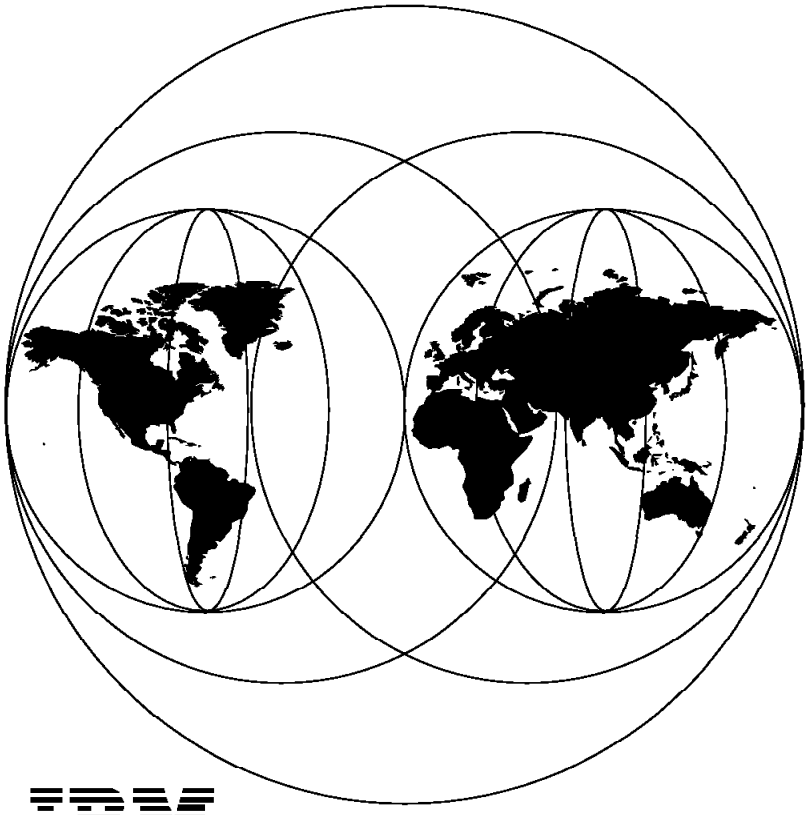
SG24-4742-00

Implementing High Availability on RISC/6000 SP

July 1996



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4742-00

Implementing High Availability on RISC/6000 SP

July 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 223.

First Edition (July 1996)

This edition applies to Version 4 Release 1 Modification 1 of HACMP for use with AIX 4.1 and RISC/6000 SP, PSSP Version 2 Release 1.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

High availability is of vital importance when running commercial and scientific applications in the RISC/6000 SP environment. Single points of failure within a system can seriously impact both performance and availability.

This redbook examines identified single points of failure, and provides detailed solutions for correcting these problems. Examples of recommended solutions that you can implement on your system are provided, as well as sample configuration scenarios.

In addition, the redbook describes High Availability Control Workstation (HACWS) implementation, the benefits of using Recoverable Virtual Shared Disk (RVSD) with HACMP, and various network and switch considerations.

This redbook will be a valuable resource to customers, system engineers and field support specialists who need to plan, configure, and implement high availability solutions within a RISC/6000 SP environment. Some knowledge of HACMP and RISC/6000 is assumed.

How This Redbook Is Organized

This redbook contains 235 pages. It is organized as follows:

- Chapter 1, “Single Points of Failure and Solutions”

  This chapter identifies the single points of failure associated with the RISC/6000 SP system and the recommended solutions. It provides a matrix that outlines the single points of failure and the possible hardware or software solutions. It presents the ideal solutions suitable for a RISC/6000 SP system where high availability is of vital importance.

- Chapter 2, “High Availability Control Workstation (HACWS)”

  This provides the detailed implementation of the high availability control workstation (HACWS). It contains all of the requirements needed to implement HACWS and the models of the RISC/6000 workstation that could be considered suitable for use as a backup control workstation.

- Chapter 3, “HACMP for RISC/6000 SP”

  This chapter addresses the recommended software solutions to some of the single points of failure within the RISC/6000 SP environment utilizing High Availability Cluster Multi-Processing. It contains some scenarios of implementing the Eprimary takeover, High Performance Switch adapter failure and dual network failover using the FDDI network.

- Chapter 4, “Network Considerations”

  This chapter focuses on the High Performance Switch and provides insight into some of the factors that should be considered when implementing a highly available RISC/6000 SP system. It deals with the different types of switches associated with the RISC/6000 SP system, the switch chip layout at

the hardware level and the importance of spreading the system resources on nodes that are in different switch chips.

- Chapter 5, “Implementing HACMP with RVSD on the RISC/6000 SP”

  This chapter provides an overview of the Virtual Shared Disk (VSD) architecture and the different communication layers used by VSD. It contains a detailed description of the Recoverable Virtual Shared Disk (RVSD) and the benefits of using RVSD together with HACMP. It uses a table matrix to explain the comparison between AHCMP and RVSD. Also it depicts some scenarios of VSD server and client configurations during mutual takeover in a RISC/6000 SP system.

- Chapter 6, “Implementing LoadLeveler for High Availability”

  This chapter describes the overview of the LoadLeveler and some of the built in capabilities for high availability that it provides. It further discusses how the overall system availability can be enhanced with the combination of High Availability Cluster Multi-Processing. It provides a sample scenario of how to best configure the scheduler to be highly available in the event of a failure.

The Team That Wrote This Redbook

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

This project was designed and managed by:

Endy Chiakpo
International Technical Support Organization, Poughkeepsie Center

The authors of this redbook are:

Gilles Eberhardt
IBM France

Rey Rodriguez
IBM US Dallas System Center

Zarah Smith
IBM UK

Hisashi Shirai
IBM Japan

Kin Tsang
IBM Hong Kong

Endy Chiakpo
International Technical Support Organization, Poughkeepsie Center

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

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

Marcelo Barrios
International Technical Support Organization, Poughkeepsie Center

David Thiessen
International Technical Support Organization, Austin Center

IBM PPS Lab Poughkeepsie:
Deepak Advani
Mark Atkins
Mike Browne
Bill Carlson
Richard Coppinger
Dave Folsom
Derrick Garmire
Kathy Lange
Dr. Gili Mendel
John Stephenson

IBM EMEA High End Center of Competence UK:
Andy McMenemy

IBM EMEA PSSC Montpellier France:
Eric Le Bail

IBM EMEA Center of Competence Munich Germany:
Bernhard Buehler

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Single Points of Failure and Solutions

This chapter identifies the potential single points of failure associated with the RISC/6000 SP and the possible solutions recommended to eliminate the single points of failure within the RISC/6000 SP system. Table 1 on page 3 is used to identify the most commonly encountered potential single points of failure in this environment, and the various recommended solutions to address them. Most of these recommended solutions are implemented in the subsequent chapters in this redbook. However, it should be understood that the RISC/6000 SP system is very reliable and so robust that the mean time between failure rate for most of the critical components such as the High Performance Switch (HiPS) is significantly high.

It is possible to eliminate most single points of failure without using any layered software product or additional hardware. As you go through this section and the subsequent chapters, 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 down time, and can provide the necessary human intervention during all hours of operation. This will help you in deciding whether you need another software product or additional hardwares to meet your availability requirements, or whether you can manage with manual intervention.

1.1.1 Disadvantages of Manual Intervention

The time to recover manually from any failure in a system 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 or hardware to meet your availability needs. If your environment cannot stand outages of more than a few minutes you should consider some of the recommendations and implementations in this redbook.

1.1.2 Definition of High Availability

High availability is the system management strategy of quickly restoring essential services in the event of system, component, or application failure. The goal is minimal service interruption, not fault tolerance.

High availability views availability as a set of system-wide, shared resources which cooperate to back up, or act as standby, for each other. It uses standard hardware and software techniques instead of specialized and imbedded hardware and software.

High availability is intended for applications that can withstand a short interruption should a failure occur. It is also for those who are willing to absorb some down time rather than pay the much higher costs of providing fault tolerance. But it is not for time-critical applications where a few seconds of down time would be disastrous.

1.1.3 Single Points of Failure

The RISC/6000 SP system is made up of many components that are essential for providing services to the end user. If the failure of any single component results in the unavailability of service to the end user, this component is called a *single point of failure* for the system.

The potential single points of failure that the RISC/6000 SP system could have are listed in the following table. They are listed according to the following RISC/6000 SP subsystems:

Frame

High Performance Switch

Node

Data

Control Workstation

The application called *LoadLeveler for AIX* is included in the matrix and in this redbook to illustrate how a distributed application can be made highly available using its own availability features and High Availability Cluster Multi-Processing.

While the goal is to eliminate all potential single points of failure, compromises may have to be made. There is usually a cost associated with eliminating a single point of failure. This cost needs to be weighed against the cost of losing services should that component fail. The goal of high availability is to provide a cost-effective, highly available computing platform than can grow to meet future processing demands.

Listed in Table 1 on page 3 are some of the solutions that address these potential points of failure. The first column in the matrix captioned ″Potential single points of failure″ lists the individual system components that can be potential single points of failure within the RISC/6000 SP system. The rest of the columns in the first row are the hardware and software applications that can be used as possible solutions to each of the components identified as a potential single point of failure, and they are:

- HACMP
- HACWS
- RVSD
- AIX
- Hardware

Each of the possible solutions represents a column heading, and under each column corresponding to an identified potential single point of failure component is one of the following:

Yes Means that the proposed solution is possible with the application or applications. For example, if we take a look at one of the identified potential single point of failure component such as the High Performance Switch, the only proposed solution is with HACMP and under this application in the matrix we have ″yes″.

Custom Means that a customized solution is possible but will have to be developed.

- Means that a solution with the application is not recommended or possible.

Table 1. Solution Matrix for Potential Points of Failure

Potential single points of failure	HACMP	HACWS	RVSD	AIX	Hardware
Frame					
• Power Supply	-	-	-	-	Yes
• Ethernet Network	-	-	-	-	Yes
• RS232 Serial Link	-	(Custom)	-	-	Yes
• Supervisor Card	-	-	-	-	Yes
High Performance Switch					
• LC8	Yes	-	-	-	-
• HiPS	Yes	-	-	-	-
• Eprimary	Yes	-	-	-	-
• HiPS Global Network	(Custom)	-	-	-	-
Node					
• Operating System	Yes	-	-	Yes	-
• Internal Disks	Yes	-	-	Yes	-
• External Disks	Yes	-	Yes	Yes	-
• HiPS Adapter	Yes	-	-	-	-
• Comm Adapter	Yes	-	-	-	-
Data					
• JFS	Yes	Yes	-	Yes	-
• Raw	Yes	Yes	Yes	Yes	-
• VSD	-	-	Yes	-	-
Control Workstation					
• Processor	Yes	Yes	-	-	-
• Data	Yes	Yes	-	Yes	-
• Disk	Yes	Yes	-	Yes	-
• RS232	Yes	Yes	-	-	-
• Ethernet	-	-	-	-	-
LoadLeveler					
• Config Files	Yes	Yes	-	Yes	-
• Central Manager	Yes	-	-	-	Yes
• Scheduler	Yes	-	-	-	-

1.2 Hardware Solutions

The ideal solution for a highly available RISC/6000 SP could be achieved by the use of additional hardware. This section describes how a RISC/6000 SP system can be configured to provide a certain level of high availability to the end users. There are a number of ways in which a RISC/6000 SP system could be configured to become highly available, and some of the methods are described below and some customization not covered in this document will be needed to implement any of the three concepts.

- Dual Frame solution
- Dual Ethernet
- Ethernet to Router Solution

1.2.1 Dual Frame System Solution

With the use of a dual frame RISC/6000 SP system, as shown in Figure 1 on page 5, the system will be protected against failures that may occur from any number of the system components. Some of these components are:

- High Performance Switch
- RISC/6000 SP Ethernet
- RISC/6000 SP Power cord cable
- RS232 Serial Link
- Supervisor Card

With the utilization of a dual frame system, in which each frame has its own High Performance Switch, power supply, and ethernet, and with the nodes in one frame serving as backup nodes for the other frame, the continuous operation of the system in the event of a failure is seemingly assured. The system as shown in Figure 1 on page 5 is an ideally configured RISC/6000 SP system that is highly available for environments where unplanned system outages are unacceptable. From the control workstation, the frame appears as a single image system. However, to ensure optimum availability, it becomes necessary to connect each frame to a different power supply (refer to 1.3, “Frame Power Failure and Recovery” on page 10). The control workstation will also be configured with a backup control workstation (HACWS). The HACWS implementation is detailed in Chapter 2, “High Availability Control Workstation (HACWS)” on page 11.

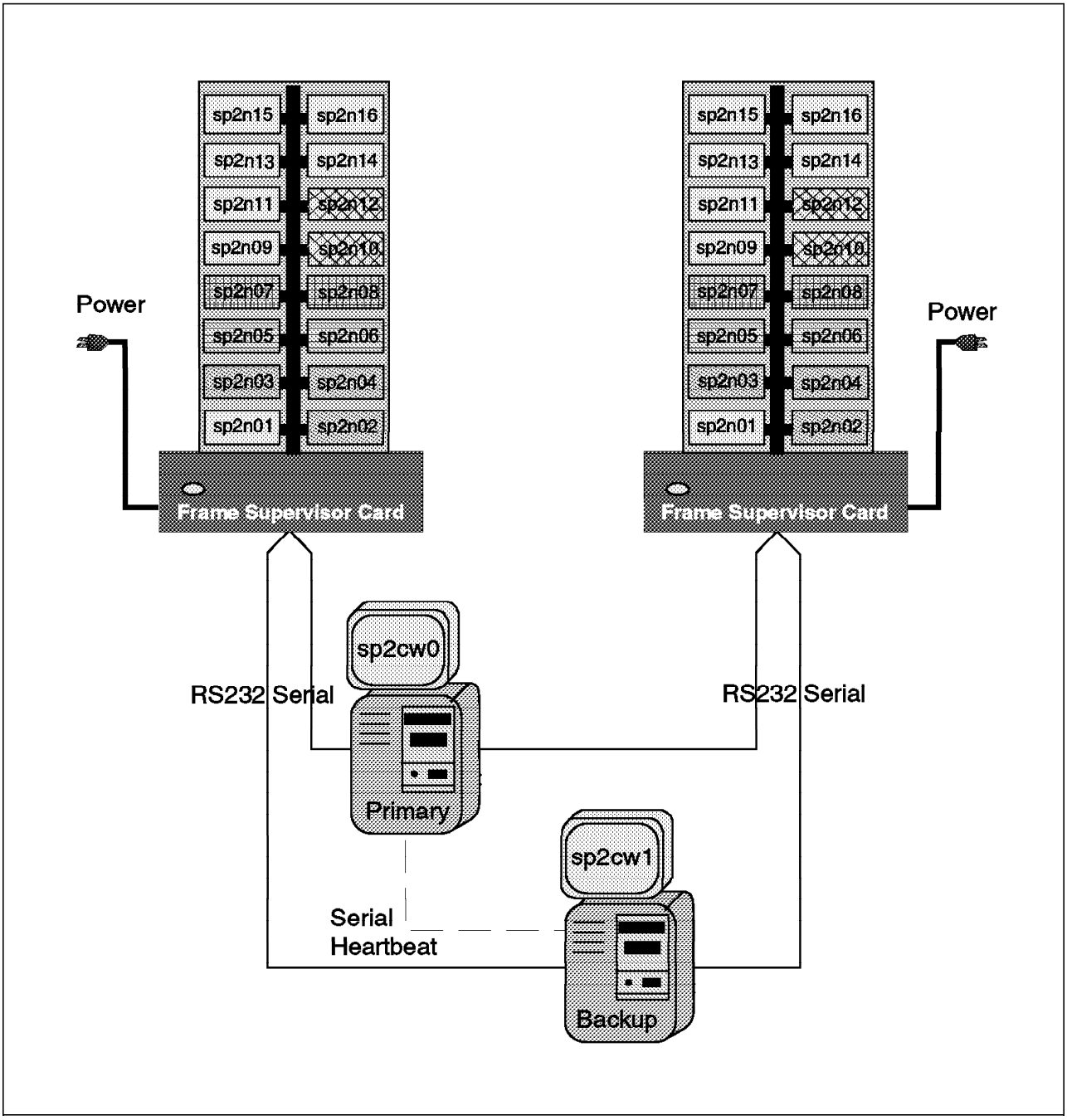


Figure 1. Configuration of a Highly Available RISC/6000 SP

1.2.2 Dual Ethernet Solution

The RISC/6000 SP Ethernet is one of the ways in which processor nodes are interconnected. The nodes are also interconnected through the High Performance Switch. However, the RISC/6000 SP Ethernet provides the communication path from the control workstation to the processor nodes, and the High Performance Switch does not provide this function from the nodes to the control workstation. Therefore, it becomes apparent that all the administrative functions on the RISC/6000 SP system, such as installing nodes, rebooting systems and mounting operating system files, are done from the control workstation to the nodes using the RISC/6000 SP Ethernet. It is also the

network used to monitor the health of the nodes. The heartbeat daemon (hb) runs on all the nodes including the control workstation, and it determines whether a node is alive or not.

In the event of an Ethernet network failure, none of the administrative tasks mentioned above can be performed and the heartbeat daemons cannot communicate. In environments where the Virtual Shared Disk (VSD) and Recoverable Virtual Shared Disk (RVSD) with Oracle database are in use, the database could be lost since the node will be declared down by the heatbeat if the Ethernet is dead.

We are faced with the challenge of how to best configure a RISC/6000 SP system in such a manner that the Ethernet network does not constitute a potential single point of failure within the system. Although we can use the High Availability Cluster Multi-Processing to monitor the RISC/6000 SP Ethernet, it cannot be used as a possible solution to recover the network. This is due to the fact that the PSSP software assumes that the IP address of the Ethernet adapter is associated with a physical node location. For this reason, the Ethernet cannot be recovered with the use of IP address takeover, which could have been implemented using High Availability Cluster Multi-Processing. One of the approaches used in this document in configuring the Ethernet network for high availability is shown in Figure 2 on page 7. In this picture, we can see that the RISC/6000 SP is a single frame system with 16 thin nodes. The control workstation, which is also configured to be highly available with HACWS, has two Ethernet adapters which are connected to different subnets. The Ethernets are connected in such a way that one of the networks serves one side of the frame with the odd number nodes, while the next one serves the other side of the frame with the even number nodes. However, when there is a failure on either adapter, the nodes affected could still be in operation and available for communication with the control workstation and other external networks to the system, through the use of the High Performance Switch and the Ethernet network of the other side of the system. This eliminates the perception of the RISC/6000 SP Ethernet as a potential single point of failure; see Figure 2 on page 7.

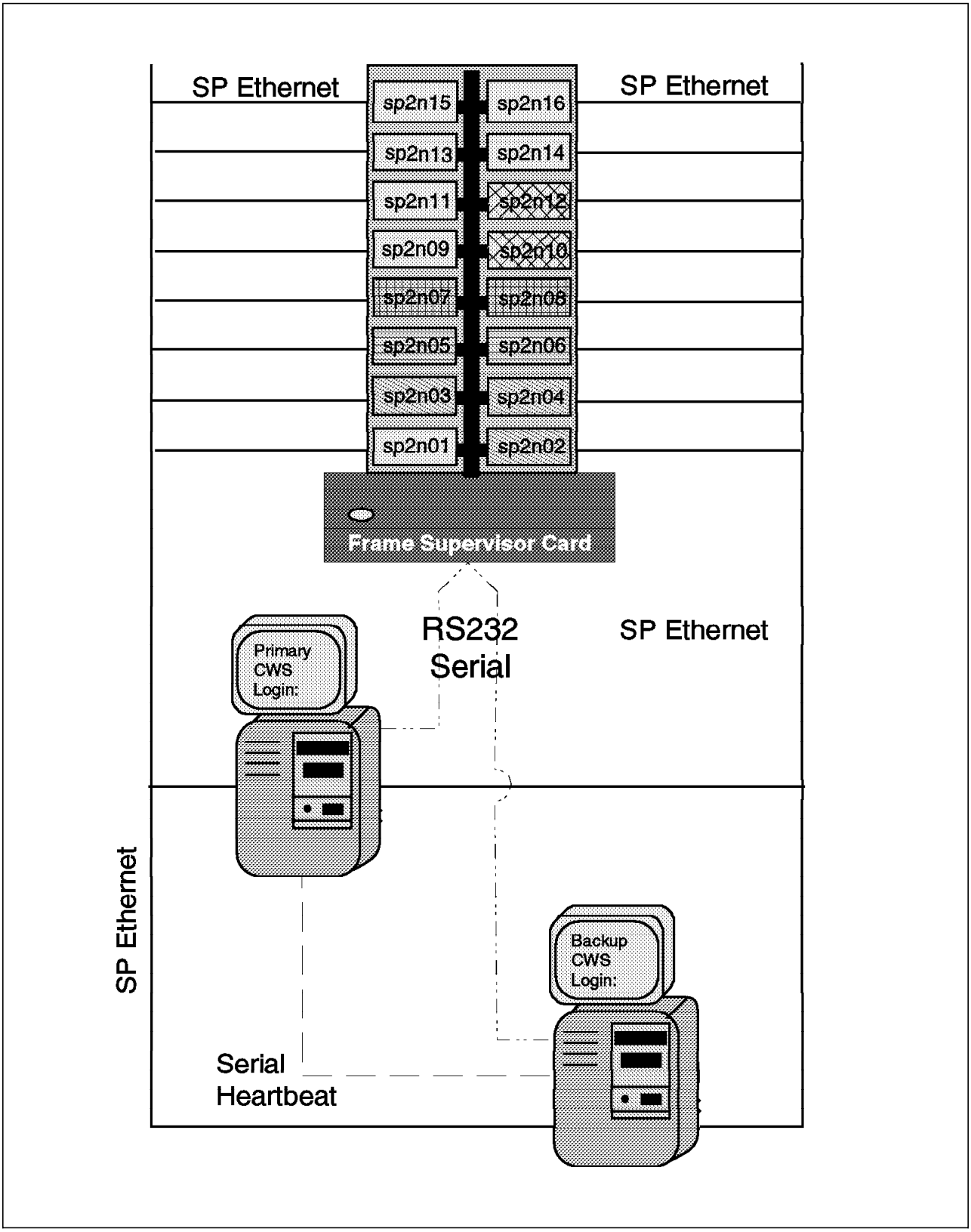


Figure 2. RISC/6000 SP Configured with Dual Ethernet

1.2.3 Ethernet to Router Solution

The use of a router to connect the RISC/6000 SP Ethernet to the control workstation provides another possible solution of configuring the Ethernet network for high availability, as shown in Figure 3 on page 9. This picture is akin to the one shown in Figure 2 on page 7 except that the control workstation has one Ethernet adapter, which is connected to a router. The control workstation is also made highly available with the use of a backup control workstation (HACWS). One segment of the system has odd number nodes (1,3,5,7,9,11,13,15), and each node has its Ethernet network configured directly to the router. The other segment of the system has even number nodes (2,4,6,8,10,12,14,16), that share a common Ethernet network, which is connected to the router.

Our reason for this choice of network configuration is to better illustrate the different methods that could be applied to eliminate or reduce the degree of ethernet outage in the system. For example, consider the case where the Ethernet network for node 1 (sp2n01) in Figure 3 on page 9 fails, perhaps due to a broken cable between the node and the router or some other fault. The fact that the Ethernet for each node on that segment of the system is directly connected to the router means that the impact of the failure will only affect node 1 (sp2n01), and the rest of the system will be operational. However, depending on the state of the node and the type of failure, access to node 1 (sp2n01) may be possible through the High Performance Switch from the other nodes in the system. It can also communicate with other external networks in the LAN with the control workstation serving as the gateway, assuming no direct link exists from the node. This kind of configuration ensures that when there is an Ethernet failure on one node, the failure will not impact the availability of the entire system or a segment of the system. Conversely, let us consider the same example for the other segment of the system with the even number nodes, and let us assume that there is a fault in the Ethernet network due to a broken cable at any point between the router to node 16 (sp2n16). It becomes apparent that this fault will affect all the nodes in this segment of the system, unlike the configuration in the segment of the system with the odd number nodes, where a similar fault will only impact the affected node. For optimum availability in a single frame system, the ethernet configuration on the segment of the system with odd number nodes would be preferable, and therefore could be implemented for both segments of the system.

The router to be used for this purpose should be the one that provides for a backup router facility. This will ensure that the router itself does not constitute a potential single point of failure in the system.

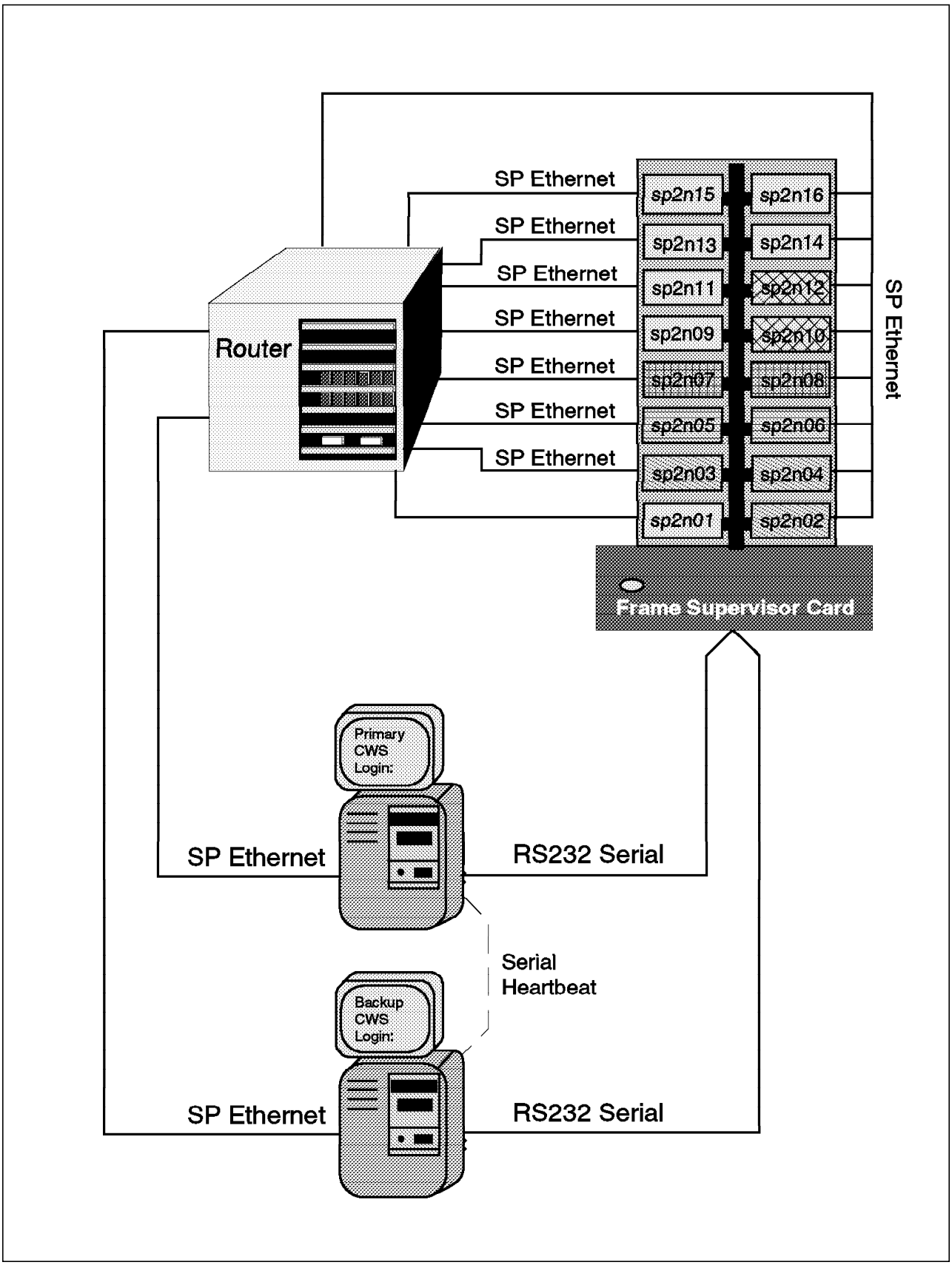


Figure 3. RISC/6000 SP Ethernet Configured with Router

1.3 Frame Power Failure and Recovery

RISC/6000 SP frames contain between one and three AC/DC 48 volt power supplies, depending on the type of frame. The N+1 feature power supply is a standard for full frame systems but, it is optional for low frames. If the N+1 feature is installed, this implies that there are at least two power supplies per frame. A failure with one of these power supplies will not interrupt the operation of the RISC/6000 SP because of the backup that will be provided by the other power supply. Furthermore, the defective power supply is hot_pluggable and can be replaced without interruption to the system. Each power supply can service up to eight nodes. With this in mind, if you have more than eight nodes per frame, it becomes necessary to consider configuring three power supplies to protect your system against unforeseen power supply failure. However, there is only one power cord from the external power network to the RISC/6000 SP. Customers should implement uninterruptible power sources (UPS) to guard themselves against a global power loss. The same is true concerning the power source for the external disks.

1.4 Internal Disk Failure and Solution

The failure of the internal disk in any of the nodes could result in the loss of data. This could have a serious implication if the node is a database or a server node upon which the operations of other nodes in the system are dependent. The solutions to protecting the failure of the internal disk are difficult and sometimes informal. One possible method would be to configure a mirrored internal disk, where the mirror is on a totally separate adapter. Since only one adapter is usually supplied on-board, this will mean installing an additional adapter and running a short cable back into the node for the second internal disk. This solution may not be cost effective in a large RISC/6000 SP system environment, and may therefore be implemented only on selected nodes in the system.

Another possible solution would be to have only AIX and paging space on the internal disks, and use external disks to store applications and database resources.

Note: While the above solutions are possibilities as a protection against internal disk failure, it is important to have in mind that rootvg mirroring is currently no longer supported with RISC/6000 SP nodes. This is due to the fact that only one bootdisk attribute can be defined in the SDR.

Chapter 2. High Availability Control Workstation (HACWS)

HACWS is a two-node configuration consisting of a primary control workstation with a backup control workstation configured as a rotating_standby HACMP cluster. The typical configuration of HACWS is shown in Figure 4.

