp -disable vxfen
# hagrp -disable ClusterService
# hagrp -offline corbenSG3lvm -any
# hagrp -offline corbenSG02
# hagrp -offline corbenSG01
# hagrp -offline vxfen
# hagrp -offline -force ClusterService -any
# hagrp -disableresources vxfen
# hagrp -disableresources ClusterService
# haconf -dump -makero
```

   In the previous steps, we are disabling the resources in order, then taking them in turn offline, and finally disabling and locking the `vxfen` and `ClusterService` so that they do not restart automatically on restart. Otherwise, VCS on a reboot/failover will grab cluster resources before or during PowerHA events.

3. Confirm the state of VCS:

```
[peter111:root] / # hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  josie112             RUNNING              0
A  peter111             RUNNING              0

-- GROUP STATE
-- Group           System               Probed     AutoDisabled    State

B  ClusterService  josie112             Y          N               OFFLINE
B  ClusterService  peter111             Y          N               OFFLINE
B  corbenSG01      josie112             Y          N               OFFLINE
B  corbenSG01      peter111             Y          N               OFFLINE
B  corbenSG02      josie112             Y          N               OFFLINE
B  corbenSG02      peter111             Y          N               OFFLINE
B  corbenSG3lvm    josie112             Y          N               OFFLINE
B  corbenSG3lvm    peter111             Y          N               OFFLINE
B  vxfen           josie112             Y          N               OFFLINE
B  vxfen           peter111             Y          N               OFFLINE
```

As you can see now everything is offline and ready to move to PowerHA.

4. Now we can start PowerHA knowing that the file systems and IP address are not in use elsewhere, as shown in Figure 4-10.

```
# smitty clstart
```

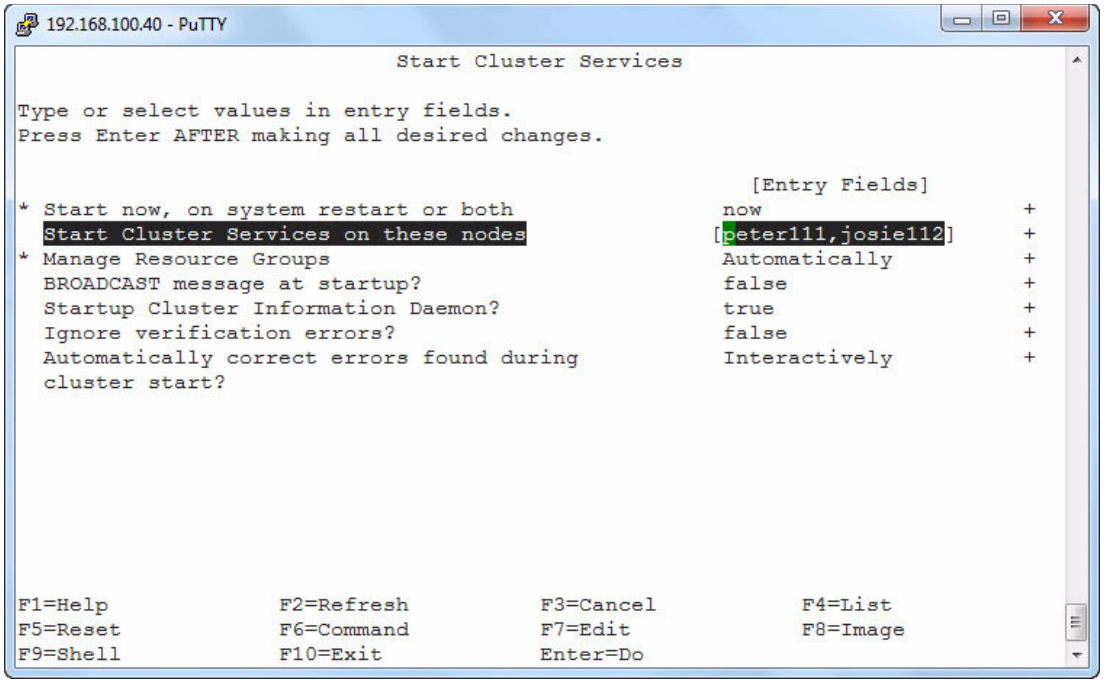

Figure 4-10 Start the cluster

5. Once this is complete, you can confirm the cluster state, first the cluster daemon:

```
[peter111:root] / # lssrc -ls clstrmgrES
Current state: ST_STABLE
sccsid = "@(#)36 1.135.1.118
src/43haes/usr/sbin/cluster/hacmprd/main.C,hacmp.pe,61haes_r713,1343A_hacmp713
10/21/"
build = "May  6 2014 15:08:06 1406D_hacmp713"
i_local_nodeid 1, i_local_siteid -1, my_handle 1
ml_idx[1]=1     ml_idx[2]=0
There are 0 events on the Ibcast queue
There are 0 events on the RM Ibcast queue
CLversion: 15
local node vrmf is 7131
cluster fix level is "1"
The following timer(s) are currently active:
monitor nfsd_mon:::RestartTimer
Current DNP values
DNP Values for NodeId - 2  NodeName - josie112
    PgSpFree = 259592  PvPctBusy = 0  PctTotalTimeIdle = 95.663888
DNP Values for NodeId - 1  NodeName - peter111
    PgSpFree = 259529  PvPctBusy = 0  PctTotalTimeIdle = 95.510347
CAA Cluster Capabilities
CAA Cluster services are active
There are 4 capabilities
Capability 0
  id: 3  version: 1  flag: 1
```

```
  Hostname Change capability is defined and globally available
Capability 1
  id: 2  version: 1  flag: 1
  Unicast capability is defined and globally available
Capability 2
  id: 0  version: 1  flag: 1
  IPV6 capability is defined and globally available
Capability 3
  id: 1  version: 1  flag: 1
  Site capability is defined and globally available
trcOn 0, kTraceOn 0, stopTraceOnExit 0, cdNodeOn 0
Last event run was JOIN_NODE_C0  on node 2
```

– Then, the cluster IP:

```
[peter111:root] / # /usr/es/sbin/cluster/utilities/cllsif
Adapter    Type     Network     Net Type  Attribute Node      IP Address
Hardware Address Interface Name  Global Name Netmask       Alias for HB
Prefix Length
josie112   boot     net_ether_01 ether    public    josie112
192.168.100.112              en0                    255.255.252.0
22
corben110  service  net_ether_01 ether    public    josie112
192.168.100.110                                     255.255.252.0
22
peter111   boot     net_ether_01 ether    public    peter111
192.168.100.111              en0                    255.255.252.0
22
corben110  service  net_ether_01 ether    public    peter111
192.168.100.110                                     255.255.252.0
22
```

– Followed by the cluster topology:

```
[peter111:root] / # /usr/es/sbin/cluster/utilities/cltopinfo
Cluster Name:    corben110
Cluster Type:    Standard
Heartbeat Type:  Unicast
Repository Disk: hdisk8 (00f6f5d0d0135ac2)

There are 2 node(s) and 1 network(s) defined

NODE josie112:
        Network net_ether_01
                corben110       192.168.100.110
                josie112        192.168.100.112

NODE peter111:
        Network net_ether_01
                corben110       192.168.100.110
                peter111        192.168.100.111

Resource Group corben110rg
        Startup Policy   Online On Home Node Only
        Fallover Policy  Fallover To Next Priority Node In The List
        Fallback Policy  Fallback To Higher Priority Node In The List
        Participating Nodes      peter111 josie112
        Service IP Label                 corben110
```

– And the local state of the cluster, confirming the mounts and IP address on the adapter:

```
[peter111:root] / # df -g|grep data
/dev/lvstripe01      3.00      2.99    1%        4     1% /data1_new
/dev/lvstripe02      3.00      2.99    1%        4     1% /data2_new
/dev/lvstripe03      3.00      2.99    1%        4     1% /data3_new
[peter111:root] / # ifconfig -a
en0:
flags=1e084863,10480<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUP
RT,64BIT,CHECKSUM_OFFLOAD(ACTIVE),CHAIN>
        inet 192.168.100.110 netmask 0xfffffc00 broadcast 192.168.103.255
        inet 192.168.100.111 netmask 0xfffffc00 broadcast 192.168.103.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
lo0:
flags=e08084b,c0<UP,BROADCAST,LOOPBACK,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,LARGESEND,CHAIN>
        inet 127.0.0.1 netmask 0xff000000 broadcast 127.255.255.255
        inet6 ::1%1/0
         tcp_sendspace 131072 tcp_recvspace 131072 rfc1323 1
```

6. Finally, we do a test failover to make sure our resources respond as expected, so on the current node holding the cluster resources:

```
# halt -q
```

7. Then, monitor the system failover. See Figure 4-11, Figure 4-12 on page 71, and Figure 4-13 on page 71.

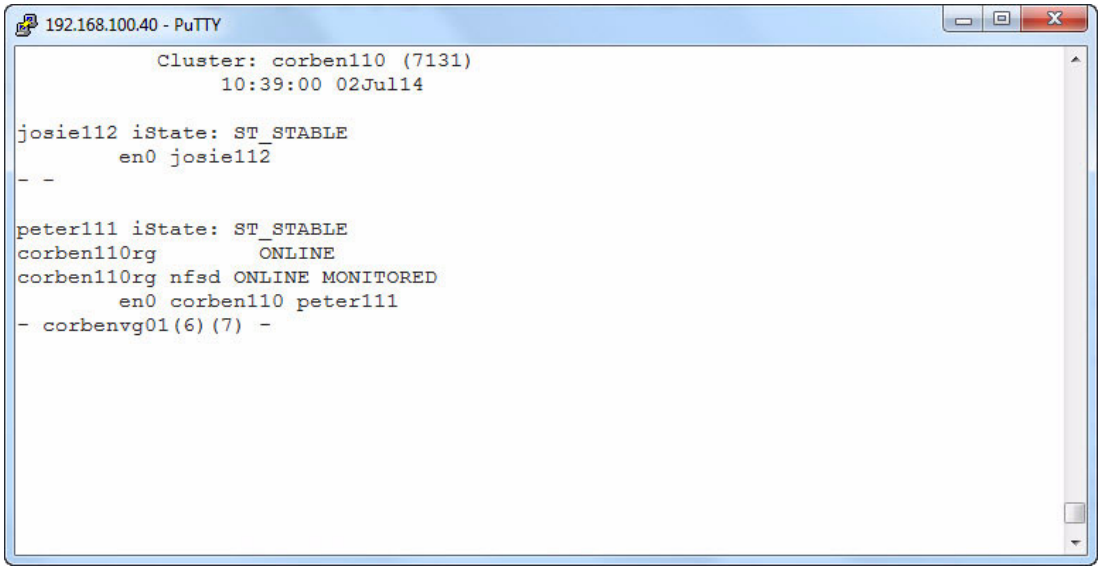

Figure 4-11 Stable cluster - qha

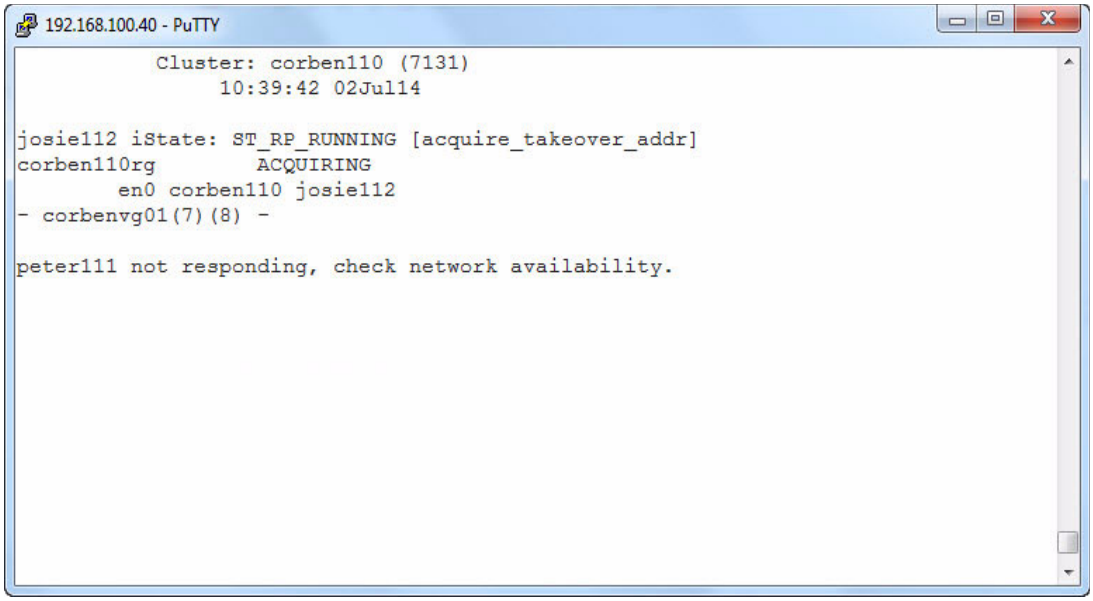

Figure 4-12 Cluster failing over - qha

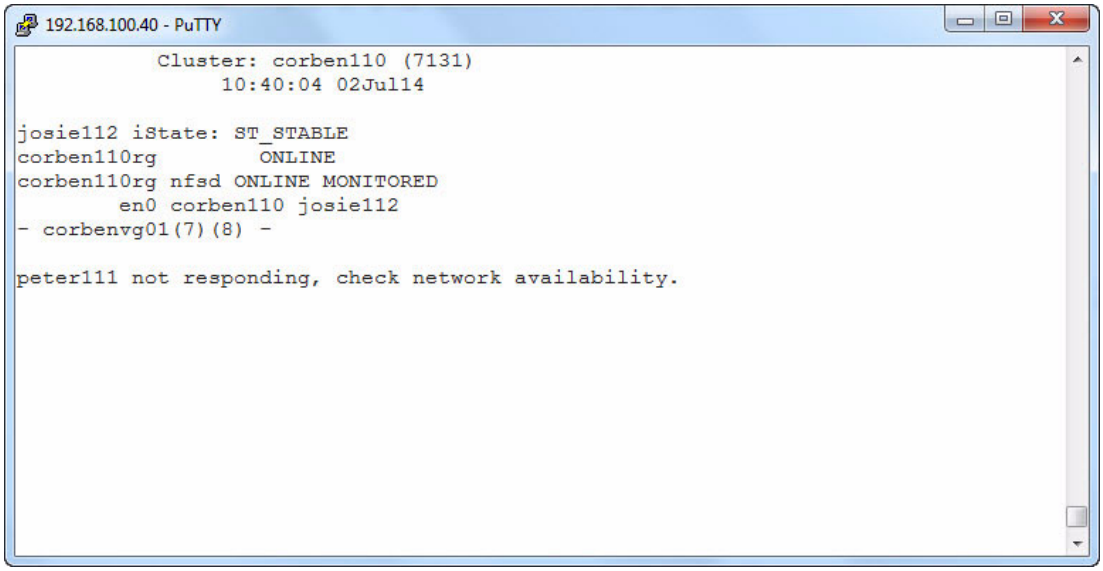

Figure 4-13 Cluster failover complete - qha

In the examples above, we used *qha* to monitor the cluster, which is a nice simple shell script that you can obtain from the following site:

http://www.powerha.lpar.co.uk

Check to ensure that you have the latest version of qha. Version 9.01 works with PowerHA 7.1.3. Earlier versions work with PowerHA 6.1 and before as they use the *clcomdES* deamon, which is no longer part of PowerHA 7.1.3.

Note: In a production environment, we always recommend that you perform full cluster testing. This could involve the removal of disks, cables, failure of servers, and more. Details of examples of this testing can be found in the *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX*, SG24-8106.

4.11 Roll-back and removal

If any issues are encountered with our new resource group, mounts, or volume groups, we may need to remove everything. This could be due to conflicts with other resources, applications, or file systems that we could not have foreseen. So in this section, we cover the commands needed to remove everything that we created in 4.6, “Converting a Symantec Cluster Server to an IBM PowerHA cluster” on page 54.

First, we take the resources offline across the cluster:

