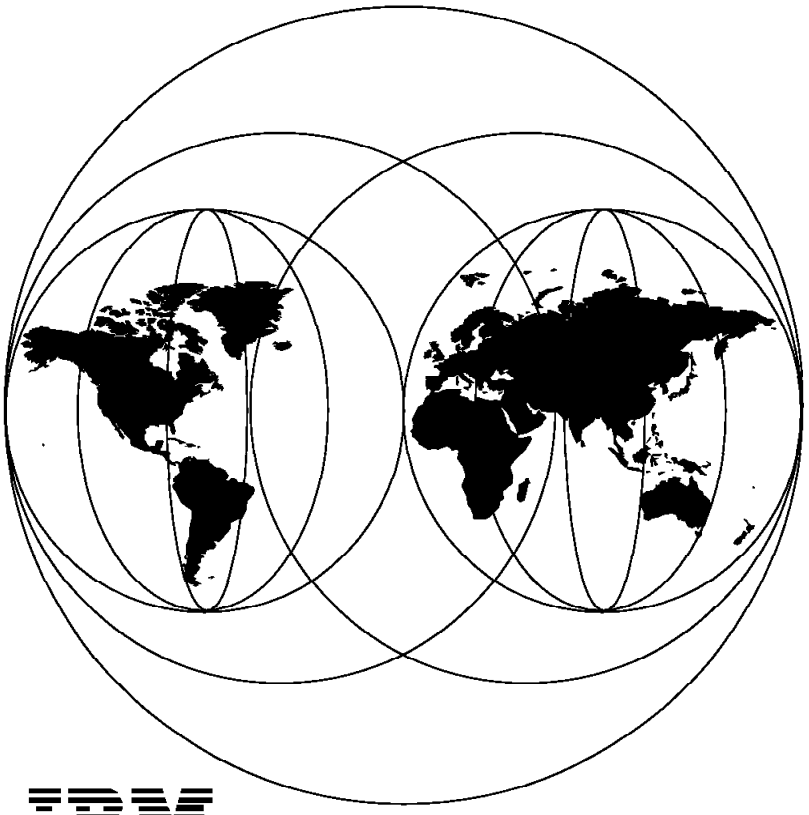
International Technical Support Organization

SG24-4551-00

High Availability on the RISC System/6000 Family

October 1995



IBM

International Technical Support Organization

Austin Center

International Technical Support Organization

SG24-4551-00

High Availability on the RISC System/6000 Family

October 1995

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xiii.

First Edition (October 1995)

This edition applies to Version 3.1 of HACMP/6000, Program Number 5696-923 for use with the AIX/6000 Version 3.2.5 Operating System and Version 4.1 of HACMP for AIX, Program Number 5696-933 for use with the AIX Version 4.1 Operating System.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. JN9B Building 821 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Abstract

This document is created for those who wish to implement a highly available AIX environment, using the HACMP for AIX product. It includes information on how to use standard AIX operating system facilities to enhance the availability of a single system, and also describes how to set up an HACMP cluster. The document also provides many helpful hints and tips to make your implementation easier. Some knowledge of the AIX operating system is assumed.

This document obsoletes and replaces the document "High Availability Strategies for AIX" (order number GG24-3684).

(241 pages)

Contents

Abstract . . . iii

Special Notices . . . xiii

Preface . . . xv
How This Document is Organized . . . xv
Related Publications . . . xvi
International Technical Support Organization Publications . . . xvi
ITSO Redbooks on the World Wide Web (WWW) . . . xvii
Acknowledgments . . . xvii

Chapter 1. Introduction to High Availability . . . 1
1.1 Introduction to Availability . . . 1
1.1.1 Why is Availability Important? . . . 2
1.2 Levels of Availability . . . 3
1.2.1 Availability Continuum . . . 6
1.3 Key Threats to System Availability . . . 6
1.4 Availability Measurements . . . 7
1.5 Availability as a Total System Concept . . . 8
1.5.1 Failure Rate and Availability . . . 8

Chapter 2. Single System Availability . . . 11
2.1 Availability Features of AIX . . . 11
2.1.1 System Management Interface Tool (SMIT) . . . 11
2.1.2 Logical Volume Manager (LVM) . . . 11
2.1.3 Journaled Filesystem (JFS) . . . 12
2.1.4 Dynamic AIX Kernel . . . 12
2.1.5 System Resource Controller (SRC) . . . 13
2.1.6 Configuration Manager . . . 13
2.1.7 AIX Update Facilities . . . 13
2.2 Availability Features of the RISC System/6000 . . . 13
2.2.1 Built-In Error Detection and Correction . . . 13
2.2.2 Backup Power Supply . . . 14
2.2.3 Power Conditioning . . . 14
2.2.4 Redundant or Spare Disks . . . 15
2.2.5 Hot Pluggable Disk Drives . . . 15
2.2.6 Multi-Tailed Disks and Shared Volume Groups . . . 15
2.2.7 RAID Disk Arrays . . . 15
2.3 Improved Availability Through System Management Practices . . . 19
2.3.1 Skills Management . . . 20
2.3.2 Operations Management . . . 20
2.3.3 Capacity Management . . . 20
2.3.4 Change Management . . . 21
2.3.5 System Test Environment . . . 21
2.3.6 Performance Management . . . 22
2.3.7 Problem Management . . . 22
2.3.8 Service Level Management . . . 22
2.3.9 Automated Operations . . . 22
2.4 Isolation . . . 23
2.4.1 Recovery Management . . . 23

2.5 Summary 24

Chapter 3. Clustering RISC System/6000s for High Availability 25
3.1 An Introduction to Clustering 25
3.1.1 What Is a Cluster? 25
3.1.2 Single Points of Failure 26
3.1.3 Eliminating Single Points of Failure in a Cluster 26
3.1.4 Disadvantages of Manual Intervention 28
3.2 High Availability Cluster Multi-Processing/6000 28
3.2.1 HACMP Technical Overview 29
3.2.2 HACMP Cluster Components 30
3.2.3 HACMP Cluster Resources 42
3.2.4 HACMP Cluster Configurations 45
3.2.5 HACMP Cluster Events 49
3.2.6 Clients in an HACMP Cluster 55
3.3 Conclusion 57

Chapter 4. Hardware Options for HACMP 59
4.1 CPU Options 59
4.2 How to Select CPU Nodes for Your Cluster 60
4.2.1 Node Configuration Guidelines 61
4.3 Storage Options 63
4.3.1 SCSI Technologies 63
4.3.2 Conventional SCSI Disk Options 65
4.3.3 RAID Disk Array Features 67
4.3.4 RAID Disk Array Options 68
4.3.5 Serial/SSA Disk Storage 70
4.3.6 Choosing a Shared Disk Technology 75
4.4 Connectivity Options 77
4.4.1 TCP/IP Networks 78
4.4.2 Choosing a Network for Your Cluster 79
4.4.3 Non-TCP/IP Networks 80

Chapter 5. Setting Up HACMP for AIX 81
5.1 Installing and Configuring HACMP/6000 Version 3.1 83
5.2 Preparing AIX for an HACMP Cluster 83
5.2.1 Configuring IP Networks 84
5.2.2 Installing Shared SCSI Disks 91
5.2.3 Defining Shared LVM Components 95
5.2.4 Additional Tasks 101
5.3 Setting Up a New HACMP Cluster 102
5.3.1 Installation Prerequisites 103
5.3.2 Installation Options 103
5.3.3 Installing the HACMP Software on Node 1 and Node 2 103
5.3.4 Installing the HACMP Client Portion on Client Systems 104
5.3.5 Rebooting Nodes and Clients 106
5.3.6 Verifying the Cluster Software 106
5.3.7 Defining the Cluster Environment 109
5.3.8 Defining Application Servers 120
5.3.9 Creating Resource Groups 121
5.3.10 Verify Cluster Environment 129
5.3.11 Starting Cluster Services 130
5.4 Upgrading a Cluster to HACMP/6000 Version 3.1 132
5.4.1 Prerequisites for Upgrade 136

5.4.2 Preparing the Cluster for the Upgrade . . . 136
5.4.3 Installing the HACMP/6000 Version 3.1 Software . . . 139
5.4.4 Upgrading from Version 2.1 to Version 3.1 . . . 140
5.4.5 Further Tasks . . . 145
5.4.6 Verification . . . 155
5.5 Upgrading a Cluster from HACMP/6000 Version 3.1 . . . 157

Chapter 6. Cluster Tuning and Customization . . . 159
6.1 When is Cluster Tuning Required . . . 159
6.2 False Takeover . . . 160
6.3 The Deadman Switch . . . 160
6.4 Tuning System I/O . . . 162
6.4.1 I/O Pacing . . . 162
6.4.2 sync Daemon (syncd) . . . 164
6.4.3 Disabling the Deadman Switch . . . 165
6.5 Cluster Manager Startup Parameters . . . 166
6.5.1 HACMP/6000 Version 2.1 . . . 166
6.5.2 HACMP/6000 Version 3.1 . . . 168
6.5.3 Pinning the Cluster Manager . . . 171
6.6 HACMP Cluster Customization Examples . . . 171
6.7 Pre-Event and Post-Event Script Parameters . . . 172
6.8 AIX Error Notification . . . 172

Chapter 7. Tips and Techniques . . . 175
7.1 Change Management . . . 175
7.1.1 Cluster Verification . . . 176
7.1.2 Software Upgrades and Fixes . . . 176
7.1.3 Cluster Maintenance - Do's and Dont's . . . 179
7.1.4 Requirement for Additional Filesystems . . . 179
7.1.5 Requirement for New Applications . . . 180
7.1.6 Requirement for New Communications Connectivity . . . 180
7.2 Mirroring the Root Volume Group (rootvg) . . . 180
7.2.1 Solution . . . 181
7.2.2 Procedure . . . 182
7.2.3 Testing Your Configuration . . . 184
7.3 Quorum . . . 184
7.3.1 Quorum in Shared Disk Configurations . . . 186
7.4 /etc/filesystems and the jfslog . . . 186
7.5 Filesystem Helper . . . 188
7.6 Phantom Disks . . . 189
7.7 /etc/inittab File . . . 190
7.8 Permissions on the /tmp Directory . . . 191
7.9 ARP Cache . . . 191
7.10 Synchronizing Time Between Cluster Nodes . . . 192
7.11 Tips on Writing Application Server and Event Scripts . . . 193
7.11.1 Problem for HACMP/6000 Version 3.1 Users Before PTF U438726 193
7.12 Recovery From Event Script Failure . . . 195
7.13 AIX Error Notification . . . 196
7.14 7135 RAIDiant Disk Array . . . 196
7.15 Resource Group Organization . . . 197
7.16 Hubs in an HACMP Cluster . . . 197

Appendix A. HACMP Software Components . . . 199
A.1 Cluster Manager . . . 199

A.1.1 Cluster Controller 200
A.1.2 Network Interface Layer 201
A.1.3 Network Interface Modules 202
A.1.4 Event Manager 203
A.2 Cluster SMUX Peer and Cluster Information Services 203
A.3 Cluster Lock Manager 204
A.3.1 UNIX System V Locking Model 204
A.3.2 CLM Locking Model 205
A.3.3 Lock Management 206
A.4 Interaction Between the HACMP Software Components 206

Appendix B. Disk Setup in an HACMP Cluster 209
B.1 SCSI Disks and Subsystems 209
B.1.1 SCSI Adapters 209
B.1.2 Individual Disks and Enclosures 212
B.1.3 Hooking It All Up 213
B.1.4 AIX's View of Shared SCSI Disks 218
B.2 RAID Subsystems 218
B.2.1 SCSI Adapters 219
B.2.2 RAID Enclosures 219
B.2.3 Connecting RAID Subsystems 219
B.2.4 AIX's View of Shared RAID Devices 223
B.3 Serial Disk Subsystems 224
B.3.1 High-Performance Disk Drive Subsystem Adapter 224
B.3.2 9333 Disk Subsystems 224
B.3.3 Connecting Serial Disk Subsystems in an HACMP Cluster 224
B.3.4 AIX's View of Shared Serial Disk Subsystems 226
B.4 Serial Storage Architecture (SSA) Subsystems 226
B.4.1 SSA Software Requirements 226
B.4.2 SSA Four Port Adapter 227
B.4.3 IBM 7133 SSA Disk Subsystem 228
B.4.4 SSA Cables 229
B.4.5 Connecting 7133 SSA Subsystems in an HACMP Cluster 230
B.4.6 AIX's View of Shared SSA Disk Subsystems 233

Appendix C. Measurements of Disk Reliability 235
C.1 Mean Time Between Failure (MTBF) 235
C.1.1 Defining Mean Time Between Failure 235
C.1.2 Mean Time Between Failure and Actual Failures. 235
C.1.3 Predicted Mean Time Between Failure 236
C.1.4 Comparison of MTBF Figures 237
C.2 Cumulative Distribution Function (CDF) 237

List of Abbreviations 239

Index 241

Figures

1. Availability Continuum 5
2. Availability is a System-Wide Consideration 8
3. The RAID Concept 16
4. RAID-0 Illustration 16
5. RAID-1 Illustration 17
6. RAID-3 Illustration 18
7. RAID-5 Illustration 19
8. HACMP Cluster Example 31
9. Disk Takeover 39
10. IP Address Takeover 40
11. Adapter Swapping 41
12. Cascading Resource Group 43
13. Rotating Resource Group 44
14. Concurrent Resource Group 45
15. Hot Standby Configuration 46
16. Mutual Takeover Configuration 47
17. Third-Party Takeover Configuration 48
18. First Node Joins Cluster 51
19. Node Joins an Active Cluster 52
20. Node Fails 53
21. Node Leaves the Cluster Gracefully with Takeover 54
22. Flow of Execution of Event Scripts 56
23. Single-Ended (SE) and Differential-Ended (DE) SCSI 64
24. 9334-501 Shared Between Two Deskside Systems 66
25. 9333 Serial Disk Drive Subsystem System Attachment 71
26. Four-Node Cluster with Shared 9333 Serial Disk Subsystem (Rear View) 72
27. Typical SSA Loop Topology 73
28. SSA-Based High Availability Servers 74
29. Sample HACMP Cluster Configuration 82
30. Alternative RS232 Serial Line Connection 89
31. Defining Shared LVM Components for Non-Concurrent Access 96
32. Cluster Running HACMP/6000 Version 2.1 132
33. HACMP/6000 Version 2.1 Cluster Configuration 133
34. HACMP/6000 Version 2.1 Node Environment Configuration 135
35. Cluster Configuration After Upgrade from HACMP/6000 Version 2.1 . . 142
36. Node 1's Resources After Upgrade from HACMP/6000 Version 2.1 . . 144
37. Cluster Configuration After Changes 149
38. Node1 Resources After Changes 151
39. Node2 Resources After Changes 152
40. Cluster Verification Output After Changes 156
41. Applying Software Fixes, Part 1 177
42. Applying Software Fixes, Part 2 178
43. Volume Group Quorum 185
44. Quorum for Shared Disks in HACMP Configurations 186
45. Phantom Disks 189
46. Protecting Your Network against Hub Failure 198
47. Cluster Manager Structure and Peer Connectivity in an HACMP Cluster 200
48. Interaction Between the HACMP Software Components in a Cluster . . 207
49. Termination Resistor Blocks on the SCSI-2 Differential Controller . . . 210

50. Termination Resistor Blocks on the SCSI-2 Differential Fast/Wide Adapter/A and Enhanced SCSI-2 Differential Fast/Wide Adapter/A . . . 210
51. 7204-215 External Disk Drives Connected on an 8-Bit Shared SCSI Bus 213
52. 7204-315 External Disk Drives Connected on a 16-Bit Shared SCSI Bus 214
53. 9334-011 SCSI Expansion Units Connected on an 8-Bit Shared SCSI Bus 216
54. 9334-501 SCSI Expansion Units Connected on an 8-Bit Shared SCSI Bus 216
55. 7134-010 High Density SCSI Disk Subsystem Connected on Two 16-Bit Shared SCSI Buses 218
56. 7135-110 RAIDiant Arrays Connected on Two Shared 8-Bit SCSI Buses 220
57. 7135-110 RAIDiant Arrays Connected on Two Shared 16-Bit SCSI Buses 221
58. 7137 Disk Array Subsystems Connected on an 8-Bit SCSI Bus 222
59. 7137 Disk Array Subsystems Connected on a 16-Bit SCSI Bus 223
60. 9333-501 Connected to Eight Nodes in an HACMP Cluster (Rear View) 225
61. SSA Four Port Aapter 227
62. IBM 7133 SSA Disk Subsystem 229
63. High Availability SSA Cabling Scenario 1 231
64. High Availability SSA Cabling Scenario 2 233

Tables

1. SCSI Adapter Characteristics 64
2. SCSI-2 External Disk Drive Enclosures 65
3. TCP/IP Network Types and Their Attributes 78
4. Cluster Manager Startup Switches 166
5. Cluster Manager Failure Detection Rates 169
6. Organization of Volume Group Descriptor Areas on Physical Disks . . 181
7. Mode Compatibility for CLM Locks 205
8. Serial Storage Architecture (SSA) Cables 230

Special Notices

This publication is intended to help RISC System/6000 users to understand, plan and configure a high availability solution for AIX. The information in this publication is not intended as the specification of any programming interfaces that are provided by High Availability Cluster Multi-Processing/6000 Version 3.1, HACMP 4.1 for AIX, or AIX Version 3.2.5 or 4.1. See the PUBLICATIONS section of the IBM Programming Announcements for AIX Version 3.2 and 4.1, HACMP/6000 Version 3.1, and HACMP 4.1 for AIX for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

AIX
AIX/6000
IBM
InfoExplorer
Micro Channel
NetView

POWERserver
RISC System/6000
RS/6000

The following terms in this publication are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.
Network File System and NFS are trademarks of SUN Microsystems, Inc.
PC Direct is a trademark of Ziff Communications Company and is
used by IBM Corporation under license.
SUN Microsystems is a trademark of SUN Microsystems, Inc.
UNIX is a registered trademark in the United States and other
countries licensed exclusively through X/Open Company Limited.
Windows is a trademark of Microsoft Corporation

Other trademarks are trademarks of their respective companies.

Preface

This document is created for those who wish to implement a highly available AIX environment, using the HACMP for AIX product. It includes information on how to use standard AIX operating system facilities to enhance the availability of a single system, and also describes how to set up an HACMP cluster. The document also provides many helpful hints and tips to make your implementation easier.

This document was written for anyone investigating or planning a highly available AIX environment. Some knowledge of the AIX operating system is assumed.

This document obsoletes and replaces the document "High Availability Strategies for AIX" (order number GG24-3684).

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction to High Availability”

  This chapter introduces availability concepts, including basic terminology, levels of availability, and various measurements of availability.

- Chapter 2, “Single System Availability”

  This chapter covers the AIX and RS/6000 features that can be exploited to increase the availability of a single AIX system. It also describes system management disciplines that are essential in maintaining a highly available system.

- Chapter 3, “Clustering RISC System/6000s for High Availability”

  This chapter introduces the concept of clustering RISC System/6000s for higher availability. It goes on to introduce the concepts and capabilities of the HACMP product.

- Chapter 4, “Hardware Options for HACMP”

  This chapter introduces the hardware options available in configuring your cluster, in terms of CPU, shared disk, and communications facilities. It also provides guidance on how to decide among the options available.

- Chapter 5, “Setting Up HACMP for AIX”

  This chapter describes how to install and set up an HACMP cluster. It also gives instruction on upgrading your cluster from a previous version of the HACMP product.

- Chapter 6, “Cluster Tuning and Customization”

  This chapter describes how to tune your cluster to avoid false takeover events, and how to customize the cluster manager behavior and event processing to suit your requirements. It also includes information on using the Error Notification facility of HACMP.

- Chapter 7, “Tips and Techniques”

  This chapter provides a number of tips and techniques, to allow smoother installation, setup, and adminstration of an AIX cluster.

- Appendix A, “HACMP Software Components”

  This appendix gives a description of the major software components of HACMP, and how they interact.

- Appendix B, “Disk Setup in an HACMP Cluster”

  This appendix gives a detailed description of adapter and cabling requirements for all the types of shared disk supported under HACMP.

- Appendix C, “Measurements of Disk Reliability”

  This appendix describes the concept of Mean Time Between Failure.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

HACMP/6000 Version 3.1

- *HACMP/6000 Concepts and Facilities*, SC23-2699
- *HACMP/6000 Planning Guide*, SC23-2700
- *HACMP/6000 Installation Guide*, SC23-2701
- *HACMP/6000 Administration Guide*, SC23-2702
- *HACMP/6000 Troubleshooting Guide*, SC23-2703
- *HACMP/6000 Programming Locking Applications*, SC23-2704
- *HACMP/6000 Programming Client Applications*, SC23-2705
- *HACMP/6000 Master Index and Glossary*, SC23-2707
- *HACMP/6000 Licensed Program Specification*, GC23-2698

HACMP 4.1 for AIX

- *HACMP 4.1 for AIX: Concepts and Facilities*, SC23-2767
- *HACMP 4.1 for AIX: Planning Guide*, SC23-2768
- *HACMP 4.1 for AIX: Installation Guide*, SC23-2769
- *HACMP 4.1 for AIX: Administration Guide*, SC23-2770
- *HACMP 4.1 for AIX: Troubleshooting Guide*, SC23-2771
- *HACMP 4.1 for AIX: Programming Locking Applications*, SC23-2772
- *HACMP 4.1 for AIX: Programming Client Applications*, SC23-2773
- *HACMP 4.1 for AIX: Master Index and Glossary*, SC23-2774
- *HACMP 4.1 for AIX: Licensed Program Specification*, GC23-2766

International Technical Support Organization Publications

- *HACMP/6000 Customization Examples*, SG24-4498
- *HACMP/6000 Mode 3 Implementation*, GG24-3685

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office. For guidance on ordering, send a PROFS note to BOOKSHOP at DKIBMVM1 or email to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called BOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following URL:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. The internal Redbooks home page may be found at the following URL:

```
http://w3.itsc.pok.ibm.com/redbooks/redbooks.html
```

Acknowledgments

This project was designed and managed by:

David Thiessen
International Technical Support Organization, Austin Center

The authors of this document are:

Andrew Beyer
IBM Australia

Rahul Bhattacharya
Tata Information Systems Ltd. (India)
An IBM and Tata Company

This document is an update and replacement for the document "High Availability Strategies for AIX" (GG24-3684).

The authors of the previous document were:

Mark Watson
IBM United Kingdom

Charlotte Brooks
IBM Australia

Bell Chang
IBM Taiwan

Ronald Daems
IBM Belgium

Miguel Crisanto
IBM Germany

John Easton
IBM United Kingdom

The advisors for the previous document were:

David Thiessen
International Technical Support Organization, Austin Center

Mark Johnson
IBM Australia

This publication is the result of a residency conducted at the International Technical Support Organization, Austin Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Marcus Brewer
International Technical Support Organization, Austin Center

Laurene Jacob
International Technical Support Organization, Austin Center

Cindy Barrett
IBM Austin

Tom Weaver
IBM Austin

Nadim Tabassum
IBM France

Chapter 1. Introduction to High Availability

When an organization purchases a new computer system, it is investing resources, both financial and human, in a new asset. This new asset will, like any other asset, require care and attention to provide its maximum return. A key attribute that influences the level of return that a system can provide is its level of *availability*. Availability is simply the proportion of the time that a system is able to be used for its intended purpose.

Availability has become a significant issue for many companies today. With computerized applications becoming more critical to business operations, the extent to which companies rely on their computer systems, has never been greater. The amount of availability a system provides is dependent on a range of issues. This book is an update to the publication entitled *High Availability Strategies for AIX*. It builds on the concepts and practices covered in the previous edition with new and updated information and focuses on technical topics relating, in particular, to improving system availability for the RISC System/6000 family. The general topic areas are:

- Single system availability. This discussion includes the features of AIX that provide a foundation for high availability and the ways that these can be enhanced further with sound system management disciplines.
- Clustering of RISC System/6000s for high availability.
- Hardware options for high availability featuring the RISC System/6000's *storage technologies*, which are the foundation for many availability approaches. The RISC System/6000's connectivity options are also featured in this edition.
- IBM's *availability management* solution for AIX Version 3.2, called High Availability Cluster Multi-Processing/6000, including how to set up a cluster, new chapters on cluster tuning, and tips for cluster implementation and administration.

Note

You should note that this book deals with *non-concurrent access clusters* only, and that concurrent access clusters are handled in a different document.

1.1 Introduction to Availability

It is important to clearly define some key terminology. The terminology of availability is loosely used in the computer industry. It is not always evident what is meant in a given context. To help ensure clarity, we will define the key terms here:

Availability Availability is a measure of the degree to which a system can be used for its intended purposes during the times required by the business.

Service Level Service level is the goal or target level of availability defined for a system. A service level could be negotiated between the users and managers of a system, or mandated by management for a particular purpose. Service levels are generally justified against cost and resource considerations. Service levels are defined in such terms as level of

availability, responsiveness, maximum permissible downtime, or maximum number of system outages over a specified period.

Outage An outage is any planned or unplanned loss of service. Unplanned outages are generally caused by defects in, or failure of, system components. Planned outages are generally periods of time scheduled for systems management activities.

Recovery Recovery is the process of restoring service after an unplanned outage. Recovery time is a key element of availability.

Storage For the purposes of this document we will define storage as all persistent forms of storage. This excludes a system's real memory or RAM, but includes disks and tapes.

Backup Backup is the process of copying selected information to some removable form of storage, so it can be retrieved later in case of a failure.

See the Glossary for definitions of other key terms.

1.1.1 Why is Availability Important?

Today, computer systems are a critical part of many businesses. It is difficult to imagine reserving a seat for a flight or taking cash from your bank account at 2:00 AM without the support of computer systems. Once a level of computer system function is available, its consistent availability can become key to a business. A reduction of system availability will incur costs to the business (either direct or indirect), taking one or more of these forms:

Direct revenue loss If system availability has a direct effect on a business's ability to take revenue, then revenue will be lost while the system is down. For example, if a business that sells tickets to entertainment and sports events cannot print tickets, then customers, and revenue may well be lost. If a newspaper publisher cannot create its daily layouts, then it may miss the next street delivery and hence, lose revenue.

Staff productivity The staff dependent on a system will often be unable to perform any useful work when a system is not available. Beyond this, they may not immediately return to their system-related tasks when the system returns to service. Frequent and lengthy system outages will also tend to induce frustration in the users, causing them to think of the system as an impediment to their work rather than an aid. They may even spend valuable time convincing others to share that view. This results in a loss of productivity and morale.

Service levels If a business transaction cannot be performed because the system is unavailable, customers may use an alternative source. Many potential customers for a particular service may also base their choice of a provider on the perceived service level of each. Lower service levels generally lead to lower levels of customer satisfaction, lower perceived quality and often loss of customers.

Circumvention costs When a system and its supported business processes are unavailable, it may be possible to provide an alternative solution, rather than simply denying service altogether. There will probably be additional costs associated with such solutions. In the ticket sales example mentioned previously, the solution may be to take the customers' money and then mail the tickets to them. This adds the costs of mailing to the business process in exchange for no additional revenue.

Before an availability strategy is developed, there must be an appreciation of the value of availability and the cost of reduced availability for each specific application to the business. This knowledge should be used to make judgements on the amount of resource to be invested in availability solutions. Inevitably, trade-offs must be made between the cost of the solution and the cost of an outage. This can only be done wisely if each system's value to the business, and the impact of its loss for a period of time, is well understood.

An availability strategy must use, as its first consideration, the level of service that is required for an application or system. That service level defines the target levels of availability. Without this knowledge, it is very difficult to make reasonable business judgements about the variety of availability alternatives that can be selected.

1.2 Levels of Availability

Availability can be seen as a continuum, where each point along the axis has an associated cost, and where improvements can be obtained through investment of additional resource or technology. Computer vendors seek to ensure that the systems they install have reliable hardware and software to provide reasonable levels of availability without additional investment. In general, for a given system, the greater the level of availability desired, the greater the costs associated with achieving it.

There are many terms used in the computer industry to define levels of availability. For the purposes of discussion in this document, we will define four major levels of availability:

1. Base Availability

   Base availability is the level of availability achieved with a single system and basic systems management practices in place. For many people, this is a sufficient level of availability. Note that a basic set of systems management procedures is required to achieve this level of availability.

   Normal cycles of operation should include planned system outages for systems management tasks. It should also be expected that such a system will have occasional unplanned outages; hence recovery procedures should be in place. Expected recovery times from failures would range from a few minutes, for a problem requiring just a system reboot, to a day for severe hardware or environmental problems.

   Selecting reliable hardware and software technology will help to provide good base availability. When planning for good base availability, one should evaluate both the proven hardware reliability of a system and any built-in

availability features that may be implemented in software. Also, the responsiveness and quality of support provided by the vendor should be factored into a decision for good base availability.

2. Improved Availability

   Improved availability systems provide greater robustness through the application of some additional technology or resource to one or more system components. This additional technology provides greater availability at a greater cost than a similar base availability system.

   Techniques such as disk mirroring, the use of an Uninterruptible Power Supply (UPS), redundant components, data journaling and checksumming, hot pluggable disks, and disk sharing can each be used to help overcome certain system failures. For example, a system with mirrored disk subsystems will offer improved availability over one with non-mirrored disks, because it can overcome certain disk failures.

   The goal of this environment is not to provide continuous uninterrupted service, but to try to ensure that system outages are primarily planned outages. However, since a single system is being used, unplanned failures will inevitably occur. The system should be designed for faster recovery from failure conditions than a base system. A more rigorous systems management strategy should also be in place to complement the investment in hardware. Recovery times in this environment would typically range from a few minutes to a few hours, and failure rates would be lower than in the base availability scenario.

3. High Availability

   High availability systems attempt to provide a continuous service within a particular operational window by minimizing the causes of failure and minimizing recovery time when failure occurs. Generally, this requires a large degree of redundancy in system components so that the continued operation of the entire system is protected from the failure of any one component. Providing this level of protection eliminates these *single points of failure*. The ultimate objective is to eliminate all single points of failure in the system. This can be accomplished by having redundant components or systems, and availability management technology that can automate the transfer of services to those redundant components or systems if a failure occurs.

   In this environment, it is crucial to ensure that the recovery time from any unplanned outage is minimal. These systems are still likely to require some planned outages for systems management purposes, but these should occur outside the operational window. Recovery times in this scenario should be in the order of tens to hundreds of seconds. If applications are written appropriately, users may not actually see this loss of service as anything other than a longer than average response time.

   To achieve this level of availability, a significant investment in systems hardware must be made, and a very strict and rigorous systems management regime must be designed and in place. Also, availability management technology should be in use to help automate the recovery process and to minimize the recovery times.

4. Continuous Availability

   At this level of availability, the system never fails to deliver its service. These systems attempt to provide 100% availability to the end user by providing both

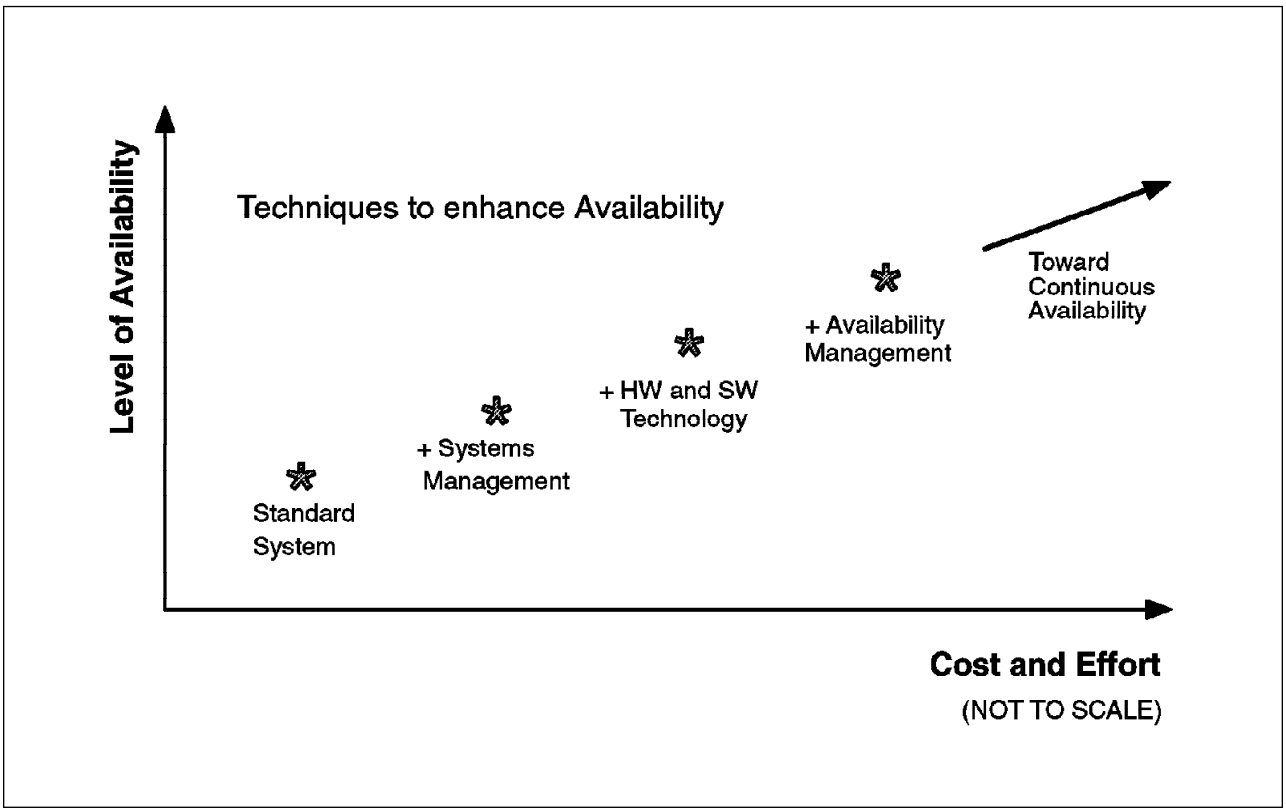


Figure 1. Availability Continuum

redundancy in components and the ability to perform all error recovery and change processes online. In this scenario, planned outages may occur, but they should not be apparent to the end user.

Fault Tolerance

Fault tolerance is not really a level of availability, but rather a characteristic of a system. It is also a term used by some vendors of high or continuous availability systems. Fault tolerance describes a system whose use of redundant components makes it impervious to component failures. What is generally called a “fault tolerant system” is usually a processor or a server. It is important to remember, however, that a fault tolerant processor cannot guarantee that an end user will see continuous availability. This is because there are such things as controllers, cables, and communications lines, all of which can fail, between the processor and that end user.

As mentioned above, the terminology of availability is somewhat loosely used in the industry. The definitions here will probably not be adopted as industry standards, but are provided to ensure that you understand the author's intent when these terms are used in this document.

1.2.1 Availability Continuum

If the levels of availability described are to be achieved, varying levels of technology and techniques will have to be used. Figure 1 on page 5 illustrates this concept, by showing a continuum, where increased levels of availability are balanced against increased levels of cost and effort.

Two points can be made from this diagram. First, increased levels of availability are achieved by both cost and effort. Higher availability cannot just be purchased in a product. It takes much planning and effort to achieve. Secondly, higher levels of availability are dependant on solidly implemented base techniques. Any high availability solution must start with rigorous systems management. If the correct systems management strategy is not in place, a large investment in special purpose hardware and software for availability may be in vain.

1.3 Key Threats to System Availability

There are many potential enemies of system availability. Some of them are outlined below. An effective systems availability plan must consider at least the following possibilities:

Environment and power

To function effectively, a system must have adequate power and operate within its environmental (temperature and humidity) bounds. These environmental requirements are detailed for the RISC System/6000 in the *IBM RISC System/6000 Planning for your System Installation* manual. If a system has no power, clearly, it cannot function. If a system is exposed to extremes of temperature or humidity, its reliability will decrease and the probability of failure will increase. The reliability of the power supply varies widely from place to place. In some locations, an average installation may encounter twenty power outages a year. For example, while this document was being prepared, the IBM office in Austin, Texas in which the authors were working, had four unplanned power outages in a six-week period, and a planned outage scheduled for a week later.

Hardware failure

While computer systems are becoming more and more reliable, hardware failures still occur. The hardware failure of a system component may impact all or part of a system's function. Further, to repair the failed component, the system may need to be shut down, either to run diagnostic procedures or to enable the replacement of hardware components.

Software failure

The operating system and/or applications may fail for a variety of reasons. Sometimes, they fail because of defects and sometimes because of errors in configuration or installation. These failures will affect operations to varying degrees.

Production time may be lost during the attempt to recreate and isolate the problem or during the process of applying updates or fixes.

Communications link failure Communications are a vital link and often the cause of a reduction in system availability. While the system itself may be physically protected, the network exists outside this protection; hence, it is exposed to a variety of potential problems. Wide area networks usually rely on a telecommunications company, for example, to provide the communications links. This variable is beyond the control of the systems manager. The role of networks in systems availability must be carefully considered. The use of network management strategies and products may be necessary to satisfy availability needs. Network availability is a complex topic and will be covered only in passing in this document.

Operational errors Human error is another common source of failures. Accidental or erroneous use of various commands and facilities provided by a system can cause the system to fail.

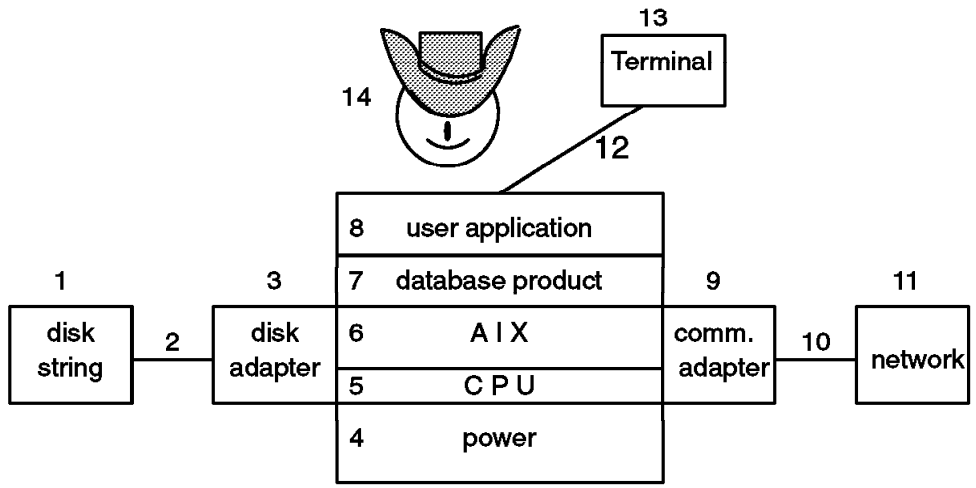
Systems management practice Poor management practices will contribute to greater down time. Unscheduled outages can be caused by such things as inopportune application of changes, poor planning for resilience and failure recovery, untested and unworkable recovery practices, incomplete backup practices, and undocumented system status. Lack of skill by the system manager can be very costly.

1.4 Availability Measurements

There are several metrics used to describe availability, all with a slightly different emphasis. The following are key measures:

Availability Level Availability level is generally expressed as a percentage. For example, 93% availability suggests that a system is available 93% of the time it is required. It is key to note that we have defined availability with respect to *normal operating time*. If a system is required ten hours a day, five days a week, we should measure availability against that operating time frame. A system's availability, or lack of availability, is not an issue when the system is not being used.

Mean Time Between Failure Mean Time Between Failure (MTBF) is a statistical measure of the average number of hours a system or a component of a system will operate before failure occurs. The greater the MTBF, the less likely it is that a failure will occur for that item.



- 14 (Components) at 99.5 % = *93 %* System Availability (SA)
- 13 @ 99.5 % and 1 @ 100 % (Mirrored Disk) = *93.5 %* SA
- 13 @ 99.5 % and 1 @ 95 % (Weak Link) = *88.5 %* SA

Figure 2. Availability is a System-Wide Consideration

Recovery Time

The time to recover from unplanned outages can be crucial. An unplanned outage may be tolerable, if recovery from it can be fast.

1.5 Availability as a Total System Concept

Availability is a system-wide consideration. No single product can guarantee system availability. High levels of availability are accomplished through careful planning, design, and integration of components from all areas of the system. It is important to understand how components interact in a system.

Consider Figure 2. Let us assume each component has an availability of 99.5%. We can calculate the availability of the system from these components:

The point of this exercise is to demonstrate that fixing only one system component may not improve systems availability substantially. Any single very weak component can easily make the system unworkable for its intended purpose.

1.5.1 Failure Rate and Availability

We can examine another scenario. Suppose we have a system like the one in Figure 2. We will consider the disks as just one component. If one disk unit with a MTBF of one year is attached, one disk failure per year on average would be expected. If instead of one disk, there are now twelve disks of the same reliability, it is likely that there will be one disk failure per month. Fortunately, disk MTBF figures these days are in the order of years, but the key point is that the more components there are in a system, the more likely a failure becomes. Assuming all components are key to the system, this has a direct bearing on overall system availability.

To assess the present availability level for any system, it is necessary to identify each of its components and try to assign an expected failure rate for each. Hardware failure rate information is clearly commercially sensitive information and is often only released by special request to the vendor of the equipment. The local power utility may keep statistics on power availability, or the average failure rates may already be generally known. Failures because of human error and software defects are much harder to quantify. Experience has shown that for smaller systems, the most frequent causes of failure are likely to be power, operational, and software defect problems. However, as a system grows and more peripherals (especially disks) are added, the likelihood of system failure resulting from hardware failure, increases. The more “moving parts” a system has, the more likely it is that one of them will break.

Chapter 2. Single System Availability

Most modern computer systems are engineered with features that provide various fundamental levels of availability, or base system availability. Base system availability can be increased further by using effective management practices to take special advantage of these built-in features. The RISC System/6000 and AIX are particularly rich in features that improve system availability. Many of these features are invisible to users and administrators, and only become obvious during system recovery; others can be harnessed to improve availability ever further.

The purpose of this chapter is to describe the features of AIX and RISC System/6000 that enhance availability in a single system environment. This leads to a discussion of system management practices that should be considered when the objective is to improve system availability further still.

2.1 Availability Features of AIX

AIX has various design features that permit many system management and configuration tasks to be performed without the need to take users off-line or to bring the system down. These tasks are also aided by features that help users to avoid errors in performing these tasks. AIX also has features that can vastly shorten the recovery time, should the system halt abnormally. By easing the burden of system management, speeding recovery times and safeguarding users against operational errors, we reduce potential and actual down-time, and maximize system availability.

2.1.1 System Management Interface Tool (SMIT)

The SMIT facility is a tool that helps administrators and users to manage and configure a RISC System/6000 system. While you need to conceptually understand the task to be performed, you do not have to remember commands and options. This lowers the probability of using a wrong or misspelled command. To perform a task through SMIT, you follow the menus until the desired configuration screen is reached. The configuration panel is completed by filling in the blanks. The SMIT facility includes context sensitive help, enforces mandatory fields, and provides lists of the valid options for many fields. All of this is designed to substantially reduce human errors.

Furthermore, SMIT logs activities in a file called *smit.log*. The smit.log file provides an audit trail, easing problem determination and isolation. SMIT also builds a file called *smit.script*, as a shell script, containing each command that has been executed. Using the smit.script file you can create and customize shell scripts containing frequently repeated series of commands.

2.1.2 Logical Volume Manager (LVM)

The Logical Volume Manager provides a simple and flexible mechanism for managing disk storage in AIX. Through SMIT, it allows you to perform tasks such as configuring a new disk to the system, or increasing the size of a filesystem or paging space while the system is online. Refer to the InfoExplorer database for more details on the logical volume manager.

2.1.2.1 Disk Mirroring

Disk mirroring is a feature of the LVM that allows a single logical filesystem to be associated with multiple physical copies in a way that is transparent to users and applications. It means that if a disk, or sectors of a disk, containing one copy of the data should fail, the data will still be accessible from another copy on another disk. Mirroring improves availability by allowing the filesystem to remain available if disks fail, but requires extra disk drives.

AIX provides disk mirroring at a logical volume level. In AIX, you can create and maintain up to three copies of a logical volume (the original and one or two mirrors). Users or applications that access files via standard AIX file manipulation routines are not aware of the fact that the files are mirrored, as AIX provides one logical view of the files.

2.1.2.2 Bad Block Relocation

To enhance availability, it is necessary for a system to be able to handle errors on the disk surfaces. Assignment of alternative disk sectors or bad block relocation is usually done by the disk subsystem. However, the LVM is able to perform bad block relocation if the disk subsystem does not provide this feature.

2.1.3 Journaled Filesystem (JFS)

AIX automatically logs all changes to a filesystem's structure in a logical volume called the journaled filesystem log, or *jfslog*. Each volume group contains at least one jfslog, if there are any filesystems in the volume group. At system restart, the `fsck` command checks the filesystem logs. If an error or inconsistency is discovered, the relevant journaled transactions are replayed to rebuild inconsistent filesystem structures. This represents a significant departure from the methods used to recover a conventional UNIX filesystem. When an unplanned outage occurs, the conventional UNIX system must check the entire filesystem, which could be hundreds of megabytes. This may take hours, or even days, to complete. The jfslog is a four megabyte logical volume that contains the necessary data to correct a filesystem error within minutes, or even seconds. The JFS significantly improves system availability, because it provides fast recovery from a system crash.

2.1.4 Dynamic AIX Kernel

Traditionally, there have been many UNIX systems management tasks which have required a rebuild of the kernel and/or a system reboot to take effect. AIX allows many changes affecting the system kernel, such as an increase of page space or the addition of a new device driver, to be activated while the system is running. Avoiding the need to reboot the system for such changes increases system availability.

AIX uses less static configuration compared with traditional UNIX. Traditional UNIX hard codes many system data structures, statically binding them to the kernel. Changes to these limits require a kernel relink and a system reboot.

Tuning kernel performance is a skill-demanding and time-consuming task. if a limit has been set too low and is exceeded in operation, the system can fail or crash. Setting the limits too high, on the other hand, is wasteful of system resources. AIX allocates only the resources that are needed and extends them dynamically as required. This eliminates the potential for failure because of unavailable resources,

and also the need for the system administration expertise and system downtime that would be necessary to reconfigure and rebuild the kernel.

2.1.5 System Resource Controller (SRC)

The SRC controls many AIX subsystems such as TCP/IP, NFS, SNA, and of course, HACMP. It can automatically handle specific events, such as abnormal termination. Furthermore, the SRC provides a consistent set of commands to start, stop, trace, and query subsystem status, to facilitate their operation.

2.1.6 Configuration Manager

At system startup, AIX automatically configures any devices added to the system. The configuration manager command, `cfgmgr`, may also be executed while the system is operational. This automatic capability significantly reduces the potential for error in the process of hardware configuration.

2.1.7 AIX Update Facilities

AIX allows updates to system software to be applied and tested, and then either committed or rejected. By allowing updates to be applied (not committed), you have a convenient mechanism for testing and ensuring that problems are not introduced by the new code. Applying an update keeps a copy of all the system files replaced, so that they can be called back if needed. If the update code causes problems, you can use the *reject* process to undo the changes and restore the operating system to its previous state. Only when the update is *committed*, are the copies of replaced files erased. This should only be done after you have had the opportunity to test the update, and are satisfied that it does not introduce any new problems.

In AIX Version 3.2, the update distribution process allows you to install or reject selective fixes, enhancements and maintenance levels, meaning that single fixes can be applied and tested individually. For AIX Version 4, the process is essentially unchanged, apart from some new naming conventions. For example, the concept of a *fileset*, replaces that of an *option* and a *subsystem*. The naming convention for PTFs has also changed to a format that consists of the fileset name plus a four field, dot-separated level identifier. For more information on AIX update facilities, refer to the IBM publication called *All About AIX Version 4*, also available by connecting to URL `http://www.austin.ibm.com/developer/aix/` from the IBM Solution Developer Support Home Page on the World Wide Web.

2.2 Availability Features of the RISC System/6000

So far, we have described some of the features of AIX that contribute significantly to enhancing system availability. The RISC System/6000 family also takes advantage of various technologies to improve availability.

2.2.1 Built-In Error Detection and Correction

Hardware and software components can be designed to automatically detect error or failure conditions and to correct some of these errors. Built-in error detection and correction will increase a system's reliability by dealing with problems which could otherwise cause failures. RISC System/6000 memory features single bit error correction, and double bit error detection, and the disks perform automatic bad block relocation.

2.2.2 Backup Power Supply

You can improve the quality and reliability of the local power supply by installing a backup power supply. A backup power supply will generally perform some level of line or power conditioning also.

2.2.2.1 Battery Backup Systems

These systems sense the failure of external power and switch the system to battery supplied power if such a failure occurs. Battery backup systems provide a limited amount of time, during which the system can be sustained on the battery. The costs of such systems increase with battery capacity (usually measured in kilovolt-amps or kVA) and the duration for which backup power can be provided.

In this environment, a customizable system interface to the power supply's electronics is normally provided to shut down the system gracefully as the battery nears the end of its power. Advanced warning of an impending shutdown will not only improve recovery time, compared to a sudden system outage, but it also warns users so that work can be saved and jobs completed. If power returns before the shutdown of the system, the system switches back to regular power and continues working without interruption. Meanwhile, the battery begins to recharge itself, to be ready to deal with the next power failure.

The rack-mounted RISC System/6000 models provide an optional Battery Backup Unit with 1500 watts of standby power, that can keep the system operating on the battery for a minimum of ten minutes.

2.2.2.2 Continuous Backup Power Source

It is also possible to install generator systems that will automatically engage when the standard power supply fails.

It is important to consider the implications of a planned power backup scheme. There is little value in keeping a system powered on if none of the peripheral devices or user terminals have power. The system should be set up so that a subset of the system can remain in operation under power loss conditions, to allow critical application users to continue working as long as possible, or to take a backup before final shutdown.

A backup power supply, also known as an Uninterruptible Power Supply or UPS, will be rated in terms of the amount of continuous power it can provide and the period over which it can provide the power. Some UPSs can be connected to a communications port in the system so they can signal a monitor process when a power failure occurs. Some UPS vendors will supply software to warn users and gracefully shut down the system, if the UPS reaches a battery power drain threshold. In some cases, if the hardware interfaces are present, but the software is not available from the UPS vendor, then it may be possible to write the code.

2.2.3 Power Conditioning

In some areas, the power supply may be reasonably reliable, but may be subject to significant fluctuations in current. The RISC System/6000 power supply will handle a degree of power fluctuation, but in some areas a power conditioning device may also be required. IBM Customer Service representatives should be able to advise on the quality of the local power supply. The details of the power supply variations that a RISC System/6000 can tolerate are contained in the publication *IBM RISC System/6000 Planning for Your System Installation*.

2.2.4 Redundant or Spare Disks

To increase the availability of a system, a spare disk can either be installed in the system or kept nearby to replace a failing disk. Mirroring, in effect, provides this capability automatically. Nevertheless, a spare disk could also be added to a system to provide a backup for any failed volumes. This technique is particularly valuable in installations that are reasonably distant from service locations, or for critical, data-dependent business environments.

2.2.5 Hot Pluggable Disk Drives

Hot pluggable disk drives allow replacement of failed units while the system is online. While this improves system availability, the data contained on the failed disk must be copied to the replacement, either by restoring from a backup, from a mirrored copy, or by the use of parity disks to recreate it (see Section 2.2.7, “RAID Disk Arrays”) before the system can be considered fully available. The 9333 Serial Disk Drive Subsystem supports drive replacement with no requirement to power down any system components. The self-docking and plugging characteristic of the disk means that no cables have to be physically plugged. The power supply provides a separate port to each disk with over-voltage and current protection to provide a high degree of safety during maintenance.

2.2.6 Multi-Tailed Disks and Shared Volume Groups

Multi-tailed disks are disks that are cabled to two or more separate system units. The data contained on a given disk is usually accessed exclusively by one of the systems at any one time. Concurrent data access by more than one system is considered a more specialized implementation. Also known as disk-sharing or bus sharing, this forms the basis for the HACMP capability to take over a disk resource from a failed system. For more information on shared bus and shared disk implementations, refer to Section 3.2.2.1, “Cluster Hardware Components” on page 30.

2.2.7 RAID Disk Arrays

RAID (Redundant Array of Independent Disk) is a disk technology that is designed to provide improved availability, security and performance over conventional disk systems. While appearing logically to the operating system as a single disk drive, a RAID array is actually made up of several disks, which have their data spread across the drives in any of several different methods. You can see this concept illustrated in Figure 3 on page 16.

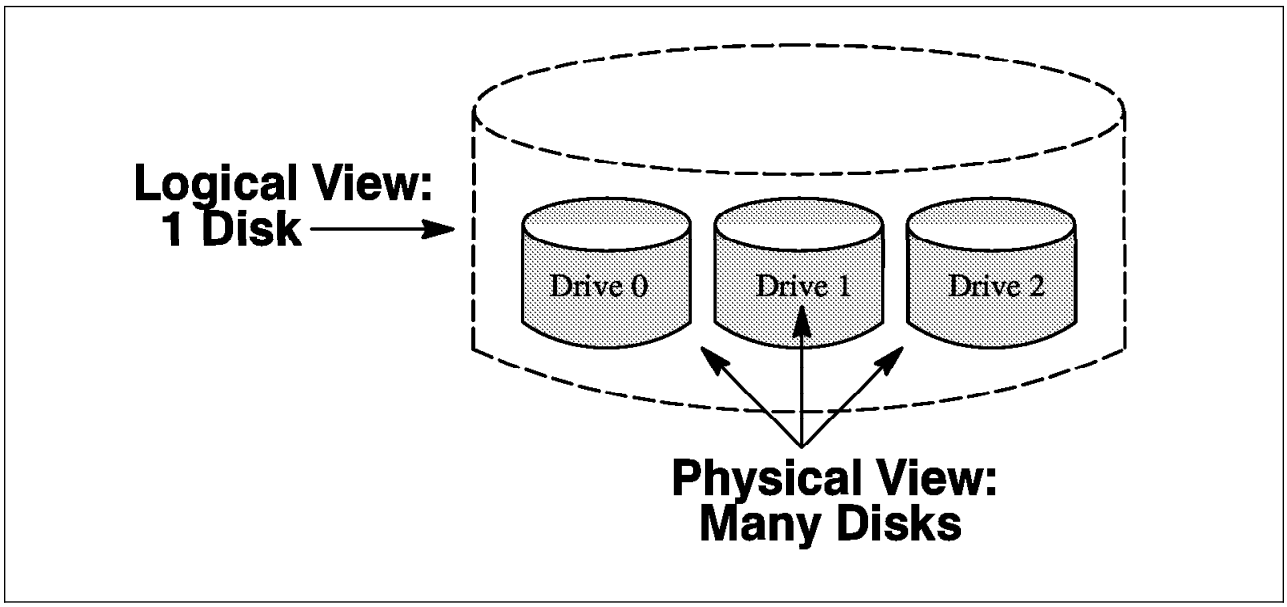


Figure 3. The RAID Concept

There are several different methods, or RAID levels, defined. Not all of these levels are implemented in products from IBM and other vendors, but they are summarized here for your reference. Three of the levels have practical application in commercial computing today, RAID-1, RAID-3, and RAID-5. RAID-2 and RAID-4 are described here for completeness, but have fallen out of favor because of their inherent disadvantages.

2.2.7.1 RAID-0

RAID-0 is not a RAID level that is well suited to applications that require any level of availability. This implementation is designed only for maximum performance. In RAID-0, sectors of user-determined size are written across all disks in the array in a sequential manner, with no mirroring or parity information being kept. This implementation is illustrated in Figure 4.

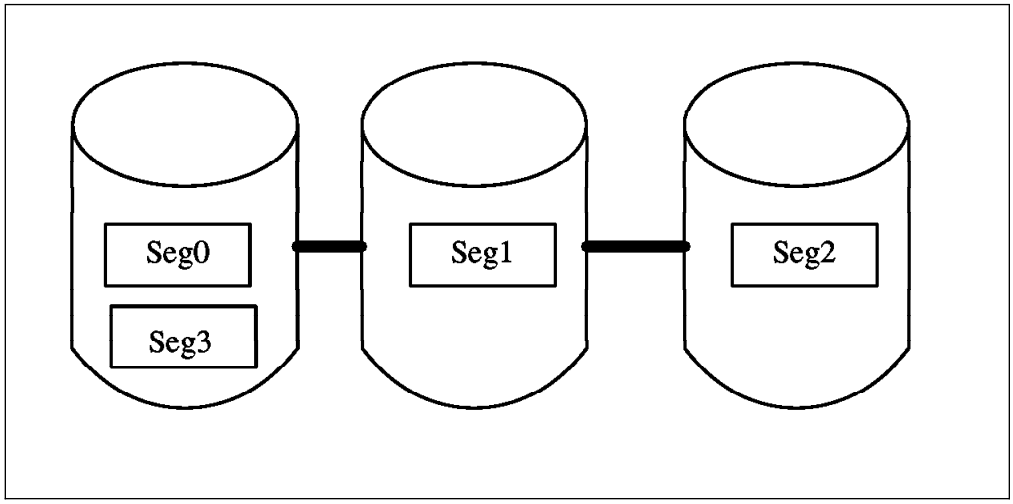


Figure 4. RAID-0 Illustration

Since the data is spread across all disks in the array, performance is enhanced, since the access load is also spread across all disks, and all disks can be active

simultaneously. Availability with this level is very poor, however, since, if one disk in the array fails, the data contained in all disks in the array is effectively lost. For this reason, this implementation is not used in high availability applications.

2.2.7.2 RAID-1

In RAID-1, data is mirrored from one disk drive to another. Disks in the array are grouped in pairs, where the data on one disk is completely duplicated on its pair. Data is written to disk in sectors of user-determined size, where each sector written to one disk is also written to its pair disk. This implementation is illustrated in Figure 5.

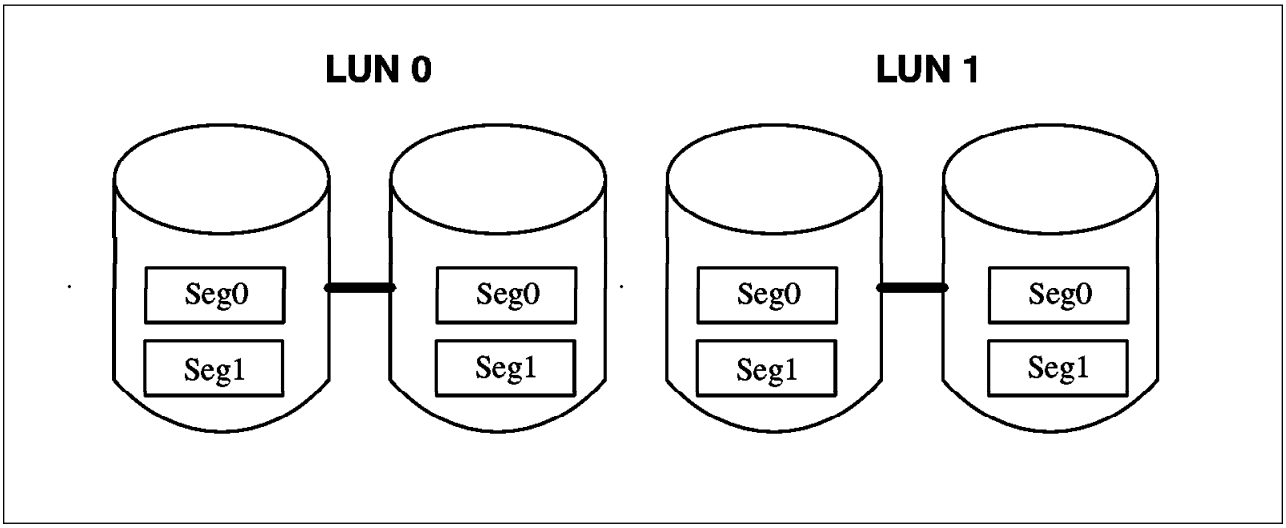


Figure 5. RAID-1 Illustration

This level provides continuous availability to the data, provided one good copy exists, but is relatively expensive, since double the amount of disk is required for all mirrored data. Compared to other RAID levels, RAID-1 can potentially decrease the machine performance. Each write request must now be performed “n” times, where n is the number of copies of the data. Note however, that read performance can be improved with disk mirroring, since mirrored read requests are dispatched to all drive controllers simultaneously. This disk which is able to service the request first will return the data. AIX provides a type of RAID-1 through the logical volume manager It attempts to overcome the potential performance impact by giving you the option of doing your writes in parallel, where all the mirrored writes are dispatched at the same time, without waiting for one to complete successfully. This is a less secure, but better performing method than the sequential write method, which may alternatively be configured. With the sequential scheme, the mirrored copy is not dispatched until the primary copy has been written successfully.

2.2.7.3 RAID-2

In RAID-2, data is interleaved across the disks on a bit-by-bit basis, and check disks are used to correct and recover from any errors. This approach requires large disk groups to maintain consistency (four check disks for 10 data disks, five check disks for 25 data disks) and has the disadvantage that all disk drives must be accessed for every I/O operation. Only one copy of the data is maintained. For the above reasons, RAID-2 is seldom supported in modern RAID implementations.

2.2.7.4 RAID-3

In RAID level 3, data is striped, on a byte-by-byte basis, across three or more drives in the array. Parity information is maintained on a dedicated disk. A single parity disk can protect up to four data disks, an overhead of twenty percent. The parity information (the Exclusive-Or of the data) is used to restore the data if a drive failure occurs. This implementation is illustrated in Figure 6.

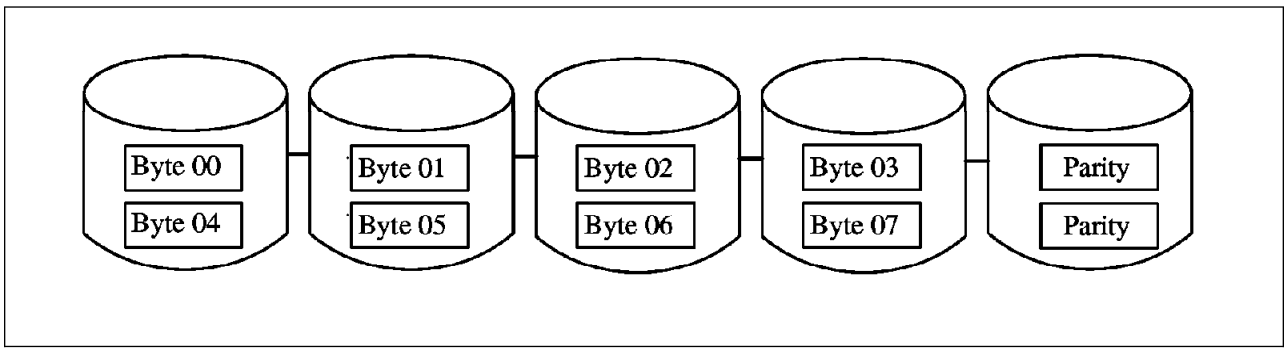


Figure 6. RAID-3 Illustration

Since data transfers to and from individual drives occur only in unit-sector multiples, the minimum amount of data that can be written to, or read from, a RAID-3 disk array is the number of data drives multiplied by the number of bytes per sector. This is known as a transfer unit.

In RAID-3, since the multiple drives in the disk array are written or read simultaneously, extremely fast data transfer rates can be achieved. This is particularly true when the size of the data being written or read is at least the size of the transfer unit. The performance of a RAID-3 implementation slows down considerably when many smaller data transfers are taking place, such as might occur in a transaction processing application. For data reads, smaller than the transfer unit, all the data in a transfer unit must be read anyway, which reduces the efficiency (similar to RAID-2).

Write operations will also be inefficient where the data writes are smaller than a transfer unit. The disk array must deal with complete transfer units even though only a small portion of data must be updated. A complete transfer unit must be read from the combined data disks, the data must be modified where appropriate, and then the entire transfer unit must be written back to the data disks, with the check disk being updated appropriately.

Select RAID-3 for applications that process large blocks of data. RAID-3 provides redundancy without the high overhead incurred by the mirroring in RAID-1. Data can be reconstructed from the check disk, and with hot pluggable drives, provides very good recovery from disk failure.

The one drawback with RAID-3 is that it is not supported by the LVM. Data in a RAID-3 array cannot be accessed as part of a volume group, even in a raw logical volume. It can only be accessed through the raw hdisk device.

2.2.7.5 RAID-4

This is similar to RAID-3, but in RAID-4, the data is written one block per disk, so that only one disk needs to be accessed for a read request. This means there is the capability for parallel reads, but writes must all access the parity drive, creating the potential for a bottleneck. Unlike RAID-3, rebuilding of data cannot be done online. For this reason, RAID-4 is seldom supported in modern RAID implementations.

2.2.7.6 RAID-5

In RAID-5, data is striped across the drives of the array in segments, and parity information is maintained. Instead of using a dedicated disk to store the parity information, as in RAID-3, RAID-5 dedicates the equivalent of one entire disk for storing check data, but distributes the parity information across all the drives in the group. This implementation is illustrated in Figure 7.

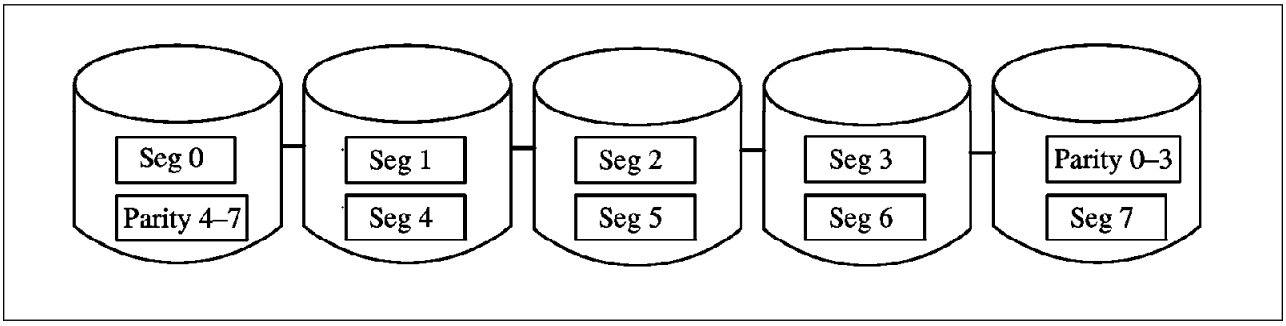


Figure 7. RAID-5 Illustration

Only two disks are involved in a single-sector write operation: the target data disk and the corresponding disk that holds the check data for that sector. This is in contrast to the RAID-3 implementation, which requires all drives in a group to be read and written when a single sector write operation occurs.

The primary benefit of the RAID-5 distributed check-data approach is that it permits multiple write operations to take place simultaneously. It also allows multiple reads to take place simultaneously and is efficient in handling small amounts of information.

Select RAID-5 for applications that manipulate small amounts of data, such as transaction processing applications.

2.3 Improved Availability Through System Management Practices

Earlier, we described the features, of the RISC System/6000 and AIX, which provide the basis for the enhanced availability of a single system.

Neither redundant system components, nor high availability design features will yield the expected system availability improvement unless effective system management is defined, developed, and used in all areas of the system. Effective system management processes are fundamental to availability management. Figure 1 on page 5 illustrates the recommended sequence for addressing availability.

After beginning with reliable base technologies and products, effective system management disciplines should be defined and used. Only after these are in place

should more sophisticated (and costly) high availability techniques and products be considered. It is necessary to proceed in this order, because the more sophisticated approaches rely on the assumption that the base technology and the system management discipline is already sound and robust.

Following the selection of reliable hardware and software, the first imperative to high availability is effective systems management. Implementation of the following systems management processes will make a sizeable contribution to availability improvement.