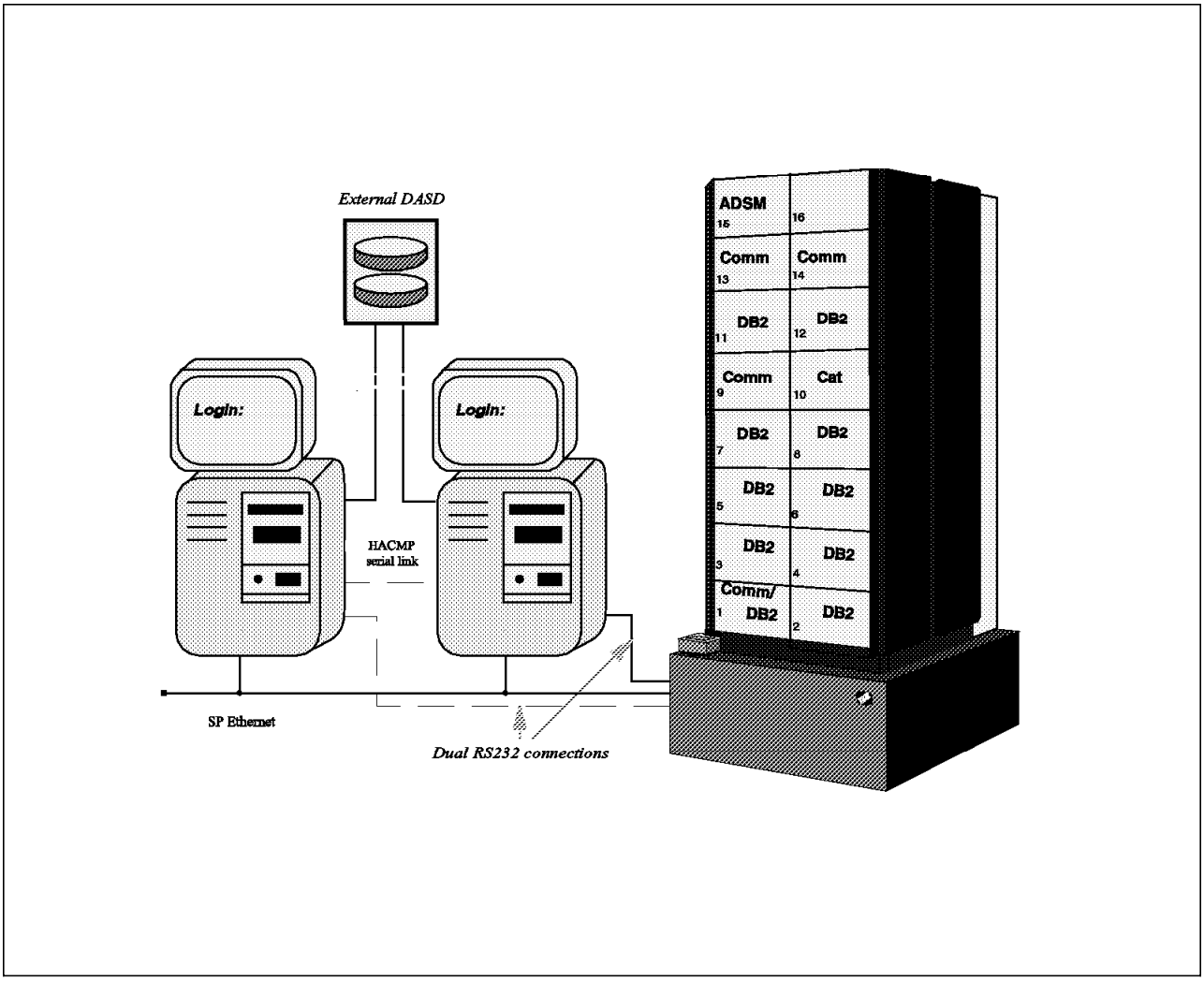


Figure 4. An HACWS Overview

This chapter provides the steps to implement the High Availability Control Workstation on the RISC/6000 SP system. Considerations for planning, detailed installation steps, and some customization samples are covered.

2.1 Planning HACWS

HACWS provides a high availability solution on the control workstation, which serves as a single point of control for managing and maintaining the RISC/6000 SP nodes using Parallel System Support Program (PSSP). The control workstation is not generally a critical resource compared to other components in the system such as the High Performance Switch. Depending on your system environment and the type of applications that are running on your control workstation, the impact of a failure in the CWS may not affect the operation of

the RISC/6000 SP. However, if your needs are such that you choose to run critical applications on the CWS, then you should consider protecting it with HACWS.

Users who do not run any critical application (for example, Name Server, LoadLeveler or DB2 Parallel Edition) on the control workstation, or do not care about spending the time to restore system backup and rebuild the control workstation after its failure, might not require HACWS. For such users, however, HACWS still provides minimum down time for maintenance of the control workstation.

The loss or failure of the control workstation will have the following effects on the management of the RISC/6000 SP system, but should have little or no impact on the active jobs on RISC/6000 SP nodes:

- It will not be possible to control the RISC/6000 SP hardware.
- System Data Repository will be unavailable.
- Existing jobs will continue to completion, but new parallel jobs cannot be started.
- No configuration changes can be made.
- Software installations cannot be done from the CWS.
- Should a switch fault occur, reset processing cannot be completed.
- Error logging of alerts raised by RISC/6000 SP nodes will be lost (though the information will still be logged on the individual nodes).
- While the control workstation is unavailable, some administrative tasks which use Parallel System Support Program (PSSP) will not be able to proceed.

2.1.1.1 Hardware Requirements

There are a number of hardware elements required to implement HACWS. They are:

1. Frame

   The following hardware is required for each RISC/6000 SP frame:

   - New Frame Supervisor Card (F/C:#1245, P/N:46H9308)
   - RS232C Y-Cable (F/C:#1245, P/N:26H7359)

   Note: Under hardware F/C #1245 a new Frame Supervisor Card and a RS232C Y-cable are provided for each SP frame.

   Figure 5 on page 13 depicts the view of the new Frame Supervisor Card and RS232C Y-cable connection. The rear view of the new Frame Supervisor Card is illustrated in Figure 6 on page 14. The LED definitions are shown in Table 2 on page 14.

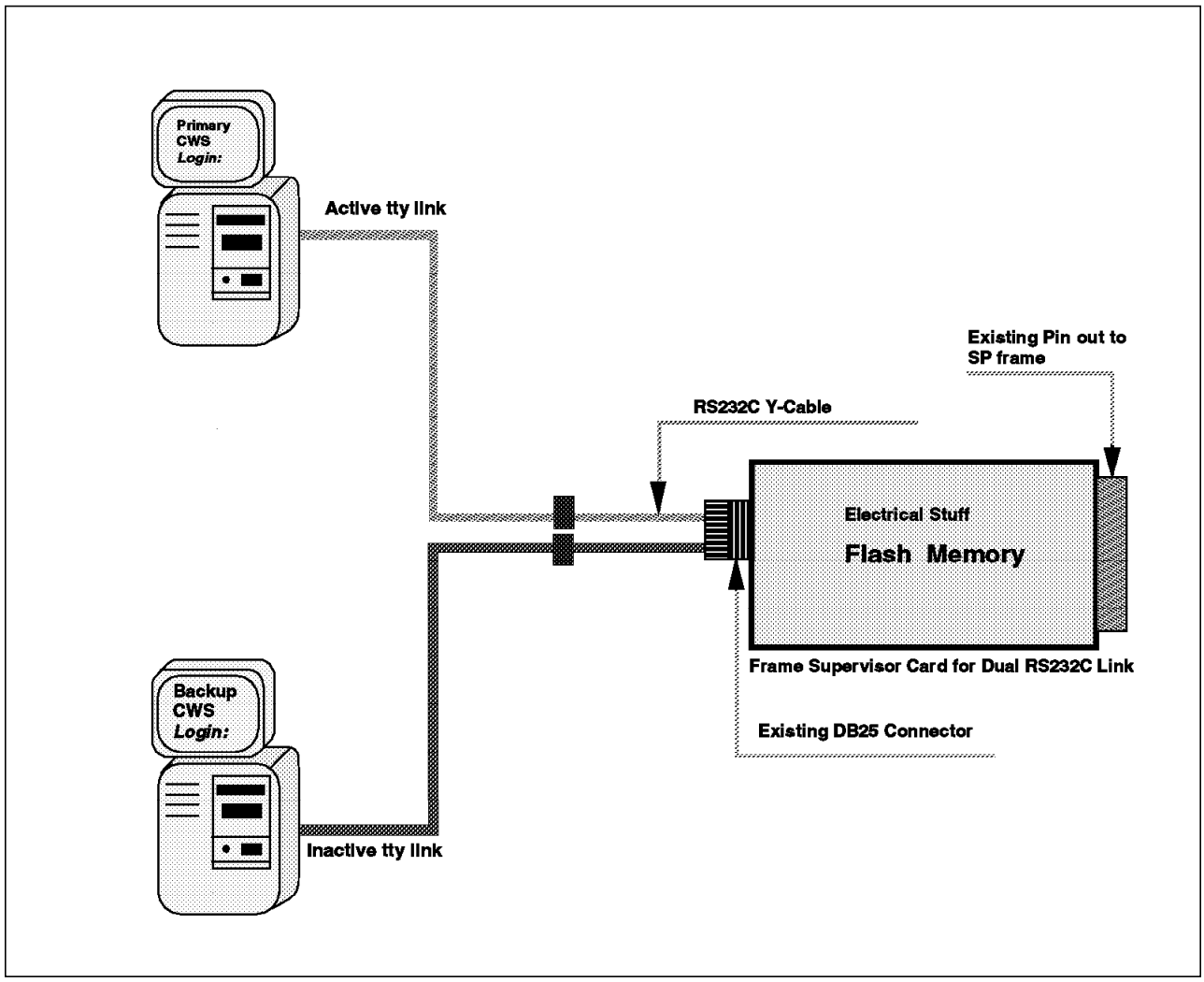


Figure 5. View of New Frame Supervisor Card and RS232C Y-Cable

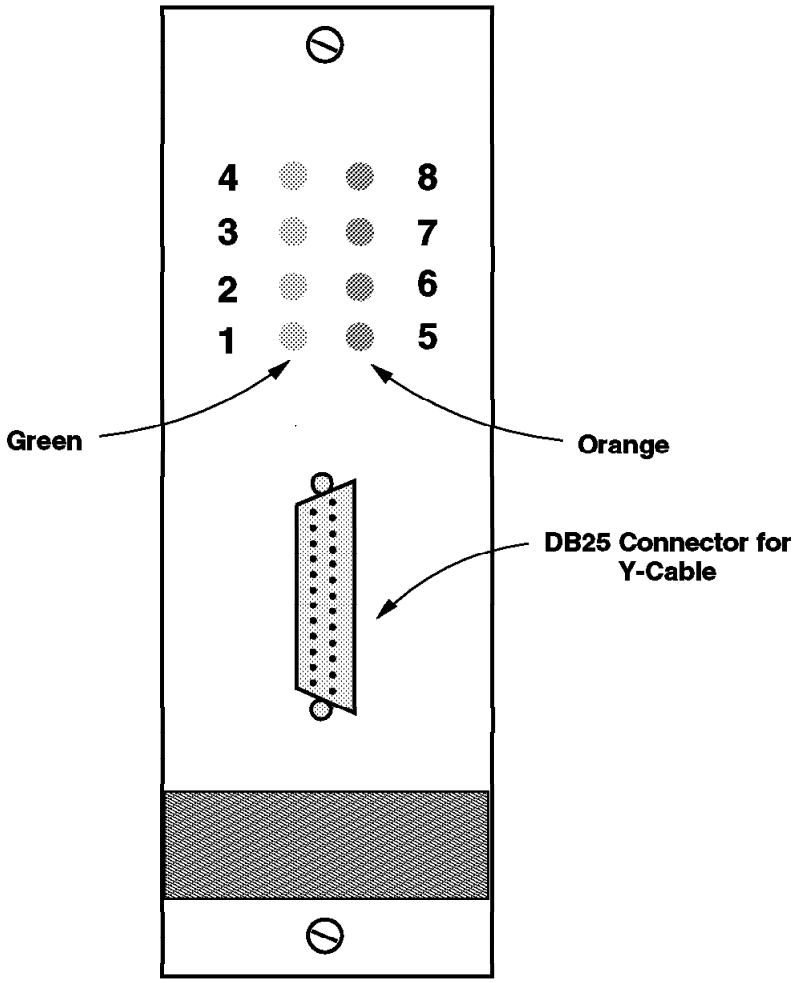


Figure 6. Rear View of New Frame Supervisor Card

Table 2 (Page 1 of 2). Frame Supervisor Card LED Definitions		
LED#	**Color**	**Signifies**
8	Orange	Control workstation current communication status (S1-RS232 data terminal ready line): • Momentarily On: Communication active • Flashing: Problem - cannot communicate with control workstation
7	Orange	Control workstation current communication status (S2-RS232 data terminal ready line): • Momentarily On: Communication active • Flashing: Problem - cannot communicate with control workstation • Off: Communicate inactive **Note:** Be aware that there may be communications to the inactive control workstation.

Table 2 (Page 2 of 2). Frame Supervisor Card LED Definitions

LED#	Color	Signifies
6	Orange	Supervisor bus communication status: • Flashing: Problem - no communication with processor nodes or switch assembly • Off: Communication okay
5	Orange	Not Used
4	Green	Connection S1 to primary control workstation • On: Active • Off: Inactive
3	Green	Connection S2 to primary control workstation • On: Active • Off: Inactive
2	Green	Bulk 48VDC power status: • On: All 48VDC power supplies operating okay • Flashing: 48VDC power supply problem • Off: No 48VDC power available
1	Green	Supervisor power status: • On: Supervisor power present • Off: No supervisor power

Note:

For LEDs #3 and #4, only one communication (S1 or S2) is active, while the other one is inactive.

2. Backup Control Workstation

   You need a RISC/6000 machine as a backup control workstation. The following shows the the current list of processors to be used as a guide in choosing a backup control workstation. It also contains the minimum memory size required for each model.

Processor	*Min. Mem. Size*	*Processor*	*Min. Mem. Size*
7009-C10	64 Meg	7013-530H	64 Meg
7009-C20	96 Meg	7013-550	64 Meg
7011-250	64 Meg	7013-55L	64 Meg
7011-25W	64 Meg	7013-560	64 Meg
7011-25T	64 Meg	7013-570	64 Meg
7012-320H	64 Meg	7013-580	96 Meg
7012-340	64 Meg	7013-58H	128 Meg
7012-340H	64 Meg	7013-590	128 Meg
7012-350	64 Meg	7013-591	128 Meg
7012-355	64 Meg	7013-59H	128 Meg
7012-360	64 Meg	7015-950	64 Meg
7012-365	64 Meg	7015-970B	64 Meg
7012-370	96 Meg	7015-980B	96 Meg
7012-375	64 Meg	7015-990	128 Meg
7012-380	128 Meg	7015-R10	64 Meg
7012-390	128 Meg	7015-R20	128 Meg
7012-39H	128 Meg	7015-R24	128 Meg
7013-520H	64 Meg		

   Note: To make the control workstation more highly available, the backup control workstation should have a different power source from the primary control workstation.

3. External Disks

The HACWS configuration requires external disks that provide a non-concurrent access feature for both primary and backup control workstations. In a typical configuration, RISC/6000 SP management data, as well as the AIX system images, PSSP and related software install filesets, NIM configuration data files, and any other software installp filesets should reside on the external shared disks. Any disks supported by HACMP V4.1.x and RS/6000 models for both the primary and backup control workstations can be used.

Note: SSA disks are not currently supported for machine model *7009*.

To avoid a single point of failure from the external disks, the use of RAID-5 disks or adopting logical volume manager (LVM) mirroring are strongly recommended. Also, two disk controllers on different power sources are recommended.

4. Network

   One of the following network connections must exist between the primary and backup control workstations:

   - A dedicated TCP/IP network link
   - An RS232C tty link (F/C:3124 or 3125)
   - The target mode SCSI across the external SCSI disks

   To avoid having the TCP/IP subsystem become a single point of failure, the use of the RS-232C link or target mode SCSI link is strongly recommended.

   Each control workstation requires the same number of connections to the RISC/6000 SP Ethernet on the same LAN segments. Each RISC/6000 SP Ethernet LAN segment must be cabled to the same Ethernet (*enx*) adapter. Standby network adapters are optional in the HACWS configuration. However, the presence of a standby adapter may avoid the need to failover (switch to backup) to the inactive control workstation with single LAN adapter failure.

   Attention

   Do not configure standby adapters in the current release of HACWS (2.1.0.0) as this is not supported. Future releases will address this requirement.

   If you require any other network LAN connections on primary and backup control workstations except SP Ethernet, you can configure those LANs as HACMP cluster networks, as long as slots for network adapters on both the primary and backup control workstations are available.

2.1.1.2 Software Requirements

The following software is required on both the primary and the backup control workstation to implement the High Availability Control Workstation:

- AIX V4.1.3 or later
- HACMP for AIX V4.1.X
- PSSP V2.1 HACWS Feature (Prod. No. 5765-529: High Avail. control workstation - ssp.hacws. 02.01.00.00

Note: If you use HACMP V4.1.1 then AIX V4.1.4 and HACWS APAR Ix57291 are required.

The following PSSP filesets are required for installing the High Availability Control Workstation. The *level*s of filesets shown in the list, however, indicate

the levels of filesets used in our testing environment. It is recommended to use the latest PTF set of PSSP.

Name	*Level*
ssp.sysman	2.1.0.5
ssp.sysctl	2.1.0.1
ssp.gui	2.1.0.4
ssp.basic	2.1.0.8
ssp.clients	2.1.0.4

2.1.2 Planning Configuration and Scenarios

Before starting to install and customize HACWS, it is a good idea to spend some time to plan the HACWS configuration and failover scenario. The recommended steps you should follow to plan your HACWS are as follows:

1. Identify single points of failure on your control workstation.

   In this step, you should identify all single points of failure in your control workstation that has no backup control workstation yet. All single points of failure in hardware and software, including operating system and applications, should be listed here.

2. Plan Backup CWS.

   Determine the RISC/6000 model used for backing up the control workstation. You can choose any model from the list in 2.1.1.1, “Hardware Requirements” on page 12, according to the CPU speed and memory size required by the applications that you want to run on the control workstation, and the number of micro channel slots that fulfill your high availability requirements.

3. Plan Networks.

   Draw a diagram of networks required by HACWS (SP Ethernet, tty link wit h Frame, and RS232C link between control workstations, and so forth); then plan how you can avoid the single points of failure on these networks.

   Note: SP Ethernet cannot be redundant, but you can minimize the impact of the SP Ethernet failure by dividing the SP Ethernet into multi-segments.

   TCP/IP addresses for control workstations should be assigned in this step. For the addressing requirements for HACWS, please refer to 2.2.1.6, “Step 6: Plan Network Configuration” on page 24.

4. Plan Shared Disks.

   At first, determine what kind of data in addition to */spdata/* filesystem should be located on the external shared disks. After that, choose how to implement the high availability on the disks. Currently, LVM mirroring and RAID 1 or 5 are your choices. Data access throughput and capacity requirements should also be taken into account for disk model selection.

5. Draw the HACWS Diagram.

   Draw the diagram of the HACWS configuration according to the results of the above steps. A simple example of the HACWS configuration is illustrated in Figure 7 on page 18. Your diagram should be more detailed and specific to your environment.

6. Plan HACMP Event Processing.

   If you still find any single point of failure on your RISC/6000 SP system, consider the HACMP event script customization to avoid it. For example, if

you have only a single adapter on an external network on each control workstation, it is recommended to promote the adapter failure to a node failure. Some customization examples covered in 2.3, “Customizing HACWS” on page 55 would be helpful.

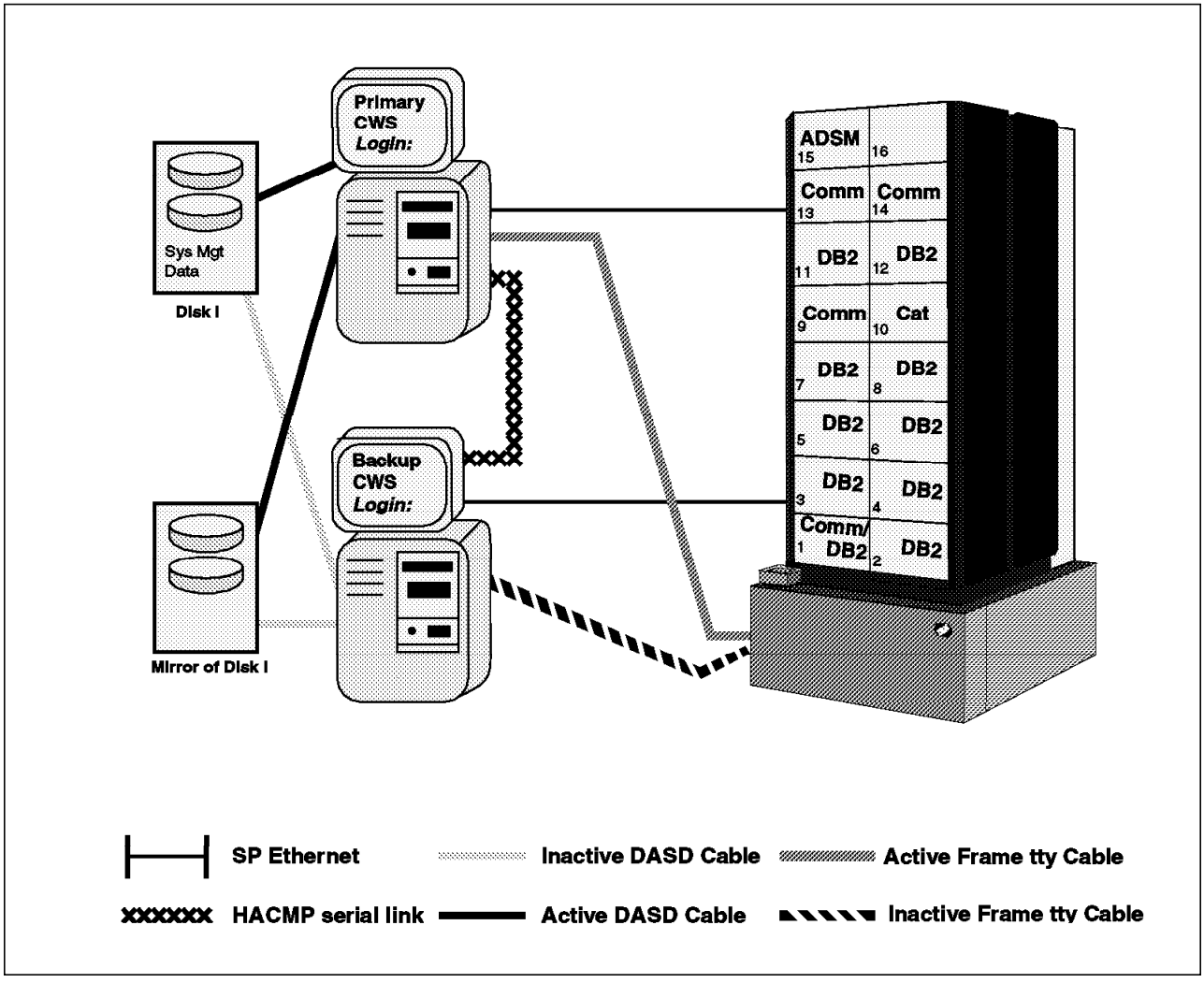


Figure 7. Simple Example of the HACWS Configuration

For the planning of the HACMP cluster configuration, see *HACMP for AIX Planning Guide*, SC23-2768.

Since HACWS uses the HACMP cluster for its configuration and failover scenario, note that HACWS can handle only the *single failure*. Table 3 on page 19 shows what kind of failures HACWS can handle.

Table 3. HACWS Failure/Recovery Matrix

Failure Point	Recovery Method	Remarks
1) System		
1-1 OS	Node Takeover	• By HACWS
1-2 CPU Planer	Node Takeover	• By HACWS
2) Disk/Disk Adapter		
2-1 Internal Disk	Node Takeover	• By HACWS
2-2 Internal Disk Adapter	Node Takeover	• By HACWS
2-3 External Disk	LVM Mirroring RAID	• Redundant disk drives required • RAID disk required
2-4 External Disk Adapter	Dual Buses Node Takeover	• Mirroring over busses required • By error notification method
3) Network		
3-1 SP Ethernet	None	• The impact for SP System can be minimized by dividing the SP Ethernet into multi-segments
3-2 SP Ethernet Adapter	Node Takeover Adapter Swap	• By HACWS (See Note: 1) • Two SP Ethernet Adapters required
3-3 RS232C Cable to Frame	Node Takeover	• By error notification (See Note: 2)
3-4 RS232C Y-Cable	None	• Hot pluggable
3-5 Frame Supervisor Card	None	• Hot pluggable
3-6 RS232C Cable for HACMP	None	• No impact for SP system
3-7 External Network	Network Switch	• Dual network required
3-8 External Network Adapter	Node Takeover Adapter Swap	• By customizing HACMP event script • Standby network adapter required
3-9 TCP/IP Subsystem	Node Takeover	• By HACWS (See Note: 1)

Note:

1. Handled by "/usr/sbin/hacws/events/network_down.post_event" script supplied by HACWS. Also, *netmon.cf* file is required for single network interface configuration (see 2.2.4.6, "Step 24: Verify the Cluster and Node Environments" on page 48).
2. See 2.3.2.1, "To Eliminate Single Point of Failure: CWS-Frame TTY Link" on page 57.

2.1.3 Other Considerations

The main operations on the hardware, such as hardware monitoring, installing a node, and rebooting a node, are available on the primary and backup control workstation. Also, you can execute configuration changes for RISC/6000 SP on the backup control workstation when the backup control workstation is the active control workstation.

However, if changes are made when the backup control workstation is active, some configuration files will need to be updated on the primary control workstation, unless the configuration files are not located on the external shared

disks. Also, several tasks related to the security functions, either AIX or Kerberos, are not available when the backup control workstation is active.

Table 4 shows which tasks can be available or unavailable on the active primary control workstation, active backup control workstation, and inactive backup control workstation.

Table 4. HACWS Task Availability Summary

Task	Active Primary CWS	Active Backup CWS	Inactive Backup CWS
Update Password (Using File-collection)	Yes	No	No
Add or change users (Using File-collection)	Yes	No	No
Change Kerberos keys	Yes	No	No
Install a node	Yes	Yes	No
Change or add partitions	Yes	Yes	No
Add a node to the system	Yes	Yes	No
Hardware monitoring	Yes	Yes	Yes
Reboot nodes	Yes	Yes	Yes
Run diagnostics	Yes	Yes	Yes
Shutdown and restart	Yes	Yes	Yes
Run parallel jobs	Yes	Yes	No
Update file collections	Yes	Yes	No
Accounting	Yes	Yes	No
Change site environment information	Yes	Yes	No

2.2 Installing HACWS

This section describes a detailed HACWS installation procedure, according to the same step sequences as in the “Installing and Configuring the High Availability Control Workstation” document. You can find the document in the “/usr/lpp/ssp.hacws/doc” directory after installing HACWS (ssp.hacws) software in the control workstation.

Also, you need to refer to the *HACMP for AIX Installation Guide*, SC23-2769 for the details of HACMP installation.

Attention

If you already have an installed and operational RISC/6000 SP system, *do not* use the mksysb image of the operational control workstation to install the backup control workstation.

Figure 8 shows the overall flow of installing HACWS.

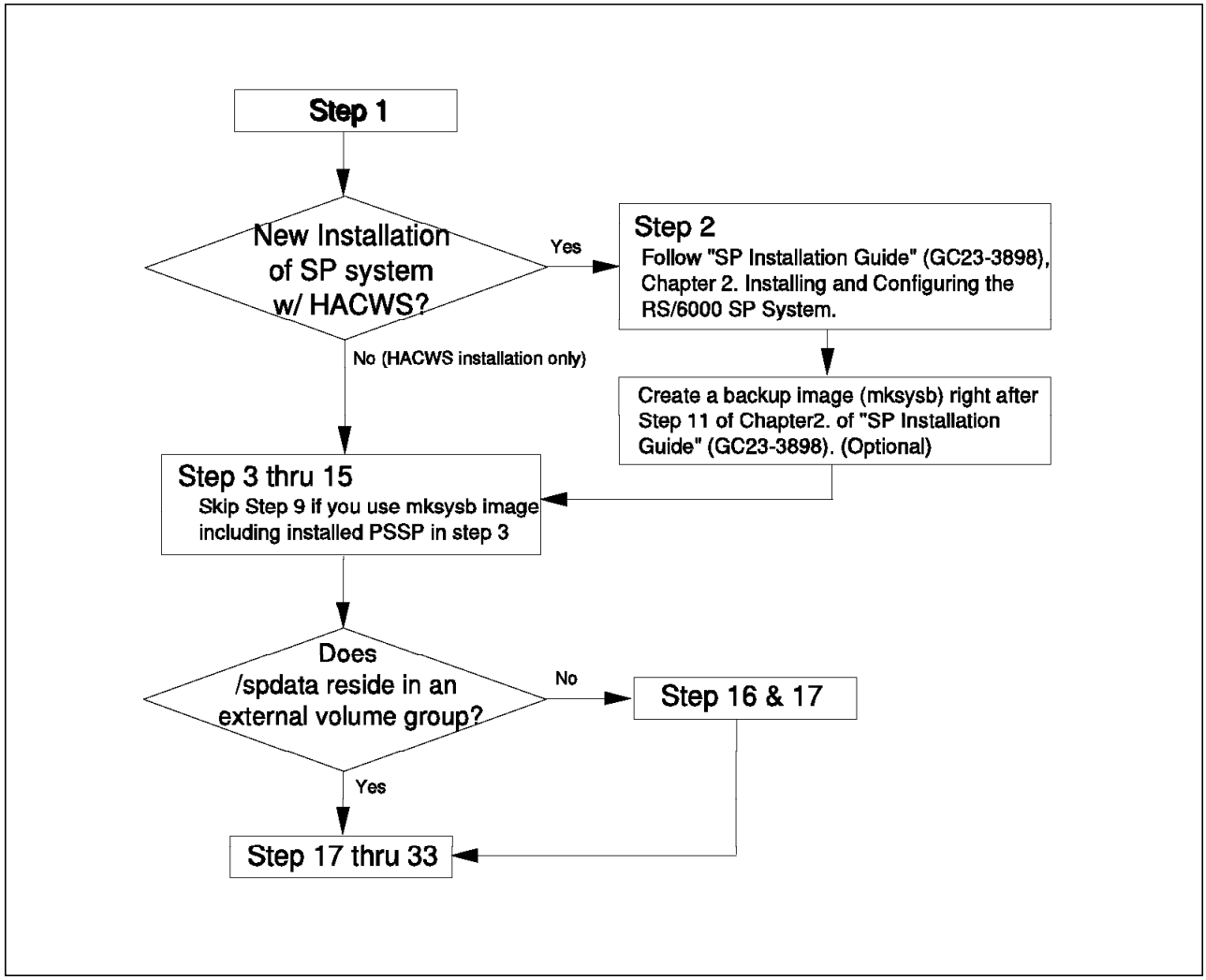


Figure 8. Flow of HACWS Installation

2.2.1 Prepare the Control Workstation

The first phase of installation is to install any necessary software on the control workstations and to plan the network configuration.