```
# hagrp -offline corbenSG3lvm -any
```

Ensure that we put the cluster into read/write mode, otherwise, our changes will not work:

```
# haconf -makerw
```

We then delete our LVM volume group details:

```
# hares -delete corbenSG1vg
```

Followed by deleting all the file system mounts:

```
# hares -delete corbenSG3lvmMNT1
# hares -delete corbenSG3lvmMNT2
# hares -delete corbenSG3lvmMNT3
```

Then, delete the service group that everything was owned by:

```
# hagrp -delete corbenSG3lvm
```

Finally, save and set the cluster back to read-only, otherwise, the changes will not take affect:

```
# haconf -dump -makero
```

Now the cluster is back to how it was set up before we added the details. We can then go and remove the related LVM file systems, logical volumes, and volume groups as needed.

4.12 Deleting the Symantec Cluster Server

Our last action will be the removal of the cluster. We can of course leave the service on until we decide it is time to remove it, so that we can ensure that everything is working first.

First, delete the Service Group and the related resources:

```
# haconf -makerw
# hares -delete <mounts>
```

Repeat as required:

```
#hares -delete <service_group>
```

Repeat as required, then delete the disk group:

```
# hares -delete RES_phantom_vxfen
# hares -delete coordpoint
# hares -delete vxfen
```

When that is deleted, you can then delete the cluster service:

```
# hares -delete csgnic
```

```
# hares -delete webip
# hares -delete ClusterService
```

Finally, save the config:

```
# haconf -dump -makero
```

Stop any remaining cluster services:

```
# hastop -all -force
```

Disable fencing:

```
# /sbin/vxfenconfig -U
```

Then, remove the packages in order based on their dependencies:

```
# installp -u VRTSamf VRTSaslapm VRTScps VRTSdbed VRTSfsadv VRTSfssdk VRTSgab
VRTSob VRTSodm VRTSperl VRTSsfcpi61 VRTSsfmh VRTSspt VRTSvbs VRTSvcs VRTSvcsag
VRTSvcsea VRTSvcswiz VRTSvxfen VRTSvxfs VRTSvxvm
# installp -u VRTSllt
# installp -u VRTSveki
# installp -u VRTSvlic
```

4.13 Troubleshooting, and known issues

As VxVM is the primary storage controller on these systems, when moving to LVM during these tests we noticed some technical issues. So in this section we cover the problems that we have seen.

4.13.1 Volume Manager holds the disk

After creating a new volume group for PowerHA on a system with an existing Symantec Cluster, we found that we could not find any disks to access, or it displayed the following errors when creating a volume group:

```
1800-050 Error exit status (1) returned by
Command_to_List; the output is:
"# No free disks found.
# Check each node to see if any disks need to have PVIDs allocated"
(The Command_to_List is:
"/usr/es/sbin/cluster/cspoc/cllspvids -O -n  'peter111' ".)
```

or

```
0516-1254 /usr/sbin/mkvg: Changing the PVID in the ODM.
0516-1397 /usr/sbin/mkvg: The physical volume hdisk6, will not be added to
the volume group.
0516-1254 /usr/sbin/mkvg: Changing the PVID in the ODM.
0516-1397 /usr/sbin/mkvg: The physical volume hdisk7, will not be added to
the volume group.
0516-862 /usr/sbin/mkvg: Unable to create volume group.
```

This is because the Symantec Storage Foundation is holding on to them:

```
[peter111:root] / # vxdisk -e list
DEVICE      TYPE          DISK  GROUP STATUS          OS_NATIVE_NAME  ATTR
disk_0      auto:LVM      -      -    LVM            hdisk0          -
```

```
disk_1       auto:LVM      -      -    LVM             hdisk8          -
ds3400-0_0   auto:cdsdisk  -      -    online          hdisk4          -
ds3400-0_1   auto:cdsdisk  -      -    online          hdisk2          -
ds3400-0_2   auto:cdsdisk  -      -    online          hdisk5          -
ds3400-0_3   auto:cdsdisk  -      -    online          hdisk1          -
ds3400-0_4   auto:cdsdisk  -      -    online          hdisk3          -
ds3400-0_5   auto:none     -      -    online invalid  hdisk6          -
ds3400-0_6   auto:none     -      -    online invalid  hdisk7          -
```

We need to remove the disk from VxVM so that LVM is allowed access to the disk:

```
# vxdisk rm <disk>
```

The *disk* name can be either the *hdisk* name or the *device* name.

Make sure that we run this on all nodes in our cluster:

```
[peter111:root] / # vxdisk -e list
DEVICE       TYPE          DISK  GROUP STATUS          OS_NATIVE_NAME  ATTR
disk_0       auto:LVM      -      -    LVM             hdisk0          -
disk_1       auto:LVM      -      -    LVM             hdisk8          -
ds3400-0_0   auto:cdsdisk  -      -    online          hdisk4          -
ds3400-0_1   auto:cdsdisk  -      -    online          hdisk2          -
ds3400-0_2   auto:cdsdisk  -      -    online          hdisk5          -
ds3400-0_3   auto:cdsdisk  -      -    online          hdisk1          -
ds3400-0_4   auto:cdsdisk  -      -    online          hdisk3          -
```

Now we have access again and we can create our volume group:

```
[peter111:root] / # mkvg -s 128 -y testvg hdisk6 hdisk7
0516-1254 mkvg: Changing the PVID in the ODM.
testvg
```

4.13.2 Volume group will not varyon with VCS after PowerHA test

After testing of the PowerHA cluster software, you may find that the Symantec Cluster Server is unable to start the LVM Service. This might be because of some locks placed on the new LVM volume group. These can be cleared by varying on and off the volume group as follows:

```
[josie112:root] / # varyonvg corbenvg01
0516-1972 varyonvg: The volume group is varied on in other node in concurrent
        mode; you cannot vary on the volume group in non-concurrent mode.  Use
        -O flag to force varyon the volume group if needed.
[josie112:root] / # varyonvg -O corbenvg01
[josie112:root] / # varyoffvg corbenvg01
```

4.14 Shared storage pools

Throughout this IBM Redbooks publication, all the servers rootvg were built on disk using Shared Storage Pools (SSPs). A number of reasons were chosen for doing this:

1. IBM PowerHA supports SSP but it has not been used before.
2. Flexibility of the storage support enables us to quickly remove, move, add, and manage the disk locally without outside support.
3. SSP snapshots allowed us to roll back changes to the environment with minimal outage time. This enabled quick retesting of scenarios and recovery from issues.

While Shared Storage Pools are not supported by Symantec Cluster Server due to the SCSI3 reserve requirements on the resource disks, we would urge that in most scenarios the benefits outweigh this issue. Like us, you can put your operating system (AIX) boot disk on SSP along with the PowerHA cluster disk using NPIV for the Symantec Cluster Server.

The following are examples of the commands that we used for the creation, and recovery of the servers from the VIO servers. Listing your snapshots:

```
# snapshot -list -clustername <cl_name> -spname <pool_name>
```

Create a snapshot:

```
# snapshot -clustername <cl_name> -spname <pool_name> -create <file_name> -lu
<node_name>
```

Rolling back to a previous snapshot (ensuring the virtual server is powered-down first):

```
# snapshot -clustername <cl_name> -spname <pool_name> -rollback <file_name> -lu
<node_name>
```

4.15 Cluster migration

This section describes the cluster migration.

4.15.1 PowerHA and Storage Foundation

In PowerHA, third-party volume groups and file systems are treated as OEM and usually require specific configuration of additional management methods to make them manageable within a PowerHA resource group. Preinstalled methods for managing Symantec volume groups and file systems until Veritas Volume Manager version 4.0 are bundled into PowerHA. For other versions and products, a set of scripts must be created and configured in PowerHA.

As mentioned before, newer versions of Storage Foundation (above 4.0) are not supported in PowerHA. Although we found that statement in the official PowerHA documentation, we tried to use the embedded feature with our recent version in an attempt to check how much of the functionalities were still working.

While most of the scripts were found to work individually with minor or no changes at all, we had problems while running the verification and synchronization processes of PowerHA, which proved the official statements about the support and forced us to abort the trial.

In order to integrate the Symantec diskgroups and file systems with our PowerHA cluster, we had to choose using the *user-defined resources* features of PowerHA. The method used will be explained in the next sections.

Note: Typically when adding an LVM volume group to a resource group, the **F4** key may be used to list the available volume groups in the system. Since our tests were performed with an unsupported version of Storage Foundation, we were not able to verify this functionality when using Storage Foundation diskgroups.

4.15.2 Configuring User-Defined Resources

As mentioned before, PowerHA does not support recent versions of VxVM volumes as OEM resources. As the test environment is running Storage Foundation version 6.1, one of our major concerns was related to how data could be managed in order to allow the migration of the cluster software.

In PowerHA 7.1, the Storage Foundation volumes and file systems can be configured as *User-Defined Resources*. These types of resources require that a set of scripts is created to accomplish the wanted tasks.

The SMIT menu to define the user-defined resources can be accessed through: ***smitty sysmirror → Custom Cluster Configuration → Resources → Configure User Defined Resources and Types***. The menu listing can be seen in Example 4-5.

Example 4-5 SMIT - Configure User Defined Resources and Types

```
                    Configure User Defined Resources and Types

Move cursor to desired item and press Enter.

Configure User Defined Resource Types
Configure User Defined Resources

Import User Defined Resource Types and Resources Definition from XML file

F1=Help             F2=Refresh          F3=Cancel           F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The menu as shown in Example 4-5 has two different entries: ***Configure User Defined Resource Types*** and ***Configure User Defined Resources***. The first entry will be used to define a new resource type handler. That is where you define all the scripts and parameters that will control the behavior of all resources associated to that type. The second entry is basically where you create a resource associated to the resource type previously defined.

The next sections illustrate how Storage Foundation resources can be enabled to work with PowerHA.

4.15.3 Enabling Storage Foundation Resources in PowerHA

Now that we talked about the *User Defined Resources*, we illustrate how to enable the Storage Foundation *diskgroups* and *file systems* in our PowerHA cluster.

Storage Foundation Diskgroups

In Example 4-6 on page 77, we create a new resource type named ***SFDG*** that performs all the activation and deactivation of the Storage Foundation *diskgroups*. The assistant requires four fields to be filled, all marked with an asterisk and shown in **bold** in the example.

First, the name of the new resource type is required. Then, the *Processing Order*, which defined the moment at which the resource will be processed during the startup or shutdown of the resource group. PowerHA provided a number of options for this field and we selected ***FIRST*** as our value because we want it to be processed when the resource acquisition starts.

Next, the startup and stop methods are required. These fields require the full path of the scripts created to handle the resources of this type. Since we are creating a resource to handle Storage Foundation *diskgroups*, we add two scripts that perform whatever tasks are necessary to start and stop the *diskgroups*.

Example 4-6 PowerHA - Adding a custom resource type for the diskgroups

```
                        Add a User Defined Resource Type

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Resource Type Name                                [SFDG]
* Processing order                                  [FIRST]                +
Verification Method                                 []
    Verification Type                               [Script]               +
* Start Method
[/usr/es/sbin/cluster/OEM/MyResources/dgstart]
* Stop Method
[/usr/es/sbin/cluster/OEM/MyResources/dgstop]
  Monitor Method                                    []
  Cleanup Method                                    []
  Restart Method                                    []
  Failure Notification Method                       []
  Required Attributes                               []
  Optional Attributes                               []
  Description                                       []


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

As you can see, the menu allows the defining additional methods to perform other tasks. It is important that these parameters are used and set wisely. Of course, there are different aspects to be considered when deciding which methods will be used and skipped.

The scripts **dgstart** and **dgstop** are simple wrappers for the **vxdg** command to *import* and *deport* the diskgroups. The scripts must receive a parameter from PowerHA, which will be the name of the *diskgroup*. Therefore, it is important that the name of the *diskgroups* of your cluster are not hard-coded.

The contents of **dgstart** script are shown in Example 4-7, which illustrates how the scripts may look. The complexity of the scripts is dictated by how your systems are configured.

Example 4-7 dgstart script to import Storage Foundation diskgroups during resource group acquisition

```
#!/usr/bin/ksh93
# This script will import a given Storage Foundation diskgroup. PowerHA will call
# this script during the activation of the resource group.
#
```

```
/usr/sbin/vxdg import ${1}
if [ $? -ne 0 ]; then
  return 1
fi

return 0
```

With the new resource type created, we can now proceed and add a *User Defined Resource* to start our Storage Foundation diskgroups. To add the resource, use **smitty sysmirror** → **Custom Cluster Configuration** → **Resources** → **Configure User Defined Resources** → **Add a User Defined Resource**. Then, select the type of resource from the list, in this case, **SFDG** and press Enter.

The creation of a new resource is shown in Example 4-8. The *Resource Name* must be the exact name of the resource being handled. This name will be passed to the start and stop scripts as a parameter to be processed.

Hint: Custom naming conventions can be created to make it easier to identify the type of the resources according to their name, as long as the scripts are able to handle the name and process it properly during all cluster operation.

In Example 4-8, our resource is called *testdg* just like the real name of our *diskgroup*.

Example 4-8 PowerHA - Add a User Defined Resource

```
                        Add a User Defined Resource

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Resource Type Name                                   SFDG
* Resource Name                                       [testdg]
  Attribute data                                      []

F1=Help            F2=Refresh          F3=Cancel           F4=List
F5=Reset           F6=Command          F7=Edit             F8=Image
F9=Shell           F10=Exit            Enter=Do
```

The configuration of the *User Defined Resource* is that simple. You can press Enter again and finish the creation of the resource.

Note: In this example, PowerHA will display a warning about the lack of a monitor method for the resource. For our demonstration, we will just ignore that message.

Storage Foundation file systems

The tasks to enable Storage Foundation file systems in PowerHA is very similar to the tasks we used for the *diskgroups*, except that this time our *Processing Order* parameter will be different.

The SMIT menu (Example 4-9 on page 79) to define the user-defined resources can be accessed through: **smitty sysmirror** → **Custom Cluster Configuration** → **Resources** → **Configure User Defined Resources and Types**.

Example 4-9 PowerHA - Adding a custom resource type for the file systems

```
                        Add a User Defined Resource Type

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Resource Type Name                                  [SFFS]
* Processing order                                    [SFDG]                +
Verification Method                                  []
    Verification Type                                 [Script]              +
* Start Method
[/usr/es/sbin/cluster/OEM/MyResources/mount]
* Stop Method
[/usr/es/sbin/cluster/OEM/MyResources/umount]
  Monitor Method                                      []
  Cleanup Method                                      []
  Restart Method                                      []
  Failure Notification Method                         []
  Required Attributes                                 []
  Optional Attributes                                 []
  Description                                         []


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Notice that the *Processing Order* parameter is now set to **SFDG**. This tells PowerHA that all resources of the **SFFS** type are to be processed after **SFDG** resources. In other words, the *file systems* will be mounted only after the *diskgroups* are made available.