2.3.1 Skills Management

Before implementing any technology or systems management regimes, the skills to form and execute plans effectively must be present within the organization. This area is most important in improving system availability. Any miscalculations in this area could have damaging effects for the business.

The systems availability planner must specify an education plan for the people who will administer and maintain the system. The system administrators must have the correct skill level to deal with many situations affecting system availability. The skill level of staff involved in implementing and operating a system will directly affect the rate of human error and its impact on operations.

If a system can only tolerate maximum outages of a few minutes, then skilled people will be required, who can respond effectively and efficiently to the pressures of a failure situation. Conversely, an unskilled staff will pose a great risk in any operational environment, and can increase the recovery time if failure occurs. A system outage, which occurs or is exacerbated by human error, is frustrating to both users and management because it is an outage which could have been avoided.

2.3.2 Operations Management

The resources to manage local systems, remote systems and networks must be properly coordinated and monitored. As mentioned before, availability is a total system concern. All systems components must be managed together. In particular, planned outages must be coordinated across all areas of the system's operation. It is also important to ensure that non-essential processing is not scheduled during the times when availability is most critical.

2.3.3 Capacity Management

Capacity management is the process of planning, controlling, and ensuring the capacity of computer resources to satisfy current and future needs. Adequate capacity is required to provide sufficient levels of responsiveness and to prevent failures caused by lack of systems resources. A common reason for operational failure of a system is the exhaustion of allocated disk storage. Systems cannot function if a key disk area is full or has inadequate space to write information. Lack of disk resources can also prevent processes such as logging, printing and editing from being effective.

Continued monitoring of systems performance, and projection of future loads can allow an organization to acquire additional systems resources before they are needed. A new disk drive will be of diminished value if it is ordered just after an application failure has been caused through lack of disk resources.

2.3.4 Change Management

Modern computer systems involve a highly complex system of interactions. Any time a change is made to a component of a system, the behavior of that component, or its interactions with other components, may be different. A program that previously functioned without error may now fail. This is why it is crucial that changes to the system are carried out in a managed fashion.

All changes to hardware, software, facilities, procedures, or networks affecting service delivery must be planned, tested, documented, and coordinated before they are committed in the production environment. The greater the level of availability desired, the more rigorous this process must be.

It is a very good idea for system administrators to keep a log book (and possibly also a file on the system) where all changes to the system are documented. For example:

```
10/12/95  - Replaced SCSI adapter, ran diagnostics
10/25/95  - Router added to the network
```

This log can assist you in investigating the possible causes for error conditions that do not immediately develop. In addition to testing changes before applying them in production, it is also essential that all changes are implemented in a manner that allows them to be easily backed out if necessary. The `updatep` facility of AIX, described in 2.1.7, “AIX Update Facilities” on page 13, assists greatly in this aspect of change management.

2.3.5 System Test Environment

A key to enhancing availability is to find potential problems in new applications and systems before using them in production. A rigorous testing environment is essential if a system is to approach continuous availability. Unfortunately, in many installations, the first test of compatibility for changes to applications or system software, is on the live production system. This involves substantial risks which cannot be tolerated where high availability is required.

As higher availability levels are demanded, it is also probable that the environment will include the added complexities of special-purpose hardware and software components or availability management technology. So, in addition to testing the application and the system environment, hardware and software that provides failure circumvention, fast recovery, and automation also requires testing. One must guard against the worst case scenario, where a change is implemented which not only crashes the operational system, but also causes the automated recovery process to fail.

An effective system test environment can be used for stress testing, performance testing, and automation testing. To really test the system, it must be “stretched” and sometimes deliberately “broken” to discover all possible weaknesses. Obviously this should not be done in the production environment. The higher the levels of availability which are to be targeted, the more rigorous testing must be.

2.3.6 Performance Management

Performance management is the process of planning, defining, measuring, analyzing, reporting, and tuning the performance of computer services. Measurement of component availability and response time is important. The process of performance management interacts with other areas such as capacity management and operation management.

While poor performance may not affect the total number of hours a system is available (its absolute availability), poor user response time or reduced throughput reduces the effective availability. If a system is available eight hours a day, but can only process 2,000 transactions rather than a required 4,000, the system is effectively only 50% available for that period. Availability measurements should also take account of periods where systems exceed agreed performance levels when computing the total system availability.

2.3.7 Problem Management

Problem management is the process of detecting, reporting, tracking, and correcting problems which impact service delivery. Since problems reduce system availability, prompt isolation, minimization, and resolution is vital. A regular analysis and review of problem reports is also important, so recurrences can be prevented. This analysis may lead to changes in the relevant management procedures.

2.3.8 Service Level Management

Customer service levels must be well defined. The integration of requirements, forecasts, services available, and costs must be agreed between the application owner, application user (or user representative) and information system supplier of service. Service levels should be reviewed and revised periodically, so they reflect current business requirements.

2.3.9 Automated Operations

Human error has a significant impact on system availability. It is possible to establish processes that will automatically handle, or greatly simplify, various system operations. Automation can be addressed in many ways. At the lowest level, systems management commands can be simplified by use of menus of tested procedures that execute a series of commands in the appropriate order and with the appropriate input. At the next level, software can be used to monitor system activity, and can be tailored to invoke system procedures when certain activities occur. Finally, software systems can monitor console or operator screens and respond to many common errors automatically. This technique is not a common one in the AIX environment because many systems are already designed to function without an operator's intervention. Any of these approaches can reduce human error to a large degree.

The following areas can be considered as candidates for potential automation:

- Console and network operations
- Application and subsystem operations
- Startup and shutdown
- Recovery
- Switching (adapters, links, modems)

The benefits of automated operations include the following:

- Reduction of human errors through reduced demands on operational personnel
- Fast recovery and reconfiguration when automated
- Repeatable procedures
- Faster problem detection
- Continuous system monitoring

2.4 Isolation

Isolation is a technique that can be used to improve availability of critical functions by partitioning them logically or physically. This can be done using hardware or software components. In Section 1.5, “Availability as a Total System Concept” on page 8, we discussed the interaction of systems components to provide a total systems availability. From this you can deduce that if a system is designed to be isolated from some failures, then availability can be improved.

In this approach, for example, a key application that requires very high levels of availability might be implemented on a different system from the remainder of the applications. The benefits of this isolation technique are:

- It minimizes the exposure of critical components to the risk associated with changes to non-critical areas. This creates a stable environment for the critical applications.
- It reduces costs associated with higher availability. This is done by allowing systems with lower availability requirements to be implemented in a lower cost, lower availability regime.
- The less critical applications will not contend for resources directly with the critical applications. This will make the environment more manageable and predictable.
- It creates simpler system test scenarios.
- It allows more orderly and stable migration of system, subsystem, and application software (from test, to non-critical production, to critical production).

2.4.1 Recovery Management

Restoring normal computer services to the user, after a failure occurs, is one of the most critical system management tasks. This process must be well planned and tested. The recovery procedures should be automated where possible, clearly documented, reviewed and approved by management, tested and updated, and show maximum recovery time. Tasks must be defined for local failures, remote failures, and disaster recovery.

The following application categories can be defined:

Critical This is when recovery must be done in minutes or less and no loss of messages, transactions, or data can be tolerated. For this kind of application or data, the use of a high availability product like HACMP may be necessary.

Essential	This is when recovery must be done in hours or days. For this kind of application data, an effective system management strategy and the use of availability features such as mirroring may be sufficient.
Non-essential	This is when recovery can be done in days or weeks. A controlled management of system backups may be enough.

2.5 Summary

In general, the better the systems management processes, the more available the systems will be. It must be re-emphasized that systems management is the foundation for any approach that attempts to increase systems availability.

In the next chapter, we will expand our discussion beyond single system availability, and begin to discuss the concept of clustering systems together.

Chapter 3. Clustering RISC System/6000s for High Availability

The level of availability provided by a single, well-managed RISC System/6000 (RS/6000) in a stable environment is often not enough to meet the requirements of an installation. In this situation, you can use more than one RS/6000 with or without the support of layered software products to achieve the desired level of availability.

It is important to keep in mind whether your requirement is for *high availability* or *fault tolerance*. The aim of fault tolerance is to provide continuous availability, with no scheduled or unscheduled outages. A highly available system may be required to be shut down at scheduled times for legitimate reasons, such as component replacement, and system maintenance. Fault tolerant systems provide a higher level of availability, but at a much higher cost than highly available systems.

If it is high availability that you are looking for in your environment, there are two ways (in increasing order of availability) in which you can use multiple RS/6000s to implement it:

- By clustering the RS/6000s without any specialized software.
- By using the High Availability Cluster Multi-Processing/6000 (HACMP) software product on a cluster of RS/6000s.

Depending on your availability requirements, one or the other of these methods will be suitable. It is up to you to decide which of these options is better suited to the installation's high availability requirement. After you have gone through this chapter, you should be able to decide which of the above methods to choose. This chapter describes and compares the options to allow you to make an effective decision on the correct solution to implement.

3.1 An Introduction to Clustering

This section describes how a cluster of RS/6000s can be configured to provide a certain level of high availability to the end users. It talks about services that are provided by a cluster and the restoration of these services to end users in the case of a failure.

3.1.1 What Is a Cluster?

A cluster is an set of independent processors, (for the purpose of our discussion, these will be RS/6000s) connected over a network, almost always with external storage devices connected to the processors on a common I/O bus. You can look at a cluster as a black box which provides certain services, critical and noncritical, to end users. A cluster contains resources, such as an interface (Local Area Network or asynchronous) over which users access the service provided, applications that the users execute, and the data that is either used or generated by these applications. The unavailability of any of these resources will result in the unavailability of some or all of the cluster's services to the end user.

Like any ideal black box system, the end user need not be aware of which processor in the cluster he is connecting to, or how and where his application

executables and data are stored. This is dependent on the design of the application, and on how connectivity between end users and the clustered processors is achieved.

3.1.2 Single Points of Failure

A cluster is made up of many components that are essential for providing services to the end user. If the failure of any single component in a cluster results in the unavailability of service to the end user, this component is called a *single point of failure* or SPOF for the cluster.

The possible single points of failure that a cluster could have are:

- Individual processors or nodes in the cluster
- Disks used to store application executables or data
- Adapters, controllers and cables used to connect the nodes to the disks
- Network adapters attached to each node
- The network backbones over which the users are accessing the cluster nodes
- Asynchronous adapters
- Application programs

The aim of any cluster designed for high availability should be to eliminate all single points of failure for *critical* services provided by the cluster. We say critical services because the elimination of a single point of failure always has a cost associated with it. You should only attempt to make a service highly available if the cost of losing the service is greater than the cost of protecting it.

3.1.3 Eliminating Single Points of Failure in a Cluster

It is possible to eliminate most single points of failure without using any layered software product. As you go through this section, try to calculate the amount of time, effort, and human intervention that would be required to recover from any particular failure. Then, consider whether your environment can withstand this amount of downtime, and can provide the necessary human intervention during all hours of operation. This will help you in deciding whether you need another software product to meet your availability requirements or whether you can manage with manual intervention.

Node Failure: When a node providing critical services in a cluster fails, another node in the cluster must be ready to take over its resources and provide the same services to the end users.

This involves the following steps:

- Configuring a network adapter on the backup node with the address of the failed node, or instructing the users (or changing the client applications) to use an alternate address.
- Importing and varying on all volume groups on the common bus between the failed and the backup nodes, and mounting all necessary file systems.
- Restoring the most recent backup of all application executables and data stored in the failed node's internal disks.
- Starting any critical application programs.

We have assumed that the backup node is not already being accessed over the network for critical services. In that case, you will need one extra network adapter for every node that this node will back up. If the users were accessing the failed node over serial connections, each terminal would have to be physically reconnected to a port on the backup node. If the external disks were not connected on a common bus between the failed and backup nodes, you would need to physically shift them from the one to the other. All critical data would have to be stored on external disks configured as one or more volume groups, separate from the rootvg. Any critical data stored on the internal disks of a node since the last backup would not be available if that node were to fail.

Shared Disk Note

It is possible to cable disks to more than one system in a cluster. This is a central part of the configuration of HACMP, to be described later, but it can also be implemented without HACMP. If you have a set of these shared disks, and are not running HACMP on your systems, and one of your systems fails with the shared disks (or more exactly, their volume groups) varied on, the other systems will not be able to access them. This is because the failed system will still be holding a SCSI RESERVE condition on the disks involved, which does not allow access from any other system. To break this condition, and continue, you may have to cycle the power on the disk(s), disk subsystems, or perhaps even the surviving system. HACMP provides routines, as part of its event scripts, that remove the RESERVE condition on shared disks in this situation.

Disks and I/O Bus Failure: To protect against the failure of any part of an external I/O channel including your disks, you should have your disks mirrored across two I/O buses or use RAID storage subsystems with dual paths from the nodes to the subsystems.

For a brief description of RAID, see 2.2.7, “RAID Disk Arrays” on page 15. For a description of RAID products available from IBM see 4.3.4, “RAID Disk Array Options” on page 68.

Choosing the right storage technology and product depends on several factors which are dealt with in detail in 4.3.6, “Choosing a Shared Disk Technology” on page 75. For instructions on how to connect external disks on a common I/O bus refer to Appendix B, “Disk Setup in an HACMP Cluster” on page 209.

Network Adapter Failure: To protect against network adapter failure, a second network adapter would be configured on each node providing critical services. This adapter would be connected to the same network backbone over which the users are accessing the main adapter. If a network adapter fails, you would change the address of this adapter to the address of the failed adapter. Another option would be to always have a spare network adapter available to replace a faulty adapter. With this option, the time taken to recover from the failure would be higher, since the system would have to be shut down to install the replacement adapter.

Network Failure: If the network backbone over which the users are communicating with a node stops functioning, one of the solutions is to manually switch all cluster nodes and client machines to a different backbone. Given the time and effort required to do this, you would want to ensure that there was no loose connection or network device (router, hub, or bridge) failure which caused the

backbone to fail. Another solution would be to connect a subset of the terminals to available serial ports on the backup nodes so that a minimum level of service could be provided. In this case, the applications would have to be designed to allow users to connect through terminals on serial ports as well as through network connections.

Failure of Asynchronous Adapter: One way to avoid the impact of the failure of an asynchronous adapter is to attach the terminals to a terminal server connected to the network instead. Here too, the application would have to be designed to allow connection either from a direct terminal attachment (on a tty) or from a network connection (on a pseudo-terminal or pty).

Application Failure: Depending on the design of your application, you could use the AIX Subsystem Resource Controller, to monitor the daemons used by your application, and to react to changes in their status.

For a description of the Subsystem Resource Controller, please see 2.1.5, “System Resource Controller (SRC)” on page 13.

3.1.4 Disadvantages of Manual Intervention

It should be obvious from the preceding discussion that the time to recover manually from any failure in a cluster could range between thirty minutes and several hours, depending on the type of failure. This excludes the time taken to detect the failure in the first place. If your environment can tolerate such outages, you do not need to use any special software product to meet your availability needs. However, if your requirement is for outages of not more than a few minutes, you should consider the HACMP product from IBM.

This product is described in detail in the next part of this chapter.

3.2 High Availability Cluster Multi-Processing/6000

High Availability Cluster Multi-Processing/6000 Version 3.1 has two distinct subsystems, which can be purchased together or separately. The high availability subsystem (feature 5050) provides a highly available environment on a cluster of RS/6000s. This provides the function to allow independent nodes, each running separate applications and accessing separate data, to provide failure protection for each other. The loosely coupled multi-processing or Concurrent Resource Manager (CRM) subsystem (feature 5051) allows two or more machines to concurrently access the same data and run the same application, also providing failure protection for each other. This redbook primarily covers the high availability subsystem.

HACMP extends the clustering model that we have discussed in Section 3.1.1, “What Is a Cluster?” on page 25 by providing an automated highly available environment for mission-critical applications. HACMP first identifies a set of cluster resources essential to providing a critical service. Cluster resources can include both hardware and software. They can be such things as disks, volume groups, file systems, network addresses, and applications. HACMP next defines relationships between cluster nodes, defining the role that each one will play in protecting the critical resources.

HACMP includes an agent program, called the Cluster Manager, running on each node. The Cluster Manager runs as a daemon (clstrmgr) in the background and is responsible for monitoring and managing the cluster.

How the cluster reacts to any of a number of cluster events is determined by shell scripts, called event scripts, that you can modify to suit your particular requirements.

The HACMP software supports a wide range of cluster configurations, allowing flexibility in building a cluster that meets your processing and availability requirements. While planning an HACMP cluster, you should always aim at a no single point of failure design. This means that the cluster is designed so that the failure of any single component cannot cause the unavailability of a critical service. Through a combination of hardware redundancy, AIX function, and customized shell scripts, you can design an HACMP cluster to quickly recover from the failure of any one of its components.

It is possible, and even likely, that you will need to bring down a cluster node at some point for the replacement of a failed component. This action is what we would call a *planned outage*. The value of HACMP is that it allows you to schedule these outages at a convenient time rather than to have them happen immediately when the component fails. Since the initial CPU or component failure is covered by the activation of one of HACMP's event scripts, and, depending on the type of failure, the interruption of service can be negligible or a matter of minutes, this otherwise *unplanned outage* is avoided, and the failure can be dealt with in a scheduled way.

An HACMP cluster is most effectively used for providing database or online transaction processing services to client applications. In this environment, you can design and implement a cluster to make a system failure transparent to end users.

3.2.1 HACMP Technical Overview

An HACMP cluster consists of up to eight cluster nodes, one or more strings of shared disks, a series of network interfaces on one or more networks, and the HACMP software. Application services on any of the cluster nodes are accessed over one or more networks by client machines or terminals.

The critical applications and data are housed on disk devices which are physically cabled to two or more cluster nodes. These disk devices are either SCSI, IBM 9333 Serial, or IBM Serial Storage Architecture disks. This shared physical connection allows the ownership of shared logical volumes and their contents to be quickly switched from one node to another.

Each cluster node has various network interfaces, over which the Cluster Managers on neighboring nodes exchange periodic messages called keepalives or heartbeats. The main task of the Cluster Manager is to use these keepalive packets (KAs) to monitor nodes and networks in the cluster for possible failures.

A change in the status of the cluster (caused by a failure or a re-integration) is called a cluster event. When the Cluster Manager detects an event, the cluster goes into reconfiguration. The Cluster Manager runs one or more of a fixed set of customizable shell scripts. These scripts are able to take care of hardware failures as well as application restarts. Depending on how the cluster has been configured,

the scripts are run at the correct time to move protected resources to a standby machine in the cluster.

The HACMP software extends the benefits of high availability to clients by providing notification of changes in the cluster status to clients through the Cluster SMUX Peer and Cluster Information Services.

The HACMP software also provides a Cluster Lock Manager daemon called cllockd, for ensuring data integrity where cluster nodes are concurrently accessing data on shared storage.

The various components that make up an HACMP cluster are described in more detail in the next section.

3.2.2 HACMP Cluster Components

The cluster is the central object in the HACMP system, combining individual hardware components (nodes, disks, and networks) into a unified whole. The software components of the HACMP product make the cluster behave as a highly available environment for critical application and data. An example of an HACMP cluster is shown in Figure 8 on page 31. A detailed description of how to set up a cluster can be found in Chapter 5, “Setting Up HACMP for AIX” on page 81. In the next section, we will describe the major components of an HACMP cluster and see how they protect against single points of failure.

3.2.2.1 Cluster Hardware Components

An HACMP cluster consists of the following hardware objects:

- Nodes
- Shared disks
- Network adapters
- Networks
- Clients

Cluster Nodes: An HACMP cluster can have from two to eight nodes. A node is a RISC System/6000 system unit that runs the HACMP software. Each node in a cluster is identified by a unique name and owns a set of resources. These resources can be disks, volume groups, filesystems, networks, network addresses, and applications. In Section 3.2.3, “HACMP Cluster Resources” on page 42, you will find a more detailed discussion on cluster resources.

The cluster nodes can be any RISC System/6000 machine from the 2XX, 3XX, 5XX, 9XX, RXX series, and the family of SMP models using HACMP 4.1 for AIX. It could also be a model C10. You should keep in mind that, although all models from the 2XX and 3XX series are supported, none of them can provide a configuration with no single point of failure because of slot limitations. These models do not have enough slots for the adapter cards necessary to protect both network resources and disk resources. Therefore, if these models are used in a cluster, only part of the total system can be protected from failure. This situation could be sufficient for the requirements of some installations.

Each of the cluster nodes should have the following minimum configuration:

- 32 MB of memory

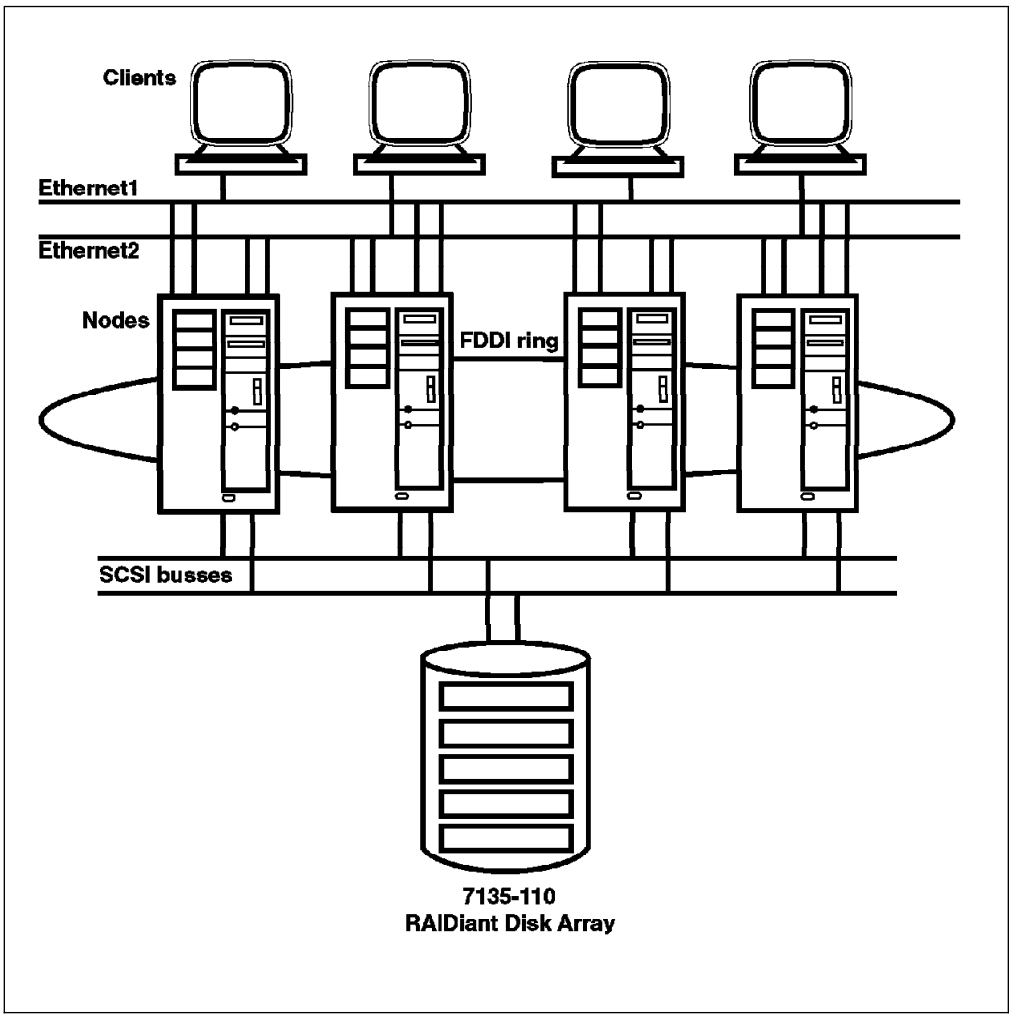


Figure 8. HACMP Cluster Example

- 320 MB internal disk drive with its own disk adapter
- Two LAN adapters for each attached LAN, supporting TCP/IP
- One external disk adapter, external disks and the necessary cables for multi-system attachment
- An available RS232 serial port

Shared Disks and Volume Groups: A shared disk is a disk that is physically connected to multiple nodes. A shared volume group is a volume group that consists entirely of shared disks and is defined to multiple systems to which the disks are physically attached. Logical volumes on a shared volume group are called shared logical volumes. Any filesystems that reside in these shared logical volumes are called shared filesystems.

In a cluster, these entities can be defined as cluster resources. In a non-concurrent environment, each disk resource is accessed by one node at a time, but is also known to other nodes which are capable of taking it over, such a failure or node shutdown occur.

In concurrent access environments, the shared disk resources are actively connected to more than one node at the same time. In this environment, disk

takeover time when a node fails is not a factor, since the disks are already actively connected before the failure.

The HACMP software supports shared external disk configurations that use SCSI-2 Differential disks and enclosures, including the IBM 7135-110 RAIDiant and IBM 7137/3514 RAID Arrays. Also supported is the IBM 9333 High-Performance Disk Drive Subsystem and the IBM 7133 Serial Storage Architecture (SSA) Subsystem.

The total number of SCSI devices (adapters and disks) on a shared bus is limited by the SCSI bus length restriction and by the number of available SCSI IDs (maximum of eight).

A chain of differential SCSI-2 disks can be connected to up to four cluster nodes. This kind of configuration requires both ends of the bus to be terminated externally. Differential SCSI-2 disks only support non-concurrent shared disk access with the exception of the RAID array products, which also support concurrent access.

Shared SCSI buses can be either SCSI-2 differential (8-bit bus) or Fast/Wide SCSI-2 differential (16-bit bus). Each of these options has its own set of adapters, disks, and cables. The total length of cabling, including internal cabling of disk enclosures, cannot exceed:

- 19 meters for differential SCSI-2.
- 25 meters for Fast/Wide differential SCSI-2.

For each shared SCSI bus, you will need the following:

- One IBM SCSI-2 Differential High-Performance External I/O Controller (feature code 2420) or one IBM SCSI-2 Differential Fast/Wide Adapter/A (feature code 2416), with the termination resistors removed, in each system.
- One SCSI-2 Differential Y-Cable to connect each SCSI adapter to the external bus

  The Y-cable will be either eight (feature code 2422) or sixteen (feature code 2426) bits wide, depending on whether the SCSI adapters and disks on the chain are SCSI-2 or Fast/Wide SCSI-2. Termination of the bus is provided at the Y-cables connected at the extremities of the bus.
- Shared disk devices

  These shared disk devices are either standalone external disk drives, or contained within a disk enclosure. The standalone disks must be of the type 7204-215 (8-bit) or 7204-315 (16-bit). The disk enclosures can be 7134-010s (16-bit) or 9334-011s (8-bit) or 9334-501s (8-bit). For AIX 3.2.5 systems, there can be up to eight SCSI ids supported on a single bus. A SCSI id is used for each adapter and each disk device on the bus. For a SCSI chain spanning four nodes, there can therefore be a maximum of four shared disk devices. These can be either standalone disks or contained within up to two 9334-XX1 enclosures or one 7134-010 enclosure.
- SCSI-2 Differential Cables (8 or 16-bit)

  These cables are used to connect the standalone disks, or 7134 or 9334 disk enclosures to the Y-cables. The cables come in a variety of lengths which can be chosen as desired, as long as the total length of the bus cabling stays within the limits mentioned above. The main thing to remember on this point is to be sure to order an extra cable for each disk or enclosure. These devices typically come with one attachment cable as

standard equipment, and additional cables as chargeable options. You will always need to order one extra cable, to be able to attach to the next device along the bus.

- SCSI-2 Differential Device-to-Device Cables

  These are used to connect standalone disk devices in a shared string or to attach two 9334s to each other.

- SCSI-2 Differential System-to-System Cables

  These are used to connect one node to another on a SCSI chain.

The IBM 7135-110 RAIDiant Disk Array consists of one or two disk array controllers with SCSI-2 Fast/Wide Differential host interfaces and five internal SCSI buses to which the physical disk drives are attached. It supports both non-concurrent and concurrent access. Configuring 7135s with two disk array controllers allows attachment to nodes on two separate SCSI buses, and provides a no single point of failure disk configuration. The adapters and cables required to connect a 7135 RAIDiant Disk Array on a shared SCSI bus are the same as those required to connect an IBM 7134 disk enclosure.

The IBM 7137/3514 Disk Array Models 2XX and higher are all supported by the HACMP software. These disk arrays support both non-concurrent and concurrent shared disk access. Concurrent access is supported only in non-mirrored configurations. The 7137/3514 Disk Array must be ordered with a special HACMP attachment kit. This kit consists of all cables, terminators, and microcode required to connect the disk arrays to up to four cluster nodes in a shared bus.

The IBM 9333 High-Performance Disk Drive Subsystem utilizes Serial-Link Disk Drives. Each subsystem requires a port on a High-Performance Disk Drive Subsystem Adapter in each of the cluster nodes that attach to it. The adapter has four ports, each supporting the attachment of a single subsystem. Models 9333-011 and 9333-501 support both non-concurrent and concurrent access from up to eight cluster nodes. Each 9333 subsystem can contain up to four disk drives, of varying capacities. The maximum total capacity in a single subsystem is 8GB.

The IBM 7133 SSA (Serial Storage Architecture) disk subsystem represents the second generation in serial storage implementations. Attached to an SSA Four Port Adapter can be loops of up to 48 disk devices each. Each adapter can support two loops. With a maximum single disk capacity of 4.5 GB, this provides 432 GB of disk capacity on a single disk adapter. Each 7133 SSA Subsystem can contain up to 16 disk drives, and can be combined with other 7133 SSA Subsystems in a single loop, or split into multiple loops. At the time of publishing, the 7133 SSA Subsystem was supported for sharing between two nodes only, on HACMP/6000 Version 3.1 only. This support is for both concurrent and non-concurrent modes of operation. Support for HACMP 4.1 for AIX and for more than two nodes is expected to be added later. You can find more information about the new SSA Subsystem in Section 4.3.5.2, “Serial Storage Architecture (SSA)” on page 73 and in Appendix B.4, “Serial Storage Architecture (SSA) Subsystems” on page 226.

A detailed description of how to attach each of these disk types is found in Appendix B, “Disk Setup in an HACMP Cluster” on page 209.

Network Interfaces: A network adapter (interface) connects a node to a network. A node typically is configured with at least two network interfaces for each network to which it connects: a service interface that handles all network traffic on the cluster, and one or more standby interfaces. Adapters in an HACMP cluster are identified by a label and a function.

Adapter Label: The adapter label, for TCP/IP networks, is the name in the /etc/hosts file associated with a specific IP address. Thus, a single node will have several adapter labels and IP addresses assigned to it. You should not confuse the adapter labels with the hostname of the machine.

Adapter Function: In an HACMP cluster, each adapter has a specific function that indicates the role it performs in the cluster. An adapter's function is either service, standby, or boot.

- Service adapter

  The service adapter is the primary connection between the node and the network. It is the interface over which the end users or client applications access the critical service that the node is offering. A node has one service adapter for each physical network to which it connects.

- Standby adapter

  A standby adapter backs up a service adapter on the same network. It can be configured to take over the IP address as well as the hardware address of that service adapter. The service adapter could be on the same node or on a different node in the cluster.

  The process of moving the IP address of a failed service adapter to the standby adapter on the same node is referred to as an *adapter swap*. The process of moving the IP address of a service adapter of a failed node to a standby adapter on a takeover node is referred to as *IP address takeover* (IPAT). The process of moving a hardware address between two network adapters is referred to as *hardware address swapping*.

  The standby adapter is configured on a different subnet from any service adapters on the same system, and its use should be reserved for HACMP only.

- Boot adapter

  IP address takeover is an HACMP facility that allows the standby adapter on one node to assume the network and/or hardware address of a failed node's service adapter. When the failed node reboots, its service adapter needs a second address to boot with in order to coexist in the same network with the takeover node. This is because its original IP address is already in use in the network by the takeover node. Hence, a boot adapter label and IP address are assigned to each service adapter for which IP address takeover is specified. The failed node boots with this address, and changes over to the service address only after the takeover node has released it during the reintegration process.

Networks: In an HACMP cluster, a network connects multiple nodes and is defined by its name and attribute.

Network Name: The network name is a symbolic value, identifying a network that participates in an HACMP cluster. Cluster processes use this name to determine which adapters are connected to the same physical network. The network name is arbitrary, but must be used consistently.

Network Attribute: A network's attribute is either public, private or serial.

- Public

  A public network connects multiple nodes in a cluster and also allows client system connections. Ethernet, Token-Ring, and FDDI are commonly used as public networks. Hardware address swapping is supported on all types of TCP/IP networks except FDDI.

- Private

  A private network provides a communication link for cluster nodes only. Possible candidates for a private network are ethernet and FDDI. This is an optional network used by cluster lock managers on cluster nodes to communicate with each other. You would use it only if your cluster was configured for concurrent shared disk access.

- Serial

  A serial network is a non-TCP/IP connection between a pair of cluster nodes for Cluster Manager control messages and heartbeat traffic to continue in the event the TCP/IP subsystem fails. A serial network can be a SCSI-2 Differential bus using Target Mode SCSI or a raw RS232 connection.

Every node in an HACMP cluster should be connected to at least one public and one serial network. The connection to a public network is to provide critical services to end users and the connection to a serial network is to prevent *node isolation*.

Node isolation occurs when all the cluster nodes are running, but cannot communicate with each other, due to the failure of the TCP/IP subsystem or physical failures on the TCP/IP networks. This leads the Cluster Managers on each of the isolated nodes to conclude that the other nodes are down and to initiate a cluster reconfiguration. Nodes configured to take over certain resources will fail to do so, since these resources are still being used by the node that owns them. The result of node isolation is unpredictable. Therefore, it is essential that each node have at least one connection to a non-TCP/IP network.

Clients: In the HACMP environment, a client is a system that can access the cluster nodes for critical services. Although, in this environment, high availability is not directly extended to clients, the cluster information daemon, *clinfo*, can be used on client processors to monitor the status of the cluster.

3.2.2.2 Cluster Software Components

There are several software components that participate in making a cluster highly available. These include the following:

- AIX operating system
- TCP/IP subsystem
- Logical Volume Manager (LVM) subsystem

- HACMP software components

AIX Operating System: The AIX operating system provides the underlying support for an HACMP cluster. Several of its features are utilized by the HACMP software to provide a high availability environment.

These include:

- Journaled File System (JFS), which uses database journaling techniques to protect the integrity of filesystem metadata.
- Object Data Manager (ODM), which stores objects describing HACMP entities, such as nodes and resources, in a global database that can be accessed by any node in a cluster.
- A Dynamic kernel, which enables adapter swapping, IP address takeover, and disk takeover.
- Device drivers, such as the SCSI target mode device driver. SCSI target mode can be utilized as a heartbeat network for HACMP.

TCP/IP Subsystem: HACMP uses TCP/IP facilities to maintain communication among cluster members. A TCP/IP facility crucial to HACMP's ability to provide high availability is IP address takeover, a networking capability that allows one node to assume the network address of another node that has left the cluster.

AIX Logical Volume Manager: The AIX LVM facilities that HACMP relies on to provide high availability are:

- Volume groups

  A volume group can include one or more physical disks, each of which contains a complete table of information about the volume group to which it belongs. If a node detaches from the cluster, the surviving nodes can provide exactly the same view of disk services.

- Disk mirroring

  This minimizes the effect of a disk failure by duplicating the contents of the disk. If a disk fails, the LVM enables the node to access a mirrored disk and continue to work.

  It is important to understand that, while HACMP relies on the disk mirroring provided by the LVM to compliment its own availability functions, the mirroring function is actually transparent to HACMP. There is no direct notification to or through HACMP of a disk failure, although the product can be customized to do so, using the Error Notification feature. This feature is described in Section 6.8, “AIX Error Notification” on page 172.

HACMP Software: The custom HACMP software that provides a highly available environment to cluster nodes consists of three major subsystems:

- Cluster Manager

  The Cluster Manager runs on each cluster node and is responsible for monitoring local hardware and software subsystems, tracking the state of the cluster peers, and acting appropriately to maintain the availability of cluster resources when there is a change in the status of the cluster. The Cluster Managers on neighboring nodes exchange periodic messages, called keepalives or heartbeats, to do this monitoring. The Cluster

Manager responds to changes in cluster status (events) by executing a set of scripts corresponding to that particular event. The cluster manager can be started explicitly through SMIT, or automatically upon startup of the system.

- Cluster SMUX Peer and Cluster Information Services

  An HACMP cluster is dynamic and can undergo various changes in its state over time. An example of this would be a node joining or leaving the cluster, or a standby adapter taking over from a service adapter. If the clients are not aware of the changes to the cluster, all the changes may not be completely transparent to the end user. If the clients are aware of the changes in the state of the cluster, they can react to these changes and possibly mask them from the end user.

  The HACMP software provides notification of cluster state changes to clients through the Cluster SMUX Peer and Cluster Information Services.

  The Cluster SMUX Peer Service provides Simple Network Management Protocol (SNMP) support to client applications. For information on SNMP, please refer to the publication *AIX Version 3.2 for the IBM RISC System/6000 Communications Concepts and Procedures*. The Cluster SMUX Peer daemon, clsmuxpd, maintains cluster status information in a special HACMP/6000 Management Information Base (MIB).

  Cluster Information daemon, clinfo, is an SNMP network monitor which can be started with the Cluster Manager from SMIT or at system startup. It runs on cluster nodes or client machines and queries the clsmuxpd daemon at regular intervals for updated cluster information. Applications running on these client machines can use the clinfo Application Programming Interfaces (APIs) documented with HACMP to track the status of the cluster. The *Programming Client Applications* manual describes how you can use these APIs.

- Cluster Lock Manager

  The Concurrent Resource Manager subsystem of HACMP implements advisory locking, to ensure the integrity of data that is being concurrently accessed by applications running on multiple nodes in a cluster. This means that all applications accessing data concurrently should first obtain a lock from the lock manager before manipulating the data. Applications can do this by using the APIs provided by the Cluster Lock Manager. The *Programming Locking Applications* manual describes how you can use these APIs. The Cluster Lock Manager runs as the cllockd daemon which can be started with the Cluster Manager from SMIT or at system startup.

A detailed description of how the HACMP daemons function is given in Appendix A, “HACMP Software Components” on page 199.

3.2.2.3 Eliminating Single Points of Failure Using HACMP

The HACMP software allows a cluster to continue to provide application services even though a key system component is no longer available. When a component becomes unavailable, HACMP detects the loss and shifts that component's workload to another component in the cluster. The failure recovery is done automatically by the event scripts, provided with HACMP, without any human intervention. The purpose of the product design is to avoid any component being a single point of failure.

The cluster components that are potential single points of failure include:

- Processors or nodes
- Networks and network adapters
- Disks and disk adapters
- Applications

The following sections describe how HACMP reacts to the failure of each of these components, to enable the cluster to continue to provide application service.

Node Failure: A node leaves a cluster either due to a planned shutdown or because of failure. A node failure is detected when the Cluster Manager on a neighboring node does not receive keepalives from it for a defined period of time. If all the operational Cluster Managers in the cluster agree that the failure is a node failure, the failed node is removed from the cluster and its resources are taken over by the nodes configured to do so.

In HACMP clusters, disks containing critical data are physically connected to multiple nodes on a common bus. In non-concurrent access environments, only one node has an active logical connection at any given time. The node with the active connection has the shared volume group(s) activated and all filesystems mounted. All the other nodes have the shared volume group(s) defined but varied off. When a node that currently owns a disk fails, a surviving node assumes control of the disk, activates all volume groups, and mounts all filesystems so that the critical data remains available. Figure 9 on page 39 illustrates disk takeover.

In a concurrent access configuration, disk takeover is not necessary because the shared disks are, at all times, actively connected to all nodes on the shared bus.

In most production environments, a service that is being provided to clients or end users is accessed over a network and tied to a specific IP address (assuming a TCP/IP network). If a node providing a service fails, disk takeover would not be enough to resume critical services, because the IP address over which the users access the data or applications on these disks would not be available.

IP address takeover occurs when a node assumes the IP address of a node that has failed. Assuming that it has also taken over the disks of the failed node, the takeover node can then provide the critical services that the failed node was providing. Figure 10 on page 40 illustrates IP address takeover.

Every network adapter connected on a TCP/IP network is uniquely identified by its hardware address. Every node connected to a TCP/IP network uses a protocol called Address Resolution Protocol (ARP) to map the IP address of a network adapter to its hardware address. The ARP cache contains this mapping information. After IP address takeover has taken place in a cluster, the ARP caches of all nodes, clients, and network devices (routers, bridges, hubs) connected to the cluster have to be refreshed or flushed to reflect the new hardware address that maps to the IP address that was taken over. All the clients and network devices connected to the cluster may not be able to flush their ARP cache. In this case, you can configure HACMP to take over the hardware address together with the IP address of the failed node.

If a critical application or process was running on a node that fails, a takeover node can restart that application or process so that the service is not lost.

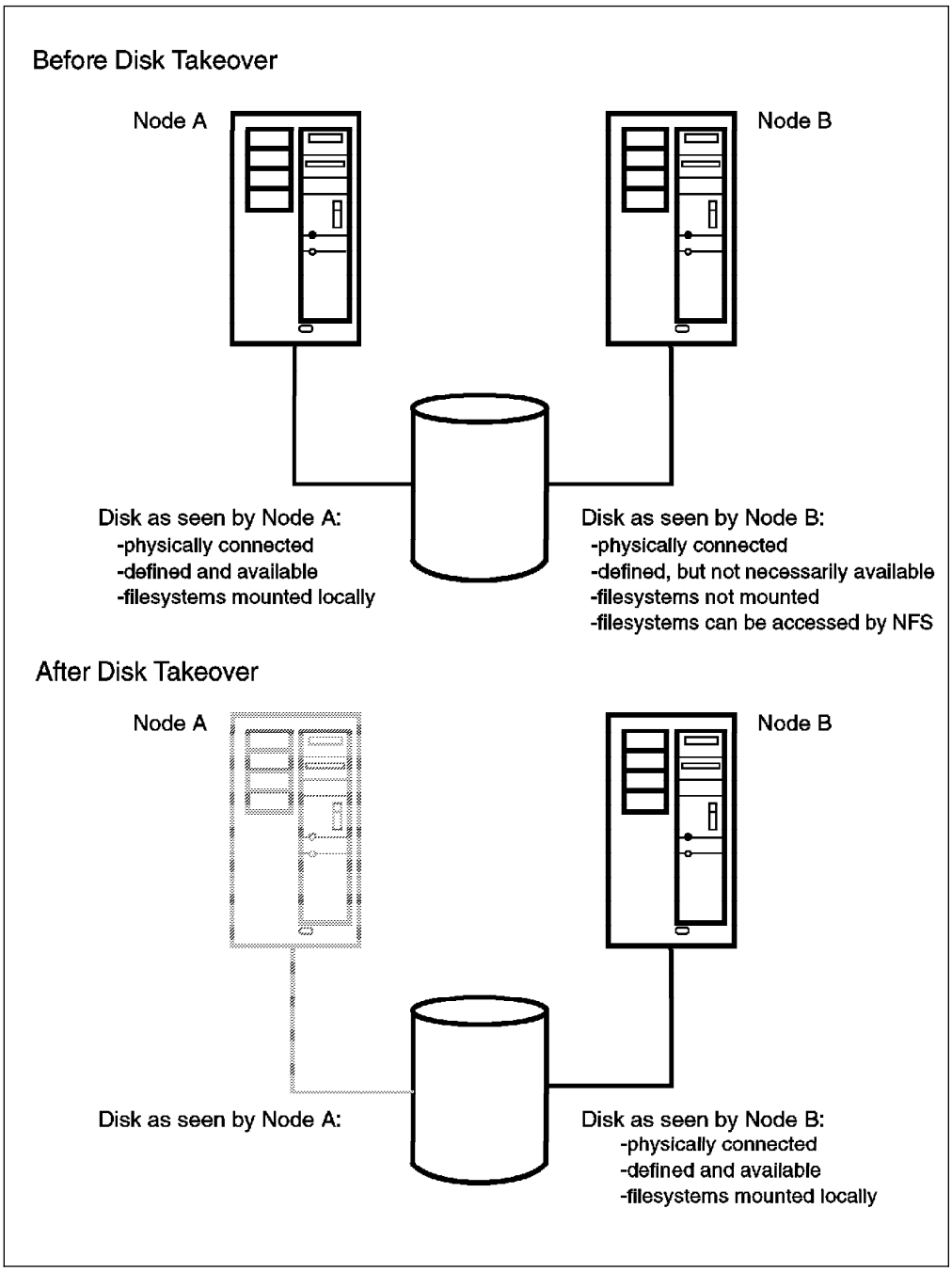


Figure 9. Disk Takeover

Network and Network Adapter Failure: A network is determined to have failed when no node in the cluster is able to communicate across it. It is recommended that you design your cluster with more than one network, whatever the attribute (public, private, or serial), so that HACMP has at least one network at all times that it can use to monitor the status of cluster nodes. If there is more than one network to which all cluster nodes are connected, the Cluster Manager detects a network failure, but takes no action. If the failing network is a public network over which critical services are accessed, it is imperative that connectivity is restored. One of the possible ways you can eliminate this single point of failure is by having the nodes in the cluster connected over multiple public networks and by customizing

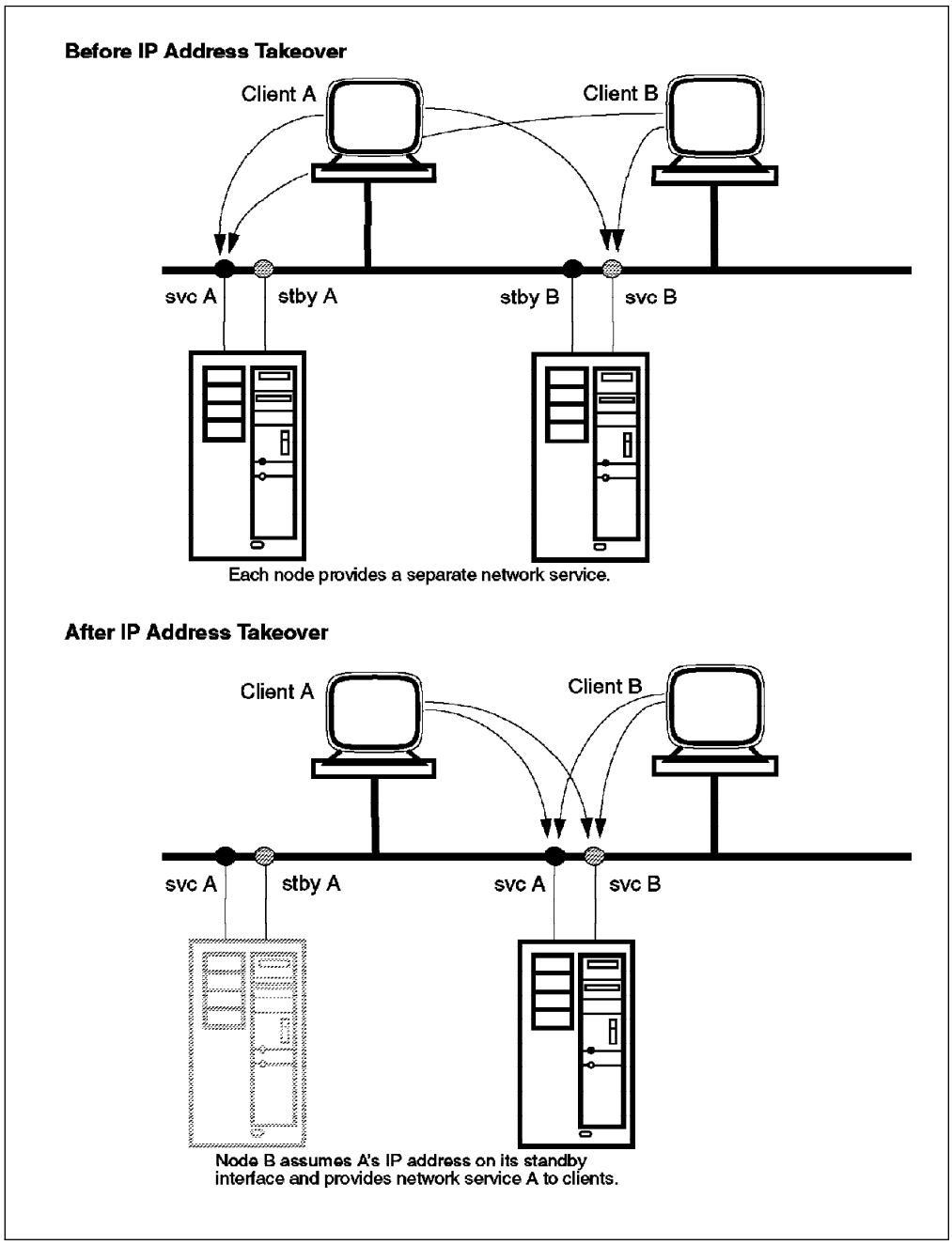


Figure 10. IP Address Takeover

the scripts that handle network failures to reroute all traffic through an alternate network.

When the service adapter on a node fails, the Cluster Manager swaps the roles of the service and standby adapters on that node. The end user just sees his screen freeze for a few seconds (the exact time depends on the type of network) before he can continue. The failure of a standby adapter is detected and logged by the Cluster Manager, but no default action is taken. If you wish the Cluster Manager to take further action when a standby adapter fails, you can customize the reaction to this event. Figure 11 on page 41 illustrates adapter swapping

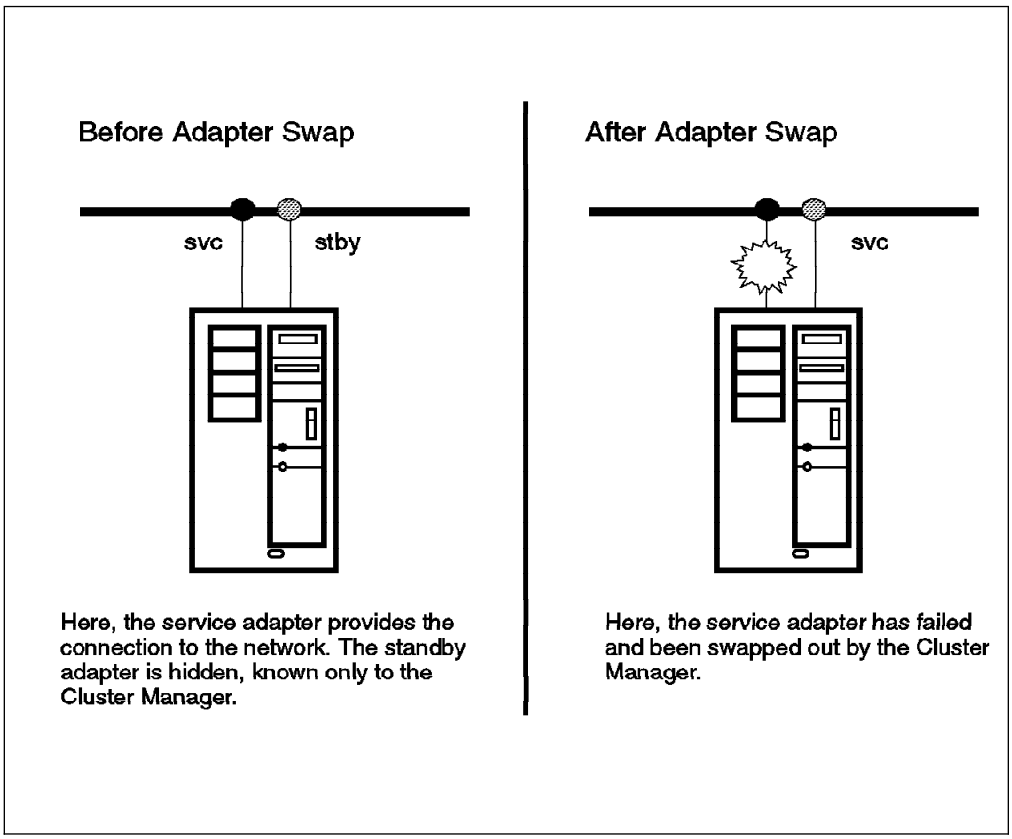


Figure 11. Adapter Swapping

Disk and Disk Adapter Failure: HACMP does not directly handle disk and disk adapter failures.

If you are using SCSI disk drives or 9333 serial disk subsystems, you can configure your cluster nodes with multiple SCSI or serial chains and have your disks mirrored across these chains. If you are using SSA disks, you can configure your cluster with multiple loops, and mirror your disks across loops. In this way, your critical data is protected against the failure of any single component of storage hardware (adapter, cabling, disks).

If you are using the 7135-110 RAIDiant Disk Array or the 7137/3514 Disk Array, data protection is already built in. You need to design the connection to these arrays to protect against failures of cables and adapters.

Please refer to Appendix B, “Disk Setup in an HACMP Cluster” on page 209 for further details of cabling different storage products in a no single point of failure configuration.

Application Failure: HACMP cannot detect the failure of an application by itself. If a node goes down, any application that was running on it can be configured to be restarted on a takeover node. The Cluster Manager can be configured to start any critical application when a node joins a cluster for the first time or after a failure.

Further, the AIX System Resource Controller (SRC) can be used to monitor for the presence or absence of an application's daemon and respond accordingly.

In conclusion, HACMP can detect most types of failure and recover from them in order to keep critical services available to the end user. The time taken to recover varies from thirty to three hundred seconds or more, depending on the amount of disk storage to be taken over in the case of a node failure and the amount of customization that you add to the recovery scripts.

3.2.3 HACMP Cluster Resources

HACMP provides a highly available environment by identifying a set of cluster-wide resources essential to uninterrupted processing and then defining relationships among nodes that ensure these resources are available to client processes.

When a cluster node fails or detaches from the cluster for a scheduled outage, the Cluster Manager redistributes its resources among any number of the surviving nodes.

These resources can include the following:

- Disks
- Volume Groups
- Filesystems
- Filesystems to be NFS mounted
- Filesystems to be NFS exported
- Service IP addresses
- Applications

3.2.3.1 Resource Groups

Each resource in a cluster is defined as part of a resource group. This allows you to combine related resources that need to be together to provide a particular service. A resource group also includes the list of nodes that can acquire those resources and serve them to clients.

A resource group is defined as one of three types:

- Cascading
- Rotating
- Concurrent

Each of these types describes a different set of relationships between nodes in the cluster, and a different set of behaviors upon nodes entering and leaving the cluster.

Cascading Resource Groups: All nodes in a cascading resource group are assigned priorities for that resource group. These nodes are said to be part of that group's resource chain. In a cascading resource group, the set of resources cascades up or down to the highest priority node active in the cluster. When a node who is serving the resources fails, the surviving node with the highest priority takes over the resources. A picture of a cascading resource group is in Figure 12 on page 43.

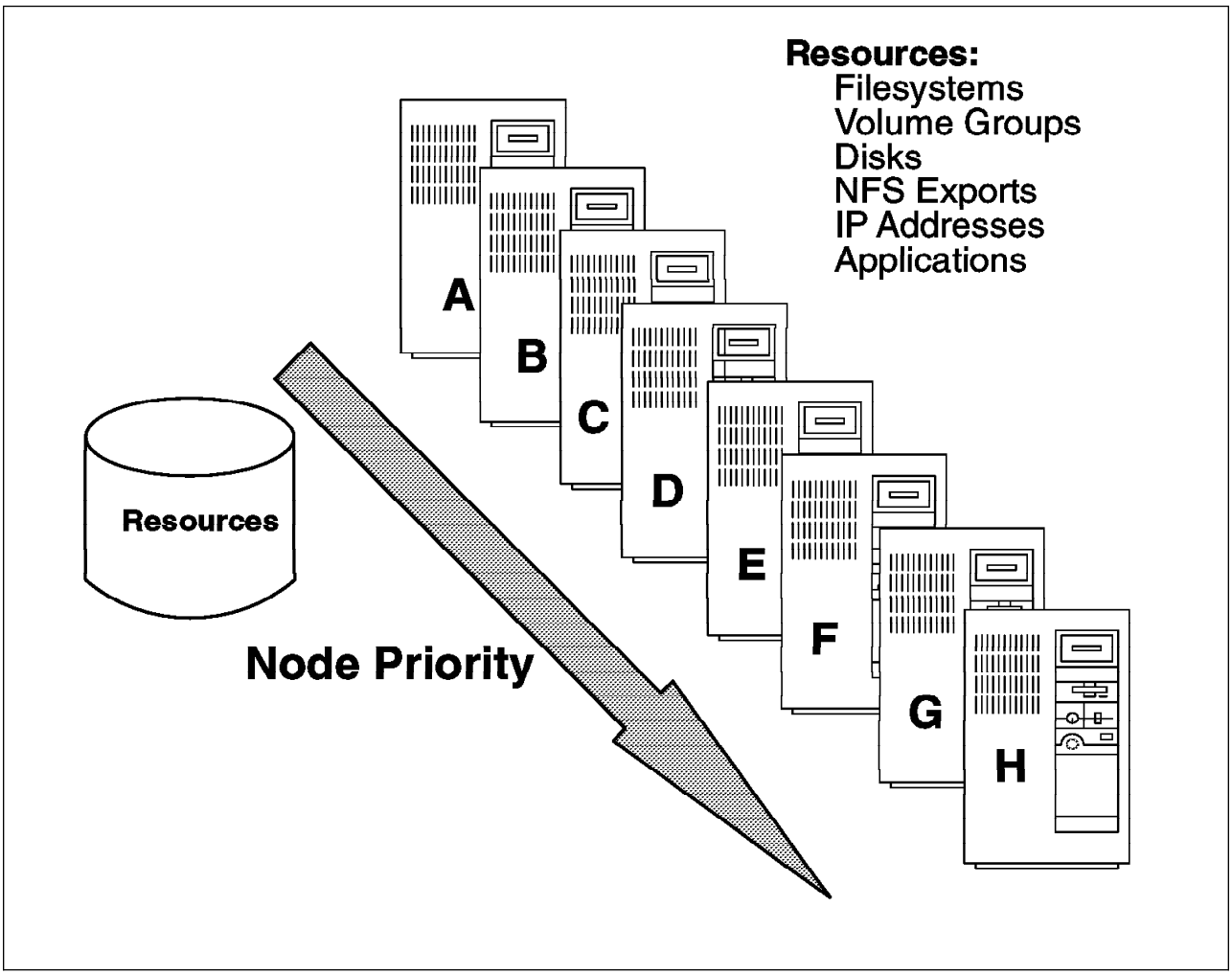


Figure 12. Cascading Resource Group

A parameter called *Inactive Takeover* decides which node takes the cascading resources when the nodes join the cluster for the first time. If this parameter is set to *true*, the first node in a group's resource chain to join the cluster acquires all the resources in the resource group. As successive nodes join, the resources cascade up to any node with a higher priority that joins the cluster. If this parameter is set to *false*, the first node in a group's resource chain to join the cluster acquires all the resources in the resource group only if it is the node with the highest priority for that group. If the first node to join does not acquire the resource group, the second node in the group's resource chain to join, if it has a higher priority than the node already active, acquires the resource group. As successive nodes join, the resource group cascades to the active node with the highest priority for the group. The default is *false*.

Member nodes of a cascading resource chain always release a resource group to a reintegrating node with a higher priority.

Rotating Resource Groups: A rotating resource group is associated with a group of nodes, rather than a particular node. A node can be in possession of a maximum of one rotating resource group per network.

As participating nodes join the cluster for the first time, they acquire the first available rotating resource group per network until all the groups are acquired. The remaining nodes remain on standby.

When a node holding a rotating resource group leaves the cluster, either because of a failure or gracefully while specifying the takeover option, the node with the highest priority and available connectivity takes over. On reintegration, a node remains as a standby and does not take back any of the resources that it had initially served. A picture of a rotating resource group is in Figure 13.

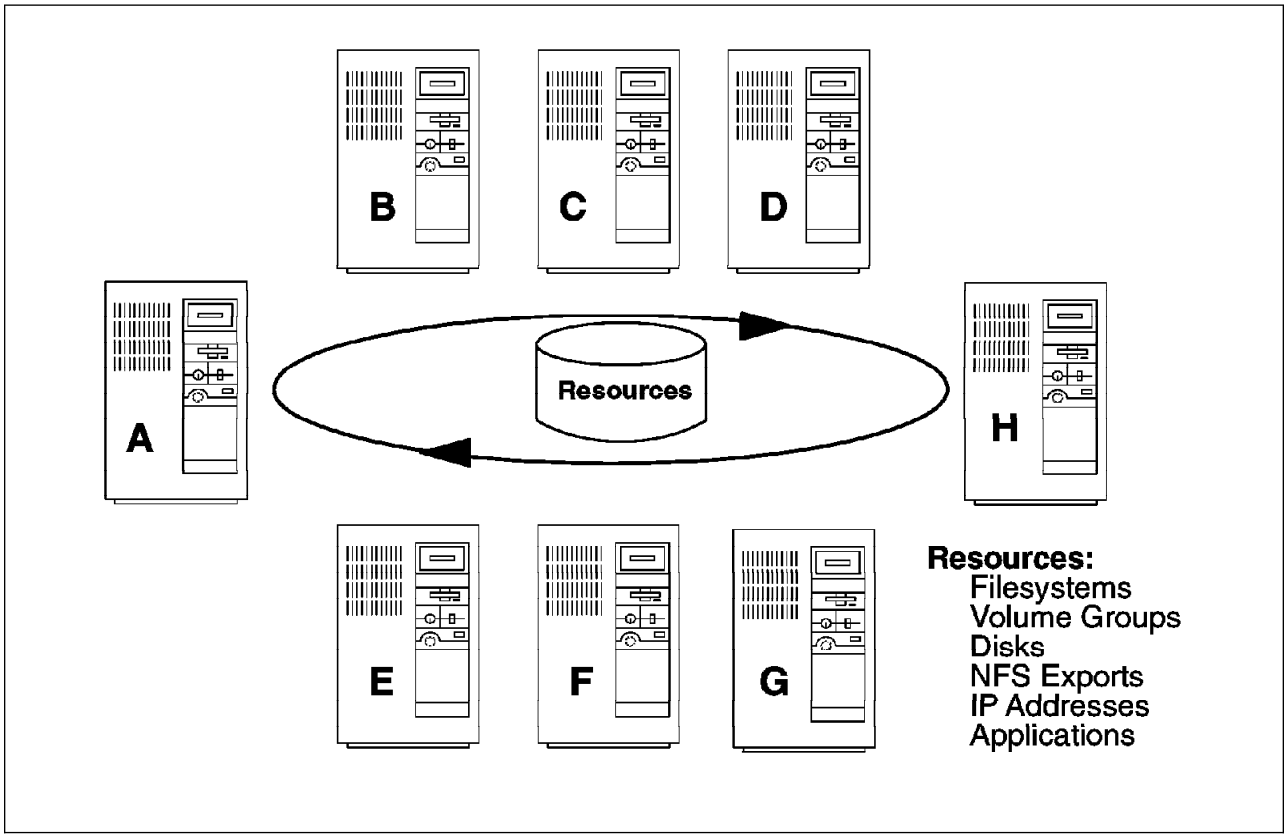


Figure 13. Rotating Resource Group

Concurrent Resource Groups: A concurrent resource group may be shared simultaneously by multiple nodes. The resources that can be part of a concurrent resource group are limited to volume groups with raw logical volumes, and raw disks.

When a node fails, there is no takeover involved for concurrent resources. On reintegration, a node once again accesses the resources simultaneously with the other nodes. A picture of a concurrent resource group is in Figure 14 on page 45.

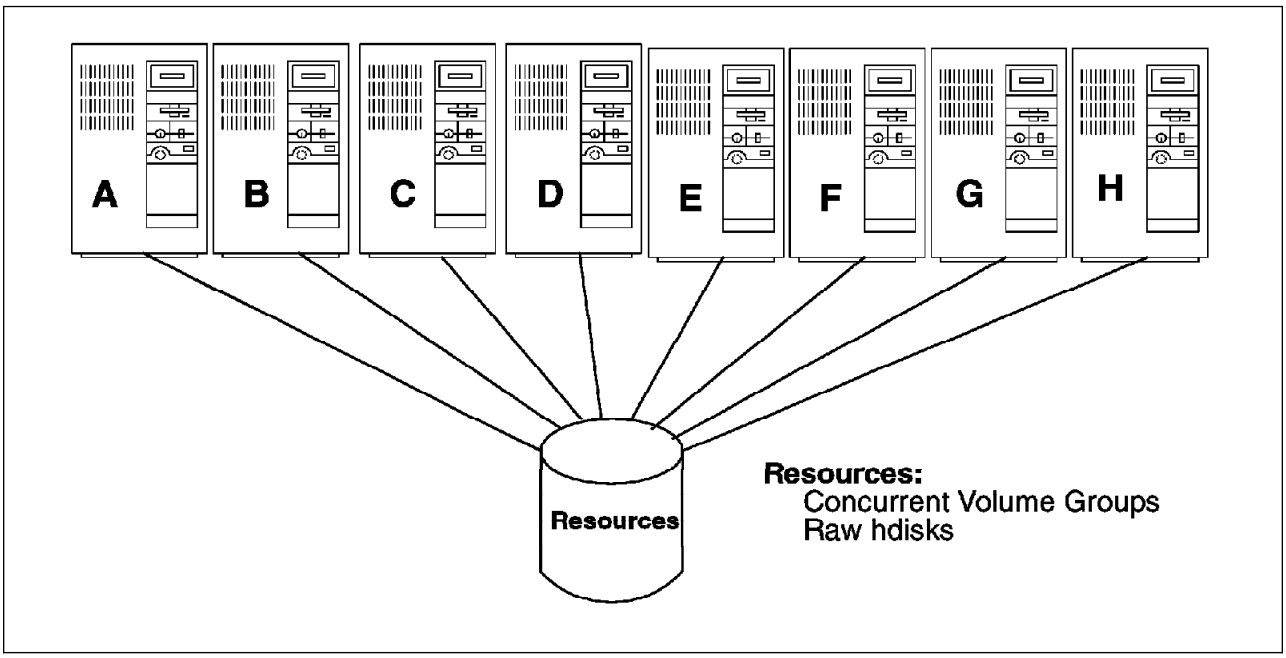


Figure 14. Concurrent Resource Group

The Cluster Manager makes the following assumptions about the acquisition of resource groups:

Cascading The active node with the highest priority controls the resource group.

Concurrent All active nodes have access to the resource group.

Rotating The node with the rotating resource group's associated service IP address controls the resource group.

3.2.4 HACMP Cluster Configurations

There are several ways in which you can configure your cluster to provide high availability. In this section, we will discuss a few basic failover configurations that you can use to design the cluster best suited to your availability requirements.

In most of the examples that follow, we will be using two node or three node clusters, but the same principles can be extended to larger clusters. When you design larger clusters, you should keep SCSI bus limitations and concurrent access limitations in mind. Please refer to Appendix B, “Disk Setup in an HACMP Cluster” on page 209 for further details of limitations related to storage.

The HACMP software has no knowledge of these failover strategies. The configurations are implemented using resource groups and their relationships with the cluster nodes. All the examples of resource groups in this section have only disks as constituent resources. This has been done for convenience only. The same concepts are easily extended to other resources such as IP addresses and applications.

The first distinction that you need to make while designing a cluster is whether you need a non-concurrent or a concurrent disk access environment.

3.2.4.1 Non-Concurrent Disk Access Configurations

The possible non-concurrent disk access configurations are:

- Hot Standby
- Rotating Standby
- Mutual Takeover
- Third-Party Takeover

Hot Standby Configuration: Figure 15 illustrates a two node cluster in a hot standby configuration.