2.2.1.1 Step 1: Understand the Procedure

Before proceeding with these steps, read through this chapter and fully understand the installation process. Also, you may need to read and become familiar with the following publications:

- *PSSP System Planning*, GC23-3902; Chapter 2
- *PSSP Administration Guide*, GC23-3897; Chapter 5
- *HACMP for AIX Planning Guide*, SC23-2768
- *RISC/6000 SP Systems: PSSP Version 2 Technical Presentation*, SG24-2542

2.2.1.2 Step 2: Install the RISC/6000 SP System

If you have not done this already, install your RISC/6000 SP system (single control workstation, and frames and nodes) as described in the *SP Installation Guide*, GC23-3898. The RISC/6000 SP system must be completely installed and operating correctly prior to proceeding to the next step.

After you complete the rest of the instructions in this chapter, the original control workstation will become the primary control workstation, and the backup control workstation will be configured to the system.

You can create a mksysb image after step 11 of Chapter 2 as described in *SP Installation Guide*, GC23-3898.

Note: This mksysb image includes AIX PSSP filesets that have not been installed and configured.

2.2.1.3 Step 3: Install AIX V4.1 on the Backup Control Workstation

If you have not done this already, install base AIX on the backup control workstation. For more information, refer to the *AIX Installation Guide*, SC23-2550.

Attention

To install the backup control workstation. *do not* use the mksysb image of the primary control workstation that has PSSP installed, configured, and operational.

Note: You can use the mksysb image that was created in 2.2.1.2, “Step 2: Install the RISC/6000 SP System” to install AIX in this step.

2.2.1.4 Step 4: Backup the Control Workstations

If you used the mksysb image in 2.2.1.2, “Step 2: Install the RISC/6000 SP System,” you can skip this step.

Otherwise, create mksysb images of both the primary and the backup control workstation, and keep them on hand in case of failure during HACWS installation. One way of recovering from a failed HACWS installation is restoring the mksysb image of the system.

Also, if the */spdata* file system is not located on the root volume group, do not forget to backup that system using the following command:

```
# cd /spdata
# tar -cvf /dev/rmt0 ./
```

2.2.1.5 Step 5: Set Up the Hardware

Perform all hardware setup in this step. This step may be skipped if your system is delivered with the new supervisor card feature (F/C #1245).

For the RISC/6000 SP frame, the old Frame Supervisor Card should be replaced with the new Frame Supervisor Card as follows:

1. Disconnect the RS232C cable from the Frame Supervisor Card.
2. Remove the old Frame Supervisor Card from the SP frame.
3. Install the new Frame Supervisor Card into the slot where the old Frame Supervisor Card resided.
4. Connect the RS232C Y-cable to the new Frame Supervisor Card.
5. Connect the RS232C cable from the primary control workstation to the Connector tagged “S1” of the RS232C Y-cable.
6. Connect the RS232C cable from the backup control workstation to the Connector tagged “S2” of the RS232C Y-cable.

   Attention

   The frame supervisor connections from each frame must be connected to the exact same tty ports on both control workstations. If not cabled correctly, frame supervisor connections will not be activated by the hardware monitor.

7. Issue the following commands to set the frame ID in the flash memory of the new Frame Supervisor Card:

   ```
   hmcmds -G setid <frame number>:0
   ```

   For example, if your RISC/6000 SP system has only one frame, issue the following command:

   ```
   hmcmds -G setid 1:0
   ```

8. Check the LEDs on the new Frame Supervisor Card. If it is correctly installed, LED “1,” “2,” and “4” are on in green color, and LED “8” is momentarily on in orange color (See the Figure 6 on page 14 and Table 2 on page 14 for the LED alignment and the definitions on the new Frame Supervisor Card).

The above procedure can be run with the SP system powered-on and running.

To set up the external disks, follow the shared disk installation instructions in Chapter 5 of the *HACMP for AIX Installation Guide*, SC23-2769 and in the hardware installation manuals of each disk model.

If you use the 7133 SSA Disk for the external shared disk between control workstations, the following drivers are required on both control workstations:

Fileset	*Level*
devices.mca.8f97.com	4.1.4.4
devices.mca.8f97.diag	4.1.4.2
devices.mca.8f97.rte	4.1.4.1
ssamc.mc	3.2.0.16

2.2.1.6 Step 6: Plan Network Configuration

The network configuration required by HACWS is somewhat tricky. Plan carefully for the network configuration and IP addresses/hostname assignment. We will use two examples to illustrate the implementation of the common type of environments, which are:

- Non-partitioned system (default partition)
- Partitioned system

The first example will address the default partition environment as depicted in Figure 9 on page 26, Figure 10 on page 27, and Figure 11 on page 28. The second example will address the partitioned environment as depicted in Figure 12 on page 30, Figure 13 on page 31, and Figure 14 on page 32.

Note: Both of the configuration examples presented here presume that the backup control workstation is configured as a secondary authentication server. Also, it should be noted that in our test environment, the IP address of the hostnames for the primary and backup control workstation are on the token ring network. This may be different from your environment where the hostname IP addresses of both control workstation may be on the RISC/6000 SP ethernet network.

At the end of your network planning, your network configuration must conform to the following:

1. On SP Ethernet, each adapter on either control workstation should have a *boot* IP address.
2. The backup control workstation must always be reachable through a network interface whose IP label matches its hostname.
3. The hostname of the primary control workstation must always be identified to HACMP as a service address.
4. All the IP addresses have been added to the */etc/hosts* file and to your name server if your site uses a name server.
5. For the SP system that has *partitions*, only the IP *alias* address corresponding to each partition (except the default partition which corresponds to the hostname and IP address of the control workstation) should be defined on the network interface on the same subnet as the interface that the control workstation *hostname* maps to. No additional *boot* addresses are required for *partitions*. To set the IP *alias* addresses for partitions, follow the procedure in 2.2.5.4, “Step 28: Add IP Address Aliases” on page 51.

Example 1 - Default Partitioned Environment: In this example, the primary and the backup control workstations are connected to the SP Ethernet and the external token-ring network. The objective is to have each SP node recognize its control workstation through the service IP address on the SP Ethernet (9.12.20.37) and the service IP address on the external token ring network (9.12.0.37). These service IP addresses will be enabled by HACMP as shown in Figure 10 on page 27. The hostname of the primary control workstation recognized by each SP node is “sp2cw0,” which is the service IP label of the token-ring network.

The primary control workstation is configured as a primary authentication server, and the backup control workstation is configured as a secondary authentication server of the primary control workstation. The backup control workstation must

be identified and accessible by its hostname, which is *sp2cw1*, in any HACMP phase. Accordingly, the hostname “sp2cw1” is always set, and the IP address “9.12.0.70” corresponding to the IP label, which is the same as the hostname, is assigned on the token-ring interface by *IP alias*.

The SP system in this example is configured as a single partition system.

Figure 9 on page 26 shows the initial network configuration before starting HACMP cluster on either control workstation. Each interface is assigned to the *boot* address. The IP alias corresponding to the hostname *sp2cw1* is defined on the token-ring interface on the backup control workstation, and this IP alias should be out of HACMP cluster definitions.

Figure 10 on page 27 shows the network configuration after starting the HACMP cluster on both the control workstations; that is, it is in the normal operation mode. The only change from Figure 9 on page 26 is that the boot addresses on both the SP Ethernet and the token-ring have been replaced by the service addresses on the primary control workstation.

In the normal operation mode, the IP packet route taken from one of the SP nodes to the active control workstation is shown below:

```
# traceroute sp2cw0
traceroute to sp2cw0.itsc.pok.ibm.com (9.12.0.37), 30 hops max, 40 byte packets
 1  sp2cw0 (9.12.0.37)  2 ms  1 ms  1 ms
```

Figure 11 on page 28 shows that the network configuration after the control workstation services have been moved from the primary control workstation to the backup control workstation. Note that the IP alias *sp2cw1* on the token-ring interface of the backup control workstation remains configured.

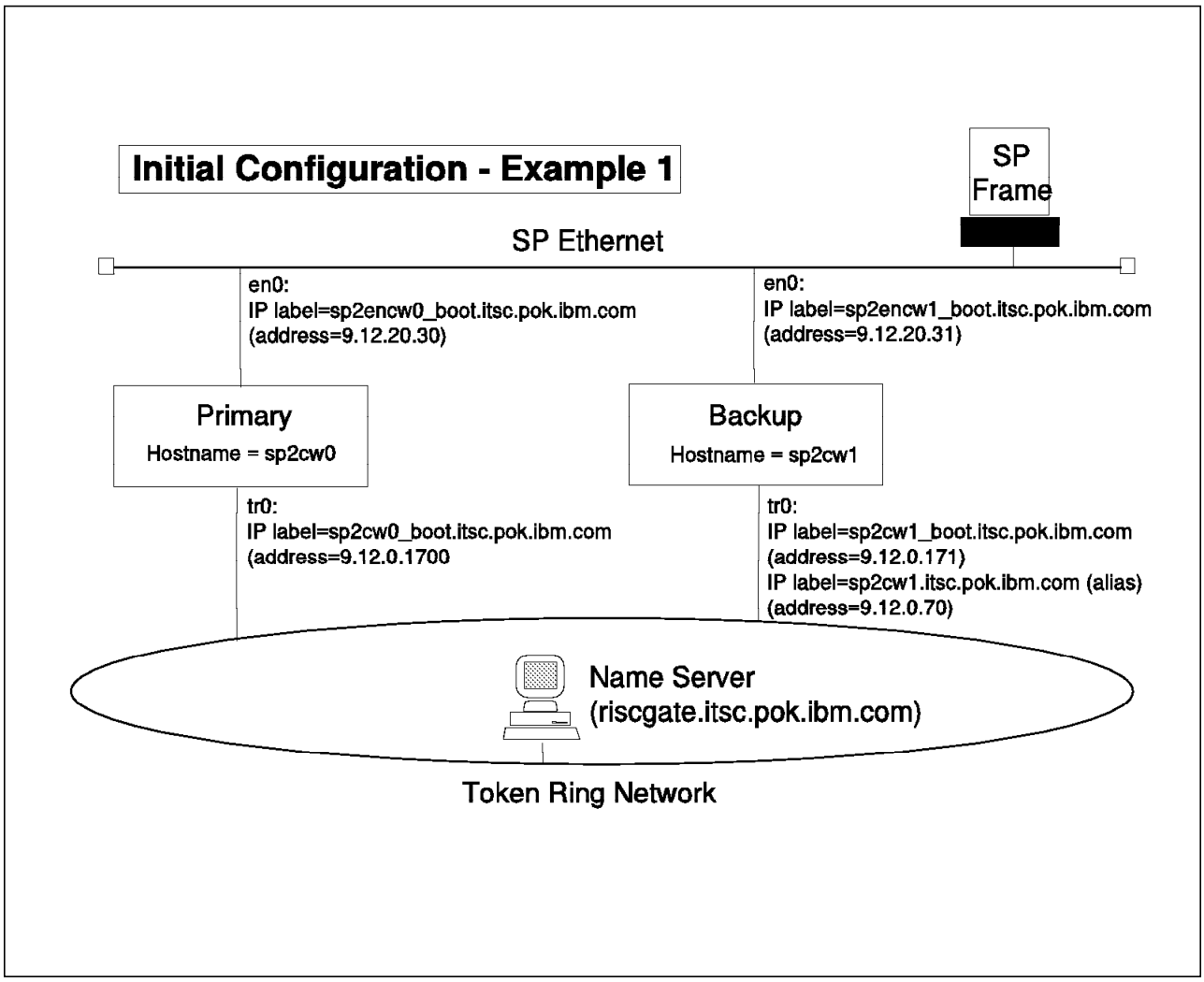


Figure 9. Initial Configuration - Example 1. Before Starting the HACMP Cluster

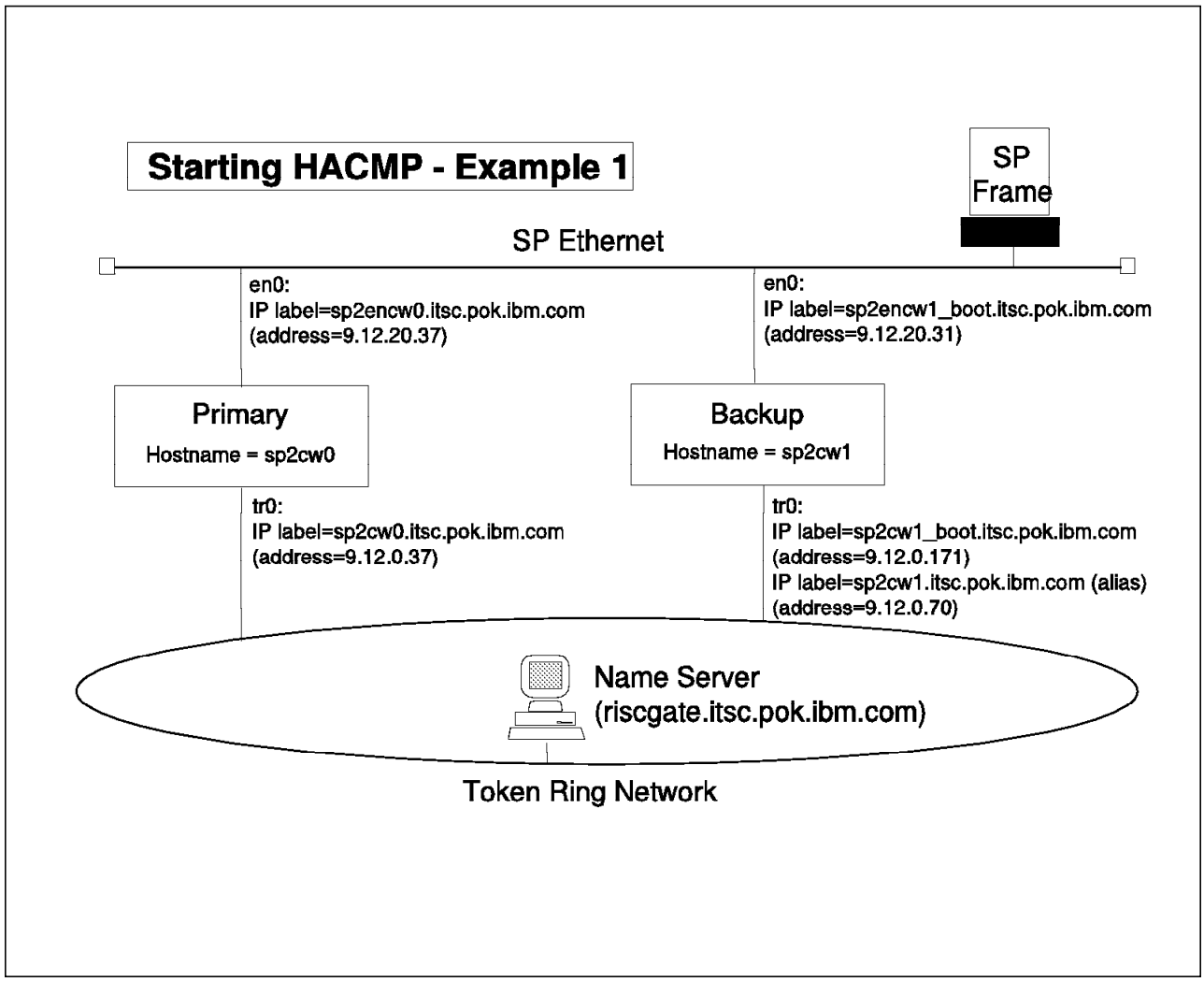


Figure 10. Starting HACMP - Example 1. After Starting the HACMP Clusters

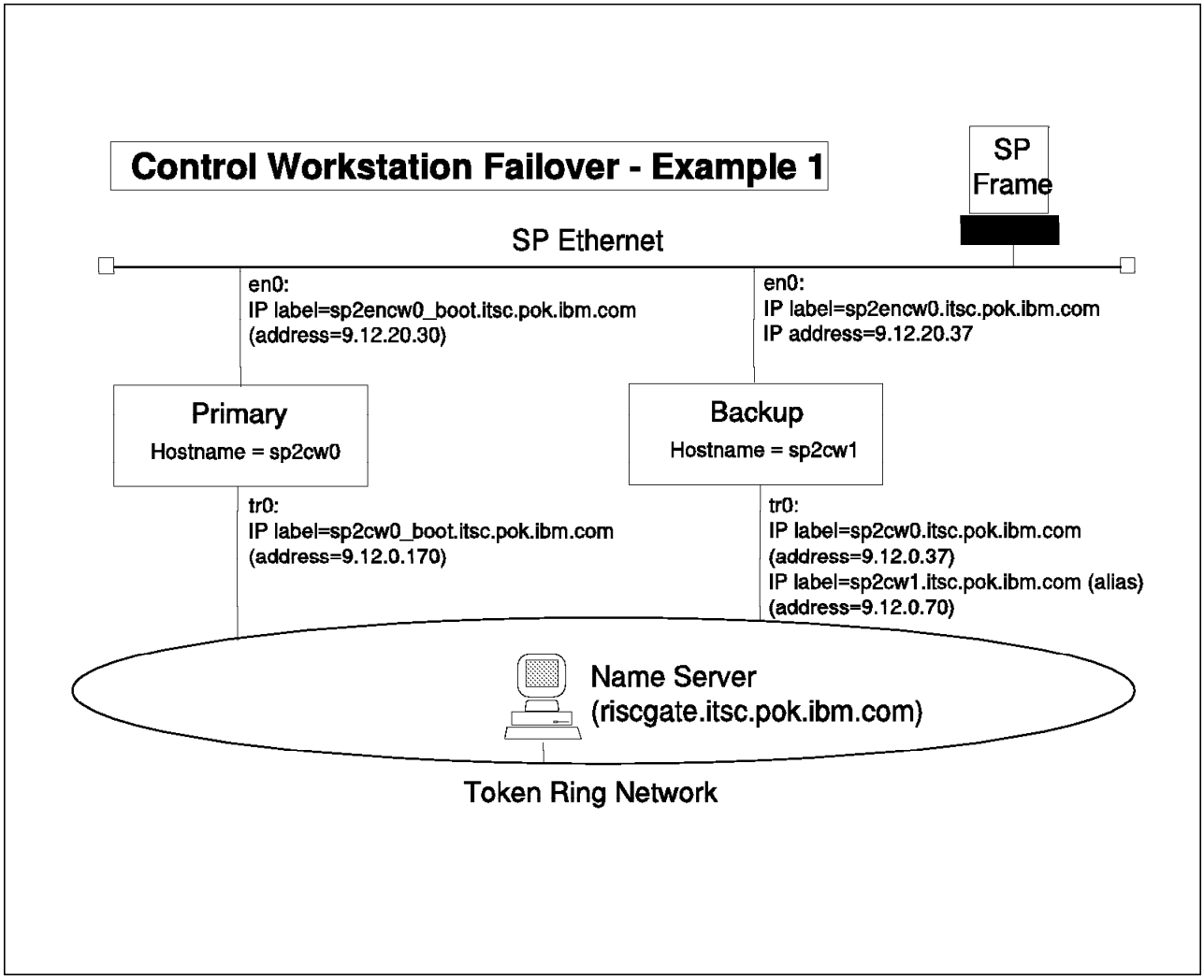


Figure 11. CWS Failover - Example 1. After Primary Control Workstation Failover

Example 2 - Partitioned Environment: In this example, the primary and the backup control workstations are also connected to the SP Ethernet and the external token-ring network. Each SP node in one partition (default partition) recognizes its control workstation by both the IP addresses of "9.12.30.37," which is the service IP address on the SP Ethernet, and "9.12.0.137," which is the service IP address on the external token-ring network. SP nodes in the other partition (secondary partition) recognize their control workstation by both the IP addresses of "9.12.30.37," which is the service IP address on the SP Ethernet, and "9.12.0.138," which is the IP alias address on the external token-ring network. The hostname of the primary control workstation recognized by SP nodes in both the partitions is "sp21cw0," which is the IP label of the token-ring service IP address.

The primary control workstation is configured as a primary authentication server, and the backup control workstation is configured as a secondary authentication server of the primary control workstation. The backup control workstation must be identified and accessible by its hostname, which is *sp21bk0*, in any HACMP phase. Accordingly, the hostname "sp21bk0" is always set, and the IP address

“9.12.0.70” corresponding to the IP label, which is same as the hostname, is assigned on the token-ring interface by IP alias.

The major difference from “Example 1 - Default Partitioned Environment” on page 24 is that the SP system in this example has two partitions; one partition is accessible through IP address “9.12.0.137” and another partition is accessible through IP address “9.12.0.138,” which is implemented by IP alias. on the token-ring interface.

Figure 12 on page 30 shows the initial network configuration before starting the HACMP cluster on either control workstation. Each interface is assigned to the *boot* address. The IP alias corresponding to the hostname *sp21bk0* is defined on the token-ring interface on the backup control workstation, and this IP alias should be out of HACMP cluster definitions.

Figure 13 on page 31 shows the network configuration after starting the HACMP cluster on both the control workstations; that is, it is in the normal operation mode. The boot addresses on both the SP Ethernet and the token-ring have been replaced by the service addresses, and the IP alias for the second SP partition “9.12.0.138” is defined on the token-ring interface on the primary control workstation.

The IP packet route taken from one of the SP nodes in the default partition to the active control workstation in Figure 13 on page 31 is as follows:

```
# traceroute sp21cw0
traceroute to sp21cw0.itsc.pok.ibm.com (9.12.0.37), 30 hops max, 40 byte packets
 1  sp21cw0 (9.12.0.137)  2 ms  1 ms  1 ms
```

Figure 14 on page 32 shows the network configuration after the control workstation failover. Note that the IP alias *sp21bk0* on the token-ring interface of the backup control workstation remains configured. The IP alias for the second SP partition is also defined on the token-ring interface of the backup control workstation, which is now the active control workstation.

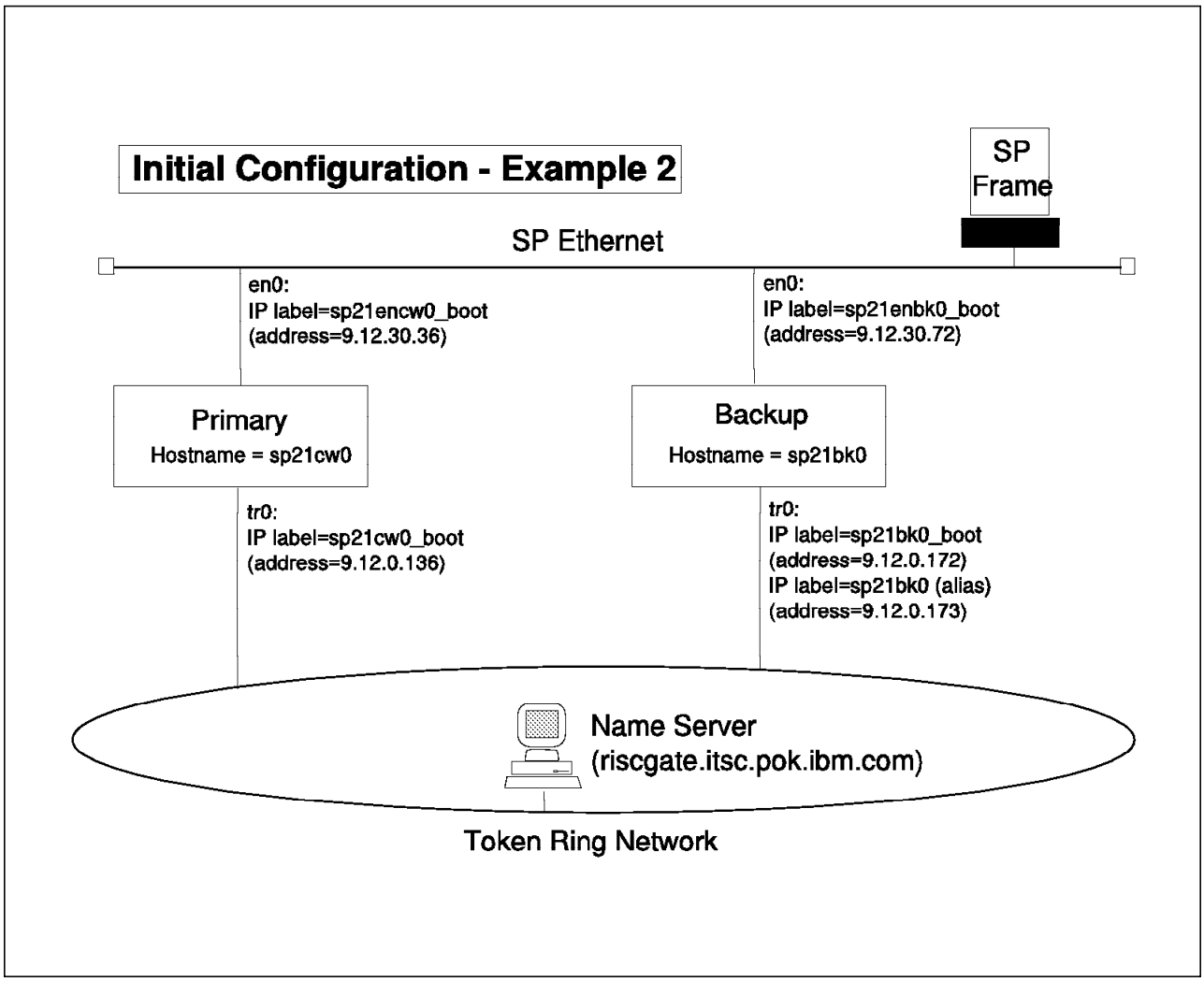


Figure 12. Initial Configuration - Example 2. Before Starting the HACMP Cluster

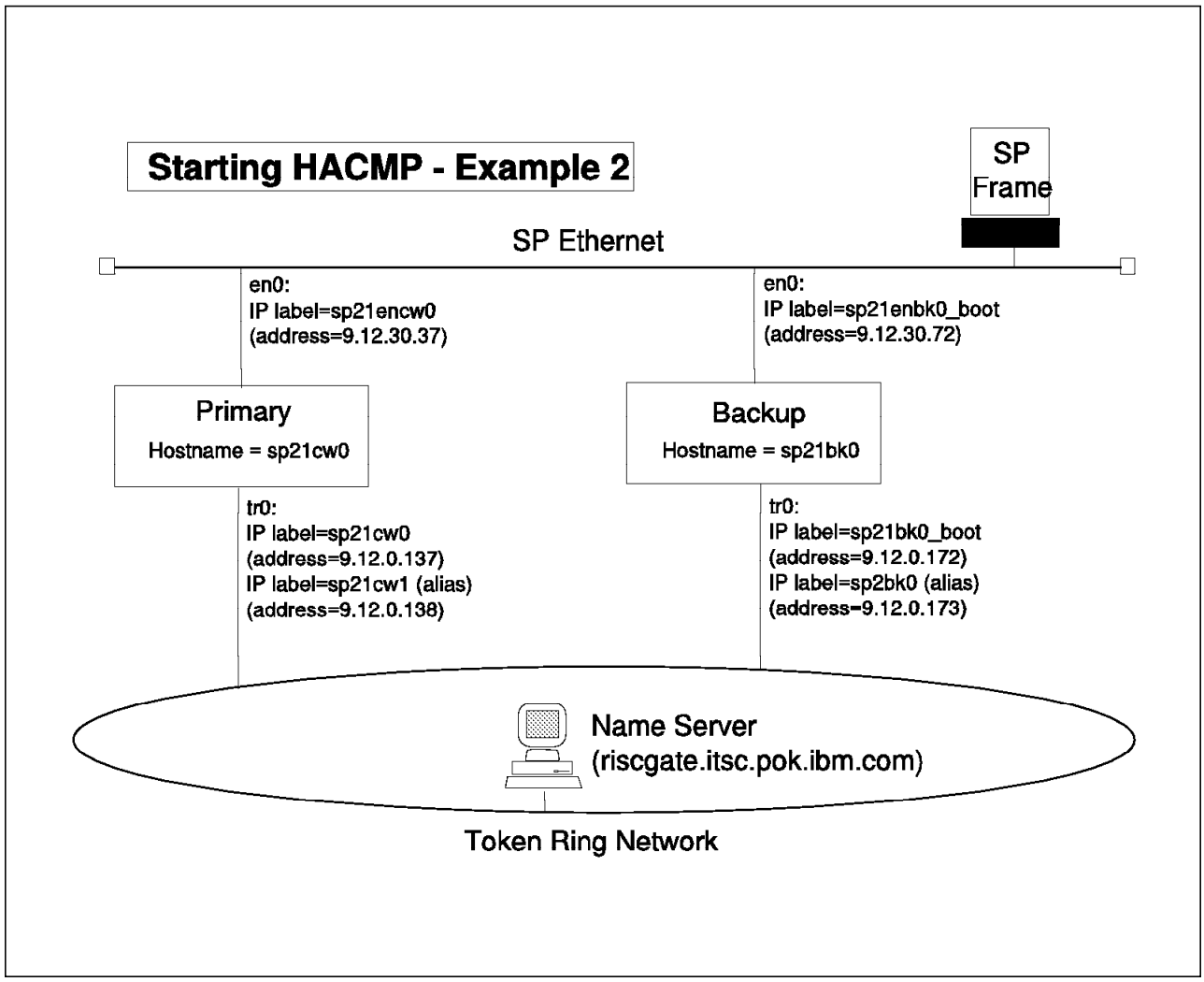


Figure 13. Starting HACMP - Example 2. After Starting the HACMP Cluster

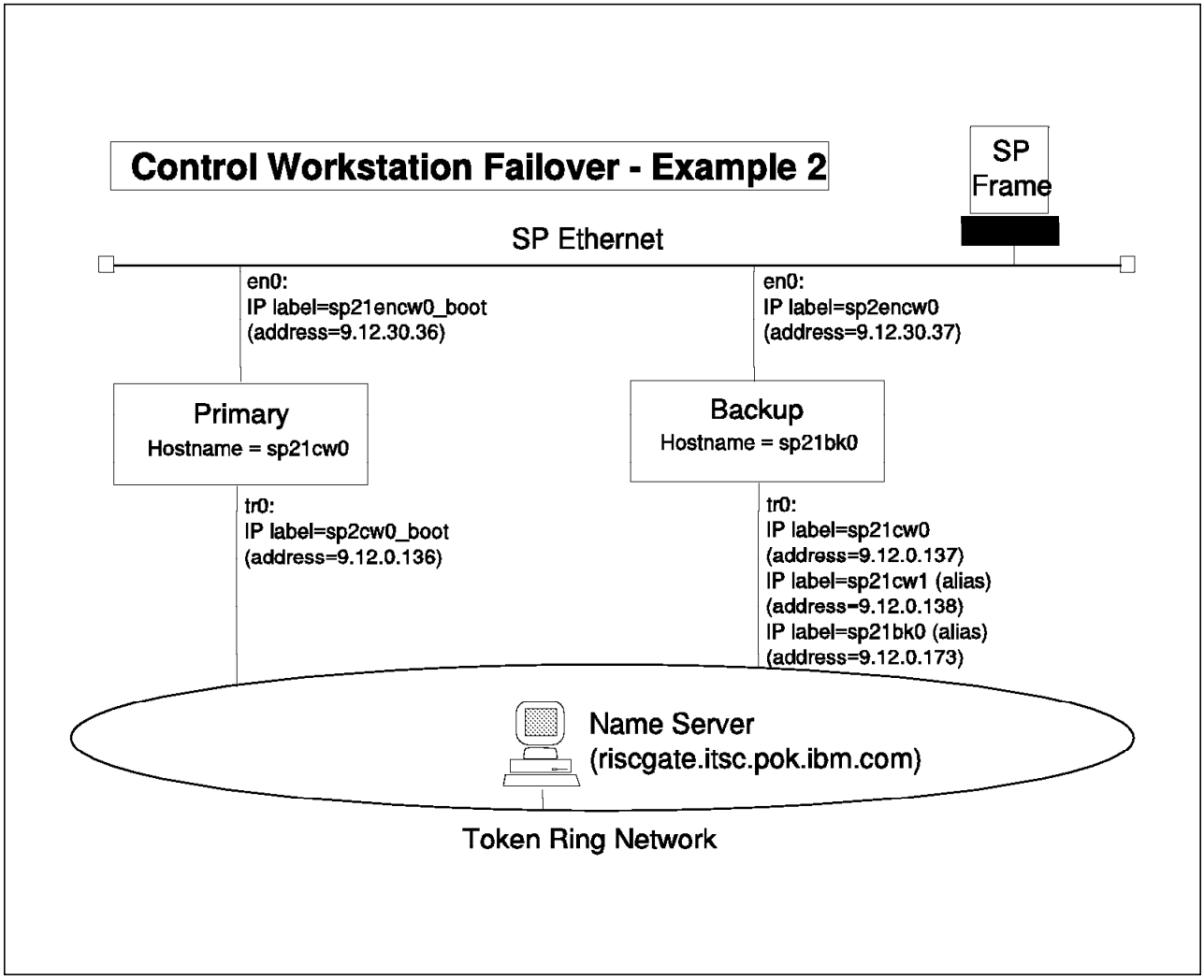


Figure 14. CWS Failover - Example 2. After Primary Control Workstation Failover

2.2.2 Update RISC/6000 SP Authentication Services on Primary CWS

In this phase of installation, you should update kerberos on the control workstations.