Next, we create the resources to activate the file systems. The process is the same as illustrated in Example 4-8 on page 78, except that this time the type of the resource must be **SFFS** and the resource name must be the entire *file system* name, as defined in the `/etc/filesystems` file.

In our test, we created the **SFFS** resource ***/test/havol01.*** Now, we are going to add the resources to the PowerHA resource group ***hagrp01***, as shown in Example 4-10. The resources cannot be added in the standard way that we usually do for native LVM volume groups and file systems. Instead, our new resources must be configured in the "**User Defined Resources**" field.

Note: The steps to create resource groups in PowerHA are not covered in this book. For more information, see *PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01.

Example 4-10 Creating the resources

```
          Change/Show All Resources and Attributes for a Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[MORE...28]                                           [Entry Fields]
```

```
  Raw Disk UUIDs/hdisks                                []                      +
  Disk Error Management?                                no                     +
  Fast Connect Services                                []                      +

  Primary Workload Manager Class                       []                      +
  Secondary Workload Manager Class                     []                      +

  Miscellaneous Data                                   []
  WPAR Name                                            []                      +
  User Defined Resources                               [testdg /test/havol01] +
  SVC PPRC Replicated Resources                        []                      +
  EMC SRDF(R) Replicated Resources                     []                      +
  DS8000 Global Mirror Replicated Resources                                    +
[MORE...3]

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

With all the resources configured, the resource group can now be started, stopped, and moved from one node to another. The Storage Foundation resources are now under PowerHA management.

4.15.4 Mixed volume manager environment

Systems are not expected to always be static. Instead, data growth is expected as the business grows.

It is important to have a strategy in mind for the moment at which your data needs to grow. Although the ideal scenario would be to have all data migrated before adding new disks to the system or expanding file systems, eventually, the need for additional space may appear before we are able to migrate all data.

When the time comes to add more data to the system, the decision about how to make it will be important. If Storage Foundation *diskgroups* are still used on the system, they can be expanded. Alternatively, if the new data requirements allow, new LVM volume groups can be created along with JFS2 file systems.

No matter what choices are made when data grows, at some point, the system may end up with a mix of different resource types.

5

Converting a local PowerHA Standard Edition cluster to a PowerHA Enterprise Edition cluster

In this chapter, we convert a local PowerHA Standard Edition (SE) cluster to a two-site disaster recovery cluster utilizing PowerHA Enterprise Edition (EE). It is always recommended to perform these steps in a test environment before ever performing in production.

This chapter covers the following topics:

- Test environment overview
- Install PowerHA Enterprise Edition software
- Delete existing cluster
- Creating a three node two-site GLVM linked cluster
  - Define linked cluster with sites
  - Configure GLVM
  - Create GMVG
  - Create resource group
  - Add GMVG into resource group
- Defining manual site split and merge policy
- Testing manual split option

The details of each step are well documented in previous PowerHA/EE IBM Redbooks publications. However, we provide a very specific example in the following sections.

© Copyright IBM Corp. 2015. All rights reserved.

5.1 Test environment overview

For our example, we start off with a local two-node cluster with a single IP network and shared storage as shown in Figure 5-1.

Though not shown, dual VIOS servers are utilized within each server. In our case, the storage is presented to the clients through VIOS via *shared storage pools*. There was no specific technical requirement for this but was requested of us since there is no other existing PowerHA documentation examples showing their use.

We also use Logical Volume Manager (LVM) mirroring across two logical sites. Our environment was not prototypical of what we would consider best practices for production environments. This is mainly because of physical resource limitations. The main idea was to demonstrate that potentially an existing cross-site LVM mirror configuration could be expanded to include Geographic Logical Volume Manager (GLVM) for a third copy. Also, GLVM was the most readily available option for us to demonstrate converting a local cluster to a dual site cluster as it does not require specialized storage to do so.

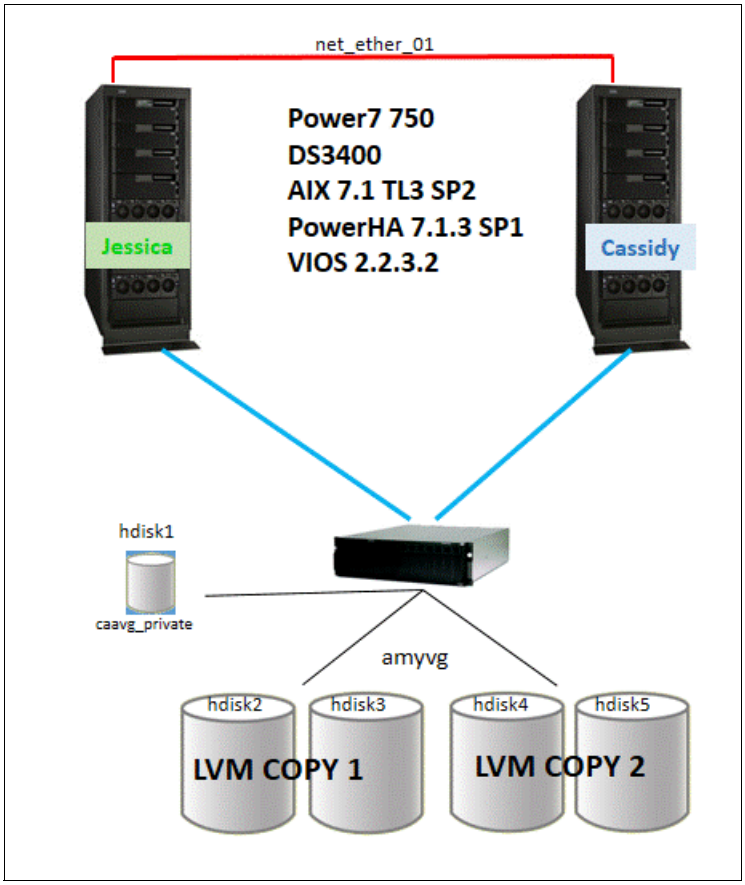

Figure 5-1 Test environment beginning configuration

The details of the cluster topology configuration are shown in Figure 5-2 on page 83. Since this is a PowerHA v7.1.3 cluster, we utilize *unicast* instead of the previously required *multicast* as the primary heartbeat mechanism.

```
[root] / # cltopinfo
Cluster Name:    PHASEtoEE
Cluster Type:    Stretched
Heartbeat Type:  Unicast
Repository Disk: hdisk1 (00f6f5d015a4310b)
Cluster Nodes:
        Site 1 (Dallas):
                Jessica
        Site 2 (FortWorth):
                Cassidy

There are 2 node(s) and 1 network(s) defined

NODE Cassidy:
        Network net_ether_01
                dallasserv      10.10.10.51
                cassidy         192.168.100.52

NODE Jessica:
        Network net_ether_01
                dallasserv      10.10.10.51
                jessica         192.168.100.51

Resource Group xsiteRG
        Startup Policy   Online On Home Node Only
        Fallover Policy  Fallover To Next Priority Node In The List
        Fallback Policy  Never Fallback
        Participating Nodes      Jessica Cassidy
        Service IP Label                 dallasserv
```

Figure 5-2 Test environment beginning topology

The cluster is set up as a two-node (*Jessica* and *Cassidy)* single resource group (*xsiteRG)* LVM mirrored hot-standby configuration. There is only one service IP (*dallasserv)* address and an application server (*banner)*. Though sites are currently defined, the member nodes assigned to each site will change during this test scenario

Ultimately, the cluster will change to include a third node (*Shanley)*, and it solely will reside in the second site *fortworth* utilizing GLVM for the data replication across sites. The second local node (*Cassidy)* will logically become part of the dallas site. No additional resource groups will be added into the configuration.

The cluster resources and resource group defined are shown in Figure 5-3 on page 84.

```
[root] / # clshowres

Resource Group Name                                 xsiteRG
Participating Node Name(s)                          Jessica Cassidy
Startup Policy                                      Online On Home Node Only
Fallover Policy                                     Fallover To Next Priority
Node In The List
Fallback Policy                                     Never Fallback
Site Relationship                                   ignore
Dynamic Node Priority
Service IP Label                                    dallasserv
Filesystems                                         ALL
Filesystems Consistency Check                       fsck
Filesystems Recovery Method                         sequential
Filesystems/Directories to be exported (NFSv2/NFSv3)
Filesystems/Directories to be exported (NFSv4)
Filesystems to be NFS mounted
Network For NFS Mount
Filesystem/Directory for NFSv4 Stable Storage
Volume Groups                                       amyvg
Concurrent Volume Groups
Use forced varyon for volume groups, if necessary   true
Disks
Raw Disks

Disk Error Management?                               no
GMVG Replicated Resources
GMD Replicated Resources
PPRC Replicated Resources
SVC PPRC Replicated Resources
EMC SRDF? Replicated Resources
TRUECOPY Replicated Resources
GENERIC XD Replicated Resources
Connections Services
Fast Connect Services
Shared Tape Resources
Application Servers                                 banner
Highly Available Communication Links
Primary Workload Manager Class
Secondary Workload Manager Class
Delayed Fallback Timer
Miscellaneous Data
Automatically Import Volume Groups                  false
Inactive Takeover
SSA Disk Fencing                                    false
Filesystems mounted before IP configured            true
WPAR Name
```

Figure 5-3 Beginning cluster resource group configuration

5.2 Install PowerHA Enterprise Edition software

In our scenario, only PowerHA Standard Edition is installed on the two local nodes. It is necessary to install PowerHA Enterprise Edition on both local nodes, along with the new remote node.

When installing, you can choose to install all, or only those filesets specific to your environment. The IBM PowerHA SystemMirror Enterprise Edition requires the installation and acceptance of license agreements for both the Standard Edition `cluster.license` fileset and the Enterprise Edition `cluster.xd.license` fileset as shown in Table 5-1, in order for the remainder of the filesets to install.

Table 5-1 PowerHA Enterprise Edition required license filesets

Required package	Filesets to install
Standard Edition license	cluster.license
Enterprise Edition license	cluster.xd.license

The base filesets in the Standard Edition are required to install the Enterprise Edition filesets. The Enterprise package levels must match those of the base runtime level (`cluster.es.server.rte`). Table 5-2 displays the itemized list of filesets for each of the integrated offerings.

Table 5-2 PowerHA Enterprise Edition - integrated offering solution filesets

Replication type	Fileset to install
ESS-Direct Management PPRC	cluster.es.pprc.rte cluster.es.pprc.cmds cluster.msg.en_US.pprc
ESS/DS6000/DS8000 Metro Mirror (DSCLI PPRC)	cluster.es.spprc.cmds cluster.es.spprc.rte cluster.es.cgpprc.cmds cluster.es.cgpprc.rte cluster.msg.en_US.cgpprc
SAN Volume Controller (SVC) Storwize V7000	cluster.es.svcpprc.cmds cluster.es.svcpprc.rte cluster.msg.en_US.svcpprc
XIV, DS8800 in-band and IBM HyperSwap®, DS8700/DS8800 Global Mirror	cluster.es.genxd.cmds cluster.es.genxd.rte cluster.msg.en_US.genxd
Geographic Logical Volume Mirroring	cluster.doc.en_US.glvm.pdf cluster.msg.en_US.glvm cluster.xd.glvm glvm.rpv.client glvm.rpv.man.en_US glvm.rpv.msg.en_US glvm.rpv.server glvm.rpv.util
EMC SRDF	cluster.es.sr.cmds cluster.es.sr.rte cluster.msg.en_US.sr

Replication type	Fileset to install
Hitachi TrueCopy/Universal Replicator HP Continuous Access	cluster.es.tc.cmds cluster.es.tc.rte cluster.msg.en_US.tc

In our scenario, we only needed the base, license, and glvm filesets as shown in Figure 5-4.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
geninstall -I "a -cgNqwXY -J"  -Z   -d . -f File 2>&1

File:
    I:cluster.xd.base              7.1.3.0
    I:cluster.xd.glvm              7.1.3.0
    I:cluster.xd.license           7.1.3.0
    I:glvm.rpv.client              7.1.3.0
    I:glvm.rpv.msg.en_US           7.1.3.0
    I:glvm.rpv.server              7.1.3.0
    I:glvm.rpv.util                7.1.3.0


[MORE...152]

F1=Help             F2=Refresh          F3=Cancel           F6=Command
F8=Image            F9=Shell            F10=Exit            /=Find
```

Figure 5-4 PowerHA Enterprise Edition installation

Upon completion, we then applied SP1 and ended up with the levels shown in Figure 5-5 on page 87.