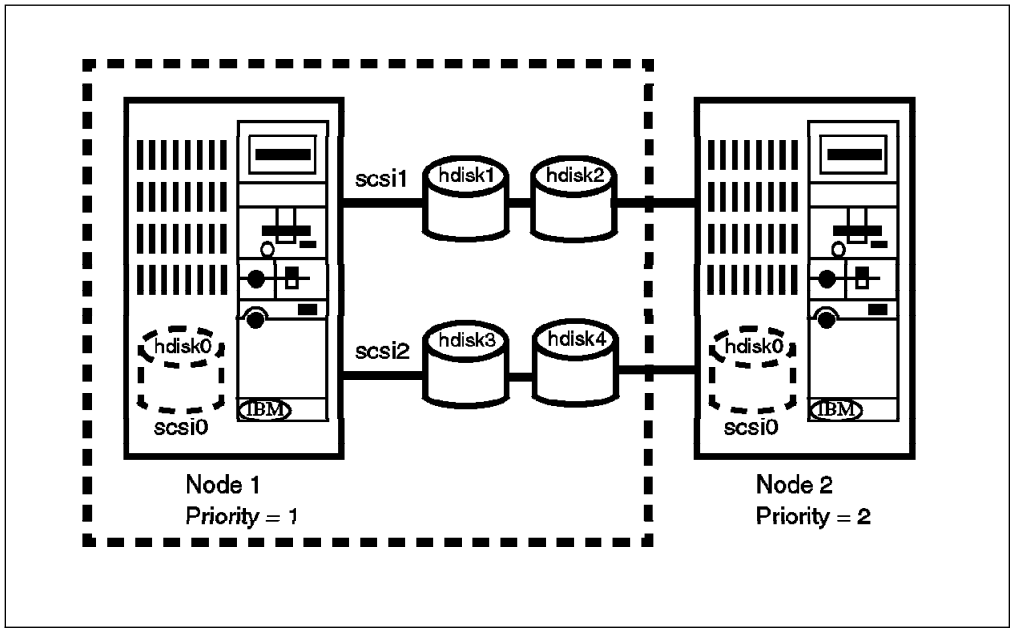


Figure 15. Hot Standby Configuration

In this configuration, there is one cascading resource group consisting of the four disks, hdisk1 to hdisk4, and their constituent volume groups and filesystems. Node 1 has a priority of 1 for this resource group while node 2 has a priority of 2. During normal operations, node 1 provides all critical services to end users. Node 2 may be idle or may be providing non-critical services, and hence is referred to as a hot standby node. When node 1 fails or has to leave the cluster for a scheduled outage, node 2 acquires the resource group and starts providing the critical services.

The advantage of this type of a configuration is that you can shift from a single system environment to an HACMP cluster at a low cost by adding a less powerful processor. Of course, this assumes that you are willing to accept a lower level of performance in a failover situation. This is a trade-off that you will have to make between availability, performance, and cost.

Rotating Standby Configuration: This configuration is the same as the previous configuration except that the resource groups used are rotating resource groups.

In the hot standby configuration, when node 1 reintegrates into the cluster, it takes back the resource group since it has the highest priority for it. This implies a break in service to the end users during reintegration.

If the cluster is using rotating resource groups, reintegrating nodes do not reacquire any of the resource groups. A failed node that recovers and rejoins the cluster becomes a standby node. You would choose a rotating standby configuration if you do not want a break in service during reintegration.

Since takeover nodes continue providing services till they have to leave the cluster, you should configure your cluster with nodes of equal power. While at a higher price in CPU hardware, a rotating standby configuration gives you better availability and performance than a hot standby configuration.

Mutual Takeover Configuration: Figure 16 illustrates a two node cluster in a mutual takeover configuration.

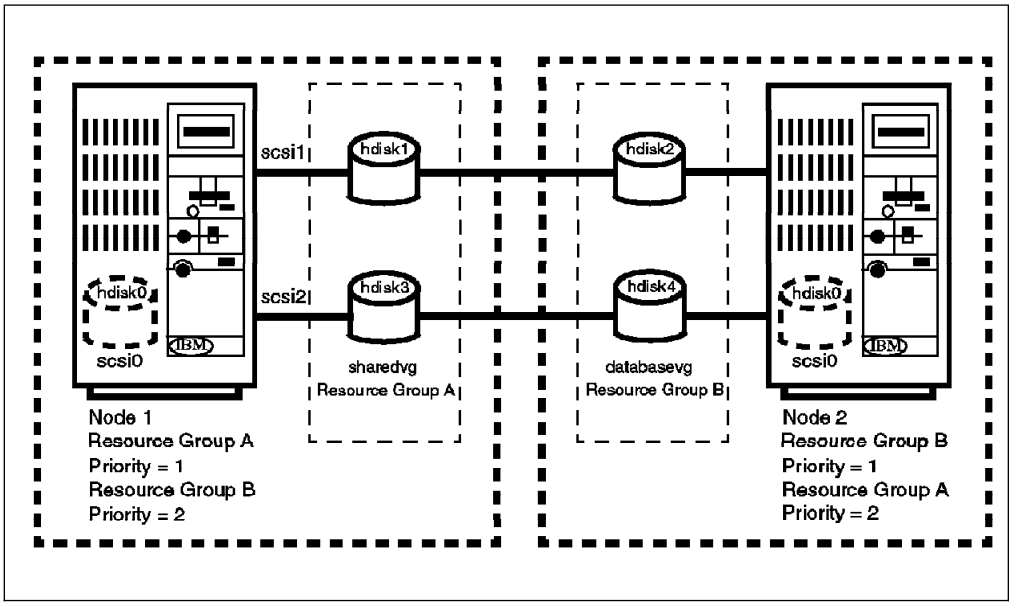


Figure 16. Mutual Takeover Configuration

In this configuration, there are two cascading resource groups A and B. Resource group A consists of two disks, hdisk1 and hdisk3, and one volume group, sharedvg. Resource group B consists of two disks, hdisk2 and hdisk4, and one volume group, databasevg. Node 1 has priorities of 1 and 2 for resource groups A and B respectively, while Node 2 has priorities of 1 and 2 for resource groups B and A respectively.

During normal operations, nodes 1 and 2 have control of resource groups A and B respectively, and both provide critical services to end users. If either node 1 or node 2 fails or has to leave the cluster for a scheduled outage, the surviving node acquires the failed node's resource groups and continues to provide the failed node's critical services.

When a failed node reintegrates into the cluster, it takes back the resource group for which it has the highest priority. Therefore, even in this configuration, there is a break in service during reintegration. Of course, if you look at it from the point of view of performance, this is the best thing to do, since you have one node doing the work of two only when any one of the nodes is down.

Third-Party Takeover Configuration: Figure 17 on page 48 illustrates a three

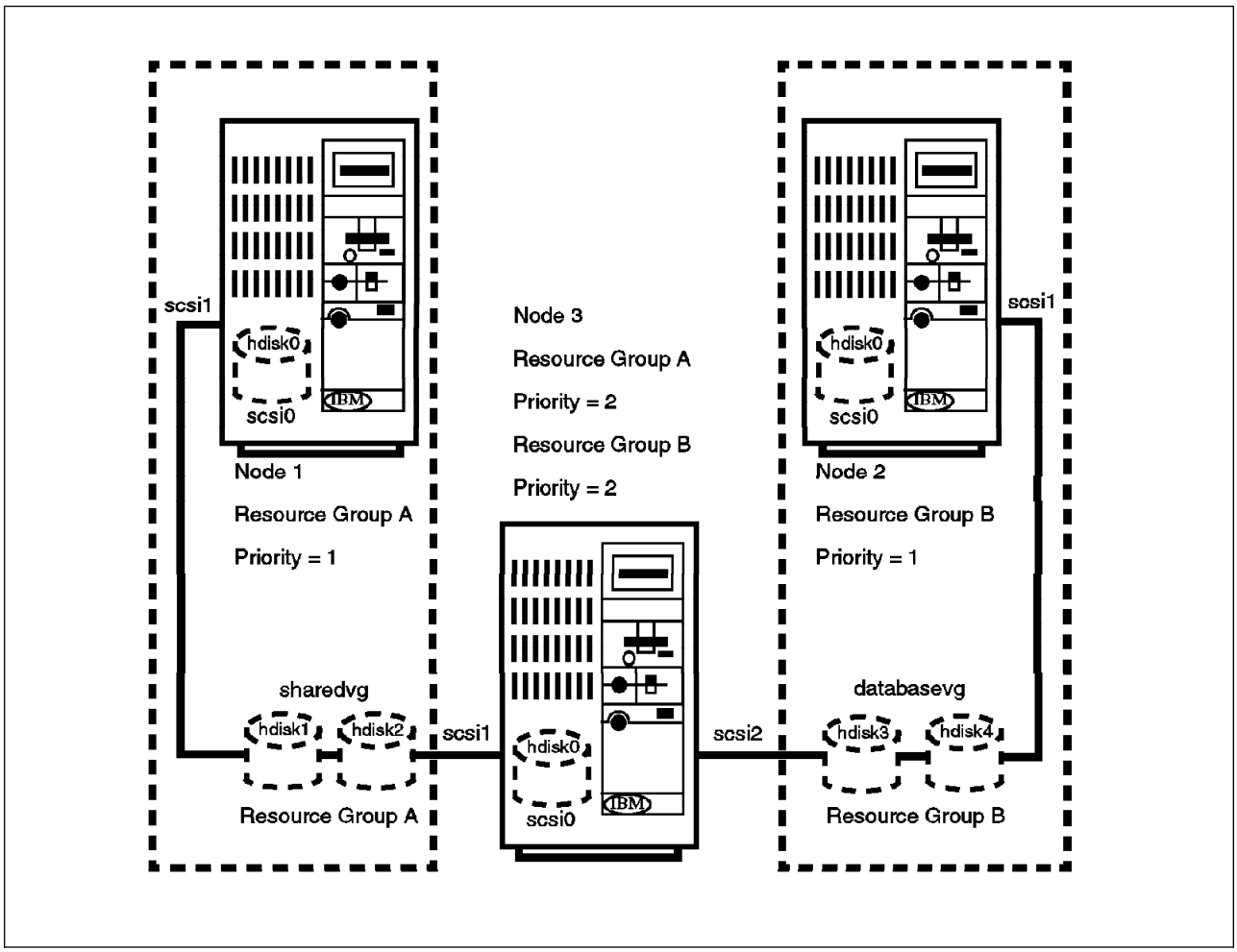


Figure 17. Third-Party Takeover Configuration

node cluster in a third-party takeover configuration. This configuration takes care of the performance degradation that results from a failover in the mutual takeover configuration.

Here the resource groups are the same as the ones in the mutual takeover configuration. Also, similar to the previous configuration, nodes 1 and 2 each have priorities of 1 for one of the resource groups, A or B. The only thing different in this configuration is that there is a third node which has a priority of 2 for both the resource groups.

During normal operations, node 3 is either idle or is providing non-critical services. In the case of either node 1 or node 2 failing, node 3 takes over the failed node's resource groups and starts providing its services. When a failed node rejoins the cluster, it reacquires the resource group for which it has the highest priority.

So, in this configuration, you are protected against the failure of two nodes and there is no performance degradation after the failure of one node.

3.2.4.2 Concurrent Disk Access Configurations

A concurrent disk access configuration usually has all its disk storage defined as part of one concurrent resource group. The nodes associated with a concurrent resource group have no priorities assigned to them.

If 7135 RAIDiant Array Subsystems or 7137/3514 Disk Array Subsystems are used for storage, you can have a maximum of four nodes concurrently accessing a set of storage resources. If you are using the 9333-501 High-Performance Disk Drive Subsystem, you can have up to eight nodes concurrently accessing it.

In the case of a node failing, a concurrent resource group is not explicitly taken over by any other node, since it is already active on the other nodes. However, in order to somewhat mask a node failure from the end users, you should also have cascading resource groups, each containing the service IP address for each node in the cluster. When a node fails, its service IP address will get taken over by another node and users can continue to access critical services at the same IP address that they were using before the node failed.

3.2.5 HACMP Cluster Events

Cluster events correspond to changes in the status of nodes, networks, and network adapters, and to reaching time limits during the reconfiguration process.

Each cluster event has an associated shell script which is executed by the Cluster Manager in response to that event. Each event script may start a series of subevent scripts, depending on the cluster configuration and the state of the cluster.

The Cluster Manager in an HACMP cluster detects and reacts to the following events:

- A node trying to join the cluster.
- A node leaving the cluster.
- A network going down.
- A network coming up.
- IP address being swapped between a service and a standby adapter.
- The cluster being unstable or in reconfiguration for too long.

3.2.5.1 Node Events

A node event could be a node joining the cluster or a node leaving the cluster. A node joins the cluster when the Cluster Manager is started on that node, and leaves the cluster when the Cluster Manager is stopped. A node can leave the cluster due to a failure or a planned outage. If a node has to be brought down for any kind of maintenance, the Cluster Manager can be shut down in one of three ways:

- Graceful Shutdown

  The node gives up all the resources that it had acquired when it joined the cluster. These resources do not get taken over by any of the other nodes in the cluster even if they were configured to do so.

- Graceful Shutdown With Takeover

The node gives up all the resources that it had acquired when it joined the cluster. Depending on the nature of the resource groups defined for the cluster and their relationships with the cluster nodes, the resources are taken over by other active nodes in the cluster.

- Forced Shutdown

  The node does not give up any of its resources and consequently these resources do not get taken over by any of the other nodes in the cluster. A forced shutdown just stops the Cluster Manager and any other HACMP daemon that was running on the node.

The flowchart in Figure 18 on page 51 shows the series of event scripts that are executed when the first node joins the cluster. The flowchart in Figure 19 on page 52 shows the sequence of execution of event scripts when a node joins a cluster that already has one or more active nodes.

The sequence in which event scripts get executed on active nodes after a cluster node fails is shown in Figure 20 on page 53. When a node leaves a cluster for a scheduled outage, event scripts are executed only if the HACMP services are shut down gracefully with takeover. The event scripts that get executed on the node that is leaving and the active nodes is shown in Figure 21 on page 54.

In the figures, please note the only node event scripts that get executed directly by the Cluster Manager are:

node_up
node_up_complete
node_down
node_down_complete

These scripts in turn execute the other scripts.

3.2.5.2 Network Events

A network event could be a network becoming available for use by cluster nodes or a network failing. When the Cluster Manager determines that a network has failed, a *network_down* event occurs. When a network becomes available for use in a cluster, the *network_up* event occurs.

A network failure can be further classified into two cases:

- Local

  When only one node has lost contact with a network, the failure is said to be local.

- Global

  When all the nodes in the cluster have lost contact with a network, the failure is said to be global.

The network_down event sends mail to the system administrator, informing him of the failure, but does not do anything else. The network_up event script takes no default action.

The *network_down_complete* and the *network_up_complete* scripts get executed only after a network_down or a network_up event gets completed successfully.

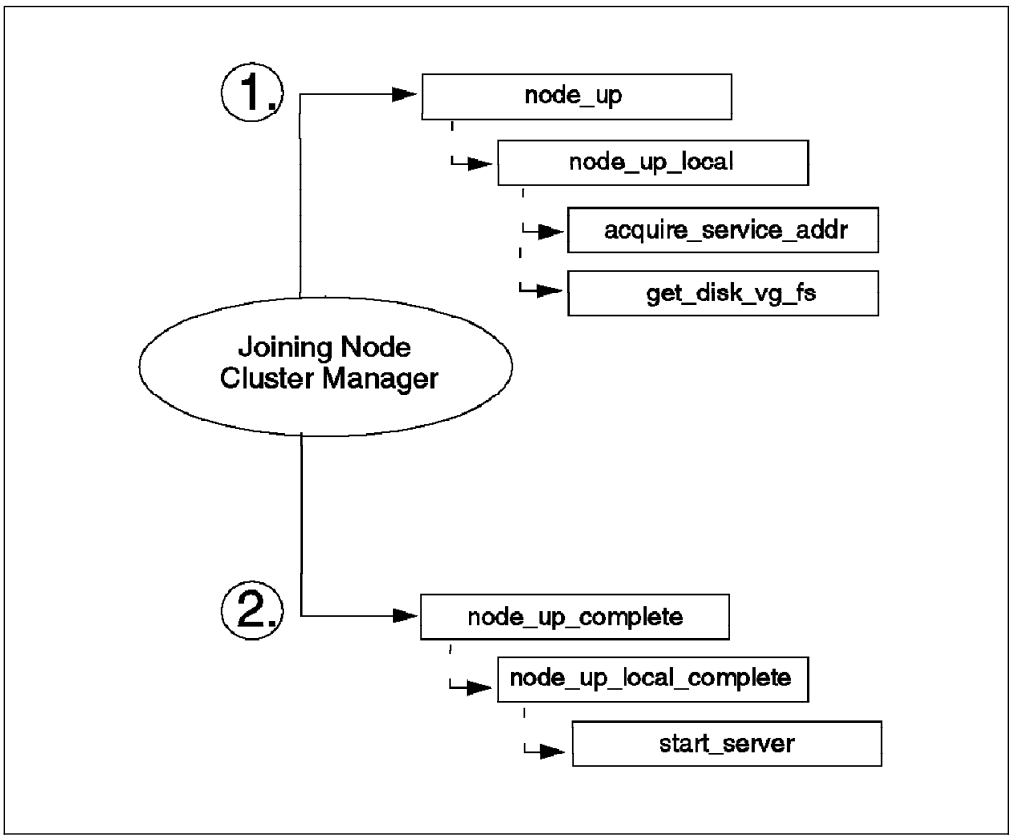


Figure 18. First Node Joins Cluster

These scripts do not do anything by default, since the appropriate action would depend on the local network configuration. You can customize them accordingly.

Network Adapter Events: The network adapter events that the Cluster Manager reacts to are:

swap_adapter

This occurs when the service adapter on a node fails and there is a standby adapter available on that node on the same network. The script for this event, by default, exchanges the IP addresses of the service and the standby adapter, and reconstructs the routing table.

swap_adapter_complete

This event occurs only after the swap_adapter event has successfully completed. The script ensures that the local ARP cache is updated by deleting entries and pinging cluster IP addresses.

fail_standby

This event occurs when the standby adapter is no longer available, either because it has taken over a service IP address or because of its own failure. The script for this event sends a message to the console indicating the loss of the standby adapter.

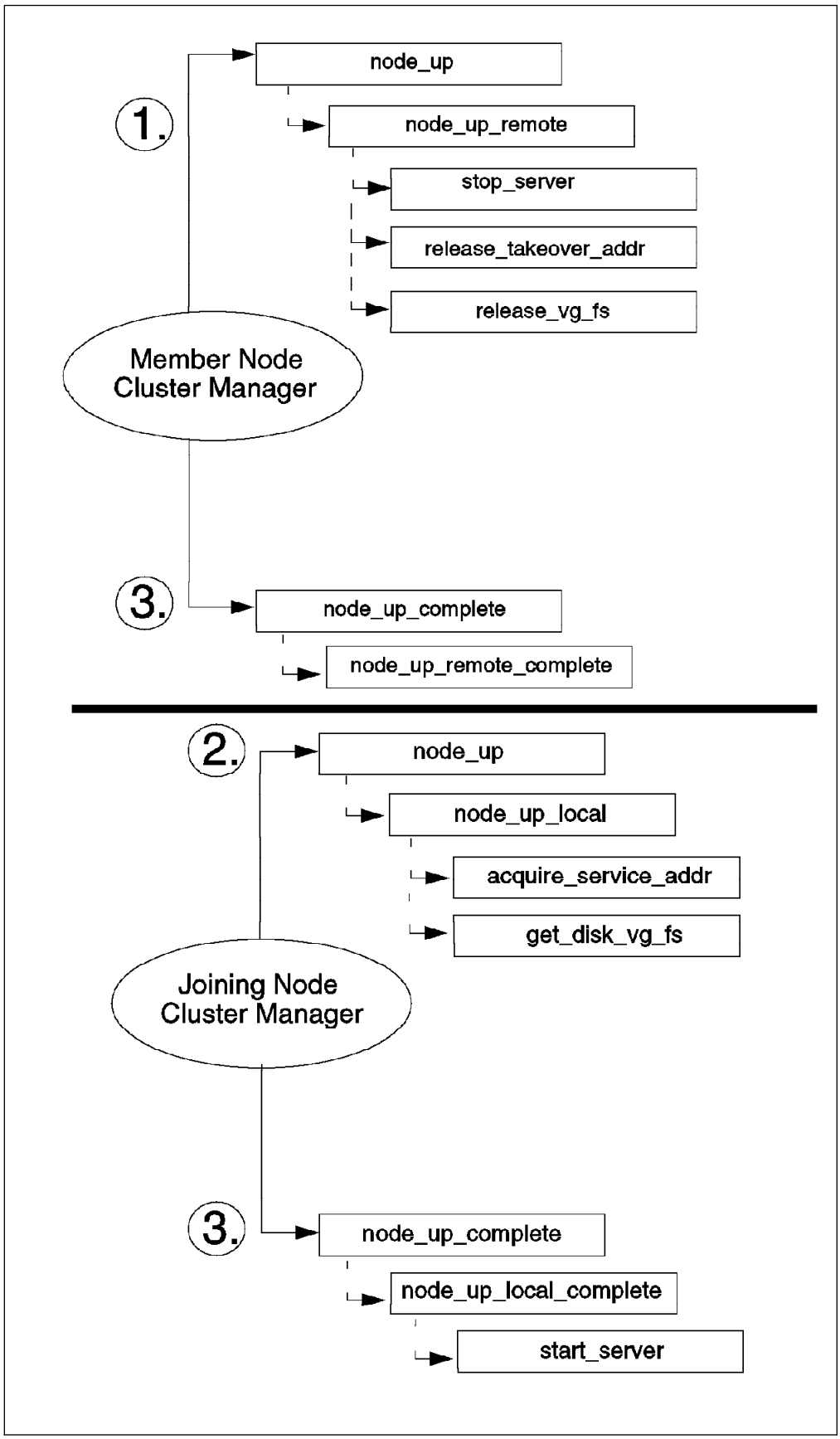


Figure 19. Node Joins an Active Cluster

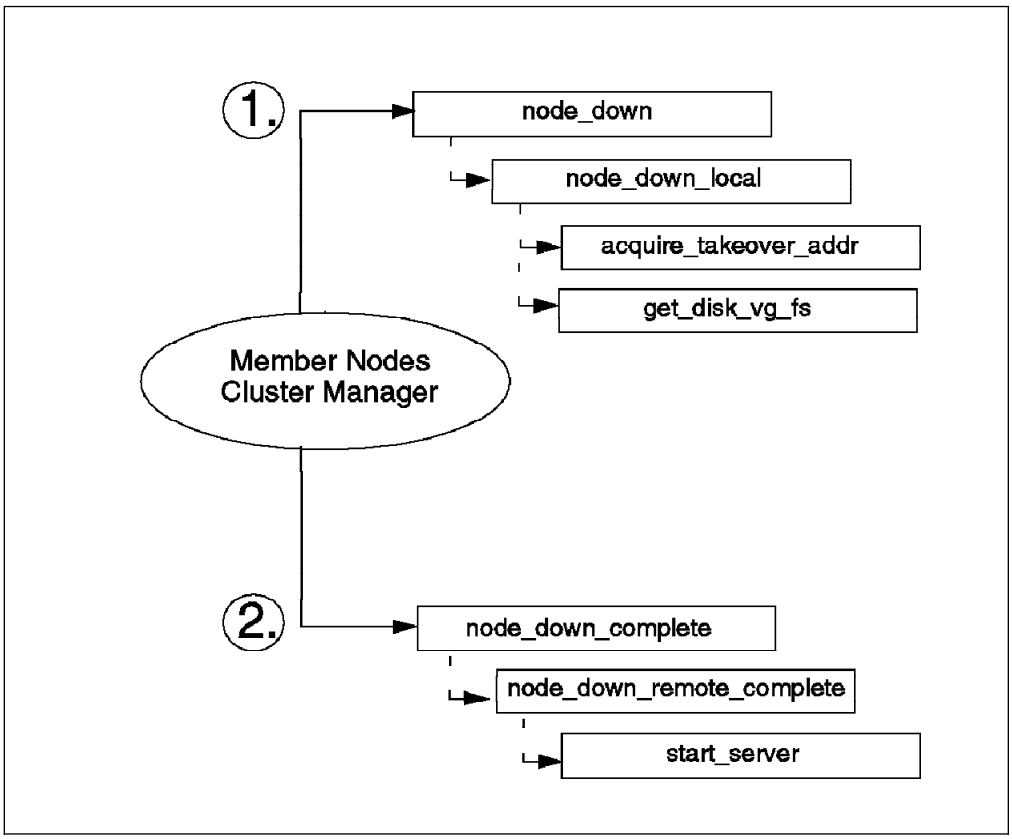


Figure 20. Node Fails

join_standby

This happens when a standby adapter becomes available for use again. The event script sends a message to the console indicating the availability of the standby adapter.

3.2.5.3 Cluster Status Events

If the cluster remains in reconfiguration for more than six minutes after a change in status due to a failure or reintegration, the *config_too_long* event is triggered, and the Cluster Manager starts displaying a message periodically on the console.

If a cluster node has been processing topology changes for more than six minutes, the *unstable_too_long* event is initiated, and the Cluster Manager starts displaying a message periodically on the console.

3.2.5.4 Cluster Event Logs

There are three log files that HACMP uses in order to keep a record of all changes in a cluster's state. You can use these files in order to find out the history of a cluster or to trouble-shoot a cluster that has been misbehaving.

The log files that HACMP writes to are:

1. /usr/adm/cluster.log

   This file contains time-stamped messages indicating the occurrence of event scripts and messages from the HACMP daemons.

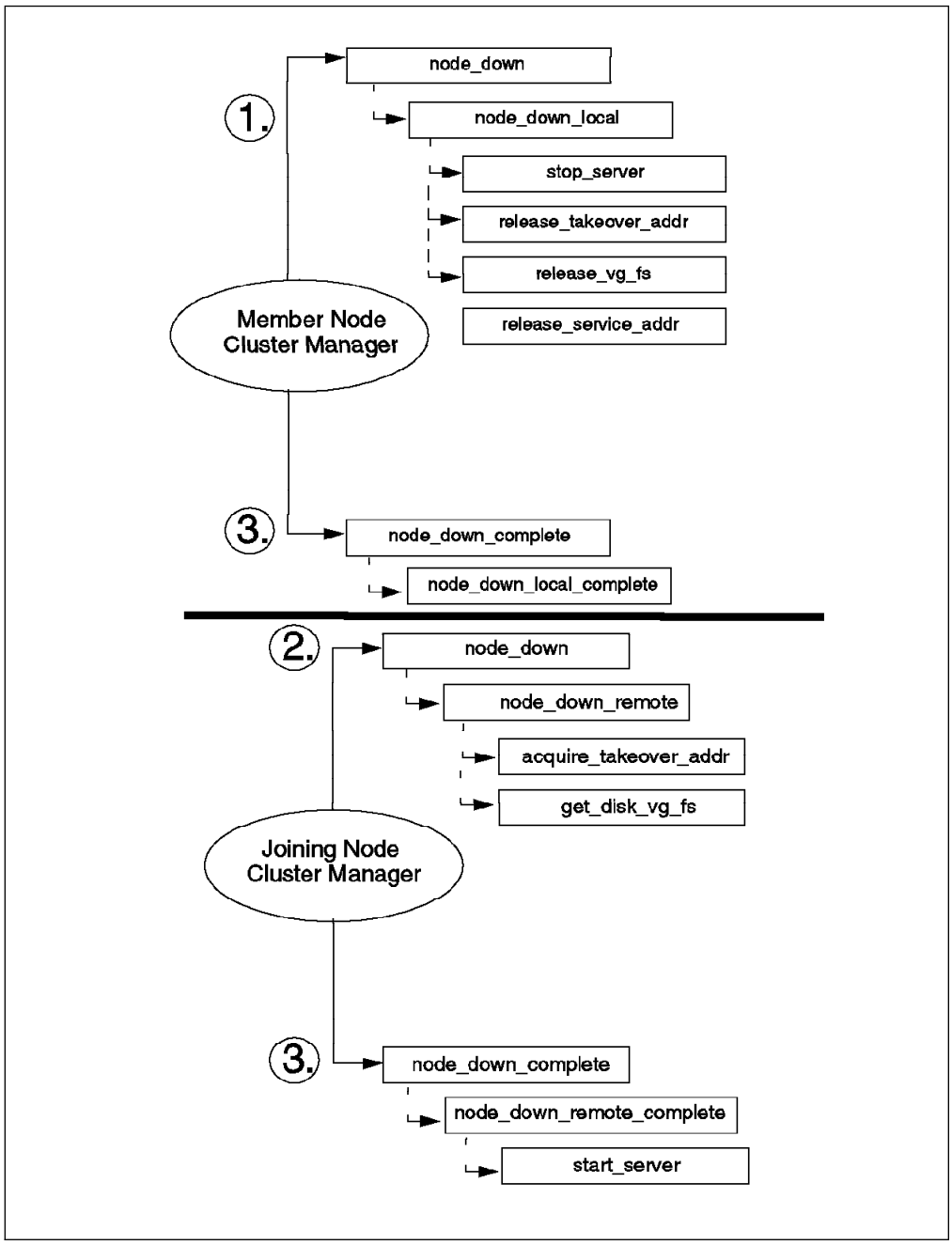


Figure 21. Node Leaves the Cluster Gracefully with Takeover

2. /tmp/hacmp.out

   This file contains messages generated by HACMP scripts. If HACMP is configured to run all scripts in a verbose manner, this file contains a line-by-line record of every command executed by the event scripts. This file is very useful for trouble-shooting.

3. /usr/sbin/cluster/history/cluster.*mmdd*

   This file contains time-stamped, formatted messages generated by HACMP scripts on a particular day. One of these files is created for every day where cluster events occur.

3.2.5.5 Customizing Cluster Behavior

The Cluster Manager has a default behavior, coded into the event scripts, in response to each event. You can add further functionality to the event processing by using the event customization facility that HACMP provides.

This facility includes:

- Pre and post-event processing

  Using this facility, you can specify scripts to be run before and after the execution of the default script for any event.

  It is preferable to use the pre and post-event scripts to do any customization rather than to change any of the default scripts that come with the HACMP product, since the default scripts may be changed by PTFs delivered by IBM over time.

- Event notification

  You can use this facility to notify users or system administrators about the occurrence of a particular event.

- Event recovery and retry

  This facility allows you to specify a script that attempts to recover from the failure of an event script (indicated by a non-zero return code). If the recovery command executes without errors, and the retry count that you have specified for the event is greater than zero, the Cluster Manager reruns the event script.

Figure 22 on page 56 shows the sequence in which these scripts are executed and the conditions that apply to the flow of execution. We will deal with customization in greater detail in Chapter 6, “Cluster Tuning and Customization” on page 159.

3.2.6 Clients in an HACMP Cluster

The HACMP Licensed Program Product comes packaged with a server component and a client component. Both these components need to be installed on the cluster nodes. You can also install the client portion of the HACMP software on any RS/6000 that you want to use as a client to your cluster.

The client portion of HACMP, when installed on an RS/6000 client, starts the clinfo daemon at system startup. It comes with a set of Application Programming Interfaces (APIs) that can be used by client applications to monitor the status of an HACMP cluster and to mask any failures of cluster components from the end user. The APIs are described fully in the HACMP/6000 Version 3.1 *Programming Client Applications* manual. The client portion of HACMP also provides a utility called `clstat`, which you can use to monitor the status of your cluster nodes, networks, and network adapters. This monitor can be viewed either in graphics form or ascii form, depending on the terminal type it is started from. The source code for the client portion of HACMP is also provided with the product, to allow it to be ported to non-RS/6000 clients.

You should have all your users who are accessing cluster services connected over a TCP/IP network and not from serial ports directly attached to one of the cluster nodes. This is because HACMP does not detect or recover from the failure of serial ports and asynchronous adapters. Also, physical replugging of terminals

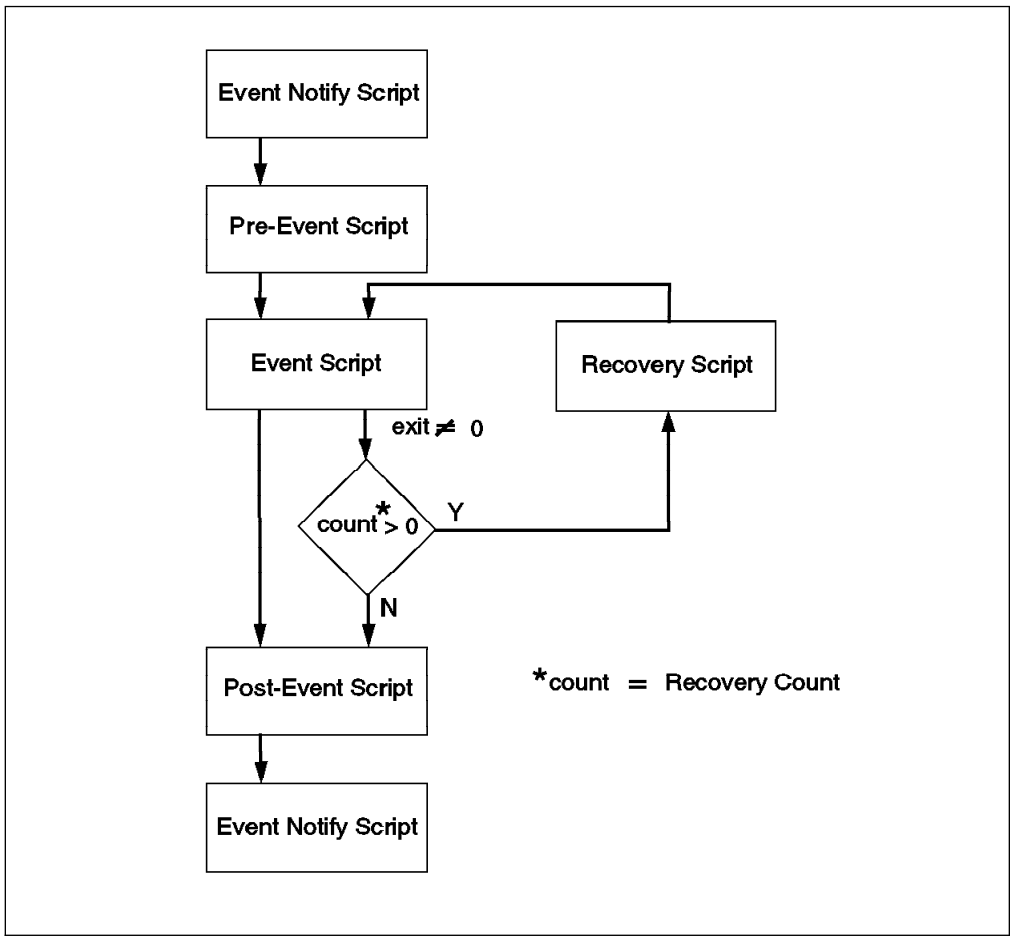


Figure 22. Flow of Execution of Event Scripts

from one machine to another would be required in the case of a node failure. This would, of course, cause much more impact to users in a failure situation.

You can have your users working on terminals connected to a terminal server which is on the public network. In this case, a network adapter failure would be transparent to the users, but a node failure would cause some impact. If the node to which users are logged in were to fail, the telnet sessions of the users would get killed. The users would then have to wait until IP address takeover had completed before they could log on again.

Different terminal servers have different capabilities. For instance, a terminal server may not be able to flush its ARP cache after an IP address takeover. If this is the case with your terminal server, you can configure your service network adapters for hardware address takeover in HACMP. If this is done, the hardware address of the service adapter is moved along with the IP address, and there is no need to flush ARP caches on client devices.

To provide the best failure transparency to your users, you should use client machines, connected to a public network and running a client application. The client application would access a server application on a cluster node at a particular service IP address.

In this scenario, an attempt to initiate a connection during failover, depending on the connection protocol being used (TCP or UDP), would either time out or retry

until IP address takeover was completed. If your client application used UDP to connect to the server node, network adapter and node failures would both be quite transparent to users, resulting in only a delay in response, and not a loss of connection.

If your cluster is configured for concurrent access, you need not wait for IP address takeover. Your client application can use the clinfo APIs to detect node failure and connect to another node where the same disks and applications are available. This would avoid the delay of sending the transaction to a failed node first, and waiting for the reconnection time.

The end users could also be shielded from the failure of a public network, if you customize the cluster to react to a network failure by routing traffic through an alternate network.

3.3 Conclusion

In this chapter, we have seen how we can provide a highly available environment for critical services by using a cluster of RS/6000s, with or without the HACMP software. We have also seen the advantages of using the HACMP product, namely the flexibility in cluster design, responsiveness, and ease of customization.

In the next chapter, we will describe the actual steps involved in setting up an HACMP cluster.

Chapter 4. Hardware Options for HACMP

One of HACMP's key design strengths is its ability to provide support across the entire range of RISC System/6000 products. Because of this built-in flexibility and the facility to mix and match RISC System/6000 products, the effort required to design a highly available cluster is significantly eased.

In this chapter, we shall outline the various hardware options supported by HACMP/6000 Version 3.1 and where appropriate, make reference to HACMP 4.1 for AIX. We realize that the rapid pace of change in products will almost certainly render any snapshot of the options to be out of date by the time it is published. This is true of almost all technical writing, though to yield to the spoils of obsolescence probably means nothing would ever make it to the printing press.

The following sections will deal with the various:

- CPU Options
- Storage Options
- Connectivity Options

available to you when you are planning your HACMP cluster.

4.1 CPU Options

HACMP is designed to execute with RISC System/6000 uniprocessors and Symmetric Multi-Processor (SMP) servers in a *no single point of failure* server configuration. The minimum configuration and sizing of each system CPU is highly dependent on the user's application and data requirements. Nevertheless, systems with 32MB of main storage and 1GB of disk storage would be practical, minimum configurations.

Almost any model of the RISC System/6000 POWERserver family can be included in an HACMP environment and new models continue to added to the list. The following RISC System/6000 model form-factors are currently supported as nodes in an HACMP/6000 Version 3.1 cluster:

- 7009 Compact Server C10
- 7011-2XX Entry-Level Desktop Servers
- 7012-3XX Desktop Servers
- 7013-5XX Deskside Servers
- 7015-9XX Rack-Mounted Servers

For a detailed description of system models supported by HACMP, you should refer to the current Announcement Letters for HACMP/6000 Version 3.1 and HACMP 4.1 for AIX.

HACMP 4.1 for AIX enhances cluster design flexibility even further by including support for the Symmetric Multi-Processor (SMP) family of machines and the Compact Server C20. With the introduction of support for SMP family of processors you are now able to mix and match processors; a cluster can consist of a combination of uniprocessor and SMP nodes. It should be noted, however, that

at the time this chapter was written, HACMP 4.1 for AIX offered concurrent access support for RAID array subsystems only. Concurrent access configurations using serial disk subsystems were not yet supported. Please check with your IBM support person to see what functions are available as you read this document.

4.2 How to Select CPU Nodes for Your Cluster

It is important to understand that selecting the system components for a cluster requires careful consideration of factors and information that may not be considered in the selection of equipment for a single system environment. In this section, we will offer some guidelines to assist you with choosing and sizing appropriate machine models to build your clusters.

Much of the decision centers around the following areas:

- Processor capacity
- Application requirements
- Anticipated growth requirements
- I/O slot requirements

These paradigms are certainly not new ones, and are also important considerations when choosing a processor for a single system environment. However, when designing a cluster, you must carefully consider requirements of the cluster as total entity. This includes understanding system capacity requirements of other nodes in the cluster beyond the requirements of each system's prescribed normal load. You must consider the required performance of the solution during and after failover, when a surviving node has to add the workload of a failed node to its own workload.

For example, in a two node cluster, where applications running on both nodes are critical to the business, each of the two nodes would function as a backup for the other, in a mutual takeover configuration. If a node were required to provide failover support for all the applications on the other node, then its capacity would need to be very much larger than would otherwise be required. Essentially, the choice of a model will depend on the requirements of highly available applications, not only in terms of CPU cycles, but also of memory and possibly disk. Approximately 15MB of disk storage is required for the HACMP software.

A major consideration in the selection of models will be the number of I/O expansion slots needed. The model selected must have enough slots to house the components required to remove single points of failure (SPOFs) and provide the desired level of availability. As mentioned in Section 3.1.2, “Single Points of Failure” on page 26, a single point of failure can be defined as any single component in a cluster whose failure would cause a service to become unavailable to end users. The more single points of failure you can eliminate, the higher your level of availability will be. Typically, you need to consider the number of slots required to support network adapters and disk I/O adapters. Your slot configuration must provide for at least two network adapters to provide adapter redundancy for one service network. If your system needs to be able to take over an IP address for more than one other system in the cluster at a time, you will want to configure more standby network adapters. You may have up to seven standby adapters in a node, slots permitting.

Your slot configuration must also allow for the disk I/O adapters you need to support the cluster's shared disk (volume group) configuration. If you intend to use disk mirroring for shared volume groups, which is strongly recommended, then you will need to use slots for additional disk I/O adapters, providing I/O adapter redundancy across separate buses. See Figure 44 on page 186 for a sample method of eliminating disk I/O adapters as single points of failure.

Because of limited slot capacity, you will not be able to include enough redundant hardware features in a model 2XX to provide a cluster with no single point of failure. For small cluster nodes, that is, nodes requiring no more than one network and two disk I/O adapters for a mirrored shared volume group, 3XX models may be suitable. The limitation is that virtually all the available slots are consumed by adapters at the time of installation just to eliminate single points of failure. This leaves very limited potential for further growth in I/O, or capacity to support additional hardware function such as serial ports, graphics adapters or wide area networks. However, HACMP is configurable to allow you to select the level of availability that can be justified. A single point of failure may be justifiable, based on a calculated risk, but this will obviously vary from site to site. Therefore, 2XX and 3XX models are supported and can be used in cluster configurations.

4.2.1 Node Configuration Guidelines

There are three well defined RISC System/6000 model categories: desktop, deskside, and rack-mounted. Each of these has appropriate application in an HACMP environment as well as limitations. The following clustering examples have been aligned to the RISC System/6000 model range and defined as small, medium, and large nodes.

We have mentioned the importance of sufficient I/O slots to support the elimination of single points of failure. It is easy to become caught up with the high availability design and to neglect the slot requirements for hardware support outside the highly available environment. For each of the categories below, the slot count required for other hardware, such as graphics adapters, 3270 connection adapters, multi-protocol adapters, async adapters, and host interface adapters, should also be considered.

Small Capacity Node: Desktop (3XX) and Compact Server (CXX) models are, in practice, suitable only for clusters consisting of two or three nodes. Beyond this number, the slot limitations of these models prevent the elimination of single points of failure in network and disk I/O resources. Typically, this is a cluster that may support one service network and up to two strings of shared disk. Disk mirroring is implemented across the two buses, to eliminate disk I/O adapters as single points of failure. For a cluster configuration of this type, you will need two network adapters (service and standby) and two disk I/O adapters in each node. If the network type is ethernet, you can use the integrated ethernet port for either the service or standby interface on a 3XX system. Given that a desktop system has four slots, this configuration will leave only one slot for an additional adapter. For example, this type of configuration may suit a highly available graphics workstation environment for newspaper publishing. For other environments, if there were no foreseeable growth in application or data processing requirements, this represents an effective and low cost solution.

The Compact Server systems have only four slots and no integrated ethernet adapter. Therefore, the Compact Server limits you even more because all slots will

be used just to satisfy the requirements for high availability. Any services that require an additional adapter will not be satisfied by a compact server.

Token-ring or FDDI network environments render the suitability of desktop systems equivalent to that of the compact servers. Whereas ethernet environments can make use of the integrated ethernet port, no such feature is available for token-ring or FDDI; you must use an I/O slot to provide token-ring adapter redundancy.

Medium Capacity Node: A medium capacity node is typically a node that supports tens to hundreds of gigabytes of shared disk storage and up to two networks (or three network interfaces).

The deskside uniprocessor servers (5XX) and, in particular, the SMP servers (JXX) are well suited to clusters of up to eight nodes. Currently, HACMP supports no more than eight nodes in a cluster. The uniprocessor servers have seven unused slots in their base configuration. The SMP servers have six unused slots. Coupled to the J02 bus expansion unit, this number increases to fourteen. If you consider the small node capacity requirement of four slots to provide no single point of failure, then the deskside servers will be a far better choice, providing the slot capacity for larger shared disk storage requirements, additional network adapters and other services.

With the ever increasing capacities of RAID subsystems and Serial Storage Architecture (SSA) subsystems, slot capacity is becoming a less significant issue for disk storage. However, if your network performance or availability requirements dictate multiple physical LANs, then multiple adapters and slots, are required. If you require network redundancy and network adapter redundancy, then slots must be available to support the additional hardware.

Large Capacity Node: A large capacity node is typically a node that supports a large amount of shared disk storage and multiple networks or network interfaces. A large amount of shared disk storage could translate to hundreds to thousands of gigabytes. Also, a large capacity node may be required to support six or more network adapters. To achieve this requires a large number of I/O slots. Because the rack-mounted systems have twin micro channel buses that provide fifteen available I/O slots, they are well suited to large capacity clustered environments of up to eight nodes. It should also be noted that the 7013-J30 SMP could also participate in this category owing to its I/O slot expansion capability.

Racks also offer a distinct packaging advantage with the ability to install two CPUs in the same rack and install disk drawers such as the 7135 RAIDiant Array, 7134 High-Density Disk Subsystem and the 9333 Serial Disk Subsystem. Separate power supplies can operate the CPU and disk drawers, so that, if power to one CPU fails, the other is not affected, and takeover of any resources can be performed if necessary. The system rack can also accommodate an uninterruptible power supply, which is a better alternative to battery backup.

For very large clusters with substantial shared disk, CPUs can be housed in dedicated racks and cabled to multiples of racks containing RAID, serial or SCSI disk subsystem drawers.

4.3 Storage Options

One of the major elements in planning a highly available AIX configuration is the shared disk storage devices. The RISC System/6000 family has a comprehensive range of disk storage choices available to HACMP. These choices can be divided into three main categories:

- Conventional SCSI disks, including SCSI-2 Differential and SCSI-2 Differential Fast/Wide.
- RAID arrays (SCSI attached).
- Serial disks, including IBM 9333 and Serial Storage Architecture.

Note: The new Serial Storage Architecture (SSA) was only supported by HACMP 3.1.1. at the time of publishing, but support for HACMP 4.1 for AIX and future releases is expected to be added over time. The HACMP 3.1.1 support is delivered in PTF U438726, and is also included in all follow-on cumulative PTFs.

The decision to use a particular storage subsystem will rely on a number of factors such as inherent availability, function, performance, and cost.

The purpose of this section is to describe the various disk storage technologies and how they fit into a highly available environment. We shall examine the features of conventional SCSI storage technologies, RAID arrays, and serial disks, and offer advice on how to choose the most appropriate type.

4.3.1 SCSI Technologies

SCSI stands for Small Computer System Interface. It is a fully documented ANSI standard (X3.131-1986). The SCSI standard has undergone a number of enhancements, with the SCSI-2 Differential Fast/Wide standard being the most current. Over the past few years, IBM has announced improved SCSI implementations in line with advancements in the SCSI standard:

- 1992 - SCSI-2 (ANSI X3T9.2 186-109)
- 1993 - SCSI-2 Differential (ANSI X3T9.2 186-109)
- 1994 - SCSI-2 Fast/Wide (ANSI X3T9.2/86-109 rev 10h)
- 1994 - SCSI-2 Differential Fast/Wide (ANSI X3T9.2/86-109 rev 10h)

In 1995, IBM announced an enhanced version of the SCSI-2 Differential Fast/Wide implementation.

Note: SCSI Differential is also sometimes called SCSI Differential-Ended. The terms are used interchangeably in this book.

SCSI is a bus-architected interface through which computers may communicate with attached intelligent devices such as fixed disks, tape drives and CD-ROMs.

All the devices attached to the SCSI bus share it by competing for control. This is known as *arbitration*. If a SCSI device wins control of the bus and is an initiator (usually an adapter), it can send commands to another device known as a target (usually a disk or tape drive). Only when that device is finished will the bus be made available to others.

A maximum of seven devices can be directly attached to SCSI, SCSI-2, and SCSI-2 Differential-Ended adapters. With enhancements to the standard, and

dual-porting of adapters like the IBM SCSI-2 Fast/Wide and SCSI-2 Differential Fast/Wide adapters, support for up to seven devices on each port (giving fourteen in total) is available. The IBM Enhanced SCSI-2 Differential Fast/Wide adapter extends device support further by permitting attachment of up to six single ended devices to the internal port and up to fifteen differential devices on the external port. Table 1 summarizes the characteristics of SCSI adapters currently supported by HACMP.

Table 1. SCSI Adapter Characteristics

	Bus	Width	Type	Data Rate	No. of devices	Bus cable length
SCSI-1	Int/Ext	8-bit	SE	4 MB/s	7	6 m
SCSI-2	Int/Ext	8-bit	SE	10 MB/s	7	3.75 m
SCSI-2 DE	Int/Ext	8-bit	DE	10 MB/s	7	19 m
SCSI-2 F/W DE	Internal	8 or 16-bit	SE	20 MB/s	7	6 m
	External	8 or 16-bit	DE	20 MB/s	7	25 m
Enhanced SCSI-2 F/W DE	Internal	8 or 16-bit	SE	20 MB/s	6	6 m
	External	8 or 16-bit	DE	20 MB/s	15	25 m

The signaling mechanism used is different between a SCSI Single-Ended bus and a SCSI Differential bus.

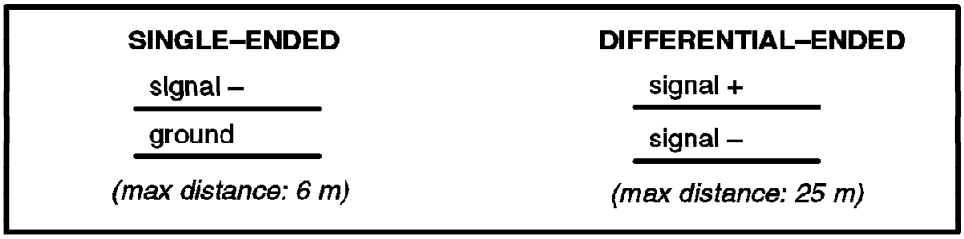


Figure 23. Single-Ended (SE) and Differential-Ended (DE) SCSI

Single-Ended SCSI carries signals on one line, with a ground line as a reference. Differential SCSI carries signals on two lines, a positive and a negative. This provides a much more robust signal, that is less susceptible to line noise. The Differential SCSI standard permits much longer SCSI buses, allowing greater installation flexibility and exploiting SCSI's full device address capability.

The SCSI-2 DE Fast/Wide bus offers the best performance with a peak rate of 20 MB/s. While its electronics run at the same speed as SCSI-2 DE, it is able to carry 16 bits in parallel (2 bytes) rather than 8 bits, and therefore has twice the throughput. The cables in a SCSI-2 DE F/W bus have 68-pin connectors as opposed to 50-pin connectors on SCSI-2 DE.

Differential-Ended SCSI buses are mandated for high availability use, because of the more reliable signal. This is why IBM has withdrawn the SCSI-SE Passthru Terminator Cables from marketing, and made them only available by RPQ. The fact that they are difficult to purchase is further encouragement to use SCSI differential. Existing SCSI-SE configurations continue to be supported, but it is highly recommended that new installations use Differential-Ended SCSI.

4.3.2 Conventional SCSI Disk Options

There are many different SCSI disk options, and the range continues to grow. In this section, we shall limit the discussion to SCSI disks that can be shared on a SCSI bus. Virtually the entire range of external disk enclosures are supported by HACMP. Disk drives installed in CPU enclosures or CPU drawers cannot be used.

The following conventional external SCSI-2 disk subsystems are supported by HACMP/6000 Version 3.1 and HACMP 4.1 for AIX:

- 9334 SCSI Expansion Unit Models 011 and 501
- 7204 External Disk Drive Models 215, 315, 317, and 325
- 7134 High Density SCSI Disk Subsystem

4.3.2.1 7204 and 9334 External SCSI Disk Storage

The 7204 and 9334 range of External SCSI-2 Differential-Ended Disk Drives are supported in shared disk configurations by HACMP. The 7204 units are single disk enclosures, suitable for clusters requiring a small amount of shared disk storage, or to provide storage increments to existing environments.

The 9334 SCSI Expansion Units provide more storage capacity in a single cabinet than the 7204 enclosures. They are suitable for environments requiring a small amount of shared disk storage, and can be shared across a maximum of four cluster nodes on a single SCSI-2 differential bus.

HACMP supports the SCSI-2 Differential-Ended disk drives shown in Table 2.

Table 2. SCSI-2 External Disk Drive Enclosures

Model	No. of Drives	Capacity	Type
7204-215	1	2GB	External SCSI-2 Differential-Ended
7204-315	1	2GB	External SCSI-2 Differential-Ended Fast/Wide
7204-317	1	2.2GB	External SCSI-2 Differential-Ended Fast/Wide
7204-325	1	4.5GB	External SCSI-2 Differential-Ended Fast/Wide
9334-501	1 to 4	2GB to 8GB	Deskside SCSI-2 Differential-Ended
9334-011	1 to 4	2GB to 8GB	Rack Drawer SCSI-2 Differential-Ended

Figure 24 on page 66 shows an example of the 9334-501 SCSI-2 Deskside Expansion Unit connected to two systems. The actual cabling is described in detail in Appendix B, “Disk Setup in an HACMP Cluster” on page 209, and pictured in Figure 54 on page 216.

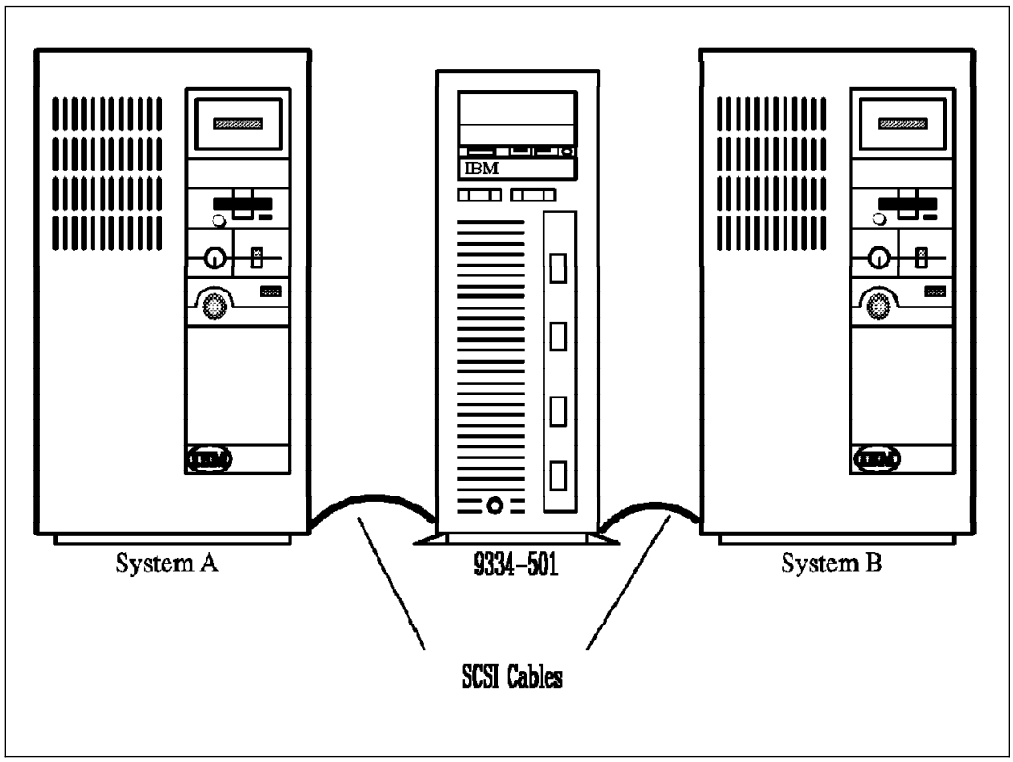


Figure 24. 9334-501 Shared Between Two Deskside Systems

4.3.2.2 The 7134-010 High Density SCSI Disk Subsystem:

The 7134-010 High Density SCSI Disk Subsystem is a rack-mounted drawer that supports from 4 GB to 72 GB of shared disk storage when attached to a system running AIX Version 4.1. Using a single SCSI-2 bus and 4.5 GB differential fast/wide SCSI-2 disk drives, the disk subsystem can accommodate up to 67.5 GB. Using two SCSI-2 buses, up to 72 GB of storage can be configured.

For a RISC System/6000 running AIX Version 3.2.5, the respective single bus and dual bus capacities are 31.5 GB and 63 GB. The base unit includes two 2 GB 3.5 inch differential fast/wide SCSI-2 disk drives. This can be expanded to eight drives (AIX Version 4.1) or seven drives (AIX Version 3.2.5).

Attachment to the RISC System/6000 is through a SCSI-2 Differential Fast/Wide Adapter/A or an Enhanced SCSI-2 Differential Fast/Wide Adapter/A.

An optional expansion feature allows the attachment of up to eight additional drives (AIX Version 4.1) for a total of sixteen, or seven additional drives (AIX Version 3.2.5) for a total of fourteen. The expansion feature has its own SCSI-2 bus and requires an additional SCSI-2 Differential Fast/Wide Adapter/A or an Enhanced SCSI-2 Differential Fast/Wide Adapter/A for attachment to the system.

If the system is running AIX Version 4.1, it is possible to use this expansion feature with a single SCSI-2 bus, by configuring a 0.69 meter Internal SCSI Bus Linkage Cable (feature code 2901). This cable links the two buses in the 7134-010 together into one bus, and allows for up to fifteen disk drives to be included on it.

For a system running AIX Version 4.1, the 7134-010, which takes only four EIA units of rack space, allows up to 405 GB (single SCSI bus) or 432 GB (dual SCSI buses) to be configured in a 7202 Expansion Rack. With dual Micro Channel rack

models, up to fifteen subsystems can be attached, giving a maximum storage capacity of 945 GB.

For more information on choosing an appropriate disk technology, refer to Section 4.3.6, “Choosing a Shared Disk Technology” on page 75.

For information on cabling shared SCSI disks, refer to Appendix B, “Disk Setup in an HACMP Cluster” on page 209.

4.3.3 RAID Disk Array Features

RAID (Redundant Array of Independent Disk) subsystems have become increasingly popular over the past few years. For many sites where data and application availability is paramount, RAID has become the storage technology of choice. Many of the design features of RAID subsystems provide intrinsic redundancy, in the following ways:

4.3.3.1 Data Redundancy

The ability to provide data redundancy is founded on the practice of logically grouping drives together in a logical unit (LUN). If data becomes unavailable because a disk drive has failed, the missing data can be reconstructed from parity data contained on other disks in a LUN.

4.3.3.2 Redundant Power Supplies

Most RAID implementations incorporate backup power supplies, so that alternate power can be supplied to the array component if a primary power supply fails.

4.3.3.3 Redundant Cooling

Extra cooling fans are often built into RAID enclosures to safeguard against fan failure.

4.3.3.4 Online Maintenance

Many RAID subsystems offer the facility to replace a failed drive without the need to take the subsystem offline. The failed drive is replaced and a replacement drive is integrated into the array by reconstructing the missing data from parity data held across other disks in the array.

4.3.3.5 Performance

Depending on the RAID level configured, RAID can offer data throughput potential that can exploit the capacity of SCSI-2 Differential Fast/Wide I/O adapters.

4.3.3.6 Standby Array Controller

A standby array controller can automatically take over for the primary array controller if it fails. IBM has a RAID implementation that allows two controllers to be active at the same time. If either controller fails, the other takes over its load. This capability provides a performance benefit in normal operations, and an availability benefit in failure situations.

RAID's popularity is also attributable to its cost effective data storage. Typically, RAID arrays can support large amounts of disk storage in a single enclosure. Once you have populated the array with, for example, ten to twenty gigabytes of storage, increments become quite cost effective. This is because you are adding disks without the need to also provide supporting hardware and enclosures.

Disk mirroring with conventional disks also provides data redundancy, but RAID makes far more efficient use of disk space, by its use of a relatively small amount of parity data, instead of complete data copies, to provide data redundancy.

For more information about RAID-5 and other RAID levels, refer to Section 2.2.7, “RAID Disk Arrays” on page 15.

4.3.4 RAID Disk Array Options

HACMP supports two different IBM RAID products:

- 7137 External Disk Array Models 412, 413, 414, 512, 513, and 514
- 7135 RAIDant Array Model 110 (HACMP/6000 Version 3.1) and Model 210 (HACMP 4.1 for AIX)

Note: The IBM 3514 Disk Array Models 212 and 213 have also been supported by HACMP, but are being withdrawn from marketing effective in September of 1995.

4.3.4.1 7137 RAID Array

The 7137 RAID Array offers a maximum capacity of 33.6 GB (RAID 0) or 29.4 GB (RAID-5) in a single unit, using 4.5 GB disk drive modules.

The 7137 supports multiple logical units. The multiple LUNs appear as separate disk drives to AIX and, therefore, can be configured into separate volume groups accessed independently by different RISC System/6000s.

The 7137 Array has the following high availability features:

- Support for RAID-5

  The 7137 requires as little as 12.5% of the total disk space for parity data using RAID-5.

- Non-volatile Removable Write Cache

  If the Disk Array controller fails or a SCSI bus fails, the integrity of the data is maintained through the use of the non-volatile removable write cache module. This module can be inserted in a replacement controller, to allow any remaining write operations to complete.

- Redundant Power Supply

  The Redundant Power Supply provides a second source of power. Each power supply provides 50% of the subsystem's needs in normal operations. If one supply fails, the remaining power supply takes over for the subsystem's total needs.

- Hot Spare Disk Option

  When using RAID-5, the 7137 supports a dynamic hot spare disk that allows automatic restoration of data to the hot spare if a drive fails, without operator intervention. The hot spare option provides an offline disk in the RAID cabinet. This capability can be enabled by the customer at install time, or at a later time, if the data is reinstalled.

Because the 7137 supports only one array controller, the array can only be connected to one shared SCSI bus. This means that the array controller, the SCSI cabling and I/O adapters each represent single points of failure. A failure in either

a SCSI adapter or cable will require failing over the affected array resources to another CPU.

4.3.4.2 7135 RAIDiant Array (Models 110 and 210)

The 7135 RAIDiant Array is offered with a range of features, with a maximum capacity of 135 GB (RAID 0) or 108 GB (RAID-5) in a single unit, using the 4.5 GB disk drive modules. The array enclosure can be integrated into a RISC System/6000 system rack, or into a deskside mini-rack. It can attach to multiple systems via a SCSI-2 Differential 8-bit or 16-bit bus.

The 7135 RAIDiant Array incorporates the following high availability features:

- Support for RAID-1, RAID-3 (Model 110 only) and RAID-5

  You can run any combination of RAID levels in a single 7135 subsystem. Each LUN can be running its own RAID level.

- Multiple Logical Unit (LUN) support

  The RAID controller takes up only one SCSI ID on the external bus. The internal disks are grouped into logical units (LUNs). The array will support up to six LUNs, which appear to AIX each as a single hdisk device. Since each of these LUNs can be configured into separate volume groups, different parts of the subsystem can be logically attached to different systems at any one time.

- Redundant Power Supply

  Redundant power supplies provide alternative sources of power. If one supply fails, power is automatically supplied by the other.

- Redundant Cooling

  Extra cooling fans are built into the RAIDiant Array to safeguard against fan failure.

- Concurrent Maintenance

  Power supplies, cooling fans, and failed disk drives can be replaced without the need to take the array offline or to power it down.

- Optional Second Array Controller

  This allows the array subsystem to be configured with no single point of failure. Under the control of the system software, the machine can be configured in *Dual Active* mode, such that each controller controls the operation of specific sets of drives. In the event of failure of either controller, all I/O activity is switched to the remaining active controller.

At the time of publishing, the 7135-110 was only supported by AIX 3.2.5 and the 7135-210 was only supported by AIX 4.1. Therefore, at this time, the 7135-110 can only be used with HACMP/6000 Version 2.1 and HACMP/6000 Version 3.1, while the 7135-210 is only usable in an HACMP 4.1 for AIX cluster. This may change over time, so you should check with your IBM representative for the most current information.

4.3.5 Serial/SSA Disk Storage

In July of 1991, IBM announced a new disk technology for the RISC System/6000 product family called the 9333 Serial Disk Drive Subsystem.

The 9333 Serial Disk Drive Subsystem was designed to provide better performance and configurability for RISC System/6000 systems than that provided by SCSI technology. Four years later, the 9333 Serial Disk Drive Subsystem still provides the best data throughput performance when compared with SCSI.

In addition, at the time of publishing, IBM has just introduced the first products to use a new technology. Called *Serial Storage Architecture* or SSA, the new technology demonstrates significant improvements in performance, function, connectivity and availability, when compared with the 9333 and SCSI technologies.

In this section, we will describe the 9333 technology and its role in a HACMP environment. Because 9333 will ultimately be superseded by SSA, we shall also introduce its concepts here. SSA will play a key role in future HACMP clusters, as a low cost storage subsystem with inherent high availability features.

4.3.5.1 IBM 9333 Serial-Link Disk Subsystem

The 9333 Serial Disk Drive Subsystem is available in two model groups, each offering a deskside and a drawer configuration. The first, and older, group is the models 500/010. Following the standard IBM disk enclosure model numbering convention, the model 500 is a deskside unit, and the 010 is a rack drawer.

The second, and preferred, group is the models 501/011. These are preferred for reasons of price performance, attachability, and function in an HACMP cluster. The 501/011 models can be used in any HACMP configuration, concurrent or non-concurrent, while the 500/010 models are restricted to non-concurrent use only.

Each 9333 Serial Disk Drive Subsystem disk drawer or tower can contain up to four disk drives. These drives can have a capacity of 857 MB, 1.07 GB, or 2.0 GB. Each tower or drawer also has two serial-link connections in its standard configuration. This means that each 9333 drawer or tower unit can be connected to two different RISC System/6000s or adapter cards. Using the optional Multiple Systems Attachment features, available on the models 501/011 only, you can expand this capability by attaching up to eight system units.

Figure 25 on page 71 shows schematically how the 9333 unit connects to two serial-link adapters (or two systems). Each of the four ports on the adapter can connect to a single 9333 Serial Disk Drive Subsystem rack or tower that contains a controller and provision for up to four disk units.