2.2.2.1 Step 7: Add the Kerberos Principal

When the *sp_cws* network interface is on the backup control workstation, the network adapter on the primary control workstation is known by an alternate name, such as *sp_cws_bt*. This alternate name is the boot address. The primary boot addresses need to be identified to kerberos so that the backup control workstation can access authenticated services on the primary while the backup control workstation is acting as the active control workstation. .

Also, you need the same principals (rcmd and hardmon) on the backup boot addresses to get the authentication service from the backup control workstation when it is on the boot addresses.

For the “Example 1 - Default Partitioned Environment” on page 24, the following procedure should be followed to add the kerberos principals *rcmd* and *hardmon*:

1. Add the *rcmd* principal on en0 (sp2encw0_boot) by entering the following command:

   /usr/kerberos/etc/kdb_edit

```
# /usr/kerberos/etc/kdb_edit
Opening database...

Enter Kerberos master key: <Enter Master Key>

Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: rcmd
Instance: sp2encw0_boot

<Not found>, Create [y] ? <Hit Enter>

Principal: rcmd, Instance: sp2encw0_boot, kdc_key_ver: 1
New Password: <Enter Password>
Verifying, please re-enter
New Password: <Enter Password>

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 1999-12-31 ] ? <Hit Enter>
Max ticket lifetime [ 255 ] ? <Hit Enter>
Attributes [ 0 ] ? <Hit Enter>
Edit O.K.
```

2. Add the *rcmd* principal on tr0 (sp2cw0_boot).

```
(Continued from the previous screen.)

Principal name: rcmd
Instance: sp2cw0_boot

<Not found>, Create [y] ? <Hit Enter>

Principal: rcmd, Instance: sp2cw0_boot, kdc_key_ver: 1
New Password: <Enter Password>
Verifying, please re-enter
New Password: <Enter Password>

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 1999-12-31 ] ? <Hit Enter>
Max ticket lifetime [ 255 ] ? <Hit Enter>
Attributes [ 0 ] ? <Hit Enter>
Edit O.K.
```

3. Add the *hardmon* principal on en0 (sp2encw0_boot).

```
(Continued from the previous screen.)

Principal name: hardmon
Instance: sp2encw0_boot

<Not found>, Create [y] ? <Hit Enter>

Principal: hardmon, Instance: sp2encw0_boot, kdc_key_ver: 1
New Password: <Enter Password>
Verifying, please re-enter
New Password: <Enter Password>

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 1999-12-31 ] ? <Hit Enter>
Max ticket lifetime [ 255 ] ? <Hit Enter>
Attributes [ 0 ] ? <Hit Enter>
Edit O.K.
```

4. Add the *hardmon* principal on tr0 (sp2cw0_boot).

```
(Continued from the previous screen.)

Principal name: hardmon
Instance: sp2cw0_boot

<Not found>, Create [y] ? <Hit Enter>

Principal: hardmon, Instance: sp2cw0_boot, kdc_key_ver: 1
New Password: <Enter Password>
Verifying, please re-enter
New Password: <Enter Password>

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 1999-12-31 ] ? <Hit Enter>
Max ticket lifetime [ 255 ] ? <Hit Enter>
Attributes [ 0 ] ? <Hit Enter>
Edit O.K.
Principal name: <Hit Enter>
#
```

5. Repeat the above procedure (steps 1 through 4) for the backup control workstation's boot addresses from the primary control workstation.

6. You can verify that the principals were correctly added by issuing the following command:

   ```
   /usr/lpp/ssp/kerberos/etc/kdb_util dump <filename>
   ```

   The file named <filename> should include the lines that follow:

```
hardmon sp2encw0_boot 255 1 1 0 1db1d44f 5026aa2b 200001010459 199603151310 * *
hardmon sp2cw0_boot 255 1 1 0 1db1d44f 5026aa2b 200001010459 199603151310 * *
rcmd sp2cw0_boot 255 1 1 0 8a3aae52 72008722 200001010459 199603151310 * *
rcmd sp2encw0_boot 255 1 1 0 8a3aae52 72008722 200001010459 199603151310 * *
hardmon sp2encw1_boot 255 1 1 0 3d44e21a 5026aa2b 200001010459 199603151311 * *
hardmon sp2cw1_boot 255 1 1 0 3d44e21a 5026aa2b 200001010459 199603151311 * *
rcmd sp2cw1_boot 255 1 1 0 74c34888 72008722 200001010459 199603151311 * *
rcmd sp2encw1_boot 255 1 1 0 74c32888 72008722 200001010459 199603151311 * *
```

After the above procedure, we have the following eight principals:

rcmd.sp2encw0_boot@ITSC.POK.IBM.COM
rcmd.sp2cw0_boot@ITSC.POK.IBM.COM
hardmon.sp2encw0_boot@ITSC.POK.IBM.COM
hardmon.sp2cw0_boot@ITSC.POK.IBM.COM
rcmd.sp2encw1_boot@ITSC.POK.IBM.COM
rcmd.sp2cw1_boot@ITSC.POK.IBM.COM
hardmon.sp2encw1_boot@ITSC.POK.IBM.COM
hardmon.sp2cw1_boot@ITSC.POK.IBM.COM

Note: The *hardmon* principal is used by the system monitor daemon (also called hardmon) on the CWS and by the logging daemon (splogd). The *rcmd* principal is used by the kshd daemon that serves the authenticated version of the remote commands rsh and rcp, as well as the sysctld and kpropd daemons.

2.2.2.2 Step 8: Add Kerberos Service Key

The following example shows the procedure you should use to add the Kerberos *rcmd* and *hardmon* keys for each primary control workstation's boot address.

1. Add the *rcmd* service key for boot addresses on both the en0 and tr0 network interfaces by entering the following command:

   ```
   /usr/lpp/ssp/kerberos/bin/ksrvutil add
   ```

```
# /usr/lpp/ssp/kerberos/bin/ksrvutil add

Name: rcmd

Instance: sp2encw0_boot

Realm: ITSC.POK.IBM.COM

Version number: 1

New principal: rcmd.sp2encw0_boot@ITSC.POK.IBM.COM; version 1

Is this correct:? (y,n)  <Hit Enter>

Password: <Enter Password>
Verifying, please re-enter
Password: <Enter Password>

Key successfully added.

Would you like to add another key? (y,n) y

Name: rcmd

Instance: sp2cw0_boot

Realm: ITSC.POK.IBM.COM

Version number: 1

New principal: rcmd.sp2cw0_boot@ITSC.POK.IBM.COM; version 1

Is this correct:? (y,n)  <ENTER>

Password: <Enter Password>
Verifying, please re-enter
Password: <Enter Password>

Key successfully added.
```

2. Add the *hardmon* service key for boot addresses on both the en0 and tr0 network interfaces.

```
(Continued from the previous screen.)

Would you like to add another key? (y,n) y

Name: hardmon

Instance: sp2encw0_boot

Realm: ITSC.POK.IBM.COM

Version number: 1

New principal: hardmon.sp2encw0_boot@ITSC.POK.IBM.COM; version 1

Is this correct:? (y,n)  <ENTER>

Password: <Enter Password>
Verifying, please re-enter
Password: <Enter Password>

Key successfully added.

Would you like to add another key? (y,n) y

Name: hardmon

Instance: sp2cw0_boot

Realm: ITSC.POK.IBM.COM

Version number: 1

New principal: hardmon.sp2cw0_boot@ITSC.POK.IBM.COM; version 1

Is this correct:? (y,n)  <ENTER>

Password: hardmon (not displayed)
Verifying, please re-enter
Password: hardmon (not displayed)

Key successfully added.

Would you like to add another key? (y,n) n

Old keyfile in /etc/krb-srvtab.old.

#
```

3. Repeat steps 1 and 2 for the backup control workstation's boot addresses.

You can verify that the service keys were correctly added by issuing the following command:

```
/usr/lpp/ssp/kerberos/bin/ksrvutil list | grep _boot
```

The result of this command should be as follows:

```
# /usr/lpp/ssp/kerberos/bin/ksrvutil list | grep _boot
1     rcmd.sp2encw0_boot@ITSC.POK.IBM.COM
1     rcmd.sp2cw0_boot@ITSC.POK.IBM.COM
1     hardmon.sp2encw0_boot@ITSC.POK.IBM.COM
1     hardmon.sp2cw0_boot@ITSC.POK.IBM.COM
1     rcmd.sp2encw1_boot@ITSC.POK.IBM.COM
1     rcmd.sp2cw1_boot@ITSC.POK.IBM.COM
1     hardmon.sp2encw1_boot@ITSC.POK.IBM.COM
1     hardmon.sp2cw1_boot@ITSC.POK.IBM.COM
```

2.2.2.3 Step 9: Install PSSP on Backup Control Workstation

To install PSSP on the backup control workstation, refer to the instructions in Step 11 ″Install PSSP on the Control Workstation″ of Chapter 2 of the *SP Installation Guide*, GC23-3898. Do not run any configuration commands (for example, *install_cw*).

Note: If you have already made a mksysb image with PSSP installed and used it in 2.2.1.3, “Step 3: Install AIX V4.1 on the Backup Control Workstation” on page 22, you can skip this step.

2.2.2.4 Step 10: Configure the Backup CWS As a Secondary Authentication Server or Client

If the primary control workstation is either a primary or secondary kerberos authentication server, then the backup control workstation must be configured as a secondary authentication server. If the primary control workstation is an authentication client, the backup control workstation must also be configured as an authentication client.

Before you start the following procedure, make sure that the hostname of the backup control workstation is set to the hostname that will continue to be used as the backup control workstation identifier (*sp2cw1* in the “Example 1 - Default Partitioned Environment” on page 24, or *sp21bk0* in the “Example 2 - Partitioned Environment” on page 28), and the IP address corresponding to the hostname is assigned to the network interface.

Configuring the Backup CWS As a Secondary Authentication Server

1. Copy the */etc/krb.conf* file from the primary authentication server to the backup control workstation with the following command on the backup control workstation:

   ```
   rcp -p <primary_host>:/etc/krb.conf /etc/krb.conf
   ```

   Note: You need to add the current IP addresses of the backup control workstation in the */.rhosts* file on the primary control workstation before issuing the previous command.

2. Add the following line to the */etc/krb.conf* file:

   ```
   ITSC.POK.IBM.COM sp2cw1.itsc.pok.ibm.com
   ```

   Note: The */etc/krb.conf* file can have two types of entries. One format is:

   ```
   <realm> <hostname> admin server
   ```

   and the other one is:

   ```
   <realm> <hostname>
   ```

   The first format indicates the primary authentication server, and the second format indicates the secondary authentication server. For example, in case of “Example 1 - Default Partitioned Environment” on page 24, the */etc/krb.conf* file will include the following lines:

   ```
   ITSC.POK.IBM.COM sp2cw0.itsc.pok.ibm.com admin server
   ITSC.POK.IBM.COM sp2cw1.itsc.pok.ibm.com
   ```

   where *ITSC.POK.IBM.COM* is the realm and *sp2cw0.itsc.pok.ibm.com* and *sp2cw1.itsc.pok.ibm.com* are the hostnames of the primary authentication server and the secondary authentication server, respectively.

3. Copy the */etc/krb.conf* file back to the primary control workstation as follows:

```
rcp -p /etc/krb.conf <primary_host>:/etc/krb.conf
```

4. Copy the */etc/krb.realms* file from the primary control workstation to the backup control workstation.
5. It is recommended to set the *PATH* environmental variable:

   ```
   export PATH=/usr/lpp/ssp/bin:/usr/lpp/ssp/kerberos/bin:$PATH
   ```

6. Make sure that the CPU times are synchronized between the primary control workstation and backup control workstation. To synchronize the CPU times, do the following:
   a. On both control workstations, issue:

      ```
      startsrc -s timed
      ```

   b. On one of either control workstations, issue:

      ```
      setclock <remote_cws_hostname>
      ```

7. Run the *setup_authent* program on the backup control workstation. In our example, *spcw1* is the backup control workstation.

```
# setup_authent
<screenclear>
********************************************************************
             Logging into Kerberos as an admin user

You must assume the role of a Kerberos administrator <user>.admin
to complete the initialization of kerberos on the local system.
The kinit command is invoked and will prompt you for the password.
If you are setting up your primary server here, you just defined it.
If your primary server is on another system, you must first enter
the user name of an administrative principal defined on that server.

You need to be authenticated as an administrator so that this
program can create the service principals required by the
authenticated services that are included in the ssp package.

       hardmon - for the System Monitor facilities
       rcmd    - for sysctl and Kerberos-authenticated rsh and rcp

For more information, see the kinit man page.
********************************************************************

setup_authent: Enter name of admin user: root

Kerberos Initialization for "root.admin"

Password: <Enter Password which should be the same as for Primary
           Authentication Server>

sp2cw1.itsc.pok.ibm.com: success.
sp2cw1.itsc.pok.ibm.com:               success.
#
```

Note: The last two messages shown in the above screen are issued by the programs that transfer the database from primary to secondary servers to indicate that the backup database has been installed.

8. Make sure that the ticket granting ticket is issued for the backup control workstation.

```
# klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@ITSC.POK.IBM.COM

  Issued           Expires          Principal
Mar 13 17:11:51  Apr 12 18:11:51  krbtgt.ITSC.POK.IBM.COM@ITSC.POK.IBM.COM
Mar 13 17:11:52  Apr 12 18:11:52  rcmd.sp21cw0@ITSC.POK.IBM.COM
#
```

9. From the primary control workstation, add an entry for the secondary authentication server to the */etc/krb.conf* file on all RISC/6000 SP nodes on which you have already initialized authentication. Do this by issuing the following command:

   ```
   hostlist -av | pcp -w - -p /etc/krb.conf /etc/krb.conf
   ```

10. If this is the first secondary authentication server, you should create a root *crontab* entry on the primary authentication server, which is the primary control workstation that invokes that script */usr/kerberos/etc/push-kprop*. Do this by:

    a. Starting crontab editor using the following command:

       ```
       crontab -e
       ```

    b. Adding the following line at the bottom of the crontab file:

       ```
       0 0 * * * /usr/kerberos/etc/push-kprop
       ```

    This periodically propagates database changes from the primary to the secondary authentication server.

Initializing As an Authentication Client System

1. Copy the */etc/krb.conf* file from the primary authentication server to the backup control workstation. Do this by using the following command on the backup control workstation:

   ```
   rcp -p <primary_host>:/etc/krb.conf /etc/krb.conf
   ```

   Note: You must put the current IP address of the backup control workstation into the */.rhosts* file on the primary control workstation before issuing the previous command.

2. Copy the */etc/krb.realms* file from the primary control workstation to the backup control workstation.

3. It is recommended to set the *PATH* environmental variable as follows:

   ```
   export PATH=/usr/lpp/ssp/bin:/usr/lpp/ssp/kerberos/bin:$PATH
   ```

4. Make sure that the CPU times are synchronized between the primary control workstation and backup control workstation. To synchronize the CPU times, do the following:

   a. On both control workstations, issue:

      ```
      startsrc -s timed
      ```

   b. On one of either control workstations, issue:

      ```
      setclock <remote_cws_hostname>
      ```

5. Run the *setup_authent* program on the backup control workstation, which is *sp2cw1* in the following example.

```
# setup_authent
setup_authent: This system is not listed as a Kerberos server in /etc/krb.conf.
Continuing setup as a Kerberos client system only.
<screenclear>
********************************************************************
             Logging into Kerberos as an admin user

You must assume the role of a Kerberos administrator <user>.admin
to complete the initialization of kerberos on the local system.
The kinit command is invoked and will prompt you for the password.
If you are setting up your primary server here, you just defined it.
If your primary server is on another system, you must first enter
the user name of an administrative principal defined on that server.

You need to be authenticated as an administrator so that this
program can create the service principals required by the
authenticated services that are included in the ssp package.

       hardmon - for the System Monitor facilities
       rcmd    - for sysctl and Kerberos-authenticated rsh and rcp

For more information, see the kinit man page.
********************************************************************

setup_authent: Enter name of admin user: root

Kerberos Initialization for "root.admin"

Password: <Enter Password which should be the same as for Primary
           Authentication Server>
#
```

6. Make sure that the ticket granting ticket is issued for the backup control workstation.

```
# klist
Ticket file:    /tmp/tkt0
Principal:      root.admin@ITSC.POK.IBM.COM

  Issued           Expires          Principal
Apr  1 20:03:27  May  1 21:03:27  krbtgt.ITSC.POK.IBM.COM@ITSC.POK.IBM.COM
#
```

2.2.2.5 Step 11: Keep Kerberos Keys in Sync

When control workstation services move back and forth between the two control workstations, the kerberos service keys must remain the same. The *krb-srvtab* file should be the same on both the primary and secondary authentication servers. To accomplish this, enter the following commands on the backup control workstation:

```
rcp -p <primary_name>:/etc/krb-srvtab /etc/krb-srvtab.primary

cp -p /etc/krb-srvtab /etc/krb-srvtab.backup

cat /etc/krb-srvtab.primary >> /etc/krb-srvtab
```

where, <primary_name> is *sp2cw0* for "Example 1 - Default Partitioned Environment" on page 24, or *sp21cw0* for "Example 2 - Partitioned Environment" on page 28.

Note: *Do not* forget to repeat this whenever you change Kerberos service keys (for example, *setup_authent*) on either of the two control workstations.

2.2.2.6 Step 12: Verify Kerberos Data

Make sure the Kerberos principal and *rcmd* key created in 2.2.2.4, “Step 10: Configure the Backup CWS As a Secondary Authentication Server or Client” on page 37 exist for the network address that matches the hostname of the backup control workstation.

```
# /usr/lpp/ssp/kerberos/bin/kadmin
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:  get_entry rcmd.sp2cw1
Admin password:
Info in Database for rcmd.sp2cw1:
   Max Life: 255   Exp Date: Fri Dec 31 23:59:59 1999

   Attribs: 00  key: 0 0
admin:  q
Cleaning up and exiting.
#
```

Also, you can check the version number and principal name in the server key file as follows:

```
# /usr/lpp/ssp/kerberos/bin/ksrvutil list
Version    Principal
   2     rcmd.sp2cw1@ITSC.POK.IBM.COM
   2     hardmon.sp2cw1@ITSC.POK.IBM.COM
   2     hardmon.sp2cw1_boot@ITSC.POK.IBM.COM
   2     rcmd.sp2cw1_boot@ITSC.POK.IBM.COM
   2     rcmd.sp2encw1_boot@ITSC.POK.IBM.COM
   2     hardmon.sp2encw1_boot@ITSC.POK.IBM.COM
   1     hardmon.sp2cw0@ITSC.POK.IBM.COM
   1     rcmd.sp2cw0@ITSC.POK.IBM.COM
   1     hardmon.sp2encw0@ITSC.POK.IBM.COM
   1     rcmd.sp2encw0@ITSC.POK.IBM.COM
   1     rcmd.sp2encw0_boot@itsc.pok.ibm.com
   1     hardmon.sp2encw0_boot@itsc.pok.ibm.com
   1     rcmd.sp2cw0_boot@ITSC.POK.IBM.COM
   1     hardmon.sp2cw0_boot@ITSC.POK.IBM.COM
#
```

2.2.3 Set Up External File System

In this phase, the external file system shared by the control workstations is set up.

2.2.3.1 Step 13: Install the HACWS Image on Both Control Workstations

Make sure that all the prerequisite filesets for HACWS are installed on both control workstations; then install *ssp.hacws* on both control workstations. For the details of prerequisite fileset for HACWS, refer to 2.1.1.2, “Software Requirements” on page 16.

After installing, check to see if the feature is correctly installed and if the necessary files are installed.

```
# lslpp -l ssp.hacws
  Fileset                  Level  State      Description
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.hacws              2.1.0.0  COMMITTED  SP High Availability Control
                                              Workstation

Path: /etc/objrepos
  ssp.hacws              2.1.0.0  COMMITTED  SP High Availability Control
                                              Workstation
# lslpp -f ssp.hacws
  Fileset               File
  ----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  ssp.hacws 2.1.0.0     /usr/sbin/hacws/events/network_down.post_event
                        /usr/sbin/hacws/events/release_service_addr.pre_event
                        /usr/sbin/hacws/events/node_up_complete.post_event
                        /usr/lpp/ssp.hacws/config/rc.hacws.BACKUP
                        /usr/sbin/hacws/spcw_verify_cabling
                        /usr/sbin/hacws/hacws_check_fileset
                        /usr/sbin/hacws/hacws_pre_event
                        /usr/sbin/hacws/events/acquire_service_addr.post_event
                        /usr/sbin/hacws/events/acquire_service_addr.pre_event
                        /usr/sbin/hacws/spcw_addevents
                        /usr/sbin/hacws/install_hacws
                        /usr/sbin/hacws/spcw_filec_update
                        /usr/lpp/ssp.hacws/README
                        /usr/lpp/ssp.hacws/README/ssp.hacws.README
                        /usr/lpp/ssp.hacws/optlevel.ssp.hacws
                        /usr/sbin/hacws
                        /usr/sbin/hacws/spcw_apps
                        /usr/lpp/ssp.hacws/config
                        /usr/sbin/hacws/events
                        /usr/sbin/hacws/hacws_verify
                        /usr/sbin/hacws/events/release_service_addr.post_event
                        /usr/lpp/ssp.hacws/config/rc.backup_cw_alias
                        /usr/lpp/ssp.hacws/config/rc.hacws.PRIMARY
                        /usr/lpp/ssp.hacws/doc/hacws_install_guide.ps
                        /usr/lpp/ssp.hacws/doc/hacws_install_guide.ascii
                        /usr/sbin/hacws/spcw_defer_ntp
                        /usr/sbin/hacws/hacws_post_event
                        /usr/lpp/ssp.hacws/doc
                        /usr/lpp/ssp.hacws/doc/README

Path: /etc/objrepos
  ssp.hacws 2.1.0.0     /spdata/sys1/hacws
                        /var/adm/hacws/events
                        /var/adm/hacws
                        /spdata/sys1/hacws/rc.syspar_aliases
                        /etc/ssp
#
```

2.2.3.2 Step 14: Stop the Primary CWS

Stop control workstation services on the primary control workstation by entering the following command:

```
/usr/sbin/hacws/spcw_apps -d
```

Check if all SP-related daemons have stopped by using `lssrc -a` command. The following subsystems (or groups) should be stopped:

```
sdr
hb
hr
```

supfilesrv
sysctld
hardmon
splogd

2.2.3.3 Step 15: Configure the Network

You need to configure each control workstation to use its boot addresses after it reboots. To do this, enter one of the following commands on both the primary and backup control workstations:

```
smit chinet
```

or

```
smit mktcpip
```

Note: Do not reboot both the primary and backup control workstations until you are instructed to do so in 2.2.5.7, “Step 31: Reboot Control Workstations” on page 52.

2.2.3.4 Step 16: Migrate Internal File System

The */spdata* directory needs to reside in an external volume group so both control workstations can access it. This can be accomplished by migrating the */spdata* files to the external volume group from a separate file system in an internal volume group. Complete the following procedure:

1. Determine which file system contains the */spdata* directory by entering the following command:

   ```
   df /spdata
   ```

   If the filesystem is separate, the mount point will be */spdata*.

2. Unmount the /spdata file system by entering the following command:

   ```
   umount /spdata
   ```

3. Enter:

   ```
   smit chjfs
   ```

4. Select **/spdata**

5. Set the new mount point to */spdata.old*. Set “mount automatically at system restart” to *no*.

6. Mount the file system at its new mount point by entering the following command:

   ```
   mount /spdata.old
   ```

2.2.3.5 Step 17: Set Up External File System

Follow the procedure below. The LVM file system configuration instructions in Chapter 6 of the *HACMP for AIX Installation Guide*, SC23-2760-01 will help you to understand the following procedure.

1. Create a volume group on the external disks from the primary control workstation. The volume group can have any name. Be sure to use a major number that is available on both control workstations. You can use the following command to find the available major number on the system:

   ```
   lvlstmajor
   ```

2. Create a file system in the new volume group. Use the following procedure:

   Create Log LV with type *jfslog*

Format the Log LV with the following command

```
logform /dev/<LV_name>
```

Create LV for the file system with unique name

Create new file system named */spdata*

Note: The current release of AIX (4.1.4) has a bug that prevents the LV name from being changed.

3. Add copies to the logical volume for both */spdata* file system and jfslog if necessary.
4. Mount the new /spdata file system.

   ```
   mount /spdata
   ```

5. Copy the */spdata* files from the old */spdata* file system to the new filesystem with the following command:

   ```
   cp -Rph /spdata.old/* /spdata
   ```

6. Unmount both spdata file systems with the following commands:

   ```
   umount /spdata.old
   ```

   ```
   umount /spdata
   ```

7. Vary off the volume group on the primary control workstation.
8. Import the volume group on the backup control workstation with the following command:

   ```
   importvg -y <vg_name> -V <major_number> <physical_volume_name>
   ```

   Note: *<major_number>* specified here must be the same as that specified on the primary control workstation when the volume group was created.

9. Change the volume group so it remains dormant at booting on the backup control workstation, and set *no* to the quorum required questions, if necessary, by issuing the following command:

   ```
   chvg -a n -Q n(or y) <spdata_vg_name>
   ```

   Note: If quorum is set to *yes* then you must have more than 50% of the disk to be operational. When quorum is set to *no*, you must have 100% of the disk to be operational at vary on time.

10. Change the */spdata* filesystem to not automatically mount on booting by using the `smit chfs` command.
11. Vary off the volume group on the backup control workstation.
12. Make the */spdata* file system available on the primary control workstation with the following command:

    ```
    varyonvg spdatavg
    ```

    ```
    mount /spdata
    ```

13. Check if the volume group and the filesystem are set to remain dormant at booting by using the following commands:

    ```
    smit chvg
    ```

    ```
    smit chfs
    ```

14. If you set **no** to the quorum required question on the backup control workstation then make sure the quorum is disabled on the primary control workstation by using `smit chvg` command. Ensure that the setting for

quorum definitions is the same on both the primary and backup control workstation.

Figure 15 and Figure 16 on page 46 show the case of using 9333 Serial Disk and 7133 SSA DISK as external disks. In both examples, /spdata is highly available through LVM mirroring.