If you are utilizing storage-specific replication, it may be required to install other additional software to accommodate it with PowerHA/EE. It is also possible that you will need additional IP network connectivity to the storage. For more information, consult the *Storage-based high availability and disaster recovery for PowerHA SystemMirror Enterprise Edition* guide at:

http://tinyurl.com/lh4nchs

A unique attribute in PowerHA v7, contributed by Cluster Aware AIX (CAA), is the cluster type definition. The types consist of *Standard (or Flat)* when no sites are used, and *Stretched and Linked* for when sites are utilized. Unfortunately, there currently is no way to change from a standard or stretched cluster to a linked cluster without deleting and re-creating the cluster. The primary reason for this is the underlying existing CAA cluster does not allow it.

```
  bos.cluster.rte            7.1.3.2    C     F    Cluster Aware AIX
  cluster.es.client.clcomd   7.1.3.1    C     F    Cluster Communication
  cluster.es.client.lib      7.1.3.1    C     F    PowerHA SystemMirror Client
  cluster.es.client.rte      7.1.3.1    C     F    PowerHA SystemMirror Client
  cluster.es.client.utils    7.1.3.0    C     F    PowerHA SystemMirror Client
  cluster.es.cspoc.cmds      7.1.3.1    C     F    CSPOC Commands
  cluster.es.cspoc.rte       7.1.3.1    C     F    CSPOC Runtime Commands
  cluster.es.migcheck        7.1.3.0    C     F    PowerHA SystemMirror Migration
  cluster.es.server.diag     7.1.3.1    C     F    Server Diags
  cluster.es.server.events   7.1.3.1    C     F    Server Events
  cluster.es.server.rte      7.1.3.1    C     F    Base Server Runtime
  cluster.es.server.testtool
  cluster.es.server.utils    7.1.3.1    C     F    Server Utilities
  cluster.license            7.1.3.0    C     F    PowerHA SystemMirror
  cluster.man.en_US.es.data  7.1.3.1    C     F    Man Pages - U.S. English
  cluster.msg.en_US.es.client
  cluster.msg.en_US.es.server
  cluster.xd.base            7.1.3.0    C     F    PowerHA SystemMirror
  cluster.xd.glvm            7.1.3.1    C     F    PowerHA SystemMirror
  cluster.xd.license         7.1.3.0    C     F    PowerHA SystemMirror
  glvm.rpv.client            7.1.3.1    C     F    Remote Physical Volume Client
  glvm.rpv.msg.en_US         7.1.3.0    C     F    RPV Messages - U.S. English
  glvm.rpv.server            7.1.3.1    C     F    Remote Physical Volume Server
  glvm.rpv.util              7.1.3.0    C     F    Geographic LVM Utilities
```

Figure 5-5 PowerHA/EE filesets with SP1 installed

5.3 Delete existing cluster

Deleting the existing cluster is an unfortunate necessary step as CAA does not support dynamically changing between stretched and linked clusters. This obviously cannot be done with an active cluster. We recommend making a snapshot of the cluster before deleting to be referenced later if needed.

To remove the existing cluster this, like most operations, can be accomplished either via **clmgr** or through SMIT. We will demonstrate both.

We first execute **smitty sysmirror** → **Cluster Nodes and Networks** → **Manage the Cluster** → **Remove the Cluster Definition**. We choose **yes** to remove the cluster definition across all nodes as shown in Figure 5-6. This will also remove the underlying CAA cluster.

```
                        Remove the Cluster Definition
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Cluster Name                                       PHASEtoEE
* Remove definition from all nodes                  [Yes]                  +

  NOTE: All user-configured cluster information
  WILL BE DELETED by this operation.
```

Figure 5-6 Delete cluster via SMIT

The clmgr equivalent of deleting the cluster is shown in Example 5-1.

Example 5-1 Delete cluster via clmgr

```
[jessica:root] / # clmgr delete cluster PHASEtoEE
One or more deletion operations have been detected.
Proceed with the deletion(s)? (y|n) y
Ensuring that the following nodes are offline: cassidy, jessica
Warning: cluster services are already offline on node "cassidy" (state is
         "ST_INIT"). Removing that node from the shutdown list.
Warning: cluster services are already offline on node "jessica" (state is
         "ST_INIT"). Removing that node from the shutdown list.
Attempting to delete node "cassidy" from the cluster...
Attempting to remove the CAA cluster from "jessica"...
Attempting to delete node "jessica" from the cluster...
```

5.4 Creating a three node two-site GLVM linked cluster

Before creating the cluster, ensure that all nodes meet the following bare minimum requirements:

- IP interfaces must be configured with addresses
- All IPs used for cluster must be defined in /etc/hosts on each node
- Hostname IP entries for all nodes must be in /etc/cluster/rhosts on each node
- clcomd must be active
- A free disk is available, with PVID, at each site for repository disk

For our scenario, we configured an additional interface and network to each of our existing cluster nodes, along with both interfaces and networks to the new remote node. The IP configuration of each system before adding them into the cluster configuration is shown in Example 5-2.

Also, both of our networks are accessible by all nodes in the cluster. This often may *not* be the case. More typically, the nodes may be attached to a LAN within a site and also another WAN between sites. Of course, it is imperative that site communications work properly, are of adequate bandwidth, and preferably multiple diversely routed physical links.

Example 5-2 IP interface configuration

```
[jessica:root] / # netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      ee.af.1.71.78.2    637436     0   600984     0     0
en0   1500  10.10.8     10.10.10.51        637436     0   600984     0     0
en0   1500  192.168.100 192.168.100.51     637436     0   600984     0     0
en1   1500  link#3      ee.af.1.71.78.3        36     0       11     0     0
en1   1500  192.168.148 192.168.150.51         36     0       11     0     0

[cassidy:root] /# netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      7a.40.c8.b3.15.2   663449     0   687192     0     0
en0   1500  192.168.100 192.168.100.52     663449     0   687192     0     0
en1   1500  link#3      7a.40.c8.b3.15.3       27     0       11     0     0
en1   1500  192.168.148 192.168.150.52         27     0       11     0     0
```

```
[shanley:root] /# netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      6e.8d.d0.21.d0.2    38633     0    21121     0     0
en0   1500  192.168.100 192.168.100.53      38633     0    21121     0     0
en1   1500  link#3      6e.8d.d0.21.d0.3       21     0       10     0     0
en1   1500  192.168.148 192.168.150.53         21     0       10     0     0
```

The contents of our /etc/hosts and /etc/cluster/rhosts entries are shown in Example 5-3, along with the clcomd status.

Example 5-3 Hosts and rhosts file contents and clcomd status

```
[jessica:root] / #cat /etc/hosts
192.168.100.21  phat01
192.168.100.40  nimres2
#net_ether_01
192.168.100.51 jessica
192.168.100.52 cassidy
192.168.100.53 shanley
#XD_data
192.168.150.51 jessica_xd
192.168.150.52 cassidy_xd
192.168.150.53 shanley_xd
#Site Service Addresses
10.10.10.51     dallasserv
10.10.10.52     ftwserv

[jessica:root] / # cat /etc/cluster/rhosts
192.168.100.51
192.168.100.52
192.168.100.53

[jessica:root] / #lssrc -s clcomd
Subsystem         Group            PID          Status
 clcomd           caa              5308598      active
```

Upon validating that the previous steps were completed, we can begin configuring the cluster.

5.4.1 Define linked cluster with sites

We begin by executing **smitty sysmirror** → **Cluster Nodes and Networks** → **Multi Site Cluster Deployment** → **Setup a Cluster, Nodes and Networks**. The options were configured as shown in Figure 5-7 on page 90.

```
               Setup Cluster, Sites, Nodes and Networks

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Cluster Name                                       [PHASEtoEE]

* Site 1 Name                                        [dallas]
* New Nodes (via selected communication paths)       [jessica cassidy]       +

* Site 2 Name                                        [fortworth]
* New Nodes (via selected communication paths)       [shanley]               +


  Cluster Type                                       [Linked Cluster]        +
```

Figure 5-7 Add a multi-site cluster via SMIT menu

Upon execution, you will see the discovery process for IP interfaces and disks run against the nodes. This will, if possible, automatically add the interfaces into the cluster topology. By default, the cluster software associates the interfaces with PowerHA networks of type *ether.* In order to change the default names and the type of a network, we used SMIT menus by executing **smitty sysmirror → Cluster Nodes and Networks → Manage Networks and Network Interfaces → Networks → Change/Show a Network**. The resulting network and interface definitions are shown in Example 5-4.

In all site configurations, it is common to have an *XD_ip* network defined. However it is mostly the same as type *ether* with only different default heartbeat parameters and subnet restrictions removed. You can have both ether networks within a site and XD_ip network across. The *XD_data* network is unique to GLVM and specifies which network is to be utilized for GLVM data replication traffic. Though it is not required to be a dedicated network, it is recommended.

In our scenario, we only have two networks and they do actually both span the sites, which may not be common overall. Because of this, we could have chosen to keep one network as type ether. However, we thought it was best to show the usage of an XD_ip instead.

Example 5-4 Network topology

```
[shanley:root] / # cllsif
Adapter              Type       Network    Net Type   Attribute  Node       IP Address
cassidy_xd           boot       GLVMnet    XD_data    public     cassidy    192.168.150.52
cassidy              boot       prodnet    XD_ip      public     cassidy    192.168.100.52
dallasserv           service    prodnet    XD_ip      public     cassidy    10.10.10.51
ftwserv              service    prodnet    XD_ip      public     cassidy    10.10.10.52
jessica_xd           boot       GLVMnet    XD_data    public     jessica    192.168.150.51
jessica              boot       prodnet    XD_ip      public     jessica    192.168.100.51
dallasserv           service    prodnet    XD_ip      public     jessica    10.10.10.51
ftwserv              service    prodnet    XD_ip      public     jessica    10.10.10.52
shanley_xd           boot       GLVMnet    XD_data    public     shanley    192.168.150.53
shanley              boot       prodnet    XD_ip      public     shanley    192.168.100.53
dallasserv           service    prodnet    XD_ip      public     shanley    10.10.10.51
ftwserv              service    prodnet    XD_ip      public     shanley    10.10.10.52
```

In the next step, we defined the repository disks for the sites. We defined both of them in a single SMIT panel by executing **smitty sysmirror → Cluster Nodes and Networks → Multi Site Cluster Deployment → Define Repository Disk and Cluster IP Address**. We selected the candidate disks for the CAA repository in each site, hdisk1 for dallas, and hdisk2 in fortworth. We used the default heartbeat mechanism of unicast as shown in Figure 5-8.

```
                Multi Site with Linked Clusters Configuration

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Cluster Name                                        PHASEtoEE
* Heartbeat Mechanism                                 Unicast                +

* Site Name                                           dallas
* Repository Disk                                    [(00f6f5d015a4310b)]    +
  Site Multicast Address                             []
      (used only for multicast heartbeat)

* Site Name                                           fortworth
* Repository Disk                                    [(00f61ab216646614)]    +
  Site Multicast Address                             []
      (used only for multicast heartbeat)
```

Figure 5-8 Defining the repository disks to each site

Note: When selecting the CAA repository disk, a PVID is automatically assigned for that disk if it has not a PVID defined already. The SMIT panel gets populated with the PVIDs of the candidate repository disks, making the cluster configuration independent of hdisk device numbers or names.

In the next step, we perform the cluster verification and synchronization by executing **smitty sysmirror → Cluster Nodes and Networks → Verify and Synchronize Cluster Configuration**. Though not necessary at this exact stage, it is recommended to do so as it will also create the CAA cluster. That way, if any problems are encountered they can be addressed early on in the configuration process.

The PowerHA cluster configuration that we have defined so far is shown in the **cltopinfo** command output in Example 5-5. Also, the specifics of the CAA cluster are shown in Example 5-6 on page 92.

Example 5-5 PowerHA cluster topology

```
[jessica:root] / # cltopinfo
Cluster Name:    PHASEtoEE
Cluster Type:    Linked
Heartbeat Type:  Unicast
Repository Disks:
        Site 1 (dallas@jessica): hdisk1
        Site 2 (fortworth@shanley): hdisk2
Cluster Nodes:
        Site 1 (dallas):
                jessica
```

```
                cassidy
        Site 2 (fortworth):
                shanley

There are 3 node(s) and 2 network(s) defined

NODE cassidy:
        Network GLVMnet
                cassidy_xd      192.168.150.52
        Network prodnet
                cassidy 192.168.100.52

NODE jessica:
        Network GLVMnet
                jessica_xd      192.168.150.51
        Network prodnet
                jessica 192.168.100.51

NODE shanley:
        Network GLVMnet
                shanley_xd      192.168.150.53
        Network prodnet
                shanley 192.168.100.53

No resource groups defined
```

Example 5-6 shows the CAA cluster configuration.

Example 5-6 CAA cluster configuration

```
[jessica:root] / # lscluster -m
Calling node query for all nodes...
Node query number of nodes examined: 3

        Node name: jessica
        Cluster shorthand id for node: 1
        UUID for node: 9837b6c4-e292-11e3-ac3a-eeaf01717802
        State of node: UP  NODE_LOCAL
        Smoothed rtt to node: 0
        Mean Deviation in network rtt to node: 0
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        PHASEtoEE          0            9846e270-e292-11e3-ac3a-eeaf01717802
        SITE NAME          SHID         UUID
        dallas             1            9837b55c-e292-11e3-ac3a-eeaf01717802

        Points of contact for node: 0
---------------------------------------------------------------------------
        Node name: shanley
        Cluster shorthand id for node: 2
        UUID for node: b599dea4-e292-11e3-854b-eeaf01717802
        State of node: UP
        Smoothed rtt to node: 14
        Mean Deviation in network rtt to node: 11
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        PHASEtoEE          0            9846e270-e292-11e3-ac3a-eeaf01717802
        SITE NAME          SHID         UUID
```

```
        fortworth          2            b59a5a28-e292-11e3-854b-eeaf01717802

        Points of contact for node: 1
        -------------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        -------------------------------------------------------------------------
        tcpsock->02   UP     IPv4         none   192.168.150.51->192.168.150.53

------------------------------------------------------------------------------
        Node name: cassidy
        Cluster shorthand id for node: 5
        UUID for node: 9837b778-e292-11e3-ac3a-eeaf01717802
        State of node: UP
        Smoothed rtt to node: 7
        Mean Deviation in network rtt to node: 3
        Number of clusters node is a member in: 1
        CLUSTER NAME       SHID         UUID
        PHASEtoEE          0            9846e270-e292-11e3-ac3a-eeaf01717802
        SITE NAME          SHID         UUID
        dallas             1            9837b55c-e292-11e3-ac3a-eeaf01717802

        Points of contact for node: 2
        -------------------------------------------------------------------------
        Interface     State  Protocol    Status     SRC_IP->DST_IP
        -------------------------------------------------------------------------
        tcpsock->05   UP     IPv4         none   192.168.100.51->192.168.100.52
        tcpsock->05   UP     IPv4         none   192.168.150.51->192.168.150.52
```

The exact order of most of the remaining steps are a bit flexible. In our case, we already have a data volume group, *amyvg,* defined with file systems, *rachaelfs,* and *meganfs,* so we will skip those steps. We chose to show configuring the GLVM option first especially when it offers an assistant that minimizes the total number of steps required.