Key features of the 9333 Disk Subsystem technology include the following:

- Better adapter slot utilization in the RISC System/6000

  Each RISC System/6000 has a limited number of adapter slots. A maximum of six disk drives, with unique SCSI IDs, can be connected to a shared SCSI bus. On the other hand, the 9333 Serial Disk Drive Subsystem allows up to sixteen disk drives to be connected to a single serial-link adapter. Using the optional Multiple Systems Attachment feature of the 9333, sixteen disk devices can be connected and shared with up to eight system CPUs. Figure 26 on page 72 shows an example of a four node cluster sharing a single 9333 Serial Disk Drive Subsystem

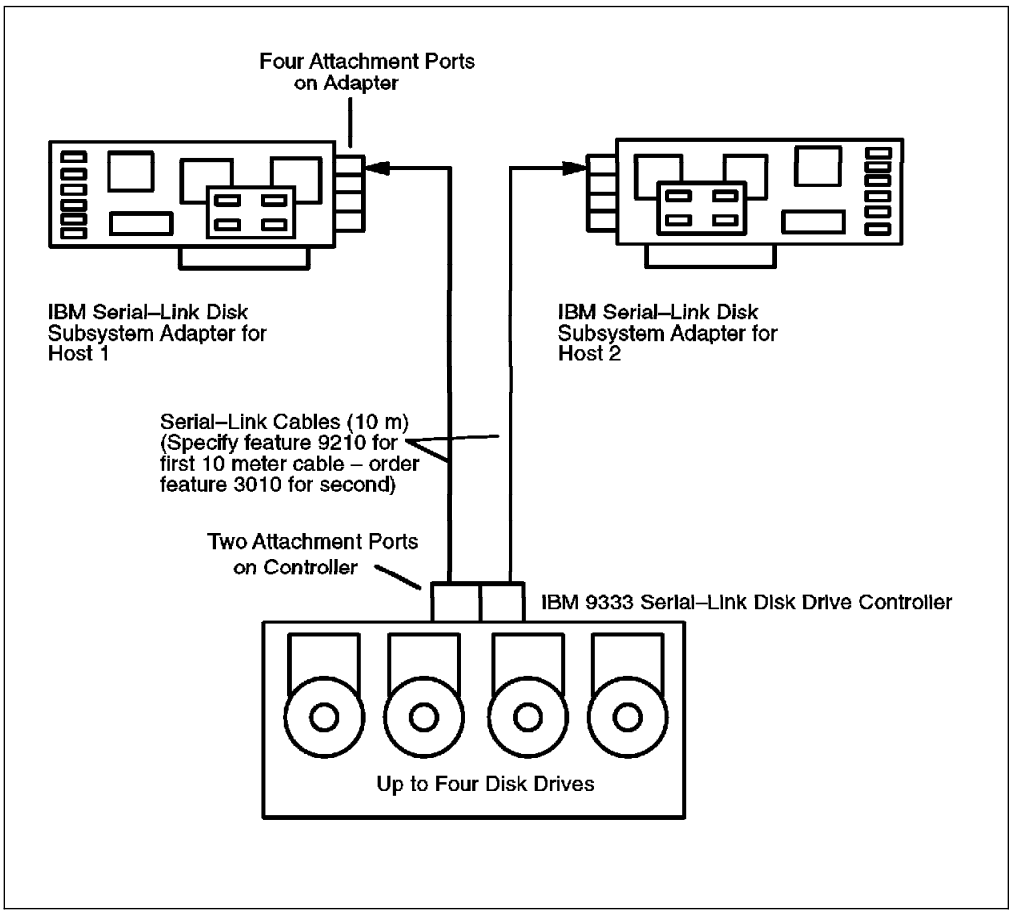


Figure 25. 9333 Serial Disk Drive Subsystem System Attachment

- Better performance

  The connection between the 9333 units (each of which contain up to four disk drives) and the adapter is an 8 MB per second, full duplex connection. Each adapter can support four of these links. The existing fastest RISC System/6000 SCSI-2 fast/wide interface features a single 20 MB per second interface. The 9333 features 32 MB per second, full duplex transfer capability when multiplexed over four disk drives. There are also many enhancements to the data access protocols that boost performance still further.

- Enhanced reliability, availability and serviceability

  The 9333 Serial Disk Drive Subsystem uses a point-to-point cabling scheme, allowing connection distances up to 10 meters. In a twin-tailed environment, this provides a potential cable length of 20 meters from adapter to adapter.

  There is also a cabling RPQ for the 9333 (number 7J0354) which permits connection distances of up to 600 meters, using optic fiber technology. With SCSI-2 Differential technology, the historical issues associated with short cabling lengths, have largely been overcome. Nevertheless, because of its simplicity, serial link cabling is more robust and reliable than its larger, multi-wired SCSI alternative.

  Also, the daisy-chain nature of SCSI requires that every disk connected to the bus is cabled twice; once to bring the cable in and once to take the cable out. Because each cable in the 9333 Serial Disk Drive Subsystem scheme is point-to-point, device problems affect only a single device. Therefore, the 9333

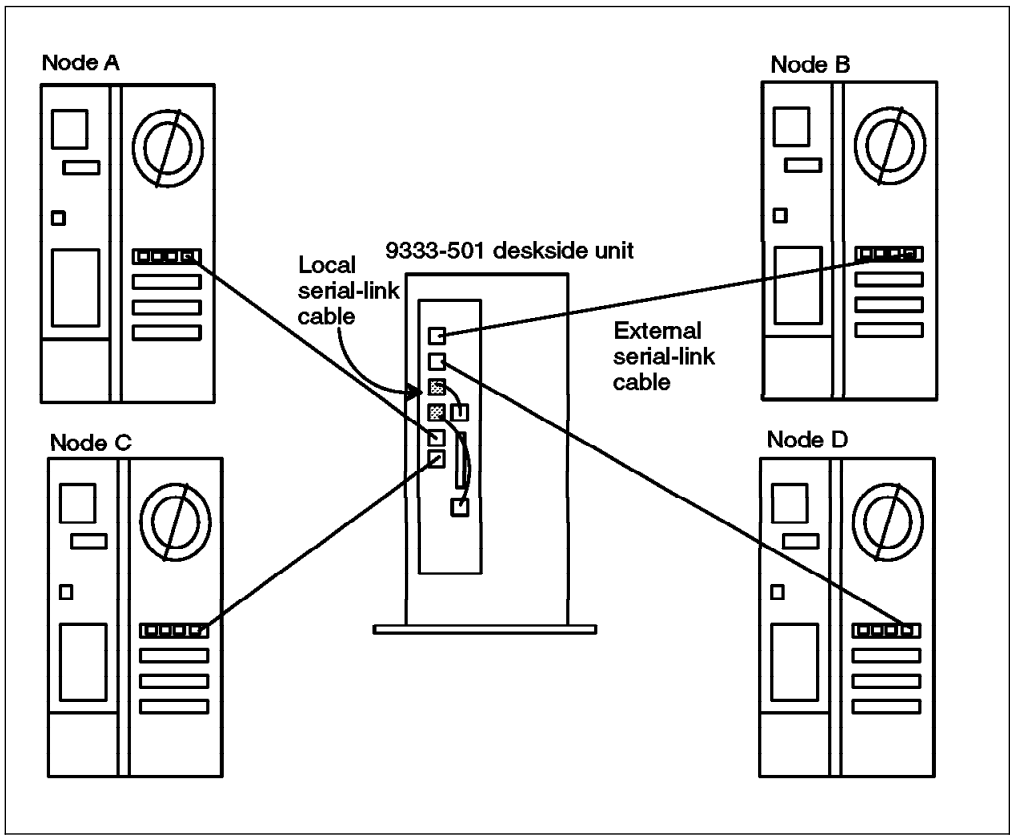


Figure 26. Four-Node Cluster with Shared 9333 Serial Disk Subsystem (Rear View)

Serial Disk Drive Subsystem provides easier installation, fault isolation and maintenance.

The 9333 Serial Disk Drive Subsystem also features advanced built-in error detection and correction functions. The 9333 Serial Disk Drive Subsystem can detect and correct many errors in the protocols of the serial-link itself, perform Power On Self Test (POST) procedures to detect problems, and detect cable faults such as open or short-circuited cables.

- Hot Plugability

  The 9333 Disk Subsystem allows addition or replacement of individual disk devices while the system is running. Disks can be added or removed, defined to the operating system, and added or removed from volume groups, all without taking down the system, or even taking the affected volume group offline. This is much less disruptive than in a SCSI environment, where the system must be shut down for any changes on the SCSI bus, and is very advantageous in a high availability configuration.

- Compatibility with SCSI-2

  To provide compatibility with existing software, the 9333 Serial Disk Drive Subsystem implements a standard SCSI-2 command set.

4.3.5.2 Serial Storage Architecture (SSA)

Serial Storage Architecture (SSA) will play a very important role in the design of future HACMP environments.

SSA is a powerful high performance serial interface designed especially for low cost connections to storage devices, subsystems, servers and workstations. It is a two signal connection (transmit and receive), providing full duplex communication. For transmission along a copper wire, it uses a differential pair method, requiring only four wires. The signal can also be transmitted along a fiber optic cable.

The characteristics of SSA are summarized as follows:

- Topology

  The SSA design allows an extremely flexible assortment of connection options. SSA networks can be connected in simple strings or loops, or more complex switched strings. The flexibility allows trade-offs to be made between cost, performance and availability.

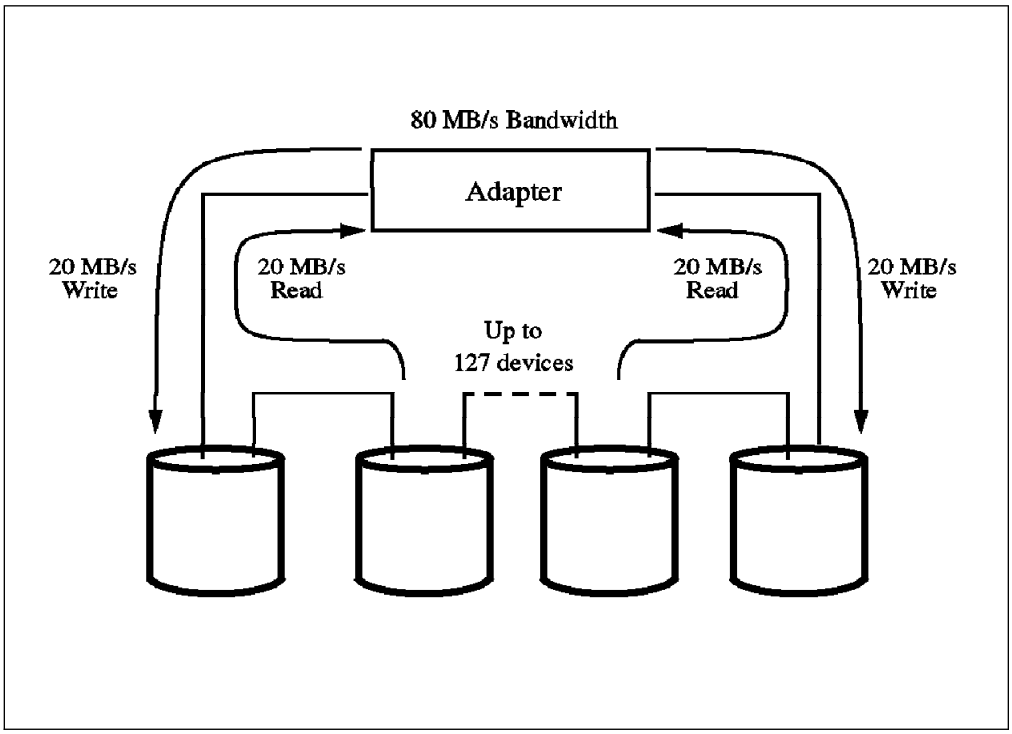


Figure 27. Typical SSA Loop Topology

- Flexible Configuration

  An SSA node is defined as any SSA-capable device. This can be either a disk drive, an adapter, or a switch. Each SSA node can support multiple independent serial ports. A typical disk drive or adapter will have two ports, but an SSA switch can have up to 126 ports.

  The connection between two SSA ports can be up to 20 meters in length, while maintaining a very low error rate. This distance can be increased to 2.5 km by using fiber optic extenders.

  SSA provides hot plugging and automatic configuration of devices. No address switches are required; SSA is architected to perform self addressing. This means that SSA networks can be easily extended or reconfigured with

minimum disruption to online operation. Multiple alternate paths to each SSA node are easily configured, providing inherent high availability and making shared storage far easier to implement than is the case with current interfaces. The addressing scheme used within SSA supports several network topologies:

String A linear network of up to 129 nodes.

Loop A cyclic network of up to 128 dual-ported nodes.

The loop topology is best suited to an HACMP environment. You can see how this topology can be configured by the example shown in Figure 28

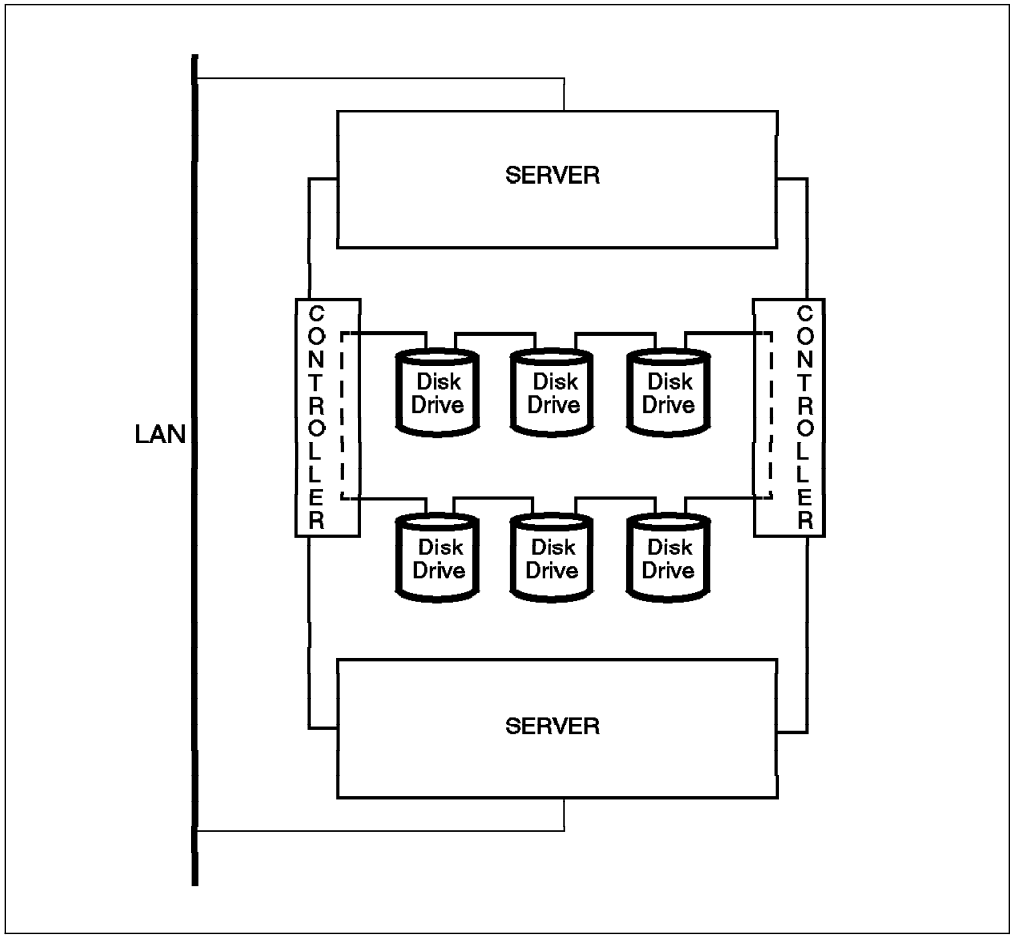


Figure 28. SSA-Based High Availability Servers

The diagram shows how SSA disk devices can be shared in a network between two server nodes in a highly available configuration. The network contains two loops. The outer loop connects the servers and the array controllers; the inner one connects the controllers and devices. Notice the cyclic paths; each device has an alternate path. This means that the SSA network is tolerant of a single fault. If a link fails, the failing link can be identified, and there may be some loss of bandwidth, but in other respects, the operation of the network is not affected. If a device fails, all other devices can still be accessed.

Switched Supports up to 126 ports per switch permitting the configuration of large numbers of node devices.

Note: The limits given above are architectural limits, and do not necessarily mean that current SSA product implementations will support the limits.

- Performance

  Each SSA link is full duplex and multiplexed, simultaneously operating at 20 MB/second in each direction. In addition, each link in an SSA network can operate independently of the others. This feature is known as *spatial reuse*. An SSA loop provides a total bandwidth of 80 MB/second at each node. This represents a significant increase in performance over the 9333 serial technology.

- Reliability and Availability

  Each SSA node performs extensive error checking. Should errors occur, SSA provides transparent recovery. When SSA devices are configured in a loop, alternate paths to each device ensure there is no single point of failure.

- Serviceability

  In keeping with the architectural philosophy of the 9333 technology, SSA networks consist of point-to-point links, not multi-dropped buses like SCSI. This makes faults easier to isolate and diagnose. In addition, each SSA node has the capability to detect open or short-circuited connections, simplifying the repair of faults. No discrete terminators are required.

  These characteristics, coupled with hot plugging, automatic configuration, and alternate path capabilities, allow SSA subsystems to be serviced with little or no disruption to online availability.

Despite the excellent performance of the 9333 technology, it is a proprietary architecture. On that basis, there has been some reluctance on the part of the industry and customers to accept it. IBM has followed a different strategy with SSA. The specifications for SSA have been reviewed and approved by the industry user group and are now with the ANSI X3T10.1 subcommittee for approval as an ANSI standard. Other disk manufacturers are also preparing to introduce their own SSA devices, so they will be available from a variety of sources.

There is more information on SSA, and on connecting it into your cluster, in Appendix B.4, “Serial Storage Architecture (SSA) Subsystems” on page 226.

4.3.6 Choosing a Shared Disk Technology

The aim of this section is to help you select the most appropriate storage technology for your cluster. Here, we are concerned only with selecting a technology for use as a shared disk resource.

Let us start by considering the factors that will potentially influence your decision. These are:

- Cost
- Technology
- Availability requirements
- Performance requirements

A simple approach is to first define your boundaries or constraints. Having done that, you are then in a position to understand your options. For instance, if cost is a major constraint, then in many cases this will cause some level of sacrifice in the technology that can be used in your design. If the elimination of all single points of failure is a major imperative, then cost may not be so important. Inevitably, there are trade-offs which result in a balance of technology and cost. You generally know when the optimum point has been reached, and an additional cost is perceived as having a diminished return. This might transform a so-called need-to-have feature into a nice-to-have feature.

Application and data storage requirements will significantly affect the choice of storage technology, especially if the storage requirement is large. In cases where large capacities are required, the cost of disk storage may far outweigh the cost of system CPU and other components put together. Also, sites needing to support a large amount of disk storage will generally be more nervous about the prospect of data loss, simply because of the sheer volume of data and the potentially long restore/recovery times that would follow a failure. This, of course, is not meant to suggest that the importance of data to a small capacity site is any less critical. Sites that place more emphasis on critical data availability will, more often than not, be looking for disk storage technologies with the most advanced availability features.

With the comprehensive range of storage products IBM has to offer, choosing the most appropriate technology for your cluster can be a daunting task. Knowing how much shared disk space is required is the starting point.

Since we are designing the storage for an HACMP cluster, data availability is a prime consideration. This means you need to factor the extra capacity required by mirroring or RAID into your design. Also, the site's future storage growth projections must be considered. For a given storage capacity, the number of I/O slots consumed by adapters varies for the different technologies. You do not want to be left in the position of having to throw away equipment because the initial choice had insufficient growth potential or because there were insufficient I/O slots to support an additional storage subsystem.

With the above in mind, the choice comes down to matching the best availability, the best performance, and the best potential for future growth with the best price. Here are some examples that hopefully make the job easier. We have adopted some arbitrary capacities of 10 GB, 100 GB and 500 GB and classified them as small, medium and large, respectively. Note that this is intended to be a guide and it is based on features, function and pricing available at the time of writing. The product range continues to evolve rapidly, and as pricing structures evolve with it, various products fall into and out of favor. Nevertheless, the objective here is to assist your decision process by examining some issues at the storage capacity breaks we have chosen.

4.3.6.1 Small Capacity (10 GB)

At this capacity, the 7137 RAID array or a mirrored implementation of the 7204 External disk drive currently offer the most cost effective solutions.

The 7137 has an important disadvantage in a high availability cluster because of its ability to support only one RAID controller. This makes the RAID controller, SCSI cabling and I/O adapters all single points of failure. A mirrored implementation of the 7204 drives, across separate SCSI buses, eliminates all single points of failure.

The 7135 is not cost effective for this low capacity requirement. The 9334 may no longer be a cost effective solution because the larger capacity 4.5GB drives, which are supported by 7137 and 7204, cannot be used in the 9334 tower. The new 7131-105 Storage Tower, although not yet supported by HACMP at the time of writing, will also provide a storage solution similar in price to that of the 7204 solution. The addition of more disks to the 7131 will also be cost effective. The 9333 subsystem is a more expensive alternative, but may be justifiable if high performance is of enough concern.

At this level, the 7134 High Density Disk Subsystem compares favorably with the 7137 and 7204 solutions. However, its rack-mount requirement may cause some concerns about floor space in a small installation.

4.3.6.2 Medium Capacity (100 GB)

For medium sized storage requirements, the 7135 RAIDiant Array is the most cost effective shared storage solution. At this capacity, it is cheaper than any of the other SCSI and RAID products. It also has the distinct advantage of offering a second RAID controller, and hence the ability to eliminate single points of failure in the configuration. For no single point of failure, two spare slots would be required in each system to accommodate the prerequisite SCSI I/O adapters.

SSA disks will also compete effectively in this capacity range. They are, at the time of publication, supported in two node configurations only, but more extensive node sharing support is expected in the future.

4.3.6.3 Large Capacity (500 GB)

Currently, the 7135 RAIDiant Array is the most cost effective storage product at this capacity level. Five units, running RAID-5 and using 4.5 GB disk drives, would be required to support this capacity. Since you are able to daisy-chain up to two 7135 RAIDiant Arrays on a single shared bus, six I/O slots would be required in each system to accommodate the SCSI adapters needed to support five subsystems, with dual controllers on separate buses.

If mirrored SCSI storage is desired, the 7134-010 is the only technology that can approach this capacity. It not possible to support a full 500 GB, with a no single point of failure mirroring configuration, because of the number of slots that would be required for SCSI adapters. For no single points of failure, fifteen slots would be needed on each node for SCSI adapters only.

In conclusion, we should mention that as SSA products continue to be introduced, the landscape for shared storage implementation will change dramatically. It is expected that, because of its ability to support large amounts of storage capacity and its low cost, SSA will compete effectively for all storage capacity requirements.

4.4 Connectivity Options

The lifeline of any cluster is the network over which the cluster nodes are connected. In an HACMP cluster, communication networks are also the channels over which the HACMP daemons interact to detect faults and serialize concurrent access to shared storage.

In this section, we will cover the different types of networks supported for an HACMP cluster, the network adapters available from IBM, and the factors to consider when choosing one or more networks for your cluster.

There are two categories of networks that must be used in an HACMP cluster:

- TCP/IP networks
- Non-TCP/IP networks

TCP/IP networks, as the name suggests, use the Transmission Control Protocol/Internet Protocol for communication. In an HACMP cluster, non-TCP/IP networks are point-to-point connections between nodes, such as serial RS-232 lines or target mode SCSI connections.

4.4.1 TCP/IP Networks

TCP/IP networks are used as public and private networks in HACMP clusters. Public networks are used by cluster nodes and client machines together, while private networks are meant to be used by the cluster nodes only, for communications of the Cluster Lock Managers on each node. Different TCP/IP network media have different characteristics, which you have to measure against your requirements.

4.4.1.1 TCP/IP Network Technologies

There are several TCP/IP network technologies available in the market today from IBM and other companies. We shall look at three of the most commonly used ones for HACMP clusters:

- Ethernet
- Token-Ring
- Fiber Distributed Data Interface (FDDI)

Table 3 lists the characteristics of these networks:

Table 3. TCP/IP Network Types and Their Attributes

Network Type	Bandwidth	Access Method	Media
Ethernet	• 10 Mbps • Shared • Half-Duplex	CSMA/CD	Coax, STP, UTP, and Fiber
Token-Ring	• 16/4 Mbps • Shared • Half-Duplex	• Token • Eight levels of access priority • Shared access under stress	STP, UTP, and Fiber
FDDI	• 100 Mbps • Shared • Half-Duplex	• Token • Access priority • Shared access under stress	STP, UTP(5), and Fiber

Note: STP = Shielded Twisted Pair, UTP = Unshielded Twisted Pair

Ethernet networks provide adequate bandwidth for character-based applications running over a workgroup. As the size of the workgroup increases, the collision domain gets larger and as a result, the access method becomes stressed. This can be alleviated to a certain degree by the segmentation of the workgroup into smaller domains.

Token-ring networks, due to their access method and higher bandwidth, provide predictable performance even under heavy loading. These networks provide a high level of management at the physical layer by automatically recovering from wiring and adapter failures, and by isolating excessive soft errors to a fault domain. The cost of components on a token-ring network is higher than an ethernet network.

FDDI networks provide a full 100 Mbps burst at the desktop and provide predictable performance under heavy loading. You can configure an FDDI ring with built-in reliability with the dual ring option. The cost of setting up an FDDI network is higher than an ethernet or a token-ring network.

4.4.1.2 IBM Network Adapters

The ethernet network adapter for the RS/6000 available from IBM is:

- Ethernet High-Performance LAN Adapter (feature code 2980)

The token-ring adapters for the RS/6000 available from IBM are:

- Token-Ring High-Performance Network Adapter (feature code 2970)

The FDDI adapters available for the RS/6000 from IBM are:

- Fiber Distributed Data Interface Adapter (feature code 2720)
- Fiber Distributed Data Interface Dual Ring Upgrade Kit (feature code 2722)

  This adapter requires a system with an existing RISC System/6000 FDDI Adapter (feature code 2720).
- FDDI-Fiber Dual-Ring Upgrade (feature code 2723)
- FDDI-Fiber Single-Ring Adapter (feature code 2724)
- FDDI - Fiber Single-Ring Adapter

4.4.2 Choosing a Network for Your Cluster

There are several factors, some of them related to HACMP, that affect your choice of a network for your cluster. These are:

- Price
- Performance
- Reliability
- Cluster event detection rate
- Support of IP and hardware address takeover

An ethernet network would be the best option for a public network if you keep the collision domain small, and if you do not need the bandwidth and reliability provided by a token-ring or an FDDI network. It is cheaper than the others, has the fastest HACMP error detection rate (six seconds), and supports IP and hardware address takeover.

If bandwidth or reliability are a major requirement, you will have to go for either a token-ring or an FDDI network. The error detection rate (12 seconds) on a token-ring network is twice as slow as on an FDDI network (six seconds). However, an FDDI network has a much higher bandwidth. On the other hand, hardware address takeover is not supported on an FDDI network (only IP address takeover), the number of slots occupied is greater, and the price is higher. Here, you need to make a trade-off.

For a private network, ethernet is the best option in most cases, given the improvements in the lock manager to minimize lock traffic. If you expect heavy lock manager traffic on the private network, you could go for an FDDI ring, since it has

the highest bandwidth. You do not need hardware address takeover on a private network.

4.4.3 Non-TCP/IP Networks

Non-TCP/IP networks are called *serial* networks in HACMP clusters. The two types of networks that can be used as serial networks in an HACMP cluster are:

- RS-232 Serial Link

  This is a point-to-point network achieved by connecting an RS-232 cable between the serial ports of two cluster nodes.

  Figure 30 on page 89 shows the components required to form the serial connection between two nodes.

- Target Mode SCSI

  This provides a communications network over a shared SCSI bus. SCSI-2 Differential (8-bit and 16-bit) adapters can act as both initiators and targets.

  You do not require any special hardware apart from the SCSI bus to set up a target mode SCSI connection.

Since serial networks in an HACMP cluster are used only for sending keepalive packets, the bandwidth of either of these networks is more than sufficient. Most clusters are configured with RS-232 links as serial networks. However, if you are using SCSI differential disks as shared disks in your cluster, you should also configure a target mode SCSI network, to provide as many keepalive paths as possible between nodes.

Chapter 5. Setting Up HACMP for AIX

In this chapter, we shall take you through the steps to set up an HACMP cluster.

The two main sections of the chapter will deal with the following subjects:

- Installing and configuring a new HACMP/6000 Version 3.1 cluster.
- Upgrading from HACMP/6000 Version 2.1 to HACMP/6000 Version 3.1.

The setup and customization of HACMP is much easier if it is well planned in advance. HACMP/6000 Version 2.1 saw the introduction of a SMIT interface for cluster configuration, which replaced the manual editing procedure required for configuring previous versions. Another improvement has been the Global ODM, which allows you to do all configuration on one node, with easy propagation to other nodes in the cluster.

We do recommend the following, as you do your planning and setup:

1. Use the Planning Worksheets in the *HACMP Planning Guide* to document your node configurations. The worksheets provide an invaluable record of the cluster's configuration that should avoid confusion when future reference is needed.
2. Follow the steps carefully, especially in the network and disk configuration. Observe any special requirements or configuration steps that may be required along the way. A systematic approach is essential.
3. Thoroughly test your setup to ensure it is demonstrating the behavior you expect.

In this chapter, we shall illustrate the installation and customization of a basic HACMP cluster. The customization methodology is similar, regardless of the cluster's operational mode (one-sided takeover, mutual takeover, and so on). Therefore, working through this example will help prepare you to work with any HACMP cluster setup.

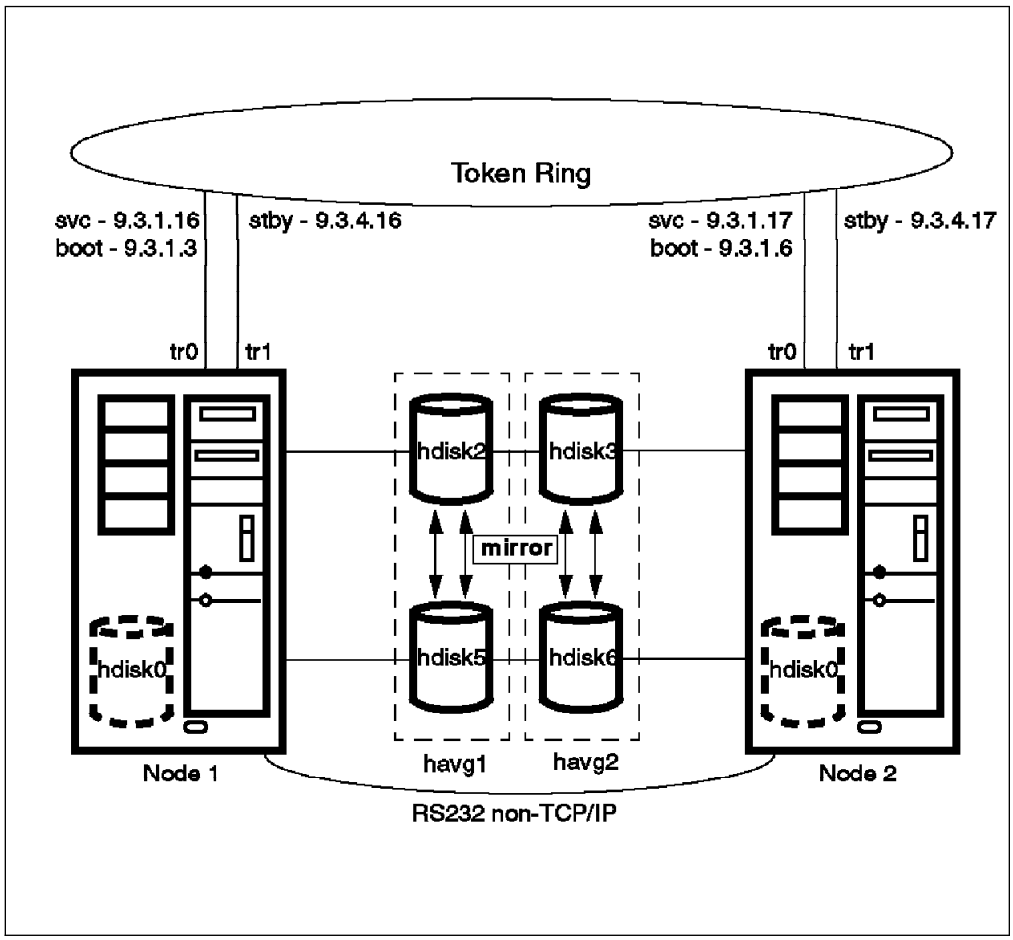


Figure 29. Sample HACMP Cluster Configuration

A sample cluster configuration is shown in Figure 29. In this configuration, the following observations can be made:

- Node 1 owns the shared volume group havg1, which is mirrored across the physical volumes hdisk2 and hdisk5. The shared volume group havg1 will contain the filesystem /sharedfs1.
- Node 2 owns the shared volume group havg2, which is mirrored across the physical volumes hdisk3 and hdisk6. The shared volume group havg2 will contain the filesystem /sharedfs2.
- The two filesystems, /sharedfs1 and /sharedfs2, will be NFS cross mounted to Node 2 and Node 1, respectively.
- If Node 2 fails, Node 1 will take over Node 2's service IP address and shared volume group havg2. If Node 1 fails, Node 2 will takeover Node 1's service IP address and shared volume group havg1.

This type of configuration is called *Mutual Takeover with Cascading Resources*.

5.1 Installing and Configuring HACMP/6000 Version 3.1

This section uses the cluster example shown in Figure 29 on page 82 and illustrates each step required to configure your cluster. The steps are divided into two major sections:

- Preparing AIX for an HACMP Cluster - Setting up the hardware and software in AIX.
- Installing and Configuring an HACMP Cluster - Configuring the cluster to handle the resources to your specifications.

The layout of this section closely parallels the *HACMP Installation Guide Version 3.1* Chapter 1. Installing and Configuring an HACMP Cluster.

5.2 Preparing AIX for an HACMP Cluster

It is important to complete all work required to configure your hardware and AIX before moving into the cluster configuration. You will save much time and effort by successfully completing this work first.

Appendix A in the *HACMP/6000 Planning Guide Version 3.1* guide contains a series of worksheets to assist you in the planning of HACMP installation. The worksheets included are the following:

- TCP/IP Networks
- TCP/IP Network Adapter
- Serial Networks
- Serial Network Adapter
- Shared SCSI-2 Differential Disk
- Shared 7135-110 RAIDiant Disk Array
- Shared 9333 Serial Disk
- Non-Shared Volume Group (Non-Concurrent and Concurrent)
- Shared Volume Group (Non-Concurrent)
- NFS-Exported File System (Non-Concurrent)
- Shared Volume Group (Concurrent)
- Application Server
- Resource Group
- Cluster Event

Sample filled out worksheets are also provided in the Planning Guide to help you understand the way in which you need to complete them.

You should fill out all appropriate worksheets with information relating to your planned configuration before beginning any configuration activities on the system. Be assured that the time you spend documenting your environment will be time well spent, as it will ease the setup and support of your environment enormously.

5.2.1 Configuring IP Networks

Ensure that all network adapters you intend to use are at least visible to AIX.

```
# lsdev -Cc adapter -t tokenring -H
```

Your machine should have at least two network adapters of the same type. In our example, we have used token-ring adapters, hence the output of the `lsdev` command looks like this:

```
name status    location description

tok0 Available 00-03    Token-Ring High-Performance Adapter
tok1 Available 00-04    Token-Ring High-Performance Adapter
```

If your system uses ethernet, FDDI, or some other network type, you can replace the argument `tokenring` with the appropriate argument in the above command syntax for `lsdev`.

We can see that our first token-ring adapter, tok0, is located in slot number 3, and that our second token-ring adapter, tok1, is located in slot number 4. In this example, tok0 will be the *service* adapter and tok1 will be the *standby* adapter.

To list the available token ring network interfaces, use the `lsdev` command again:

```
# lsdev -Cc if -s TR -H
```

The output is:

```
name status    location description

tr0  Available          Token Ring Network Interface
tr1  Available          Token Ring Network Interface
```

If your system uses another type of network, you can replace the argument `TR` with the appropreiate argument for your network in the above command syntax for `lsdev`.

When configuring your cluster, you will need to run the above commands on all nodes intended for use in the cluster. If the adapters are marked `Defined` rather than `Available`, it means the interfaces are not configured. This could be because there is a hardware problem with the adapter, or because an adapter, which was once in the machine, has been removed or become unseated in its slot. If any of the adapters are marked as `Defined`, the problem must be fixed before continuing.

To fix this kind of problem, you could enter `smit devices` and run the **Configure Devices Added After IPL** option. Also, the output from the `diag -a` command can help you to trace possible causes.

Subnet Issue in AIX 4.1

The AIX 4.1 operating system does not allow you to create more than one adapter on the same subnet using the SMIT mktcpip screen. AIX returns an error on the `ifconfig` of each adapter beyond the first on that subnet, because it detects that a route already exists for that subnet. Although all adapters remain up and usable, the extra adapters get marked as *down* in the ODM. In this state, when the system is rebooted, the adapter is not up. Since all standby adapters for a given network must be configured on the same subnet, this presents a major problem for HACMP in configurations where more than one standby adapter per network is desired on a node.

The `ifconfig` issued as a result of a takeover also gets this error returned, as can be seen by a bad return code in the /tmp/hacmp.out file, but in this case the ODM is not changed so no problem results.

5.2.1.1 Set Up Network Interfaces on Node 1

In this step, IP addresses and names are assigned to the service and standby network adapters on Node 1. Also, the /etc/hosts file is updated to include IP addresses and names for all network interfaces on Node 1 and Node 2.

When using SMIT to configure an adapter, the `HOSTNAME` field changes the default hostname. For example, if you configure the first adapter as `node1`, and then configure the second adapter as `node1_stby`, the default hostname at system boot time will be `node1_stby`. To avoid this problem, it is recommended that you configure the desired default hostname *last*. Therefore, it is preferable to configure all standby interfaces first, then the boot interfaces (a node will have a boot IP address associated with an adapter only if its service IP address can taken over by another system). The adapter IP addresses should be added to /etc/hosts automatically if you do your definitions using SMIT, but this should be checked as a final step.

1. Configure all standby interfaces first. Use the `smit mktcpip` command, select **tr1**, and fill out the menu as follows:

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* HOSTNAME                                             [node1_stby]
* Internet ADDRESS (dotted decimal)                    [9.3.4.16]
  Network MASK (dotted decimal)                        [255.255.255.0]
* Network INTERFACE                                     tr1
  NAMESERVER
           Internet ADDRESS (dotted decimal)           []
           DOMAIN Name                                 []
  Default GATEWAY Address                              []
          (dotted decimal or symbolic name)
  RING Speed                                           16
  START TCP/IP daemons Now                             no
```

2. Set up the boot IP address on `node1`, by entering `smit mktcpip`, selecting **tr0**, and filling out the menu as follows:

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* HOSTNAME                                             [node1_boot]
* Internet ADDRESS (dotted decimal)                    [9.3.1.3]
  Network MASK (dotted decimal)                        [255.255.255.0]
* Network INTERFACE                                    tr0
  NAMESERVER
           Internet ADDRESS (dotted decimal)           []
           DOMAIN Name                                 []
  Default GATEWAY Address                              []
          (dotted decimal or symbolic name)
  RING Speed                                           16
  START TCP/IP daemons Now                             no
```

3. Define node1's service IP address by entering the command `smit mkhostent` and filling out the menu:

```
                                Add a Host Name

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

                                                        [Entry Fields]
* INTERNET ADDRESS (dotted decimal)                    [9.3.1.16]
* HOST NAME                                            [node1]
  ALIAS(ES) (if any - separated by blank space)        []
  COMMENT (if any - for the host entry)                []









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

5.2.1.2 Set Up Network Interfaces on Node 2

1. Repeat Steps 1 on page 85 through 3 to configure all network adapters on node2.

   Node 2's standby adapter:

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* HOSTNAME                                             [node2_stby]
* Internet ADDRESS (dotted decimal)                    [9.3.4.17]
  Network MASK (dotted decimal)                        [255.255.255.0]
* Network INTERFACE                                    tr1
  NAMESERVER
           Internet ADDRESS (dotted decimal)           []
           DOMAIN Name                                 []
  Default GATEWAY Address                              []
          (dotted decimal or symbolic name)
  RING Speed                                           16
  START TCP/IP daemons Now                             no
```

   Node 2's boot IP address:

Note that if Node 2 were a Hot Standby node, there would be no boot IP address. You would configure the adapter's *service IP address* at this panel and omit the next step.

```
                        Minimum Configuration & Startup

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* HOSTNAME                                             [node2_boot]
* Internet ADDRESS (dotted decimal)                    [9.3.1.6]
  Network MASK (dotted decimal)                        [255.255.255.0]
* Network INTERFACE                                    tr0
  NAMESERVER
           Internet ADDRESS (dotted decimal)           []
           DOMAIN Name                                 []
  Default GATEWAY Address                              []
          (dotted decimal or symbolic name)
  RING Speed                                           16
  START TCP/IP daemons Now                             no
```

Node 2's service IP address:

```
                                Add a Host Name

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

                                                        [Entry Fields]
* INTERNET ADDRESS (dotted decimal)                    [9.3.1.17]
* HOST NAME                                            [node2]
  ALIAS(ES) (if any - separated by blank space)        []
  COMMENT (if any - for the host entry)                []

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

2. On each node, update the /etc/hosts file with any required entries to make the files identical. You can do this by editing the /etc/hosts file, or by using the `smit mkhostent` command in the same way that you created the service adapter's IP address.

   In summary, the /etc/hosts files on each node should look like this:

```
9.3.4.16        node1_stby
9.3.1.3         node1_boot
9.3.1.16        node1
9.3.4.17        node2_stby
9.3.1.6         node2_boot
9.3.1.17        node2
```

Note that the standby interface is on a different subnet from the service interface. This is necessary to verify that each adapter is indeed receiving and responding to the keepalive packets sent during normal cluster operations.

3. Certain network options should also be set for HACMP. The following lines should be added at the end of the /etc/rc.net file on each node:

```
/usr/sbin/no -o ipsendredirects=0
/usr/sbin/no -o ipforwarding=0
```

   HACMP uses the subnet feature of IP addressing to force IP datagrams to be sent across the designated network interface. To enable this feature, the `subnetsarelocal` option of the `/etc/no` command must be set to **0**. If `subnetsarelocal` is left at its default of 1, datagrams may not go to their intended destination, and adapter swapping may not work correctly. The `/usr/sbin/cluster/etc/rc.cluster` startup script sets the `subnetsarelocal` option to 0.

   Although HACMP does not strictly require that `ipforwarding` and `ipsendredirects` be enabled (set to **0**), it is a good idea to do so. This is why you add the lines shown above to the end of the /etc/rc.net file.

4. Use the `ping` command on both machines to check that all the interfaces are reachable.

5.2.1.3 Configuring an RS232 Serial Line

The *HACMP/6000 Installation Guide Version 3.1* strongly recommends that you make a direct serial connection between nodes that share a common resource. The serial network allows Cluster Managers to continue to exchange keepalive packets, even if the TCP/IP subsystem, or its networks, or network adapters fail. The serial network prevents nodes from becoming isolated and attempting to take over shared resources when they shouldn't.

The direct serial connection can be either the SCSI-2 Differential bus using target mode SCSI or a raw RS232 serial line. Experience shows that the raw RS232 serial line is a more robust connection method than target mode SCSI.

If the target mode SCSI connection goes down, a system reboot is required to restart it. In the case of RS232, a down connection is automatically restarted when the cables are correctly reconnected.

Also, if a SCSI adapter fails, node isolation could still occur, even though the cluster may be able to handle the failure by running an event script to fail over the disk resources to another node.. Therefore, we advocate the use of an RS232 connection whenever possible. If there is a unused serial port on each of the nodes, use it for this purpose.

This section describes how to configure an RS232 serial line to connect two cluster nodes sharing external disk resources.

1. Before configuring the RS232 serial line, you must have physically installed the line between the two nodes. The HACMP serial line, a null-modem line, is most easily implemented by connecting one of the following two orderable IBM features:
   - Feature 3124 (PN 88G4853) Serial to Serial Port Cable (3.7 meters)

- Feature 3125 (PN 88G4854) Serial to Serial Port Cable (8.0 meters)

Each of these features provides a null modem cable, to connect between serial ports on the your cluster nodes.

If it is not possible to obtain one of these cables, you can construct your own null modem cable for the serial line, as shown in Figure 30.

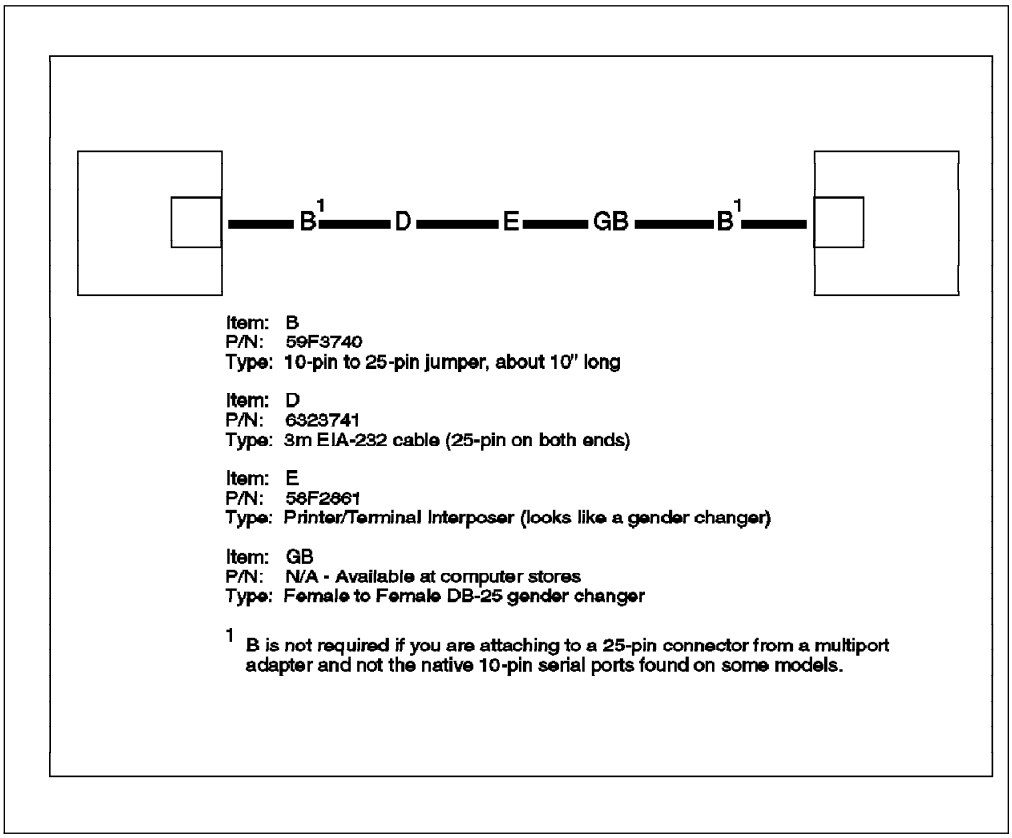


Figure 30. Alternative RS232 Serial Line Connection

2. Use the `smit tty` fastpath to create a tty device on Node 1 and Node 2. On the resulting panel, you can add an RS232 tty by selecting a native serial port, or a port on an asynchronous adapter, and filling out the fields as in the following example:

```
                                Add a TTY

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

[TOP]                                                 [Entry Fields]
  TTY type                                           tty
  TTY interface                                      rs232
  Description                                        Asynchronous Terminal
  Parent adapter                                     sa1
* PORT number                                        [s2]                  +
  BAUD rate                                          [9600]                +
  PARITY                                             [none]                +
  BITS per character                                 [8]                   +
  Number of STOP BITS                                [1]                   +
  XON-XOFF handshaking                               yes                   +
  RTS-CTS handshaking                                no                    +
  TERMINAL type                                      [dumb]
  STATE to be configured at boot time                [available]           +
  Read Trigger                                       [3]                   +#
[MORE...13]

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

Select the port number corresponding to the serial port you wish to use, for example, **s2**. Make sure that the `Enable LOGIN` field is set to **disable**. You do not want a `getty` process being spawned on this interface. When the command has finished, take note of the tty device number of the new tty (for example, tty0 or tty1).

3. Ensure you have connected the serial line between Node 1 and Node 2, then test the connection by doing the following:

   - On the first node, enter the following command:

     ```
     # stty < /dev/ttyx
     ```

     where `ttyx` is the newly added tty device. The command line on the first node should hang.

   - On the second node, enter the following command:

     ```
     # stty < /dev/ttyx
     ```

     where `ttyx` is the newly added tty device.

     If the nodes are able to communicate over the serial line, both nodes display their tty settings and return to the prompt.

     Note

     This is a valid communication test of a newly added serial connection before the HACMP clstrmgr daemon has been started. This test yields different results after the clstrmgr daemon has been started, because this daemon changes the initial settings of the tty devices and applies its own settings.

See the *HACMP/6000 Planning Guide Version 3.1*, Chapter 4. Planning Serial Networks, for more information on serial networks.

5.2.2 Installing Shared SCSI Disks

In a typical HACMP installation, two or more nodes will share disk resources. In some cases, all disk resources may be included in one resource group normally attached to one node and backed up by another (hot standby). In other environments, some disks may be included in one resource group and other disks in another. In this configuration, each resource group can be normally attached to a different node, with each node providing backup protection to the other (mutual takeover). Regardless of the mode of operation, the hardware setup for SCSI disks varies little. We have chosen SCSI disks for our example setup, since their setup is the most complex. Please refer to Appendix B, “Disk Setup in an HACMP Cluster” on page 209 for descriptions of how to set up other types of disks in a clustered environment.

Our example has disks owned by both Node 1 and Node 2. They are mirrored across separate SCSI buses to eliminate the I/O adapters and cables as single points of failure. See Figure 29 on page 82 for a picture of our configuration.

The following points summarize the steps required to set up SCSI-2 Differential adapters and disks:

- Remove terminating resistors from the SCSI-2 DE adapters.
- Install and configure the SCSI-2 DE adapters.
- Connect the disks between the nodes.

5.2.2.1 Removing Terminating Resistors from SCSI-2 DE Adapters

Any SCSI bus needs to be terminated at each end. In a conventional bus of SCSI devices, the bus is terminated at one end on the SCSI adapter, and at the other end on the last device on the bus. A shared SCSI bus in an HACMP environment also needs to terminated at each end. However, in this case, all the terminations are external. External terminations are used to enable a machine (adapter) to be disconnected from the bus without disrupting the other devices or machines connected to the same bus. To terminate each card externally, the terminating resistor blocks must be removed from the SCSI adapters. See Figure 49 on page 210 and Figure 50 on page 210 for illustrations of the locations of these terminating resistor blocks on the various SCSI differential adapters available from IBM.

5.2.2.2 Installing and Configuring SCSI2 DE Adapters

After you have finished removing the terminating resistor blocks from the SCSI adapters, install them into your cluster nodes. To avoid confusion, it is good practice to install respective SCSI adapters in the same slot numbers in both nodes.

Next time you boot up your cluster nodes, the SCSI adapters should be automatically configured. List the installed SCSI adapters using the following command:

```
# lscfg | grep scsi
```

The output should look like:

```
name  status    location description

scsi0 Available 00-08    SCSI I/O Controller
scsi1 Available 00-06    SCSI I/O Controller
scsi2 Available 00-05    SCSI I/O Controller
```

The internal disks are usually connected to `scsi0`, which resides in slot 8 on a 5XX model. One shared disk cabinet will be connected to `scsi1`, which resides in slot 5, and the other will be connected to `scsi2` in slot 6. Both adapters must be in the `Available` state.

Now we want to find the SCSI IDs for each of the adapters. To find the SCSI ID of the `scsi1` adapter, enter the command:

```
# lsattr -E -l scsi1 -a id
```

The output should look like:

```
id 7 Adapter card SCSI ID
```

Then do the same for `scsi2`. The SCSI IDs for both adapters should be `7`, which is the default value.

Each device or adapter on a SCSI bus needs to have a unique SCSI ID. This means that, since each bus will be connected to our two machines, each of the disks on the bus, as well as the adapters at either end, must have unique SCSI IDs. The SCSI adapters on each machine need not have unique IDs from each other, but rather each device on a single shared bus must be unique. It is probably a good practice to use the same SCSI ID for each shared bus SCSI adapter on a node, in order to more easily manage your system.

Because, at this point, the SCSI IDs for the corresponding adapters on Node 2 will also have the default value of `7`, we must change them to something else, to make them unique on their buses.

On a SCSI bus, only one device can have control of the bus at a time. In case of contention for control of the bus, the device with the highest SCSI ID wins control. Therefore, you always want your SCSI adapters to have the highest SCSI IDs on the bus. This is why SCSI adapters, in AIX 3.2.5, have default ids of `7`, the highest possible address. For the other SCSI adapters on the bus, we will want to use the next highest addresses, starting with `6`.

Administration Tip

Some implementers of HACMP do not like to use the SCSI ID 7 on any of their shared buses. This is because, if you have one system with SCSI IDs of 7 up and running, and then do maintenance on another system attached to the bus, you may inadvertently cause that second system to come up with SCSI IDs of 7 also.

Reinstallation of the system, some `mksysb` restorations, replacement of adapters, and booting a system model that does not support the SCSI-2 Differential F/W adapter for boot disks but has F/W for the shared bus, are all examples of situations that could cause another system to also come up with conflicting SCSI IDs of 7.

This is certainly not a rule, but is worth keeping in mind.

Set the SCSI IDs for both SCSI adapters in Node 2 to a value of `6` using SMIT. On Node 2, enter the command `smit chgscsi`, and change the SCSI ID for adapter `scsi1` as follows:

```
                Change/Show Characteristics of a SCSI Adapter

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

                                                        [Entry Fields]
  SCSI adapter                                         scsi1
  Description                                          SCSI I/O Controller
  Status                                               Available
  Location                                             00-05
  Adapter card SCSI ID                                [6]
  BATTERY backed adapter                               no
  DMA bus memory LENGTH                               [0x202000]
  Enable TARGET MODE interface                         no
  Target Mode interface enabled                        no
  PERCENTAGE of bus memory DMA area for target mode   [50]
  Name of adapter code download file                   /etc/microcode/8d77.44>
  Apply change to DATABASE only                        no
```

If you have not yet attached any disks to the shared buses, you should be able to leave the last field, `Apply change to DATABASE only`, set to **no** for an immediate change to be made. If you have already attached disk devices to the shared bus, this will not be possible. In this case, you will have to change this field to **yes**, and reboot the machine, for the change to become effective.

Repeat this change for the other SCSI adapter, `scsi2`.

Check the SCSI IDs of the SCSI adapters on Node 2 to ensure that they are now set to `6`. Enter the following command to find the SCSI id of the adapters named `scsi1` and `scsi2`:

```
# lsattr -E -l scsi1 -a id
id 6 Adapter card SCSI ID True
# lsattr -E -l scsi2 -a id
id 6 Adapter card SCSI ID True
```

If the SCSI IDs are not set to 6, then you should enter the command `smit chgscsi`, as shown above, and change the values to 6.

5.2.2.3 Connecting Shared SCSI Disks Between Nodes

Now that the SCSI adapters that will be connected to the shared bus have unique SCSI IDs, you can connect the shared disk devices. It is also possible to do this before configuring any of the SCSI adapters. However, if the disks have been powered on and defined by AIX, then you will have difficulty changing the characteristics (SCSI IDs) of the adapters. You will receive a `device busy` error because disk devices have been defined to the adapter. In this case, you would need to make the change of characteristics to the database only, and then reboot the machine to make the change effective.

Follow the steps below to connect and configure the shared disk resources.

1. Shut down and power off both machines.
2. Connect the disk cabinets between Node 1 and Node 2 using the required cables. Our example uses 9334-501 Expansion Units. You can see an example of the cabling requirements for these devices in Figure 54 on page 216.
3. Given that the SCSI IDs of the adapters on each bus are 6 and 7, each disk must now have a unique SCSI ID between 0 and 5. Also, take note of the maximum cable length (see Figure 54 on page 216) for the SCSI-2 Differential Ended bus. The maximum total cable length, including cables between devices and cables inside the disk expansion units, must not exceed 19 meters.

   Now, power on the disks first, and then power on Node 1. After Node 1 has booted, you can power up Node 2. The reason for staggering the boot times is to avoid having the two systems trying to configure the shared disks at the same time. This would happen if they both reached the same point in their boot process simultaneously.
4. On Node 1, use the `lsdev` command to list the disks:

   ```
   # lsdev -Cc disk -H
   ```

   ```
   name   status    location    description

   hdisk0 Available 00-08-00-00 2.0 GB SCSI Disk Drive
   hdisk2 Available 00-06-00-00 1.0 GB SCSI Disk Drive
   hdisk3 Available 00-06-00-10 1.0 GB SCSI Disk Drive
   hdisk5 Available 00-05-00-00 1.0 GB SCSI Disk Drive
   hdisk6 Available 00-05-00-10 1.0 GB SCSI Disk Drive
   ```

   Notice that the shared disks are `hdisk2, hdisk3, hdisk5`, and `hdisk6`.
5. Now list the disk devices on Node 2:

```
# lsdev -Cc disk -H
```

```
name   status    location    description

hdisk0 Available 00-08-00-00 2.0 GB SCSI Disk Drive
hdisk2 Available 00-06-00-00 1.0 GB SCSI Disk Drive
hdisk3 Available 00-06-00-10 1.0 GB SCSI Disk Drive
hdisk5 Available 00-05-00-00 1.0 GB SCSI Disk Drive
hdisk6 Available 00-05-00-10 1.0 GB SCSI Disk Drive
```

For simplicity, both nodes have the same disk configurations. This, of course, does not have to be the case. If nodes have different quantities of internal disks installed, then the names on different nodes for the same shared disk may be different. Having the same name on each node for a disk aids clarity, but this is not always possible.

You should now update the Shared SCSI-2 Differential Disk Worksheet (Appendix A, Page 11 of *HACMP/6000 Planning Guide Version 3.1*).

At this point, your shared disk installation is complete and you can now define the shared LVM components.

5.2.3 Defining Shared LVM Components

Now you can create the shared volume groups and filesystems that will reside on the shared disk devices. Our configuration will have two volume groups. Volume group havg1 will be owned by Node 1, and volume group havg2 will be owned by Node 2.

The volume group havg1 contains two disks, hdisk2 in one 9334 cabinet and hdisk5 in the second 9334 cabinet. We shall mirror hdisk2 on hdisk5. The volume group havg2 also contains two disks, hdisk3 in the same cabinet as hdisk2 and hdisk6 in the same cabinet as hdisk5. We shall mirror hdisk3 on hdisk6.

Creating the volume groups, logical volumes, and file systems shared by the nodes in an HACMP/6000 cluster requires that you perform steps on all nodes in the cluster. In general, you first define all the components on one node (in our example, this is Node 1) and then import the volume groups on the other nodes in the cluster (in our example, this is Node 2). This ensures that the ODM definitions of the shared components are the same on all nodes in the cluster.

Non-concurrent access environments typically use journaled file systems to manage data, while concurrent access environments use raw logical volumes. Our example deals with non-concurrent access environments only.

Figure 31 on page 96 lists the steps you complete to define the shared LVM components for non-concurrent access environments using SCSI disk subsystems.

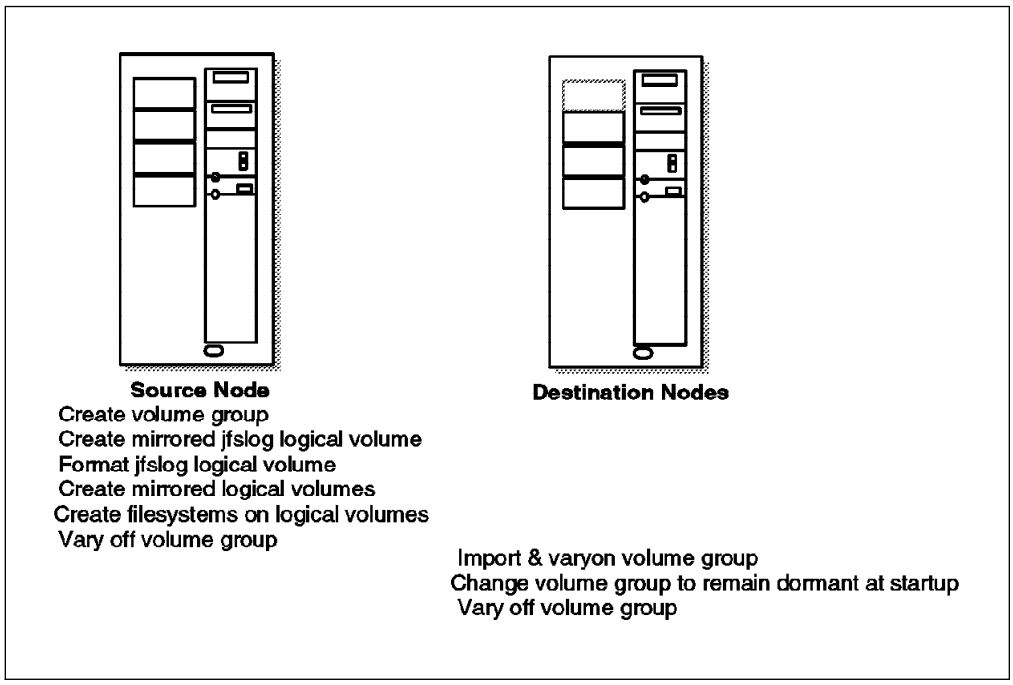


Figure 31. Defining Shared LVM Components for Non-Concurrent Access

5.2.3.1 Create Shared Volume Groups on Node 1

Use the `smit mkvg` fastpath to create a shared volume group.

1. As root user on Node 1 (the source node), enter `smit mkvg`:

```
                              Add a Volume Group

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

                                                      [Entry Fields]
  VOLUME GROUP name                                  [havg1]
  Physical partition SIZE in megabytes                4                      +
* PHYSICAL VOLUME names                              [hdisk2 hdisk5]        +
  Activate volume group AUTOMATICALLY                 no                     +
    at system restart?
* ACTIVATE volume group after it is                  yes                    +
    created?
  Volume Group MAJOR NUMBER                          [44]                   +#

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

Here, you provide the name of the new volume group, the disk devices to be included, and the major number to be assigned to it. It is also important to specify that you do not want the volume group activated (varied on) automatically at system restart, by changing the setting of that field to **no**. The varyon of shared volume groups needs to be under the control of HACMP, so that it is coordinated correctly.

Regardless of whether you intend to use NFS or not, it is good practice to specify a major number of the volume group. To do this, you must select a major number that is free on each node. Be sure to use the same major number on all nodes. Use the `lvlstmajor` command on each node to determine a free major number common to all nodes.

2. Because `havg1` and `havg2` contain mirrored disks, you can turn off quorum checking. On the command line, enter `smit chvg` and set quorum checking to **no**

```
                              Change a Volume Group

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

                                                      [Entry Fields]
* VOLUME GROUP name                                   havg1
* Activate volume group AUTOMATICALLY                 no                      +
    at system restart?
* A QUORUM of disks required to keep the volume       no                      +
   group on-line ?




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

   Now repeat the two steps above for volume group `havg2`.

3. Varyon the two volume groups on Node 1:

```
# varyonvg havg1
# varyonvg havg2
```

4. Before you create any filesystems on the shared disk resources, you need to explicitly create the *jfslog logical volume*. You need to explicitly create the jfslog logical volume, so that you can give it a unique name of your own choosing, which is used on all nodes in the cluster to refer to the same log. If you do not do this, it is possible and likely that naming conflicts will arise between nodes in the cluster, depending on what user filesystems have already been created.

   Use SMIT to add the log logical volumes `sharedloglv1` for the filesystems in volume group `havg1`, and `sharedloglv2` for the filesystems in volume group `havg2`. Enter `smit mklv`, and select the volume group **havg1** to which you are adding the first new jfslog logical volume.

```
                              Add a Logical Volume

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

[TOP]                                                 [Entry Fields]
  Logical volume NAME                                [sharedloglv1]
* VOLUME GROUP name                                   havg1
* Number of LOGICAL PARTITIONS                       [1]                    #
  PHYSICAL VOLUME names                              [hdisk2 hdisk5]        +
  Logical volume TYPE                                [jfslog]
  POSITION on physical volume                         midway                +
  RANGE of physical volumes                           minimum               +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                 []                     #
    to use for allocation
  Number of COPIES of each logical                    2                     +
    partition
  Mirror Write Consistency?                           yes                   +
  Allocate each logical partition copy                yes                   +
    on a SEPARATE physical volume?
[MORE...9]

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

The fields that you need to change or add to are shown in **bold** type.

After you have created the jfslog logical volume, be sure to format the log logical volume with the following command:

```
# /usr/sbin/logform /dev/sharedloglv1
logform: destroy /dev/sharedloglv1 (y)?
```

Answer yes (`y`) to the prompt about whether to destroy the old version of the log.

Now create the log logical volume `sharedloglv2` for volume group `havg2` and format the log, using the same procedure.

5. Now use SMIT to add the logical volumes `sharedlv1` and `sharedlv2`.

   It would be possible to create the filesystems directly, which would save some time. However, it is recommended to define the logical volume first, and then to add the filesystem on it. This procedure allows you set up mirroring and logical volume placement policy for performance. It also means you can give the logical volume a unique name.

   On Node 1, enter `smit mklv`, and select the volume group **havg1**, to which you will be adding the new logical volume.

```
                              Add a Logical Volume

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

[TOP]                                                 [Entry Fields]
  Logical volume NAME                                [sharedlv1]
* VOLUME GROUP name                                   havg1
* Number of LOGICAL PARTITIONS                       [50]                    #
  PHYSICAL VOLUME names                              [hdisk2 hdisk5]         +
  Logical volume TYPE                                []
  POSITION on physical volume                         center                 +
  RANGE of physical volumes                           minimum                +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                 []                      #
    to use for allocation
  Number of COPIES of each logical                    2                      +
    partition
  Mirror Write Consistency?                           yes                    +
  Allocate each logical partition copy                yes                    +
    on a SEPARATE physical volume?
  RELOCATE the logical volume during                  yes                    +
    reorganization?
  Logical volume LABEL                               []
  MAXIMUM NUMBER of LOGICAL PARTITIONS               [128]
  Enable BAD BLOCK relocation?                        yes                    +
  SCHEDULING POLICY for writing logical               sequential             +
    partition copies
  Enable WRITE VERIFY?                                no                     +
  File containing ALLOCATION MAP                     []
[BOTTOM]

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

The **bold** type illustrates those fields that need to have data entered or modified. Notice that SCHEDULING POLICY has been set to **sequential**. This is the best policy to use for high availability, since it forces one mirrored write to complete before the other may start. In your own setup, you may elect to leave this option set to the default value of parallel to maximize disk write performance.

6. Now, create the filesystems on the logical volumes you have just defined. At the command line, you can enter the following fastpath: smit crjfslv. Our first filesystem is configured on the following panel:

```
       Add a Journaled File System on a Previously Defined Logical Volume


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

                                                      [Entry Fields]
* LOGICAL VOLUME name                                 sharedlv1              +
* MOUNT POINT                                        [/sharedfs1]
  Mount AUTOMATICALLY at system restart?              no                     +
  PERMISSIONS                                         read/write             +
  Mount OPTIONS                                      []                      +
  Start Disk Accounting?                              no                     +




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

Repeat the above step to create the filesystem /sharedfs2 on logical volume sharedlv2.

7. Mount the filesystems to check that creation has been successful.

```
# mount /sharedfs1
# mount /sharedfs2
```

8. If there are problems mounting the filesystems, there are two suggested actions to resolve them:
   a. Execute the fsck command on the filesystem.
   b. Edit the /etc/filesystems file, check the stanza for the filesystem, and make sure it is using the new jfslog you have created for that volume group. Also, make sure that jfslog has been formatted correctly with the logform command.

   Assuming that the filesystems mounted without problems, now unmount them.

```
# umount /sharedfs1
# umount /sharedfs2
```

9. Vary off the two volume groups.