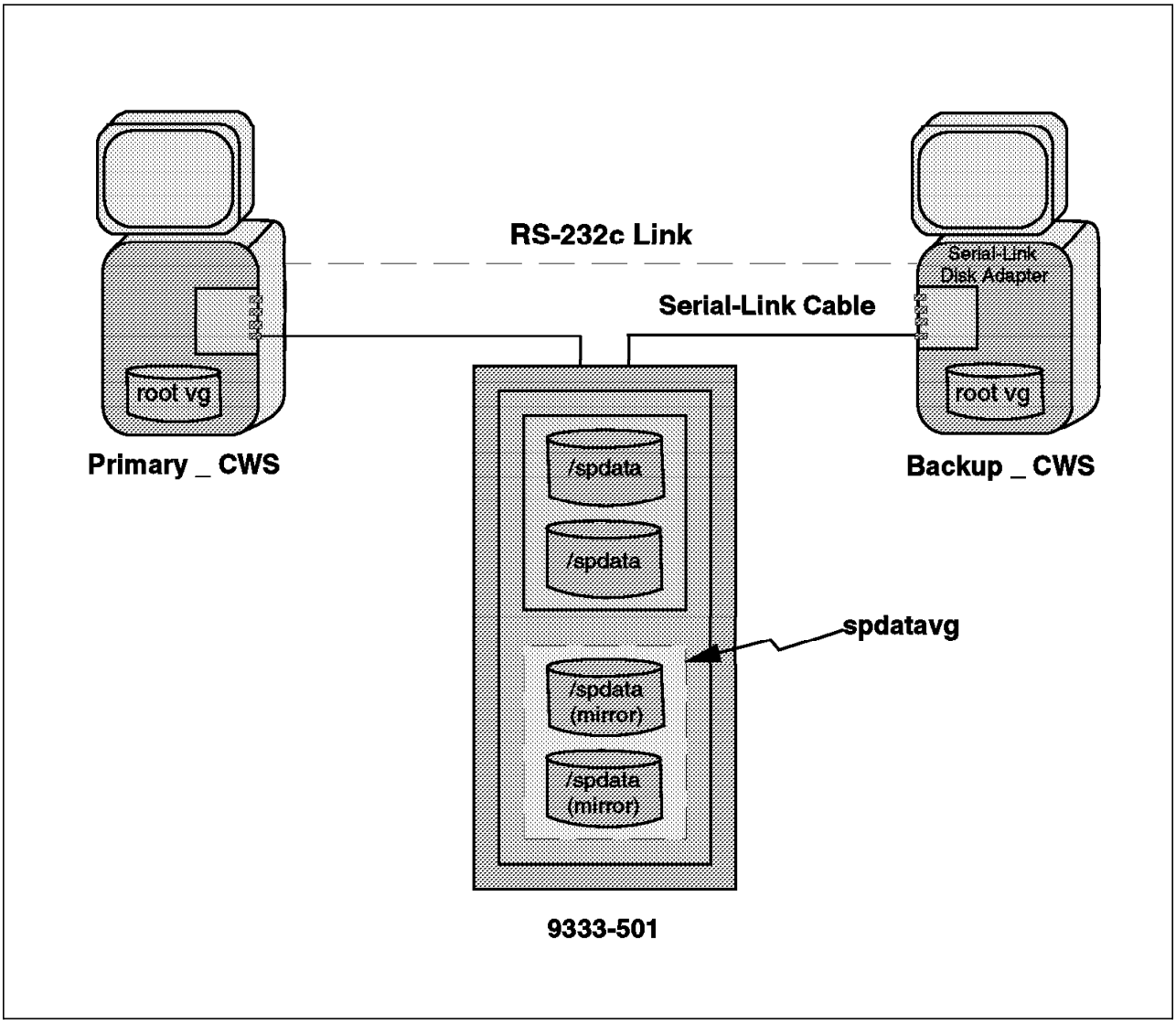


Figure 15. SDR Data Allocation Example for HACWS with 9333

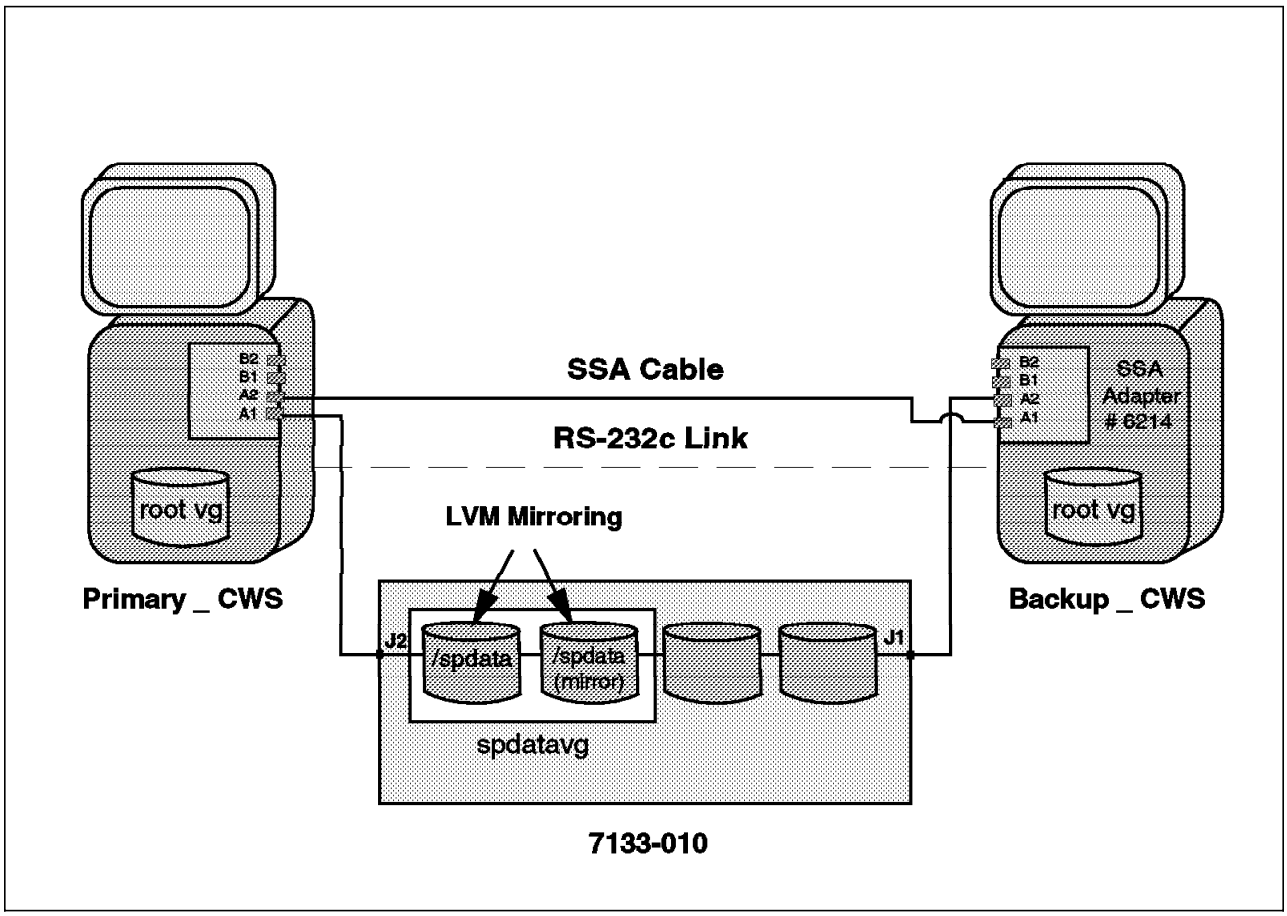


Figure 16. SDR Data Allocation Example for HACWS with 7133

2.2.3.6 Step 18: Complete Administration Tasks

Use the following list to make the control workstations ready for configuring the HACMP cluster and HACWS:

- The *Install_hacws* command you use in 2.2.5.2, “Step 26: Set Up the HACWS Configuration” on page 49 sets I/O pacing to the HACMP recommended starting points. If you want to let *install_hacws* set these values, skip this step.
- A boot address should be defined as the service address for each ethernet adapter on either control workstation as well as other external networks.

  Use the following commands to assign a boot address on each adapter:

  ```
  smit mktcpip
  ```

  or

  ```
  smit chinet
  ```

  If you use the latter command, you should add a line including the IP address and IP label in the */etc/hosts/* file as follows:

  ```
  9.12.0.171   sp2cw0_boot
  ```

- The */etc/hosts* file should contain both short and long hostnames for each network interface, so that both types of references (for example, *sp2cw0_boot* and *sp2cw0_boot.itsc.pok.ibm.com*) can be resolved.

- Set up the */.rhosts* file on both control workstations so that HACMP can use normal *rsh* and *rcp* commands. If you do not want to set up */.rhosts* files, you can make HACMP use the authenticated versions of the *rsh* and *rcp* commands by changing your *PATH* with the following command:

  ```
  export PATH=/usr/lpp/ssp/rcmd/bin:$PATH
  ```

  Note: If your *PATH* is already set up as in this way, the authenticated version of the *rsh* and *rcp* commands will be used even if you set up */.rhosts* files.

- Make sure *tty* devices are available on the backup control workstation by using the following command:

  ```
  lsdev -Cctty
  ```

  Two tty devices should be available, one for the Frame Supervisor Card and another for the HACMP serial link. If you do not see two tty devices available, create them with the `smit tty` command, using all default settings.

2.2.4 Install High Availability Cluster Multi-Processing

This section describes the steps you take to setup HACMP to work with HACWS.

2.2.4.1 Step 19: Install HACMP on the Primary and Backup CWSs

Follow the instructions in Chapter 8 of the *HACMP for AIX Installation Guide*, SC23-2769-01 to install the HACMP software.

Note: To retain the network changes you made in 2.2.3.3, “Step 15: Configure the Network” on page 43, do not reboot the control workstations at this time.

2.2.4.2 Step 20: Verify the Cluster Software

Follow the instructions in Chapter 10 of the *HACMP for AIX Installation Guide*, SC23-2769-01, and make sure that both control workstations have no problems.

2.2.4.3 Step 21: Define the Cluster Environment

1. Define the Cluster ID and name.
2. Define nodes to HACMP.
3. Define adapters to HACMP.

   You must configure HACMP to use hardware address takeover in conjunction with IP address takeover on any service adapter belonging to the SP Ethernet network or whose name matches the hostname of the primary control workstation. In our examples, the hardware address takeover on SP Ethernet and token-ring network should be defined.
4. Synchronize the Cluster definition on all nodes.

2.2.4.4 Step 22: Configure the HACWS Application Server

Follow the instructions in Chapter 12 of the *HACMP for AIX Installation Guide*, SC23-2769.

For the HACWS application server, use the following definitions:

```
Server Name :  hacws_server
Start Script : /usr/sbin/hacws/spcw_apps -ua
Stop Script : /usr/sbin/hacws/spcw_apps -di
```

2.2.4.5 Step 23: Define the Resource Group

Follow the instructions in Chapter 12 of the *HACMP for AIX Installation Guide*, SC23-2769 to define the resource group. For the HACWS resource group, use the following definitions:

- Resource group name: **hacws_group1**
- Node relationship: **rotating**
- Participating node names: The node names of the primary control workstation and backup control workstation
- Service IP label: The service IP label(s) planned in 2.2.1.6, “Step 6: Plan Network Configuration” on page 24
- Filesystems: **/spdata**
- Volume groups: **spdatavg**

Attention

When you define the resource group, you must enter not only the shared filesystem name (*/spdata*) but also the shared volume group name (*spdatavg*), which is not required by HACMP. If you do not do this, */usr/sbin/hacws/spcw_verify* script, which you will use in 2.2.5.5, “Step 29: Verify the HACWS Configuration” on page 51, will abend with the following messages:

```
# /usr/sbin/hacws/install_hacws
sp2cw0: hacws_verify: Resource group hacws_group1 contains no
volume group.
sp2cw1_boot: hacws_verify: Resource group hacws_group1
contains no volume group.
hacws_verify: Execution on node sp2cw0 failed.
#
```

2.2.4.6 Step 24: Verify the Cluster and Node Environments

Follow the instructions in Chapter 13 of the *HACMP for AIX Installation Guide*, SC23-2769 to verify the cluster and the node environments.

Note: If a control workstation is configured without a standby adapter on one of its service networks, you need to create the */usr/sbin/cluster/netmon.cf* file that specifies additional network addresses to which ICMP ECHO requests can be sent. In “Example 1 - Default Partitioned Environment” on page 24, the *netmon.cf* file should include the following lines:

```
sp2n01
sp2n02
```

where *sp2n01* and *sp2n02* are hostnames of SP nodes. With this file, HACMP can identify *node_network_failure*, that is, the failure of the SP Ethernet adapter or TCP/IP subsystem, from *global_network_failure*.

2.2.5 Set Up and Test HACWS

In this phase, the steps you take to customize HACWS and verify its function are provided.

2.2.5.1 Step 25: Make Each Control Workstation Addressable by Its Hostname

Section 2.2.5.2, “Step 26: Set Up the HACWS Configuration” requires that each control workstation to be addressable by its hostname. For the “Example 1 - Default Partitioned Environment” on page 24, issue the following commands:

```
<On the Primary CWS>

ifconfig en0 sp2encw0.itsc.pok.ibm.com netmask 255.255.255.0 up

ifconfig tr0 sp2cw0.itsc.pok.ibm.com netmask 255.255.255.0 up

<On the Backup CWS>

ifconfig tr0 sp2cw1.itsc.pok.ibm.com netmask 255.255.255.0 up
```

Also, make sure that the hostnames of the primary and backup control workstations are set to the correct hostnames (here, “sp2cw0” and “sp2cw1”). If not, set the hostname by the `smit hostname` command on each control workstation.

Note: If you use name service, please check if the addresses can be resolved correctly by using the `netstat -i` command. Do this after using the preceeding commands. If the command does not respond correctly, stop using name service, then try this step again. In such a case, do not forget to restart name service before 2.2.5.8, “Step 32: Start Cluster Services” on page 53.

2.2.5.2 Step 26: Set Up the HACWS Configuration

Configure the primary and backup control workstation as an HACWS configuration.

Note: You can do this step either from the primary or backup control workstation.

1. Issue `smit hacws`.

```
                High Availability Control Workstation Management

Move cursor to desired item and press Enter.

Install and Configure HACWS
Identify Event Scripts to HACMP
Verify HACWS Installation and Configuration
Verify Frame to Control Workstation Cabling
```

2. Select **Install and Configure HACWS**.

```
                Install and Configure HACWS

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

                                                [Entry Fields]
* HOSTNAME of primary control workstation       [sp2cw0]
* HOSTNAME of backup control workstation        [sp2cw1]
  Execute on both primary and backup?            yes              +
```

3. Press the Enter key.
4. After completion of this command, please check if the hostnames and the IP addresses of the backup control workstation are set correctly in the System Data Repository.

```
# /usr/lpp/ssp/bin/SDRGetObjects SP backup_cw
backup_cw
sp2cw1
# /usr/lpp/ssp/bin/SDRGetObjects SP ipaddrs_bucw
ipaddrs_bucw
9.12.0.171:9.12.20.31
```

Notes:

1. If any optional ssp filesets (for example, ssp.jm or ssp.top) are installed on one control workstation, the same filesets must be installed on another control workstation.
2. If you use HACMP for AIX V4.1.1 (not V4.1) before starting this step, you need APAR Ix57291 or you can modify the *install_hacws* script as follows. Otherwise, this script will soon abend with the messages of the prerequisite checking failure.

```
 ...
done <<EOF
ssp.clients     yes     2.1.0.0
ssp.authent     no      2.1.0.0
ssp.basic       yes     2.1.0.0
ssp.css         no      2.1.0.0
ssp.jm          no      2.1.0.0
ssp.sysman      yes     2.1.0.0
ssp.sysctl      yes     2.1.0.0
ssp.gui         yes     2.1.0.0
ssp.top         no      2.1.0.0
ssp.hacws       yes     2.1.0.0
cluster.client  yes     4.1.0.0
cluster.server  yes     4.1.0.0
EOF
 ...
```

Figure 17. install_hacmp - Before Modification

```
 ...
done <<EOF
ssp.clients     yes     2.1.0.0
ssp.authent     no      2.1.0.0
ssp.basic       yes     2.1.0.0
ssp.css         no      2.1.0.0
ssp.jm          no      2.1.0.0
ssp.sysman      yes     2.1.0.0
ssp.sysctl      yes     2.1.0.0
ssp.gui         yes     2.1.0.0
ssp.top         no      2.1.0.0
ssp.hacws       yes     2.1.0.0
EOF
#cluster.client  yes     4.1.0.0
#cluster.server  yes     4.1.0.0
 ...
```

Figure 18. install_hacws - After Modification

2.2.5.3 Step 27: Customize Cluster Event Processing

1. Issue `smit hacws`.

```
     High Availability Control Workstation Management

Move cursor to desired item and press Enter.

Install and Configure HACWS
Identify Event Scripts to HACMP
Verify HACWS Installation and Configuration
Verify Frame to Control Workstation Cabling
```

2. Select **Identify Event Scripts to HACMP** and press Enter.

The following commands are defined as a pre-event command and post-event command, respectively, for all HACMP events after the completion of this step.

/usr/sbin/hacws/hacws_pre_event
/usr/sbin/hacws/hacws_post_event

Note: You can see this definition by issuing the `odmget HACMPevent` command.

2.2.5.4 Step 28: Add IP Address Aliases

In 2.2.1.6, “Step 6: Plan Network Configuration” on page 24 you planned the network configuration for both control workstations. If your configuration requires IP address aliases, you need to provide the appropriate commands to HACWS.

If you need to use an IP address alias to configure the hostname of the backup control workstation as a network interface, edit the */etc/rc.backup_cw_alias* script on the backup control workstation. Also, you must add the following line near the end of the */etc/rc.net* file:

```
/etc/rc.backup_cw_alias
```

If you need to use an IP address alias to configure a network interface for an SP system partition, or if you need to configure an IP address alias on the active control workstation for some other reason, edit the */spdata/sys1/hacws/rc.syspar_aliases* script. This script runs on the active control workstation before it starts the control workstation function services. See the comments in the script for more details.

Note: If the */etc/rc.net* file on the backup control workstation does not include the following statement, add the preceding line near the end of the */etc/rc.net* file to enable the backup control workstation to work as an IP router.

```
/usr/sbin/no -o ipforwarding=1
```

Otherwise, you will not be able to log in to the RISC/6000 SP nodes from the backup control workstation when it becomes the active control workstation.

2.2.5.5 Step 29: Verify the HACWS Configuration

1. Type `smit hacws`.
2. Select **Verify HACWS Installation and Configuration**.

Note: If you use HACMP for AIX V4.1.1 (not V4.1), you will need to install APAR Ix57291.

```
 ...
done <<EOF
ssp.clients     yes     2.1.0.0
ssp.authent     no      2.1.0.0
ssp.basic       yes     2.1.0.0
ssp.css         no      2.1.0.0
ssp.jm          no      2.1.0.0
ssp.sysman      yes     2.1.0.0
ssp.sysctl      yes     2.1.0.0
ssp.gui         yes     2.1.0.0
ssp.top         no      2.1.0.0
ssp.hacws       yes     2.1.0.0
cluster.client  yes     4.1.0.0
cluster.server  yes     4.1.0.0
EOF
 ...
```

Figure 19. hacws_verify - Before Modification

```
 ...
done <<EOF
ssp.clients     yes     2.1.0.0
ssp.authent     no      2.1.0.0
ssp.basic       yes     2.1.0.0
ssp.css         no      2.1.0.0
ssp.jm          no      2.1.0.0
ssp.sysman      yes     2.1.0.0
ssp.sysctl      yes     2.1.0.0
ssp.gui         yes     2.1.0.0
ssp.top         no      2.1.0.0
ssp.hacws       yes     2.1.0.0
EOF
#cluster.client  yes     4.1.0.0
#cluster.server  yes     4.1.0.0
 ...
```

Figure 20. hacws_verify - After Modification

2.2.5.6 Step 30: Verify the Hardware Connections

1. Type `smit hacws`.
2. Select **Verify Frame to Control Workstation Cabling**.

2.2.5.7 Step 31: Reboot Control Workstations

Reboot the primary control workstation. After it finishes booting, reboot the backup control workstation.

Attention

Until the HACMP cluster completes starting, do not use the *Common Desktop Environment (CDE)* session on the primary and backup control workstations. Instead, use *command line* mode or *X-Window*. If you start the HACMP cluster from a CDE window, CDE will get out of control as soon as HACMP changes the IP addresses from *boot* to *service* on the cluster network. This is because CDE requires strict consistency between the hostname and the actual IP address on the adapter.

2.2.5.8 Step 32: Start Cluster Services

Follow the instructions in Chapter 2 of the *HACMP for AIX Administration Guide*, SC23-2769 to start cluster services on each control workstation. Start cluster services on the primary control workstation first.

1. On the primary control workstation, type `smit clstart` and press Enter.
2. Select **now** for "Start now" option.

   Note: Do not use the *on system restart* option.
3. Select **true** for "Startup Cluster Information Daemon?" option.
4. Press Enter.
5. After the *OK* message is displayed on the smit menu, repeat steps 1 thru 4.

2.2.5.9 Step 33: Verify HACWS Installation

To make sure you installed HACWS correctly, start control workstation services and move them between the primary and backup control workstations.

1. Verify control workstation services.

   a. Service addresses should be configured on the primary control workstation. Try to telnet to a service address or use the following command:

      ```
      netstat -i
      ```

   b. */spdata* should be mounted on the primary control workstation. To verify this, issue the following command:

      ```
      df /spdata
      ```

      The */spdata* file system should be listed.

   c. The SDR should be available. To verify this, issue the following command:

      ```
      # /usr/lpp/ssp/bin/SDRGetObjects SP
      control_workstation cw_ipaddrs   install_image remove_image primary_
      fig   ntp_server   ntp_version  amd_config   print_config print_id
      config passwd_file  passwd_file_loc homedir_server homedir_path file
      supman_uid   supfilesrv_port spacct_enable spacct_actnode_thresh spa
      enable acct_master  cw_has_usr_clients code_version layout_dir   aut
      backup_cw    ipaddrs_bucw active_cw
      sp2cw0       9.12.20.37:9.12.0.70:9.12.0.37: bos.obj.ssp.41 false
            consensus     ""           3            true         false
          true         /etc/passwd  sp2cw0       sp2cw0       /home/sp2cw0
        102          8431         false                  80 false
      false        PSSP-2.1     ""           ssp          sp2cw1       9.1
      0.171:9.12.20.31: backup_cw
      #
      ```

2. Cause a failover.

   After control workstation services have started on the primary control workstation, tell HACMP to move control workstation services to the backup control workstation.

   a. On the primary control workstation, type `smit clstop`

   b. Select **now** for Stop now options.

   c. Select **takeover** for Shutdown mode option.

   d. Press Enter.

e. Make sure control workstation function services move to the backup control workstation. Follow the procedure described in step 1, “Verify control workstation services.”

   Also, you can check the LEDs on the Frame Supervisor Card. If the backup control workstation is correctly recognized by the Frame Supervisor Card as the active control workstation, LED “1,” “2,” and “3” are on in green color, and LED “7” is momentarily on in orange color. (See Figure 6 on page 14 and Table 2 on page 14 for the LED alignment and the definitions on the new Frame Supervisor Card.)

When you finish testing, restart cluster services on the primary control workstation.

Note: The first time you start control workstation services on the backup control workstation, the *spcw_apps* script runs and takes about a half hour for *setup_server* to complete. You must wait for it to complete before you move control workstation services back to the primary control workstation. You can tell whether the *spcw_apps* script has been completed by issuing the following command:

```
grep "SPCW_APPS COMPLETE" /tmp/hacmp.out
```

2.3 Customizing HACWS

This section describes how you can customize HACWS to make your RISC/6000 SP system more resilient to any component failures.

2.3.1 Typical HACWS Functional Flow

Before addressing customization, you should understand how HACWS works, that is, the typical functional flow of control workstation boot and failover. Also, you should be familiar with HACMP event processing functional flow in the HACWS environment.

2.3.1.1 Control Workstation HACMP Event Processing Flow

The HACWS software supplies HACMP pre- and post-event scripts (*HACWS_pre_event* and *HACWS_post_event*) for all HACMP events (this event customization is explained in 2.2.5.3, “Step 27: Customize Cluster Event Processing” on page 51). To provide additional pre- and post-event scripts to be called by the pre- and post-event scripts supplied with HACWS, the additional scripts must be located in the */var/adm/hacws/events* directory. The name of the scripts you supply must adhere to the following naming convention:

```
<event_name>.pre_pre_event

<event_name>.post_pre_event

<event_name>.pre_post_event

<event_name>.post_post_event
```

where *<event_name>* is the HACMP event name. For example, *node_up_local_complete.pre_pre_event* script will run at the *node_up_local_complete* event and before the *HACWS_pre_event* script. Figure 21 on page 56 shows the execution flow of HACMP and HACWS event scripts.

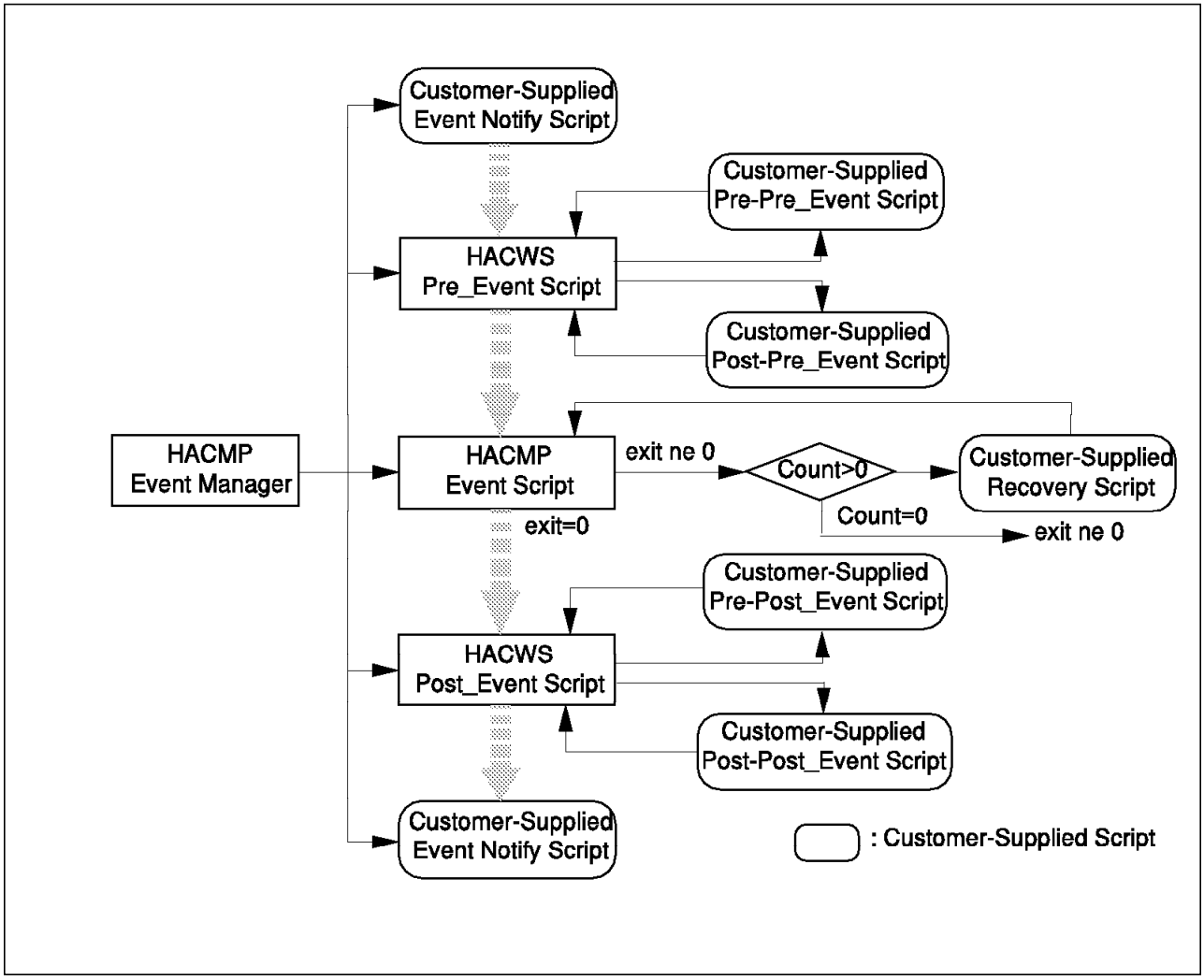


Figure 21. HACWS Event Processing Flow

2.3.1.2 Control Workstation Boot Functional Flow

1. The administrator boots the primary control workstation.
2. It boots to run level 2 without network initialization.
3. The HACWS state is set to *inactive*.
4. The administrator starts the HACMP cluster services.
5. HACMP runs `rc.net -boot` to configure and start the TCP/IP daemons correctly.
6. HACMP cluster manager performs takeover.
7. The *node_up* event starts with a message to the console.
8. The *node-up_complete* event script completes and the primary control workstation is acting as the control workstation.
9. The administrator starts the HACMP cluster manager on the backup control workstation.
10. The primary control workstation receives a message stating that the backup control workstation is joining the cluster.

11. The primary control workstation then receives a message stating the backup control workstation *node_up* event is complete.
12. Issue the HACMP `clstat` command to verify the cluster is operational.

2.3.1.3 Control Workstation Failover Functional Flow

Note: This occurs on the inactive control workstation.

1. As a result of a post *node_down_remote* event script, the following occurs:
   - */etc/hacws.state* is changed to active.
   - The virtual hostname is set if this script is running on the backup control workstation.
2. An *acquire_takeover_addr* event is generated.
3. A *get_disk_vg_fs* event is generated.
4. *node_down_complete* events are generated.
5. A *start_server* event is generated.
6. The `spcw_apps -u` command is executed.
7. As a post *start_server* event, the SDR is updated if required.

2.3.2 HACWS Customization Example

2.3.2.1 To Eliminate Single Point of Failure: CWS-Frame TTY Link

Some single points of failure are inevitable even after HACWS is installed. As shown in Table 3 on page 19, however, you can avoid some single points of failure by creating a user-supplied event script or by using AIX error notification. The following steps show how you can promote the failure of the RS232C cable between the control workstation and the RISC/6000 SP frame from a single point of failure to a recoverable node failure.

Attention

This procedure was tested only on the RS232C link failure. The failure of the frame supervisor card could cause the same error log entry. In such a case, you should modify the sample script to neglect the failure of the frame supervisor card, because the sample script will also cause the node failover. It is highly recommended to test your script intensively before you apply this procedure to your production system.

1. Create a shell script to be called by the error daemon and to cause node failover.

   The sample script is shown in Figure 22 on page 58. In this example, the script file should be located in the */tmp/hisashi* directory on both the primary and backup control workstations, and must have an *executable* mode.

```
#!/bin/sh
#############################################################################
#
#   Name:  fsc_tty_down
#
#   Purpose:  This is an script called by error daemon when RS232C link
#             between the CWS and the SP frame fails.
#             If this node is an active control workstation, then this
#             script causes a failover to the backup control workstation.
#
#   Returns:  0        -  success
#             nonzero  -  failure
#
#############################################################################
export PATH=/usr/bin:/usr/sbin:/usr/sbin/cluster/utilities
PROGNAME=$0
HACWS_RESOURCE_GRP=hacws_group1
LOCAL_NODE=get_local_nodename
STATUS=0

# Does the HACWS resource group exist?
NODE_LIST=clgetgrp -g $HACWS_RESOURCE_GRP -f nodes 2>/dev/null
if [[ $? -ne 0 ]]
then     # resource group doesn't exist
    exit 0
fi

# Does the local node belong to the resource group chain?  If not, then
# this cannot be an active control workstation, so don't do anything.
FOUND_LOCAL=no
for NODE in $NODE_LIST
do
    if [[ $NODE = $LOCAL_NODE ]]
    then        # Local node belongs to resource group chain.
        FOUND_LOCAL=yes
    fi
done
if [[ $FOUND_LOCAL = no ]]
then
    exit 0
fi

# Does the local node control the volume group(s).  If this is true, then
# we assume the local node currently controls the resource group.
eval clsetenvres $HACWS_RESOURCE_GRP node_down
for VG in $VOLUME_GROUP
do
    lsvg $VG >/dev/null 2>/dev/null
    if [[ $? -ne 0 ]]
    then
        # This node doesn't control the resource group.
        exit 0
    fi
done

# If we get to this point, then we need to failover.
exec clstop -N -gr -y -s             # exec should not return.

# If we reach this point, something is wrong.
echo "$PROGNAME: Cannot perform control workstation failover." > /dev/console
exit 1
```

Figure 22. Sample Script fsc_tty_down

2. Set the AIX Error Notification Method as follows:

   a. Enter the *HACMP Error Notification* menu by issuing the following command:

      ```
      smit cm_add_notifymeth.dialog
      ```

      or

      1) Type `smit hacmp`
      2) Select **RAS Support**.
      3) Select **Error Notification**.
      4) Select **Add a Notify Method**.

   b. Fill each field as follows:

```
                                Add a Notify Method

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

                                                  [Entry Fields]
* Notification Object Name                       [fsc_tty_failure]
* Persist across system restart?                  Yes
  Process ID for use by Notify Method            [0]
  Select Error Class                              Hardware
  Select Error Type                               Perm
  Match Alertable errors?                         None
  Select Error Label                             [SPMON_EMSG100_ER]
  Resource Name                                  [sphwlog]
  Resource Class                                 []
  Resource Type                                  []
* Notify Method                                  [/tmp/hisashi/fsc_tty_down]
```

   c. Press Enter.

   d. Verify your setting by issuing the following command:

      ```
      odmget errnotify
      ```

      Your definition will be found at the bottom of the following screen:

```
# odmget errnotify

 .. skipping ..
        en_class = ""
        en_type = ""
        en_alertflg = ""
        en_resource = ""
        en_rtype = ""
        en_rclass = ""
        en_symptom = ""
        en_method = "/usr/lib/ras/notifymethscarray -l $1 -r $6 -t $9"

errnotify:
        en_pid = 0
        en_name = "fsc_tty_down"
        en_persistenceflg = 1
        en_label = "SPMON_EMSG100_ER"
        en_crcid = 0
        en_class = "H"
        en_type = "PERM"
        en_alertflg = ""
        en_resource = ""
        en_rtype = ""
        en_rclass = ""
        en_symptom = ""
        en_method = "/tmp/hisashi/sp21cw0/fsc_tty_down"
#
```

3. Cause a Node Failover.

   After setting up the error notification method, cause a node failover by pulling out the RS232C cable from the tty port of the primary control workstation. A few seconds after you pull out the cable, the node failover process should start. You can monitor the process by entering the following command on both the primary and backup control workstations:

   ```
   tail -f /tmp/hacmp.out
   ```

   In a real failover case, you can exchange the failed RS232C cable while the control workstation service is running on the backup control workstation.