5.4.2 Configure GLVM

The topic of configuring GLVM has been covered in numerous sources including the base PowerHA publications and Redbooks publications. This section is not intended to cover all the detailed options that are available but to focus on one specific implementation as covered in our scenario.

There is some flexibility in the order of the following steps but these we felt are the best order to perform them for consistent results. In PowerHA configurations, it is common, and often required, to have each node in the cluster be both an *rpvserver* and *rpvclient.* The rpvserver presents the remote disks to the clients. The clients of course get new disk definitions locally as though it is just another disk. However, they are only accessed via the rpvserver over the IP network.

Add rpvservers to dallas site

For our scenario, we need the four local dallas site disks (*hdisk2-5*) to be presented to the fortworth site. We also need the two disks from the fortworth site (*hdisk3-4*) to be presented to both nodes at the dallas site. To accomplish this, we must first define each node to an RPV server site. Then, we must make each disk within each site an rpvserver.

Note: When configuring GLVM, it is expecting the disks to not be existing volume group members. If so, they will not show up in the menu pick lists. If you want to use an existing volume group as we have, you must export the volume group via **exportvg**.

We start off on node jessica by executing **smitty glvm_utils** → **Remote Physical Volume Servers** → **Define / Change / Show Remote Physical Volume Server Site Name**. We then enter site name dallas as shown in Figure 5-9. This step is then repeated for node cassidy.

```
        Define / Change / Show Remote Physical Volume Server Site Name

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Remote Physical Volume Server Site Name            [dallas]
```

Figure 5-9 Define RPV server site to dallas nodes

While still on node jessica, we execute **smitty glvm_utils** → **Remote Physical Volume Servers** → **Add Remote Physical Volume Servers**. We then chose our four disks as shown in Figure 5-10 on page 95.

After selecting the disks and pressing Enter, we are presented with the final rpvserver menu as shown in Figure 5-11 on page 95. We manually enter the IP address that corresponds to the XD_data network interface on node shanley in the fortworth site. We also chose the default to not allow the devices to configure automatically on system restart. Much like volume group activation, we want PowerHA to control this action. Upon pressing Enter to complete the rpvserver configuration, we see rpvserver devices present as shown in Figure 5-12 on page 96. We repeat these steps on the second node, cassidy, in the dallas site with one exception. We also chose the option to not start the devices immediately. This will create the rpvservers and put them in the defined state instead of available.

```
                        Remote Physical Volume Servers

Move cursor to desired item and press Enter.

  Remote Physical Volume Server Site Name Configuration
  +--------------------------------------------------------------------------+
  |                       Physical Volume Identifiers                        |
  |                                                                          |
  | Move cursor to desired item and press F7.                                |
  |     ONE OR MORE items can be selected.                                   |
  | Press Enter AFTER making all selections.                                 |
  |                                                                          |
  |   # Physical Volume        Physical Volume Identifier                    |
  |   # ------------------------------------------------                    |
  | > hdisk2           00f6f5d01660fbd1                                      |
  | > hdisk3           00f6f5d015a44307                                      |
  | > hdisk4           00f6f5d0166106fa                                      |
  | > hdisk5           00f6f5d0166114f3                                      |
  |   hdisk6           00f6f5d029906df4                                      |
  |                                                                          |
  | F1=Help                 F2=Refresh              F3=Cancel                |
  | F7=Select               F8=Image                F10=Exit                 |
F1| Enter=Do                /=Find                  n=Find Next              |
F9+--------------------------------------------------------------------------+
```

Figure 5-10 Rpvserver disk selection on node jessica in dallas

```
                      Add Remote Physical Volume Servers

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Physical Volume Identifiers                         00f6f5d01660fbd1 00f6>
* Remote Physical Volume Client Internet Address     [192.168.150.53]        +
  Configure Automatically at System Restart?         [no]                    +
  Start New Devices Immediately?                     [yes]                    +
```

Figure 5-11 Rpvserver final SMIT window on node jessica

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

rpvserver0 Available
rpvserver1 Available
rpvserver2 Available
rpvserver3 Available
```

Figure 5-12 Rpvserver devices are available on node jessica

Add rpvservers to fortworth site

On node shanley, we begin by executing **smitty glvm_utils** → **Remote Physical Volume Servers** → **Define / Change / Show Remote Physical Volume Server Site Name** as shown in Figure 5-13. We then enter site name dallas as shown in Figure 5-9 on page 94. This step is then repeated for node cassidy.

```
        Define / Change / Show Remote Physical Volume Server Site Name

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Remote Physical Volume Server Site Name           [fortworth]
```

Figure 5-13 Define RPV server site to fortworth node

We next execute **smitty glvm_utils** → **Remote Physical Volume Servers** → **Add Remote Physical Volume Servers**. We then chose only the two local disks (hdisk3-4).

After selecting the disks and pressing Enter, we are presented with the final rpvserver menu as shown in Figure 5-14. We manually enter the IP address that corresponds to the XD_data network interface on node jessica in the dallas site. We also chose the default to not allow the devices to configure automatically on system restart. Much like volume group activation, we want PowerHA to control this action.

```
                        Add Remote Physical Volume Servers

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Physical Volume Identifiers                        00f61ab21664556b 00f6>
* Remote Physical Volume Client Internet Address    [192.168.150.51]        +
  Configure Automatically at System Restart?        [no]                    +
  Start New Devices Immediately?                    [yes]
```

Figure 5-14 Rpvserver final SMIT window on node shanley in fortworth site

Upon pressing Enter to complete the rpvserver configuration, we see rpvserver devices present as shown in Figure 5-15. Since there is only one node in the fortworth site, no additional rpvserver definitions are required at this time. Though later, we change the definition temporarily to present the disks to node cassidy.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

rpvserver0 Available
rpvserver1 Available
```

Figure 5-15 Rpvserver devices are available on node shanley

The current status of the rpvservers on each node can be seen in Example 5-7.

Example 5-7 Current rpvserver states on all nodes

```
[jessica:root] / # clcmd lsdev -t rpvstype
------------------------------
NODE cassidy
------------------------------
rpvserver0 Defined  Remote Physical Volume Server
rpvserver1 Defined  Remote Physical Volume Server
rpvserver2 Defined  Remote Physical Volume Server
rpvserver3 Defined  Remote Physical Volume Server
------------------------------
NODE shanley
------------------------------
rpvserver0 Available  Remote Physical Volume Server
rpvserver1 Available  Remote Physical Volume Server
------------------------------
NODE jessica
------------------------------
rpvserver0 Available  Remote Physical Volume Server
rpvserver1 Available  Remote Physical Volume Server
rpvserver2 Available  Remote Physical Volume Server
rpvserver3 Available  Remote Physical Volume Server
```

Adding rpvclients to fortworth site

On node shanley, we execute **smitty glvm_utils** → **Remote Physical Volume Clients** → **Add Remote Physical Volume Clients**. In the first menu regarding IPv6, we keep the default of *no* and press Enter to continue. We then manually enter the IP address of the XD_data network interface on node jessica. We then get another menu to choose the rpv volume local address. In this case, we chose the XD_data IP of node shanley as shown in Figure 5-16 on page 98.

```
                        Add Remote Physical Volume Clients

Type or select a value for the entry field.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* Remote Physical Volume Server Internet Address     [192.168.150.51]        +

  +--------------------------------------------------------------------------+
  |              Remote Physical Volume Local Internet Address               |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   192.168.100.53   shanley                                               |
  |   192.168.150.53   shanley_xd                                            |
  |                                                                          |
  | F1=Help                 F2=Refresh              F3=Cancel                |
F1| F8=Image                F10=Exit                Enter=Do                 |
F5| /=Find                  n=Find Next                                      |
  +--------------------------------------------------------------------------+
```

Figure 5-16 Rpvclient initial SMIT menu on node shanley

We then get a pop-up menu to choose which disks from node jessica we want to set up as clients. In our case, we choose all four of the disks presented. Upon pressing Enter, we are presented with the final rpvclient SMIT menu as shown in Figure 5-17.

```
                        Add Remote Physical Volume Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Remote Physical Volume Server Internet Address      192.168.150.51
  Remote Physical Volume Local Internet Address       192.168.150.53
  Physical Volume Identifiers                         00f6f5d015a4430700000>
  I/O Timeout Interval (Seconds)                     [180]                  #
  Start New Devices Immediately?                     [yes]                 +
```

Figure 5-17 Rpvclient final SMIT menu on node shanley

This step results in four new hdisk definitions presented locally onto node shanley as seen in Figure 5-18 on page 99.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

hdisk5 Available
hdisk6 Available
hdisk7 Available
hdisk8 Available
```

Figure 5-18 New hdisks on node shanley

Adding rpvclients to dallas site

On node jessica, we repeat the previous rpvclient steps. We execute **smitty glvm_utils** → **Remote Physical Volume Clients** → **Add Remote Physical Volume Clients**. In the first menu regarding IPv6, we keep the default of *no* and press Enter to continue. We then manually enter the IP address of the XD_data network interface on node shanley. We then get another menu to choose the rpv volume local address. In this case, we chose the XD_data IP of node shanley.

We then get a pop-up menu to choose which disks from node shanley we want to utilize as clients. In our case, we choose both of the disks (hdisk3-4) presented. Upon pressing Enter, we are presented with the final rpvclient SMIT menu as shown in Figure 5-19.

```
                      Add Remote Physical Volume Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Remote Physical Volume Server Internet Address      192.168.150.53
  Remote Physical Volume Local Internet Address       192.168.150.51
  Physical Volume Identifiers                         00f61ab21664556b00000>
  I/O Timeout Interval (Seconds)                     [180]                  #
  Start New Devices Immediately?                     [yes]                 +
```

Figure 5-19 Rpvclient final SMIT menu on node jessica

This step results in two new hdisk definitions presented locally onto node jessica as seen in Figure 5-20 on page 100.

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

hdisk7 Available
hdisk8 Available
```

Figure 5-20 New hdisks on node jessica

Next, we need to repeat these steps on node cassidy. However, before doing so we have to change the rpvserver configuration on node shanley to change the rpvclient IP address to be the XD_data IP network of node cassidy. This is done easily on node shanley by executing **smitty glvm_utils** → **Remote Physical Volume Servers** → **Change Multiple Remote Physical Volume Servers** and manually typing in the IP address as shown Figure 5-21.

```
                Change Multiple Remote Physical Volume Servers

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Remote Physical Volume Servers                     rpvserver0 rpvserver1
  Remote Physical Volume Client Internet Address     [192.168.150.52]       +
  Configure Automatically at System Restart?         [Do not change]
```

Figure 5-21 Change Rpvservers on node shanley to point to node cassidy

Now we repeat the rpvclient steps on node cassidy that we previously executed on note jessica. We execute **smitty glvm_utils** → **Remote Physical Volume Clients** → **Add Remote Physical Volume Clients**. In the first menu regarding IPv6, we keep the default of *no* and press Enter to continue. We then manually enter the IP address of the XD_data network interface on node shanley. We then get another menu to choose the rpv volume local address. In this case, we chose the XD_data IP of node shanley.

We then get a pop-up menu to choose which disks from node shanley we want to set up as clients. In our case, we choose both of the disks (hdisk3-4) presented. Upon pressing Enter, we are presented with the final rpvclient SMIT menu as shown in Figure 5-22 on page 101. This step will result in two new hdisk definitions presented locally onto node cassidy as seen in Figure 5-23 on page 101.

Upon completion, we changed the rpvservers on node shanley back to the XD_data IP address on node jessica as the client. This is because in our PowerHA cluster, jessica will normally be the primary production system. If a fallover occurs, both the rpvserver and rpvclient definitions and status are automatically controlled by PowerHA.

```
                        Add Remote Physical Volume Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Remote Physical Volume Server Internet Address      192.168.150.53
  Remote Physical Volume Local Internet Address       192.168.150.52
  Physical Volume Identifiers                         00f61ab21664556b00000>
  I/O Timeout Interval (Seconds)                     [180]                  #
  Start New Devices Immediately?                     [yes]                 +
```

Figure 5-22 Rpvclient final SMIT menu on node cassidy

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

hdisk6 Available
hdisk7 Available
```

Figure 5-23 New hdisks on node cassidy

All the disk and rpvclient definitions as shown Example 5-8.

Example 5-8 Hdisks and rpvclients

```
[jessica:root] / # clcmd lsdev -Cc disk
------------------------------
NODE cassidy
------------------------------
hdisk0 Available  Virtual SCSI Disk Drive
hdisk1 Available  Virtual SCSI Disk Drive
hdisk2 Available  Virtual SCSI Disk Drive
hdisk3 Available  Virtual SCSI Disk Drive
hdisk4 Available  Virtual SCSI Disk Drive
hdisk5 Available  Virtual SCSI Disk Drive
hdisk6 Available  Remote Physical Volume Client
hdisk7 Available  Remote Physical Volume Client
------------------------------
NODE shanley
------------------------------
hdisk0 Defined    Virtual SCSI Disk Drive
hdisk1 Available  Virtual SCSI Disk Drive
hdisk2 Available  Virtual SCSI Disk Drive
hdisk3 Available  Virtual SCSI Disk Drive
hdisk4 Available  Virtual SCSI Disk Drive
hdisk5 Available  Remote Physical Volume Client
hdisk6 Available  Remote Physical Volume Client
hdisk7 Available  Remote Physical Volume Client
```

```
hdisk8 Available  Remote Physical Volume Client
-------------------------------
NODE jessica
-------------------------------
hdisk0 Available  Virtual SCSI Disk Drive
hdisk1 Available  Virtual SCSI Disk Drive
hdisk2 Available  Virtual SCSI Disk Drive
hdisk3 Available  Virtual SCSI Disk Drive
hdisk4 Available  Virtual SCSI Disk Drive
hdisk5 Available  Virtual SCSI Disk Drive
hdisk6 Available  Virtual SCSI Disk Drive
hdisk7 Available  Remote Physical Volume Client
hdisk8 Available  Remote Physical Volume Client
```

5.4.3 Create GMVG

In our scenario, we already have an existing LVM mirrored scalable volume group, amyvg, with two copies utilizing mirror pools. We previously exported the volume group in order to allow the disks to show up in the GLVM pick lists. We begin by reimporting the volume group and varying it on node jessica in order to create the third copy and third mirror pool. Example 5-9 shows our current LV copy maps and mirror pools.

Example 5-9 Existing mirror pools and LVM copies

```
[jessica:root] / # lsmp -A amyvg
VOLUME GROUP:       amyvg               Mirror Pool Super Strict: yes

MIRROR POOL:        primp               Mirroring Mode:           SYNC
MIRROR POOL:        secmp               Mirroring Mode:           SYNC