```
# varyoffvg havg1
# varyoffvg havg2
```

5.2.3.2 Import Shared Volume Groups to Node 2

The next step is to import the volume groups you have just created to Node 2. Login to Node 2 as root and do the following steps:

1. Enter the fastpath command: smit importvg and fill out the fields as shown:

```
                          Import a Volume Group

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

                                                    [Entry Fields]
  VOLUME GROUP name                                [havg1]
* PHYSICAL VOLUME name                             [hdisk2]          +
* ACTIVATE volume group after it is                 yes                    +
    imported?
  Volume Group MAJOR NUMBER                        [44]                   +#




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

2. Change the volume group to prevent automatic activation of havg1 at system restart and to turn off quorum checking. This must be done each time you import a volume group, since these options will reset to their defaults on each import. Enter smit chvg:

```
                              Change a Volume Group

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

                                                      [Entry Fields]
* VOLUME GROUP name                                   havg1
* Activate volume group AUTOMATICALLY                 no                      +
    at system restart?
* A QUORUM of disks required to keep the volume       no                      +
   group on-line ?




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

3. Repeat the two steps above for volume group `havg2`, being sure to specify its major number on the import panel.
4. Vary on the volume groups and mount the filesystems on Node 2 to ensure that there are no problems.

5.2.4 Additional Tasks

After setting up the networks and shared disks for your cluster, there are some other tasks you need to complete before you can start installing and configuring HACMP. These tasks are covered in the following sections.

5.2.4.1 Setting Network Options

This step involves editing the /etc/rc.net file on each cluster node to add the following lines:

```
if [ -f /usr/sbin/no ] ; then
   .
   .
   .
   /usr/sbin/no -o ipsendredirects=0
   /usr/sbin/no -o ipforwarding=0
fi
```

5.2.4.2 Editing the /etc/hosts File and Nameserver Configuration

Make sure that the service, standby, and boot addresses of all cluster nodes are listed in the /etc/hosts file on each cluster node. In case you are using DNS (Domain Name Serving), it is also a good idea to also add the fully qualified name (<IP Label>.<Domain Name>) as an alias for the service and boot IP labels.

Ensure that the /etc/hosts file on each cluster node has the following entry:

```
127.0.0.1   loopback localhost
```

Make sure that at least the service and boot addresses of your cluster nodes are defined in your nameserver configuration.

5.2.4.3 Editing the /.rhosts File

Make sure that the IP labels and fully qualified names corresponding to all service, standby, and boot addresses of all TCP/IP networks in the cluster are present in the /.rhosts file of each cluster node.

For our example, the /.rhosts file has the following entries:

```
node1_stby
node1_stby.austin.ibm.com
node1_boot
node1_boot.austin.ibm.com
node1
node1.austin.ibm.com
node2_stby
node2_stby.austin.ibm.com
node2_boot
node2_boot.austin.ibm.com
node2
node2.austin.ibm.com
```

You must also ensure that the /.rhosts has the correct permissions. It must be owned by root, with read/write access for the owner, and no permissions for group or others, as shown in the following listing:

```
# ls -l /.rhosts
-rw-------   1 root     system       461 May 26 18:05 /.rhosts
```

This file ensures that the cluster configuration and node environment configuration synchronization functions, and the Global ODM feature of HACMP work correctly between nodes.

Having successfully completed the steps in the previous sections, unmount your shared filesystems and varyoff the shared volume groups. You are now in a position to install and configure your HACMP cluster.

5.3 Setting Up a New HACMP Cluster

This section describes installing the HACMP/6000 Version 3.1 software on cluster nodes and clients. It contains instructions for installing HACMP from scratch.

Note

The timing of the HACMP 4.1 for AIX announcement made it impossible to include a description of its installation in this book. However, while HACMP 4.1 for AIX is an SMP enabled version, its installation differs very little from that of HACMP/6000 Version 3.1.

If you have a previous version of HACMP, but are not going to save your existing cluster configuration, you should follow the instructions in this section.

If you wish to save your existing cluster configuration and upgrade to HACMP/6000 Version 3.1, see Section 5.4, “Upgrading a Cluster to HACMP/6000 Version 3.1” on page 132 for upgrade and conversion instructions.

5.3.1 Installation Prerequisites

The following prerequisites apply to the installation of HACMP/6000 Version 3.1:

1. Each node must have AIX Version 3.2 installed.
2. Each node in the cluster requires its own HACMP license.
3. You must be root user to perform the installation.
4. The /usr filesystem must have 15 MB of free disk space for nodes in a non-concurrent access environment. 18 MB is required on each node in a concurrent access environment. Concurrent access is not covered in this book. If you are installing the client portion of HACMP on client systems, then /usr requires only eight MB of free disk space.

5.3.2 Installation Options

You can install HACMP from tape, or load an installation image from tape onto one node and use it as an installation server to install the other nodes. Because HACMP is fairly small, as an alternative, you could transfer copies of the installation image with ftp to the install directory (/usr/sys/inst.images) of the other nodes and install HACMP from there. Installing each node in parallel will be much quicker than installing from tape, especially if you have more than two or three nodes to install.

5.3.3 Installing the HACMP Software on Node 1 and Node 2

In our example, we will install HACMP from tape. Complete the following steps to install the HACMP/6000 Version 3.1 software on a node. You should note that the client subsystem (clinfo) is automatically installed on a node when the `Automatically install PREREQUISITE software?` field is set to **yes**.

1. Put the HACMP install tape into the tape drive of Node 1 and enter:

```
# smit install_selectable_all
```

2. Then enter the name of the input device or select it from the list presented by pressing F4. For example, your input device could be a tape drive called **/dev/rmt0.1**.
3. Press Enter, and accept the default values offered in the following SMIT panel:

```
                Install From All Available Software Packages

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

                                                      [Entry Fields]
* INPUT device / directory for software               /dev/rmt0.1
* SOFTWARE to install                                 [all]                  +
  Automatically install PREREQUISITE software?        yes                   +
  COMMIT software?                                    no                    +
  SAVE replaced files?                                yes                   +
  VERIFY Software?                                    no                    +
  EXTEND file systems if space needed?                yes                   +
  REMOVE input file after installation?               no                    +
  OVERWRITE existing version?                         no                    +
  ALTERNATE save directory                            []



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

4. Press Enter to start the installation.

   You will see a number of messages as the installation proceeds. When the installation has completed, you are instructed to read the README3.1 file located in the /usr/sbin/cluster directory. It is important to read this file in case there are further instructions relating to tasks that may need to be performed after installation of the software.

Repeat this installation procedure on Node 2.

5.3.4 Installing the HACMP Client Portion on Client Systems

In most cases, your users will be connecting to the cluster nodes over a TCP/IP LAN, from a client system. In most cases, these client systems are separate UNIX systems or PCs, but in some cases, they can also be terminal servers. If the client system is an AIX system, it is possible to install the client portion of HACMP on it. This client portion is called *clinfo*, or Cluster Information Services, and it provides several functions.

First, it provides information about the cluster, through the clinfo API, which can be used to build *cluster-aware* applications. These are applications that, from a client machine, are able to query information about the state of the cluster. This information can also be obtained through a clinfo application called *clstat*. This application is provided with HACMP, and provides, in either graphical or ascii form, information about the state of the cluster, its nodes, and network adapters.

Secondly, clinfo can be used to trigger automatic actions on the client system whenever a failure event happens in the cluster. The clinfo subsystem provides a shell script, `/usr/sbin/cluster/clinfo.rc`, which by default, flushes the arp cache of the system it is running on whenever a failure occurs, but can be customized to do whatever else is desired.

The installation of the clinfo subsystem on a client system is optional. If you will be implementing the takeover of hardware addresses along with IP addresses in your cluster setup, the arp cache flushing function of clinfo is not needed. In this case,

you would only need to install clinfo if you want to use the clstat cluster status application, or if you want to use the clinfo API in your application, to make it cluster-aware.

Note: The source code for clinfo is included with the HACMP product, to allow you to port it to other client machines, other than RS/6000s.

If you wish to install the client portion of the high availability software on a workstation, complete the following steps:

1. Put the HACMP install tape into the tape drive and enter:

```
# smit install_selectable_all
```

2. Enter the name of the input device or select it from the list presented by pressing F4. For example, your input device may be a tape drive called **/dev/rmt0.1**.
3. Press Enter. Make sure that the cursor is located on the field called `SOFTWARE to install` and press F4. Then, from the list of available packages, use F7 to select **cluster.client**.
4. You are presented with the following panel:

```
                Install From All Available Software Packages

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

                                                      [Entry Fields]
* INPUT device / directory for software               /dev/rmt0.1
* SOFTWARE to install                                [3.1.0.0  cluster.clien> +
  Automatically install PREREQUISITE software?       yes                      +
  COMMIT software?                                   no                       +
  SAVE replaced files?                               yes                      +
  VERIFY Software?                                   no                       +
  EXTEND file systems if space needed?               yes                      +
  REMOVE input file after installation?              no                       +
  OVERWRITE existing version?                        no                       +
  ALTERNATE save directory                           []

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

5. Press Enter to start the installation.

   You will see a number of messages as the installation proceeds. When the installation has completed, you are instructed to read the README3.1 file located in the /usr/sbin/cluster directory. It is important to read this file in case there are further instructions relating to tasks that may need to be performed after installation of the client portion.

   One task that you must perform on a client system is to update the /usr/sbin/cluster/etc/clhosts file. In this file, you must list the IP label or IP address of each of the boot and service adapters on all nodes of the cluster. This is required so that the clinfo subsystem on the client system can connect to one of the nodes in the cluster to receive information about the cluster state.

5.3.5 Rebooting Nodes and Clients

The final step in the installation process is to reboot each node and client in your cluster environment.

5.3.6 Verifying the Cluster Software

HACMP provides a utility for you to verify that there are no problems with the HACMP software or your configuration at several steps in the process. This utility can be invoked using the `/usr/sbin/cluster/diag/clverify` command.

The `clverify` utility is most often run in an interactive mode, in which you can step through several menus to reach the particular verification option you want to run.

We will now execute various options of the `clverify` utility on our cluster, to confirm that there are no problems so far. As we go along, we will describe each option and the output produced by executing it.

On executing the `/usr/sbin/cluster/diag/clverify` command, the following screen appears:

```
------------------------------------------------------------
To obtain help on a specific option, type: help <option>
To return to previous menu, type: back
To quit the program, type: quit
------------------------------------------------------------

Valid options:
software
cluster

clverify>
```

The **software** option notifies you if any of the following problems exist on your cluster:

- An incompatible or unsupported level of AIX, SNMP, or TCP/IP is installed.
- The LPP installation itself is incorrect.
- A required PTF is missing.
- A PTF incompatible with the HACMP software has been installed.

The **cluster** option checks for the following:

- All resources used by HACMP are validly configured.
- Ownership and takeover of these resources are defined and are in agreement across nodes.

Choosing the **software** option gives the following options:

```
Valid options:
prereq
bos
badptfs
lpp
clverify.software>
```

The functions of these options are as follows:

- The **prereq** option verifies that all known AIX PTFs required for HACMP are installed.
- The **bos** option verifies that the installed version and release number of the system's Base Operating System and the TCP/IP and SNMP subsystems are compatible with HACMP and prints a warning message if the installed version does not meet the established requirements.
- The **badptfs** option verifies that no PTFs known to be incompatible with HACMP are installed.
- The **lpp** option verifies that all the HACMP files are properly installed.

On entering **prereq** at the `clverify.software` prompt, we get:

```
Retrieving installed APARs from system...
Comparing installed APARs with requirements listed in clvreq.dat...
The fix for APAR IX46408 is not installed on your system.
IX46408
Exit Code: 1
Command completed.

      --------- Hit Return To Continue ---------
```

The `prereq` verify reveals that our installation requires the PTF created for APAR IX46508. You should install a required PTF as soon as possible. Do not allow an inconsistent software environment to persist.

The **bos** option gives the following output:

```
clverify.software> bos
Comparing installed software with requirements listed in clvbos.dat...
All Base Operating System software requirements met.
Command completed.

      --------- Hit Return To Continue ---------
```

If, for example, the required version of TCP/IP were not installed, we would have received a message such as the following:

```
ckprereq:  The following software products must be applied first:
 (as defined by "/usr/sbin/cluster/diag/clvbos.dat", the failing products
requisite file)

 * bosnet.tcpip.obj v>2, r=2 or r>2
 * prereq
 * prereq
 * prereq
```

On entering **badptfs**, we get the following message:

```
clverify.software> badptfs
Comparing installed PTFs with incompatible PTFs listed in clvinval.dat.
No installed PTFs known to be incompatible with HACMP.
Command completed.

      --------- Hit Return To Continue ---------
```

If an unwanted PTF were present, a message indicating its number would appear.

```
PTF Uxxxxxx is not compatible with HACMP.
```

The **lpp** option gives the following output:

```
clverify.software> lpp
The files for package cluster are being verified.
This may take several minutes, please wait.
The files for package cluster are being verified.
This may take several minutes, please wait.

Checking AIX files for HACMP-specific modifications...
Warning: There is no cluster found.
cllsif: Error reading configuration.

*/etc/inittab not configured for HACMP.
        If IP Address Takeover is configured,
        or the Cluster Manager is to be started on boot,
        then /etc/inittab must contain the proper HACMP entries.

Command completed.

      --------- Hit Return To Continue ---------
```

If the HACMP software were not installed correctly, a message would appear informing us of the error. However, you will notice here that AIX files are checked for HACMP-specific modifications. The errors displayed are to be expected because a cluster has not yet been defined.

After verifying the HACMP software, enter **back** at the `clverify.software` prompt then enter **quit** at prompt to exit `clverify`.

5.3.7 Defining the Cluster Environment

Now that the HACMP software has been successfully installed, it is time to define the cluster itself. In HACMP terms, this is called defining the cluster environment.

The cluster environment is defined by entering information describing the following components:

- Cluster
- Nodes in the cluster
- Network adapters

Again, SMIT provides the interface to make the tasks straightforward. The methods used for defining the cluster differ from those in HACMP Version 2.1 and provide a greater level of flexibility and refinement.

HACMP/6000 Version 3.1 introduces a new entity called a *Network Interface Module (NIM)*. NIMs are preloaded at install time and allow you the flexibility to change the characteristics of a particular network type, such as the keepalive or heartbeat rate. This capability allows you to fine-tune the behavior of the cluster, and is most often used to prevent false takeovers from occurring in an environment where occasional peaks in CPU or network load are present.

5.3.7.1 Defining the Cluster ID and Name

The first step is to create a cluster ID and name that uniquely identifies the cluster. This is necessary in case there is more than one cluster on a single physical network. Refer to your completed planning worksheets the data and complete the following steps to define the cluster ID and name.

1. Enter the `smit hacmp` command to display the system management menu for HACMP: The HACMP menu is the starting point for the definition and management of all HACMP characteristics and function.

```
                          HACMP/6000

Move cursor to desired item and press Enter.

  Manage Cluster Environment
  Manage Application Servers
  Manage Node Environment
  Show Environment
  Verify Environment
  Manage Cluster Services
  Cluster Recovery Aids
  Cluster RAS Support




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

2. Select **Manage Cluster Environment** and press Enter to display the following menu:

```
                    Manage Cluster Environment

Move cursor to desired item and press Enter.

  Configure Cluster
  Configure Nodes
  Configure Adapters
  Synchronize All Cluster Nodes
  Show Cluster Environment
  Configure Network Modules

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

3. Select **Configure Cluster** and press Enter to display the following menu:

```
                              Configure Cluster

Move cursor to desired item and press Enter.

  Add a Cluster Definition
  Change / Show Cluster Definition
  Remove Cluster Definition

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

4. Choose the **Add a Cluster Definition** option and press Enter to display the following panel. Our examples are shown, however, you will want to fill out the fields with the cluster ID and name of your choice:

```
                         Add a Cluster Definition

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

                                                    [Entry Fields]
  **NOTE: Cluster Manager MUST BE RESTARTED
          in order for changes to be acknowledged.**

* Cluster ID                                       [1]                    #
* Cluster Name                                     [itso_austin]

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

5. Press Enter. The cluster ID and name are entered in HACMP's own configuration database managed by the ODM.

6. Press F3 to return to the Manage Cluster Environment screen. From here, we will move to the next stage, defining the cluster nodes.

5.3.7.2 Defining Nodes

Other parts of the cluster definition refer to the cluster nodes by their node names. In this section, we are simply defining the names that will identify each node in the cluster.

1. Select **Configure Nodes** on the Manage Cluster Environment screen to display the following menu:

```
                                  Configure Nodes

Move cursor to desired item and press Enter.

  Add Cluster Nodes
  Change / Show Cluster Node Name
  Remove a Cluster Node




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

2. Choose the **Add Cluster Nodes** option and press Enter to display the following screen. Our examples are shown, however, you will want to fill out the fields with the node names of your choice:

```
                              Add Cluster Nodes

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

                                                      [Entry Fields]
* Node Names                                         [Node1 Node2]







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

   Remember to leave a space between names. If you use a duplicate name, an error message will be displayed. You need only to enter this information on one node, because you can later execute **Synchronize All Cluster Nodes** to propagate the information, using HACMP's Global ODM (GODM), to all other nodes configured in the cluster.

3. Press Enter to update HACMP's configuration database.
4. Press F3 to return to the `Manage Cluster Environment` screen. From here, we will move to the next stage, defining the network adapters to HACMP.

5.3.7.3 Defining Network Adapters

Having defined the node names, you can now proceed with defining the network adapters associated with each node. Again, you can define all the network adapters for all nodes on one node. You can later synchronize all the information to the other nodes' ODMs.

We shall use the values for our sample cluster You should refer to the planning worksheets for TCP/IP and serial networks for your own cluster definitions. If you refer to Figure 29 on page 82, you will notice that both Node1 and Node 2 contain two token ring network adapters. One adapter is configured as a service adapter and the other is configured as a standby adapter. If the service adapter in one node fails, its standby adapter will be reconfigured by the Cluster Manager to take over that service adapter's IP address. If a node fails, the standby adapter in the surviving node will be reconfigured to take over the failed node's service IP address and masquerade as the failed node.

Notice also the RS232 connection between Node 1 and Node 2. The RS232 link provides an additional path for keepalive (or heartbeat) packets and allows the Cluster Managers to continue communicating if the network fails. It is important to understand also that the RS232 network is not a TCP/IP network. Instead it uses HACMP's own protocol over the raw RS232 link.

Having this non-TCP/IP RS232 network is a very important requirement, since it provides us protection against two single points of failure:

1. The failure of the TCP/IP software subsystem
2. The failure of the single token-ring network

In either of these cases, if the RS232 network were not there, all keepalive traffic from node to node would stop, even though the nodes were still up and running. This is known as *node isolation*. If node isolation were to occur, Node 1 and Node 2 would both attempt to acquire their respective takeover resources. However, since the partner nodes would still be up and running, these attempts would fail, with the respective Cluster Managers endlessly attempting to reconfigure the cluster.

With the RS232 link in place, either of these failures would be interpreted as a network failure, instead of a node failure, allowing the administrator to take the appropriate action (restarting TCP/IP on a node, or fixing a network problem), without the cluster nodes trying to take over each other's resources inappropriately.

Defining Node 1's Network Adapters: Complete the following steps to define Node 1's network adapters:

1. Select **Configure Adapters** on the `Manage Cluster Environments` panel to display the following menu:

```
                              Configure Adapters

Move cursor to desired item and press Enter.

  Add an Adapter
  Change / Show an Adapter
  Remove an Adapter




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

2. Choose the **Add an Adapter** option. Press Enter to display the following panel, where you will fill out the fields for the service adapter:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node1]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    service               +
  Adapter Identifier                                 [9.3.1.16]
  Adapter Hardware Address                           [0x42005a4f4165]
  Node Name                                          [Node1]                +




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

3. Press Enter to store the details in HACMP's configuration database.

   The following observations can be made about the fields to be filled in on this panel:

   Adapter Label — This is the IP label of the adapter, which should be the same as the label you have defined in the /etc/hosts file and in your nameserver.

   Network Type — If you list this field with F4, you will see the various Network Interface Modules (NIMs) available. There is a NIM for each type of network medium supported, as well as a Generic IP NIM. Since this adapter is on a token-ring network, we have selected the **token** NIM.

   Network Name — This is an arbitrary name of your own choosing, to define to HACMP which of its adapters are on the same physical network. It is important that you use the same network name for all of the adapters on a physical network.

Network Attribute This field can either be set to public, private, or serial. A *public network* is one that is used by cluster nodes and client systems for access, as is this token-ring network. A *private network* is used for communications between cluster nodes only. The Cluster Lock Manager uses any private networks that are defined for its first choice to communicate between nodes. The most common reason to define a network as private is to reserve it for the exclusive use of the Cluster Lock Manager. A *serial network* is a non-TCP/IP network. This is the value you will define for your RS232 connection, and your SCSI Target Mode network if you have one.

Adapter Function This field can either be set to service, standby, or boot. A *service adapter* provides the IP address that is known to the users, and that is in use when the node is running HACMP and is part of the cluster. The *standby adapter*, as we have said before, is an adapter that is configured on a different subnet from the service adapter, and whose function is to be ready to take over the IP address of a failed service adapter in the same node, or the service adapter address of another failed node in the cluster. The *boot adapter* provides an alternate IP address to be used, instead of the service IP address, when the machine is booting up, and before HACMP Cluster Services are started. This address is used to avoid address conflicts in the network, because if the machine is booting after previously failing, its service IP address will already be in use, since it will have been taken over by the standby adapter on another node. A node rejoining the cluster will only be able to switch from its boot to its service address, after that service address has been released by the other node.

Adapter Identifier For a TCP/IP network adapter, this will be the IP address of the adapter. If you have already done your definitions in the /etc/hosts file, as you should have at this point, you do not have to fill in this field, and the system will find its value, based on the Adapter IP Label you have provided. For a non-TCP/IP (serial) network adapter, this will be the device name of the adapter, for instance `/dev/tty0` or `/dev/tmscsi0`.

Adapter Hardware Address This is an optional field. If you want HACMP to also move the hardware address of a service adapter to a standby adapter at the same time that it moves its IP address, you will want to fill in a hardware address here. This hardware address is of your own choosing, so you must make sure that

it does not conflict with that of any other adapter on your network. For token-ring adapters, the convention for an alternate hardware address is that the first two digits of the address are “42.” In our example, we have found out the real hardware address of the adapter by issuing the command `lscfg -v -l tok0`. Our alternate hardware address is the same as the real address, except that we have changed the first two digits to “42.” This ensures that there is not a conflict with any other adapter, since all real token-ring hardware address start with “10....” If you fill in an alternate hardware address here, HACMP will change the hardware address of the adapter from its real address that it has at boot time, to the alternate address, at the same time as it is changing the IP address from the boot address to the service address. If this is done, client users, who only know about the service address, will always have a constant relationship between the service IP address and its hardware address, even through adapter and node failures, and will have no need to flush their arp caches when these failures occur. Alternate hardware address are only used with service adapters, since these are the only adapters that ever have their IP addresses taken over.

Node Name — This is the name of the node to which this adapter is connected. You can list the nodes that you have defined earlier with the F4 key, and choose the appropriate node.

4. Select the **Add an Adapter** option again. Press Enter to display the following panel and fill out the fields for the boot adapter:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node1_boot]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    boot                  +
  Adapter Identifier                                 [9.3.1.3]
  Adapter Hardware Address                           []
  Node Name                                          [Node1]                +









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

Notice that we have defined this adapter having the same network name as the service adapter. Also, you should note that the IP address for the boot adapter is on the same subnet as the service adapter. These two HACMP adapters, boot and service, actually represent different IP addresses to be used on the same physical adapter. In this case, token-ring adapter tok0 will start out on the boot IP address when the machine is first booted, and HACMP will switch the adapter's IP address to the service address (and the hardware address to the alternative address we have defined) when HACMP Cluster Services are started.

5. Press Enter to store the details in HACMP's configuration database.
6. Select the **Add an Adapter** option again. Press Enter and fill out the fields for the IP details for the standby adapter:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node1_stby]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    standby               +
  Adapter Identifier                                 [9.3.4.16]
  Adapter Hardware Address                           []
  Node Name                                          [Node1]                +



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

Notice again that we have used the same network name, since this adapter is on the same physical network. We should also point out that this adapter has been configured on a different subnet from the boot and service adapter definitions. Our netmask was set earlier in the TCP/IP setup to 255.255.255.0.

7. Press Enter to store the details in HACMP's configuration database.
8. Select the **Add an Adapter** option again. Press Enter and fill out the details for the RS232 connection:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [Node1_tty1]
* Network Type                                       [rs232]                +
* Network Name                                       [rs232_net]            +
* Network Attribute                                   serial                +
* Adapter Function                                    service               +
  Adapter Identifier                                 [/dev/tty1]
  Adapter Hardware Address                           []
  Node Name                                          [Node1]                +



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

Note here that we have chosen a different network type and network attribute, and assigned a different network name. Also, the adapter identifier is defined as the device name of the tty being used.

Defining Node 2's Network Adapters: Repeat steps 2 on page 113 through 8 on page 116 to configure the adapters on Node 2. Remember that all the configuration work can be done on one node because you can later synchronize this information to the other node(s) using HACMP's GODM facility.

Enter the service adapter details for Node 2:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node2]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    service               +
  Adapter Identifier                                 [9.3.1.17]
  Adapter Hardware Address                           [0x42005ac908bd]
  Node Name                                          [Node2]                +



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

Here note that we have defined an alternate hardware address for this adapter also, which corresponds to the real hardware address of adapter tok0, with the first two digits changed to “42.”

Enter the boot adapter details for Node 2:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node2_boot]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    boot                  +
  Adapter Identifier                                 [9.3.1.6]
  Adapter Hardware Address                           []
  Node Name                                          [Node2]                +




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

Enter the IP details for Node 2's standby adapter:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [node2_stby]
* Network Type                                       [token]                +
* Network Name                                       [token_net]            +
* Network Attribute                                   public                +
* Adapter Function                                    standby               +
  Adapter Identifier                                 [9.3.4.17]
  Adapter Hardware Address                           []
  Node Name                                          [Node2]                +




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

Enter the details for Node 2's RS232 connection:

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter Label                                      [Node2_tty]
* Network Type                                       [rs232]                +
* Network Name                                       [rs232_net]            +
* Network Attribute                                   serial                +
* Adapter Function                                    service               +
  Adapter Identifier                                 [/dev/tty1]
  Adapter Hardware Address                           []
  Node Name                                          [Node2]                +


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

5.3.7.4 Synchronizing the Cluster Definition on All Nodes

The HACMP configuration database must be the same on each node in the cluster. If the definitions are not synchronized across the nodes, a run-time error message is generated at cluster startup time.

You will use the **Synchronize All Cluster Nodes** option on the `Manage Cluster Environment` panel to copy the cluster definition from Node 1 to Node 2.

```
                            Manage Cluster Environment

Move cursor to desired item and press Enter.

  Configure Cluster
  Configure Nodes
  Configure Adapters
  Synchronize All Cluster Nodes
  Show Cluster Environment
  Configure Network Modules




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

1. Select the **Synchronize All Cluster Nodes** option on the `Manage Cluster Environment` menu and press Enter.

   SMIT responds: `ARE YOU SURE?`

2. Press Enter.

Note:

Before synchronizing the cluster definition, all nodes must be powered on, and the /etc/hosts and /.rhosts files must include all HACMP IP labels.

The cluster definition, including all node, adapter, and network module information, is copied from Node 1 to Node 2.

For more information, refer to Chapter 8, Defining the Cluster Environment, in the *HACMP Installation Guide Version 3.1*.

5.3.8 Defining Application Servers

Application Servers define a highly available application to HACMP. The definition consists of the following:

- Name
- Application start script
- Application stop script

Using this information, the application can be defined as a resource protected by HACMP, and HACMP will be able to start and stop the application at the appropriate time, and on the correct node. Tips for writing Application Server start and stop scripts are included in Chapter 7, “Tips and Techniques” on page 175. Application Server start and stop scripts should be contained on the internal disks of each node, and must be kept in the same path location on each node. To define an Application Server, perform the following tasks:

1. At the command prompt, enter the SMIT fastpath `smit hacmp`. The following panel is presented:

```
                        HACMP/6000

Move cursor to desired item and press Enter.

  Manage Cluster Environment
  Manage Application Servers
  Manage Node Environment
  Show Environment
  Verify Environment
  Manage Cluster Services
  Cluster Recovery Aids
  Cluster RAS Support



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

2. Select **Manage Application Servers** to display the following screen:

```
                    Manage Application Servers

Move cursor to desired item and press Enter.

  Add an Application Server
  Change / Show an Application Server
  Remove an Application Server




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

3. Choose **Add an Application Server** to display the following screen:

```
                          Add an Application Server

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

                                                      [Entry Fields]
* Server Name                                        [itso_server]
* Start Script                                       [/usr/sbin/cluster/itso_sc>
* Stop Script                                        [/usr/sbin/cluster/itso_sc>




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

4. Enter an arbitrary Server Name, then enter the full pathnames for the start and stop scripts. Remember that the start and stop scripts must reside on each participating cluster node. Our script names are:

   - `/usr/sbin/cluster/itso_scripts/startsvr`
   - `/usr/sbin/cluster/itso_scripts/stopsvr`

   Once this is done, an Application Server named `itso_server` has been defined, and can be included in a resource group to be controlled by HACMP.

5.3.9 Creating Resource Groups

In this section we shall go through the steps of defining two *cascading resource groups*, rg1 and rg2, to HACMP. Both nodes will participate in each resource group. Node 1 will have a higher priority for resource group rg1 and Node 2 will have a higher priority for resource group rg2. In other words, Node 1 will own the resources in resource group rg1, and will be backed up by Node 2, while Node 2 will own the resources in resource group rg2, backed up by Node 1. This is called *mutual takeover with cascading resources*.

Resource group rg1 will consist of the following resources:

- /sharedfs1 filesystem
- Node 1 IP address
- NFS export of the /sharedfs1 filesystem

- NFS mount of the /sharedfs1 filesystem
- Application Server itso_server

Resource group rg2 will consist of the following resources:

- /sharedfs2 filesystem
- Node 2 IP address
- NFS export of the /sharedfs2 filesystem
- NFS mount of the /sharedfs2 filesystem

The steps required to set up this configuration of resource groups are as follows:

1. Configure the resource group rg1 on Node 1 by the using SMIT fastpath:

```
# smit cl_mng_res
```

   Then select **Add / Change / Show / Remove a Resource Group** from the following menu:

```
                              Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters




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

2. Select **Add a Resource Group** from the next menu:

```
                  Add / Change / Show / Remove a Resource Group

Move cursor to desired item and press Enter.

  Add a Resource Group
  Change / Show a Resource Group
  Remove a Resource Group




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

3. In the panel that follows, fill out the fields as shown:

```
                              Add a Resource Group

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

                                                      [Entry Fields]
* Resource Group Name                                [rg1]
* Node Relationship                                   cascading              +
* Participating Node Names                           [Node1 Node2]          +




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

In the field `Participating Node Names`, be sure to name the highest priority node *first*. For resource group rg1, this is Node1, since it is the owner. Other nodes participating then get named, in decreasing order of priority. In a two node cluster, there is only one other name, but in a larger cluster, you may have more than two nodes (but not necessarily all nodes) participating in any resource group.

4. Press Enter to store the information in HACMP's configuration database.
5. Press F3 twice to go back to the `Manage Resource Groups` panel. Select **Configure Resources for a Resource Group**.

```
                            Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters




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

6. The list that appears should show only one resource group, rg1. Select this item.

```
                            Select a Resource Group

   Move cursor to desired item and press Enter.

     rg1

   F1=Help                 F2=Refresh              F3=Cancel
   F8=Image                F10=Exit                Enter=Do
   /=Find                  n=Find Next
```

7. In the SMIT panel that follows, fill out the fields as shown. Make sure that the `Inactive Takeover Activated` and the `9333 Disk Fencing Activated` fields are set to **false**.

```
                    Configure Resources for a Resource Group

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

                                                      [Entry Fields]
  Resource Group Name                                 rg1
  Node Relationship                                   cascading
  Participating Node Names                            Node1 Node2

  Service IP label                                    [node1]                 +
  Filesystems                                         [/sharedfs1]            +
  Filesystems to Export                               [/sharedfs1]            +
  Filesystems to NFS mount                            [/sharedfs1]            +
  Volume Groups                                       []                    +
  Concurrent Volume groups                            []                      +
  Raw Disk PVIDs                                      []                      +
  Application Servers                                 [itso_server]                   +
  Miscellaneous Data                                  []

  Inactive Takeover Activated                          false                  +
  9333 Disk Fencing Activated                          false                  +



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
```

The following comments should be made about some of these parameters:

Service IP label By filling in the label of **node1** here, we are activating IP address takeover. If Node 1 fails, its service IP address (and hardware address since we have defined it) will be transferred to the other node in the cluster. If we had left this field blank, there would be no IP address takeover from Node 1 to Node 2.

Filesystems Any filesystems that are filled in here will be mounted when a node takes over this resource group. The volume group that contains the filesystem will first be automatically varied on as well.

Filesystems to Export Filesystems listed here will be NFS exported, so they can be mounted by NFS client systems or other nodes in the cluster.

Filesystems to NFS mount Filling in this field sets up what we call an *NFS cross mount*. Any filesystem defined in this field will be NFS mounted by all the participating nodes, other than the node that currently is holding the resource group. If the node holding the resource group fails, the next node to take over breaks its NFS mount of this filesystem, and mounts the filesystem itself as part of its takeover processing.

Volume Groups This field does not need to be filled out in our case, because HACMP will automatically discover which volume group it needs to vary on in order to mount the filesystem(s) we have defined. This field is there, so that we could specify one or more volume

groups to vary on, in the case where there were no filesystems, but only raw logical volumes being used by our application.

Raw Disk PVIDs — This field is very rarely used, but would be used in the case where an application is not using the logical volume manager at all, but is accessing its data directly from the hdisk devices. One example of this might be an application storing its data in a RAID-3 LUN. RAID-3 is not supported at all by the LVM, so an application using RAID-3 would have to read and write directly to the hdisk device.

Application Servers — For any Application Servers that are defined here, HACMP will run their start scripts when a node takes over the resource group, and will run the stop script when that node leaves the cluster.

8. In the same way, set up the second resource group rg2.

```
# smit cl_mng_res
```

The following panel is displayed:

```
                              Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters









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

Select **Add / Change / Show / Remove a Resource Group**.

```
                  Add / Change / Show / Remove a Resource Group

Move cursor to desired item and press Enter.

  Add a Resource Group
  Change / Show a Resource Group
  Remove a Resource Group




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

Select **Add a Resource Group**. On the resulting panel, fill in the fields, as shown below, to define your second resource group.

```
                              Add a Resource Group

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

                                                      [Entry Fields]
* Resource Group Name                                [rg2]
* Node Relationship                                   cascading              +
* Participating Node Names                           [Node2 Node1]          +



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

Use F3 to go back to the `Manage Resource Groups` panel.

```
                            Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters



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

Select **Configure Resources for a Resource Group**.

```
                          Select a Resource Group

   Move cursor to desired item and press Enter.

     rg1
     rg2

   F1=Help                 F2=Refresh              F3=Cancel
   F8=Image                F10=Exit                Enter=Do
   /=Find                  n=Find Next
```

Choose the resource group **rg2**.

```
                    Configure Resources for a Resource Group

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

                                                      [Entry Fields]
  Resource Group Name                                 rg2
  Node Relationship                                   cascading
  Participating Node Names                            Node2 Node1

 Service IP label                                    [node2]                 +
 Filesystems                                         [/sharedfs2]            +
 Filesystems to Export                               [/sharedfs2]            +
 Filesystems to NFS mount                            [/sharedfs2]            +
 Volume Groups                                       []                    +
 Concurrent Volume groups                            []                      +
 Raw Disk PVIDs                                      []                      +
 Application Servers                                 []                      +
 Miscellaneous Data                                  []

 Inactive Takeover Activated                          false                  +
 9333 Disk Fencing Activated                          false                  +



F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
```

Fill in the appropriate fields, as shown above, and hit Enter to save the configuration.

9. The next job is to synchronize the node environment configuration to the other node. Hit F3 three times to return you to the `Manage Node Environment` panel, as shown below:

```
                                        Manage Node Environment

Move cursor to desired item and press Enter.

  Manage Resource Groups
  Change/Show Cluster Events
  Sync Node Environment




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

Select **Sync Node Environment**. You will see a series of messages, as the ODMs on the other node(s) are updated from the definitions on your node.

You can also synchronize the resource group configuration from the command line by executing the `/usr/sbin/cluster/diag/clconfig -s -r` command.

Note for HACMP Version 2.1 Users

For those users that have used HACMP Version 2.1, it is important for you to note that in HACMP/6000 Version 3.1 and HACMP 4.1 for AIX, the node environment must also be synchronized explicitly, along with the cluster environment. This is a change from HACMP Version 2.1, where the node environment was automatically synchronized by the Global ODM.

10. There are two ways that you can look at your resource group definitions, by node and by resource group.

    You can use the command:

```
# /usr/sbin/cluster/utilities/clshowres -n'<node name>'
```

    to see the resource groups in which a particular node participates. For example:

```
# /usr/sbin/cluster/utilities/clshowres -n'Node1'

Resource Group Name                             rg2
Node Relationship                               cascading
Participating Node Name(s)                      Node2 Node1
Service IP Label                                node2
Filesystems                                     /sharedfs2
Filesystems to be exported                      /sharedfs2
Filesystems to be NFS mounted                   /sharedfs1
Volume Groups                                   havg2
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing

Resource Group Name                             rg1
Node Relationship                               cascading
Participating Node Name(s)                      Node1 Node2
Service IP Label                                node1
Filesystems                                     /sharedfs1
Filesystems to be exported                      /sharedfs1
Filesystems to be NFS mounted                   /sharedfs2
Volume Groups                                   havg1
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
Run Time Parameters:

Node Name                                       Node1
Debug Level                                     high
Host uses NIS or Name Server                    false
```

    Alternatively, you can use the command:

```
# /usr/sbin/cluster/utilities/clshowres -g<resource group name>
```

    to see the configuration of a particular resource group. For example:

```
# /usr/sbin/cluster/utilities/clshowres -g'rg2'

Resource Group Name                          rg2
Node Relationship                            cascading
Participating Node Name(s)                   Node2 Node1
Service IP Label                             node2
Filesystems                                  /sharedfs2
Filesystems to be exported                   /sharedfs2
Filesystems to be NFS mounted                /sharedfs1
Volume Groups                                havg2
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing
Run Time Parameters:

Node Name                                    Node2
Debug Level                                  high
Host uses NIS or Name Server                 false

Node Name                                    Node1
Debug Level                                  high
Host uses NIS or Name Server                 false
```

5.3.10 Verify Cluster Environment

Once you have completed the cluster and node environment definitions, you should verify that the node configurations are consistent and correct over the entire cluster. To verify the cluster enter the SMIT fastpath:

```
# smit hacmp
```

Select **Verify Environment** from the following panel:

```
                        HACMP/6000

Move cursor to desired item and press Enter.

  Manage Cluster Environment
  Manage Application Servers
  Manage Node Environment
  Show Environment
  Verify Environment
  Manage Cluster Services
  Cluster Recovery Aids
  Cluster RAS Support




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

The following panel is presented:

```
                        Verify Environment

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

                                              [Entry Fields]
    Verify Cluster Networks, Resources, or Both         both            +
    Error Count                                        []              #









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

Take the default on this panel, which is to verify both the network configurations and the resource configurations. The Global ODM of HACMP will check the definitions on all nodes, to make sure they are correct and consistent. It will also check various AIX system parameters and system files, to make sure they are set correctly for HACMP, and will check any application server scripts you have defined, to make sure they are on all the nodes where they need to be, and that they are executable. You should see several verification messages, but the results should yield no errors. If you encounter errors, you must diagnose and rectify them before starting the cluster managers on each node. Failure to rectify verification errors will cause unpredictable results when the cluster starts.

5.3.11 Starting Cluster Services

Provided your verification has run without highlighting any errors, you are now ready to start cluster services on one node at a time. Each node should be able to finish its *node_up* processing, before another node is started.

To start cluster services on a node, issue the smit fastpath command `smit clstart`, to bring up the following panel:

```
                        Start Cluster Services

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

                                              [Entry Fields]
* Start now, on system restart or both                now                +

  BROADCAST message at startup?                       false              +
  Startup Cluster Lock Services?                      false              +
  Startup Cluster Information Daemon?                 false              +









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

Here, you can select all the defaults, and hit Enter to start cluster services on the node.

Here are some comments on some of the fields:

Start now, on system restart or both

The recommended setting for this field is to **now**. If you set it to `system restart` or `both`, it will put a record into the /etc/inittab file, so that HACMP cluster services are started automatically on the machine each time it boots. This is not a very good idea, because it may result in a node trying to join the cluster before fixes have been fully tested, or at a time when the impact of resource group movement in the cluster is not desired.

It is much better to have explicit control over when cluster services are started on a node, and for that reason, the **now** setting is recommended.

Startup Cluster Lock Services?

Cluster Lock Services are, almost in all cases, only needed in a concurrent access configuration. The Cluster Lock Manager is normally used to control access to concurrently varied on volume groups. Therefore, it is not needed in this case.

Startup Cluster Information Daemon?

The cluster information daemon, or clinfo, is the subsystem that manages the cluster information provided through the clinfo API to applications. This option would need to be set to true if you were going to be running applications directly on the cluster node that used the clinfo API. An example of such an application would be the cluster monitor `clstat`, which is provided as part of the product. If you are not running such an application, or are running such an application, but on a client machine, this option can be left with its default of `false`.

If you are running a clinfo application on a client machine, it gets its information from the *clsmuxpd* daemon on a cluster node, and does not need clinfo to be running on that cluster node.

When you start cluster services on a node, you will see a series of messages on the SMIT information panel, and then its status will switch to `OK`. This does not mean the cluster services startup is complete, however. To track the cluster

processing, and to know when it is completed, you must watch the two main log files of HACMP:

- /var/adm/cluster.log

  This log file tracks the beginning and completion of each of the HACMP event scripts. Only when the `node_up_complete` event completes is the node finished its cluster processing.

- /tmp/hacmp.out

  This is a more detailed log file, as it logs each command of the HACMP event scripts as they are executing. In this case, you not only see the start and completion of each event, but also each command being executed in running those event scripts.

It is recommended to run the `tail -f` command against each of these log files when you start up nodes in the cluster, so that you can track the successful completion of events, and so that you can know when the processing is completed.

5.4 Upgrading a Cluster to HACMP/6000 Version 3.1

In this section, we will document the procedure to upgrade a cluster running HACMP/6000 Version 2.1 to HACMP/6000 Version 3.1. To do this, we shall continue with our cluster example shown in Figure 32.

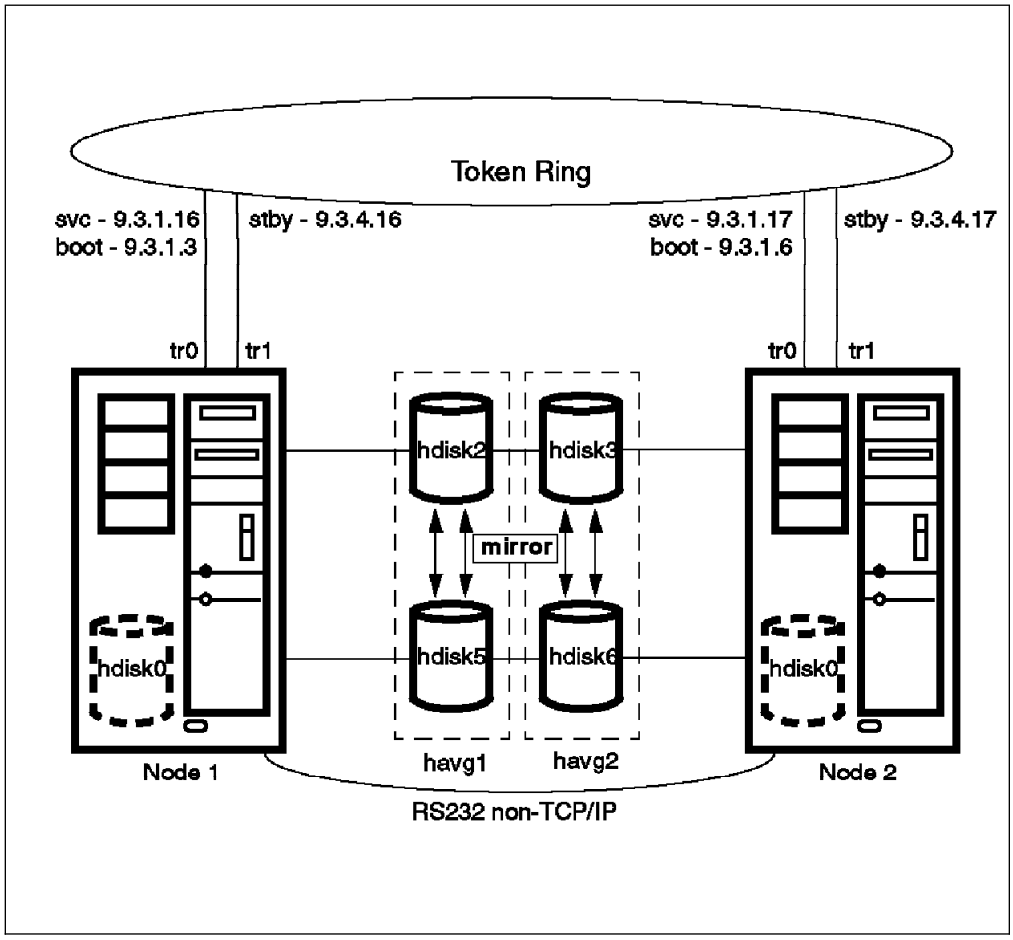


Figure 32. Cluster Running HACMP/6000 Version 2.1

The current configuration of the cluster topology, as shown by the /usr/sbin/cluster/cllscf command is shown in Figure 33 on page 133:

```
Cluster Description of Cluster itso_austin
Cluster ID: 1
There were 2 networks defined : rs232_net, token_net
There are 2 nodes in this cluster.

NODE 1:
        This node has 2 service interface(s):

        Service Interface node1_tty0:
                IP address:     /dev/tty0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node1_tty0 has no standby interfaces.


        Service Interface node1:
                IP address:     9.3.1.16
                Hardware Address:       0x42005a4f4165
                Network:        token_net
                Attribute:      public

        Service Interface node1 has a possible boot configuration:
                Boot (Alternate Service) Interface: node1_boot
                IP address:     9.3.1.3
                Network:        token_net
                Attribute:      public

        Service Interface node1 has 1 standby interfaces.
                Standby Interface 1: node1_stby
                IP address:     9.3.4.16
                Network:        token_net
                Attribute:      public

NODE 2:
        This node has 2 service interface(s):

        Service Interface node2_tty0:
                IP address:     /dev/tty0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node2_tty0 has no standby interfaces.


        Service Interface node2:
                IP address:     9.3.1.17
                Hardware Address:       0x42005ac908bd
                Network:        token_net
                Attribute:      public
```

Figure 33 (Part 1 of 2). HACMP/6000 Version 2.1 Cluster Configuration

```
        Service Interface node2 has a possible boot configuration:
                Boot (Alternate Service) Interface: node2_boot
                IP address:     9.3.1.6
                Network:        token_net
                Attribute:      public

        Service Interface node2 has 1 standby interfaces.
                Standby Interface 1: node2_stby
                IP address:     9.3.4.17
                Network:        token_net
                Attribute:      public

Breakdown of network connections:

Connections to network rs232_net
        Node 1 is connected to network rs232_net by these interfaces:
                node1_tty0

        Node 2 is connected to network rs232_net by these interfaces:
                node2_tty0


Connections to network token_net
        Node 1 is connected to network token_net by these interfaces:
                node1_boot
                node1
                node1_stby

        Node 2 is connected to network token_net by these interfaces:
                node2_boot
                node2
                node2_stby
```

Figure 33 (Part 2 of 2). HACMP/6000 Version 2.1 Cluster Configuration

The current configuration of the cluster node environment, as shown by the command:

```
# /usr/sbin/cluster/clshowres -n'All'
```

is shown in Figure 34 on page 135:

```
Node ID                                     1

Run Time Parameters:
Debug Level                                 high
Takeover for inactive node                  false
Host uses NIS or Name Server                false

Owned Resources:
Filesystems                                 /sharedfs1 /usr/adsmshr
Filesystems to be exported                  /sharedfs1
Volume Groups                               havg1
Concurrent Volume Groups
Disks
Application Servers

Take Over Resources:
Take over from node id                      2
Service IP Label                            node2
Filesystems                                 /sharedfs2
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups                               havg2
Disks
Application Servers

Node ID                                     2

Run Time Parameters:
Debug Level                                 high
Takeover for inactive node                  false
Host uses NIS or Name Server                false

Owned Resources:
Filesystems                                 /sharedfs2
Filesystems to be exported
Volume Groups                               havg2
Concurrent Volume Groups
Disks
Application Servers

Take Over Resources:
Take over from node id                      1
Service IP Label                            node1
Filesystems                                 /sharedfs1
Filesystems to be NFS mounted
Volume Groups                               havg1
Disks
Application Servers
```

Figure 34 (Part 1 of 2). HACMP/6000 Version 2.1 Node Environment Configuration

```
Node ID                                     3

Run Time Parameters:

Owned Resources:

Take Over Resources:

Node ID                                     4

Run Time Parameters:

Owned Resources:

Take Over Resources:
```

Figure 34 (Part 2 of 2). HACMP/6000 Version 2.1 Node Environment Configuration

5.4.1 Prerequisites for Upgrade

Before you start upgrading your cluster, make sure the following prerequisites are met:

1. All the information from your Version 2.1 planning worksheets and diagrams has been transferred to the Version 3.1 worksheets and diagrams.
2. Each node in the cluster has AIX 3.2.5 installed.
3. Each node in the cluster has its own HACMP license.
4. You have root authority.
5. The /usr directory on each cluster node has 3 MB of free space.
6. The /usr directory on each client RS/6000 has 2 MB of free space.

5.4.2 Preparing the Cluster for the Upgrade

We shall now take our cluster through the preparatory steps involved in upgrading a cluster:

1. Stop HACMP services gracefully on both nodes by using the command:

```
# /usr/sbin/cluster/clstop -y -N -s -g
```

2. Ensure that HACMP was stopped successfully on both nodes by looking for the following lines in the /tmp/hacmp.out files on nodes 1 and 2, respectively:

```
EVENT COMPLETED: node_down_complete 1 graceful
 .
 .
 .
EVENT COMPLETED: node_down_complete 2 graceful
```

3. Also check that the service interfaces on both nodes are on their boot addresses and that the only active volume groups are the ones containing internal disks. Run the following commands on Node 1:

```
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                       6801514     0  6801514     0     0
lo0   1536  127         localhost        6801514     0  6801514     0     0
tr1   1492  <Link>                       1313566     0  1187320     0     0
tr1   1492  9.3.4       node1_stby       1313566     0  1187320     0     0
tr0   1492  <Link>                          2678     0     2142     0     0
tr0   1492  9.3.1       node1_boot          2678     0     2142     0     0
# lsvg -o
rootvg
```

Also run the same commands on Node 2:

```
# netstat -i
Name  Mtu   Network     Address         Ipkts    Ierrs Opkts    Oerrs Coll
lo0   1536  <Link>                       6777835     0  6777835     0     0
lo0   1536  127         loopback         6777835     0  6777835     0     0
tr1   1492  <Link>                       1313842     0  1187454     0     0
tr1   1492  9.3.4       node2_stby       1313842     0  1187454     0     0
tr0   1492  <Link>                          1827     0     1485     0     0
tr0   1492  9.3.1       node2_boot          1827     0     1485     0     0
# lsvg -o
rootvg
```

4. Verify that the cluster is configured properly by using the command:

```
# /usr/sbin/cluster/tools/clver -tb
```

This command needs only to be issued on one of the nodes. The output should be similar to this:

```
Verification to be performed on the following:
        Network Topology
        Owned Resources
        Takeover Resources

Retrieving Cluster Topology...


Verifying Network Topology...


Retrieving Cluster Resources...


Verifying File Systems, Volume Groups and Disks on Node 1...


Verifying File Systems, Volume Groups and Disks on Node 2...


Verifying Logical Volume Name Consistency on all nodes...


Verifying Application Servers on all nodes...


Verifying Cluster Events on all nodes...


Verification has completed normally.
```

5. For each node, archive the /usr/sbin/cluster directory to a readily accessible place on disk, so that it is easy to retrieve and compare localized script and configuration files.
6. Check the state of the HACMP/6000 Version 2.1 software and its PTFs on cluster nodes and clients and commit any of these that are in the applied state.

```
# lslpp -h "cluster*"
  Name
  --------------------
    Fix Id  Release         Status     Action     Date       Time     User Name
    ------- --------------- ---------- ---------- ---------- -------- ----------

Path: /usr/lib/objrepos
  cluster.client
            02.01.0000.0000 COMPLETE   COMMIT     09/28/94   14:33:53 root
    U432018 02.01.0000.0000 COMPLETE   COMMIT     09/28/94   14:33:54 root
    U436930 02.01.0000.0000 COMPLETE   APPLY      04/19/95   16:20:32 root

  cluster.server
            02.01.0000.0000 COMPLETE   COMMIT     09/28/94   14:33:55 root
    U432018 02.01.0000.0000 COMPLETE   COMMIT     09/28/94   14:33:56 root
    U436930 02.01.0000.0000 COMPLETE   APPLY      04/19/95   16:20:32 root
```

Here, we need to commit the client and server portions of PTF U436930. Issue the SMIT fast command `smit install_commit`. The following panel is displayed:

```
               Commit Applied Software (Remove Previous Version)

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

                                                      [Entry Fields]
* SOFTWARE name                                      [all]                   +
  COMMIT older version if above version uses it?      yes                    +
  EXTEND file systems if space needed?                yes                    +









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

With the cursor on the field `SOFTWARE name`, press F4 to view the list of uncommitted software:. Select the following items:

```
cluster.client                        02.01.0000.0000.U436930
cluster.server                        02.01.0000.0000.U436930
```

After the commit operation has completed, the SMIT output information should confirm that the software is now in committed state. If you are in any doubt, you can execute the following command to confirm the software states:

```
# lslpp -l "cluster*"
```

The output should show all HACMP software and PTFs to be in a committed state.

7. Make a backup of each node using the `mksysb` command.

5.4.3 Installing the HACMP/6000 Version 3.1 Software

The actual install method will depend on the source from which you are installing the software. You can install from tape or from an image you have loaded into the /usr/sys/inst.images directory.

1. To overwrite the HACMP/6000 Version 2.1 software with the Version 3.1 software, use the SMIT fastpath:

```
# smit install_selectable_all
```

2. Select the appropriate INPUT device or directory.
3. From the next panel, select the required portions of the HACMP software from the list of available `SOFTWARE to install`.
4. Press Enter and check the entries in the following panel:

```
                Install From All Available Software Packages

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

                                                      [Entry Fields]
* INPUT device / directory for software               /usr/sys/inst.images
* SOFTWARE to install                                [3.1.0.0  cluster.clien> +
  Automatically install PREREQUISITE software?        no                      +
  COMMIT software?                                    no                      +
  SAVE replaced files?                                yes                     +
  VERIFY Software?                                    no                      +
  EXTEND file systems if space needed?                yes                     +
  REMOVE input file after installation?               no                      +
  OVERWRITE existing version?                         yes                     +
  ALTERNATE save directory                           []

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

There are two points that you need to consider while installing the Version 3.1 software:

a. The `Automatically Install PREREQUISITE software?` field should be set to **no**.

b. The `OVERWRITE existing version?` field should be set to **yes**.

5. The client and server portions of HACMP/6000 Version 3.1 and of PTF U438726 (the most recent PTF for HACMP/6000 Version 3.1 at the time of writing) are installed on the nodes of our cluster. The command `lslpp -h "cluster*"`, returns the following output on both nodes:

```
  Name
  --------------------
    Fix Id  Release         Status     Action     Date       Time     User Name
    ------- --------------- ---------- ---------- ---------- -------- ----------

Path: /usr/lib/objrepos
  cluster.client
            03.01.0000.0000 COMPLETE   COMMIT     08/05/95   14:34:26 root

  cluster.client
    U438726 03.01.0000.0001 COMPLETE   APPLY      08/05/95   14:49:10 root

  cluster.server
            03.01.0000.0000 COMPLETE   COMMIT     08/05/95   14:34:28 root

  cluster.server
    U438726 03.01.0000.0001 COMPLETE   APPLY      08/05/95   14:49:10 root

Path: /etc/objrepos
  cluster.client
            03.01.0000.0000 COMPLETE   COMMIT     08/05/95   14:34:28 root

  cluster.client
    U438726 03.01.0000.0001 COMPLETE   APPLY      08/05/95   14:51:36 root

  cluster.server
            03.01.0000.0000 COMPLETE   COMMIT     08/05/95   14:34:29 root

  cluster.server
    U438726 03.01.0000.0001 COMPLETE   APPLY      08/05/95   14:51:36 root
```

5.4.4 Upgrading from Version 2.1 to Version 3.1

The steps you need to follow to convert your cluster are:

1. At the command line of any one node, enter:

```
# /usr/sbin/cluster/conversion/clconvert
```

2. You will be prompted to enter the network type for all networks that were defined for your cluster.

```
Input network type.

        Valid Network Type
        ------------------
         ether
         token
         rs232
         socc
         fddi
         IP
         slip
         tmscsi
         fcs
         hps

Type of network "rs232_net" is : rs232
Type of network "token_net" is : token

Network you chose:

        Network "rs232_net" type: rs232
        Network "token_net" type: token
                Any changes? (y/n): n

clconvert successful for this node.  Be sure to synchronize
the cluster before starting HACMP/6000.
```

The highlighted words in the above screen are the ones that need to be entered. The others are prompts from `clconvert`.

3. Synchronize the cluster configuration to all other nodes in the cluster:

```
# /usr/sbin/cluster/diag/clconfig -s -t
```

This can also be done through SMIT, as shown in Section 5.3.7.4, “Synchronizing the Cluster Definition on All Nodes” on page 119.

4. Synchronize the node configuration to all other nodes in the cluster:

```
# /usr/sbin/cluster/diag/clconfig -s -r
```

This can also be done through SMIT, as shown in 9 on page 127.

5. Reboot all cluster nodes.

The `clconvert` utility can be run with the `-F` option. If this option is not used, the `clconvert` utility checks for configured data in new ODM classes. If data is present, it is not upgraded to version 3.1. If the `-F` option is specified, data conversion occurs unconditionally, no new ODM data class checks are made, and all previous data is overwritten.

Now, the configuration of the cluster topology, as shown by the `/usr/sbin/cluster/utilities/cllscf` command, is shown in Figure 35 on page 142.

```
Cluster Description of Cluster itso_austin
Cluster ID: 1
There were 2 networks defined : rs232_net, token_net
There are 2 nodes in this cluster.

NODE 1:
        This node has 2 service interface(s):

        Service Interface node1_tty0:
                IP address:     0.0.0.0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node1_tty0 has no standby interfaces.


        Service Interface node1:
                IP address:     9.3.1.16
                Hardware Address:       0x42005a4f4165
                Network:        token_net
                Attribute:      public

        Service Interface node1 has a possible boot configuration:
                Boot (Alternate Service) Interface: node1_boot
                IP address:     9.3.1.3
                Network:        token_net
                Attribute:      public

        Service Interface node1 has 1 standby interfaces.
                Standby Interface 1: node1_stby
                IP address:     9.3.4.16
                Network:        token_net
                Attribute:      public

NODE 2:
        This node has 2 service interface(s):

        Service Interface node2_tty0:
                IP address:     0.0.0.0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node2_tty0 has no standby interfaces.


        Service Interface node2:
                IP address:     9.3.1.17
                Hardware Address:       0x42005ac908bd
                Network:        token_net
                Attribute:      public
```

Figure 35 (Part 1 of 2). Cluster Configuration After Upgrade from HACMP/6000 Version 2.1

```
        Service Interface node2 has a possible boot configuration:
                Boot (Alternate Service) Interface: node2_boot
                IP address:     9.3.1.6
                Network:        token_net
                Attribute:      public

        Service Interface node2 has 1 standby interfaces.
                Standby Interface 1: node2_stby
                IP address:     9.3.4.17
                Network:        token_net
                Attribute:      public

Breakdown of network connections:

Connections to network rs232_net
        Node 1 is connected to network rs232_net by these interfaces:
                node1_tty0

        Node 2 is connected to network rs232_net by these interfaces:
                node2_tty0


Connections to network token_net
        Node 1 is connected to network token_net by these interfaces:
                node1_boot
                node1
                node1_stby

        Node 2 is connected to network token_net by these interfaces:
                node2_boot
                node2
                node2_stby
```

Figure 35 (Part 2 of 2). Cluster Configuration After Upgrade from HACMP/6000 Version 2.1

The relationship of the cluster nodes and the resource groups, can be seen using the command:

```
# /usr/sbin/cluster/utilities/clshowres -n<node name>
```

For example, the output for node 1 is shown in Figure 36 on page 144.

```
# /usr/sbin/cluster/utilities/clshowres -n1

Resource Group Name                          grp_cas_1
Node Relationship                            cascading
Participating Node Name(s)                   1
Service IP Label
Filesystems                                  /usr/adsmshr
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing

Resource Group Name                          grp_cas_2_1
Node Relationship                            cascading
Participating Node Name(s)                   2 1
Service IP Label                             node2
Filesystems                                  /sharedfs2
Filesystems to be exported
Filesystems to be NFS mounted                /sharedfs1
Volume Groups                                havg2
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing


Resource Group Name                          grp_cas_1_2
Node Relationship                            cascading
Participating Node Name(s)                   1 2
Service IP Label                             node1
Filesystems                                  /sharedfs1
Filesystems to be exported                   /sharedfs1
Filesystems to be NFS mounted
Volume Groups                                havg1
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing
Run Time Parameters:

Node Name                                    1
Debug Level                                  high
Host uses NIS or Name Server                 false
```

Figure 36. Node 1's Resources After Upgrade from HACMP/6000 Version 2.1

The command

```
# /usr/sbin/cluster/utilities/clshowres -n2
```

will return similar output, but without the `grp_cas_1` resource group, because Node 2 does not, at any time, own that resource group.

In HACMP/6000 Version 3.1, you can also see the configuration by resource groups using the command:

```
# /usr/sbin/cluster/utilities/clshowres -g'<resource group name>'
```

For example:

```
# /usr/sbin/cluster/utilities/clshowres -g'grp_cas_1_2'

Resource Group Name                          grp_cas_1_2
Node Relationship                            cascading
Participating Node Name(s)                   1 2
Service IP Label                             node1
Filesystems                                  /sharedfs1
Filesystems to be exported                   /sharedfs1
Filesystems to be NFS mounted
Volume Groups                                havg1
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing
Run Time Parameters:

Node Name                                    1
Debug Level                                  high
Host uses NIS or Name Server                 false

Node Name                                    2
Debug Level                                  high
Host uses NIS or Name Server                 false
```

5.4.5 Further Tasks

Although the `clconvert` utility does most of the work involved in the conversion process, you will usually need to do some further configuration. Among other things, this may include:

- Changing the names of cluster objects
- Changing the organization of resource groups

5.4.5.1 Changing Names of Cluster Objects

The `clconvert` utility generates names for nodes and resource groups. Since nodes in HACMP/6000 Version 2.1 are numbered and not named, the `clconvert` by default uses those numbers for the new node names. You will probably want to change these assigned alphanumeric names to more meaningful ones.

For example, this is how we changed the name of a node in our cluster from `1` to `Node1`:

1. Execute `smit cm_config_nodes` from the command line of any cluster node.
2. Select **Change/Show Cluster Node Name** from the menu that comes up.

```
                              Configure Nodes

Move cursor to desired item and press Enter.

  Add Cluster Nodes
  Change / Show Cluster Node Name
  Remove a Cluster Node









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

3. Select the node whose name you want to change:

```
                   Cluster Node Name to Change/Show

   Move cursor to desired item and press Enter.

     1
     2

   F1=Help                F2=Refresh             F3=Cancel
   F8=Image               F10=Exit               Enter=Do
   /=Find                 n=Find Next
```

4. Specify the new name for this node:

```
                         Change/Show a Cluster Node

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

                                                      [Entry Fields]
* Node Name                                           1
* New Node Name                                      [Node1]









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

You can also use the command:

```
# /usr/sbin/cluster/utilities/clnodename -o'<old name>' -n'<new name>'
```

to change the name of a cluster node. For example:

```
# /usr/sbin/cluster/utilities/clnodename -o'2' -n'Node2'
```

changes the name of the second node in our cluster from `2` to `Node2`.

Remember to change the resource groups after you have changed node names, to have the correct participating nodes. This is not done automatically by changing the node names. Using the same SMIT screen, you can also change the names of the resource groups from the names that they have been automatically given by `clconvert` to more meaningful names.

Changing the name and the participating node names of a resource group is illustrated by the following steps:

1. Execute `smit cm_add_res` from the command line of any cluster node.
2. Select **Change / Show a Resource Group** from the panel that appears:

```
                Add / Change / Show / Remove a Resource Group

Move cursor to desired item and press Enter.

  Add a Resource Group
  Change / Show a Resource Group
  Remove a Resource Group









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

3. Select the resource group that you want to change from the list of resource groups:

```
                        Select a Resource Group

   Move cursor to desired item and press Enter.

     grp_cas_1
     grp_cas_1_2
     grp_cas_2_1

   F1=Help                 F2=Refresh              F3=Cancel
   F8=Image                F10=Exit                Enter=Do
   /=Find                  n=Find Next
```

4. Enter the new name for the resource group into the `New Resource Group Name` field and the new node names in the `Participating Node Names` field:

```
                        Change/Show a Resource Group

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

                                                    [Entry Fields]
  Resource Group Name                               grp_cas_1_2
  New Resource Group Name                           [rg1]
  Node Relationship                                 cascading              +
  Participating Node Names                          [Node1 Node2]          +









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

You can do the same task from the command line. For example, to change the name of the `grp_cas_2_1` resource group to `rg2` and the participating node names to `Node1` and `Node2`, you need to execute the following command:

```
# /usr/sbin/cluster/utilities/clchgrp -g 'grp_cas_2_1' -G 'rg2' -n 'Node2 Node1'
```

That is a long command, so you will probably want to use SMIT.

Important!

Please make all changes to cluster configuration from one node only, and then propagate these changes to other cluster nodes by synchronizing cluster and node configuration across all nodes.