4. Reintegrate the Control Workstation Cluster after node failover and exchange the RS232C cable. If necessary, you should move the control workstation service back to the primary control workstation by issuing the following command:

   ```
   /usr/sbin/cluster/clstop -N -gr -y -s
   ```

   Do not forget to start the cluster on the backup control workstation again by entering the following command:

   ```
   /usr/sbin/cluster/clstart -m -b
   ```

2.3.2.2 To Failover Name Server (NIS)

See Chapter 2.3 of *HACMP/6000 Customization Examples*, SG24-4498 for setting up HACMP with the NIS environment. For HACWS, configure the primary control workstation as an NIS master server, and configure the backup control workstation as an NIS slave server.

Chapter 3. HACMP for RISC/6000 SP

As RISC/6000 SP systems are being integrated more and more into today′s commercial environment, high availability of the systems has become an important issue for the customers who run their mission critical applications on them. From the start, RISC/6000 SP has been designed with availability in mind. The hardware components, such as the RISC System/6000 and the High Performance Switch, and its operating system AIX, are all market proven to be particularly rich in features that improve system availability. However, in a mission critical commercial environment where a business′s wins and losses depend on the up time of the system, enhanced availability functions are definitely a plus. HACMP for AIX 4.1.1 on the RISC/6000 SP is an additional layer of software that could be used to further improve the availability of the RISC/6000 SP system.

HACMP for AIX Version 4.1.1 allows up to eight RISC/6000 SP processor “nodes” to be configured in a highly available cluster on a RISC/6000 SP. If one fails, its function can be assumed or “taken over” by another processor within the cluster. The backup processor can be doing highly available work, and can be backed up by another processor in the cluster. Multiple HACMP for AIX 4.1.1 clusters can be installed on a single RISC/6000 SP system, and a cluster can have nodes in a different RISC/6000 SP “frame” or “rack.”

Due to this flexibility, or perhaps to the complicated nature of a RISC/6000 SP system, it is important to consider potential single points of failure from various perspectives when designing clusters on the RISC/6000 SP system. For instance, if left unattended, a RISC/6000 SP node failure can be a single point of failure for a cluster application on a RISC/6000 SP system. However, if proper AIX error notification techniques are utilized, these failures can prompt HACMP to transfer affected workloads to backup processors, maintaining cluster availability. Understanding the system and the HACMP software is essential, and careful configuration and installation planning are critical for any successful High Availability implementation, either on a RISC/6000 SP system or cluster of RISC System/6000 systems. There are two excellent redbooks that cover the topics of high availability and HACMP extensively. They are *High Availability on the RISC System/6000 Family*, SG24-4551-00 and *HACMP/6000 Customization Examples*, SG24-4498-00.

The following chapter describes in detail how HACMP for AIX 4.1.1 can be used to cover some of the potential single points of failure on the system, and the “how to” implementation methods of installing those HACMP availability features. It is our intention to provide the generic high availability solutions for each of the identified single points of failure on a RISC/6000 SP system by using HACMP for AIX 4.1.1. Therefore, it is up to you to pick and choose the individual solution provided in this redbook to form the frame for your high availability solution.

3.1 HACMP Solutions Matrix for Potential Single Points of Failure on RS/6000 SP

This chapter provides High Availability solutions for potential, commonly known, single points of failure on a RISC/6000 SP system using HACMP. Table 5 lists the most common potential single points of failure existing on a RISC/6000 SP, and whether HACMP can provide a solution to cover each single point of failure. In addition, the table points to the sections in this chapter where the actual implementation steps are presented.

Table 5. HACMP Solution Matrix for Potential Single Points of Failure on RS/6000 SP

Potential single point of failure	HACMP Solution	Sections on HACMP Solution Implementation Methods
Node		
• Operating System	Yes	Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68
• Internal Disk	Yes	Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68
• Disk Adapter	Yes	Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68
• Communication Adapter	Yes	Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68
• HiPS Adapter	Yes	Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68, Section 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86
High Performance Switch		
• HiPS adapter	Yes	Section 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86
• Eprimay	Yes	Section 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86
• HiPS Global Network	Yes	Section 3.7, “Implementing High Availability for HiPS Network Failure” on page 105
Networks		
• Communication Network	Yes	Same solution methodology as HiPS Global Network (Section 3.7, “Implementing High Availability for HiPS Network Failure” on page 105), or customized solution should be implemented

For example, the High Performance Switch adapter can be a single point of failure because there is only one High Performance Switch adapter per node. By looking at the table, you can see that HACMP can provide a solution for this single point of failure, and the section that covers the implementation procedures for this solution is under Implementing HA for Eprimary and HiPS Adapter Failure. For a node failure, look into the section called Implementing HA for SP nodes implementation procedures. Therefore, if your environment has the need for a high availability solution for nodes and the High Performance Switch adapter, you can easily cut and paste the two sections together to form your implementation procedures.

3.2 RISC/6000 SP Sample Configuration

In order to cover as many High Availability Cluster Multi-Processing solutions on a RISC/6000 SP system, we have set up several clusters within our 16 thin nodes RISC/6000 SP frame for covering different common fail-over scenarios.
Figure 23 shows the structuring scheme of the RISC/6000 SP for this redbook.

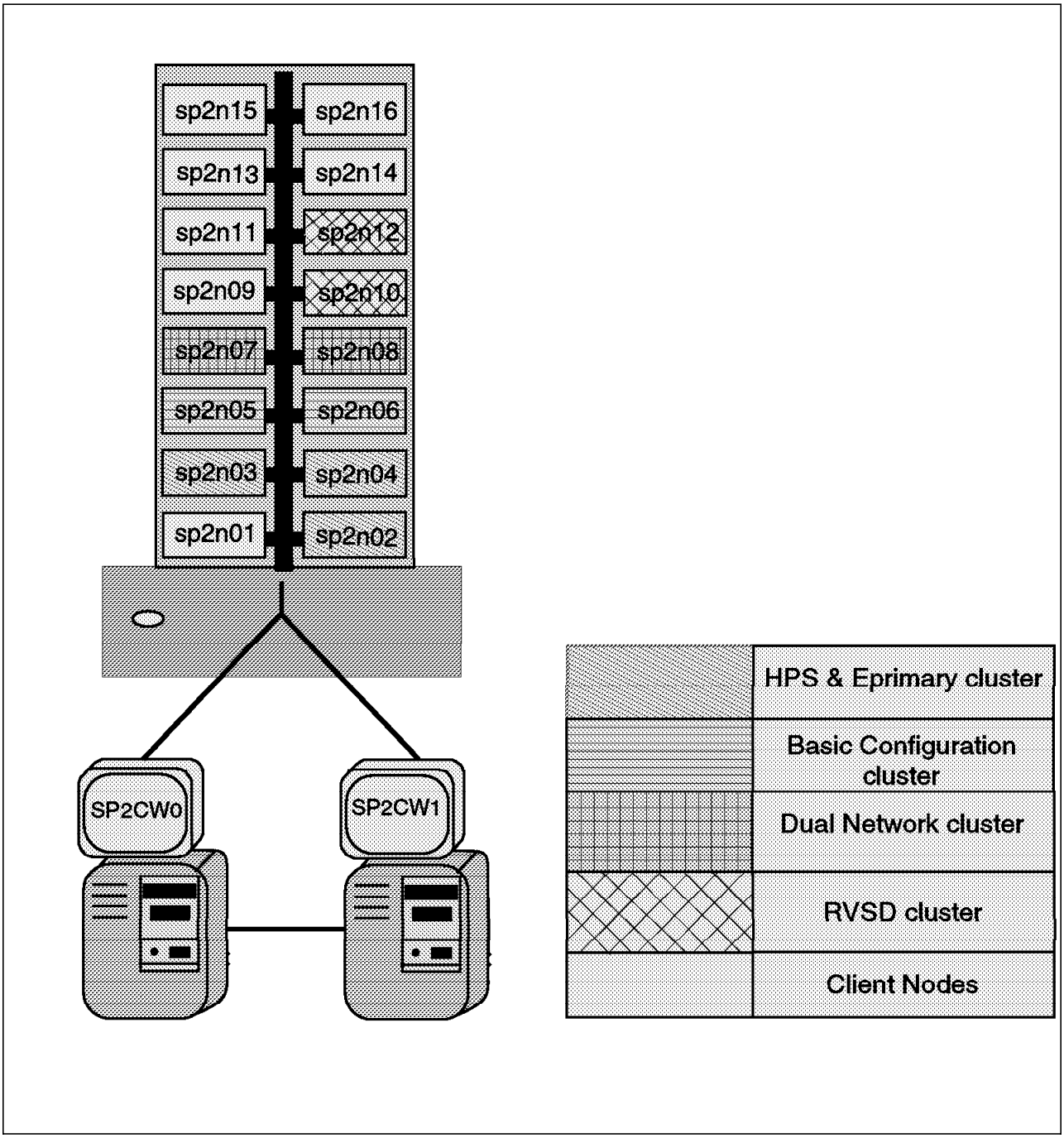


Figure 23. The System Structure Used in the Residency

Note that the RISC/6000 SP environment shown in this table is set up in such way that it is easy for us to illustrate the key points of the concept. Your environment could be quite different from ours; however, once the concepts are clearly understood, you can build your own High Availability Cluster Multi-Processing environment based on our models presented here.

3.3 Installing HACMP on the RISC/6000 SP System

As the basic requirement, HACMP for AIX Version 4.1.1 on the RISC/6000 SP requires IBM AIX 4.1.4 to be installed on all RISC/6000 SP nodes, which will be used to form the HACMP clusters. Please consult the *HACMP for AIX Installation Guide*, SC23-2769-01, to determine which filesets (install images) you will need to install. There are no unique HACMP for AIX 4.1.1 filesets for the RISC/6000 SP system.

3.3.1 HACMP Prerequisites

In addition to the basic requirement of IBM AIX Version 4.1.4, the following prerequisites must also be installed on the RISC/6000 SP control workstation and the RISC/6000 SP nodes, that are participating in the cluster or are clients of the HACMP clusters prior to the installation of HACMP:

- POWERparallel SP version 2 release 1 of the PSSP (AIX Parallel System Support Programs).
- Latest service level of the PSSP. At the time of this redbook, the minimum service level is level 8. This must include a PTF for component ssp.css at level 02.01.00.06.
- HACMP's prerequisite AIX filesets (for example X11.Dt, X11.vsm, bos.compat, bos.adt, and bos.data filesets), on all participating RISC/6000 SP cluster nodes.
- HACMP for AIX 4.1.1 filesets, on all participating RISC/6000 SP cluster nodes. For client nodes, you need HACMP for AIX 4.1.1 clients filesets only.
- At the time of this redbook, PTF U440482 is needed for the AIX 4.1.1 to fix the problem associated with changing a Logical Volume name. Changing the Logical Volume name may be required by HACMP during the configuration process. You may want to get the fix for PMR 0X957.

3.3.2 Installation Procedures

Use the following procedures to install the HACMP software on the control workstation and the nodes that will be used to form an HACMP cluster:

1. On the control workstation, load the software media and install the HACMP images on the directory by entering the following:

   ```
   /spdata/sys1/install/lppsource
   ```

   This screen shows the results of your input:

```
# smit install

          Install and Update Software ->
              Copy Software to Hard Disk for Future Installation ->
                  Select Input device
                  Select Software to copy
                        (all cluster filesets)
                  Input the name of the directory to copy the software
                        (/spdata/sys1/install/lppsource)
```

2. Update the .toc file on the directory /spdata/sys1/install/lppsource by entering the command:

   ```
   inutoc /spdata/sys1/install/lppsource
   ```

3. On the control workstation, create a file called /HACMPHOSTS. It should contain host names of the nodes in the SP machine that will have HACMP for AIX installed.

4. On the control workstation, enter the command:

   ```
   export WCOLL=/HACMPHOSTS
   ```

5. Ensure all SP hosts in the HACMPHOSTS are up (check hostResponds from spmon).

6. Add the directory /spdata/sys1/install/lppsource to the NFS export list (if it is not already added), by entering the command:

   ```
   /usr/sbin/mknfsexp -d '/spdata/sys1/install/lppsource' -t 'ro' '-B'
   ```

7. On the control workstation, enter the following command to NFS mount the directory /spdata/sys1/install/lppsource on all SP hosts included in the HACMPHOSTS as /mnt:

   ```
   dsh mount <CWNAME>:/usr/sys1/install/lppsource /mnt
   ```

   where CWNAME is the hostname of the control workstation.

8. On the control workstation, enter the following command to install the HACMP images on all SP hosts included in the HACMPHOSTS file. This must be done for each fileset you need to install.

   ```
   dsh "/usr/bin/installp -Xagd /mnt cluster.adt, cluster.base,
   cluster.man, cluster.vsm"
   ```

9. On the control workstation, enter the command to unmount the NFS file system on all nodes:

   ```
   dsh umount <CWNAME>:/spdata/sys1/install/lppsource /mnt
   ```

10. To verify that the HACMP was successfully installed on each node, enter the following command and check to see if all the HAMCP images have been installed.

    ```
    dsh "installp -s | grep cluster"
    ```

11. Reboot the nodes in the SP machine.

Note: In regard to HACMP for AIX 4.1.1 PTFs: At the time of this redbook, there is no formal PTF for the HACMP for AIX 4.1.1. However, there are several Emergency Fixes (Efix) available for the product, and their Efix numbers are

189568, 189571, 189573, 189574, and 189575. There is also a fix for PMR 0X957 that will be required.

3.4 Planning Considerations for HACMP on the RISC/6000 SP System

There are some specific considerations for HACMP on RISC/6000 SP for which one must plan carefully before and during the implementation phases of the HACMP cluster on a SP system, as follows:

1. IP Address Takeover (IPAT) for the SP HPS:
   - Address Resolution Protocol (ARP) must be enabled for the HiS network for any IP address takeover on the HiPS to work. This can be configured by an SP "customize" operation, or during the initial SP setup. Another method can also be used to enable the ARP for the HiPS, and it is presented in Appendix B, "Enabling Address Resolution Protocol on HiPS" on page 213. However, if this method is used, you must carefully, and completely follow the steps presented.
   - In an HACMP environment, HiPS boot and service addresses will be alias addresses on the HiPS css0 IP interface. The css0 interface can have more than one alias IP address; therefore it can support IP takeover addresses. At present, only one boot address and up to seven takeover addresses can be defined per HiPS css0 interface. Standby adapters/addresses are not required nor used for HiPS IP address takeover.
   - A User-Defined HACMP network name must contain the string "HiPS" in order for HACMP to recognize and execute IP takeover for HiPS addresses.
   - All HiPS addresses must be defined as a private network.
   - The css0 "base IP address," or the first IP address configured for the css0 interface (not an alias address), is unused and should not be configured for IP address takeover in HACMP. This ensures and maintains compatibility with the SP PSSP product.
   - The HiPS alias addresses for IPAT can be configured as a part of a cascading or rotating resource group.
   - Applications that make use of the HiPS switch network must be aware of fault characteristics of the switch. For example, during switch initialization and switch faults, power outage of a node will cause a switch fault. The switch network is unavailable on all nodes while the Eprimary node re-initializes the switch. These will appear as HACMP network down events followed by network up events for the HiPS interfaces on all cluster nodes. Applications that make use of the switch must be "tolerant" of these brief outages. In general, most TCP/IP applications are not affected by the switch re-initialization.
   - HACMP will attempt to initialize the switch network if it is not up. Since this utilizes the Eprimary node, two scenarios could happen.
     - If the Eprimary node is being managed by HACMP cluster, then the first node to enter the cluster will be assigned to be the Eprimary node and initialize the switch network.
     - If the Eprimary node is not being managed by HACMP and the Eprimary node is not available, then the switch initialization will not occur for any node.

2. Network Option:

   The following network options should be added to the last line of the /etc/rc.net file on all SP HACMP cluster nodes for the HiPS switch network.

```
no -o ipforwarding=0
     no -o ipsendredirects=0
     no -o thewall=8192
     no -o lowclust=72
     no -o lowmbufs=72
```

   Consult the "Tuning the System" section of the *SP Administration Guide*, GC23-3897-01, for more information about changing other network options to maximize performance based on your expected SP worktype and workload.

3. Kerberos:

   Kerberos is an authentication mechanism used on the SP system. Some commands such as rsh and rcp are modified for use in this authentication environment. However, HACMP uses /bin/rsh for global ODM updating done during verification and/or synchronization. Therefore, HACMP still requires that all nodes in a cluster have rsh (/.rhosts) root authority. In addition, should HACMP manage the switch but not the Eprimary node, .rhosts permissions should include that Eprimary node. Generally speaking, all nodes in any SP HACMP cluster should be able to execute /bin/rsh (through .rhosts permissions) to each other.

4. Automount Daemon:

   For SP installations that require the automount daemon (AMD) on HACMP nodes, a modification is needed to insure that AMD starts properly (with NFS available and running) on node bootup. This is due to the way HACMP for AIX manages the inittab file and run levels upon startup. To enable AMD on nodes that have HACMP for AIX installed, add the following line as the last line of the file /usr/sbin/cluster/etc/harc.net:

```
startsrc -s nfsd
```

5. SP Administrative Ethernet:

   Since some of the SP software has an assumption that an IP address on the SP administrative Ethernet (en0) is associated with a specific node, IP address takeover (as defined in cascading or rotating resource groups), cannot be configured on this network. However, you should configure this adapter into the HACMP cluster so that the network will be monitored by the cluster manager. This is done by defining the SP Ethernet IP address as a service address to HACMP, but it must *not* be configured for IP address takeover (do not configure a boot address). In addition, no owned or takeover resources can be associated with this adapter.

6. Serial or Non-IP Network:

   It is strongly recommended (but not required) that a non-IP network be present between nodes that share resources, in order to eliminate TCP/IP as

a single point of failure. HACMP for AIX 4.1.1 supports target mode SCSI, serial network with the use of an 8-port EIA-232 serial adapter card.

7. High Performance Switch:

The switch is designed in such a manner that each switch chip services 4 nodes, and in an SP frame, there will be 4 switch chips handling connections to all the nodes. If one of the chips fails, then communication through the switch for all 4 nodes serviced by the chip will be lost. Therefore, when configuring HACMP on an RISC/6000 SP, one must consider how the nodes should be chosen so that they will not be on the same switch chip and become a single point of failure. For more details on the High Performance Switch, see 4.1, “High Performance Communication Network” on page 137.

8. The /etc/hosts File and nameserver Configuration:

Make sure all nodes can resolve all cluster addresses. Edit the /etc/hosts file (and the /etc/resolv.conf file, if using the nameserver configuration) on each node in the cluster to make sure the IP addresses of all cluster interfaces are listed. Also, make sure that /etc/hosts file on each node has the following entry:

```
127.0.0.1   loopback localhost
```

3.5 Implementing High Availability for RISC/6000 SP Nodes

The previous section identified some of the potential single points of failure in an RISC/6000 SP node. The occurrence of any one of these failures results in that node′s failure or isolation.

Node Failure Event

Potential single points of failure in an RISC/6000 SP node

1. RISC/6000 SP node processor failure
2. Operating system crashes or fails
3. Internal disk adapter fails
4. Internal disk fails
5. External disk adapter fails
6. HiPS adapter fails
7. Integrated Ethernet port fails

The objective is to keep the RISC/6000 SP node′s applications running and highly available for its clients. The strategy is to keep the host operational with as much internal redundancy and operating system safeguards as possible or practical. And, if all else fails or as designed, the strategy is to failover to a backup node and restart the applications there.

3.5.1 Sample Two-Node RISC/6000 SP HACMP Cluster

An application called LoadLeveler is made highly available as shown in Figure 24 on page 70. As described in Chapter 6, “Implementing LoadLeveler for High Availability” on page 185, LoadLeveler job queues are maintained by LoadLeveler scheduler nodes. The failure of a scheduler node results in its job queues and job status becoming unavailable to its client users. Jobs in the queue are not lost, but their disappearance from status screens results in users mistakenly considering their jobs lost. Users then resubmit their jobs, thus creating duplicate jobs. This can be disastrous if the jobs are very compute-intensive.

In this example, the application needs to maintain:

- The same IP address as the failed scheduler.
- The same hostname as the failed scheduler.

Table 6. HACMP Solution Matrix for RISC/6000 SP Node Failure Configuration

Potential Single Point of Failure	HACMP Solution	Coverage in Section
RISC/6000 SP Node		
• Processor failure	Failover to backup node	Covered in this section
• Operating system crashes	Failover to backup node	Covered in this section
• Internal disk adapter fails	Failover to backup node	Covered in this section
• Internal disk fails	Failover to backup node	Covered in this section
• External disk adapter fails	Failover to backup node	Covered in this section
• Integrated Ethernet port fails	Failover to backup node	Covered in this section
• HiPS adapter fails	Failover to backup node	Covered in this section and 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86

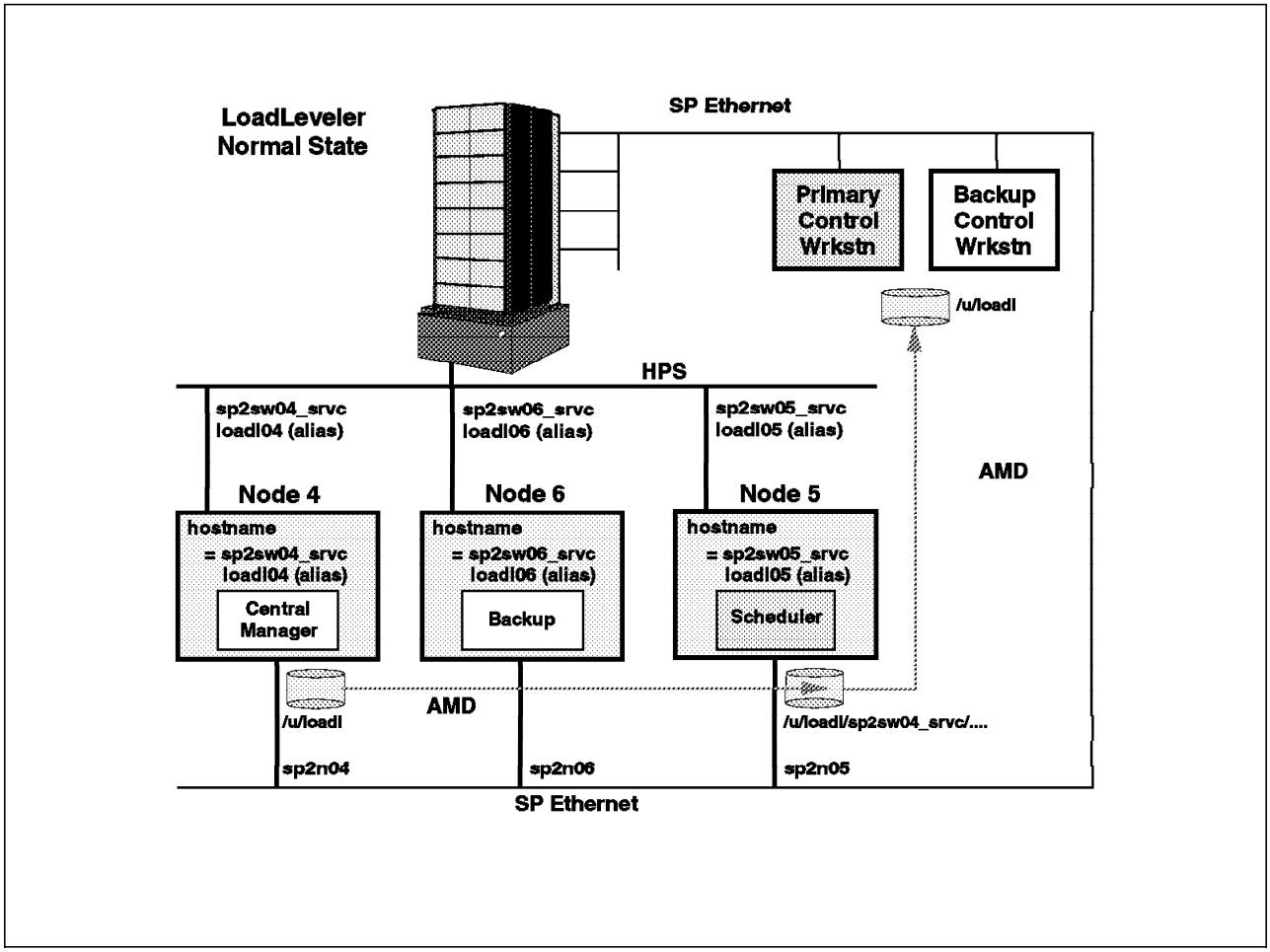


Figure 24. Sample RISC/6000 SP Basic High Availability Configuration

3.5.2 Overview of HACMP Installation Process

The HACMP installation process for the RISC/6000 SP is basically the same as for the clustered RISC System/6000. The steps are divided into two major areas:

Preparing AIX for an HACMP cluster and setting up the hardware.

Installing the software and configuring HACMP.

The following sections highlight the steps involved and focus on the RISC/6000 SP specific implementation topics. Refer to the *High Availability Cluster Multi-Processing 4.1 for AIX Installation Guide*, SC23-2769, for more detailed information on High Availability Cluster Multi-Processing planning and installation.

3.5.3 Set the RISC/6000 SP Hardware and Software for HACMP

After identifying which RISC/6000 SP nodes will be configured, verify the system components and hardware installation assigned for this cluster. Once everything is found ready, set up the AIX and the system environment as follows:

Step 1. Configuring the Network

Step 2. Install the Shared Disk Devices

Step 3. Define Shared LVM Components

Step 4. Customize AIX

3.5.3.1 Step 1. Configure the Network and Network Adapters

Typically the RISC/6000 SP is installed and configured prior to HACMP. As such, the RISC/6000 SP node's reliable Ethernet address and High Performance Switch adapter base address are already defined and configured. Do not reconfigure or change these interfaces. At this point, configure only the standby adapters and external communications adapters required.

See *High Availability Cluster Multi-Processing 4.1 Installation Guide*, SC23-2769, for details on configuring the different network types, including the RS232 serial line. Ensure that either an RS232 serial line or target mode SCSI line is configured for HACMP heartbeat.

In our test configuration, the RS232 serial line was the only other communications network setup for the cluster.

3.5.3.2 Step 2. Install Shared Disk Devices

Dual adapters and dual paths to disks are the recommended configurations for high availability. If using SCSI devices, this is accomplished by twin-tailing the supported devices and mirroring their filesystems or raw files across disks and adapters. If you are using a 7133 disk, the SSA technology and cabling system provides bi-directional access to any disk in the loop. With multiple loop configurations, mirroring of the filesystems and files can also be accomplished.

See *High Availability on the RISC System/6000 Family, SG24-4551* and *High Availability Cluster Multi-Processing 4.1 Installation Guide*, SC23-2769, for detailed information on these disk subsystems.

Our test configuration used a 7133 SSA disk subsystem to demonstrate the use of shared disk devices.

3.5.3.3 Step 3. Define Shared LVM Components

Once the shared disk devices are defined and configured, the next step is creating the volume groups, logical volumes, and the file systems shared by the nodes in an HACMP cluster. In general, you configure the LVM components from one node, then use importvg, which is the component on the other node. This ensures that the ODM definitions of the shared components are the same on all the cluster nodes.

See *High Availability Cluster Multi-Processing 4.1 Installation Guide* for different instructions for defining the concurrent and non-concurrent shared LVM components. You may want to refer to 2.2.3.5, “Step 17: Set Up External File System” on page 43.

3.5.3.4 Step 4. Customize AIX for HACMP

To ensure that your HACMP for AIX cluster environment works as planned, consider the following issues as they pertain to the cluster. These issues include:

I/O pacing

User and group IDs

Network option settings

/etc/hosts file and nameserver edits

/.rhosts file edits

NFS configurations

See *High Availability Cluster Multi-Processing 4.1 Installation Guide* for addressing some of these considerations.

In this test configuration and application, the scheduler application is given a hostname *sp2sw05_srvc* with a service address associated to it in the /etc/hosts file.

3.5.4 Configure High Availability Cluster Multi-Processing

Once the system environments are set, we proceed with the HACMP configuration process as follows:

Step 5. Installing the HACMP software on the RISC/6000 SP nodes

Step 6. Configuring the cluster

Step 7. Defining the nodes

Step 8. Defining the adapters

Step 9. Defining the resource groups

Step 10. Adding application servers

Step 11. Adding resources to resource groups

Step 12. Synchronizing the cluster nodes

Step 13. Verifying and testing the configuration

3.5.4.1 Step 5. Install HACMP Software on RISC/6000 SP Nodes

Installing HACMP on the RISC/6000 SP nodes can be done through an install server or through a remotely mounted install image. For details, see section 3.3, “Installing HACMP on the RISC/6000 SP System” on page 64.

After installing HACMP, enter `smit hacmp` to access the configuration panels. For those familiar with versions earlier than 4.1.1, you may experience some differences when navigating through this *smit* configuration. Once you get to the panel you need, the underlying commands are the same. Note the new main panels, shown below.

To access the High Availability Cluster Multi-Processing configuration panels and services, enter:

```
# smit hacmp
```

```
                                HACMP for AIX

Move cursor to desired item and press Enter.

=>Cluster Configuration
  Cluster Services
  Cluster Recovery Aids
  RAS Support
```

These items are explained as follows:

Cluster Configuration
Where to start configuration process.

Cluster Services
Where to start and stop the cluster.

Cluster Recovery Aids
Tools to recover from failed scripts or to reset disk fencing.