[jessica:root] / # lsvg -P amyvg
Physical Volume   Mirror Pool
hdisk3            primp
hdisk2            primp
hdisk4            secmp
hdisk5            secmp

[jessica:root] / # lslv -m rachaellv
rachaellv:/rachaelfs
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0017 hdisk3            0097 hdisk5
0002  0097 hdisk2            0097 hdisk4
0003  0018 hdisk3            0098 hdisk5
0004  0098 hdisk2            0098 hdisk4
0005  0019 hdisk3            0099 hdisk5
0006  0099 hdisk2            0099 hdisk4
0007  0020 hdisk3            0100 hdisk5
0008  0100 hdisk2            0100 hdisk4
0009  0021 hdisk3            0101 hdisk5
0010  0101 hdisk2            0101 hdisk4
0011  0032 hdisk3            0113 hdisk4
0012  0113 hdisk2            0112 hdisk5
0013  0033 hdisk3            0114 hdisk4
0014  0114 hdisk2            0113 hdisk5
0015  0034 hdisk3            0115 hdisk4
```

```
0016  0115 hdisk2            0114 hdisk5
0017  0035 hdisk3            0116 hdisk4
0018  0116 hdisk2            0115 hdisk5
0019  0036 hdisk3            0117 hdisk4
0020  0117 hdisk2            0116 hdisk5

[jessica:root] / # lslv -m meganlv
meganlv:/meganfs
LP    PP1  PV1               PP2  PV2               PP3  PV3
0001  0022 hdisk3            0102 hdisk5
0002  0102 hdisk2            0102 hdisk4
0003  0023 hdisk3            0103 hdisk5
0004  0103 hdisk2            0103 hdisk4
0005  0024 hdisk3            0104 hdisk5
0006  0104 hdisk2            0104 hdisk4
0007  0025 hdisk3            0105 hdisk5
0008  0105 hdisk2            0105 hdisk4
0009  0026 hdisk3            0106 hdisk5
0010  0106 hdisk2            0106 hdisk4
0011  0027 hdisk3            0108 hdisk4
0012  0108 hdisk2            0107 hdisk5
0013  0028 hdisk3            0109 hdisk4
0014  0109 hdisk2            0108 hdisk5
0015  0029 hdisk3            0110 hdisk4
0016  0110 hdisk2            0109 hdisk5
0017  0030 hdisk3            0111 hdisk4
0018  0111 hdisk2            0110 hdisk5
0019  0031 hdisk3            0112 hdisk4
0020  0112 hdisk2            0111 hdisk5
```

Create GLVM copy

To create the GLVM copy also involves utilizing **`smitty glvm_utils`**. Upon execution, we choose **`Geographically Mirrored Logical Volumes`** → **`Add a Remote Site Mirror Copy to a Logical Volume`**. We are then presented with a pop-up menu with a list of logical volumes to choose from. In our first pass we choose *meganlv*. We next choose the remote site, fortworth, and then are presented with a remote candidate disk list with which we can create the mirror. In our case, it is hdisk7 and hdisk8. We chose both disks and are presented with the final SMIT menu as shown in Figure 5-24 on page 104.

In our environment, since we already have two copies we are creating a third copy. So it is necessary to set the option of *NEW TOTAL number of logical partition copies* to “3”. We also chose not to synchronize the copy of data at this time. We will synchronize all at the same time in a later step.

```
              Add a Remote Site Mirror Copy to a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* LOGICAL VOLUME name                                 meganlv
* NEW TOTAL number of logical partition copies        3                    +
* REMOTE PHYSICAL VOLUME Name                         hdisk7 hdisk8
  POSITION on physical volume                         outer_middle         +
  RANGE of physical volumes                           maximum              +
  Allocate each logical partition copy on a SEPARATE  superstrict
   physical volume?
  SYNCHRONIZE the data in the new logical partition   no                   +
  copies?
```

Figure 5-24 Create remote mirrored logical volume

We then repeat this step for both remaining logical volumes *rachaellv* and *shfloglv.* After completion, the third copy is stale as shown in Example 5-10. To synchronize, we simply execute **syncvg -v amyvg**.

Example 5-10 Mirrors stale after creation

```
[jessica:root] / # lsvg -l amyvg
amyvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
rachaellv           jfs2       20      60      6    closed/stale  /rachaelfs
meganlv             jfs2       20      60      6    closed/stale  /meganfs
shfloglv            jfs2log    1       3       3    closed/stale  N/A
```

Upon completing, the mirrors are now synchronized as shown Example 5-11.

Example 5-11 Mirrors in sync

```
[jessica:root] / # lsvg -l amyvg
amyvg:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
rachaellv           jfs2       20      60      6    closed/syncd  /rachaelfs
meganlv             jfs2       20      60      6    closed/syncd  /meganfs
shfloglv            jfs2log    1       3       3    closed/syncd  N/A
```

Note: GMVGs cannot be configured through C-SPOC. So create on one node, then varyoff and import on the next node. We show later after adding a mirror pool how to accomplish it.

Add a mirror pool

We need to add the rpvclient disks, hdisk7 and hdisk8, into a new mirror pool. We must first add the drives into amyvg via **extendvg**. Then, we use the **chpv** command to create and add them to a mirror pool as shown in Example 5-12.

Example 5-12 Create new mirror pool

```
[jessica:root] / # chpv -p thirdmp hdisk7 hdisk8
```

We verify that there are now three mirror pools by executing the **lsmp** and **lsvg** commands as shown in Example 5-13.

Example 5-13 All mirror pools

```
[jessica:root] / # lsmp -A amyvg
VOLUME GROUP:       amyvg               Mirror Pool Super Strict: yes

MIRROR POOL:        primp               Mirroring Mode:          SYNC
MIRROR POOL:        secmp               Mirroring Mode:          SYNC
MIRROR POOL:        thirdmp             Mirroring Mode:          SYNC
[jessica:root] / # lsvg -P amyvg
Physical Volume   Mirror Pool
hdisk3            primp
hdisk2            primp
hdisk4            secmp
hdisk5            secmp
hdisk7            thirdmp
hdisk8            thirdmp
```

Importing GMVG to each node

To this point, the RPV servers have been available on the remote node, shanley, and the RPV clients available on the local node, jessica. To import the GMVGs to the other local node, cassidy, the RPV clients need to be available to it. The RPV server must also be changed in order to make the RPV clients available on node cassidy.

This is done as follows:

1. Varyoff the volume groups on the local node, jessica.
2. Make the RPV clients defined on the local node, jessica.
3. Change the RPV servers defined on the remote node, shanley, to client cassidy.
4. Make the RPV clients available on the local node, cassidy.
5. Import the volume group to local node cassidy.

On the local node, jessica, execute:

- **varyoffvg amyvg**
- **rmdev -l hdisk7**
- **rmdev -l hdisk8**

This puts both disks in the defined state on node jessica. Next, we need to change the client address on the RPV servers on node shanley. This can be done in SMIT as we did previously in Figure 5-21 on page 100. However, we will perform it from the command line this time.

On the remote node shanley, execute:

- **chdev -l rpvserver0 -a client_address=192.168.150.52**
- **chdev -l rpvserver1 -a client_address=192.168.150.52**

On the local cassidy, the disks were still in available state. If they were not we could do so by executing:

- **mkdev -l hdisk7**
- **mkdev -l hdisk8**

At this point, we recommend to verify that disk access is indeed possible from node cassidy. This can be done by executing:

- **lquerypv -h /dev/hdisk6**
- **lquerypv -h /dev/hdisk7**

Upon positive confirmation of disk access, we can now import the volume group as follows:

- **importvg -y amyvg hdisk7**

The results of the command execution and the disk listing is shown in Example 5-14.

Example 5-14 GMVG amyvg imported on second local node cassidy

```
[cassidy:root] / # importvg -y amyvg hdisk7
amyvg
0516-783 importvg: This imported volume group is concurrent capable.
        Therefore, the volume group must be varied on manually.
[cassidy:root] / # lspv
hdisk0          00f70c99013e28ca                    rootvg          active
hdisk1          00f6f5d015a4310b                    caavg_private   active
hdisk2          00f6f5d015a44307                    amyvg
hdisk3          00f6f5d01660fbd1                    amyvg
hdisk4          00f6f5d0166106fa                    amyvg
hdisk5          00f6f5d0166114f3                    amyvg
hdisk6          00f61ab21664556b                    amyvg
hdisk7          00f61ab21664742e                    amyvg
```

We put the RPV clients, hdisk6 and hdisk7, back in the defined state on node cassidy by executing:

- **rmdev -l hdisk6**
- **rmdev -l hdisk7**

Next, we need to import the GMVG amyvg to the remote node shanley. Currently, all four RPV servers are available on node jessica and have node shanley set up as their client as shown in Example 5-15.

Example 5-15 RPVserver status on jessica

```
[jessica:root] / # lsdev -t rpvstype
rpvserver0 Available  Remote Physical Volume Server
rpvserver1 Available  Remote Physical Volume Server
rpvserver2 Available  Remote Physical Volume Server
rpvserver3 Available  Remote Physical Volume Server

[jessica:root] / # lsattr -El rpvserver0
auto_online n                                 Configure at System Boot   True
client_addr 192.168.150.53                    Client IP Address          True
rpvs_pvid   00f6f5d01660fbd100000000000000000 Physical Volume Identifier True

[jessica:root] / # lsattr -El rpvserver1
auto_online n                                 Configure at System Boot   True
client_addr 192.168.150.53                    Client IP Address          True
rpvs_pvid   00f6f5d015a4430700000000000000000 Physical Volume Identifier True

[jessica:root] / # lsattr -El rpvserver2
auto_online n                                 Configure at System Boot   True
```

```
client_addr 192.168.150.53                    Client IP Address          True
rpvs_pvid   00f6f5d0166106fa0000000000000000 Physical Volume Identifier True

[jessica:root] / # lsattr -El rpvserver3
auto_online n                                 Configure at System Boot   True
client_addr 192.168.150.53                    Client IP Address          True
rpvs_pvid   00f6f5d0166114f30000000000000000 Physical Volume Identifier True
```

The RPV clients are also still available on node shanley as shown in Example 5-16.

Example 5-16 RPV client status on shanley

```
[shanley:root] /# lsdev -t rpvclient
hdisk5 Available  Remote Physical Volume Client
hdisk6 Available  Remote Physical Volume Client
hdisk7 Available  Remote Physical Volume Client
hdisk8 Available  Remote Physical Volume Client
```

Just like before, it is recommended to verify that disk access is indeed possibly on node cassidy. This can be done by executing:

- **lquerypv -h /dev/hdisk5**
- **lquerypv -h /dev/hdisk6**
- **lquerypv -h /dev/hdisk7**
- **lquerypv -h /dev/hdisk8**

Upon positive confirmation of disk access, we can now import the volume group as follows:

- **importvg -y amyvg hdisk8**

The results of the command execution and the disk listing is shown in Example 5-17.

Example 5-17 GMVG amyvg imported on remote node shanley

```
[shanley:root] / # importvg -y amyvg hdisk8
amyvg
0516-783 importvg: This imported volume group is concurrent capable.
        Therefore, the volume group must be varied on manually.
[shanley:root] / # lspv
hdisk1          00f61ab215ad00d4                    rootvg          active
hdisk2          00f61ab216646614                    caavg_private   active
hdisk3          00f61ab21664556b                    amyvg
hdisk4          00f61ab21664742e                    amyvg
hdisk5          00f6f5d015a44307                    amyvg
hdisk6          00f6f5d01660fbd1                    amyvg
hdisk7          00f6f5d0166106fa                    amyvg
hdisk8          00f6f5d0166114f3                    amyvg
```

Next, we put all RPV clients and servers in the defined state via the **rmdev -l** command as shown previously. The status of all RPV servers and clients is shown in Example 5-18.

Example 5-18 RPV server and client status on all nodes

```
[jessica:root] / # clcmd lsdev -t rpvstype
------------------------------
NODE cassidy
------------------------------
rpvserver0 Defined  Remote Physical Volume Server
```

```
rpvserver1 Defined  Remote Physical Volume Server
rpvserver2 Defined  Remote Physical Volume Server
rpvserver3 Defined  Remote Physical Volume Server

------------------------------
NODE shanley
------------------------------
rpvserver0 Defined  Remote Physical Volume Server
rpvserver1 Defined  Remote Physical Volume Server

------------------------------
NODE jessica
------------------------------
rpvserver0 Defined  Remote Physical Volume Server
rpvserver1 Defined  Remote Physical Volume Server
rpvserver2 Defined  Remote Physical Volume Server
rpvserver3 Defined  Remote Physical Volume Server

[jessica:root] / # clcmd lsdev -t rpvclient
------------------------------
NODE cassidy
------------------------------
hdisk6 Defined  Remote Physical Volume Client
hdisk7 Defined  Remote Physical Volume Client

------------------------------
NODE shanley
------------------------------
hdisk5 Defined  Remote Physical Volume Client
hdisk6 Defined  Remote Physical Volume Client
hdisk7 Defined  Remote Physical Volume Client
hdisk8 Defined  Remote Physical Volume Client
------------------------------
NODE jessica
------------------------------
hdisk7 Defined  Remote Physical Volume Client
hdisk8 Defined  Remote Physical Volume Client
```

5.4.4 Create resource group

Before creating the resource group, we re-created our application server, *banner*, along with two service IPs, *dallasserv* and *ftwserv*, and assigned them to their respective sites.

To create the resource group, execute **smitty sysmirror** → **Cluster Applications and Resources** → **Resource Groups** → **Add a Resource Group**. We are presented with, and fill out the fields to, the SMIT menu as shown in Figure 5-25 on page 109.

```
                        Add a Resource Group (extended)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Resource Group Name                                [xsiteGLVMRG]

  Inter-Site Management Policy                       [Online on Primary Site] +
* Participating Nodes from Primary Site              [jessica cassidy]       +
  Participating Nodes from Secondary Site            [shanley]               +

  Startup Policy                                      Online On Home Node O> +
  Fallover Policy                                     Fallover To Next Prio> +
  Fallback Policy                                     Never Fallback         +
```

Figure 5-25 Creating two site resource group

5.4.5 Add GMVG into resource group

We now add all of our resources, including GMVG of amyvg, into the resource group. To do so, we execute **smitty sysmirror** → **Cluster Applications and Resources** → **Resource Groups** → **Change/Show Resources and Attributes for a Resource Group** and choose the only resource group, xsiteGLVMRG, and press Enter. For clarity, only the fields we utilized are shown in Figure 5-26. All other fields, we left them at their default values.