After these changes, the `/usr/sbin/cluster/utilities/cllscf` command gives the output found in Figure 37 on page 149.

```
Cluster Description of Cluster itso_austin
Cluster ID: 1
There were 2 networks defined : rs232_net, token_net
There are 2 nodes in this cluster.

NODE Node1:
        This node has 2 service interface(s):

        Service Interface node1_tty0:
                IP address:     0.0.0.0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node1_tty0 has no standby interfaces.


        Service Interface node1:
                IP address:     9.3.1.16
                Hardware Address:       0x42005a4f4165
                Network:        token_net
                Attribute:      public

        Service Interface node1 has a possible boot configuration:
                Boot (Alternate Service) Interface: node1_boot
                IP address:     9.3.1.3
                Network:        token_net
                Attribute:      public

        Service Interface node1 has 1 standby interfaces.
                Standby Interface 1: node1_stby
                IP address:     9.3.4.16
                Network:        token_net
                Attribute:      public

NODE Node2:
        This node has 2 service interface(s):

        Service Interface node2_tty0:
                IP address:     0.0.0.0
                Hardware Address:
                Network:        rs232_net
                Attribute:      serial

        Service Interface node2_tty0 has no standby interfaces.


        Service Interface node2:
                IP address:     9.3.1.17
                Hardware Address:       0x42005ac908bd
                Network:        token_net
                Attribute:      public
```

Figure 37 (Part 1 of 2). Cluster Configuration After Changes

```
        Service Interface node2 has a possible boot configuration:
                Boot (Alternate Service) Interface: node2_boot
                IP address:     9.3.1.6
                Network:        token_net
                Attribute:      public

        Service Interface node2 has 1 standby interfaces.
                Standby Interface 1: node2_stby
                IP address:     9.3.4.17
                Network:        token_net
                Attribute:      public

Breakdown of network connections:

Connections to network rs232_net
        Node Node1 is connected to network rs232_net by these interfaces:
                node1_tty0

        Node Node2 is connected to network rs232_net by these interfaces:
                node2_tty0


Connections to network token_net
        Node Node1 is connected to network token_net by these interfaces:
                node1_boot
                node1
                node1_stby

        Node Node2 is connected to network token_net by these interfaces:
                node2_boot
                node2
                node2_stby
```

Figure 37 (Part 2 of 2). Cluster Configuration After Changes

The command:

```
# /usr/sbin/cluster/utilities/clshowres -nNode1
```

gives the output found in Figure 38 on page 151.

```
Resource Group Name                          grp_cas_1
Node Relationship                            cascading
Participating Node Name(s)                   Node1
Service IP Label
Filesystems                                  /usr/adsmshr
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing

Resource Group Name                          rg2
Node Relationship                            cascading
Participating Node Name(s)                   Node2 Node1
Service IP Label                             node2
Filesystems                                  /sharedfs2
Filesystems to be exported
Filesystems to be NFS mounted                /sharedfs1
Volume Groups                                havg2
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing

Resource Group Name                          rg1
Node Relationship                            cascading
Participating Node Name(s)                   Node1 Node2
Service IP Label                             node1
Filesystems                                  /sharedfs1
Filesystems to be exported                   /sharedfs1
Filesystems to be NFS mounted
Volume Groups                                havg1
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                            false
9333 Disk Fencing                            false
Run Time Parameters:

Node Name                                    Node1
Debug Level                                  high
Host uses NIS or Name Server                 false
```

Figure 38. Node1 Resources After Changes

The command:

```
# /usr/sbin/cluster/utilities/clshowres -nNode2
```

gives the output found in Figure 39 on page 152.

```
Resource Group Name                             rg2
Node Relationship                               cascading
Participating Node Name(s)                      Node2 Node1
Service IP Label                                node2
Filesystems                                     /sharedfs2
Filesystems to be exported
Filesystems to be NFS mounted
Volume Groups                                   havg2
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing

Resource Group Name                             rg1
Node Relationship                               cascading
Participating Node Name(s)                      Node1 Node2
Service IP Label                                node1
Filesystems                                     /sharedfs1
Filesystems to be exported                      /sharedfs1
Filesystems to be NFS mounted
Volume Groups                                   havg1
Concurrent Volume Groups
Disks
Application Servers
Miscellaneous Data
Inactive Takeover                               false
9333 Disk Fencing                               false
Run Time Parameters:

Node Name                                       Node2
Debug Level                                     high
Host uses NIS or Name Server                    false
```

Figure 39. Node2 Resources After Changes

5.4.5.2 Changing the Organization of Resource Groups

In most cases, you will not have to change the way your resource groups are organized by the `clconvert` utility. The main reason for you to reorganize your resources would be if your verification of the configuration produced any errors.

If you look at the node configuration of our cluster before it was upgraded, you will notice that node 1 owned a filesystem called /usr/adsmshr which was not configured to be taken over by node 2. If you examine the resource group configuration after the cluster was upgraded, you will notice that the `clconvert` utility has created a separate resource group called `grp_cas_1` which contains only the /usr/adsmshr filesystem and has Node1 as the only participating node. The `clconvert` utility has also created another resource group called `grp_cas_1_2` (changed by us to `rg1`) which has a filesystem, /sharedfs1, in the same volume group as /usr/adsmshr, and both nodes as participating nodes.

On running the cluster verification utility, an error is returned saying that the PVIDs of disks required by resource groups `grp_cas_1` and `rg1` are the same. To resolve this error, we have to either totally remove the definition of the /usr/adsmshr filesystem from HACMP and mount it by using pre-event or post-event scripts, or include it in the `rg1` resource group.

It was decided to include it in `rg1`. The steps required to do this are as follows:

1. Change the `rg1` resource group to include the /usr/adsmshr filesystem. Enter the SMIT fastpath command `smit cl_mng_res`. You are presented with the following panel:

```
                          Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters









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

Select **Configure Resources for a Resource Group**. You will now be asked to select which resource group:

```
                          Select a Resource Group

   Move cursor to desired item and press Enter.

     rg1
     rg2

   F1=Help                 F2=Refresh              F3=Cancel
   F8=Image                F10=Exit                Enter=Do
   /=Find                  n=Find Next
```

Here, we select resource group **rg1**. The following panel is presented:

```
                  Configure Resources for a Resource Group

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

                                                      [Entry Fields]
  Resource Group Name                                 rg1
  Node Relationship                                   cascading
  Participating Node Names                            Node1 Node2

  Service IP label                                   [node1]                  +
  Filesystems                                        [/sharedfs1 /usr/adsmshr] +
  Filesystems to Export                              [/sharedfs1]             +
  Filesystems to NFS mount                           []                       +
  Volume Groups                                      [havg1]                  +
  Concurrent Volume groups                           []                       +
  Raw Disk PVIDs                                     []                       +
  Application Servers                                []                       +
  Miscellaneous Data                                 []

  Inactive Takeover Activated                         false                   +
  9333 Disk Fencing Activated                         false                   +

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

On this panel, we add the **/usr/adsmshr** filesystem to those already defined in the `Filesystems` field.

2. Remove the `grp_cas_1` resource group. Enter the SMIT fastpath command `smit cl_mng_res`. You are presented with the following panel:

```
                          Manage Resource Groups

Move cursor to desired item and press Enter.

  Add / Change / Show / Remove a Resource Group
  Configure Resources for a Resource Group
  Configure Run Time Parameters

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

Select **Add / Change / Show / Remove a Resource Group**. The following panel is presented:

```
                Add / Change / Show / Remove a Resource Group

Move cursor to desired item and press Enter.

  Add a Resource Group
  Change / Show a Resource Group
  Remove a Resource Group




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

Select **Remove a Resource Group**. You are now prompted to select a resource group to remove.

```
                          Select a Resource Group

   Move cursor to desired item and press Enter.

     grp_cas_1
     rg1
     rg2

   F1=Help                 F2=Refresh              F3=Cancel
   F8=Image                F10=Exit                Enter=Do
   /=Find                  n=Find Next
```

We choose resource group **grp_cas_1** and hit Enter to remove it.

5.4.6 Verification

After all changes have been made and you are satisfied with the configuration, you need to verify the cluster configuration. From the command line, execute:

```
# /usr/sbin/cluster/diag/clconfig -v '-tr'
```

The output of this command, for our cluster, is shown in Figure 40 on page 156.

```
Contacting node Node2 ...
HACMPcluster ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPnode ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPnim ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPnetwork ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPadapter ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPgroup ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPresource ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPserver ODM on the local node is the same as that on the remote node Node2.

Contacting node Node2 ...
HACMPnode ODM on the local node is the same as that on the remote node Node2.

Verification to be performed on the following:
        Network Topology
        Resources

Retrieving Cluster Topology...


Verifying Network Topology...


Verifying Configured Resources...

Retrieving Resources from Node: Node1...

Retrieving Resources from Node: Node2...

Verification has completed normally.

Verifying Resource Group: rg2
-------------------------

Verifying Resources on each Node...

Verifying Filesystem: /sharedfs2
Verifying NFS-Mounted Filesystem: /sharedfs1
Verifying Volume Group: havg2
```

Figure 40 (Part 1 of 2). Cluster Verification Output After Changes

```
Verifying Resource Group: rg1
-------------------------

Verifying Resources on each Node...

Verifying Filesystem: /sharedfs1
Verifying Filesystem: /usr/adsmshr
Verifying Export Filesystem: /sharedfs1
Verifying Volume Group: havg1


Verifying Resources across Resource Groups
------------------------------------------


Verifying Application Servers
-----------------------------

Verifying Cluster Events on Individual Nodes
--------------------------------------------
Verifying Events on Node: Node1
Verifying Events on Node: Node2

Verification completed normally.
```

Figure 40 (Part 2 of 2). Cluster Verification Output After Changes

If the verification utility reports no errors, you can be confident that there are no problems with your configuration. You can now go ahead and integrate any customization scripts that you might have had in the cluster running HACMP/6000 Version 2.1.

5.5 Upgrading a Cluster from HACMP/6000 Version 3.1

Before HACMP/6000 Version 3.1, you could not have different versions of HACMP coexisting in an active cluster. Version 3.1 and all future versions of HACMP can coexist in a cluster. This means that you need not bring down the entire cluster to upgrade to a higher version of HACMP.

Note

The ability to have nodes at different HACMP levels in the same active cluster is meant to be a migration aid only. Nodes in an active cluster should not be left at different levels of HACMP for an extended time.

The steps that you need to follow to upgrade your cluster from HACMP/6000 Version 3.1 to the next version are:

1. Do a *graceful shutdown with takeover* of HACMP services on one node.
2. Upgrade this node to the next version. (As in the case of HACMP 4.1 for AIX, this may also involve an upgrade of the AIX operating system.)
3. Start up HACMP services on this node and allow it to take back any resource groups for which it has the highest priority in the cluster. At this point, there are two different versions of HACMP running in your cluster.
4. Repeat steps 1 to 3 for the rest of the cluster nodes, one at a time.

The same procedure also applies to installing HACMP or AIX software PTFs on your cluster.

Chapter 6. Cluster Tuning and Customization

By now, you are probably quite aware that HACMP cannot be considered in the same way as an out-of-the-box product. Almost every processing environment is different, and requires careful customization of its HACMP installation. Furthermore, in some cases, the configuration of a cluster may require additional customization to improve performance or to provide stable and predictable cluster behavior. This is what we call *cluster tuning*.

The purpose of this chapter is to discuss various cluster tuning mechanisms and examine the conditions under which a cluster may require tuning. The key areas for discussion are as follows:

- When is Cluster Tuning Required
- False Takeover
- The Deadman Switch
- Tuning System I/O
- Cluster Manager Parameters
- Cluster Customization Examples
- Pre-Event and Post-Event Script Parameters
- AIX Error Notification

6.1 When is Cluster Tuning Required

Once testing of the customization work is complete and the cluster has been commissioned into production, there is usually a period of observation or burn-in, where the system administrator and/or users will observe the cluster to ensure that it is behaving as expected. In many cases, only standard customization procedures are required to tailor the cluster for stable behavior. However, in some environments, the test period and early production may reveal some undesirable behavior.

For example, one or more cluster nodes may have a kernel panic (LED 888) condition for no apparent reason. When the node crashes, a surviving node will usually attempt to acquire any resources for which takeover has been configured. Such a condition is one instance of what is known as *false takeover*.

Experience has shown that unstable cluster behavior is most commonly caused by environmental factors affecting the cluster, rather than by faulty customization. This is supported further when verification of the cluster reveals no configuration errors. If unstable cluster behavior is not corrected promptly, you will find that the cluster environment will provide reduced availability, to the extent where interruptions to application services may occur more frequently than in a single system environment. Clearly this is not the aim of a clustered environment.

How do you eliminate the unstable cluster behavior? By making the cluster less susceptible to environmental factors, you can make it more resilient. You can do this by applying one or more tuning measures to increase the robustness of the cluster.

6.2 False Takeover

As described earlier, false takeover occurs when an active node is failed by its own Cluster Manager for abnormal reasons, or when the Cluster Managers on other nodes incorrectly diagnose that failure of an active node has occurred. A surviving node will then attempt to acquire resources from the node it thinks has failed. The surviving partner Cluster Managers also mark that node with a status of *down*.

Generally speaking, problems associated with false takeover relate to the Cluster Manager on an active node being unable to send keepalive packets to its partner node or nodes, and have them received there successfully. In a stable cluster, all Cluster Managers transmit keepalive packets at a default rate of one every half second. If, for some reason, a Cluster Manager is unable to transmit any keepalive packets out to its partner nodes within a default period of five seconds, a timeout occurs and a kernel panic is initiated. This fails the node, and thereby makes its resources available for clean takeover by a partner node. A node crash of this type is initiated by a feature called the *Deadman Switch*.

6.3 The Deadman Switch

The Deadman Switch (DMS) is a kernel extension that permits the Cluster Manager to fail or crash the node on which it is running, if that node is not able to send keepalive packets for a long enough time such that the other nodes would start takeover processing. The Deadman Switch is reset by the Cluster Manager, on each pass through its main processing loop. If the Cluster Manager is blocked and cannot get through its loop within a set interval of its most recent loop, it also will not have reset the DMS timer. The system is crashed by a kernel panic when a DMS timeout occurs. The timeout value varies according the particular release of HACMP. In Version 2.1 the default value is five seconds. In Versions 3.1 and 4.1, the value depends on the type of network interfaces installed in the cluster nodes.

The DMS is a useful mechanism, since it works to prevent false takeovers from occurring. In other words, it prevents other nodes trying to take over resources from a node that is still running. The node that is not able to send keepalive packets for the designated time, but is still up, is able to take itself out of the cluster, and prevent the other nodes from having to fight over its resources. However, experience has revealed that the application environments of some clusters can cause sporadic DMS timeouts, and cause inappropriate node crashes.

6.3.1.1 Finding the Culprit

It is important to first establish whether or not the DMS is responsible for the node crash. This can be done by analyzing the system dump using the following steps.

1. Use the `crash` command to determine if the crash has been caused by the Deadman Switch. The `crash` command displays system images for examining a dump. Use the following syntax after rebooting the failed node:

```
# crash /dev/hd7
```

2. At the crash program prompt, type `stat`. If the Deadman Switch is responsible for the node crash, the panic message will indicate a “dms timeout.” The `trace -m` subcommand will also give further information.

```
> stat
        sysname: AIX
        nodename: hadave1_boot
        release: 2
        version: 3
        machine: 000007913400
        time of crash: Mon Jun  5 16:27:07 1995
        age of system: 9 day, 22 hr., 27 min.
            panic: HACMP dms timeout - halting hu

>       trace -m
        0x211db0 (excpt=205f88:40000000:0:205f88:106) (intpri=5)
        IAR:        0006bebc  .get_last_msg + 44:      lbzx   r4,r9,r7
        *LR:        0006c010  .panic + a8
        *00211c78:  00000000  <invalid>
        00211c78:   014e8284  [<dms>:dead_man_sw_handler] + 1bc
        00211cc8:   014e88ac  [<dms>:timeout_end] + 7e4
        00211d28:   00033d2c  .clock + f0
        00211d78:   000215a4  .i_poll + 64
        00008714:   000094e0  <invalid>
```

3. At this point, it is useful to determine which process had the kernel lock at the time of the crash. This will help identify a process involved in heavy I/O activity. Type the following:

```
> od *kernel_lock
```

   This command will display a process id (PID) in hexadecimal, if the kernel lock was involved in the crash. The crash command lists processes in PID sequence with an associated slot number. To make it easier to locate the correct process, you can perform a binary sort on the process table.

   For example, type `p 10` to list the process in slot 10:

```
> p 10
SLT ST    PID   PPID   PGRP   UID  EUID  PRI   CPU   EVENT  NAME
10 s      a54     1   a54     0     0    60     0 05de52a4 cron
        FLAGS: swapped_in wake/sig
```

   Repeat this until you find the process with the correct PID (that is, the one returned by the `od *kernel_lock` command).

Note

The advice about the kernel lock is only valid for AIX 3.2.5. Since the AIX 4.1 kernel is multi-threaded, processes typically do not use a global kernel lock, so the same bottleneck is not present.

Once you have determined that the Deadman Switch has caused your system crash, you can tune your environment to avoid the DMS timeout.

In the following sections, we will examine the conditions which can cause the Deadman Switch to crash a system and then discuss ways of tuning the cluster to avoid such conditions. In particular, we shall focus on the following areas:

1. System I/O
   - Disk I/O pacing

- syncd daemon
- Disabling the Deadman Switch

2. Cluster Manager startup parameters
3. Pinning the Cluster Manager

6.4 Tuning System I/O

Most situations of a node crash caused by a DMS timeout relate to an excessively high system I/O or processing load.

If a cluster node is expected to carry a high level of I/O throughput, it may have an excessive amount of I/O wait time. It is well known that excessive I/O wait time has an undesirable impact on system performance. It is also regarded as an enemy of the Cluster Manager. The reasons for this have to do with the kernel's locking mechanism.

The *kernel lock* is a lock used by kernel processes to serialize access to global data structures. Since AIX Version 3.2 is single-threaded, there is only one kernel lock. Only one process can hold the kernel lock at a time. The Cluster Manager must acquire the kernel lock during each keepalive cycle, so that it can check the status of each of its adapter resources. Furthermore, in the case of non-TCP/IP networks, like RS232 and SCSI Target Mode, the Cluster Manager must have the kernel lock in order to send keepalive packets. During periods of intense I/O activity, the Cluster Manager process may be required to wait before it is able to obtain the kernel lock. If conditions are particularly severe, the timeout period for the Deadman Switch may pass before the Cluster Manager can get the lock. Five seconds may seem like a substantial amount of time for a process to be blocked. However, there are situations where this can happen. For instance, an I/O intensive process that is actually paged out while holding the kernel lock, will require many CPU cycles to complete. The system must page in the process and its associated data sets so that outstanding I/Os may be completed and the kernel lock then released. Also, a large buildup of I/O will take a long time to complete before the kernel lock is released.

It should be noted that we are not necessarily referring to a system environment where the I/O workload is nearing the limits of the system's capacity. Transient peaks in the I/O workload are sufficient to create these conditions for the Cluster Manager.

AIX Version 4, being multi-threaded, allows more layers of granularity in terms of kernel locking, and does alleviate the problem of the cluster manager being denied the kernel lock when requested.

6.4.1 I/O Pacing

Let us first turn to a brief discussion of disk I/O pacing and then examine the way in which it can be used to make your cluster resistant to false takeover.

Users of AIX occasionally encounter long interactive application response times when another application on the system is doing large writes to disk. Because most writes are asynchronous, FIFO I/O queues of several MB can build up, which can take several seconds to complete. The performance of an interactive process is severely impacted if every disk read spends several seconds working its way

through a queue of pending I/Os. In response to this problem, the Virtual Memory Manager has an option called *I/O Pacing* to control the amount of resource given to disk writes. This feature was introduced in AIX Version 3.2.

I/O pacing does not change the interface or logic of I/O. It simply limits the number of I/Os that can be outstanding against a given file. When a process tries to exceed that limit, it is suspended until enough outstanding requests have been processed to reach a lower threshold.

6.4.1.1 Use of Disk I/O Pacing

In systems where disk I/O pacing is disabled, programs that generate a very large amount of output may saturate the system's I/O facilities and cause the response times of less demanding programs to deteriorate. Remember that large I/O build ups may lock the kernel for several seconds, and that the Cluster Manager competes for the same lock every half second to complete a keepalive cycle. If, under these conditions, the Cluster Manager cannot successfully get keepalive packets out within the time limit set for the Deadman Switch, the node will panic.

I/O pacing allows you to tune the system in a way that more equitably distributes system resources during large disk writes. This is done by setting high-water and low-water marks on the sum of pending I/Os. When a process tries to write to a file that already has pending writes at the high-water mark, the process is put to sleep until enough I/Os have been completed to make the number of pending writes less than or equal to the low-water mark.

We recommend that you start with the following high-water and low-water marks:

high-water mark = 33

low-water mark = 24

The above values should be sufficient to permit the Cluster Manager to process its keepalive cycles, even during temporary periods of high I/O activity. However, if the active node often suffers from an I/O intensive load, the above settings could impact interactive performance. Some experimentation will be required to find the best settings for your workload.

The high-water and low-water marks can be set with SMIT. Enable disk I/O pacing by changing the high-water and low-water marks from their default values of zero (disk-I/O pacing disabled) to the values shown in the following SMIT panel, obtained by entering the `smit chgsys` command:

```
              Change / Show Characteristics of Operating System

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

                                                      [Entry Fields]
  Maximum number of PROCESSES allowed per user       [40]                    +#
  Maximum number of pages in block I/O BUFFER CACHE  [20]                    +#
  Maximum Kbytes of real memory allowed for MBUFS    [2048]                  +#
  Automatically REBOOT system after a crash           true                   +
  Continuously maintain DISK I/O history              true                   +
  HIGH water mark for pending write I/Os per file    [33]                    +#
  LOW water mark for pending write I/Os per file     [24]                    +#
  Enable memory SCRUBBING                             false                  +
  Amount of usable physical memory in Kbytes          32768
  Primary dump device                                 /dev/hd7
  Secondary dump device                               /dev/sysdumpnull
  Error log file size                                 1048576
  State of system keylock at boot time                normal
  Size of data cache in bytes                         32K
  Size of instruction cache in bytes                  8K

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

6.4.2 sync Daemon (syncd)

A reduction in I/O buildup in disk-active nodes can also be assisted by changing the behavior of the *sync* daemon. This can be done by allowing the sync daemon to run more often, so that frequency with which the I/O buffers are flushed is increased. The aim here is to prevent the build up of long I/O queues, which may take several seconds to complete, by forcing writes more often. Since the flush operation requires the kernel lock, this technique reduces the length of time the kernel is locked.

To change the interval between successive instances of sync, change the following line in the /sbin/rc.boot file:

```
nohup /etc/syncd 60 > /dev/null 2>&1 &
```

If you intend to make this change, a recommended starting point is a change from the default value of 60 seconds to a value of 10 seconds. Your line in /sbin/rc.boot must look like this:

```
nohup /etc/syncd 10 > /dev/null 2>&1 &
```

You will need to reboot the system after making this change for it to take effect. When the cluster is stable and all applications are running, you can monitor the system CPU utilization for excessive use by the sync daemon. The interval value could then be increased if necessary.

In the same way that finding the most suitable values for disk I/O pacing is a process of trial and error, some experimentation may be required to find the most

appropriate value for the sync daemon. That is, increasing the interval for syncd makes the system performance acceptable, but leaves the problem of I/O build up and the DMS system crash unresolved. You can see that there are trade-offs in this approach. A good solution for one cluster may not be appropriate for a different one.

6.4.3 Disabling the Deadman Switch

In extreme circumstances where all tuning methods have failed to prevent the Deadman Switch from crashing a cluster node, you may wish to disable it. Be aware, however, that disabling the Deadman Switch also disables the ability of a node to take itself out of the cluster if it finds that it is unable to send keepalive packets. This can lead to a false takeover, where other nodes try to take over resources when the owning node is still up. In most cases, disabling the Deadman Switch is not a recommended solution.

Since HACMP is started by the System Resource Controller, startup information for the Cluster Manager is kept in the ODM object class SRCsubsys. Before you modify the Cluster Manager subsystem, list any arguments that may already be active for Cluster Manager startup, by invoking the following command:

```
# lssrc -Ss clstrmgr | awk -F: '{print $3}'
```

Note any arguments which appear after the word *cmdargs* in the output. The following command can then be used to disable the Deadman Switch under HACMP/6000 Version 2.1 and HACMP/6000 Version 3.1 by making the appropriate changes to the SRCsubsys object class. Be sure to also include any arguments that you listed before, using the `lssrc` command. Arguments must be separated by spaces and enclosed in single quotation marks.

```
# chssys -s clstrmgr -a '-D'
```

Since the `chssys` command with the `-a` flag overwrites any existing arguments, verify that the desired change has been made by executing the `lssrc` command.

```
# lssrc -Ss clstrmgr | awk -F: '{print $3}'
```

Once you have restarted the Cluster Manager, it is a good idea to check that it is running with the parameters you expect. You can run the `ps` command to check your work. For the example above, enter the following:

```
# ps -ef | grep clstrmgr
    root 13316  6050   4 17:22:35  pts/0  0:00 grep clstrmgr
    root 14934  1338   4 17:20:55      -  0:04 /usr/sbin/cluster/clstrmgr -D
```

Check the command tail for the `clstrmgr` process. This will indicate if the correct parameters are present.

6.5 Cluster Manager Startup Parameters

So far, we have discussed the methods of tuning cluster behavior by exploiting the intrinsic functionality of AIX. In some situations, the tuning of AIX's I/O subsystem may not improve cluster stability to an acceptable level. Therefore, it may be necessary to modify the run-time characteristics of the Cluster Manager to enhance cluster stability and reduce the chance of false takeover. Most importantly, modifying the Cluster Manager parameters allows you to change the timeout period for the Deadman Switch and the elapsed time required for cluster event detection.

In this section we shall describe the use of the Cluster Manager startup parameters which modify some important characteristics of the Cluster Manager subsystem.

The mechanisms for tuning the cluster parameters in HACMP/6000 Version 3.1 differ from the those in HACMP/6000 Version 2.1. We recognize that many clusters may remain at HACMP/6000 Version 2.1 for some time, so both mechanisms are presented here.

6.5.1 HACMP/6000 Version 2.1

In earlier releases of HACMP, the value of Deadman Switch timeout was a constant, fixed at five seconds. The Deadman Switch, as well as other Cluster Manager characteristics, can now be modified by changing certain variables. These are defined as follows:

normal_ka_cycles This is the number of microseconds between keepalives. The default value is 500000, which is 0.5 seconds.

cycles_to_fail This value is used to define the number of missed keepalives that determine a minor failure. It is also used as the number of cycles each node will wait for synchronization prior to continuing. The default value is 4.

ifs_fails_til_netfail This value is used for determining the number of minor failures permitted prior to failing a remote interface. The default value is 4.

The switches used to change the Cluster Manager variables are defined in Table 4.

Table 4. Cluster Manager Startup Switches

Switch	Variable Modified
-n	normal_ka_cycles
-f	cycles_to_fail
-a	ifs_fails_til_netfail

Note that each parameter requires an integer value. Modifying one or more of these switches will change the run-time behavior of the Cluster Manager.

Now, let us explain these variables in more detail and discuss cases where it is appropriate to change them.

Changes to the Cluster Manager parameters affect failover time, and the timeout limit imposed by the Deadman Switch Remember that the timeout for the Deadman Switch in prior releases of HACMP was a constant set to five seconds. The

change from a constant to a value represented by an expression was made available in PTF U432018 and in subsequent cumulative PTFs. It is calculated in the following way:

```
zombie_timeout = (normal_ka_cycle * (cycles_to_fail - 1)
                  * ifs_fails_til_netfail) - 1
```

If you base your calculation on the default values for the Cluster Manager, the Deadman Switch timeout will be set to five seconds:

```
normal_ka_cycle       = 0.5 sec (500,000 microseconds)
cycles_to_fail        = 4
ifs_fails_til_netfail = 4

zombie_timeout = (0.5 * (4 - 1) * 4) - 1
               = 6 - 1
               = 5 seconds
```

Detection of a network failure is calculated in the following way:

```
net_fail_time = normal_ka_cycle * (cycles_to_fail - 1)
                * ifs_fails_til_netfail
```

Therefore, based on the above values, a network failure would be detected at 6 seconds; given by:

```
net_fail_time = 0.5 * (4-1) * 4
              = 6 seconds
```

Note that this would also constitute a node failure if the keepalive packets were missed on all networks. Note also that the Deadman Switch is designed to time out one second before a network failure detection. If the Deadman Switch is disabled and the Cluster Manager on that node is unable to send keepalive packets, then a node failure would be detected by partner nodes when the threshold (six seconds in the above example) for missed keepalive packets is exceeded.

6.5.1.1 Changing the Cluster Manager Parameters

When you change the Cluster Manager parameters, it is important you make consistent changes on all nodes in the cluster. To activate any changes, you must first stop and restart the Cluster Manager.

Length of Keepalive Cycles

The length of the keepalive cycle can be made longer or shorter with the `-n` switch. The `-n` switch modifies the value of the normal_ka_cycles variable. To change its value, you must change the subsystem definition for the Cluster Manager:

```
# chssys -s clstrmgr -a '-n NNNNN'
```

where `NNNNN` is the cycle duration in microseconds. The default value of 500000 microseconds is equivalent to 0.5 seconds. To change the length of the keepalive cycle to 0.75 seconds, enter:

```
# chssys -s clstrmgr -a '-n 750000'
```

Realize that lengthening the keepalive cycle will slow down rate at which the Cluster Manager sends keepalive packets, causing it to take longer for cluster events to be detected and initiated. In most instances, a better method would be to alter the cycles_to_fail variable.

Number of Missed Keepalive Packets
The number of missed keepalive packets defines a minor event (or failure). Its value is changed by modifying the `-f` switch. The `-f` switch modifies the value of the cycles_to_fail variable. To change its value, you must change the subsystem definition for the Cluster Manager:

```
# chssys -s clstrmgr -a '-f N'
```

where `N` is the number of keepalive cycles missed. The default value is 4. To change the the number of keepalive cycles missed to 8, for instance, enter:

```
# chssys -s clstrmgr -a '-f 8'
```

Four minor failures constitute a major failure, at which point the surviving nodes in a cluster will mark the status of another node or network as down. Adjusting this value will allow additional time to pass before the nodes of a cluster will determine that a node or network has failed. As mentioned before, this is a better option than modifying the value of normal_ka_cycles, because here we are increasing the time required for a fallover, but not slowing the rate of keepalive processing.

In the above example, if you set the cycles_to_fail equal to 8, the Deadman Switch will be set as follows:

```
deadman switch timeout = (0.5 * (8 - 1) * 4) - 1
                       = 13 seconds
```

By definition, a network or node failure would occur one second later, at 14 seconds.

6.5.2 HACMP/6000 Version 3.1

With HACMP/6000 Version 3.1, you can configure the parameters of the Cluster Manager more easily than with HACMP/6000 Version 2.1. Version 3.1 offers the ability to do this through its SMIT interface. It allows you to set a failure detection rate of Slow, Normal, or Fast, independently for each network type that you have configured in your cluster. HACMP/6000 Version 3.1 distinguishes between “network types” such as Ethernet/IP, Token Ring, Target Mode SCSI or RS232, FDDI, SLIP and SOCC. Each of the network types that you have in your cluster has its own Network Interface Module (or NIM) in each node, responsible for sending and receiving keepalive packets, and detecting network failures. The effect

of the Slow, Normal or Fast settings for the various network types are shown in Table 5 on page 169.

Table 5. Cluster Manager Failure Detection Rates

Interface		Slow	Normal	Fast
Ethernet/IP	KA Rate (sec)	1.0	0.5	0.5
	Missed KAs	12	12	9
Token Ring	KA Rate (sec)	1.0	0.5	0.5
	Missed KAs	24	24	12
TMSCSI/RS232	KA Rate (sec)	3.0	1.5	0.5
	Missed KAs	8	6	5
SOCC	KA Rate (sec)	1.0	0.5	0.5
	Missed KAs	12	12	9
FDDI	KA Rate (sec)	1.0	0.5	0.5
	Missed KAs	12	12	9
SLIP	KA Rate (sec)	2.5	1.0	0.5
	Missed KAs	18	12	12

Normal and *slow* are the recommended settings. *Slow* is useful in cluster environments where temporary network or CPU load situations might otherwise cause false takeovers to occur. In particular, a setting of *slow* is useful in situations where there might be temporarily intense I/O load. A setting of *fast* will give you a faster reaction to failures, but exposes a node to the danger of false takeover. Node failure processing is based on the *Missed KA* value for the fastest network.

For example, if a cluster uses ethernet and RS232 networks, and if the failure detection rates for both networks are set to *normal*, event processing will commence after twelve keepalive packets are missed on the ethernet network. This means that event processing will commence after six seconds. If the RS232 network were the faster of the networks, event processing would commence after nine seconds. This differs from HACMP/6000 Version 2.1, where failure detection migrated to the slowest network.

Event (or failure) detection time is defined by:

```
(KA_rate * missed_KAs)
```

which is equivalent to:

```
(KA_cycle_time * cycles_to_fail)
```

in Version 2.1 terminology. This does differ slightly from the HACMP/6000 Version 2.1 formula, since the variable *ifs_fails_til_netfail* is no longer used.

The Deadman Switch timeout is defined by the formula:

```
(KA_rate * missed_KAs) - 1
```

For an ethernet interface configured for a *normal* failure detection rate, the Deadman Switch timeout would be defined on the basis that:

```
KA_rate = 0.5 sec
missed_KAs = 12
```

Therefore:

```
Deadman Switch timeout = (0.5 * 12) - 1
                       = 5 seconds
```

Notice that this is the same value for Deadman Switch timeout in Version 2.1. However, if you perform the same calculation for other interface types, the value for the Deadman Switch timeout will be different.

You can change the failure detection rate by entering the SMIT fastpath command `smit cm_config_networks`. The following panel will be displayed:

```
                          Configure Network Modules

Move cursor to desired item and press Enter.

  Add a Network Module
  Change / Show a Network Module
  Remove a Network Module









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

Using ethernet as an example, you can change the value of the Failure Detection Rate:

```
               Change / Show Cluster Network Module

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

                                                      [Entry Fields]
* Network Module Name                                 ether
  New Network Module Name                             []
  Description                                         [Ethernet Protocol]
  Address Type                                        Address              +
  Path                                                [/usr/sbin/cluster/nims>]
  Parameters                                          []
  Failure Detection Rate                              normal               +









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

6.5.3 Pinning the Cluster Manager

In early releases of HACMP, the Cluster Manager was treated by the Virtual Memory Manager in the same way as any other process. That is, it could be swapped out of memory. In HACMP/6000 Version 2.1 and Version 3.1, the Cluster Manager is *pinned* in memory by default at invocation time. This is done to prevent paging in being a factor of delays in sending out keepalive packets, and therefore to prevent one cause of false takeovers.

Nevertheless, it is possible to invoke the Cluster Manager so that it is pageable. To prevent the Cluster Manager from being pinned at invocation time, you can change the characteristics of the Cluster Manager subsystem by using the following command:

```
# chssys -s clstrmgr -a '-l'
```

You may want to make this change if an active node is suffering from a shortage of memory resources. Of course, this is not a recommended remedy, not only for the reasons stated above, but also because this suggests a more pervasive memory shortage that is affecting the application workload. A better solution is to leave the Cluster Manager pinned in memory and upgrade the system's memory capacity.

6.6 HACMP Cluster Customization Examples

In this section, we will list the possible customizations that most production environments usually require. We will not discuss any of these in detail as there is another redbook, *HACMP Customization Examples*, that is dedicated to this subject.

Most customizations that you undertake while setting up a production cluster will require you to write pre or post-event scripts that will automatically do the following:

- Enable and disable user accounts on nodes joining the cluster or on nodes backing up a failed node.
- Forward mail from a takeover node to a failed node when it reintegrates into the cluster.
- Copy customization scripts across from a surviving node to a failed node after the failed node reintegrates.

  This ensures that any change that may have been made in the configuration of a cluster's reactions to events is propagated to a node that was down at the time the change was made.
- Kill any processes that continue to run on the takeover node after the failed node has reintegrated and has started providing normal services to users.
- Notify system administrators and/or users about the occurrence of events.
- Escalate local network failures to node failures.
- Notify system administrators about the failure of system hardware or software using the AIX error notification facility.

  This is especially important for failures in the storage subsystem in your cluster. Since HACMP does not detect any failures on the I/O bus, and

you would have designed your storage with redundancy to protect against failures, you may not notice a disk, adapter, controller, or cable failure.

- Shut down the cluster gracefully if there is a power failure and the battery backup starts providing power.
- Transfer print queues from a failed node to a takeover node.
- Continue a backup that was running on a failed node on a takeover node.

6.7 Pre-Event and Post-Event Script Parameters

When you specify pre or post-event scripts to HACMP, these scripts get called with a fixed set of parameters by the Cluster Manager.

All pre-event scripts get called with the following parameters:

- The name of the corresponding event as the first parameter.
- All the parameters passed to the corresponding event script as trailing parameters.

All post-event scripts get called with the following parameters:

- The name of the corresponding event as the first parameter.
- The exit code of the corresponding event script as the second parameter.
- All the parameters passed to the corresponding event script as trailing parameters.

6.8 AIX Error Notification

HACMP provides an interface to the AIX error logging facility. Through one of the HACMP SMIT screens, you can specify a command or script to be executed whenever an entry corresponding to a particular error is made in the AIX error log.

You would usually configure this facility to take some action for permanent hardware and software errors. The error identifiers can be found in the /usr/include/sys/errids.h file. A resource name, class, and type will uniquely identify a device to the error notification facility. The notify method that you specify can be a command, a script, or an executable.

For example, in order to set up error notification to inform the root user about a permanent hardware error on a SCSI Adapter (scsi1), enter the following command:

```
# smit cm_EN_menu
```

From the SMIT menu that appears, select **Add a Notify Method**

```
                              Error Notification

Move cursor to desired item and press Enter.

  Add a Notify Method
  Change/Show a Notify Method
  Delete a Notify Method









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

Use the screen that appears to set up the notification method.

```
                            Add a Notify Method

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

                                                    [Entry Fields]
* Notification Object Name                         [P_SCSI_ERROR]
* Persist across system restart?                    Yes                    +
  Process ID for use by Notify Method               []                     +#
  Select Error Class                                Hardware               +
  Select Error Type                                 PERM                   +
  Match Alertable errors?                           None                   +
  Select Error Label                                [SCSI_ERR1]            +
  Resource Name                                     [scsi1]                +
  Resource Class                                    [adapter]              +
  Resource Type                                     [adapter]              +
* Notify Method                                     [echo scsi1 Failed |
                                                    mail root]




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

The following arguments are automatically passed on to all notify methods:

$1 Sequence number form the error log entry

$2 Error ID from the error log entry

$3 Error class from the error log entry

$4 Error type from the error log entry

$5 Alert flag values from the error log entry

$6 Resource name from the error log entry

$7	Resource type from the error log entry
$8	Resource class from the error log entry

Once you have set up a notify method, you can delete it or change it at any time through the menu you obtain using the SMIT fast path `cm_EN_menu`.

Chapter 7. Tips and Techniques

The aim of this chapter is to give you some tips and techniques to help you manage your clusters more effectively and to enable you to recognize and remedy some of the more common problems that occur from time to time. The following topic areas are covered in this chapter:

- Change Management in an HACMP cluster
- Mirroring the Root Volume Group (rootvg)
- Quorum
- JFS Log and /etc/filesystems
- Filesystem Helper
- Phantom Disks
- /etc/inittab File
- Permissions on the /tmp directory
- ARP Cache
- Synchronizing time between cluster nodes
- Tips on writing application server and event scripts
- Recovery from event script failure
- AIX error notification
- 7135 RAIDiant disk array
- Resource group organization
- Hubs in an HACMP cluster

7.1 Change Management

A commonly held view about HACMP clusters is that there should be little or essentially no system downtime. While this one of the goals of clustering, the role of HACMP is really to eliminate *unplanned* downtime. It is a fallacy to propose an HACMP cluster based on the pretext that downtime will never occur or will never be required. This is because all computer systems, no matter how highly available, will require some periodic downtime to enable various system management activities to be performed. The discipline associated with performing such activities is called *change management*.

There are several reasons why change management is an essential discipline in a cluster environment. Several administration and maintenance functions require *planned* downtime. Some of these are:

- Cluster verification
- Software upgrades and fixes
- Cluster maintenance
- Additional filesystems or disk resource
- New applications

- New communications connectivity

When you implement a change management discipline, by definition, it requires *planned* downtime. Let us examine some of the topics mentioned above.

7.1.1 Cluster Verification

An interesting feature of HACMP is its apparent dormancy. When the cluster is stable, you hardly know that HACMP exists, but when something goes wrong (a cluster resource fails), its existence becomes unquestionably prominent. The problem is that when a cluster is stable and has been running for a long period of time without a failure, how can you be sure, when something does fail, that failover will succeed as expected. Something may go wrong on a node that affects the Cluster Manager's ability to correctly handle failover.

To maximize your confidence that the cluster is healthy, you should periodically test node failover in the same way it was tested before acceptance. This requires planned downtime, and is an essential form of preventative maintenance to safeguard against potential catastrophe.

7.1.2 Software Upgrades and Fixes

As with everything else in a cluster, applying software fixes should be done in a controlled fashion. The normal method of applying AIX fixes is to do the following:

- Perform a controlled failover (shutdown graceful with takeover) from one node in the cluster. For example, in Figure 41 on page 177, Node A's resources are failed over to Node B.
- Apply AIX PTFs to Node A, which has now left the cluster.
- Fully test the fixes while the node is outside of the cluster.
- Reintegrate Node A into the cluster, during a period of low usage if possible.

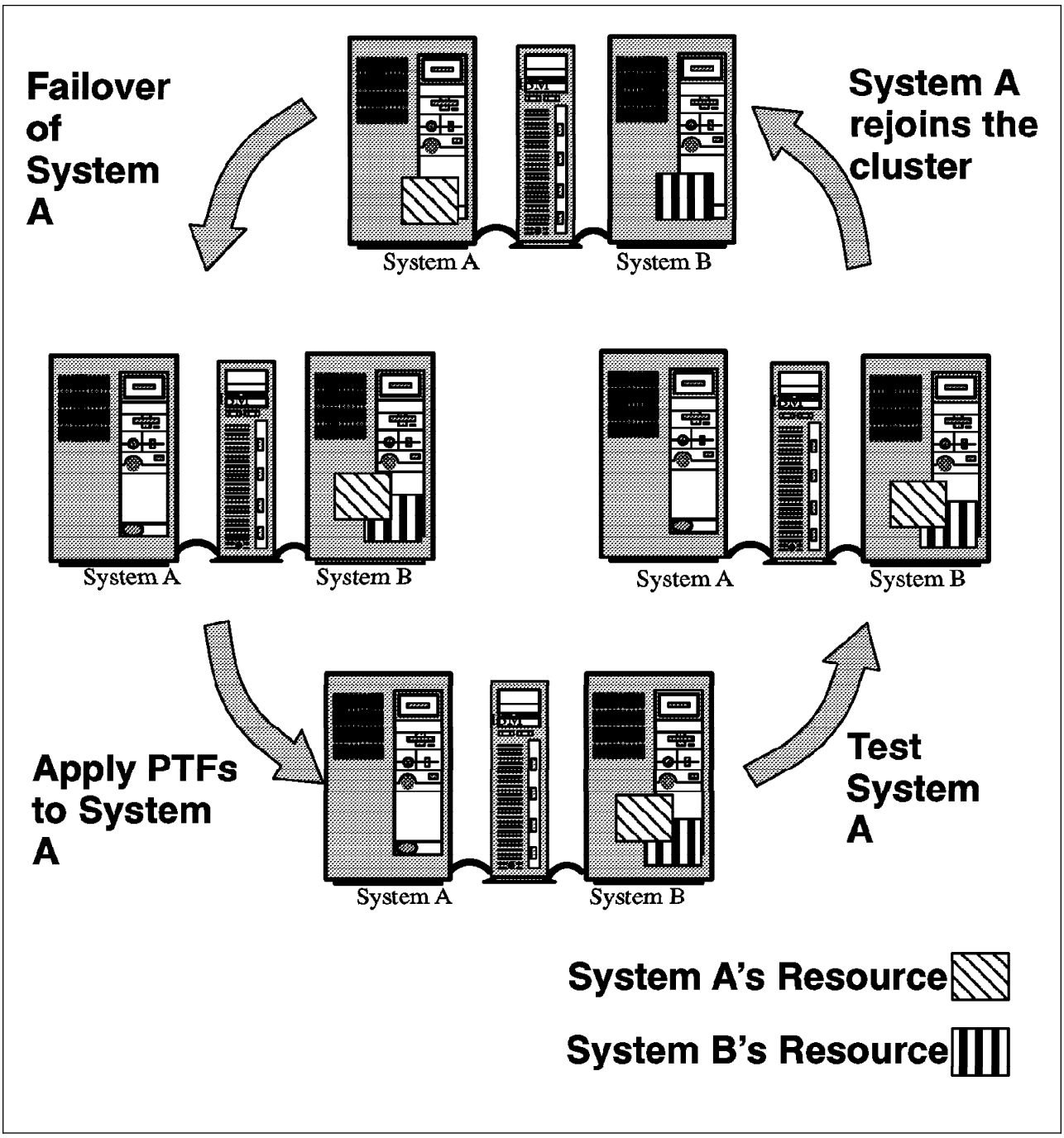


Figure 41. Applying Software Fixes, Part 1

You can then repeat the cycle for the other nodes. Refer to Figure 42 on page 178.

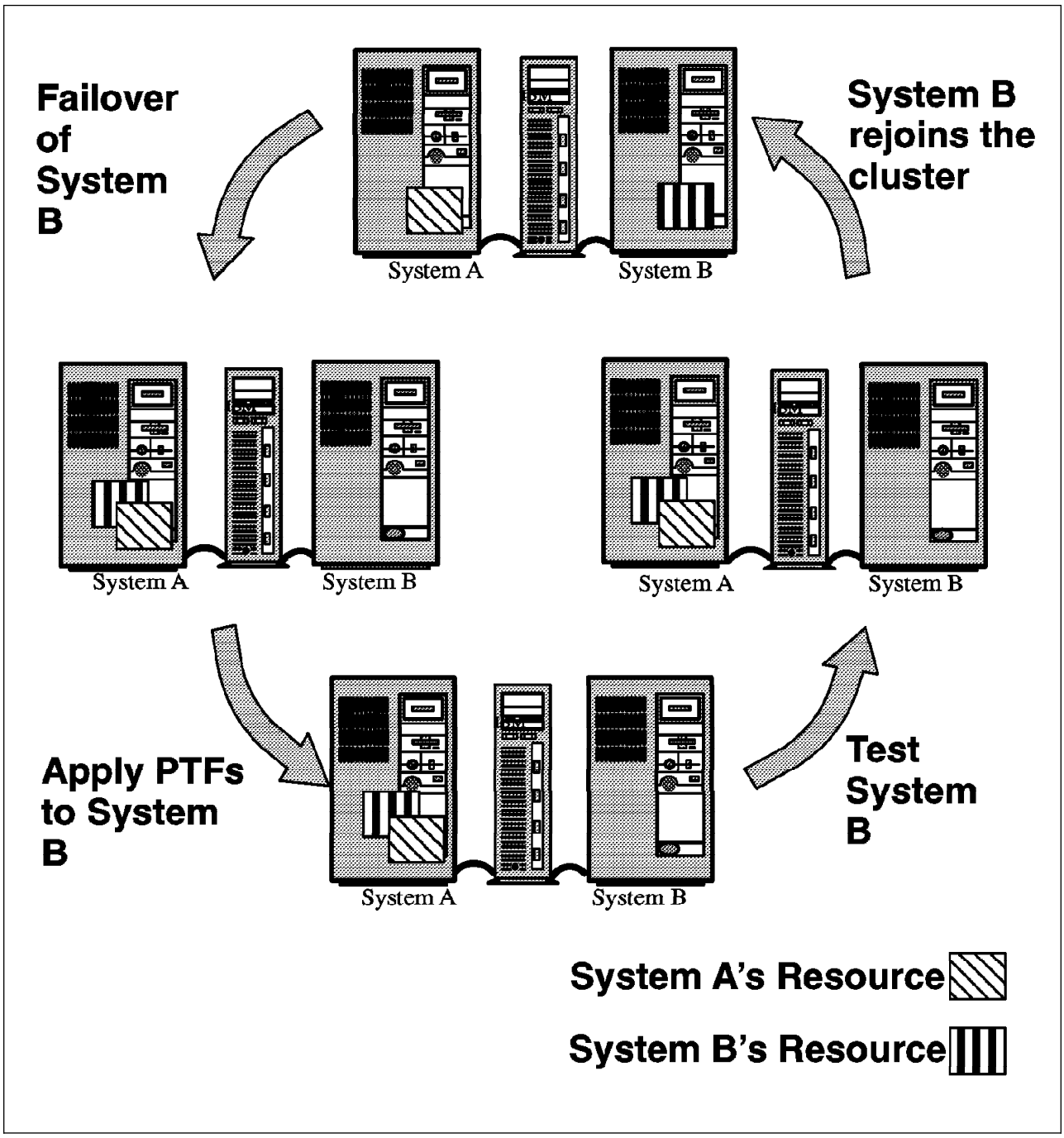


Figure 42. Applying Software Fixes, Part 2

Along with the normal rules for applying updates, the following general points should be observed for HACMP clusters:

- Cluster nodes should be kept at the same AIX maintenance levels wherever possible. This will, of course, not be true while the update is being applied, but should be true at all other times.
- Cluster nodes should be running the same HACMP maintenance levels. There will be incompatibilities between various maintenance levels of HACMP, so you must ensure that consistent levels are maintained across all cluster nodes. The cluster must be taken down to update the maintenance levels.

HACMP 4.1 for AIX does, however, make upgrading from HACMP/6000 Version 3.1 less disruptive to a customer's environment. Version compatibility, a new feature beginning in HACMP/6000 Version 3.1, allows you to upgrade an existing cluster running HACMP/6000 Version 3.1 to HACMP 4.1 for AIX without taking the entire cluster offline. During the upgrade process, individual nodes in the cluster can be removed from the cluster, upgraded one at a time, and then reintegrated into the cluster. Nodes running both HACMP/6000 Version 3.1 and HACMP 4.1 for AIX are able to coexist while the rest of the nodes are upgraded.

As with earlier versions, HACMP 4.1 for AIX provides conversion tools to assist existing customers in converting their configuration files from the previous version.

7.1.3 Cluster Maintenance - Do's and Dont's

- DO repair a failed system quickly - the cluster is running without a backup while the system is down.
- DO plan maintenance carefully.
- DO test thoroughly before reintegrating a failed machine.
- DO test thoroughly before putting a software fix into production.
- DO allow sufficient time for maintenance and testing.
- DO find out what a fix actually updates before applying it. Read the Memo_to_Users document first.
- DON'T commit a fix before testing it fully.
- DON'T apply fixes to AIX just because they are available. Subscribe to the old adage, “If it ain't broke....don't fix it!”
- DON'T keep running with intermittent hardware errors. Fix them.
- DON'T run `cfgmgr` on a running cluster node.

7.1.4 Requirement for Additional Filesystems

In the same way that the processing and resource demands of a single system environment will grow over time, so too will those of a cluster. Data requirements grow, applications grow in number and more users are added to systems. This generally leads to a requirement for increased storage capacity. When an HACMP cluster is commissioned, a defined set of resources are known to the Cluster Managers on each node. Adding more physical volumes or filesystems to a cluster, unfortunately, does not mean that these new resources are automatically discovered by the Cluster Managers. Consider the example of adding a new filesystem to a shared volume group. Assuming you have already created the filesystem, you would then add it to the list of owned, takeover or rotating resources for the various nodes in your Version 2.1 cluster, or to a resource group for a Version 3.1 cluster. At this point, you need to schedule some downtime to perform the following tasks:

1. Execute a graceful shutdown of the Cluster Managers on all cluster nodes.
2. Ensure that the shared volume group is varied off (it should be already, since the Cluster Manager will perform this task).

3. Export the shared volume group from any node which expects to take over this resource during failover, or any node on which this resource is configured for rotating standby.
4. Reimport the shared volume group on each of the nodes, one at a time.
5. Varyon the volume group and check that the new filesystem can be successfully mounted.
6. Unmount the filesystem and varyoff the shared volume group.
7. Restart the Cluster Manager on each of the cluster nodes.

There are plans to enhance HACMP so that configuration changes you make to resources and applications will be dynamically configurable across all cluster nodes. That is, in situations where system shutdown is not required, such as for adding a filesystem, it would not be required to bring the cluster down to propagate the changes across all nodes. But for now, this process requires the planning and scheduling of an appropriate change window.

7.1.5 Requirement for New Applications

If you wish to add a new application and have it protected and started by the Cluster Manager, some downtime must be scheduled to restart all Cluster Managers. This allows a new application server to be recognized. It is possible to install the application while the cluster is running. The decision to do this should be made with careful consideration of the effects it may have on any other subsystems that are running, and on system performance.

Also, it is important to understand the effects that enabling different services or subsystems will have on your cluster. For instance, if you enable Network Information Services (NIS) without changing the cluster to make it aware of the change, there is a very good chance that a false takeover will occur when the Cluster Manager attempts to reconfigure adapters at startup time.

7.1.6 Requirement for New Communications Connectivity

Again it is important to consider not only the effects of additional hardware on your cluster, but also the way you want the cluster to function with the additional hardware. If you wish HACMP to be aware of new network interfaces and indeed, to provide failover capability for the new equipment, then a change window must be scheduled for the necessary installation and configuration activity.

7.2 Mirroring the Root Volume Group (rootvg)

Of all the components used to build a computer system, physical disk devices are usually the most susceptible to failure. Because of this, disk mirroring is a frequently used technique for increasing system availability.

Filesystem mirroring and disk mirroring are easily configured using the Logical Volume Manager. However, conventional filesystem and disk mirroring offer no protection against failure of the operating system or against the failure of the disk from which the operating system normally boots.

Operating system failure does not always occur instantaneously, as demonstrated by a system that gradually loses access to operating system services. This

happens as code and data that was previously being accessed from memory gradually disappears in response to normal paging.

Normally, in an HACMP environment, it is not necessary to think about mirroring the root volume group, because the node failure facilities of HACMP can cover for the loss of any of the rootvg physical volumes. However, it is possible that a customer with business-critical applications will justify mirroring rootvg, to avoid the impact of the failover time involved in a node failure. In terms of maximizing availability, this technique is just as valid for increasing the availability of a cluster as it is for increasing single system availability.

The following procedure contains information that will enable you mirror the root volume group (rootvg), using the advanced functions of the Logical Volume Manager (LVM). It contains the steps required to:

- Mirror all the filesystems in rootvg
- Create an additional boot logical volume (blv)
- Create a secondary dump device
- Modify the bootlist to contain all boot devices

You may mirror logical volumes in the rootvg in the same way as any AIX logical volume may be mirrored, either once (two copies), or twice (three copies). The following procedure is designed for mirroring rootvg to a second disk only. Upon completion of these steps, your system will remain available if one of the disks in rootvg fails, and will even automatically boot from an alternate disk drive, if necessary.

This technique has been written using AIX Version 3.2.5. Be careful using these techniques and procedures, because as product and system interfaces change, so may the applicability and accuracy of this information.

7.2.1 Solution

Each disk belonging to a volume group will contain at least one Volume Group Descriptor Area (VGDA). The VGDA contains information about the layout of the logical volumes in the volume group. All VGDAs in a volume group will contain a copy of the same data. The number of VGDAs contained on a single disk varies depending on the number of disks in the volume group as follows:

Table 6. Organization of Volume Group Descriptor Areas on Physical Disks

Number of Disks	**Number of VGDAs**
One	Two
Two	Two on the first disk One on the second disk
More than two	One on each disk

LVM requires that a quorum, or majority of VGDAs, be accessible in order for a volume group to remain active, or to be varied on. In a two disk system, the failure of the first disk will cause two of the three VGDAs to be inaccessible, and therefore cause the varyoff of the entire volume group. If the second of the two disks fails, however, only one of the three VGDAs will be inaccessible, and the volume group will remain varied on, albeit with missing data. If you have three or more disks in

the volume group, a quorum will always remain upon failure of any one of the disks.

7.2.2 Procedure

The following procedure assumes that the operating system was installed on hdisk0, and is contained solely on that disk. The example assumes also that the mirror disk is called hdisk1 and is not currently allocated to a volume group. If the disk names are different on your system configuration, you can check them by entering the `lspv` command and making the appropriate substitutions for the disk names given here. You must be logged in as root to proceed.

1. Extend rootvg to include hdisk1:

```
# extendvg rootvg hdisk1
```

2. Even though rootvg is forced to vary on at boot time, if quorum is lost during operations, rootvg will be varied off. Since this would defeat the purpose of mirroring rootvg, turn off quorum checking to avoid operating system failure if a disk is lost:

```
# chvg -Qn rootvg
```

3. For the boot logical volume (blv) and secondary dump logical volume that will be created on hdisk1, edit the /etc/filesystems file, and add the following stanzas to the bottom of the file. It is important that the new stanzas be entered at the bottom of the file. Otherwise, if they appear before the entries for the primary boot and dump logical volumes, the filesystem helper will find these first and ignore the correct ones.

```
/blv:
        dev     = /dev/hd5x
        vol     = "spare"
        mount   = false
        check   = false
        free    = false
        vfs     = jfs
        log     = /dev/hd8

/mnt:
        dev     = /dev/hd7x
        vol     = "spare"
        mount   = false
        check   = false
        free    = false
        vfs     = jfs
        log     = /dev/hd8
```

   (Where x = a number from 1 to 9).

4. **Important:** Before mirroring or creating any other filesystems on hdisk1, create a new boot logical volume, and update it with a boot image. The boot logical volume must reside in the first two physical partitions of the disk:

```
# mklv -y hd5x -t boot -a e rootvg 2 hdisk1
# bosboot -a -l /dev/hd5x -d /dev/hdisk1
```

(Where x is the same number chosen in step 3).

Note

Sometimes, when updates to AIX are being performed, a new boot image is created and written to the boot logical volume, /dev/hd5. This process does not know about a second boot logical volume, and will not update it. Therefore the user must remember to perform a `bosboot -a` against the second boot logical volume each time updates are installed on the system.

5. Create a new system dump device on the second disk, and set that device to be the secondary dump device.

   Execute the command:

```
# sysdumpdev -e
```

   The output of `sysdumpdev` is the estimated value of the dump space required for a system dump to complete successfully (LED 0c0). The output value is in units of bytes, so you will need to divide it by 1,048,576 to convert to it megabytes, and then by 4 to convert to Physical Partitions (PPs). Round up the result to the next whole PP size. It is a good idea to then round up by an another one or two PPs for safety. Set the value of `n` in the following `mklv` command to the dump space size you require:

```
# mklv -y hd7x -t sysdump -a e rootvg n hdisk1
# sysdumpdev -P -s /dev/hd7x
```

   (Where x is the same number chosen in step 3 on page 182).

6. Create a mirrored copy of all filesystems, the filesystem log, and the paging space on hdisk1:

```
# mklvcopy hd6 2 hdisk1
# mklvcopy hd8 2 hdisk1
# mklvcopy hd2 2 hdisk1
# mklvcopy hd9var 2 hdisk1
# mklvcopy hd4 2 hdisk1
# mklvcopy hd1 2 hdisk1
# mklvcopy hd3 2 hdisk1
```

   In the example above, the logical volume names are specified in the order of performance priority with reference to where they should be placed on the disk. However, your own specific performance requirements may be used to modify this order if desired. If you have additional logical volumes in the root volume group, make mirror copies of them too.

7. Update non-volatile memory (NVRAM) with a new list of disks from which the system is allowed to boot.

```
# bootlist -m normal hdisk0 hdisk1
```

NVRAM contains a list of the devices from which the system will attempt to boot. The `-m normal` indicates the boot devices that should be used while the system unit key is in the normal position.

8. Synchronize (update) all of the new copies created:

```
# syncvg -v rootvg
```

Do not run any LVM commands while the `syncvg` command is processing. They may cause it to fail. If `syncvg` fails, it may usually be rerun without problems.

Once the steps above have been completed, either, but not both, of the two disks may fail without affecting the operation of the system. Normally, the system will attempt to boot from hdisk0 because it appears first in the bootlist. However, if hdisk0 is unavailable at boot time, the system will attempt to boot from hdisk1.

7.2.3 Testing Your Configuration

Having completed the procedure outlined above, you can test your work by failing hdisk0. The following steps describe a simple way of doing this.

1. Remove the power supply cable from its connector socket on the physical drive. Once you have removed the power, it is a good idea to access an application that is not in memory. Any application that resides in rootvg will do. InfoExplorer may be a good choice. This will prove that the mirror disk is providing application services.
2. Now shut down the system and observe that it boots successfully from hdisk1.

If the results of the tests are positive, then you have successfully mirrored the root volume group. All that remains to be done now is to reconnect the power cable to the physical disk drive. This must be done with the system powered down otherwise the system's power management will automatically cut power to all consumers in the system. Repower the system and observe that it boots correctly.

7.3 Quorum

The concept of quorum is a very important one to understand correctly if you are to manage an HACMP installation. Quorum is a policy that is enforced (now optionally) by the LVM, to ensure that the most recent, valid and up-to-date version of both the Volume Group Descriptor Area (VGDA) and Volume Group Status Area (VGSA) is always available and used. It does this by using set theory to enforce that, each time a volume group is varied on, it has available at least one of the VGDAs that was active up until the last time the volume group was successfully varied off.

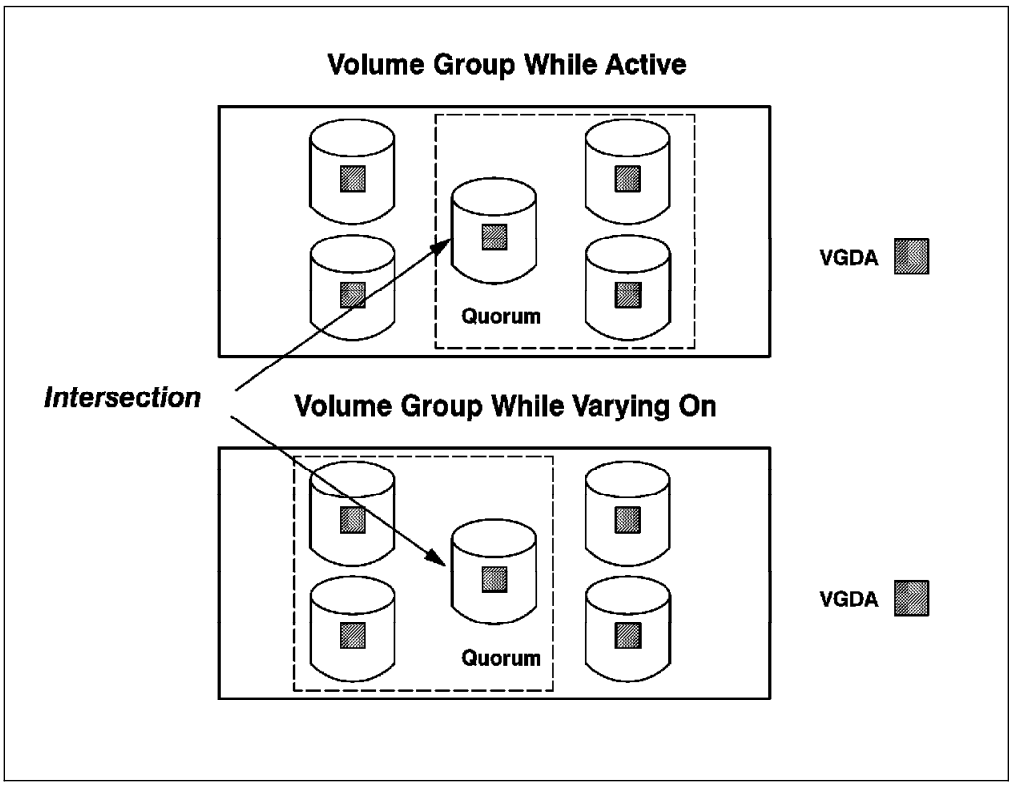


Figure 43. Volume Group Quorum

A quorum is defined as greater then 50% of the VGDAs in the volume group. If quorum protection is turned on for a volume group:

- Each time an update is made to the VGDA, the LVM checks to see that it is able to write at least a quorum (greater than 50%) of the VGDAs in the volume group. If it cannot do this, the volume group is immediately varied off.
- Each time a volume group is varied on, the LVM checks to see if it has a quorum of the VGDAs available. If there is a quorum available, the varyon continues successfully, even if some physical volumes are missing. If any physical volumes are missing, the system lists all PVs and their state before continuing. If a quorum is not available, the varyon fails.

By enforcing two rules:

- That the set of VGDAs available whenever the volume group is active must be greater than 50%
- That the set of VGDAs available whenever the volume group is varied on must be greater than 50%

the system ensures that these two sets intersect with at least one valid VGDA.

In this way, the system automatically ensures the integrity of the VGDAs it is using, and it is not left up to the user to do so.

7.3.1 Quorum in Shared Disk Configurations

For clusters using disk mirroring, quorum is an important concept. Consider, for example, a cluster that requires eight gigabytes of disk storage (four gigabytes mirrored). This requirement could be met with two 9333 or 9334 disk subsystems and two disk I/O adapters in each node. For data availability reasons, logical volumes would be mirrored across disk subsystems.

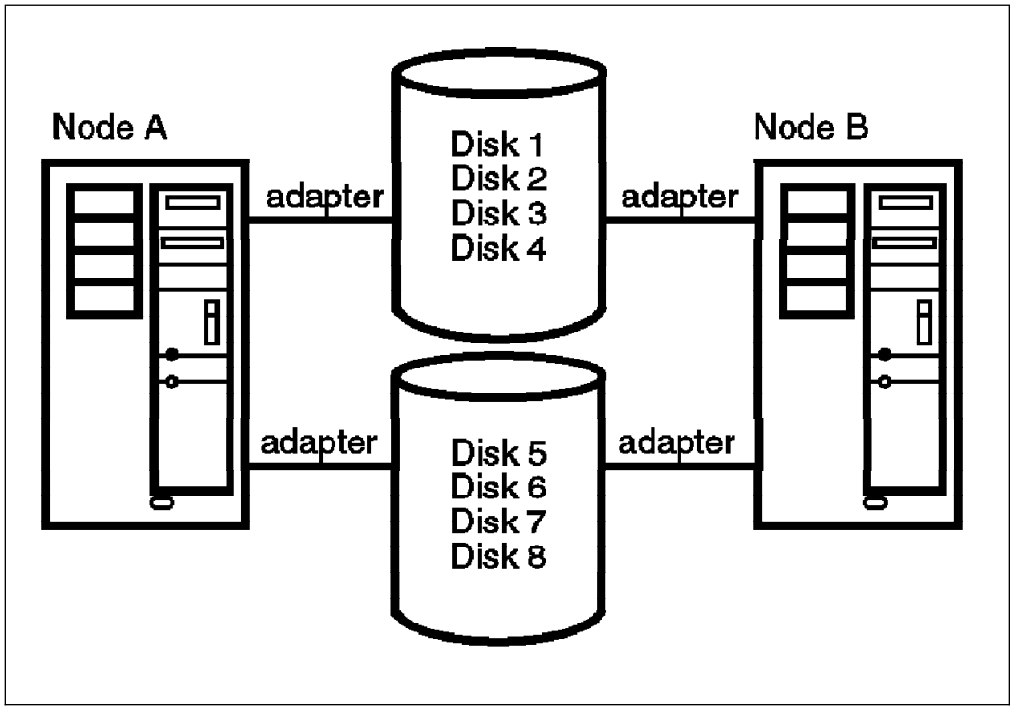


Figure 44. Quorum for Shared Disks in HACMP Configurations

With quorum enabled, the failure of a single disk I/O adapter, cable, or disk subsystem power supply would cause exactly half the disks to be inaccessible. Quorum would be lost and the volume group varied off, even though a copy of all mirrored logical volumes would still be available. The solution is to turn off quorum checking for the volume group. With quorum checking turned off, the volume group will remain active even if half the disks became inaccessible in the manner described above.

The trade-off is that, with quorum disabled, if *even one* physical volume is missing from the volume group, any varyon will fail. You will be required to use the `-f` or force option to varyon the volume group. This is like one last warning from the system that it cannot absolutely guarantee a valid VGDA. It is the user's responsibility to make this determination.

7.4 /etc/filesystems and the jfslog

AIX assigns a logical volume name to each logical volume it creates. Examples of system-defined logical volumes are /dev/lv00 and /dev/lv01. Within an HACMP cluster, the name of any shared logical volume must be unique in the cluster. The journaled filesystem log (jfslog) is a logical volume, and as such, also requires a unique name within the cluster. The reason is this. Say, for instance, you have two nodes: Node A and Node B. You have created some volume groups and logical volumes on the internal disks of Node B and allowed the system to assign

its default names, starting with /dev/vg00 and /dev/lv00. On Node A, you then create the shared volume group and logical volumes, and you allow the LVM to choose the names.

When you attempt to import the shared volume group onto Node B, you will find a device name conflict between logical volume names in the internal volume group of Node B and the logical volume names of the shared volume group. For example, you would already have an instance of /dev/loglv00 and /dev/lv00 on Node B. The LVM will prevent multiple instances of the same logical volume name by renaming the conflicting ones. When this is done, the same logical volumes will have different names on Node A and Node B. This is not only a bad management situation, but it also will certainly cause some of the HACMP scripts to fail.

If you have this problem, you can uniquely rename the logical volumes using SMIT. However, even though the corresponding stanzas in the /etc/filesystems file will be updated with the new logical volume names, they will not be updated with the new name of the jfslog. You will need to change each stanza manually using the `chfs` command. For example:

```
# chfs -a log=sharedlvlog /shared_filesystem_name
```

(Where `sharedlvlog` is the name of your log logical volume and `shared_filesystem_name` is the name of your filesystem).

If you have many filesystems, this becomes a tedious task which is prone to error. Therefore, before creating any filesystems on the shared disk resources, it is good practice to create the jfslog logical volume first. For example:

```
# smit mklv
```

Then select the volume group you require:

```
                              Add a Logical Volume

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

[TOP]                                                   [Entry Fields]
  Logical volume NAME                                  [sharedlvlog]
* VOLUME GROUP name                                     sharedvg
* Number of LOGICAL PARTITIONS                        [1]                        #
  PHYSICAL VOLUME names                                [hdisk2]                  +
  Logical volume TYPE                                  [jfslog]
  POSITION on physical volume                           midway                   +
  RANGE of physical volumes                             minimum                  +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                   [1]                        #
    to use for allocation
  Number of COPIES of each logical                      1                        +
    partition
  Mirror Write Consistency?                             yes                      +
  Allocate each logical partition copy                  yes                      +
    on a SEPARATE physical volume?
[MORE...9]

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

Remember to format the log logical volume with the following command:

```
# /usr/sbin/logform /dev/sharedlvlog
```

(Where `sharedlvlog` is the name of your log logical volume).