RAS Support
Tools to debug and trace cluster events.

```
                          Cluster Configuration

Move cursor to desired item and press Enter.

=>Cluster Topology
  Cluster Resources
  Cluster Snapshots
  Cluster Verification
```

These items are explained as follows:

Notes:

1. *Cluster Topology* covers steps 1 thru 4 of the process.
2. *Cluster Resources* covers steps 5 thru 8 of the process.
3. *Cluster Verification* covers steps 9 and 10 of the process.

Cluster Topology
Where to add/change/show nodes and node adapters.

Cluster Resources
Where to add/change/show application resources and groups.

Cluster Snapshots
Tools for capturing cluster configurations. Snapshots can be used for debugging, change control or for reference.

Cluster Verification
Tools for verifying cluster configuration and resources.

3.5.4.2 Step 6. Configure the Cluster

To configure the cluster, select:

= > **Configure Cluster**

```
                                Cluster Topology

Move cursor to desired item and press Enter.

=>Configure Cluster
  Configure Nodes
  Configure Adapters
  Configure Network Modules
  Show Cluster Topology
  Synchronize Cluster Topology
```

Then select:

= > **Add Cluster Definition**

```
                                Configure Cluster

Move cursor to desired item and press Enter.

  Add a Cluster Definition
  Change / Show Cluster Definition
  Remove Cluster Definition
```

```
                             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                                        [10]
* Cluster Name                                      [Cluster10]
```

Note: Subsequent *smit* screen illustrations show the entry panels only. The steps leading to these panels will be shown in text with the *= >* symbol preceding them.

The preceding screen will execute the following.

/usr/sbin/cluster/utilities/claddclstr

The values entered for the following will be used:

Cluster ID Enter an integer value as a cluster ID.

Cluster Name
Assign a name that uniquely identifies the cluster. This is necessary in case there is more than one cluster on a single physical network.

3.5.4.3 Step 7. Define Nodes

Enter:

```
# smit hacmp
```

Select the following options:

= > **Cluster Configuration**
= > **Cluster Topology**
= > **Configure Nodes**
= > **Add Cluster Nodes**

```
                                Add Cluster Nodes

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

                                                [Entry Fields]
* Node Names                                   [sp2n05 sp2n06]
```

The preceding screen will execute the following:

```
/usr/sbin/cluster/utilities/clnodename
```

The values entered for the following will be used:

Node Names

Assign a unique name for each node in the cluster. List all the node names, using a space character to delimit each name. Node names need not be the same as the hostnames given in the /etc/hosts file. The use of descriptive names provides more clarity and ease of use.

3.5.4.4 Step 8. Add Node Adapters

Enter:

```
# smit hacmp
```

Select the following options:

= > **Cluster Configuration**
= > **Cluster Topology**
= > **Configure Adapters**
= > **Add Adapter**

```
                                Add an Adapter

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

                                                [Entry Fields]
* Adapter IP Label                             [sp2sw05_srvc]
* Network Type                                 [hps]
* Network Name                                 [HPS1]
* Network Attribute                              private
* Adapter Function                               service
  Adapter Identifier                           []
  Adapter Hardware Address                     []
  Node Name                                    [sp2n05]
```

The preceding screen will execute the following:

```
/usr/sbin/cluster/utilities/claddnode
```

The values entered for the following will be used:

Adapter Label
This is the name of the adapter that you have chosen to define the High Availability Cluster Multi-Processing. The naming conventions used in the example are <interface>_srvc for the service adapters and <interface>_boot for the boot adapters.

Network Type
This determines the type of network to which the adapter is connected. On the RISC/6000 SP, the internal Ethernet is defined as type *ether* and the High Speed Switch is defined as type *HPS* (HiPS). These are both preinstalled network interface modules (NIM) and can be obtained from the smit panel picklist.

Network Name
For High Availability Cluster Multi-Processing Version 4.1.1. on the RISC/6000 SP, the High Speed Switch network must be defined with the string *HPS* somewhere in the name of the network. For example, *HPS_net* enables the scripts to identify it as the Switch network.

Note: The High Performance Switch abbreviation use to be HPS and it has been changed to HiPS. Some of the SMIT menu display may still have it as HPS.

Network Attribute
This can be either *public*, *private*, or *serial*. The HiPS must be defined as a private network and the RISC/6000 SP internal Ethernet must be defined as a public network. The RS232 serial connection is defined as a serial network.

Adapter Function
Defined as either *service*, *standby*, or *boot*. The HiPS has a service and boot adapter defined for it, but no standby. The internal Ethernet is defined to HACMP with only a service address so that it can be monitored by HACMP for AIX. It cannot be used for IP address takeover. Some RISC/6000 SP software such as the SDR associates the en0 internal Ethernet adapter with a specific node. The internal Ethernet can also not have any resources associated with it.

Adapter Identifier
Enter the IP address associated with the adapter. This is optional, if you are defining an adapter whose IP label is already listed in the /etc/hosts file. If this is the adapter for the serial network, the Adapter Identifier should be entered as */dev/tty* n.

Note: The IP addresses specified for the HiPS service and boot addresses are alias addresses on the css0 interface. They are

separate from the base IP address (the initial IP address configured on the css0 adapter). The base IP address must not be configured for IP address takeover in HACMP for compatibility reasons with the PSSP software, which associates the base IP address with a specific node. This address can therefore not be moved with IP address takeover. The boot and service addresses must also be defined on a different subnet, otherwise the Base IP address clverify will fail when it is run.

Adapter Hardware address
This is optional unless hardware address swapping is to be appropriately configured. IP hardware address swapping is not possible on the HiPS adapters.

Node Name
This associates the address with a particular node. The only case where the node name is not associated with a node is when the service address is shared as part of a rotating resource.

Upon completing Step 4, verify the whole cluster topology by entering:

```
# smit hacmp
```

Select the following options:
= > **Cluster Configuration**
= > **Cluster Topology**
= > **Show Cluster Topology**

A sample output of this verification step can be seen in Figure 25 on page 78, followed by Figure 26 on page 79.

```
Cluster Description of Cluster Cluster10
Cluster ID: 10
There were 2 networks defined : HPS1, sp_ether
There are 2 nodes in this cluster.

NODE sp2n05:
        This node has 2 service interface(s):

        Service Interface sp2sw05_srvc:
                IP address:     9.12.23.22
                Hardware Address:
                Network:        HPS1
                Attribute:      private

        Service Interface sp2sw05_srvc has a possible boot configuration:
                Boot (Alternate Service) Interface: sp2sw05_boot
                IP address:     9.12.23.72
                Network:        HPS1
                Attribute:      private

        Service Interface sp2sw05_srvc has no standby interfaces.

        Service Interface sp2n05:
                IP address:     9.12.20.55
                Hardware Address:
                Network:        sp_ether
                Attribute:      public

        Service Interface sp2n05 has no standby interfaces.

        Service Interface sp2n05_tty:
                IP address:     /dev/tty1
                Hardware Address:
                Network:        Serial_net
                Attribute:      serial

        Service Interface sp2n05_tty has no standby interfaces.
```

Figure 25. Cluster 10 Topology (1 of 2)

```
NODE sp2n06:
        This node has 2 service interface(s):

        Service Interface sp2sw06_srvc:
                IP address:     9.12.23.23
                Hardware Address:
                Network:        HPS1
                Attribute:      private

        Service Interface sp2sw06_srvc has a possible boot configuration:
                Boot (Alternate Service) Interface: sp2sw06_boot
                IP address:     9.12.23.73
                Network:        HPS1
                Attribute:      private

        Service Interface sp2sw06_srvc has no standby interfaces.

        Service Interface sp2n06:
                IP address:     9.12.20.56
                Hardware Address:
                Network:        sp_ether
                Attribute:      public

        Service Interface sp2n06 has no standby interfaces.

Breakdown of network connections:

Connections to network HPS1
        Node sp2n05 is connected to network HPS1 by these interfaces:
                sp2sw05_boot
                sp2sw05_srvc

        Node sp2n06 is connected to network HPS1 by these interfaces:
                sp2sw06_boot
                sp2sw06_srvc

Connections to network sp_ether
        Node sp2n05 is connected to network sp_ether by these interfaces:
                sp2n05

        Node sp2n06 is connected to network sp_ether by these interfaces:
                sp2n06
```

Figure 26. Cluster 10 Topology (2 of 2)

3.5.4.5 Step 9. Define the Resource Group

In this section we define two cascading resource groups:

5to6_res_grp

6_res_grp

Resource group *5to6_res_grp* will consist of the service address and a set of start/stop scripts that will define which LoadLeveler configuration to use.

Resource group *6_res_grp*will consist of just the service address of node 6. This acts as a place holder for the service address and avoids any unwarranted clverify errors or messages regarding not having configured a service address with a boot address.

Enter:

```
# smit hacmp
```

Select the following options:
= > **Cluster Configuration**
= > **Cluster Resources**
= > **Define Resource Groups**
= > **Add Resource Group**

```
                              Add a Resource Group

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

                                                [Enter Fields]
* Resource Group Name                           [5to6_res_grp]
* Node Relationship                               cascading
* Participating Node Names                      [sp2n05 sp2n06]
```

```
                              Add a Resource Group

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

                                                [Enter Fields]
* Resource Group Name                           [6_res_grp]
* Node Relationship                               cascading
* Participating Node Names                      [sp2n06]
```

The preceding screen will execute the following:

```
/usr/sbin/cluster/utilities/claddgrp
```

The values entered for the following will be used:

Resource Group Name
Creates a string of characters that describes the resource group. The name must not have any spaces or special High Availability Cluster Multi-Processing reserved characters.

Node Relationship
Select *cascading*, *rotating*, or *concurrent*. In this example, *cascading* was configured to allow the primary host to reclaim its resources and restart its application upon rejoining the cluster.

Participating Node Names
Name the node with the highest priority *first*. In this example the primary is first on the list for claiming the resources.

3.5.4.6 Step 10. Add Application Servers

Application servers are names given to High Availability Cluster Multi-Processing resources which provide the nodes with the start and stop scripts to use for proper application start, restart and shutdown. The definition consists of:

- Server name
- Application start script
- Application stop script

Once you have created the scripts necessary to start and stop the application, copy them to both nodes and provide their full pathnames in this section.

Enter:

```
# smit hacmp
```

Select the following options:
= > **Cluster Configuration**
= > **Cluster Resources**
= > **Define Application Servers**
= > **Add an Application Server**

```
                          Add an Application Server

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

                                  Entry Fields
* Server Name                   [str_stp_5on6]
* Start Script                  [/u/loadl/hascripts/start_5on6]
* Stop Script                   [/u/loadl/hascripts/stop_5on6]
```

The preceding screen will execute the following:

```
/usr/sbin/cluster/utilities/claddserv
```

The values entered for the following will be used:

Server Name
Enter a suitable name which describes the application being made highly available.

Start Script
Enter the full pathname for the start script.

Stop Script Enter the full pathname for the stop script.

In this example, &l. is started on the backup node with:

```
/u/loadl/hascripts/start_5on6
```

When the primary rejoins the cluster and reclaims the application, LoadLeveler is stopped on the backup node by:

```
/u/loadl/hascripts/stop_5on6
```

Once this is done, an application server named *5to6_server* has been defined. It can be added into a resource group and controlled by High Availability Cluster Multi-Processing.

3.5.4.7 Step 11. Add Resources to Resource Groups

Once a Resource group has been defined, resources can be added to it.

Enter:

```
# smit hacmp
```

Select the following options:

= > **Cluster Configuration**
= > **Cluster Resources**
= > **Change/Show Resources for a Resource Group**
= > **Configure Resources for a Resource Group**

```
                    Configure Resources for a Resource Group

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

TOP                                          [Enter Fields]
  Resource Group Name sp2n08 sp2n07             5to6_res_grp
  Node Relationship   sp2n08 sp2n07             cascading
  Participating Node Names                      sp2n05 sp2n06

  Service IP label                             [sp2sw05_srvc]
  Filesystems                                  []
  Filesystems to Export                        []
  Filesystems to NFS mount                     []
  Volume Groups                                []
  Concurrent Volume groups                     []
  Raw Disk PVIDs                               []
  Application Servers                          [str_stp_5]
  Miscellaneous Data                           []
```

The preceding screen will execute the following:

```
/usr/sbin/cluster/utilities/claddres -g
```

The values entered for the following will be used:

Service IP Label

By filling in the label of *sp2sw05_srvc* here, we activate IP address takeover. If the primary node *sp2n05*fails, its service IP address will be transferred to the backup node. If we had left this field blank, there would be no IP address takeover from the primary to the backup node.

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 be automatically varied on.

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 is currently 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.

Volume Groups

This field does not need to be filled in if a filesystem has already been specified above. High Availability Cluster Multi-Processing discovers which volume group it needs to vary on in order to mount the filesystems we have defined.

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 LVM, so an application using RAID-3 would have to read and write directly to the hdisk service.

Application Server

For any application servers that are defined here, High Availability Cluster Multi-Processing 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.

3.5.4.8 Step 12. Synchronize the Cluster Nodes

Once the definitions have been entered for the topology, they must be copied to the other nodes so that the definitions across all nodes are the same. This will update the ODM on all nodes defined in the cluster. Perform the synchronization from the node where all the cluster configuration have been made. To execute:

Enter:

```
# smit hacmp
```

Select the following options:

= > **Cluster Configuration**
= > **Cluster Resources**
= > **Synchronize Cluster Resources**

3.5.4.9 Step 13. Verifying and Testing the Cluster

To ensure that the cluster configuration is correct with regards to the rules governed by High Availability Cluster Multi-Processing, use the */usr/sbin/cluster/diag/clverify* utility. This can be run from the command line or through the smit option *Cluster Verification*.

Another option that is available, but only from the command line, is the clverify topology check option which checks consistency across all nodes in the cluster. You can also use the *sync* option to synchronize all the cluster nodes with the local node's definition.

/usr/sbin/cluster/diag/clverify cluster topology check
/usr/sbin/cluster/diag/clverify cluster topology sync

When High Availability Cluster Multi-Processing verifies the cluster, clverify may fail with the following error messages that follow:

```
Verifying Cluster Topology...

ERROR: Node sp2n05 is not on its boot address.
ERROR: Node sp2n06 is not on its boot address.
```

The reason for the error messages is because the HiPS adapters, at the present time, have not been aliased with the boot addresses. Therefore, when HACMP performs the checking, it will notice that the HiPS adapter is not on the proper address, and issue the messages. However, these messages can be ignored, since HACMP will actually place the alias boot address on the HiPS adapter if it cannot find one. It will then change the boot address to the alias service address during the startup process. However, if you want to eliminate the messages, you could either manually assign the alias boot address onto the HiPS adapter, or set an entry in the /etc/rc.net to alias the boot address when the system restarts. Again, even though these messages can be ignored, please scan the entire clverify output for other ERROR messages before proceeding.

To verify that the cluster is behaving as designed, we performed the following tests:

1. Verify IP address takeover by simulating node failure of sp2n05 and verifying that the backup node takes over the required service address: *sp2sw05_srvc* and the disk resource, *demo_vg*

   Test this by issuing the command:

   ```
   smit clstop
   ```

   ```
                              Stop Cluster Services

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

                                                          [Entry Fiels]
   * Stop now, on system restart or both                  now           +

     BROADCAST cluster shutdown?                          true          +
   * Shutdown mode                                        takeover      +
            (graceful, graceful with takeover, forced)
   ```

   The expected behavior is the following:

   - The service address, *sp2sw05_srvc* should be taken over by the HiPS adapter on the backup node sp2n06.

   Method used for testing:

   - Issue the command *netstat -i* on the backup node, sp2n06. Find the *sp2sw05_srvc* address up on its css0 HiPS interface together with its own *sp2sw06_srvc* service address.

- On the takeover node sp2n06, we should see the following with the netstat -i:

```
sp2n06-root / -> netstat -i

Name  Mtu   Network     Address       Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 <Link>                      377042     0   378384     0
lo0   16896 127         localhost.in-ad   377042     0   378384     0
en0   1500  <Link>10.0.5a.fa.3.33        1290017     0  1155986     0
en0   1500  9.12.20     sp2n06.itsc.pok  1290017     0  1155986     0
css0  65520 <Link>0.0.0.0.0.0             915425     0   718255 51230
css0  65520 9.12.6      sp2sw06.itsc.po   915425     0   718255 51230
css0  65520 9.12.23     sp2sw06_srvc.it   915425     0   718255 51230
css0  65520 9.12.23     sp2sw05_srvc.it   915425     0   718255 51230
```

- Issue the lsvg -o command on the backup node, sp2n06, and verify that the *demo_vg* is varied on and available.

```
# lsvg -o

rootvg
demo_vg
```

2. Verify proper reintegration of the primary node, sp2n05. The Primary node, sp2n05, should reclaim its *sp2sw05_srvc* and the disk resource: *demo_vg* upon rejoining the cluster.

   Verify this by issuing the command:

   `smit clstart`

   on the primary node, sp2n05.

```
                          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?                      true
  Startup Cluster Lock Services?                     false
  Startup Cluster Information Daemon?                true
```

   The expected behavior is the following:

   - The service address, *sp2sw05_srvc*, should return to the primary node, sp2n05. The disk resource should return to the primary as well.

   The method used for testing is the following:

   - Issue the command, netstat -i, on the primary node, and you should find the service address, *sp2sw05_srvc*, back on the node.

```
sp2n05-root / -> netstat -i

Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Co
lo0   16896 <Link>                       264665     0   266027     0
lo0   16896 127         localhost.in-ad   264665     0   266027     0
en0   1500  <Link>10.0.5a.fa.1a.92       1156301     0   977862     0
en0   1500  9.12.20     sp2n05.itsc.pok  1156301     0   977862     0
css0  65520 <Link>0.0.0.0.0.0             956805     0   711860  6400
css0  65520 9.12.6      sp2sw05.itsc.po   956805     0   711860  6400
css0  65520 9.12.23     sp2sw05_srvc.it   956805     0   711860  6400
```

- Issue the command:

```
# lsvg -o
rootvg
demo_vg
```

3. Repeat verification steps 1 and 2, but instead of a graceful takeover with a shutdown of the primary node, do an actual power off on the system. The cluster should behave as in steps 1 and 2.

3.6 Implementing High Availability for Eprimary and HiPS Adapter Failure

The High Performance Switch adapters in the nodes and the Eprimary node are frequently cited as potential single points of failure. The Eprimary node is the node designated to manage the High Performance Switch initialization and recovery for the RISC/6000 SP system. If the Eprimary node failed, then the function of the Eprimary would not be available, and this would potentially impact the whole system since there is no in-built failover mechanism for the function. There is only one High Performance Switch adapter in each node, and the failure of the adapter in the Eprimary node would result in the loss of initialization and recovery control for the entire system. Therefore, the failure of the Eprimary node or the switch adapter on the Eprimary could impact not just the node itself, but also the operation of the entire RISC/6000 SP system.

However, with the recent announcement of the new SP Switch, the perception of the Eprimary node as a potential single point of failure is eliminated with an built-in recovery mechanism that provides for a backup Eprimary node in the event of a failure.

If your system is installed with the High Performance Switch (HiPS), the use of HACMP for AIX 4.1.1 running on RISC/6000 SP could provide the solution for both scenarios. HACMP can effectively reassign the Eprimary node in case of Eprimary node failure. This reassignment is made among all nodes in the cluster with the High Performance Switch, and is done by way of a rotating resource algorithm (the lowest alphabetical node that is up acquires the Eprimary reassignment). Also, to eliminate the High Performance Switch adapter as a single point of failure, one will need to use the AIX error notification configuration in HACMP to promote the High Performance Switch adapter failure to a node failure, so that the IP address of the failed High Performance Switch adapter could be taken over by its backup node. This would eliminate this single point of failure.

The following figure shows the cluster environment that was set up in our RISC/6000 SP system to test and to illustrate the implementation of High Availability Cluster Multi-Processing solution for the High Performance Switch adapter and Eprimary on a RISC/6000 SP:

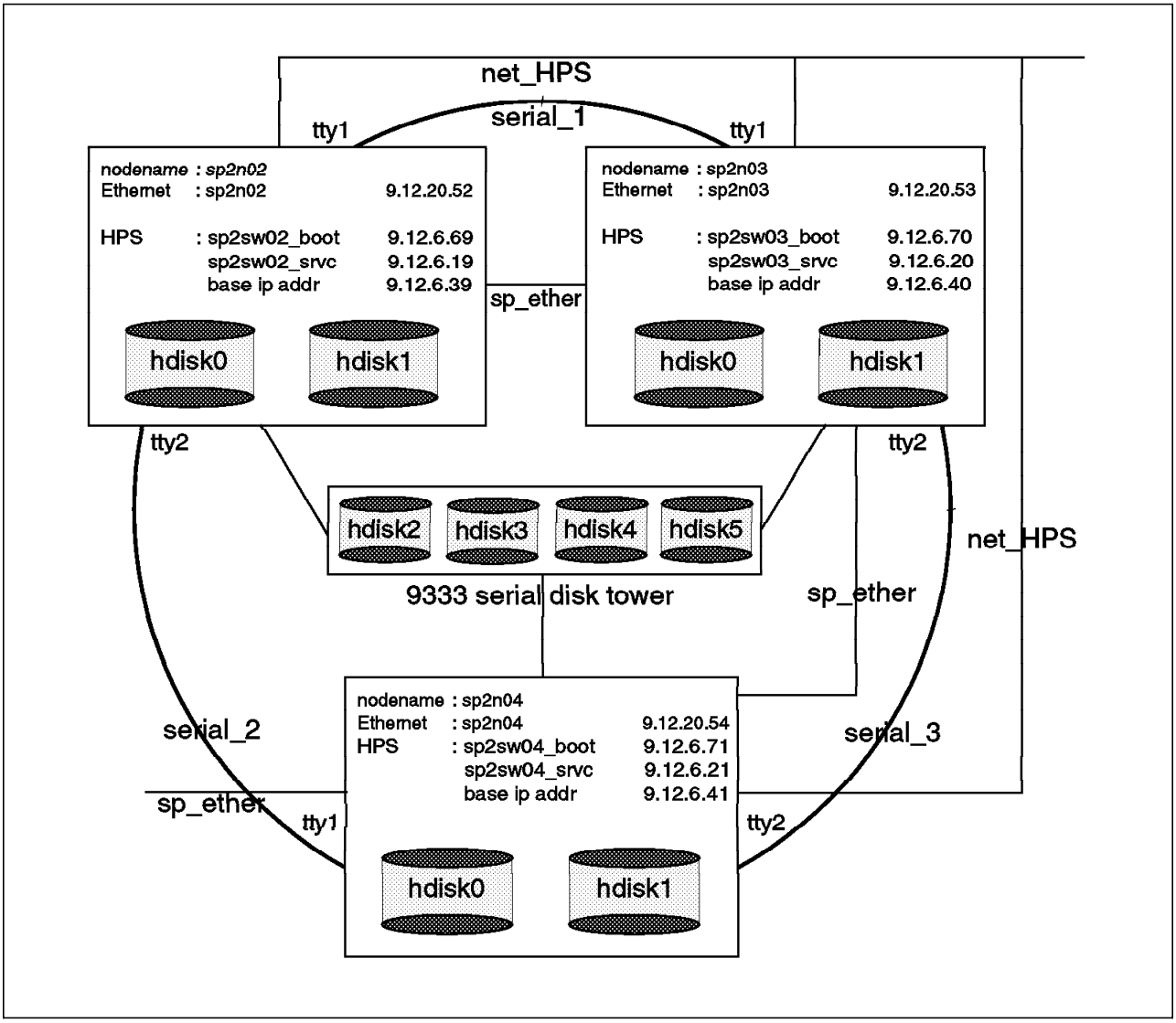


Figure 27. The System Configuration for HiPS & Eprimary Cluster

3.6.1 Configure HACMP to Manage Eprimary and HiPS Adapter Failures

The first step in providing a high availability solution for Eprimary and HiPS adapters is to set up a HACMP cluster. The procedures in the HACMP cluster setup are provided in section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68. However, for this section, the cluster configuration that we use is a three-node cluster, and it is configured in such way that all three HiPS service addresses are mutually backed up among all three nodes in a cascading take-over fashion.

In this cluster configuration, there are three types of networks.

- The High Performance Switch, which provides the high speed link between each node (sp2n02, sp2n03, sp2n04) within the SP system.
- The Serial (RS232) link which provides a non-tcpip communication only between the nodes within the cluster.
- The SP Administrative Ethernet, and this network should not be taking part in any IP address takeover scheme (see the section 3.4, “Planning

Considerations for HACMP on the RISC/6000 SP System" on page 66 for more information).

Refer to Figure 27 on page 87 for system configuration of the cluster, and refer to Figure 28 on page 89 through Figure 31 on page 92 for the network topology for this particular cluster as well. For more details on HACMP installation procedures, refer to the *HACMP Installation Guide*, SC23-2769.

```
Cluster Description of Cluster cluster_8
Cluster ID: 8
There were 5 networks defined : net_HPS, serial_1, serial_2, serial_3, sp_ether
There are 3 nodes in this cluster.

NODE sp2n02:
        This node has 4 service interface(s):

        Service Interface sp2sw02_srvc:
                IP address:     9.12.23.19
                Hardware Address:
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw02_srvc has a possible boot configuration:
                Boot (Alternate Service) Interface: sp2sw02_boot
                IP address:     9.12.23.69
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw02_srvc has no standby interfaces.


        Service Interface sp2n02_tty1:
                IP address:     /dev/tty1
                Hardware Address:
                Network:        serial_1
                Attribute:      serial

        Service Interface sp2n02_tty1 has no standby interfaces.


        Service Interface sp2n02_tty2:
                IP address:     /dev/tty2
                Hardware Address:
                Network:        serial_2
                Attribute:      serial

        Service Interface sp2n02_tty2 has no standby interfaces.


        Service Interface sp2n02:
                IP address:     9.12.20.52
                Hardware Address:
                Network:        sp_ether
                Attribute:      public

        Service Interface sp2n02 has no standby interfaces.
```

Figure 28. Cluster Topology for the 3-Node Cluster (1 of 4)

```
NODE sp2n03:
        This node has 4 service interface(s):

        Service Interface sp2sw03_srvc:
                IP address:     9.12.23.20
                Hardware Address:
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw03_srvc has a possible boot configuration:
                Boot (Alternate Service) Interface: sp2sw03_boot
                IP address:     9.12.23.70
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw03_srvc has no standby interfaces.


        Service Interface sp2n03_tty1:
                IP address:     /dev/tty1
                Hardware Address:
                Network:        serial_1
                Attribute:      serial

        Service Interface sp2n03_tty1 has no standby interfaces.


        Service Interface sp2n03_tty2:
                IP address:     /dev/tty2
                Hardware Address:
                Network:        serial_3
                Attribute:      serial

        Service Interface sp2n03_tty2 has no standby interfaces.


        Service Interface sp2n03:
                IP address:     9.12.20.53
                Hardware Address:
                Network:        sp_ether
                Attribute:      public

        Service Interface sp2n03 has no standby interfaces.
```

Figure 29. Cluster Topology for the 3-Node Cluster (2 of 4)

```
NODE sp2n04:
        This node has 4 service interface(s):

        Service Interface sp2sw04_srvc:
                IP address:     9.12.23.21
                Hardware Address:
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw04_srvc has a possible boot configuration:
                Boot (Alternate Service) Interface: sp2sw04_boot
                IP address:     9.12.23.71
                Network:        net_HPS
                Attribute:      private

        Service Interface sp2sw04_srvc has no standby interfaces.


        Service Interface sp2n04_tty1:
                IP address:     /dev/tty1
                Hardware Address:
                Network:        serial_2
                Attribute:      serial

        Service Interface sp2n04_tty1 has no standby interfaces.


        Service Interface sp2n04_tty2:
                IP address:     /dev/tty2
                Hardware Address:
                Network:        serial_3
                Attribute:      serial

        Service Interface sp2n04_tty2 has no standby interfaces.


        Service Interface sp2n04:
                IP address:     9.12.20.54
                Hardware Address:
                Network:        sp_ether
                Attribute:      public

        Service Interface sp2n04 has no standby interfaces.
```

Figure 30. Cluster Topology for the 3-Node Cluster (3 of 4)

```
Breakdown of network connections:

Connections to network net_HPS
        Node sp2n02 is connected to network net_HPS by these interfaces:
                sp2sw02_boot
                sp2sw02_srvc

        Node sp2n03 is connected to network net_HPS by these interfaces:
                sp2sw03_boot
                sp2sw03_srvc

        Node sp2n04 is connected to network net_HPS by these interfaces:
                sp2sw04_boot
                sp2sw04_srvc


Connections to network serial_1
        Node sp2n02 is connected to network serial_1 by these interfaces:
                sp2n02_tty1

        Node sp2n03 is connected to network serial_1 by these interfaces:
                sp2n03_tty1


Connections to network serial_2
        Node sp2n02 is connected to network serial_2 by these interfaces:
                sp2n02_tty2

        Node sp2n04 is connected to network serial_2 by these interfaces:
                sp2n04_tty1


Connections to network serial_3
        Node sp2n03 is connected to network serial_3 by these interfaces:
                sp2n03_tty2

        Node sp2n04 is connected to network serial_3 by these interfaces:
                sp2n04_tty2


Connections to network sp_ether
        Node sp2n02 is connected to network sp_ether by these interfaces:
                sp2n02

        Node sp2n03 is connected to network sp_ether by these interfaces:
                sp2n03

        Node sp2n04 is connected to network sp_ether by these interfaces:
                sp2n04
```

Figure 31. Cluster Topology for the 3-Node Cluster (4 of 4)

After all the components of the resources such as network and shared disks are configured into the HACMP environment, you can easily configure the HACMP cluster for Eprimary management by entering the following command on one of the cluster nodes at the command line. See the ATTENTION note before performing this step:

```
/usr/sbin/cluster/events/utils/cl_HPS_Eprimary manage
```

The command will automatically synchronize the HACMP ODM across all nodes within the cluster, and this cluster shall be the only cluster on an SP to manage Eprimary. If there is ever a need for the cluster to not manage Eprimary, you can issue the following command on one of the cluster node to unconfigure the HACMP cluster for Eprimary management:

```
/usr/sbin/cluster/events/utils/cl_HPS_Eprimary unmanage
```

Again, the command will automatically synchronize the HACMP ODM across all nodes within the cluster.

ATTENTION:

If there is more than one HACMP cluster defined in the SP, only one cluster can be configured to manage Eprimary. Therefore, before configuring an HACMP cluster for Eprimary management, you need to make sure that NO other cluster is already managing Eprimary. To check this, you can enter the following command on the control workstation:

```
dsh -a "odmget -q'name=EPRIMARY' HACMPsp2"
```

If the output contains nothing, or responds with something similar to the following for all nodes, then there is no other cluster currently configured to manage Eprimary:

```
HACMPsp2:
            name = "EPRIMARY"
            value  = ""
```

Notes:

1. Only one HACMP cluster on a RISC/6000 SP can manage Eprimary.
2. The first node that is brought up in an Eprimary managing cluster will check to see if another node is currently Eprimary. If another node is the Eprimary node, and that node (switch) is up, its switch device will be unloaded prior to the Eprimary reassignment (requiring it to be rebooted to reload these devices).
3. A graceful shutdown (shutdown -F) of an Eprimary node will remove that node's switch device for graceful failover of Eprimary to another node in the cluster. The node will have to be rebooted to reintegrate it into the cluster. Taking down the node will not unload the devices.