```
  Resource Group Name                                 xsiteGLVMRG
 Inter-Site Management Policy                        Online On Primary Site
 Participating Nodes from Primary Site               jessica cassidy
  Participating Nodes from Secondary Site             shanley

  Startup Policy                                      Online On Home Node O>
  Fallover Policy                                     Fallover To Next Prio>
  Fallback Policy                                     Never Fallback

  Service IP Labels/Addresses                        [dallasserv ftwserv]    +
  Application Controller Name                        [banner]                +

  Volume Groups                                      [amyvg]                 +
  Use forced varyon of volume groups, if necessary    true
```

Figure 5-26 Adding GMVG resource to resource group

The ending cluster resource group configuration looks very much like a cross-site LVM mirroring configuration. However, it is both an LVM mirrored solution within the dallas site. It is a GLVM configuration across sites to fortworth.

It is recommended next to synchronize the cluster to determine whether errors exist. An overview of our cluster is shown in Figure 5-27 on page 110.

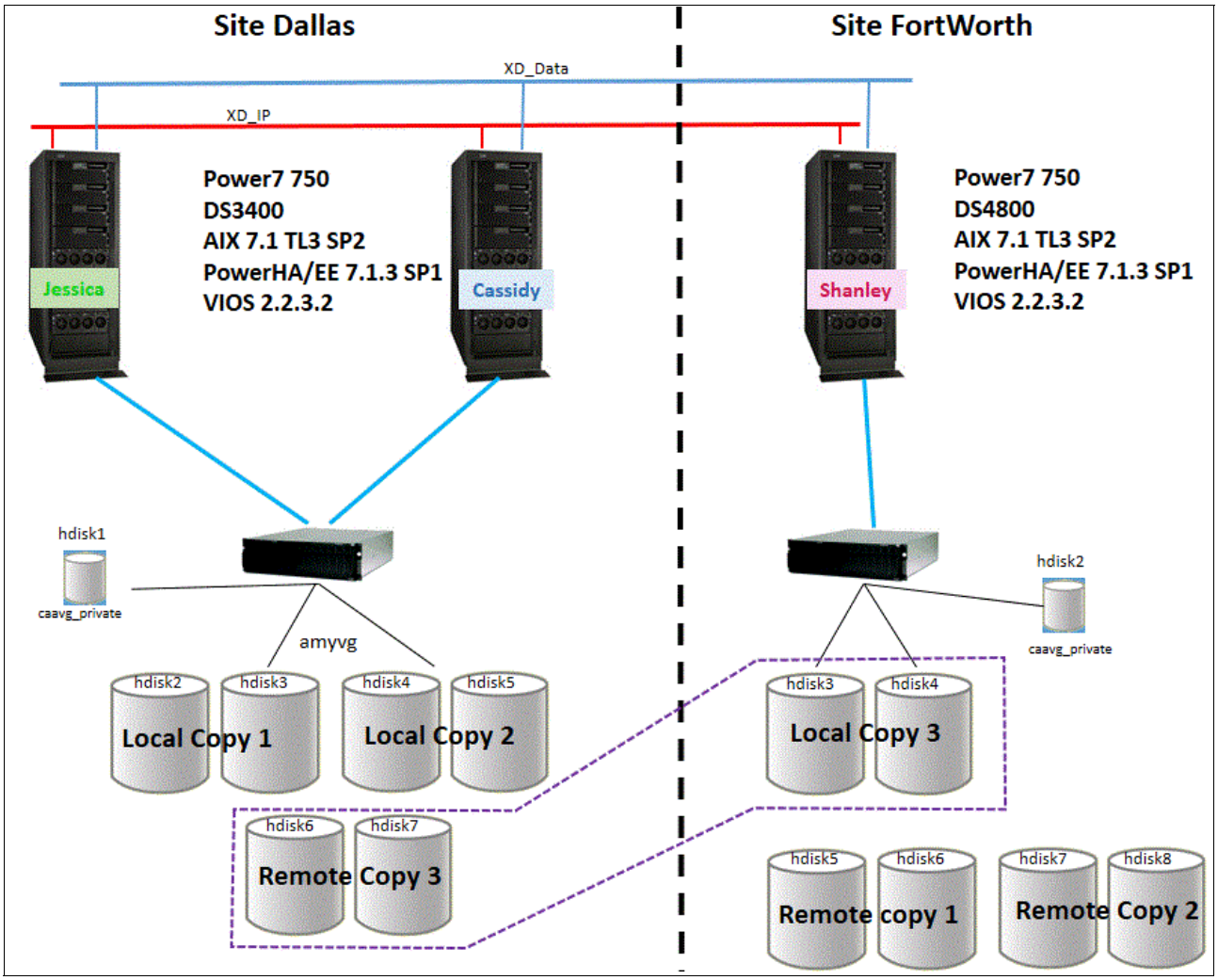

Figure 5-27 Test cluster final configuration

5.5 Defining manual site split and merge policy

Overall, PowerHA is designed for automatic fallover. However, in cases of sites with data replication involved there is some reluctance to allow it. Especially in the case of site isolation resulting in false fallover. This can lead to very undesired results by having both copies of data active. There are also times, especially when only one node at each site, that customers want to evaluate the initial site failure to determine if it is recoverable quickly before deciding to fallover to the disaster recovery site. This is generally because getting back to the primary production site involves at syncing up in deltas in the data. Though it does not have to be process intensive, it can be time intensive.

In previous versions of PowerHA when utilizing storage replication there was, and still is, an option to choose a *Manual* recovery action. Initially, this only applied to scenarios in which PowerHA detected that the replicated pairs where not in sync at the time of site failure. So it is codependent on the status of the replicated resources. This means that it was still possible for automatic fallover to proceed even with the manual option set.

In PowerHA v7.1.3, there are new *manual* split and merge policies. It can, and would be, applied globally across the cluster. However, there is an option to specify if it should apply to storage replication recovery or not.

Now when using GLVM specifically, there historically has been no method for manual fallover within PowerHA. If that was warranted, it was simply suggested to implement stand-alone GLVM.

Restriction: The manual split/merge option *only* applies to linked clusters but *cannot* be used with those replication methods that utilize the genxd filesets. At time of writing, this consisted of DS8000 and XIV.

To configure this feature, we execute **smitty sysmirror** → **Custom Cluster Configuration** → **Cluster Nodes and Networks** → **Initial Cluster Setup (Custom)** → **Configure Split and Merge Policy for a Linked Cluster** and then are presented with the SMIT menu as shown in Figure 5-28.

```
            Configure Split and Merge Policy for a Linked Cluster

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
  Split Handling Policy                           Manual                +
  Merge Handling Policy                           Manual                +

  Split and Merge Action Plan                     Reboot

  Select Tie Breaker                                                    +

  Manual Choice Options
         Notify Method                           []
         Notify Interval (seconds)               [20]                  +#
         Maximum Notifications                   [15]                  +#
         Default Surviving Site                  [dallas]              +
         Apply to Storage Replication Recovery    No                   +
```

Figure 5-28 Configure manual split and merge policy

The description and options of each field are shown in Table 5-3 on page 112.

Note: The *Manual choice option* was added in PowerHA 7.1.3. If you specify a number of notifications, you must choose a default surviving site. Inversely, if you do not choose maximum notifications, you *cannot* choose a default surviving site.

Table 5-3 Configure Cluster Split and Merge Policy fields

Field	Description
Split handling policy	Select **None**, the default setting, for the partitions to operate independently of each other after the split occurs. Select **Tie breaker** to use the disk that is specified in the **Select tie breaker** field after a split occurs. When the split occurs, one site wins the SCSI reservation on the tie breaker disk. The site that losses the SCSI reservation uses the recovery action that is specified in the policy setting. **Note:** If you select the **Tie breaker** option in the **Merge handling policy** field, you must select **Tie breaker** for this field. Select **Manual** to wait for manual intervention when a split occurs. PowerHA SystemMirror does not perform any actions on the cluster until you specify how to recover from the split. **Note:** If you select the **Manual** option in the **Merge handling policy** field, you must select **Manual** for this field.
Merge handling policy	Select **Majority** to choose the partition with the highest number of nodes as the primary partition. Select **Tie breaker** to use the disk that is specified in the **Select tie breaker** field after a merge occurs. **Note:** If you select the **Tie breaker** option in the **Split handling policy** field, you must select **Tie breaker** for this field. Select **Manual** to wait for manual intervention when a merge occurs. PowerHA SystemMirror does not perform any actions on the cluster until you specify how to handle the merge.
Split and merge action plan	**Reboot** will reboot all nodes in the site that does not win the tie breaker or is not responded to manually when using the manual choice option below. This is *not* an editable option.
Select tie breaker	Select an iSCSI disk or a SCSI disk that you want to use as the tie breaker disk. It must support either SCSI-2 or SCSI-3 reserves.

Field	Description
Manual Choice Option	**Notify Method** is a method to be invoked in addition to a message to /dev/console to inform the operator of the need to choose which site will continue after a split or merge. The method is specified as a path name, followed by optional parameters. When invoked, the last parameter will be either “split” or “merge” to indicate the event. **Notify Interval** The frequency of the notification time, in seconds, between the message to inform the operator of the need to choose which site will continue after a split or merge. The supported values are 10 - 3600. **Maximum Notifications** is the maximum number of times that PowerHA SystemMirror will prompt the operator to choose which site will continue after a split or merge. The default, blank, is infinite. Otherwise, the supported values are 3 - 1000. However, this value *cannot* be blank when a surviving site is specified. **Default Surviving Site** If the operator has not responded to a request for a manual choice of surviving site on a “split” or “merge”, this site will be allowed to continue. The other site will take the action chosen under “Action Plan”. The time the operator has to respond is “Notify Interval” times “Maximum Notifications+1”. **Apply to Storage Replication Recovery** determines if the manual response on a split also applies to those storage replication recovery mechanisms that provide an option for “Manual” recovery. If “Yes” is selected, the partition that was selected to continue on a split will proceed with takeover of the storage replication recovery. This *cannot* be used when DS8k and XIV replication is used.

In our scenario, we chose a *Manual* for both the split and merge policy. We also specified a *Notification Interval* of 20 seconds and *Maximum Notification* of 15. This is roughly equivalent to 5 minutes. Though in actuality, it is closer to five minutes and twenty seconds as the time for an operator is actually one additional notification interval above the maximum notifications as explained in the table above.

Synchronizing the cluster is required for these settings to take effect.

5.6 Testing manual split option

The requirement of a split is that all the IP communications between sites are lost. This can be achieved in numerous ways. In our case, we simply pulled the physical Ethernet cables from node shanley. However, since most environments are virtualized this is rare to have the ability to pull cables and it not affect other systems beyond the cluster nodes. In those cases, other options like disabling switch ports or dynamically changing the virtual adapters can also give the wanted results.

To test the cluster, we begin will all nodes active in the cluster. We then perform the following steps:

1. Halt first node, jessica, in dallas site.
2. Sever links between sites.
3. Respond to prompt to continue on remote site.
4. Respond to prompt to recover on primary site.
5. Restart primary site nodes to rejoin cluster.
6. Manually move resource group back to primary site.

5.6.1 First node failure in site dallas

In example Example 5-19, we show the current status of the resource group and resources in the cluster as we begin. Notice that all disks in amyvg are active. This will change after fallover to the remote site and will be highlighted again later.

Example 5-19 All nodes active and beginning resource status

```
[jessica:root] / # clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
-----------------------------------------------------------------------------
xsiteGLVMRG    ONLINE                         jessica@dallas
               OFFLINE                        cassidy@dallas
               ONLINE SECONDARY               shanley@fortwo

[jessica:root] / # lspv
hdisk0          00f6f5d00146570c                    rootvg          active
hdisk1          00f6f5d015a4310b                    caavg_private   active
hdisk2          00f6f5d01660fbd1                    amyvg           active
hdisk3          00f6f5d015a44307                    amyvg           active
hdisk4          00f6f5d0166106fa                    amyvg           active
hdisk5          00f6f5d0166114f3                    amyvg           active
hdisk6          00f6f5d029906df4                    None
hdisk7          00f61ab21664556b                    amyvg           active
hdisk8          00f61ab21664742e                    amyvg           active

[jessica:root] / # lsvg -p amyvg
amyvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk3            active            78          58          16..00..11..15..16
hdisk2            active            478         457         96..75..95..95..96
hdisk4            active            478         457         96..75..95..95..96
hdisk5            active            478         458         96..76..95..95..96
hdisk7            active            478         457         96..75..95..95..96
hdisk8            active            478         458         96..76..95..95..96

[jessica:root] / # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      ee.af.1.71.78.2   1015673     0   962125     0     0
en0   1500  10.10.8     dallasserv        1015673     0   962125     0     0
en0   1500  192.168.100 jessica           1015673     0   962125     0     0
en1   1500  link#3      ee.af.1.71.78.3   1454861     0  1644616     0     0
en1   1500  192.168.150 jessica_xd        1454861     0  1644616     0     0
```

There are several ways to fail a node. Generally, either an immediate shutdown via the Hardware Management Console (HMC) or via **halt -q**. In our case, we execute the **halt** command. This results in a fallover to second local dallas node, cassidy. The results of the fallover look very similar to what it did on node jessica. It is just now all resources are online to node cassidy, as shown in Example 5-20.

Example 5-20 Resources active on cassidy after local node failure

```
[cassidy:root] / # clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
```

```
------------------------------------------------------------------------------
xsiteGLVMRG    OFFLINE                    jessica@dallas
               ONLINE                     cassidy@dallas
               ONLINE SECONDARY           shanley@fortwo

[cassidy:root] / # lspv
hdisk0          00f70c99013e28ca                    rootvg          active
hdisk1          00f6f5d015a4310b                    caavg_private   active
hdisk2          00f6f5d015a44307                    amyvg           active
hdisk3          00f6f5d01660fbd1                    amyvg           active
hdisk4          00f6f5d0166106fa                    amyvg           active
hdisk5          00f6f5d0166114f3                    amyvg           active
hdisk6          00f61ab21664556b                    amyvg           active
hdisk7          00f61ab21664742e                    amyvg           active

[cassidy:root] / # lsvg -p amyvg
amyvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk2            active            78          58          16..00..11..15..16
hdisk3            active            478         457         96..75..95..95..96
hdisk4            active            478         457         96..75..95..95..96
hdisk5            active            478         458         96..76..95..95..96
hdisk6            active            478         457         96..75..95..95..96
hdisk7            active            478         458         96..76..95..95..96

[cassidy:root] / # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      7a.40.c8.b3.15.2  1266099     0  1267951     0     0
en0   1500  10.10.8     dallasserv        1266099     0  1267951     0     0
en0   1500  192.168.100 cassidy           1266099     0  1267951     0     0
en1   1500  link#3      7a.40.c8.b3.15.3  1019856     0   980645     0     0
en1   1500  192.168.150 cassidy_xd        1019856     0   980645     0     0
```

Notice again that all the disks are active. This is normal because cassidy has access to all the local disks and during fallover RPV server changes to the new client address of cassidy. However, we will see something different after failing cassidy as this results in a site failure.

5.6.2 Site split

This case is a continuation of the previous section. Dallas site node jessica is still down but both local node cassidy and remote shanley are still active in the cluster. We physically pull the Ethernet cables from node shanley. This results in a site split.

Note: A demo of split site recovery on this exact cluster can be found at:

http://youtu.be/rcSjnKFksQk

Upon detection of the site split, and the fact that we chose manual split merge options previously, we are prompted with the following via the console terminal window on the HMC on *both* nodes, as shown in Figure 5-29 on page 116.

The manual prompt status can also be found on either node without access to a console terminal window by executing **smitty sysmirror** → **Problem Determination Tools** → **Manual Response to Split or Merge** → **Display any needed Manual Response**.