Answer yes to the prompt `Destroy /dev/sharedlvlog?`.

Then create the logical volumes and the filesystems on the logical volumes you have just defined. If you do this, the /etc/filesystems file will not require any modification. Additionally, when you import the shared volume group onto a takeover node, /etc/filesystems will be updated correctly with no naming conflicts.

7.5 Filesystem Helper

In some clusters, one or more nodes may display the following message while booting:

```
Filesystem Helper: 0506-519 Device Open failed
```

The message will relate to filesystems that are not configured to be automatically mounted at boot time. Typically, these are filesystems that reside in shared volume groups, which will not be activated (varyonvg) at boot time. The error message shown above is issued when the `fsck` command indicates that a filesystem could not be opened for checking. The Filesystem Helper will attempt to check a filesystem if the stanza in /etc/filesystems for that filesystem contains the entry `check = true`. The `check` attribute will be set to true if the filesystem was originally

created to be mounted automatically at boot time. Changing its characteristics later, so that the filesystem does not mount automatically at boot time, does not change the check entry in /etc/filesystems from `true` to `false`. On the other hand, if you create the filesystem with the automatic mount attribute set to `false`, then the check attribute will be set to `false` in /etc/filesystems. As long as the filesystem is not available for known reasons, as in our example with HACMP, this error message does not create a problem.

Nevertheless, you can prevent the message from appearing if it is causing concern. Do not change the attributes by editing /etc/filesystems directly. This will render the information stored in the VGDAs inconsistent. Set the attribute when the volume group is varied on with the following command:

```
# chfs -a check=false /shared_filesystem_name
```

(Where `shared_filesystem_name` is the name of your filesystem).

7.6 Phantom Disks

Phantom disks occasionally are seen in a SCSI environment. This is the situation where you see a second hdisk device created for the same real device, with the same SCSI ID. The original device then appears in a Defined state. This comes about when Logical Volume Manager activates (varies on) and creates a SCSI RESERVE on the disks in the particular volume group. The varyon, and the resulting SCSI RESERVE, creates an exclusive relationship between the initiator (the system) and the target (each disk in the volume group). If, for example, a single shared disk called hdisk1 is activated by System A, when System B boots and performs a SCSI inquiry, it cannot read the PVID from the disk. System B has already got hdisk1 defined, but since it cannot now read the PVID of the disk it now sees at the same SCSI ID, it cannot assume that it is the same device. Therefore it puts its original hdisk1 device into a Defined state, and configures a new device.

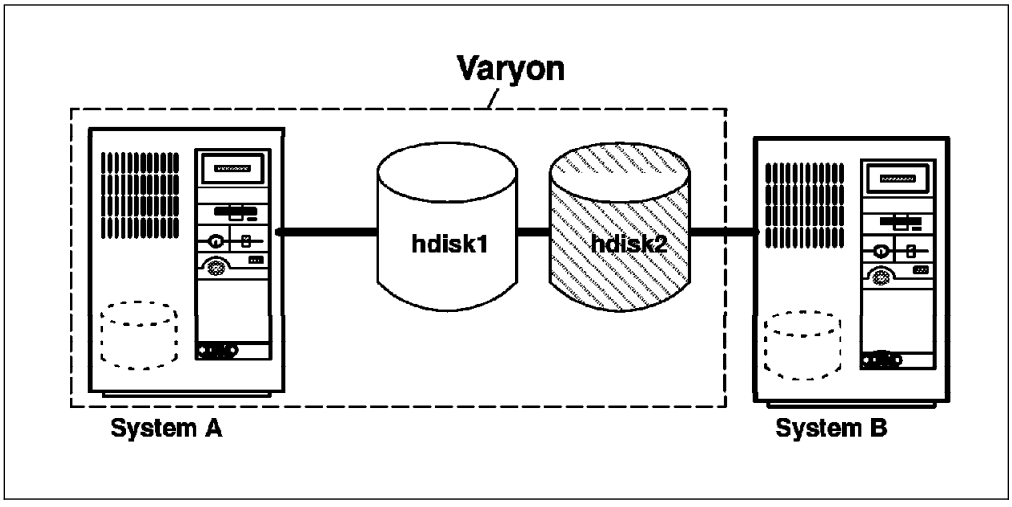


Figure 45. Phantom Disks

The appearance of phantom disks has occasionally caused confusion in customer environments and there have been instances where users have mistakenly deleted

the disks that are listed in a Defined state. This naturally has a disastrous effect on the process of failover because the takeover node is unable to acquire the disk resources it expects.

It is important to note that you do not need to take any special action if one or more nodes displays phantom disks. The HACMP event scripts are smart enough to correct this situation on failover by removing all phantom disk devices and restoring the real disk devices to a configured state.

7.7 /etc/inittab File

When you install and configure HACMP, the /etc/inittab file is modified during the process. In particular, some entries are added and some are modified depending on how you wish to configure the cluster nodes.

Clusters can occasionally run into problems related to the /etc/inittab file, even though its entries on all nodes may have been correct after the installation and configuration of the Cluster Manager. Here are some cases:

1. One or more nodes have undergone some configuration changes which have modified the /etc/inittab file, therefore preventing the Cluster Manager from starting one or more daemons correctly. Typically this can happen when you install an application that adds an entry to the inittab file. If the particular application binds an IP address at runtime, then it is important that the entry for it in the inittab file is set to a run level of `a`. This allows the daemon to be started by the Cluster Manager after all network interfaces have been configured rather than being started at system boot time. Furthermore, for any node configured for IP address takeover, it is important that the following line appears at the end of the inittab file.

   ```
   clinit:a:wait:touch /usr/sbin/cluster/.telinit # HACMP6000 This must be last
   entry in inittab!
   ```

   When you configure a node for IP address takeover, the above line will be written into the inittab file. During its startup, the Cluster Manager runs the event script `acquire_service_addr`, which configures network adapters and starts any daemons set to run level `a`, by executing the command `telinit a`. The Cluster Manager needs to know that TCP/IP and all daemons that bind a network address have been started before it activates any volume groups or starts any user applications. To do this, the script checks for the existence of the file `/usr/sbin/cluster/.telinit`, and will not exit until it has been created by the `touch` command. If the `clinit` entry is not the last entry in the /etc/inittab file, the Cluster Manager may start user applications before all IP-dependent applications have started. This may lead to unpredictable behavior of the application environment. After installing a new application, or when performing problem determination work, it is good practice to check that the above-mentioned entry is still the last line in the /etc/inittab file.

2. A problem has occurred with one or more of the IP-related services and remedial action has modified one or more of the run-levels of subsystem entries in the /etc/inittab file. For example, this may happen where the Network File System is enabled. A problem has occurred and NFS needs to be stopped and restarted using SMIT. Inadvertent misuse of the startup options `restart` or `both`, submitted to the `rmnfs` or `mknfs` commands, can erase the `rcnfs` entry

from the /etc/inittab file or change its run level back to 2. This must be set to a to avoid problems with NFS.

7.8 Permissions on the /tmp Directory

The permissions and ownerships for /tmp created by AIX at install time are the following:

```
drwxrwxrwt   7 bin      bin         1536 Jun 26 13:49 tmp
```

When you install HACMP/6000 Version 2.1 or HACMP/6000 Version 3.1, the permissions of the /tmp directory will be changed:

```
drwxr-xr-x   7 root     system      1536 Jun 26 15:01 tmp
```

Write permission for group and other are disabled and uid and gid are changed to uid=root and gid=system, respectively.

The altered permissions cause many problems for non-root users. Users and applications that require /tmp for temporary work space, may be denied access because they do not have permission. APAR IX41422 was opened for this problem in 1994, and was subsequently closed with no intention to build a PTF, leaving the responsibility to the user to ensure the correct permissions and ownerships. After the installation of HACMP, check or change the permissions on /tmp to ensure that they are consistent with your applications' requirements.

7.9 ARP Cache

The ARP (Address Resolution Protocol) cache contains entries which map IP addresses of network interfaces to their corresponding hardware addresses. ARP caches present a potential problem in HACMP configurations, during adapter swapping or IP address takeover scenarios.

Once an entry is in the ARP cache, systems use the hardware address rather than an IP address to communicate to interfaces on the network. When an IP address takeover or adapter swap occurs, the IP address becomes associated with a new adapter's hardware address. However, the new address mapping is not automatically updated to the ARP cache of a client machine.

There are various solutions available to the ARP cache problem. The easiest one is to use a feature called *hardware address swapping*. This feature, which was first introduced in HACMP/6000 Version 2.1, allows the hardware address of an adapter to be taken over along with the IP address. Using this technique means that no change to the ARP cache of a client machine is necessary. The disadvantage is that the changing of hardware addresses increases the time required to do adapter swapping and IP address takeover.

Other methods all involve some form of flushing the ARP caches on client systems and routers. The optional client component of HACMP, *clinfo*, can be used for this.

For clients running clinfo, the `/usr/sbin/cluster/etc/clinfo.rc` script, which is executed whenever a cluster event occurs, updates the system's ARP cache.

On clients that are not running clinfo, the product documentation suggests two methods to correct a non-clinfo client's ARP cache by pinging it from a cluster node. The Methods are described below:

1. To notify a client, you must add the hostname or IP address of the client host to the PING_CLIENT_LIST variable in the `/usr/sbin/cluster/etc/clinfo.rc` script on each of the cluster nodes. After this is done, whenever a cluster event occurs, `/usr/sbin/cluster/etc/clinfo.rc` executes the following command for each host specified in PING_CLIENT_LIST:

   ```
   /etc/ping hostname 1024 2
   ```

2. The second method involves writing a shell script for each cluster node, and using the *post-event customization* facility to call it after the *acquire_takeover_addr* event:

   a. Ping the client system you wish to update:

      ```
      ping -c1 client_IP_address
      ```

   b. Add a route to the client system you wish to update using the relocated interface:

      ```
      route add client_IP_address failed_node_service_IP_address 0
      ```

   c. Ping the client system again:

      ```
      ping -c1 client_IP_address
      ```

   d. Delete the route you previously added:

      ```
      route delete client_IP_address failed_node_service_IP_address
      ```

   This procedure forces the ping to go out over the interface that has taken over the IP address for the failed system. The address resolution triggered by this ping will provide the client with the new hardware address now asssociated with the service IP address.

We will repeat that neither of these procedures are necessary if either of the following are true:

- You have implemented hardware address takeover in your configuration.
- Your client systems are running the clinfo component of HACMP.

For more information on the Cluster Information Program (clinfo), see the *HACMP Installation Guide Version 3.1* Chapter 13, Setting Up the Cluster Information Program.

7.10 Synchronizing Time Between Cluster Nodes

Time should be synchronized between nodes in the cluster. If this is not done, you have the potential for problems with shared data after a failover. For example, if a database resides in a shared volume group, and time is not being synchronized between cluster nodes, when a node fails, and the disks join the other node, the timestamps on database logs will be different from the time on the new system. If the takeover node has a time ahead of the failed node, the replay logs should work correctly. If the time on the takeover node is behind that of the failed node, you can have problems, ranging from errors on the log replays, to lost data. You could

choose to have one of the cluster nodes as the time server (internal) or have a facility outside the cluster supply the time (external).

Time serving and the *timed* daemon are covered in more detail in the redbook *HACMP Customization Examples* Chapter 8, Synchronizing Time Clocks in Your Cluster.

7.11 Tips on Writing Application Server and Event Scripts

You can provide start and stop scripts to HACMP for application servers such as database backends. While writing these scripts, you should keep the following things in mind:

- Where applicable, you must check for abnormal termination of the application server in your start script.
- You should protect against data or transaction loss in the stop script for a server.

As mentioned earlier, you should try to avoid making any changes to the default event scripts provided by HACMP. You should rather try to implement all required customization in the pre and post-event scripts.

While writing both event as well as application server scripts, you should keep the following points in mind:

- Fork a shell for your script to execute in by having #!/bin/sh as the first line in your script.
- Set the PATH variable at the beginning of every script to ensure that all the commands you are using in your script are visible.
- Check for correct usage and return a message indicating the correct usage if the script is executed incorrectly.
- Check if the cluster has been set for verbose logging of script commands, and accordingly run your script with or without the verbose option. Verbose logging is the default setting for HACMP, meaning that the maximum amount of information from the event scripts being run is logged into the /tmp/hacmp.out file. You can check your current setting on any node by entering `smit cm_run_time.select` and checking the setting of the `Debug Level` field. If it is set to `High`, you have verbose logging enabled.
- Use exit status codes in your scripts.
- Document your scripts as extensively as possible.
- Handle bad return codes from commands that your script calls.

Be sure to test all your scripts extensively before incorporating them into HACMP.

7.11.1 Problem for HACMP/6000 Version 3.1 Users Before PTF U438726

The following problem has been fixed by HACMP/6000 Version 3.1 PTF U438726, but was found in our testing, and will be experienced by any users before they have applied this PTF. If you have already applied this PTF, you can disregard this section in total.

If you declare a pre or post-event script, recovery script, or notify method for any event, and later decide to remove the script from the HACMP configuration, you will face a problem.

For example, assume that you have provided a pre-event command to HACMP for the node_up_local_complete event. Now, after further customization, you realize that this script is not required. In order to remove the script from ODM, you use the following SMIT screen:

```
                          Change/Show Cluster Events

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

                                                        [Entry Fields]
  Node ID                                              dave1

  Event Name                                           node_up_local_complete

  Description                                          Script run after the n>

  Event Command                                       [/usr/sbin/cluster/even>

  Notify Command                                      []
  Pre-event Command                                   []
  Post-event Command                                  [/tmp/post_node_up_loc_comp.
sh]
  Event Recovery Command                              []
  Recovery Counter                                    [0]                      #



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

In this screen, if you delete the entry for **Post-event Command** and execute, you will get the following error message:

```
                                COMMAND STATUS

Command: failed        stdout: no            stderr: yes

Before command completion, additional instructions may appear below.

/usr/sbin/cluster/events/cmd/clchev: option requires an argument -- a
Usage: /usr/sbin/cluster/events/cmd/clchev [-i nodeName] [-e new-eventname] [-d
description] [-f NLSCatalogfile] [-m CatalogMessageNo] [-t CatalogSetNo] [-s scr
ipt] [-n script] [-b script] [-a script] [-r script] [-c count] [-o odmdir] Even
tname
Or:
Usage: /usr/sbin/cluster/events/cmd/clchev [-i nodeName] [-O Eventname] [-e new-
eventname] [-d description] [-f NLSCatalogfile] [-m CatalogMessageNo] [-t Catalo
gSetNo]  [-s script] [-n script] [-b script] [-a script] [-r script] [-c count]
[-o odmdir]




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

In order to work around this problem, remove all commands except `exit 0` from the script that is no longer required. Do not attempt to remove it from the HACMP configuration.

7.12 Recovery From Event Script Failure

There are two ways in which you can recover from an event script failure. You can either write a failure recovery script and declare it to the Cluster Manager or you can manually bring the cluster back to a stable state.

If you have not specified a recovery script, or if your recovery script fails, you will most probably see a message on your console to the effect that the cluster has been in reconfiguration mode for too long. At this point, you can browse the /tmp/hacmp.out file and, if the scripts were set to run in verbose mode, find the script that failed and the exact point at which the failure occurred.

In many cases, you can manually correct whatever caused the script to fail. Then you can ask HACMP to continue event processing from where it had left off, by giving the following command:

```
# /usr/sbin/cluster/tools/cld_debug -t clruncmd -h'hadave1_boot'
```

Alternatively, you can use the SMIT fast path `clrecover.dialog`, provide the IP label of any adapter on the node on which the event script has failed, and let SMIT execute the above command.

7.13 AIX Error Notification

One of the problems with setting up error notification in HACMP has been that the Select Error Label field in the Error Notification menu in SMIT was truncated to 10 characters in the ODM and the display area of the SMIT screen. This problem is corrected by PTF U435385 for HACMP/6000 Version 2.1 and all superseding PTFs for that version.

At the time of publication of this document, this problem had reappeared in HACMP/6000 Version 3.1, and had not been fixed yet. You should be sure to test your error notification methods thoroughly to see if this problem exists at your level of the product.

The workaround for this problem is to explicitly type the entire error label into the Select Error Label field, rather than to select an item from the list presented by the F4 key. The list of error labels can be found in the header file /usr/include/sys/errids.h.

7.14 7135 RAIDiant Disk Array

The software required to configure a 7135 RAIDiant Array Model 110 in a highly available manner is the 7135 Disk Array Manager with the `scarray` device driver programs. For a detailed description of how to use the 7135 Disk Array Manager to configure a 7135 disk array, please refer to *7135 RAIDiant Array Installation and Service Guide*.

There are several points that you should keep in mind while configuring a 7135 in an HACMP cluster. These are:

- Load balancing across controllers should be disabled in a dual-active controller environment.

  The 7135 Disk Array Manager allows you to configure a 7135 with two controllers such that data traffic to and from the disks in the array is split across the two controllers. This is called a *dual-active* configuration.

  You can configure the 7135 such that if traffic over one of the controllers is heavier, control of some of the highly accessed disks is shifted to the other controller. This capability is called load balancing.

  In an HACMP cluster, where more than one node is accessing the same disk array, if the Disk Array Manager on one node switches control of some disks from one controller to the other, this switch is not known to the other nodes. Consequently, the other nodes are not able to access these disks. Therefore, load balancing should be disabled on all nodes having access to a 7135 in a cluster.

- Array configuration should be done from only one node at any given time.

  In an HACMP cluster, a 7135 is connected to more than one node on the same SCSI bus. When a node configures a 7135, it resets the connections of any other node that was accessing it on the same bus. This may lead to errors and data loss.

  Since a node would exchange configuration related information with the 7135 at system startup, you should boot cluster nodes attached to a shared 7135 one at a time.

- Independent of the total number of disks in a 7135, the maximum number of Logical UNits (LUNs) that can be created on it is eight.

7.15 Resource Group Organization

For easier management of resources and resource groups, you should group service IP addresses separately from disk and application resources. Also, you should group an application server together with the disk resource that that application accesses. This is best illustrated through an example.

Let us consider a two node cluster in a mutual takeover configuration. There are two cascading resource groups, A and B, each containing an IP address, two volume groups, their filesystems, and two application servers. One node has a priority of 1 on resource group A and a priority of 2 on resource group B, while the other node has a priority of 1 on resource group B and a priority of 2 on resource group A.

After working on this cluster for some time, you realize that the workload is not balanced between the nodes. You also realize that the problem can be solved if you swap one application from each node over to the other node.

In order to do this, you will have to change the resource groups. If, on the other hand, you had separate resource groups for each IP address and for each set of application server and associated storage, all you would need to do is change the node priorities for two of these resource groups.

The difference in effort between changing a resource group and changing the node priorities may not be appreciable in the above example. However, if you extend the same exercise to a cluster containing three or more nodes, or ten or more applications, each with its own storage, you will realize that isolation in the design of resource groups gives you a great deal of flexibility for administration and for introducing changes to the cluster.

7.16 Hubs in an HACMP Cluster

Production environments with a large number of users accessing services over LANs often use hubs for easier management of networks. If all cluster nodes are connected to one hub, the hub becomes a single point of failure for the cluster.

Figure 46 on page 198 shows how you can connect two hubs in a cluster so that the failure of any one of the hubs does not result in the failure of the entire network. In this figure, if any one of the hubs fail, the cluster nodes lose their connections to either the service adapters or the standby adapters, but never to both adapters on the same node.

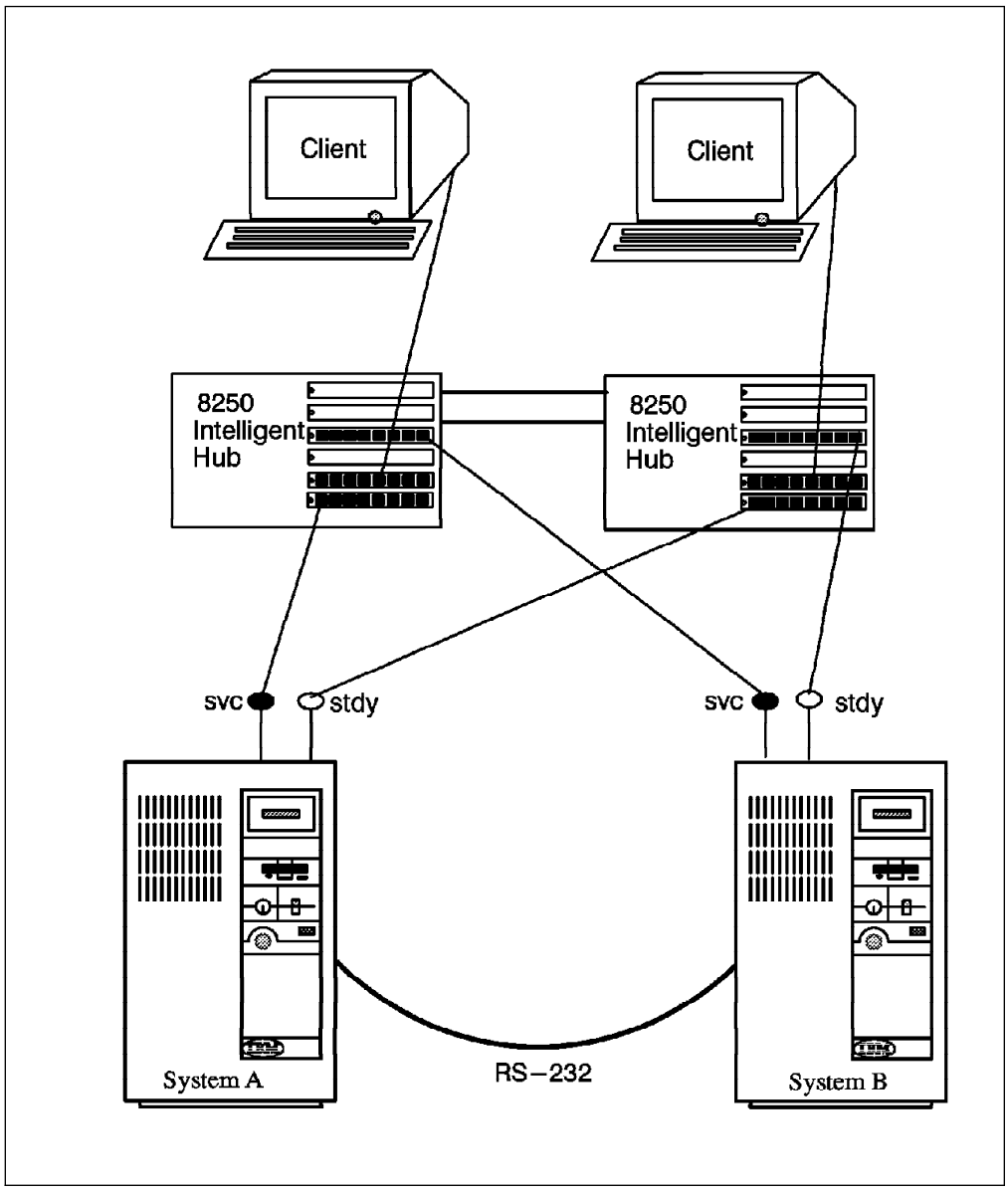


Figure 46. Protecting Your Network against Hub Failure

If the hub to which the service adapters are connected fails, the IP addresses of the service and standby adapters get swapped. If the hub to which the standby adapters are connected fails, a message is displayed on the consoles of the cluster nodes informing you about the loss of the standby adapters.

Appendix A. HACMP Software Components

This appendix gives a detailed description of the different software components and corresponding daemons of HACMP that provide high availability, client support, and concurrent access.

These components are:

- Cluster Manager (Cluster Manager daemon)
- Cluster SMUX (SNMP MUltipleXer) Peer Service (clsmuxpd daemon)
- Cluster Information Service (clinfo daemon)
- Cluster Lock Manager (cllockd daemon)

The Cluster Manager and clsmuxpd daemons are started every time HACMP cluster services are started. HACMP cluster services can be started either automatically at system startup or explicitly through SMIT. The clinfo and cllockd daemons are started only if you request the start of the Cluster Information Service and the Cluster Lock Service respectively.

The only HACMP daemon that can run on a client machine is the clinfo daemon. On a client, the clinfo daemon is started automatically at system startup, on the next reboot after the client portion of HACMP has been installed.

All the HACMP daemons are defined to the AIX System Resource Controller (SRC). Therefore, all of them are started, stopped, and monitored by the SRC. The Cluster Manager, clsmuxpd, and the clinfo daemons are all part of an SRC group called cluster. The cllockd daemon is in a separate group by itself called lock.

A.1 Cluster Manager

As described in “HACMP Software” on page 36, the Cluster Manager runs on all cluster nodes. It monitors local hardware and software subsystems, tracks the state of other nodes in the cluster, and executes event scripts in response to cluster events.

The Cluster Manager has four functional components. These are:

- Cluster Controller
- Event Manager
- Network Interface Layer
- Network Interface Modules (NIMs)

While the Cluster Controller, Event Manager, and Network Interface Layer are all part of the Cluster Manager executable, the Network Interface Modules are separate executables (one for each type of network). Figure 47 on page 200 shows the structure of the Cluster Manager and the connectivity between neighboring peer nodes in a cluster.

Aside from the different parts of the Cluster Manager, the figure also shows that the Cluster Manager daemon is controlled by the Subsystem Resource Controller

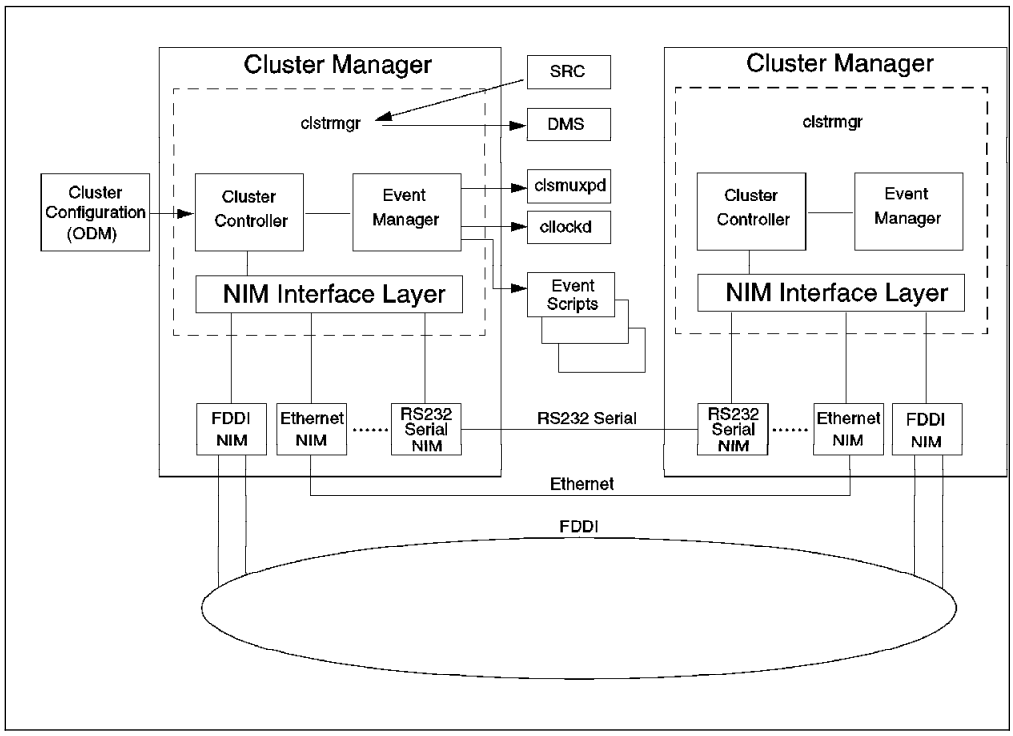


Figure 47. Cluster Manager Structure and Peer Connectivity in an HACMP Cluster

(SRC) and that the Cluster Manager daemon controls the Deadman Switch (DMS). Section 6.3, “The Deadman Switch” on page 160 gives a detailed description of the DMS and how the Cluster Manager daemon controls it.

A.1.1 Cluster Controller

The Cluster Controller is the brains of the Cluster Manager. On startup of cluster services, the Cluster Controller retrieves information about the cluster configuration from the Global ODM (godm) and determines which nodes in the cluster are its neighbors. It does this by sorting through an ordered list of adapter labels in such a way as to avoid a node from sending KAs to itself in normal circumstances. Simplistically speaking, adjacent nodenames in an alphabetical list of nodenames are considered a node's neighbors. The Cluster Manager then exchanges keepalives (KAs) only with its neighbor nodes.

Note: This is a change from previous versions of HACMP, where all nodes exchanged keepalives with all other nodes.

The Cluster Controller receives information about the cluster from the NIMs by way of the Network Interface Layer and uses this information to maintain a current view of the cluster topology (node membership and the state of networks).

When the status of the cluster changes, the Cluster Controller queues cluster events to the Event Manager. It can also escalate events. For example, it can detect the failure of multiple networks in the queue and translate this into a response for a node failure. As nodes join or leave the cluster, the node membership changes and the Cluster Controller accordingly adjusts the neighbor node it is exchanging keepalive packets with, if necessary.

The Cluster Controller registers a failure when it starts missing KA packets that it expects. Three missed KAs are called a *minor failure* and four minor failures are

considered a *major failure*. It translates four missed KAs over one adapter as a failure for that adapter and swaps its address with the standby adapter. If there are no KAs being received on all adapters from a network, the Cluster Controller waits until it has missed a total of twelve KAs before it assumes a network failure. It then enters a stabilization period of ten KAs, to ensure that KAs are still being received over other networks, and that it is not a node failure. Once it is sure that it is a network failure, it queues the corresponding event on the event queue.

A node failure event is initiated when a node misses twelve KAs on the fastest network (network with the highest KA rate on its interfaces). There is a stabilization period of ten KAs during which it sends messages to its missing neighbor over the other networks to check if it is a network failure. If these messages do not get acknowledged, the Cluster Controller concludes that its neighbor has left the cluster and queues a node down event on the event queue.

The Cluster Controller also handles a condition named *cluster partitioning*. A Cluster gets partitioned when one or more active nodes in a cluster lose all communications with each other. This results in two or more isolated partitions, each containing two or more cluster nodes, within one cluster.

It is possible that a partitioned cluster might not be detected, depending on where the communications break were, and which cluster nodes were trading keepalives as neighbors. For this reason, the Cluster Controller also sends out a regular *slow heartbeat message* to each other node, one at a time, every two seconds. It also expects to receive slow heartbeat messages back from each of the other nodes. If a slow heartbeat message is not received back from each of the other nodes within three complete slow heartbeat cycles, a failure event is generated.

If the failure in communication is an intermittent one, and the Cluster Managers on the cluster nodes are later able to communicate with each other after having been isolated, the nodes in the biggest partition send reset packets to the nodes in the other partitions to bring them down. This is done because the events to reflect the other nodes as down would already have been generated, with their accompanying resource takeover actions. If the nodes were to continue to stay up, they would resist the takeover of their resources, and would cause these events to fail. If the size of each partition is the same, the partition having the node with the lowest node name (first on an alphabetical list of node names) survives.

A.1.2 Network Interface Layer

The Network Interface Layer is a liaison between the Cluster Controller and the NIMs. It initiates NIMs (one per logical network) as required by the configuration of the cluster. It then monitors the NIMs and restarts them if they hang or exit.

The Network Interface Layer determines the rate at which the NIMs send out KAs and the number of KAs that a NIM must miss to register a failure. The actual KA rates differ based on network type and cluster configuration. KA rates and how you can configure them is discussed in Section 6.5.2, “HACMP/6000 Version 3.1” on page 168. The Network Interface Layer also sends each NIM a list of addresses that it has to communicate with. Multicasting (sending messages to multiple specific addresses as opposed to broadcasting messages to all addresses) is also managed by the Network Interface Layer.

A.1.3 Network Interface Modules

There is a unique Network Interface Module executable for each type of network supported by HACMP. A different instance of a NIM is started by the Network Interface Layer for each logical network defined to an HACMP cluster. NIMs communicate with the Network Interface Layer over pipes.

The NIMs associated with each type of network are:

Ethernet	nim_ether
RS-232 Serial	nim_sl
SOCC	nim_SOCC
FDDI	nim_fddi
Token Ring	nim_tok
Target Mode SCSI	nim_tms
SLIP	nim_slip
Generic IP	nim_genip

NIMs are responsible for sending and receiving KAs and messages, for acknowledging the delivery of messages, and for detecting network failures. They do not make any decisions or initiate any actions based on the information that they receive, but rather just pass the results to the Network Interface Layer and on to the Cluster Controller for the decisions to be made.

A.1.3.1 Packet Types

Keepalive (KA) packets are UDP-based. They are exchanged only between NIMs on neighboring nodes over all configured interfaces that are common to the two nodes. How a node determines who it's neighbors are has been described earlier in Section A.1.1, “Cluster Controller” on page 200. Among other things, a KA packet contains the version of the NIM that is sending it, the process IDs of the sending and receiving Cluster Managers, and the nodenames of the sending and receiving nodes.

Message packets, on the other hand, are exchanged between any two nodes in a cluster only when necessary. Some of the conditions under which message packets are exchanged are:

- Nodes or networks joining the cluster
- Detection of events
- Synchronization of Cluster Managers on different cluster nodes

The contents of a message packet are:

- Version number of the Cluster Manager
- Nodenames of the sending and receiving nodes
- Retry and hop count (number of nodes to pass through to reach the destination node from the source node)
- Broadcast flag
- Text of the message

A.1.4 Event Manager

The Event Manager pulls events, that have been queued by the Cluster Controller, off the event queue. It first sets all necessary environment variables and then forks a process to execute the script(s) associated with an event. Finally, it directly informs `clsmuxpd` and `cllockd` of the event.

A.2 Cluster SMUX Peer and Cluster Information Services

The Cluster SMUX Peer daemon is an SNMP sub-agent that uses an extension MIB (Management Information Base) table specific to the HACMP environment. When the Cluster Manager is started on a cluster node, the Cluster SMUX Peer daemon (clsmuxpd) is also started automatically. The clsmuxpd daemon first registers itself with the SNMP agent (snmpd) on the local node and then contacts the local Cluster Manager for information about the cluster.

Once clsmuxpd has got the cluster information, it maintains an updated topology map of the cluster in the HACMP MIB, as it receives information about changes in cluster status from the Event Manager.

The Cluster Information Service allows applications on cluster nodes and client machines to receive information about the state of the cluster, using C and C++ APIs. It also executes the `clinfo.rc` script whenever it detects that a node, network, or cluster event has occurred. This script flushes the ARP cache of the system on which it is running, and can be customized to take other actions as required.

The Cluster Information daemon (clinfo) is an SNMP monitor which receives information about a cluster from a clsmuxpd daemon running on any of the cluster nodes. Please note that clinfo does not communicate directly with clsmuxpd. Any exchange of information is through the snmpd daemon.

When clinfo starts running, it reads its configuration file, named /usr/sbin/cluster/clhosts. This file lists the IP addresses or IP labels for the service and boot interfaces of all the nodes in the cluster. Starting with the first address in this file, clinfo tries to connect with an active clsmuxpd process on a cluster node.

Once clinfo has established contact with a clsmuxpd daemon, it queries it at regular intervals for updated cluster information. The default interval is fifteen seconds, which can be customized as desired.

Alternately, clinfo can be run with an *asynchronous trap option*. This makes clsmuxpd send a trap message to clinfo as soon as a cluster event occurs, instead of waiting for the next query.

Compatibility

It must be noted, if you run clinfo with the asynchronous trap option, you cannot also run network monitoring programs, such as Netview for AIX, on the same system. This is because both use the same UDP port number, 162, to receive traps. If clinfo is running in asynchronous trap mode first, Netview for AIX cannot be successfully started on the system.

The clinfo daemon remains bound to the first clsmuxpd daemon that it finds in the cluster as long as it can. If the node that clinfo is bound to fails, it uses its internal cluster topology map to establish contact with the clsmuxpd daemon on another node.

A.3 Cluster Lock Manager

The Cluster Lock Manager provides two different lock models:

- UNIX System V locking model
- CLM locking model

There are two sets of Cluster Lock Manager APIs available to application developers, one for each locking model. The same application can use both types of locks on the same data. One lock model is not aware of the locks being held by the other. Therefore, if you are using both lock models in an application, you should ensure that each model is locking a different region of data. The Cluster Lock Manager implements advisory locking, so it does not ensure data integrity by itself. The application must be written to ensure that all access to the data is done through the Cluster Lock Manager.

A.3.1 UNIX System V Locking Model

The UNIX System V locking model works with lock resources, which are first registered, and later linked to specific regions to be locked. A lock resource can be any entity, for instance a file, a database, or a data structure. When an application registers a lock resource, a resource handle is returned to it. Subsequently, specific regions within it can be locked and unlocked (using offset and length parameters).

A UNIX System V lock can be one of two types:

- Shared

  This is a *read* type of lock. Multiple processes can request and be granted shared locks on the same region at the same time. An exclusive lock cannot be granted on a region that has any shared locks held.

- Exclusive

  This is a *write* type of lock. It prevents any other process from accessing its specified region, in any way.

A UNIX System V lock request can be in one of two states:

- Granted

  The requested region is available in the requested mode.

- Blocked

  This happens when any part of the requested region has an exclusive lock active on it.

In this model, a new shared lock can overlap an existing shared lock. The entire combined region of the two overlapping locks is now considered to be locked as a single unit. The subsequent release of either of the two locks will result in the entire region getting unlocked.

A.3.2 CLM Locking Model

The CLM locking model works on named lock resources, which can be anything as defined by the requesting application. A lock resource is created when a lock on that resource is requested for the first time, and it exists until the last lock request on it is released. A lock resource consists of a resource name, a Lock Value Block, and three lock queues (grant, convert, and wait).

In the CLM model, there are six locking modes that increasingly restrict access to a resource. These are:

- Null (NL) - no access, just interest
- Concurrent Read (CR) - read while others read or write
- Concurrent write (CW) - write while others read or write
- Protected Read (PR) - read while others are only permitted to read
- Protected Write (PW) - write while others are only permitted to read
- Exclusive (EX) - write while no access is granted to others

Locks can be promoted or demoted by conversion from one mode to another.

An application can request a lock with a specific mode on a lock resource. Many such locks can be requested for one lock resource. Multiple lock requests can be granted for one lock resource as long as they are compatible. The compatibility of lock requests is shown in Table 7. All the requests for a particular resource are managed in the lock queues.

Table 7. Mode Compatibility for CLM Locks

	NL	**CR**	**CW**	**PR**	**PW**	**EX**
NL	Yes	Yes	Yes	Yes	Yes	Yes
CR	Yes	Yes	Yes	Yes	Yes	No
CW	Yes	Yes	Yes	No	No	No
PR	Yes	Yes	No	Yes	No	No
PW	Yes	Yes	No	No	No	No
EX	Yes	No	No	No	No	No

Locks in the CLM model can be in one of three states, each with its own associated lock queue:

Granted State Requested lock is compatible with any existing locks. The lock request is granted, and is added to the *grant queue*.

Blocked State Requested lock is not compatible with at least one existing lock. The lock request is not granted, but is added to the *wait queue*, to await the release of the existing incompatible locks.

Converting State A blocked lock request, which is on the wait queue, can then request to be changed to a different mode, in hopes of acquiring a compatible lock. If the lock mode now requested is compatible with any existing locks, the lock is granted, and the request goes to the grant queue. If the new mode is still incompatible with an existing lock, the lock request is placed on the *convert queue*, until the incompatible lock is released.

The *Lock Value Block* in a lock resource is used by concurrent access applications to exchange data with each other. This is a data area that can be used by applications to make information from one lock request available to another.

A.3.3 Lock Management

The Cluster Lock Manager keeps a database of locked resources, distributed over all cluster nodes that have access to the concurrent storage. The node from which the Cluster Lock Manager controls a particular lock resource is called the *master site* for that resource and its data.

When an application requests a lock for a particular resource, the Cluster Lock Manager on the local node checks to see if it has control of that resource. If it does not have control for that resource, it communicates with other Cluster Lock Managers on other cluster nodes over the private network to find the master site for that resource.

The information regarding a lock request resides on the node from which the request was made and the node mastering the requested lock resource (which may be the same node as the requesting node). Therefore, if a node mastering some lock resources fails, the information regarding lock requests from other nodes would still be present on those nodes.

Since accessing a resource from the node where it is mastered is more efficient, the Cluster Lock Manager manager is designed to be able to dynamically resite the mastering of a resource to a node where its usage is the highest. The threshold at which this resiting is done, if at all, is customizable by the user.

Further information on the Cluster Lock Manager can be found in the manual *HACMP Locking Applications*, for either HACMP/6000 Version 3.1 or HACMP 4.1 for AIX, as listed in the *Related Publications* section in the beginning of this redbook.

A.4 Interaction Between the HACMP Software Components

Figure 48 on page 207 shows how the different software components of HACMP that we have seen in the previous sections interact with each other on the same node as well as across nodes.

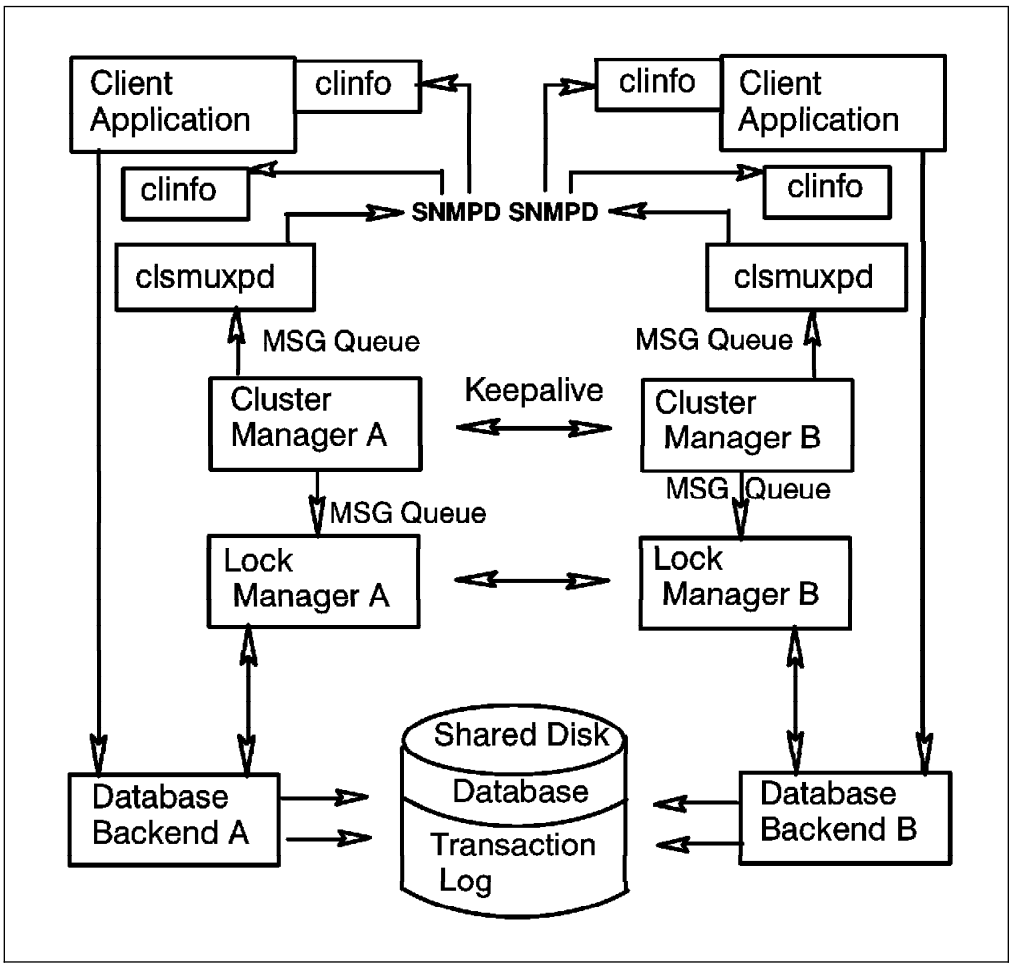


Figure 48. Interaction Between the HACMP Software Components in a Cluster

Appendix B. Disk Setup in an HACMP Cluster

This appendix gives detailed descriptions of the setup of different kinds of shared disk devices for HACMP. You will see how cluster nodes are connected to shared disks and how the storage space on these devices becomes visible to the operating system.

The appendix is divided into three sections, each of which deals with a particular type of disk or subsystem. These sections are:

- SCSI disks and subsystems
- RAID subsystems
- 9333 Serial disk subsystems
- Serial Storage Architecture (SSA) disk subsystems

B.1 SCSI Disks and Subsystems

The SCSI adapters that can be used on a shared SCSI bus in an HACMP cluster are:

- SCSI-2 Differential Controller (FC: 2420, PN: 43G0176)
- SCSI-2 Differential Fast/Wide Adapter/A (FC: 2416, PN: 65G7315)
- Enhanced SCSI-2 Differential Fast/Wide Adapter/A (FC: 2412, PN: 52G3380)

  (This adapter was only supported under AIX 4.1 and HACMP 4.1 for AIX at the time of publishing, but testing was underway to certify the adapter under HACMP/6000 Version 3.1)

The non-RAID SCSI disks and subsystems that you can connect as shared disks in an HACMP cluster are:

- 7204 Models 215, 315, 317, and 325 External Disk Drives
- 9334 Models 011 and 501 SCSI Expansion Units
- 7134-010 High Density SCSI Disk Subsystem

B.1.1 SCSI Adapters

The SCSI-2 Differential Controller is used to connect to 8-bit disk devices on a shared bus. The SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A is usually used to connect to 16-bit devices but can also be used with 8-bit devices.

In a dual head-of-chain configuration of shared disks, there should be no termination anywhere on the bus except at the extremities. Therefore, you should remove the termination resistor blocks from the SCSI-2 Differential Controller and the SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A. The positions of these blocks (U8 and U26 on the SCSI-2 Differential Controller, and RN1, RN2 and RN3 on the SCSI-2 Differential Fast/Wide Adapter/A and Enhanced SCSI-2 Differential Fast/Wide Adapter/A) are shown in Figure 49 on page 210 and Figure 50 on page 210 respectively.

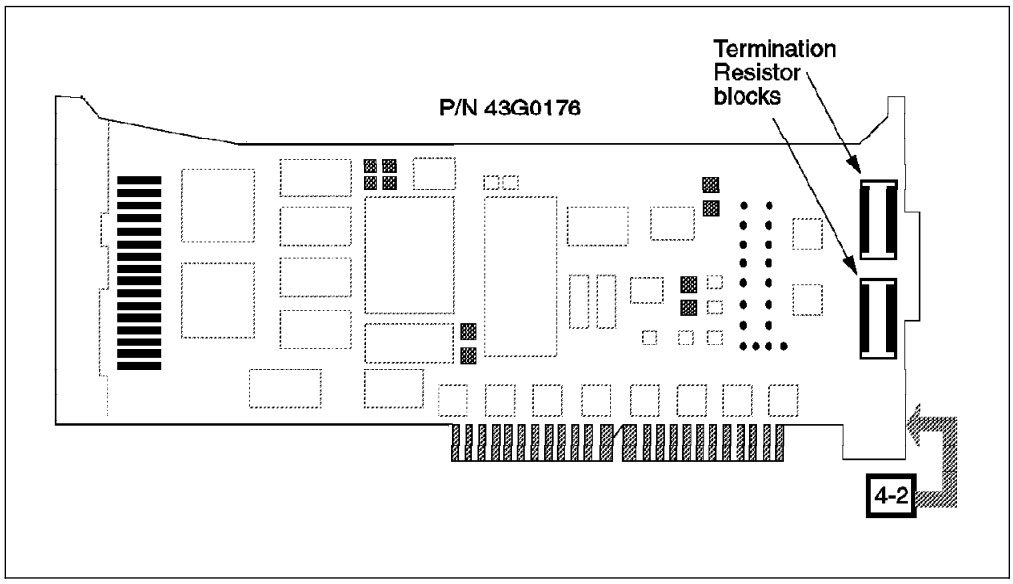


Figure 49. Termination Resistor Blocks on the SCSI-2 Differential Controller

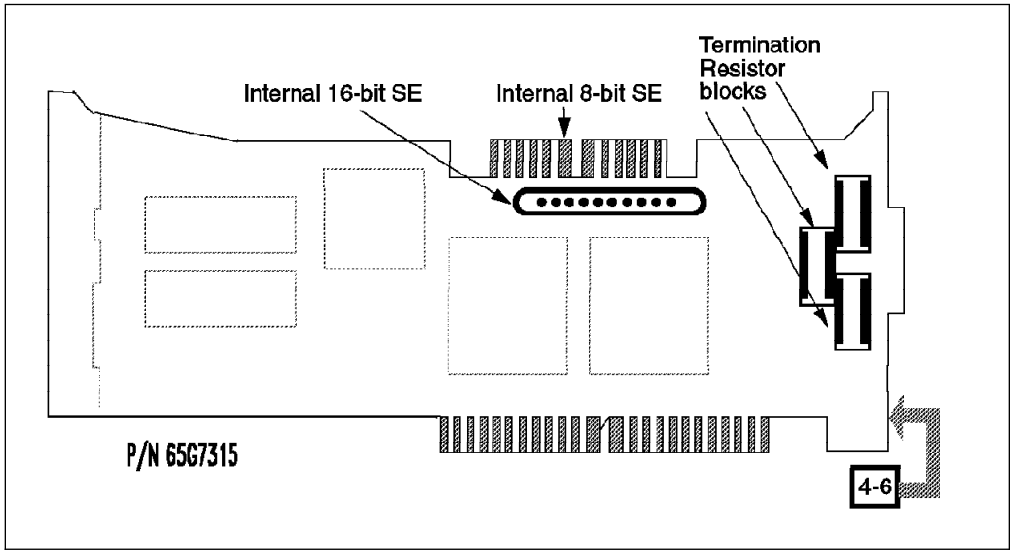


Figure 50. Termination Resistor Blocks on the SCSI-2 Differential Fast/Wide Adapter/A and Enhanced SCSI-2 Differential Fast/Wide Adapter/A

The ID of a SCSI adapter, by default, is 7. Since each device on a SCSI bus must have a unique ID, the ID of at least one of the adapters on a shared SCSI bus has to be changed.

The procedure to change the ID of a SCSI-2 Differential Controller is:

1. At the command prompt, enter `smit chgscsi`.
2. Select the adapter whose ID you want to change from the list presented to you.

```
                                SCSI Adapter

  Move cursor to desired item and press Enter.

    scsi0 Available 00-02 SCSI I/O Controller
    scsi1 Available 00-06 SCSI I/O Controller
    scsi2 Available 00-08 SCSI I/O Controller
    scsi3 Available 00-07 SCSI I/O Controller

  F1=Help                 F2=Refresh              F3=Cancel
  F8=Image                F10=Exit                Enter=Do
  /=Find                  n=Find Next
```

3. Enter the new ID (any integer from 0 to 7) for this adapter in the `Adapter card SCSI ID` field. Since the device with the highest SCSI ID on a bus gets control of the bus, set the adapter's ID to the highest available ID. Set the `Apply change to DATABASE only` field to **yes**.

```
               Change / Show Characteristics of a SCSI Adapter

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

                                                      [Entry Fields]
  SCSI Adapter                                       scsi1
  Description                                        SCSI I/O Controller
  Status                                             Available
  Location                                           00-06
  Adapter card SCSI ID                               [6]                    +#
  BATTERY backed adapter                             no                     +
  DMA bus memory LENGTH                              [0x202000]             +
  Enable TARGET MODE interface                       yes                    +
  Target Mode interface enabled                      yes
  PERCENTAGE of bus memory DMA area for target mode  [50]                   +#
  Name of adapter code download file                 /etc/microcode/8d77.a0>
  Apply change to DATABASE only                      yes                    +



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

4. Reboot the machine to bring the change into effect.

The same task can be executed from the command line by entering:

```
# chdev -l scsi1 -a id=6 -P
```

Also with this method, a reboot is required to bring the change into effect.

The procedure to change the ID of a SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A is almost the same as the one described above. Here, the adapter that you choose from the list you get after executing the `smit chgsys` command should be an `ascsi` device. Also, as, shown below, you need to change the external SCSI ID only.

```
                Change/Show Characteristics of a SCSI Adapter


  SCSI adapter                                          ascsi1
  Description                                           Wide SCSI I/O Control>
  Status                                                Available
  Location                                              00-06
  Internal SCSI ID                                      7                      +#
  External SCSI ID                                      [6]                    +#
  WIDE bus enabled                                      yes                    +
  ...
  Apply change to DATABASE only                         yes
```

The command line version of this is:

```
# chdev -l ascsi1 -a id=6 -P
```

As in the case of the SCSI-2 Differential Controller, a system reboot is required to bring the change into effect.

The maximum length of the bus, including any internal cabling in disk subsystems, is limited to 19 meters for buses connected to the SCSI-2 Differential Controller, and to 25 meters for those connected to the SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A.

B.1.2 Individual Disks and Enclosures

The 7204-215 External Disk Drive is an 8-bit disk that can be connected to the SCSI-2 Differential Controller, the SCSI-2 Differential Fast/Wide Adapter/A, or the Enhanced SCSI-2 Differential Fast/Wide Adapter/A. While there is a theoretical limit of six such disks in an I/O bus connected to two nodes, HACMP supports up to four in a single bus. This support limit is based only on what has been specifically tested by development.

As there are typically choices to be made in lengths of cable connecting disks and adapters in the bus, it is important to keep in mind the bus length limits stated in the last section, while configuring your hardware.

The 7204 Model 315, 317, and 325 External Disk Drives are 16-bit disks that can only be connected to the SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A. For HACMP, the tested limit of these disks in a single shared 16-bit bus is six for the 7204-315, and fourteen for the 7204-317 and 7204-325.

The 9334 Model 011 and 501 SCSI Expansion Units can each contain up to four 8-bit disks. Because of the bus length limitation, you can daisy-chain a maximum of two such units on a shared bus. The number of disks in the enclosures is determined by the number of free SCSI IDs in the bus. The enclosure itself does not have any SCSI ID.

The 7134-010 High Density SCSI Disk Subsystem can contain up to six 16-bit disks in the base unit and six more in the expansion unit. You can either configure your 7134 with just the base unit connected to one shared SCSI bus, or you can

configure it with the base and the expansion unit attached to two different shared SCSI buses. The maximum number of disks in each unit is determined by the number of available SCSI IDs on the shared bus to which it is attached.

B.1.3 Hooking It All Up

In this section we will list the different components required to connect SCSI disks and enclosures on a shared bus. We will also show you how to connect these components together.

B.1.3.1 7204-215 External Disk Drive

To connect a set of 7204-215s to SCSI-2 Differential Controllers on a shared SCSI bus, you need the following:

- SCSI-2 Differential Y-Cable

  FC: 2422 (0.765m), PN: 52G7348

- SCSI-2 Differential System-to-System Cable

  FC: 2423 (2.5m), PN: 52G7349

  This cable is used only if there are more than two nodes attached to the same shared bus.

- SCSI-2 DE Controller Cable

  FC: 2854 or 9138 (0.6m), PN: 87G1358 - OR -

  FC: 2921 or 9221 (4.75m), PN: 67G0593

- SCSI-2 DE Device-to-Device Cable

  FC: 2848 or 9134 (0.66m), PN: 74G8511

- Terminator

  Included in FC 2422 (Y-Cable), PN: 52G7350

Figure 51 shows four RS/6000s, each represented by one SCSI-2 Differential Controller, connected on an 8-bit bus to a chain of 7204-215s.

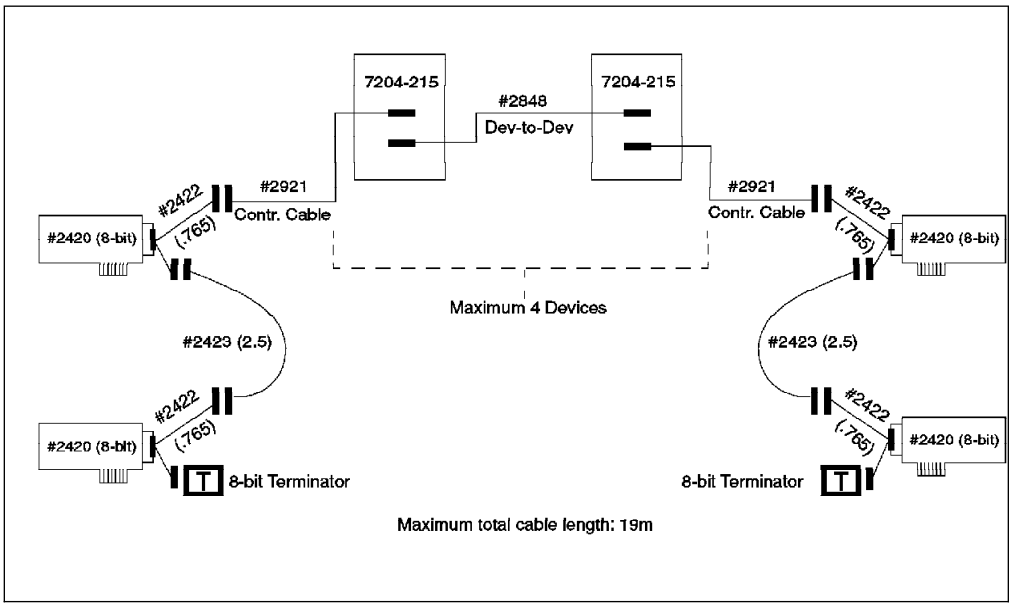


Figure 51. 7204-215 External Disk Drives Connected on an 8-Bit Shared SCSI Bus

B.1.3.2 7204 Model 315, 317, and 325 External Disk Drives

To attach a chain of 7204 Model 315s, 317s, or 325s, or a combination of them to SCSI-2 Differential Fast/Wide Adapter/As or Enhanced SCSI-2 Differential Fast/Wide Adapter/As on a shared 16-bit SCSI bus, you need the following 16-bit cables and terminators:

- 16-Bit SCSI-2 Differential Y-Cable

  FC: 2426 (0.94m), PN: 52G4234

- 16-Bit SCSI-2 Differential System-to-System Cable

  FC: 2424 (0.6m), PN: 52G4291 - OR -

  FC: 2425 (2.5m), PN: 52G4233

  This cable is used only if there are more than two nodes attached to the same shared bus.

- 16-Bit SCSI-2 DE Device-to-Device Cable

  FC: 2845 or 9131 (0.6m), PN: 52G4291 - OR -

  FC: 2846 or 9132 (2.5m), PN: 52G4233

- 16-Bit Terminator

  Included in FC 2426 (Y-Cable), PN: 61G8324

Figure 52 shows four RS/6000s, each represented by one SCSI-2 Differential Fast/Wide Adapter/A, connected on a 16-bit bus to a chain of 7204-315s. The connections would be the same for the 7204-317, and Model 325 drives. You could also substitute the Enhanced SCSI-2 Differential Fast/Wide Adapter/A (feature code 2412) for the SCSI-2 Differential Fast/Wide Adapter/As shown in the figure, if you are running HACMP 4.1 for AIX.

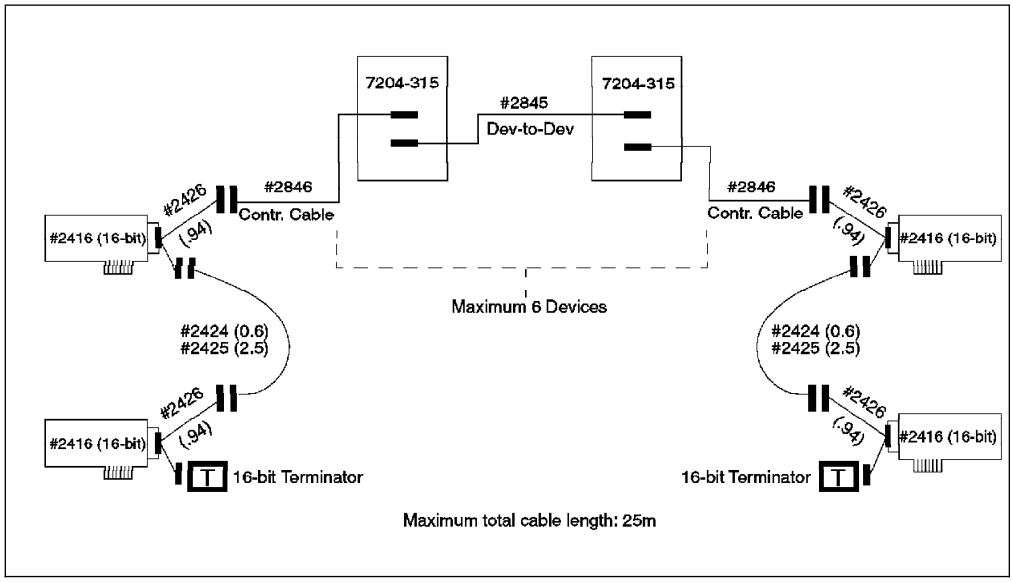


Figure 52. 7204-315 External Disk Drives Connected on a 16-Bit Shared SCSI Bus

B.1.3.3 9334-011 and 9334-501 SCSI Expansion Units

For connecting 9334 Models 011 or 501 to SCSI-2 Differential Controllers on a shared 8-bit SCSI bus, you require the following, in all cases:

- SCSI-2 Differential Y-Cable

  FC: 2422 (0.765m), PN: 52G7348

- SCSI-2 Differential System-to-System Cable

  FC: 2423 (2.5m), PN: 52G7349

  This cable is used only if there are more than two nodes attached to the same shared bus.

- Terminator

  Included in FC 2422 (Y-Cable), PN: 52G7350

In addition to the common set of cables, the 9334-011 requires:

- SCSI-2 DE Controller Cable

  FC: 2921 or 9221 (4.75m), PN: 67G0593 - OR -

  FC: 2923 or 9223 (8.0m), PN: 95X2494

- SCSI-2 DE Device-to-Device Cable

  FC: 2925 or 9225 (2.0m), PN: 95X2492

In addition to the common set of cables, the 9334-501 requires:

- SCSI-2 DE Controller Cable

  FC: 2931 (1.48m), PN: 70F9188 - OR -

  FC: 2933 (2.38m), PN: 45G2858 - OR -

  FC: 2935 (4.75m), PN: 67G0566 - OR -

  FC: 2937 (8.0m), PN: 67G0562

- SCSI-2 DE Device-to-Device Cable:

  FC: 2939 or 9239 (2.0m), PN: 95X2498

Figure 53 on page 216 shows four RS/6000s, each represented by one SCSI-2 Differential Controller, connected on an 8-bit bus to a chain of 9334-011s.
Figure 54 on page 216 shows four RS/6000s, each represented by one SCSI-2 Differential Controller, connected on an 8-bit bus to a chain of 9334-501s.

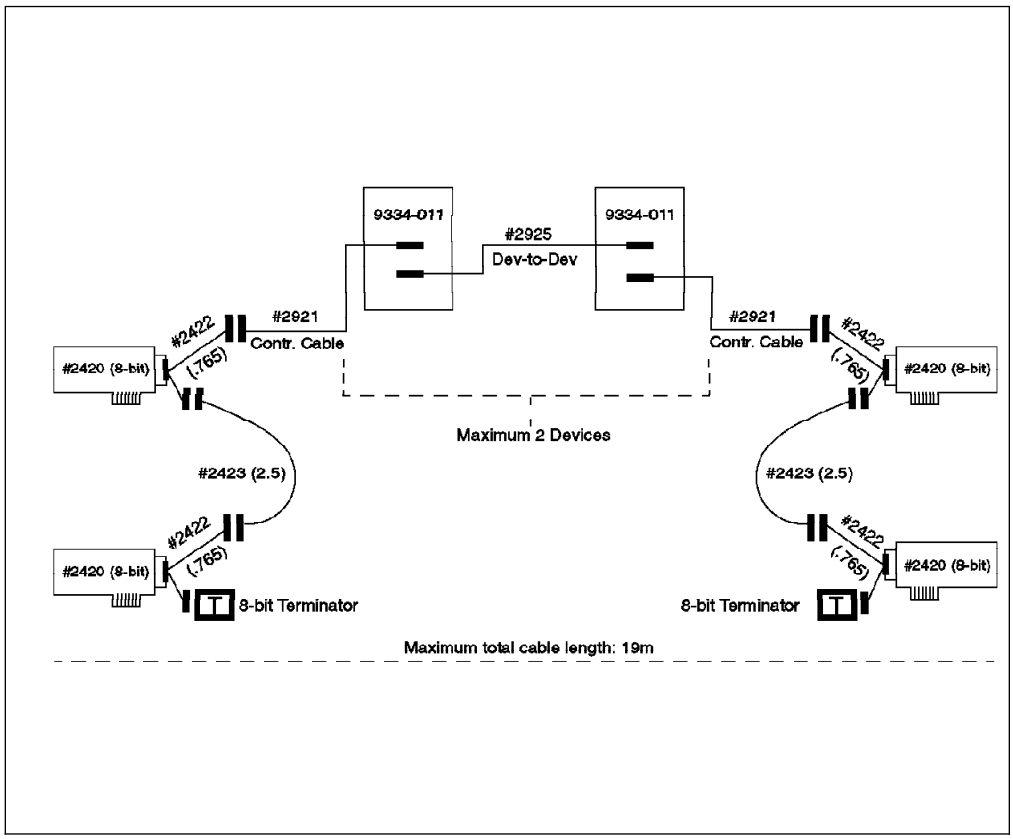


Figure 53. 9334-011 SCSI Expansion Units Connected on an 8-Bit Shared SCSI Bus

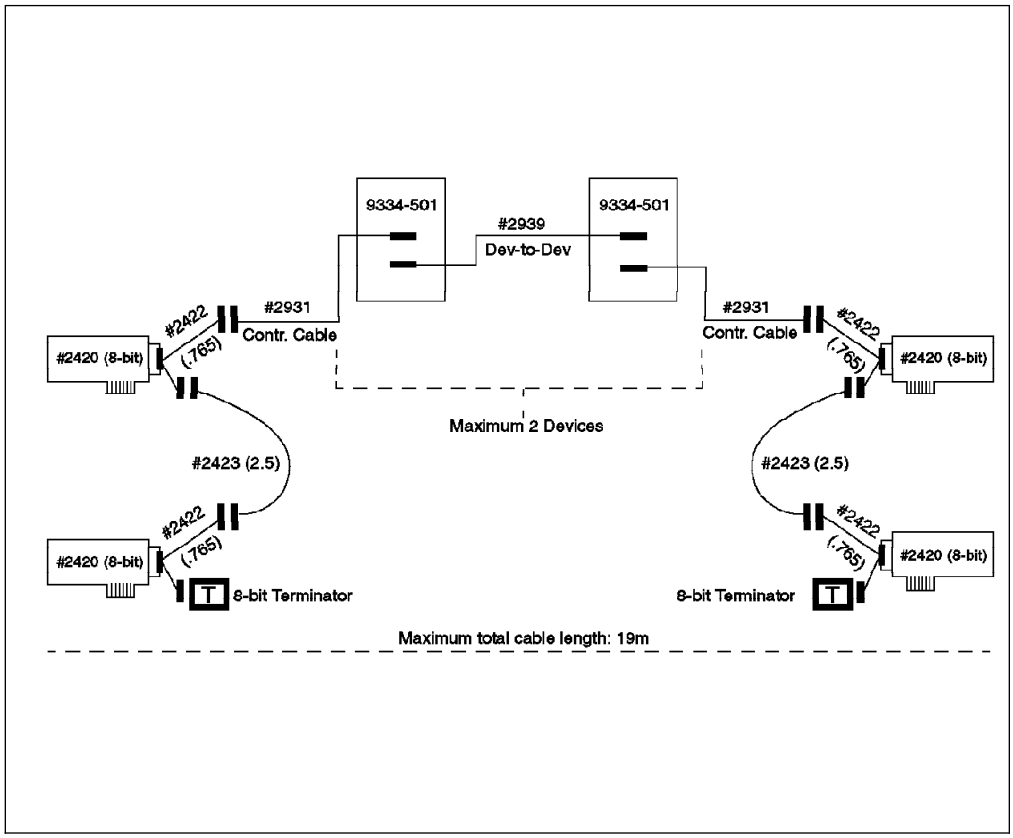


Figure 54. 9334-501 SCSI Expansion Units Connected on an 8-Bit Shared SCSI Bus

B.1.3.4 7134-010 High Density SCSI Disk Subsystem

To attach a 7134-010 to a SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A on a shared 16-bit SCSI bus, you need the following:

- 16-Bit SCSI-2 Differential Y-Cable

  FC: 2426 (0.94m), PN:52G4234

- 16-Bit SCSI-2 Differential System-to-System Cable

  FC: 2424 (0.6m), PN: 52G4291 - OR -

  FC: 2425 (2.5m), PN: 52G4233

  This cable is used only if there are more than two nodes attached to the same shared bus.

- 16-Bit Differential SCSI Cable

  FC: 2902 (2.4m), PN: 88G5750 - OR -

  FC: 2905 (4.5m), PN: 88G5749 - OR -

  FC: 2912 (12.0m), PN: 88G5747 - OR -

  FC: 2914 (14.0m), PN: 88G5748 - OR -

  FC: 2918 (18.0m), PN: 88G5746

- 16-Bit Terminator (T)

  Included in FC 2426 (Y-Cable), PN: 61G8324

Figure 55 on page 218 shows four RS/6000s, each represented by two SCSI-2 Differential Fast/Wide Adapter/As, connected on a 16-bit bus to a 7134-010 with a base and an expansion unit. You could also substitute the Enhanced SCSI-2 Differential Fast/Wide Adapter/A (feature code 2412) for the SCSI-2 Differential Fast/Wide Adapter/As shown in the figure, if you are running HACMP 4.1 for AIX.

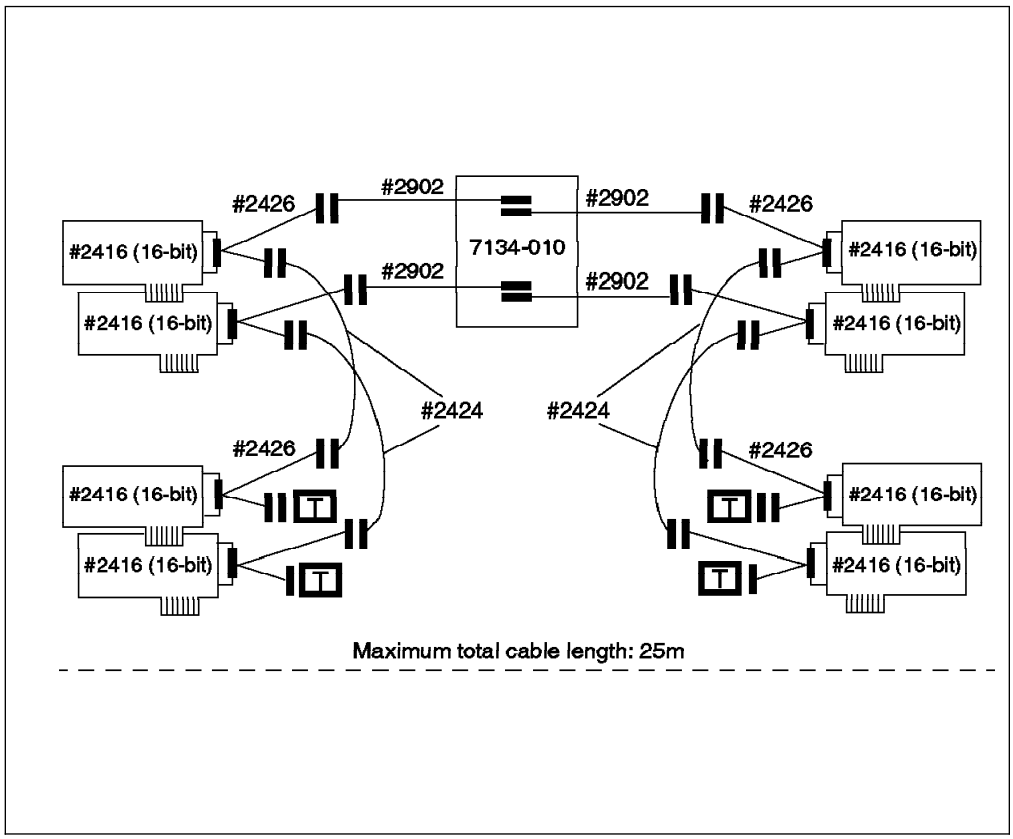


Figure 55. 7134-010 High Density SCSI Disk Subsystem Connected on Two 16-Bit Shared SCSI Buses

B.1.4 AIX's View of Shared SCSI Disks

If your shared SCSI bus has been set up without violating any of the restrictions for termination, SCSI IDs, or cable length, the nodes connected to the shared bus should be able to configure each disk, including the ones inside a 9334 or a 7134, as a separate hdisk device at the next system restart.

B.2 RAID Subsystems

The SCSI adapters that can be used to connect RAID subsystems on a shared SCSI bus in an HACMP cluster are:

- SCSI-2 Differential Controller (FC: 2420, PN: 43G0176)
- SCSI-2 Differential Fast/Wide Adapter/A (FC: 2416, PN: 65G7315)
- Enhanced SCSI-2 Differential Fast/Wide Adapter/A (FC: 2412)

  (This adapter was only supported under AIX 4.1 and HACMP 4.1 for AIX at the time of publishing, but testing was underway to certify the adapter under HACMP/6000 Version 3.1)

The RAID subsystems that you can connect on a shared bus in an HACMP cluster are:

- 7135-110 (HACMP/6000 Version 3.1 only, at the time of publishing) and 7135-210 (HACMP 4.1 for AIX only) RAIDiant Array

- 7137 Model 412, 413, 414, 512, 513, and 514 Disk Array Subsystems

  Note: Existing IBM 3514 RAID Array models continue to be supported as shared disk subsystems under HACMP, but since this subsystem has been withdrawn from marketing, it is not described here. As far as cabling and connection characteristics are concerned, the 3514 follows the same rules as the 7137 Disk Array subsystems.

B.2.1 SCSI Adapters

A description of the SCSI adapters that can be used on a shared SCSI bus is given in Section B.1.1, “SCSI Adapters” on page 209.

B.2.2 RAID Enclosures

The 7135 RAIDiant Array can hold a maximum of 30 single-ended disks in two units (one base and one expansion). It has one controller by default, and another controller can be added for improved performance and availability. Each controller takes up one SCSI ID. The disks sit on internal single-ended buses and hence do not take up IDs on the external bus. In an HACMP cluster, each 7135 should have two controllers, each of which is connected to a separate shared SCSI bus. This configuration protects you against any failure (SCSI adapter, cables, or RAID controller) on either SCSI bus.

Because of cable length restrictions, a maximum of two 7135s on a shared SCSI bus is supported by HACMP.

The 7137 Model 412, 413, 414, 512, 513, and 514 Disk Array Subsystems can hold a maximum of eight disks. Each model has one RAID controller, that takes up one SCSI ID on the shared bus. You can have a maximum of two 7137s connected to a maximum of four nodes on an 8-bit or 16-bit shared SCSI bus.

B.2.3 Connecting RAID Subsystems

In this section, we will list the different components required to connect RAID subsystems on a shared bus. We will also show you how to connect these components together.

B.2.3.1 7135-110 or 7135-210 RAIDiant Array

The 7135-110 RAIDiant Array can be connected to multiple systems on either an 8-bit or a 16-bit SCSI-2 differential bus. The Model 210 can only be connected to a 16-bit SCSI-2 Fast/Wide differential bus, using the Enhanced SCSI-2 Differential Fast/Wide Adapter/A.

To connect a set of 7135-110s to SCSI-2 Differential Controllers on a shared 8-bit SCSI bus, you need the following:

- SCSI-2 Differential Y-Cable

  FC: 2422 (0.765m), PN: 52G7348

- SCSI-2 Differential System-to-System Cable

  FC: 2423 (2.5m), PN: 52G7349

  This cable is used only if there are more than two nodes attached to the same shared bus.

- Differential SCSI Cable (RAID Cable)

FC: 2901 or 9201 (0.6m), PN: 67G1259 - OR -

FC: 2902 or 9202 (2.4m), PN: 67G1260 - OR -

FC: 2905 or 9205 (4.5m), PN: 67G1261 - OR -

FC: 2912 or 9212 (12m), PN: 67G1262 - OR -

FC: 2914 or 9214 (14m), PN: 67G1263 - OR -

FC: 2918 or 9218 (18m), PN: 67G1264

- Terminator (T)

  Included in FC 2422 (Y-Cable), PN: 52G7350

- Cable Interposer (I)

  FC: 2919, PN: 61G8323

  One of these is required for each connection between a SCSI-2 Differential Y-Cable and a Differential SCSI Cable going to the 7135 unit, as shown in Figure 56.

Figure 56 shows four RS/6000s, each represented by two SCSI-2 Differential Controllers, connected on two 8-bit buses to two 7135-110s each with two controllers.

Note

The diagrams in this book give a logical view of the 7135 subsystem. Please refer to the *7135 Installation and Service Guide* for the exact positions of the controllers and their corresponding connections.

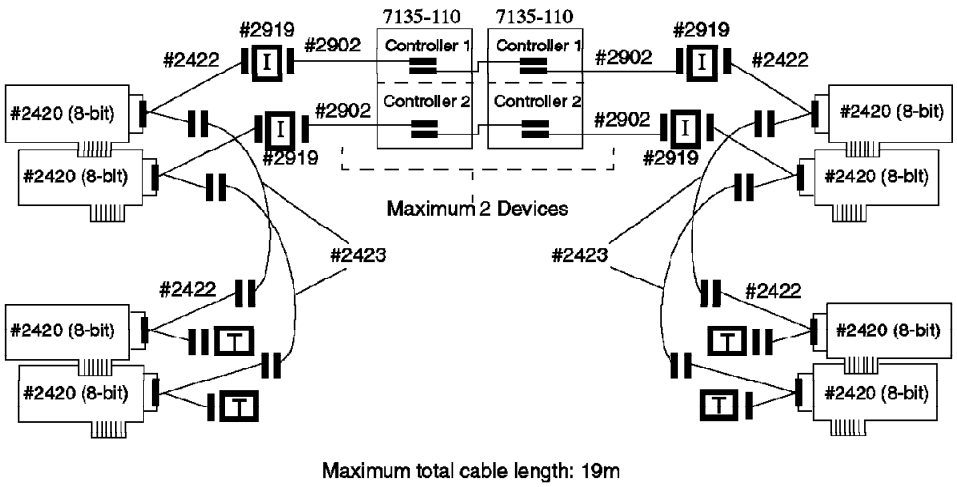


Figure 56. 7135-110 RAIDiant Arrays Connected on Two Shared 8-Bit SCSI Buses

To connect a set of 7135s to SCSI-2 Differential Fast/Wide Adapter/As or Enhanced SCSI-2 Differential Fast/Wide Adapter/As on a shared 16-bit SCSI bus, you need the following:

- 16-Bit SCSI-2 Differential Y-Cable

FC: 2426 (0.94m), PN: 52G4234

- 16-Bit SCSI-2 Differential System-to-System Cable

  FC: 2424 (0.6m), PN: 52G4291 - OR -

  FC: 2425 (2.5m), PN: 52G4233

  This cable is used only if there are more than two nodes attached to the same shared bus.

- 16-Bit Differential SCSI Cable (RAID Cable)

  FC: 2901 or 9201 (0.6m), PN: 67G1259 - OR -

  FC: 2902 or 9202 (2.4m), PN: 67G1260 - OR -

  FC: 2905 or 9205 (4.5m), PN: 67G1261 - OR -

  FC: 2912 or 9212 (12m), PN: 67G1262 - OR -

  FC: 2914 or 9214 (14m), PN: 67G1263 - OR -

  FC: 2918 or 9218 (18m), PN: 67G1264

- 16-Bit Terminator (T)

  Included in FC 2426 (Y-Cable), PN: 61G8324

Figure 57 shows four RS/6000s, each represented by two SCSI-2 Differential Fast/Wide Adapter/As, connected on two 16-bit buses to two 7135-110s, each with two controllers.

The 7135-210 requires the Enhanced SCSI-2 Differential Fast/Wide Adapter/A adapter for connection. Other than that, the cabling is exactly the same as shown in Figure 57, if you just substitute the Enhanced SCSI-2 Differential Fast/Wide Adapter/A (FC: 2412) for the SCSI-2 Differential Fast/Wide Adapter/A (FC: 2416) in the picture.

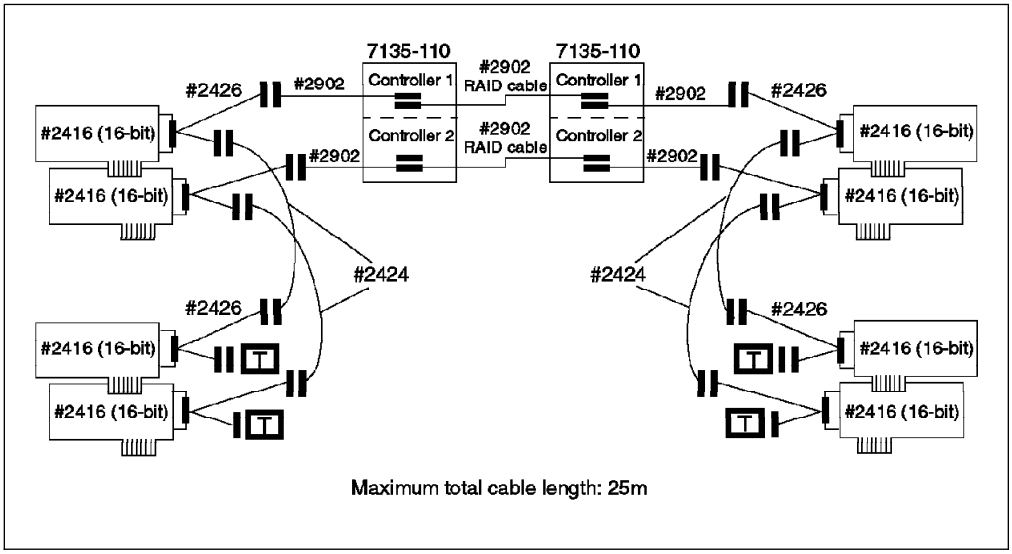


Figure 57. 7135-110 RAIDiant Arrays Connected on Two Shared 16-Bit SCSI Buses

B.2.3.2 7137 Model 412, 413, 414, 512, 513, and 514 Disk Array Subsystems

To connect two 7137s to SCSI-2 Differential Controllers on a shared 8-bit SCSI bus, you need the following:

- SCSI-2 Differential Y-Cable

  FC: 2422 (0.765m), PN: 52G7348

- SCSI-2 Differential System-to-System Cable

  FC: 2423 (2.5m), PN: 52G7349

  This cable is used only if there are more than two nodes attached to the same shared bus.

- Attachment Kit to SCSI-2 Differential High-Performance External I/O Controller

  FC: 2002, PN: 46G4157

  This includes a 4.0-meter cable, an installation diskette, and the *IBM 7137 (or 3514) RISC System/6000 System Attachment Guide*.

- Multiple Attachment Cable

  FC: 3001, PN: 21F9046

  This includes a 2.0-meter cable, an installation diskette, and connection instructions.

- Terminator (T)

  Included in FC 2422 (Y-Cable), PN: 52G7350

Figure 58 shows four RS/6000s, each represented by one SCSI-2 Differential Controller, connected on an 8-bit bus to two 7137s.

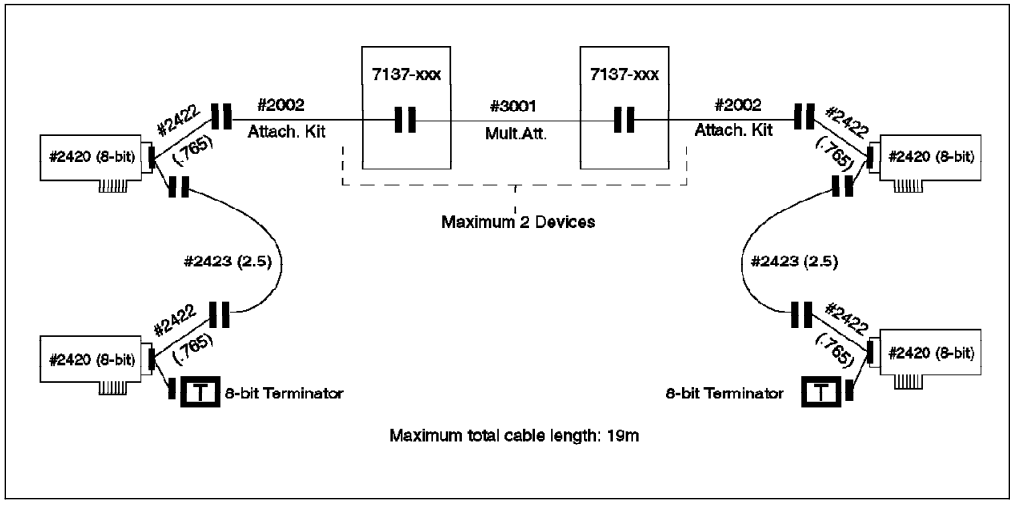


Figure 58. 7137 Disk Array Subsystems Connected on an 8-Bit SCSI Bus

To connect two 7137s to SCSI-2 Differential Fast/Wide Adapter/As or Enhanced SCSI-2 Differential Fast/Wide Adapter/As on a shared 16-bit SCSI bus, you need the following:

- 16-Bit SCSI-2 Differential Y-Cable

  FC: 2426 (0.94m), PN: 52G4234

- 16-Bit SCSI-2 Differential System-to-System Cable

  FC: 2424 (0.6m), PN: 52G4291 - OR -

  FC: 2425 (2.5m), PN: 52G4233

  This cable is used only if there are more than two nodes attached to the same shared bus.

- Attachment Kit to SCSI-2 Differential Fast/Wide Adapter/A or Enhanced SCSI-2 Differential Fast/Wide Adapter/A

  FC: 2014, PN: 75G5028

  This includes a 4.0-meter cable, an installation diskette, and the *IBM 7137 (or 3514) RISC System/6000 System Attachment Guide*.

- Multiple Attachment Cable

  FC: 3001, PN: 21F9046

  This includes a 2.0-meter cable, an installation diskette, and connection instructions.

- 16-Bit Terminator (T)

  Included in FC 2426 (Y-Cable), PN: 61G8324

Figure 59 shows four RS/6000s, each represented by one SCSI-2 Differential Fast/Wide Adapter/As, connected on a 16-bit bus to two 7137s. The Enhanced SCSI-2 Differential Fast/Wide Adapter/A uses exactly the same cabling, and could be substituted for the SCSI-2 Differential Fast/Wide Adapter/A in an AIX 4.1 and HACMP 4.1 for AIX configuration.

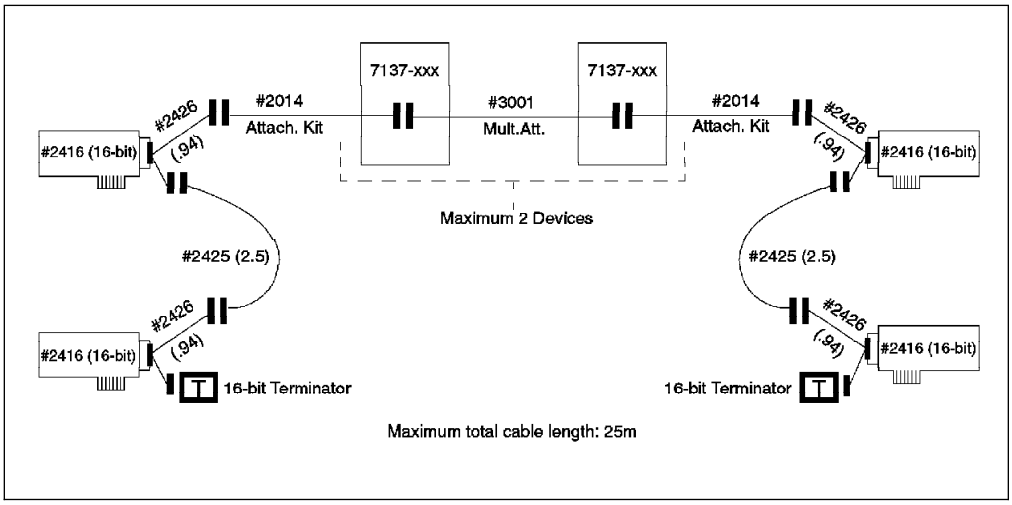


Figure 59. 7137 Disk Array Subsystems Connected on a 16-Bit SCSI Bus

B.2.4 AIX's View of Shared RAID Devices

The 7135 and 7137 subsystems come preconfigured with Logical Units (LUNs) from the factory. Please refer to Section 4.3.3, “RAID Disk Array Features” on page 67 for a description of LUNs. Each LUN gets recognized by nodes on the shared bus as an hdisk device. You can reconfigure the LUNs in a 7135 to suit your requirements by using the 7135 Disk Array Manager software. A 7137 can be reconfigured by using the operator panel on the subsystem itself.

The procedure for configuring LUNs is beyond the scope of this book. Please refer to *7135 RAIDiant Array for AIX - Installation and Reference* for instructions on using the 7135 Disk Array Manager software to create and manage LUNs in a 7135. Please refer to the product documentation that comes with the 7137 subsystem for instructions to set up LUNs on that subsystem.

B.3 Serial Disk Subsystems

To connect serial disk subsystems as shared devices in an HACMP cluster, the adapter that you will use is:

- High-Performance Disk Drive Subsystem Adapter 40/80 MB/sec. (FC: 6212, PN: 67G1755)

The serial disk subsystems that you can connect as shared devices in an HACMP cluster are:

- 9333 Model 011 and 501 High-Performance Disk Drive Subsystems

B.3.1 High-Performance Disk Drive Subsystem Adapter

The High-Performance Disk Drive Subsystem Adapter has four ports, with each port supporting the attachment of a single 9333-011 or 501 controller. Since each controller can drive up to a maximum of four disks, of 2 GB capacity each, you can access up to 32 GB of data with one High-Performance Disk Drive Subsystem Adapter. There is no limit on the number of serial disk adapters that you can have in one node. You do not need to worry about device addresses or terminators with serial disks, since the subsystem is self-addressing. This feature makes it much easier to install and configure than the SCSI options discussed previously.

B.3.2 9333 Disk Subsystems

The 9333 Model 011 and 501 High-Performance Disk Drive Subsystems can each contain a maximum of four disks. The 9333-011 is in a drawer configuration, and is used on rack-mounted models. The 9333-501 is in a mini-tower configuration, and is used on all other models of the RS/6000. Each 9333 subsystem requires a dedicated port on a High-Performance Disk Drive Subsystem Adapter. A maximum of four 9333s can attach to one High-Performance Disk Drive Subsystem Adapter, one for each port. Each 9333 subsystem can be shared with a maximum of eight nodes in a cluster. To connect 9333s to an RS/6000, you need to have AIX Version 3.2.4 or later, and AIX feature 5060 (IBM High-Performance Disk Subsystem Support) installed.

B.3.3 Connecting Serial Disk Subsystems in an HACMP Cluster

To connect a 9333-011 or 501 to two systems, each containing High-Performance Disk Drive Subsystem Adapters, you need the following:

- Serial-Link Cable (Quantity 2)

  FC: 9210 or 3010 (10m)

  FC: 9203 or 3003 (3m)

To connect a 9333-011 or 501 to three or more systems, each containing High-Performance Disk Drive Subsystem Adapters, you need the following:

- Serial-Link Cable (One for each system connection)

FC: 9210 or 3010 (10m)

FC: 9203 or 3003 (3m)

- Multiple System Attachment Feature(s)

  FC: 4001 (Connect up to four systems)

  FC: 4002 (Connect up to eight systems)

  Feature 4001 is a prerequisite for feature 4002.

Figure 60 shows eight RS/6000s, each having a High-Performance Disk Drive Subsystem Adapter, connected to one 9333-501 with the Multiple System Attachment Features 4001 and 4002 installed.

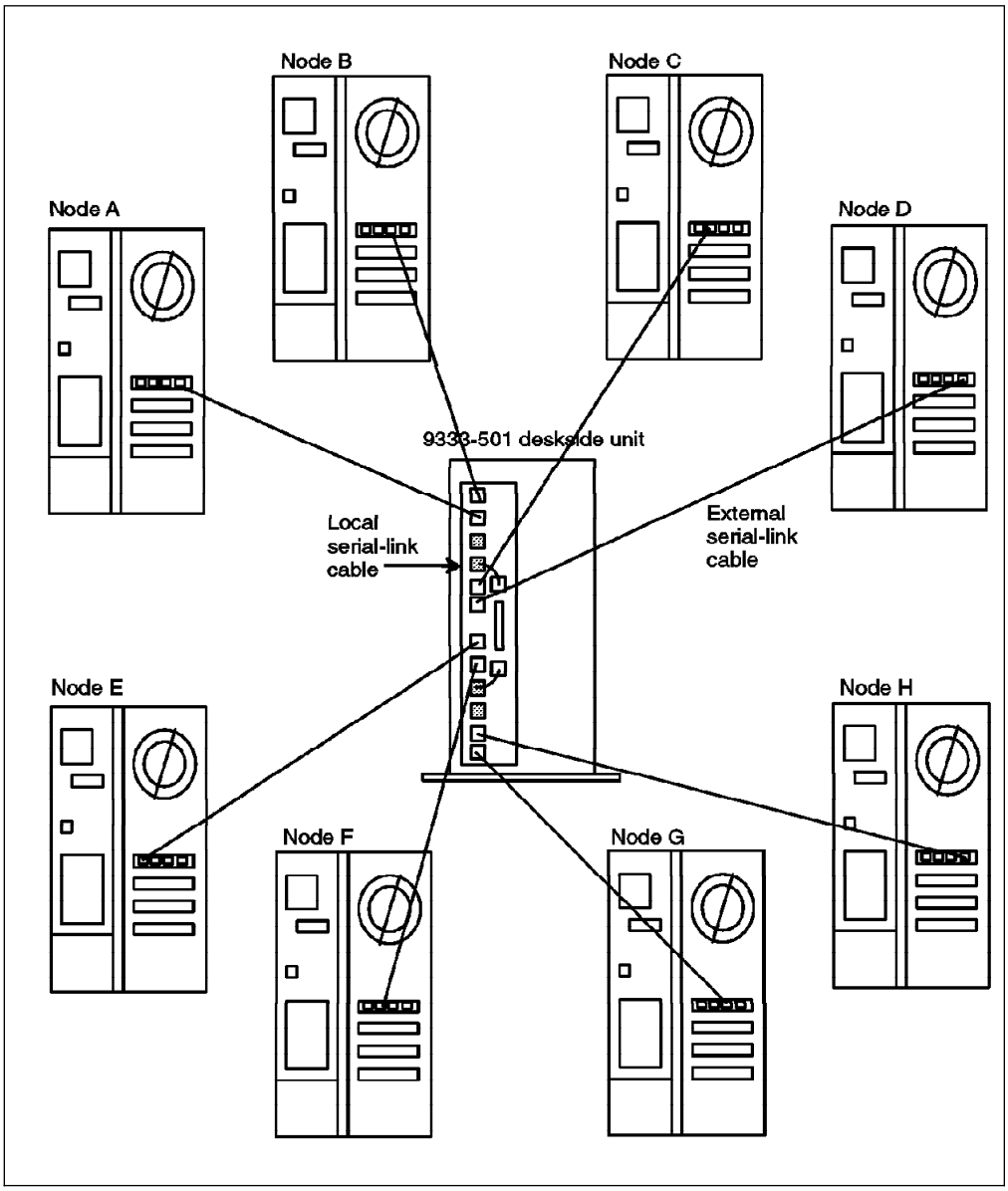


Figure 60. 9333-501 Connected to Eight Nodes in an HACMP Cluster (Rear View)

B.3.4 AIX's View of Shared Serial Disk Subsystems

Each individual serial disk inside a 9333 subsystem appears as a separate hdisk device on all nodes connected to the subsystem.

B.4 Serial Storage Architecture (SSA) Subsystems

Serial Storage Architecture is a second generation of the high performance serial disk subsystems, started with the IBM 9333 subsystems. SSA subsystems provide new levels of performance, reliability, and flexibility, and are IBM's strategic high performance disk subsystems for the future.

SSA Support in HACMP

At the time of publishing, the IBM 7133 SSA subsystem was supported for sharing between two nodes only, in a cluster running AIX 3.2.5 and HACMP/6000 Version 3.1. Support for sharing a subsystem between larger numbers of nodes, and support for the the use of the 7133 in an AIX 4.1 and HACMP 4.1 for AIX cluster are expected to be added at a later date. Please check with your IBM representative for the latest support information.

To connect SSA subsystems as shared devices in your HACMP cluster, the adapter that you will use is:

- SSA Four Port Adapter (FC: 6124)

This adapter is shown in Figure 61 on page 227.

The SSA disk subsystems that you can connect as shared devices in an HACMP cluster are:

- IBM 7133-010 SSA Disk Subsystem

  This model is in a drawer configuration, for use in rack mounted systems.

- IBM 7133-500 SSA Disk Subsystem

  This model is in a standalone tower configuration, for use in all models.

B.4.1 SSA Software Requirements

The IBM 7133 SSA Disk Subsystem is supported by AIX Version 3.2.5 with additional program temporary fixes (PTFs), and the AIX 3.2.5 device driver shipped with the SSA Four Port Adapter (FC 6214 on the attaching system). For ease of installation, these PTFs are packaged with the device driver on the CD-ROM shipped with the adapter.

Customers without access to CD-ROM drives on their machines or network can obtain the device driver and required PTFs through the FIXDIST system. The device driver is available as APAR IX52018. The required PTFs, on FIXDIST, are identified as PMP3251.

For alternative delivery, contact your Software Service representative for the appropriate PTFs. The additional Version 3.2.5 PTFs (without the AIX 3.2.5 device driver for the adapter) are included on all AIX Version 3.2.5 orders shipped after May 19, 1995, labelled *AIX 3.2.5 Enhancement 5 (3250-05-00)*.

At the time of publishing, SSA support for AIX 4.1 was expected to be announced by the end of 1995. Please check with your IBM representative for its most current status.

B.4.2 SSA Four Port Adapter

The IBM SSA Four Port Adapter supports the connection of a large capacity of SSA storage. The basic concept of SSA storage connection is that of a loop. An SSA loop starts at one port on the SSA Four Port Adapter continues through a number of SSA disk drives, and concludes at another port on an SSA Four Port Adapter. Each loop can include up to 48 disk devices. Since you can support two loops on each SSA Four Port Adapter, you can support up to 96 disk devices on each adapter. If all those disk devices were of the 4.5 GB capacity, this would provide a potential capacity of 432 GB on an adapter. The adapter itself is shown in Figure 61.

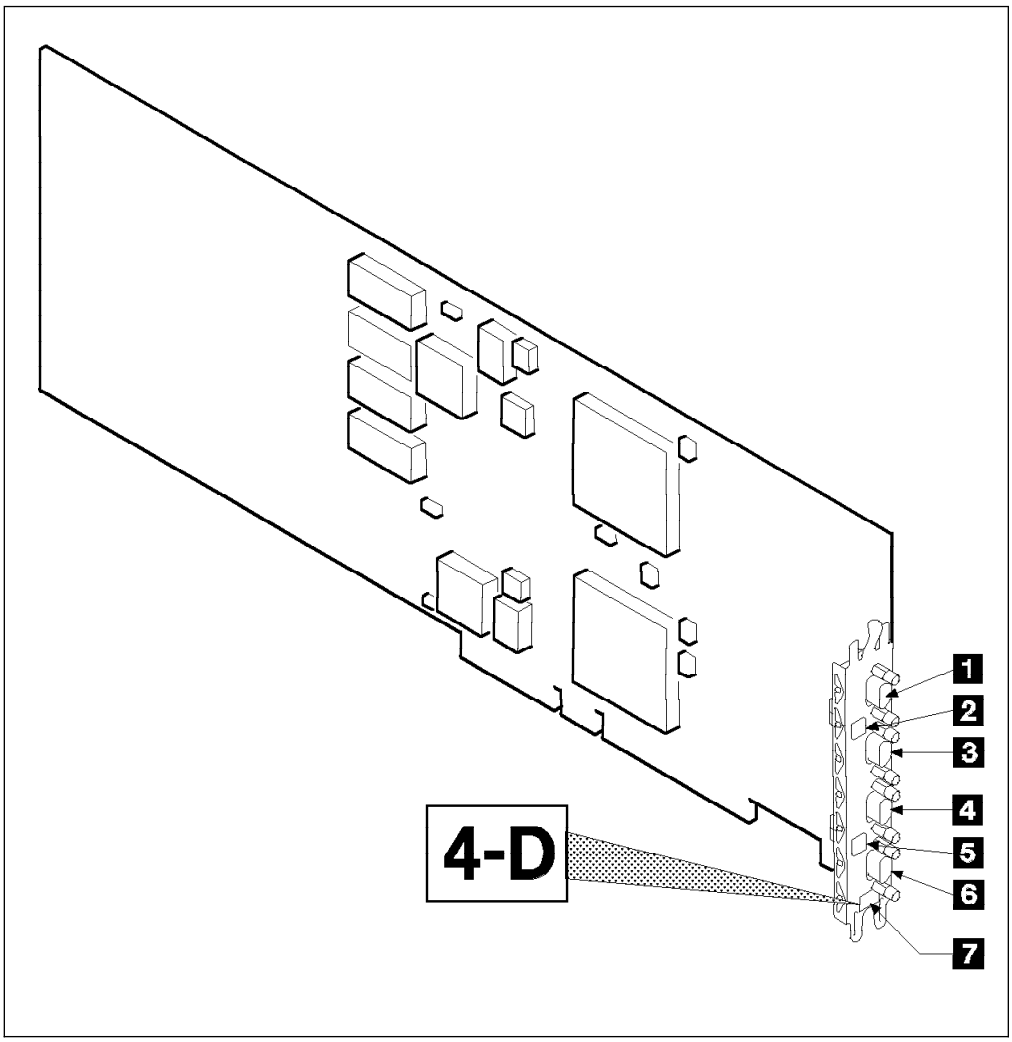


Figure 61. SSA Four Port Aapter

The labeled components of the adapter in the figure are as follows:

1. Connector B2
2. Green light for adapter port pair B
3. Connector B1

4. Connector A2
5. Green light for adapter port pair A
6. Connector A1
7. Type-number label

The green lights for each adapter port pair indicate the status of the attached loop as follows:

Off	Both ports are inactive. If disk drives are connected to these ports, then either the modules have failed or their SSA links have not been enabled.
Permanently on	Both ports are active.
Slow flash	Only one port is active.

The SSA loop that you create need not begin and end on the same &ssaadt.. Loops can be made to go from one adapter to another adapter in the same system or in a different system. There can at most be two adapters on the same loop.

B.4.3 IBM 7133 SSA Disk Subsystem

The IBM 7133 SSA Disk Subsystem is available in two models, the rack drawer model 010 and the standalone tower model 500. While these models hold their disk drives in different physical orientations, they are functionally the same. Each model is capable of holding up to 16 SSA disk drives, each of which can be 1.1 GB, 2.2 GB, or 4.5 GB drives. The subsystem comes standard with four 2.2 GB drives, which can be traded for higher or lower capacity drives at order time.

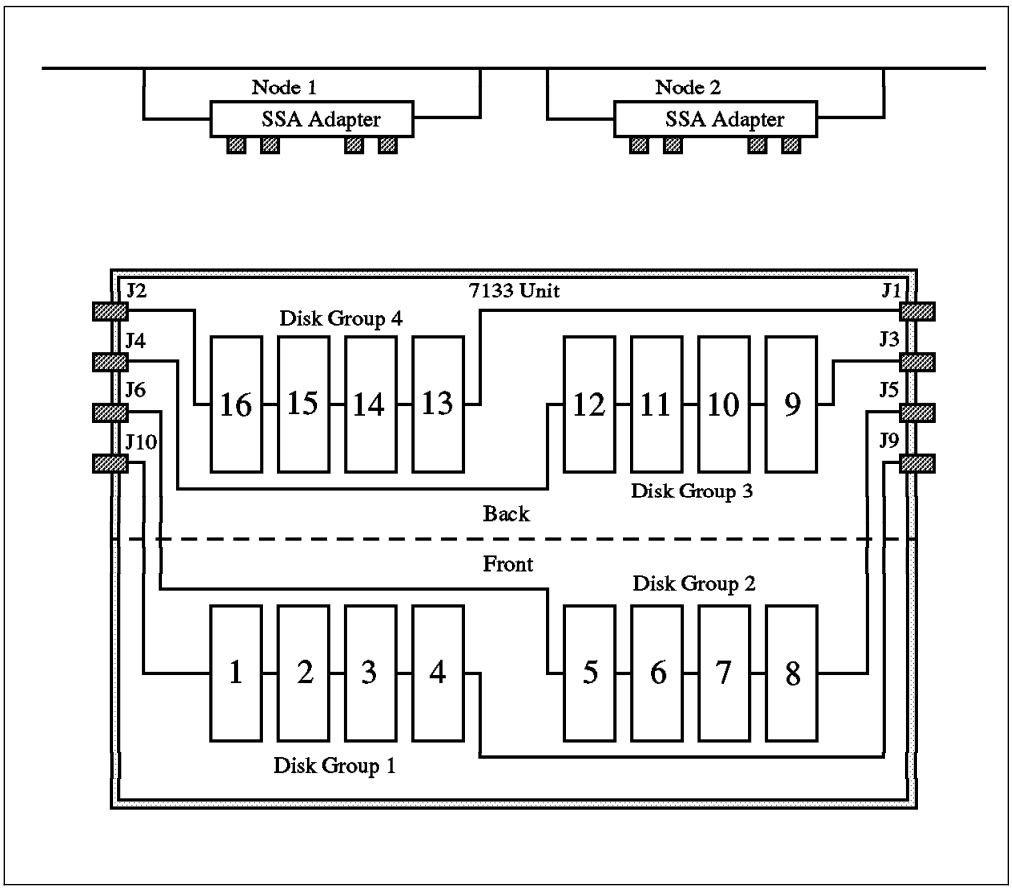


Figure 62. IBM 7133 SSA Disk Subsystem

As you can see in Figure 62, each group of four disk drives in the subsystem is internally cabled as a loop. Disk Group 1 includes disk drive positions 1-4 and is cabled between connectors J9 and J10. Disk Group 2 includes disk drive positions 5-8 and is cabled between connectors J5 and J6. You can also see Disk Groups 3 and 4 in the picture. These internal loops can either be cabled together into larger loops, or individually connected to SSA Four Port Adapters. For instance, if you were to connect a short cable between connectors J6 and J10, you would have a loop of eight drives that could be connected to the SSA Four Port Adapter from connectors J5 and J9.

B.4.4 SSA Cables

SSA cables are available in a variety of different lengths. The connectors at each end are identical, which makes them very easy to use. These cables can be used to connect four disk internal loops together into larger loops within the 7133 subsystem itself, to connect multiple 7133 subsystems together in a larger loop, or to connect a 7133 subsystem to an SSA Four Port Adapter. The same cable can be used for any of these connections, as long as it is long enough. In Table 8 on page 230 is a list of cable feature codes, along with their lengths, and part numbers:

Table 8. Serial Storage Architecture (SSA) Cables		
Cable Description	**Feature Code**	**Part Number**
SSA Copper Cable (0.18 meters)	5002	07H9163
SSA Copper Cable (0.6 meters)	5006	31H7960
SSA Copper Cable (1.0 meter)	5010	07H8985
SSA Copper Cable (2.5 meters)	5025	32H1465
SSA Copper Cable (5.0 meters)	5050	88G6406
SSA Copper Cable (10 meters)	5100	32H1466
SSA Copper Cable (25 meters)	5250	88G6406

The feature code numbers start with the number 5, and the next three digits give a rounded length in meters, which makes the feature numbers easy to understand and remember. As was mentioned before, the only difference between these cables is their length. They can be used interchangeably to connect any SSA components together.

If you obtain an announcement letter for the 7133 SSA Subsystem, you will also see a number of other cable feature codes listed, with the same lengths (and same prices) as those in Table 8. You needn't worry or be confused about these, since they are the same cables as those in the tables. As long as you have the correct length of cable for the components you need to connect, you have the right cable.

The maximum distance between components in an SSA loop using IBM cabling is 25 meters. With SSA, there is no special maximum cabling distance for the entire loop. In fact, the maximum cabling distance for the loop would be the maximum distance between components (disks or adapters), mulitplied by the maximum number of components (48) in a loop.

B.4.5 Connecting 7133 SSA Subsystems in an HACMP Cluster

The flexibility of the SSA subsystem creates many different options for attaching SSA subsystems in a cluster, with varying levels of redundancy and availability. Since SSA subsystems are currently only supported for sharing between two nodes, these are the examples that we will use. However, it is expected that you will be able to expand these examples by adding more nodes into the loop(s) in the future. We will illustrate two simple scenarios of SSA connection in this section.

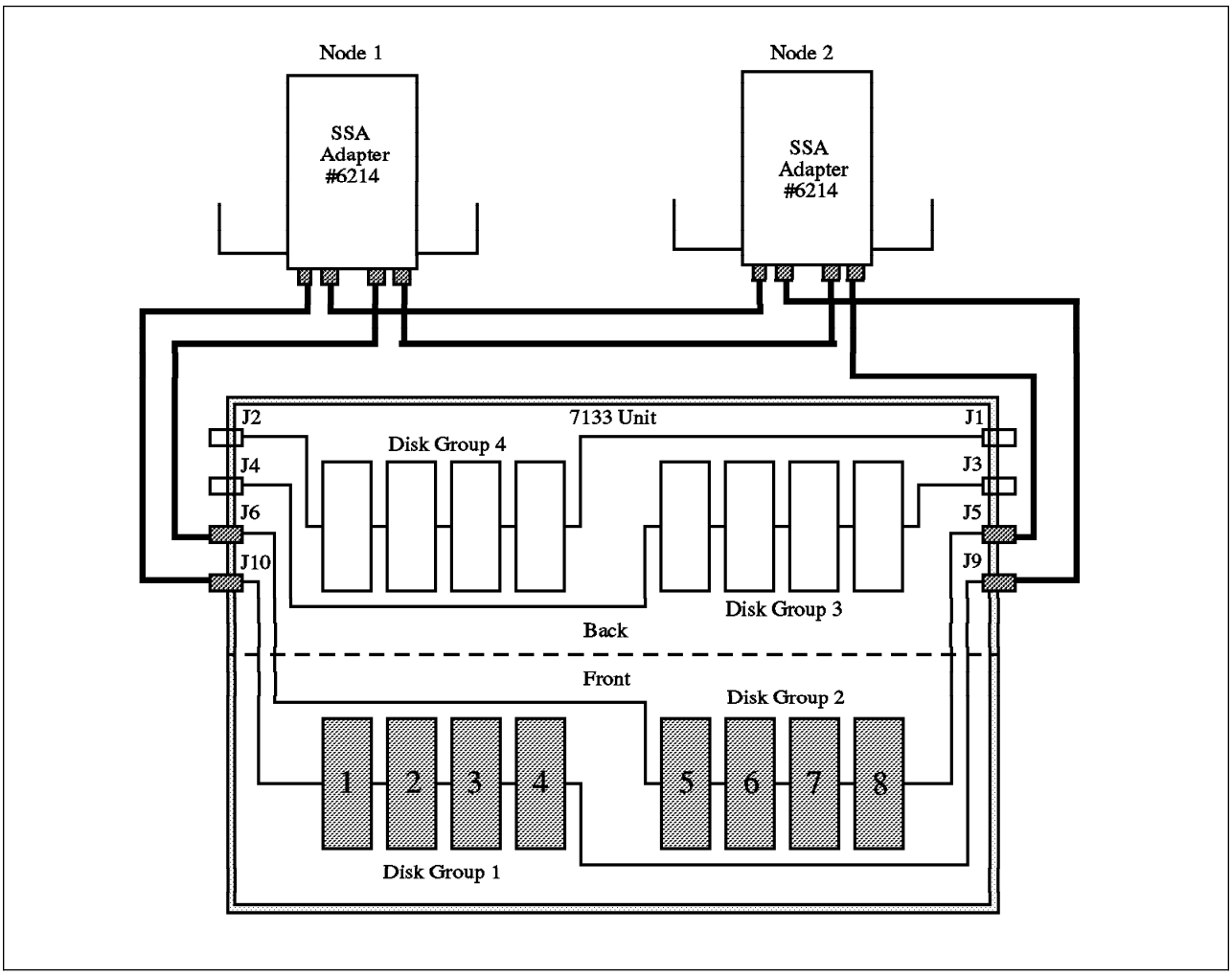


Figure 63. High Availability SSA Cabling Scenario 1

The first scenario, shown in Figure 63, shows a single 7133 subsystem, containing eight disk drives (half full), connected between two nodes in a cluster. We have not labeled the cables, since their lengths will be dependent on the characteristics of your location. Remember, the longest cable currently marketed by IBM is 25 meters, and there are many shorter lengths, as shown in Table 8 on page 230. As we said before, all cables have the same connectors at each end, and therefore are interchangeable, provided they have sufficient length for the task.

In the first scenario, each cluster node has one SSA Four Port Adapter. The disk drives in the 7133 are cabled to the two machines in two loops, the first group of four disks in one loop, and the remaining four in the other. Each of the loops is connected into a different port pair on the SSA Four Port Adapters.

In this configuration, LVM mirroring should be implemented across the two loops; that is, a disk on one loop should be mirrored to a disk on the other loop. Mirroring in this way will protect you against the failure of any single disk drive.

The SSA subsystem is able to deal with any break in the cable in a loop by following the path to a disk in the other direction of the loop, even if it does go through the adapter on the other machine. This recovery is transparent to AIX and HACMP.

The only exposure in this scenario is the failure of one of the SSA Four Port Adapters. In this case, the users on the machine with the failed adapter would lose their access to the disks in the 7133 subsystem. The best solution to this problem is to add a second SSA Four Port Adapter to each node, as shown in Figure 64 on page 233. However, this adds an amount of cost to the solution that might not be justifiable, especially if there is a relatively small amount of disk capacity involved.

An alternative solution would be to use HACMP's Error Notification feature to protect against the failure. You could define an error notification method, which is triggered on the AIX error log record on the failure of the adapter, and which would run a script to shut down the cluster manager in a *graceful with takeover* mode. This would migrate the users to the other node, from which they would still have access to the disks.

HACMP's Error Notification facility is described, with examples, in 6.8, “AIX Error Notification” on page 172.

Our second scenario, in Figure 64 on page 233, shows a second SSA Four Port Adapter added to each node. This allows each system to preserve its access to the SSA disks, even if one of the adapters were to fail. This solution does leave an adapter port pair unused on each adapter. These could be used in the future to attach additional loops, if the remaining disk locations in the 7133 were filled, and if additional 7133 subsystems were added into the loops.

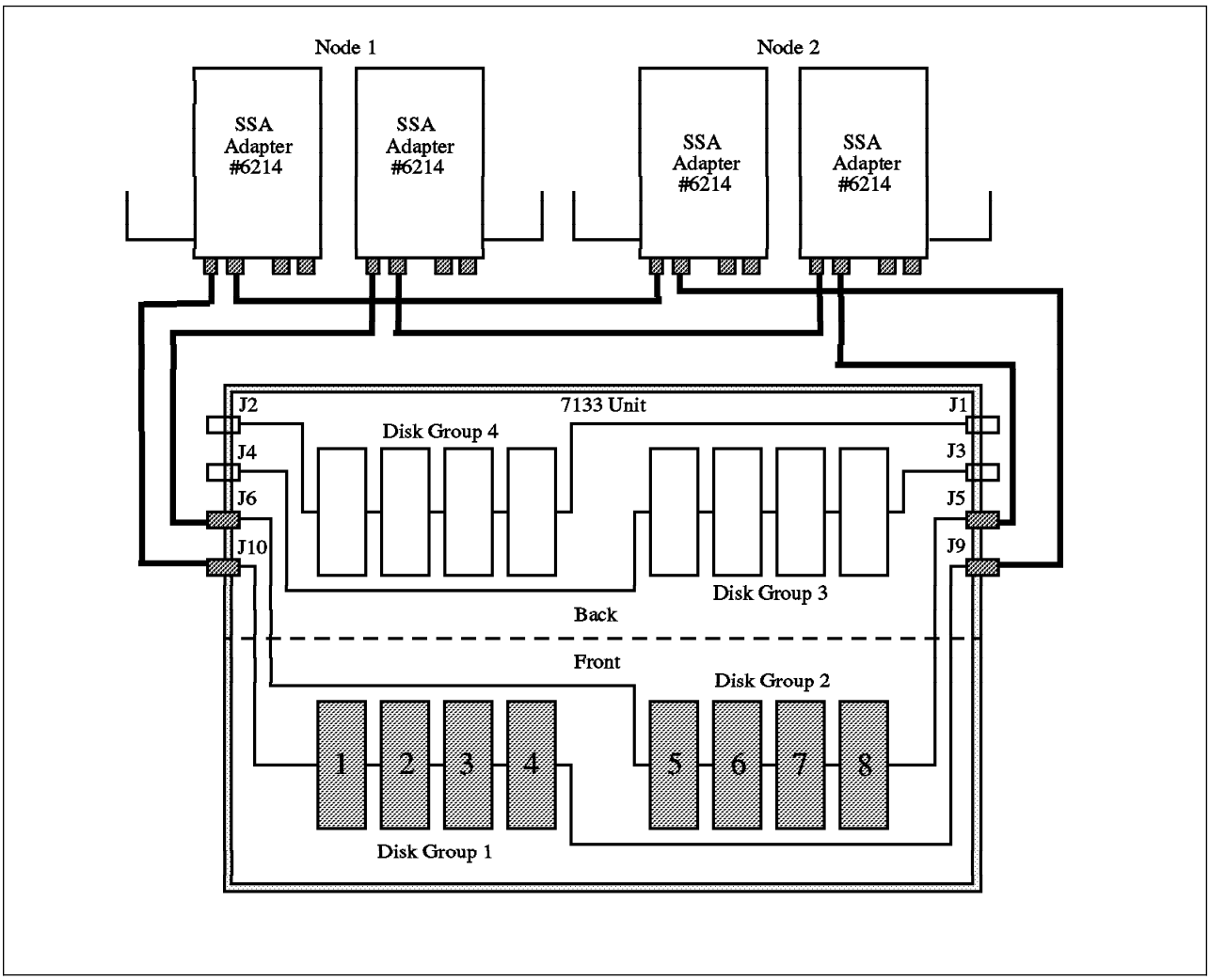


Figure 64. High Availability SSA Cabling Scenario 2

Any of the loops can be extended at any time, by reconnecting the cabling to include the new disks in the loop. If these additions are planned correctly, and cables are unplugged and plugged one at a time, this addition of disks can be done in a “hot-pluggable” way, such that the system does not have to be brought down, access to existing disks is not lost, and the new disks can be configured while the system continues running.

B.4.6 AIX's View of Shared SSA Disk Subsystems

The AIX operating system configures each disk drive in a shared SSA subsystem as a separate hdisk device on each node.

Appendix C. Measurements of Disk Reliability

One of the components most prone to failure in any single system or cluster is the disk drive. In this appendix, we will see how to interpret the reliability rating of a disk and how it relates to the probability of a drive failing.

C.1 Mean Time Between Failure (MTBF)

Mean time between failure (MTBF) is widely used as a reliability measure by suppliers of small-form-factor disk drives and is regarded throughout the industry as an effective gauge of reliability. However, it is important that you be able to interpret the relationship between MTBF claims and actual drive reliability.

In general, higher MTBF correlates with fewer drive failures; but an MTBF claim is not a guarantee of product reliability and does not represent a condition of warranty.

C.1.1 Defining Mean Time Between Failure

MTBF is the mean of a distribution of product life times, often estimated by dividing the total operating time accumulated by a defined group of drives within a given time period, by the total number of failures in that time period.

The defined group of drives is a group of drives that:

- Have not reached end-of-life (typically five to seven years)
- Are operated within a specified reliability temperature range under specified normal usage conditions
- Have not been damaged or abused

A failure is any event that prevents a drive from performing its specified operations, given that the drive meets the group definition described above.

This includes drives that fail during shipment and during what is frequently referred to as the *early life period* (failures typically resulting from manufacturing defects).

It does not include drives that fail during integration into Original Equipment Manufacturer (OEM) system units or as a result of mishandling, nor does it encompass drives that fail beyond end-of-life.

C.1.2 Mean Time Between Failure and Actual Failures.

If you purchase a drive with an MTBF of one million hours (114 years), it does not mean that the drive will operate without failure for that period of time. That is because the drive will reach end-of-life before reaching one million hours.

For example, a continuously operated drive with a five-year useful life will reach end-of-life in less than 45000 hours. But, theoretically, if the drive is replaced by a new drive when it reaches end-of-life, and the new drive is replaced with another new drive when it reaches end-of-life, and so on, then the probability that a million hours would elapse before a failure occurs would be greater than 30 percent in most cases.

Let us now consider the case of multiple drives. If you purchase 1000 drives with an MTBF of a million hours each, let us calculate the number of disks that you can expect to fail over a five-year period.

Assuming that any failed drive is replaced with a new drive having the same reliability characteristics and that the drives are used continuously, then the number of failures, **r**, you can expect is:

$$r \simeq \frac{(1000 \text{ drives}) \times (43{,}800 \frac{\text{hours}}{\text{drive}})}{1{,}000{,}000 \frac{\text{hours}}{\text{failure}}}$$

$$\simeq 44 \text{ failures}$$

Note!

In this example because of statistical variation, there is approximately a 90 percent probability that the actual number of failures will be between 33 and 55.

If the drives are operated for 16 hours per day instead of 24 hours per day, then the number of failures you can expect is:

$$r \simeq \frac{(1000 \text{ drives}) \times (29{,}200 \frac{\text{hours}}{\text{drive}})}{1{,}000{,}000 \frac{\text{hours}}{\text{failure}}}$$

$$\simeq 29 \text{ failures}$$

C.1.3 Predicted Mean Time Between Failure

It is very costly and time-consuming to actually measure high MTBF's with a reasonable degree of precision. Therefore, to assess the reliability of a new disk drive prior to volume production, reliability data from past products and component and assembly tests is merged to create a mathematical model of the drive reliability characteristics. The outcome of this modeling process is the predicted MTBF.

After volume production gets under way, actual field failure data is used to check the validity of the model. If you buy drives that have a predicted MTBF of a million hours and these drives meet the conditions stated earlier in the definition of a defined group, it is quite likely that these drives will actually achieve a million hours MTBF.

The actual MTBF measured from any specific set of drives will depend on the usage and the environmental conditions the drives experience. Stressing a drive beyond normal usage conditions may reduce the actual MTBF to a point below the predicted MTBF.

Some of the conditions under which MTBF, and consequently reliability, of a drive may reduce are:

- Warm environments with poor airflow
- A high seek rate operational environment
- Being part of portable equipment, and hence, subject to higher levels of shock and vibration

Furthermore, because MTBF can only be measured using statistical methods, any measurement will be subject to statistical variation. The degree of variation is inversely proportional to the number of drives included in the measurement.

C.1.4 Comparison of MTBF Figures

There are no established industry standards for calculating or reporting MTBF. So, while comparing MTBF figures for drives produced by different vendors, you must ensure that the assumptions behind the claims are the same.

C.2 Cumulative Distribution Function (CDF)

Cumulative Distribution Function (CDF) is a mathematical function using which you can define the probability that a drive will fail prior to some point in time. For example, a drive with a CDF equal to four percent at five years has a four percent chance of failing sometime within its first five years of operation.

CDF can also be used to determine the number of expected failures from a group of drives. For example, say that 1000 drives are put into service simultaneously. If the CDF equals four percent at five years, then four percent, or 40, drives can, on average, be expected to fail after five years. It should be noted that if, when a drive fails, it is replaced with a new drive, the total number of failures over the five year period will, on average, be higher than 40 since some of the replacement drives may also fail.

List of Abbreviations

ADSM/6000	Adstar Distributed Storage Manager/6000
AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report The description of a problem to be fixed by IBM defect support. This fix is delivered in a PTF (see below).
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
AS/400	Application System/400
CDF	Cumulative Distribution Function
CD-ROM	Compact Disk - Read Only Memory
CLM	Cluster Lock Manager
CLVM	Concurrent Logical Volume Manager
CPU	Central Processing Unit
CRM	Concurrent Resource Manager
DE	Differential Ended
DLC	Data Link Control
DMS	Deadman Switch
DNS	Domain Name Service
DSMIT	Distributed System Management Interface Tool
FDDI	Fiber Distributed Data Interface
F/W	Fast and Wide (SCSI)
GB	Gigabyte
GODM	Global Object Data Manager
GUI	Graphical User Interface
HACMP	High Availability Cluster Multi-Processing
HANFS	High Availability Network File System
HCON	Host Connection Program
IBM	International Business Machines Corporation
I/O	Input/Output
IP	Interface Protocol
IPL	Initial Program Load (System Boot)
ITSO	International Technical Support Organization
JFS	Journaled Filesystem
KA	Keepalive Packet
KB	Kilobyte
kb	kilobit
LAN	Local Area Network
LU	Logical Unit (SNA definition)
LUN	Logical Unit (RAID definition)
LVM	Logical Volume Manager
MAC	Medium Access Control
MB	Megabyte
MIB	Management Information Base
MTBF	Mean Time Between Failure
NETBIOS	Network Basic Input/Output System
NFS	Network File System
NIM	Network Interface Module **Note:** This is the definition of NIM in the HACMP context. NIM in the AIX 4.1 context stands for Network Installation Manager.
NIS	Network Information Service
NVRAM	Non-Volatile Random Access Memory
ODM	Object Data Manager
PAD	Packet Assembler/Disassembler
POST	Power On Self Test
PTF	Program Temporary Fix A fix to a problem described in an APAR (see above).
RAID	Redundant Array of Independent (or Inexpensive) Disks
RISC	Reduced Instruction Set Computer
SCSI	Small Computer Systems Interface
SLIP	Serial Line Interface Protocol

SMIT System Management Interface Tool

SMP Symmetric Multi-Processor

SMUX SNMP (see below) Multiplexor

SNA Systems Network Architecture

SNMP Simple Network Management Protcol

SOCC Serial Optical Channel Converter

SPOF Single Point of Failure

SPX/IPX Sequenced Package Exchange/Internetwork Packet Exchange

SRC System Resource Controller

SSA Serial Storage Architecture

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Interface Protocol

UDP User Datagram Protocol

UPS Uninterruptible Power Supply

VGDA Volume Group Descriptor Area

VGSA Volume Group Status Area

WAN Wide Area Network

Index

Special Characters

/etc/filesystems
  filesystem helper 188
  jfslog 186
/etc/inittab
  cluster problems 190
  entries 190
  IP takeover 190
  run level 190
/tmp filesystem
  permissions 191

Numerics

9333 optical extender cabling RPQ 71
9333 Serial-Link Disk Subsystem
  benefits 70
  compatibility with SCSI-2 72
  use with HACMP/6000 33
9334 SCSI Disk Subsystem
  use with HACMP/6000 32

A

abbreviations 239
acronyms 239
AIX
  dynamic kernel 12
  update facilities 13
application server definition 120
ARP cache
  Clinfo 191
  hardware address swapping 114, 191
  IP takeover 191
asynchronous trap option, clinfo 203
automated operations 22
availability
  availability continuum 6
  base availability 3
  continuous availability 4
  features of AIX 11
  high availability 4
  importance of 2
  improved availability 4
  levels of 3
  measurements of 7
  single system 11
  techniques to enhance 19
  threats to 6

B

bad block relocation 12
boot adapter 34, 114

C

cabling
  7134-010 High Density SCSI Disk Subsystem 217
  7135-110 or 7135-210 RAIDiant Array 219
  7137 Model 412, 413, 414, 512, 513, and 514 Disk Array Subsystems 222
  7204 Model 315, 317, and 325 External Disk Drives 214
  7204-215 External Disk Drive 213
  9333 Serial-Link Subsystems 224
  9334-011 and 9334-501 SCSI Expansion Units 215
cascading resource groups 42, 121
change management 21
  additional communications connectivity 180
  additional filesystems 179
  cluster maintenance 179
  cluster verification 176
  new applications 180
  software fixes 176
  software upgrades 176
clinfo (also see HACMP/6000) 30, 37
clstat program 55
cluster lock manager (also see HACMP/6000) 30
cluster manager
  changing startup parameters 167
  cycles_to_fail 166
  failure detection rates 168
  HACMP/6000 Version 2.1 166
  HACMP/6000 Version 3.1 168
  KA_rate 168
  missed_KA 168
  normal_ka_cycles 166
  pinning 171
  startup parameters 166
cluster manager (also see HACMP/6000) 29
cluster shutdown modes 49
cluster tuning 159
cluster.log file 53
concurrent resource groups 44
config_too_long event 53
Configuration Manager (cfgmgr) 13
cross mount 124
cycles_to_fail 166, 168

D

deadman switch (DMS)
  crash command 160
  disabling 165
  false takeover 160
  I/O pacing 163
  keepalive packets 160
  sync daemon 165
  timeout 162, 169
  timer 160
disk mirroring 4, 12
  mirroring with RAID 17
  quorum 184
  root volume group 180
dual-active configuration 196

E

error detection 13
error notification 172
events
  cluster status events 53
  customization 55
  logfiles 53
  network events 50
  node events 49

F

fail_standby event 51
failure rate 8
false take-over
  deadman switch 160
  pinning the Cluster Manager 171
fault tolerance 5, 25
filesystem helper 188
forced shutdown 50
fsck Command 12

G

graceful shutdown 49
graceful shutdown with takeover 49

H

HACMP
  clinfo 30, 37
  cluster lock manager 30
  cluster manager 29
  disk space requirements 60
  hardware requirements 30
  network interface 29
  network interface types 34
  shared disk devices 29
  shared SCSI disk setup 91
HACMP/6000 Version 2.1
  changing startup parameters 167
  startup parameters 166
HACMP/6000 Version 3.1
  failure detection rates 168
hacmp.out file 54
hardware address swapping 114
hot pluggable disk drive 4, 15
hot standby configuration 46

I

I/O Pacing 162
  deadman switch 163
  use of 163
IP takeover
  ARP cache 191
isolation

J

jfslog 12
join_standby event 53
journaled filesystem (jfs) 12
  /etc/filesystems 186
  logform 98, 188

K

KA_rate 168
keep-alive packets
keepalive packets 29, 88

L

lock value block 206
logical volume
  name conflict 187
Logical Volume Manager (LVM) 11
lvlstmajor command 97

M

major failure 201
major number 97
Mean Time Between Failure
  definition 7
minor failure 200
missed_KA 168
multi-tailed disks 15
mutual takeover configuration 47

N

network interface (see HACMP)
Network Interface Module (NIM) 109

NFS cross mount 124
NIM (Network Interface Module) 109
node environment definition
  application server definition 120
  resource group definition 121
node isolation 112
normal_ka_cycles 166, 167

O

outage, planned and unplanned 2

P

passthru terminator cable 65
permissions
  /tmp filesystem 191
phantom disks 189
power supply
  backup power supply 14
  power conditioning 14
  Uninterruptible Power Supply (UPS) 14
private network 35, 114
problem management 22
public network 35, 114

Q

quorum 182
  disk mirroring 184
  volume group descriptor area 181, 184

R

RAID disk arrays 15
recovery time 8
redundancy 15
Request for Price Quotation (RPQ)
  9333 optical extender cabling 71
resource group definition 121
resource groups
  cascading 42
  concurrent 44
  rotating 43
rotating resource groups 43
rotating standby configuration 46

S

SCSI reserve
serial network 35, 114
serial networks 80
Serial Storage Architecture (SSA) 73
service adapter 34, 114
service level 1
  service level management 22
shared disk cabling
  7134-010 High Density SCSI Disk Subsystem 217
  7135-110 or 7135-210 RAIDiant Array 219
  7137 Model 412, 413, 414, 512, 513, and 514 Disk Array Subsystems 222
  7204 Model 315, 317, and 325 External Disk Drives 214
  7204-215 External Disk Drive 213
  9333 Serial-Link Subsystems 224
  9334-011 and 9334-501 SCSI Expansion Units 215
shared logical volume
  name conflict 187
shared volume groups 15
shutdown modes 49
single point of failure
  definition 4
single points of failure, eliminating 26
Small Computer Systems Interface (SCSI)
  arbitration 63
spatial reuse 75
standby adapter 34, 114
swap_adapter event 51
Symmetric Multi-Processor (SMP) 59
sync daemon 164
  deadman switch 164
  I/O buildup 164
System Management Interface Tool (smit) 11
System Resource Controller (SRC) 13
  starting Cluster Manager 165
system test environment 21
systems management
  automated operations 22
  capacity management 20
  change management 21
  isolation 23
  operations management 20
  performance management 22
  problem management 22
  recovery management 23
  service level management 22
  skills management 20
  system test environment 21

T

takeover
  disk 38
  IP address 38
target mode, SCSI 78, 80
third-party takeover configuration 47
time synchronization 192
timeserver 192
tuning
  false takeover 159
  I/O pacing 162
  kernel lock 162

tuning *(continued)*
LED 888 159
system I/O 162
when to do 159

U

Uninterruptible Power Supply (UPS) 4, 14
unstable_too_long event 53
upgrade of cluster from HACMP/6000 Version 2.1 to HACMP/6000 Version 3.1 132

V

VGDA 181
volume group descriptor area 181

ITSO Technical Bulletin Evaluation
RED000

International Technical Support Organization
High Availability on the RISC System/6000 Family
October 1995

Publication No. SG24-4551-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:
   Do you provide billable services for 20% or more of your time? Yes____ No____
   Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____
   If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

Name _____ Address _____

Company or Organization _____

Phone No. _____

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department JN9, Building 821
Internal Zip 2834
11400 BURNET ROAD
AUSTIN TX
USA 78758-3493

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.