```
[shanley:root] / # May 26 13:46:11 shanley local0:crit clstrmgrES[12648580]: Mon May
26 13:46:11 Removing 5 from ml_idx
A cluster split has been detected.
You must decide if this side of the partitioned cluster is to continue.
To have it continue, enter

       /usr/es/sbin/cluster/utilities/cl_sm_continue

To have the recovery action - Reboot - taken on all nodes on this partition, enter

       /usr/es/sbin/cluster/utilities/cl_sm_recover
```

Figure 5-29 Manual operator response prompt upon site split

In our case, we do want the remote site, fortworth, to continue with takeover. To do as the console message indicates, we simply execute **`cl_sm_continue`**. However, this too can be performed within SMIT via **smitty sysmirror** → **Problem Determination Tools** → **Manual Response to Split or Merge** → **Provide a Manual Response**. Then, choose either *Continue* or *Recover* as shown in Figure 5-30.

We also want the local site node, cassidy, to reboot. We execute **`cl_sm_recover`** in order for it to be rebooted for us.

```
                          Provide a Manual Response

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
* This site should                                       Recover +



  +--------------------------------------------------------------------------+
  |                             This site should                             |
  |                                                                          |
  | Move cursor to desired item and press Enter.                             |
  |                                                                          |
  |   Continue                                                               |
  |   Recover                                                                |
  |                                                                          |
  | F1=Help                F2=Refresh              F3=Cancel                 |
F1| F8=Image               F10=Exit                Enter=Do                  |
F5| /=Find                 n=Find Next                                       |
F9+--------------------------------------------------------------------------+
```

Figure 5-30 Manual response via SMIT

Now since the resources are activated on node shanley there are a couple of key differences. They are the fact that the site-specific service address of *ftwserv* is active and that since the primary site is down, the four remote disks are now in the *missing* state as shown in Example 5-21 on page 117.

Example 5-21 Remote takeover completed

```
[shanley:root] / # cl_sm_continue
Resource Class Action Response for ResolveOpQuorumTie
[shanley:root] / # clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
-----------------------------------------------------------------------------
xsiteGLVMRG    OFFLINE                        jessica@dallas
               OFFLINE                        cassidy@dallas
               ONLINE                         shanley@fortwo

[shanley:root] / # lspv
hdisk1          00f61ab215ad00d4                    rootvg          active
hdisk2          00f61ab216646614                    caavg_private   active
hdisk3          00f61ab21664556b                    amyvg           active
hdisk4          00f61ab21664742e                    amyvg           active
hdisk5          00f6f5d015a44307                    amyvg           active
hdisk6          00f6f5d01660fbd1                    amyvg           active
hdisk7          00f6f5d0166106fa                    amyvg           active
hdisk8          00f6f5d0166114f3                    amyvg           active

[shanley:root] / # lsvg -p amyvg
amyvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            missing           78          58          16..00..11..15..16
hdisk6            missing           478         457         96..75..95..95..96
hdisk7            missing           478         457         96..75..95..95..96
hdisk8            missing           478         458         96..76..95..95..96
hdisk3            active            478         457         96..75..95..95..96
hdisk4            active            478         458         96..76..95..95..96

[shanley:root] / # lsvg amyvg |grep STALE
STALE PVs:          4                       STALE PPs:      14

[shanley:root] / # netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en0   1500  link#2      6e.8d.d0.21.d0.2  1532614     0  1530640     0     0
en0   1500  10.10.8     ftwserv           1532614     0  1530640     0     0
en0   1500  192.168.100 shanley           1532614     0  1530640     0     0
en1   1500  link#3      6e.8d.d0.21.d0.3  1613987     0  1183422     0     0
```

5.6.3 Restart primary site nodes

In this case we perform two steps:

1. Restart both LPARs via the HMC
2. Restart cluster services on both nodes

Upon successful reintegration of the primary site, PowerHA/EE will automatically perform the following actions:

- Activate RPV servers on dallas site node jessica
- Activate RPV clients on node shanley
- Reactive volume group to change disks from missing to active
- Automatically resync the stale partitions in the volume group

The end result of these actions is shown in Example 5-22.

Example 5-22 Auto resync after site reintegration

```
[shanley:root] / # clRGinfo
-----------------------------------------------------------------------------
Group Name     State                          Node
-----------------------------------------------------------------------------
xsiteGLVMRG    ONLINE SECONDARY               jessica@dallas
               OFFLINE                        cassidy@dallas
               ONLINE                         shanley@fortwo

[shanley:root] / # lsvg -p amyvg
amyvg:
PV_NAME           PV STATE          TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk5            active            78          58          16..00..11..15..16
hdisk6            active            478         457         96..75..95..95..96
hdisk7            active            478         457         96..75..95..95..96
hdisk8            active            478         458         96..76..95..95..96
hdisk3            active            478         457         96..75..95..95..96
hdisk4            active            478         458         96..76..95..95..96
[shanley:root] / # lsvg amyvg |grep STALE
STALE PVs:          0                         STALE PPs:      0
```

5.6.4 Move resource group back to primary site

To move the resource group back to the primary site, like most tasks, it can be done both via SMIT or the **clmgr** command line. It also can be executed on any active node in the cluster. In our case, we used the **clmgr** command on node shanley as shown in Example 5-23.

Example 5-23 Resource group move via clmgr

```
[shanley:root] / # clmgr move rg xsiteGLVMRG site=dallas
Attempting to move group xsiteGLVMRG to site dallas.

Waiting for the cluster to process the resource group movement request....

Waiting for the cluster to stabilize.....

Resource group xsiteGLVMRG is online on site dallas.


Cluster Name: PHASEtoEE

Resource Group Name: xsiteGLVMRG
Node                          Primary State    Secondary State
----------------------------- ---------------
jessica@dallas                ONLINE           OFFLINE
cassidy@dallas                OFFLINE          OFFLINE
shanley@fortworth             OFFLINE          ONLINE SECONDARY
```

To perform the same operation via SMIT, execute **smitty cspoc** → **Resource Group and Applications** → **Move Resource Groups to Another Site** and choose the online resource group *xsiteGLVMRG* as shown in Figure 5-31 on page 119.

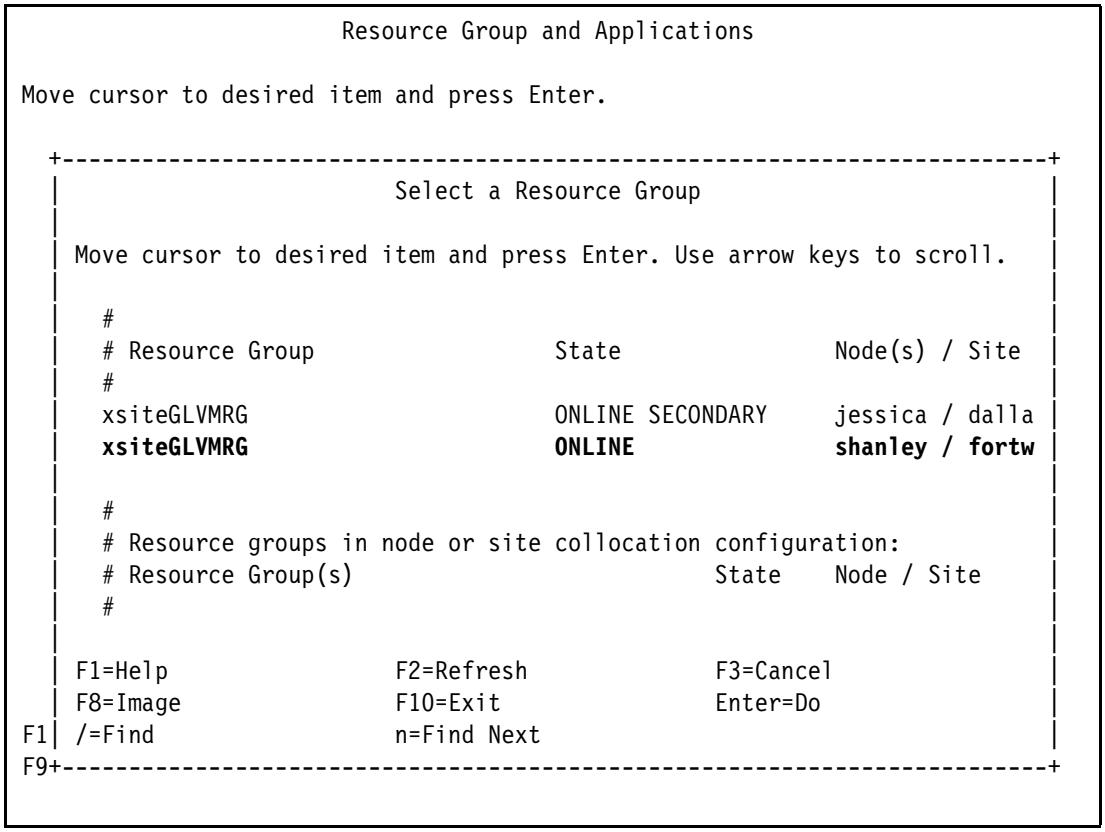

Figure 5-31 Select resource group to move

We then choose site *dallas* from the pop-up picklist as shown in Figure 5-32 on page 120 as it is the only other site possible to move it to. We are then presented with the final SMIT menu as shown in Figure 5-33 on page 120.

Upon successful execution, we get identical feedback as we did from using the **clmgr** command as shown in Example 5-23 on page 118.

```
                        Resource Group and Applications

Move cursor to desired item and press Enter.

  Show the Current State of Applications and Resource Groups
  Bring a Resource Group Online
  Bring a Resource Group Offline
  Move Resource Groups to Another Node
  Move Resource Groups to Another Site

  Suspend/Resume Application Monitoring
  Application Availability Analysis
  +-------------------------------------------------------------------------+
  |                        Select a Destination Site                        |
  |                                                                         |
  | Move cursor to desired item and press Enter.                            |
  |                                                                         |
  |   # *Denotes Originally Configured Primary Site                         |
  |   dallas                                                                |
  |                                                                         |
  | F1=Help                 F2=Refresh              F3=Cancel               |
  | F8=Image                F10=Exit                Enter=Do                |
F1| /=Find                  n=Find Next                                     |
F9+-------------------------------------------------------------------------+
```

Figure 5-32 Select site to move resource group to

```
                     Move Resource Group(s) to Another Site

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Resource Group(s) to be Moved                      xsiteGLVMRG
  Destination Site                                   dallas
  Name of site containing data to be preserved                              +
  by data divergence recovery processing
  (Asynchronous GLVM Mirroring only)
```

Figure 5-33 Move resource group final SMIT window

5.7 Testing manual merge option

We start off the same way as in the manual split test. We cause a split to occur; however, we simply do not answer the prompt. We are strictly trying to create a merge scenario and all event processing is held up when the prompt occurs. So we have to let each one think they are the only one in the cluster.

We then restore communications between the sites. The prompt notification changes on both nodes as shown in Figure 5-29 on page 116. But this time, the message indicates a *merge* was detected.

In our case, we told the local site to continue and the remote site to recover. This caused the remote site to reboot. We then restarted cluster services on the remote node and it joined the cluster and stabilized.

Now if we would have told the remote site to recover when the split was detected instead, and communications restored before rejoining it into the cluster, a merge condition would not have occurred.

In either case of a split or merge, the options must be very carefully considered. Better yet, they also must be fully tested to ensure the wanted results.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *IBM PowerHA Cookbook for AIX Updates*, SG24-7739
- *IBM PowerHA SystemMirror for AIX Cookbook Update*, SG24-7739-01
- *IBM PowerHA SystemMirror Standard Edition 7.1.1 for AIX Update*, SG24-8030
- *IBM PowerVM Enhancements What is New in 2013, SG24-8198*
- *IBM PowerHA SystemMirror 7.1.2 Enterprise Edition for AIX, SG24-8106*

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Symantec Storage Foundation Documentation

  https://sort.symantec.com/documents/doc_details/sfha/6.0/Linux/ProductGuides
- Fix Level Recommendation Tool (FLRT)

  http://www14.software.ibm.com/webapp/set2/flrt/home
- Symantec Operations Readiness Tools (SORT)

  http://sort.symantec.com

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

© Copyright IBM Corp. 2015. All rights reserved.

IBM Redbooks

IBM PowerHA SystemMirror for AIX 7.1.3 Best Practices and Migration Guide

(0.2”spine)
0.17”<->0.473”
90<->249 pages

IBM PowerHA SystemMirror for AIX 7.1.3 Best Practices and Migration Guide

Positions technically IBM PowerHA SystemMirror

Includes best practices guidelines

Describes migration scenarios

This IBM Redbooks publication positions high availability solutions for IBM Power Systems with IBM PowerHA SystemMirror Standard and Enterprise Editions (hardware, software, best practices, reference architectures, migration, and tools) with a well-defined and documented deployment model within an IBM Power Systems environment allowing customers a planned foundation for a dynamic high available infrastructure for their enterprise applications.

This Redbooks publication documents topics to leverage the strengths of IBM PowerHA SystemMirror Standard and Enterprise Editions 7.1.3 for IBM Power Systems to solve customers' application high availability challenges, and maximize systems' availability, and management.

This Redbooks publication focuses on providing the readers with technical information and references on the capabilities of each edition, functionalities, usability, and features that make IBM PowerHA SystemMirror a premier solution for high availability and disaster recovery for IBM Power Systems servers.

This Redbooks publication helps strengthen the position of the IBM PowerHA SystemMirror solution with a well-defined and documented best practices, usability, functionality, migration and deployment model within an IBM POWER system virtualized environment allowing customers a planned foundation for business resilient infrastructure solutions.

This Redbooks publication is targeted toward technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) responsible for providing high availability solutions and support with the IBM PowerHA SystemMirror on IBM POWER.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8234-00 ISBN 0738440205