Due to the limitation on the number of adapters allowed on each node, one must consider the scenario that if the High Performance Switch adapter on the Eprimary node fails, the whole High Performance Switch initialization and recovery process control also goes down with it. Therefore, it is a single point of failure not only to the node, but also to the Eprimary. To eliminate this as a cluster single point of failure, one will need to use the AIX error notification configuration in HACMP to promote adapter failure to node failure, as illustrated in the following screen.

Errors that should be provided to AIX error notification are HPS_FAULT9_ER and HPS_FAULT3_RE. To perform the notification set up, enter:

```
# smit hacmp
```

Select the following from the HACMP system management menu:

= > **RAS Support**
= > **Error Notification**
= > **Add a Notify Method**

On the *Add a Notify Method screen*, create the error notification for HPS_FAULT9_ER by entering the following information:

```
                          Add a Notify Method

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

                                                  [Entry Fields]
* Notification Object Name                         HPS_ER9
* Persistence across system restart?               Yes             +
  Process ID for use by Notify Method              []             +#
  Select Error Class                               All             +
  Select Error Type                                PERM            +
  Match ALERTable errors?                          None            +
  Select Error Label                               [HPS_FAULT9_ER] +
  Resource Name                                    [All]          +
  Resource Class                                   [All]          +
  Resource Type                                    [All]          +
* Notify Method                          [/usr/sbin/cluster/utilitie
                                          /clstop '-grsy'>
```

Note: In the Notify Method field, we simply use the HACMP's stop cluster script with take over option to promote the failover of resources to the backup node. However, you could customize your own script to take whatever actions are necessary for your environment.

Create the error notification for HPS_FAULT3_RE by entering the following information:

```
                        Add a Notify Method

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

                                                [Entry Fields]
* Notification Object Name                       HPS_ER3
* Persistence across system restart?             Yes             +
  Process ID for use by Notify Method            []             +#
  Select Error Class                             All             +
  Select Error Type                              TEMP            +
  Match ALERTable errors?                        None            +
  Select Error Label                             [HPS_FAULT3_RE] +
  Resource Name                                  [All]           +
  Resource Class                                 [All]           +
  Resource Type                                  [All]           +
* Notify Method                          [/usr/sbin/cluster/utilitie
                                          /clstop '-grsy'>
```

Repeat the same procedures on all nodes within the cluster environment since, the error notification information will not be synchronized by HACMP.

In addition to the single point of failure at the High Performance Switch adapter, there are other single points of failure in our cluster example. They are the 9333 Serial Link adapter and the 9333 Disk Subsystem. To eliminate the 9333 Serial Link adapter as single point of failure, use the AIX error notification in HACMP to promote the adapter failure (SDA_ERR1, SDA_ERR3) to node failure. An example of the setup follows.

```
# smit hacmp
```

Select the following from the HACMP system management menu:

= > **RAS Support**
= > **Error Notification**
= > **Add a Notify Method**

On the *Add a Notify Method screen*, create the error notification for SDA_ERR1 by entering the following information:

```
                          Add a Notify Method

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

                                                      [Entry Fields&rbk.
* Notification Object Name                            9333_ADP_ER1
* Persistence across system restart?                  Yes              +
  Process ID for use by Notify Method                 []              +#
  Select Error Class                                  All              +
  Select Error Type                                   PERM             +
  Match ALERTable errors?                             None             +
  Select Error Label                                  [SDA_ERR1]       +
  Resource Name                                       [All]            +
  Resource Class                                      [All]            +
  Resource Type                                       [All]            +
* Notify Method                                       [halt -q]
```

Perform the same step for error SDA_ERR3, and then repeat the above procedures on all nodes within the cluster environment. Another method for eliminating the single point of failure for both the 9333 Serial Link adapter and 9333 Disk Subsystem is to have additional adapters placed on each node within the cluster and to have additional Disk Subsystems in the cluster (or use the RAID disk subsystem). Again, the notify method in the above example is by no means the best solution for every environment. You should tailor your notify method script to suit your environment.

3.6.2 Verify the High Availability Solution for Eprimary and HiPS Adapter Failure

To verify that the cluster is behaving as designed, we have performed the following set of tests: (Assumption: The current Eprimary node is sp2n02, and all nodes in the cluster are operational with HACMP, as shown in the Figure 32 on page 97).

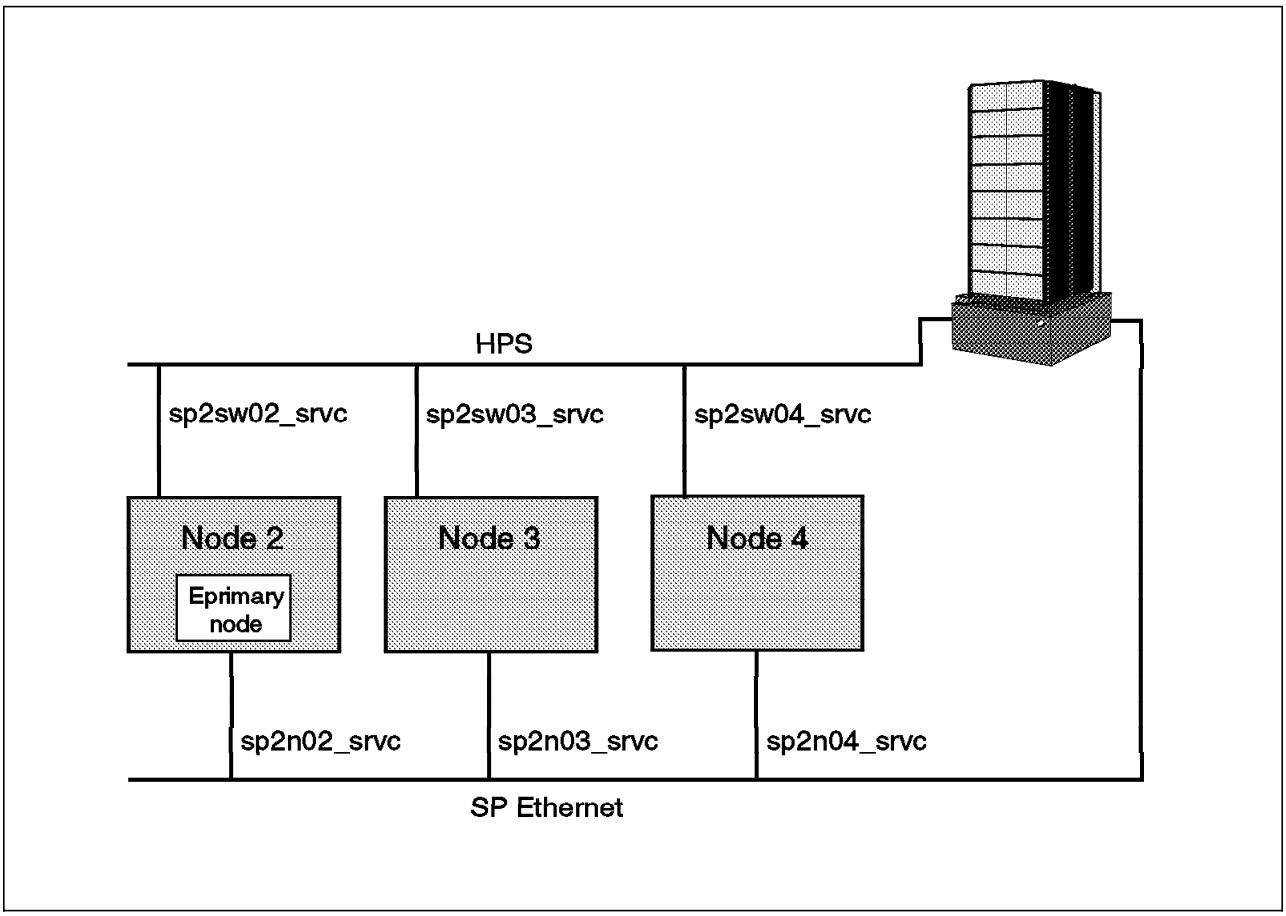


Figure 32. Normal Operation of the Running Cluster

1. Shutdown of HACMP with takeover option on node sp2n03, which is not the Eprimary node. Issue the following command:

   `smit clstop`

   The following screen will appear:

```
                              Stop Cluster Services

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

                                                      [Entry Fiels]
* Stop now, on system restart or both                 now           +

  BROADCAST cluster shutdown?                         true          +
* Shutdown mode                                       takeover      +
          (graceful, graceful with takeover, forced)



F1=Help             F2=Refresh          F3=Cancel       F4=List
Esc+5=Reset         F6=Command          F7=Edit         F8=Image
F9=Shell            F10=Exit            Enter=Do
```

   The behavior expected is the following:

- The service address of HiPS should be taken over by the backup node sp2n04, and the boot address should be reinstated.
- The Eprimary node should remain on sp2n02.
- The pictorial presentation of the expected behavior is shown in Figure 33.

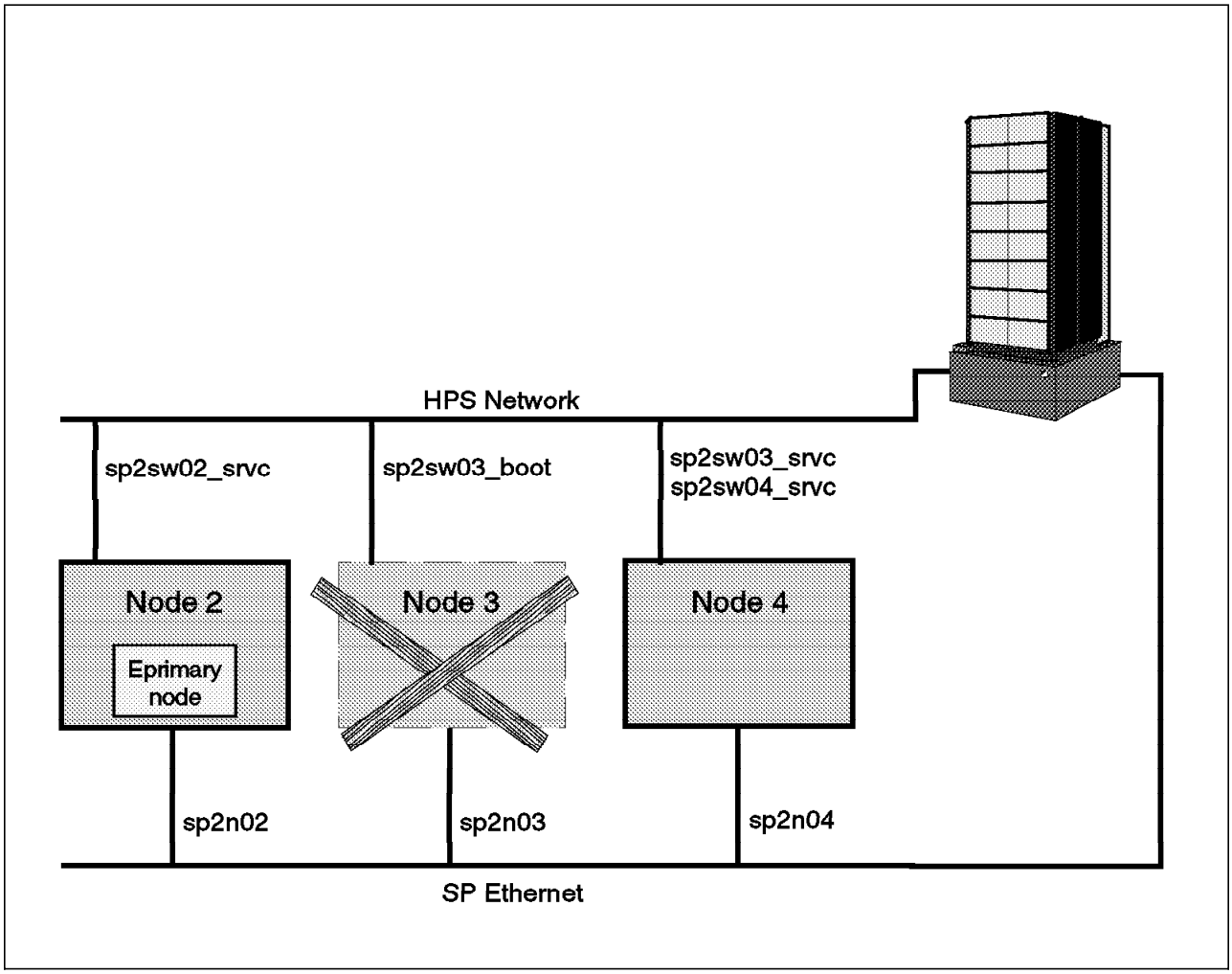


Figure 33. Expected Behavior of the Cluster after sp2n03 is Down

Method used for testing:

- Issue the command, `netstat -i`, on nodes sp2n03. We should see the following on the screen:

```
sp2n03-root / -> netstat -i

Name  Mtu   Network     Address      Ipkts  Ierrs    Opkts  Oerrs  Coll
lo0   16896 <Link>                   662      0      1060      0     0
lo0   16896 127         localhost    662      0      1060      0     0
en0   1500  <Link>10.0.5a.fa.1d.b8   34822    0      26409     0     0
en0   1500  9.12.20     sp2n03       34822    0      26409     0     0
css0  65520 <Link>0.0.0.0.0.0        22900    0      18998   294     0
css0  65520 9.12.6      sp2sw03      22900    0      18998   294     0
css0  65520 9.12.23     sp2sw03_boot 22900    0      18998   294     0
```

- On the takeover node sp2n04, we should see the following with the netstat -i:

```
sp2n04-root / -> netstat -i

Name  Mtu   Network     Address       Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 <Link>                    1415     0     1589     0     0
lo0   16896 127         localhost     1415     0     1589     0     0
en0   1500  <Link>10.0.5a.fa.12.1d    6008     0     5357     0     0
en0   1500  9.12.20     sp2n04        6008     0     5357     0     0
css0  65520 <Link>0.0.0.0.0.0         3763     0     2795     0     0
css0  65520 9.12.6      sp2sw04       3763     0     2795     0     0
css0  65520 9.12.23     sp2sw04_srvc  3763     0     2795     0     0
css0  65520 9.12.23     sp2sw03_srvc  3763     0     2795     0     0
```

- Issue the command, Eprimary, on any of the SP nodes. You should see the return number is 2, which corresponds to sp2n02.

```
sp2n06-root / -> Eprimary
2
```

2. Shutdown of HACMP with takeover option on the Eprimary node sp2n02. Do this by issuing the following command:

   `smit clstop`

   The following screen will appear:

```
                            Stop Cluster Services

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

                                                        [Entry Fields]
* Stop now, on system restart or both                   now           +

  BROADCAST cluster shutdown?                           true          +
* Shutdown mode                                        takeover      +
          (graceful, graceful with takeover, forced)




F1=Help             F2=Refresh          F3=Cancel       F4=List
Esc+5=Reset         F6=Command          F7=Edit         F8=Image
F9=Shell            F10=Exit            Enter=Do
```

   The behavior expected is the following:

   - The service address of HiPS and Eprimary should be taken over by the backup node, and since its first backup node sp2n03 is not available (shutdown in step 1), the resources will go to the next available node sp2n04.
   - The HiPS adapter of sp2n02 will become undefined and this node will have to be rebooted. The estart command has to be issued in order to reintegrate the node and its switch back into the cluster. (See the note

in section 3.6.1, “Configure HACMP to Manage Eprimary and HiPS Adapter Failures” on page 87).

- The pictorial presentation of the expected behavior is shown in Figure 34.

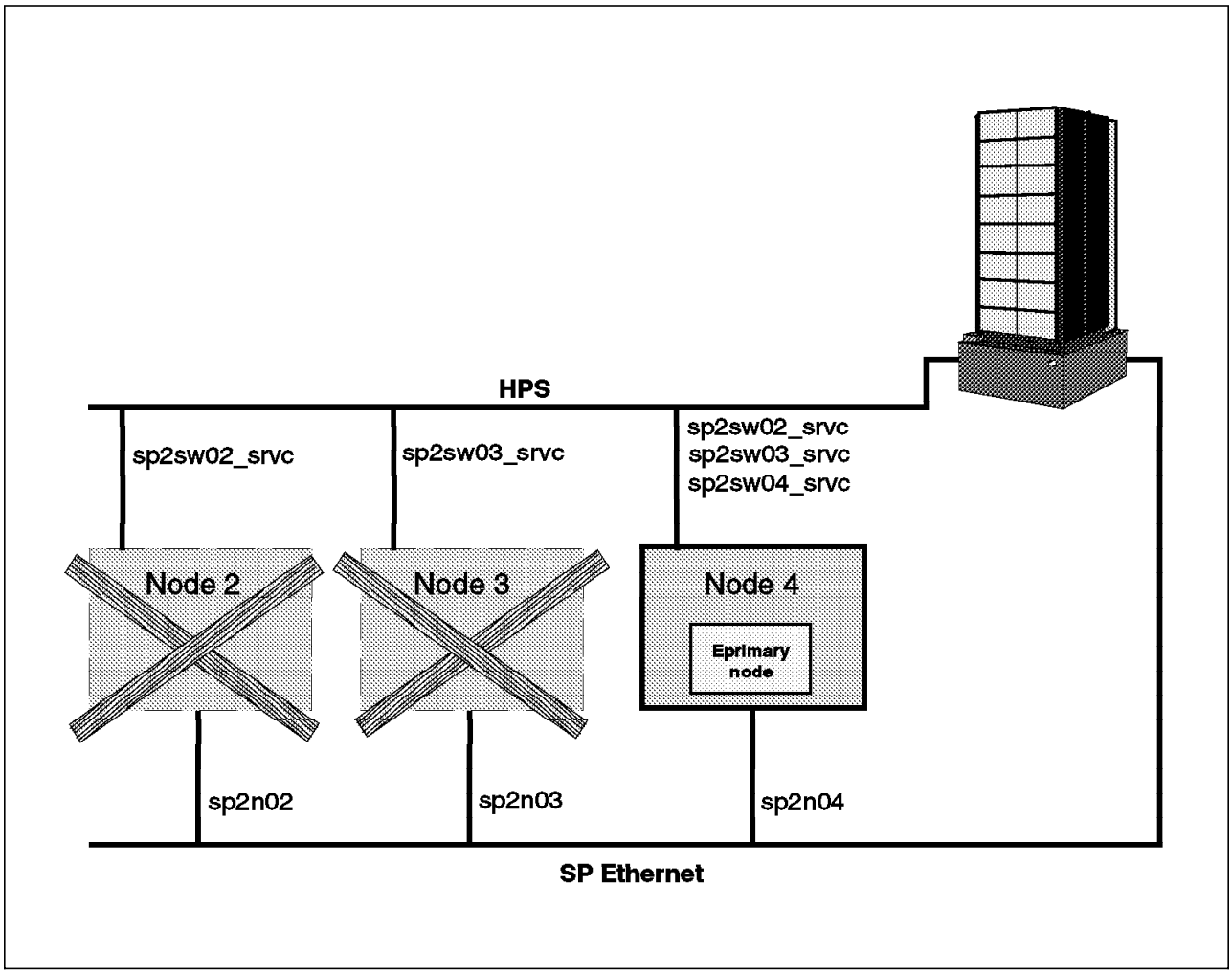


Figure 34. Results after sp2n03 and sp2n02 Have Failed

Method used for testing:

- Issue the command, `netstat -i`, on the backup node, and you should see that the service address of HiPS on the down node is being taken over, as seen on the following screen.

```
sp2n04-root / -> netstat -i

Name  Mtu   Network     Address      Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 <Link>                   1415     0     1589     0     0
lo0   16896 127         localhost    1415     0     1589     0     0
en0   1500  <Link>10.0.5a.fa.12.1d   6008     0     5357     0     0
en0   1500  9.12.20     sp2n04       6008     0     5357     0     0
css0  65520 <Link>0.0.0.0.0.0        3763     0     2795     0     0
css0  65520 9.12.6      sp2sw04      3763     0     2795     0     0
css0  65520 9.12.23     sp2sw04_srvc 3763     0     2795     0     0
css0  65520 9.12.23     sp2sw02_srvc 3763     0     2795     0     0
css0  65520 9.12.23     sp2sw03_srvc 3763     0     2795     0     0
```

- Issue the command Eprimary on any node or control workstation. You should see that the return number corresponds to the backup node.

```
sp2n06-root / -> Eprimary
4
```

3. Start HACMP on nodes sp2n02 and sp2n03. Remember, you will need to reboot sp2n02 before the node can rejoin the cluster.

   Expected behavior:

   - Node sp2n04 will release the service address of node sp2n02 and node sp2n03 back to its original owner.
   - The Eprimary node should still be sp2n04.
   - The pictorial presentation of the expected behavior is shown in the following Figure 35.

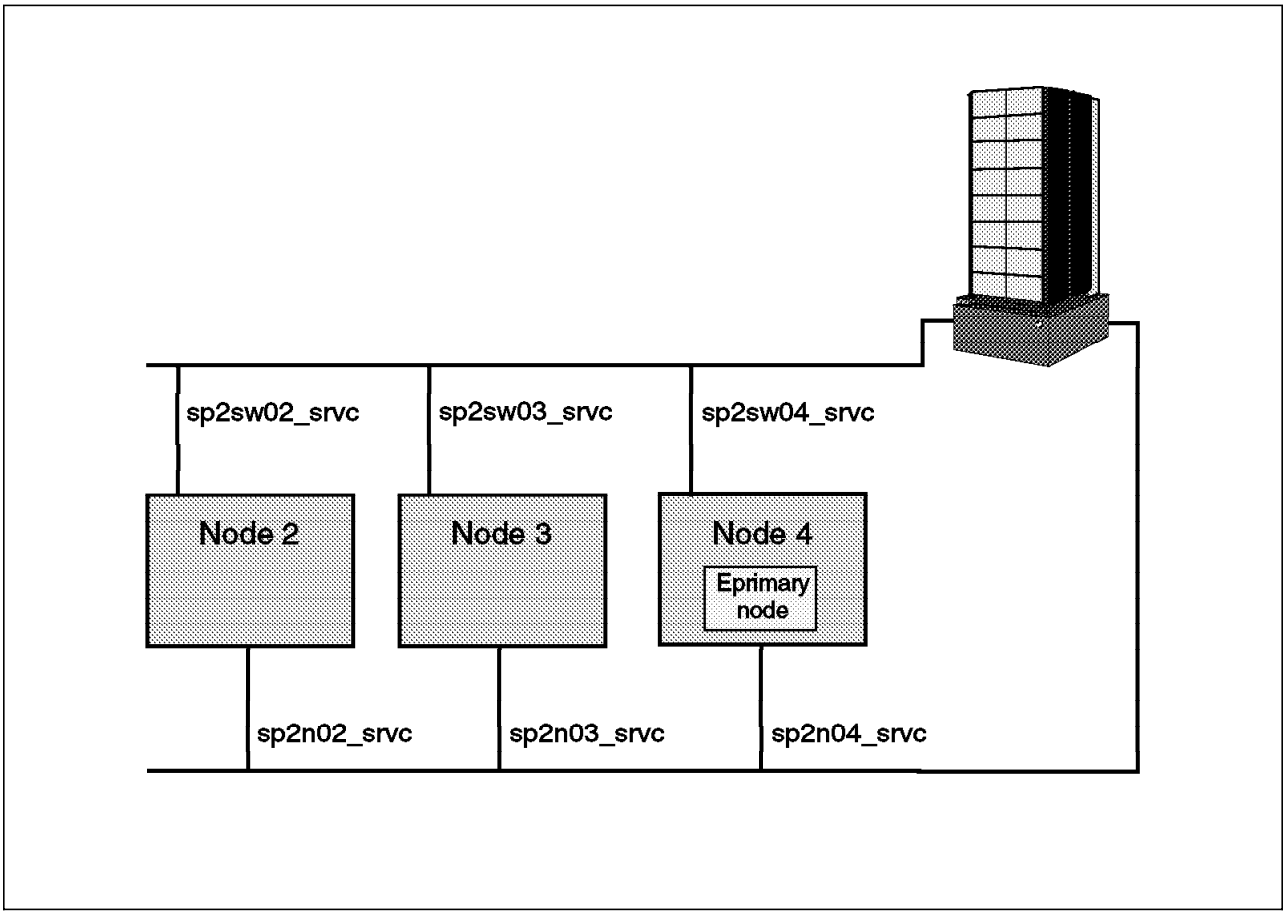


Figure 35. Results after Reintegration of Nodes sp2n03 and sp2n02

Method used for testing:

- Issue the command, netstat -i, on all three nodes. The service address should be reinstated to its respective node.

- Issue the command `Eprimary` at any node within the SP system, and the return value should be 4, which corresponds to node sp2n04. This is because the Eprimary node is configured as a rotating resource.

4. Repeat step 1 thru 3, except this time the machine is being powered down to simulate the hard crash of a node. Also, you may want to power down the machine in a different sequence than described above. You should, however, expect to see similar results.

5. Set the Eprimary node to a node that is outside of the cluster even before the cluster is brought up.

   Expected behavior:

   - The cluster will look at all RISC/6000 SP nodes to see if anyone is holding the Eprimary. If a node is not a member of the Eprimary Management Cluster, HACMP will take the Eprimary from that node and will assign Eprimary to the first member node that comes up within the cluster.
   - The node that just lost Eprimary will have to reboot, and Estart will have to be issued in order for the switch of that node to work again.

   Method used for testing:

   - Take down HACMP on all cluster nodes. Assign the Eprimary to a node outside of the cluster by issuing the command,

     ```
     Eprimary 5
     ```

     to assign the Eprimary to node 5.
   - Start the HACMP on one of the cluster member nodes (for example, sp2n02). After the HACMP is stabilized on that node, issue the command, `Eprimary`, and the return value should be the number corresponding to node sp2n02; in this case, it is 2.

6. Shutdown of HACMP with the graceful option on one of the cluster nodes that is not the Eprimary node.

   Expected behavior:

   - The service address of HiPS should be released, but it is not taken over by the backup node. The boot address should be reinstated.
   - The Eprimary node should remain as it is.

   Method used for testing:

   - Issue the command, `netstat -i`, on the nodes. You should see the boot address of the HiPS adapter.

7. Shutdown of HACMP with the graceful option on the Eprimary node.

   Expected behavior:

   - The Eprimary should be taken over by the backup node.
   - The HiPS adapter of the node will become undefined. This node will have to be rebooted, and Estart has to be issued in order for the node

and its switch to reintegrate itself back to the cluster. (See the note in section 3.6.1, “Configure HACMP to Manage Eprimary and HiPS Adapter Failures” on page 87).

- The service address will not be taken over by its backup node.

Method used for testing:

- Issue the command, `netstat -i`, on all cluster member nodes, and you should not see the node’s service address of HiPS appear on any of the nodes.
- Issue the command, `Eprimary`, on any node or control workstation. You should see that the return number is associated with the backup node.

8. Simulate the error HPS_FAULT9_ER for the HiPS adapter.

   Expected behavior:

   - HACMP will trap the error, and promote it to a shutdown of HACMP with takeover option.
   - The node will shutdown HACMP, and its resource (HiPS service address and/or Eprimary) will be taken over by its backup node.

   Method used for testing:

   - We found one of many tools used to simulate the error in the redbook, *HACMP Cook Book*, SG24-4553. The tools were developed for use with HACMP 2.1 or HACMP 3.1. However, this particular tool also works for HACMP for AIX 4.1.1. If the tools are installed properly, then you could generate the error by issuing the following:

     ```
     /usr/HACMP_ANSS/tools/ERROR_TOOL/error_gen
     28DB7CA6 HP_FAULT9_ER
     ```

9. Simulate the error HPS_FAULT3_ER for the HiPS adapter.

   Expected behavior:

   - HACMP will trap the error, and promote it to a shutdown of HACMP with takeover option.
   - The node will shut down HACMP, and its resource will be taken over by its backup node.

   Method used for testing:

   - We found one of many tools used to simulate the error in the redbook, *HACMP Cook Book*, SG24-4553. The tools were developed for use with HACMP 2.1 or HACMP 3.1. However, this particular tool also works for HACMP for AIX 4.1.1. If the tools are installed properly, then you could generate the error by issuing the following:

     ```
     /usr/HACMP_ANSS/tools/ERROR_TOOL/error_gen
     F066D4C0 HP_FAULT3_RE
     ```

10. Check client’s node communication.

    Expected behavior:

- After the IP takeover of a HiPS, client nodes should be notified, and some actions should be taken, such as updating the ARP cache. If HACMP client software is installed and configured properly, the default action would at least perform the update of the ARP cache. However, you can add new functionality to the clinfo.rc script in order to customize it to your requirement. For more details on the topic, refer to 3.7.5, “Client Considerations” on page 127 of this chapter.

Method used for testing:

- Use the `ping` command to test the connection between the client node and the cluster nodes. If you have customized the script for your environment, then you should define the test scenario to make sure the client environment is behaving in the same way as you would expect.

3.6.3 A Few Useful Tips

Here are some tips that could be useful in debugging problems during the set up and testing of the HACMP cluster for Eprimary and High Performance Switch adapter:

- The startup of HACMP cluster fails with some error message saying the HiPS address is not in its boot address.

  This problem is fixed in the Efix number 189568. HACMP is supposed to check if the HiPS adapter is on its boot address. If not, it will configure the HiPS adapter with its boot address, and later in the startup process, change it back to its service address. Please check to make sure you have applied the fix for the problem.

- The startup of the HACMP cluster fails with an error message saying there is no HiPS boot or service interface.

  The problem could be that the node was being released as the Eprimary node, and it must be rebooted to reintegrate itself back to the cluster.

- The HiPS could not start on the node after HACMP comes up.

  The problem could be that the fault_service_Worm_RTG daemon is not started. Verify that the fault service daemon is running on the node by entering:

  ```
  ps -ef | grep Worm
  ```

  Check for a message similar to this:

  ```
  root 14028 1 0 18:34:55  - 0:01 /usr/lpp/ssp/css/fault_service_Worm
  _RTG -r 1 -s 100015 -p 2 -a TB2 -d SW
  ```

  If the deamon is not running, try to start the daemon by entering:

  ```
  /usr/lpp/ssp/css/rc.switch
  ```

  and issue:

  ```
  Estart
  ```

  If it is still not working, then a reboot of the node may be necessary.

3.7 Implementing High Availability for HiPS Network Failure

The aim of this section is to discuss the requirements, configuration and suggested implementation of a recovery procedure to cope with the global network failure of the High Performance Switch, which has been identified as a single point of failure. Refer to the section 3.1, "HACMP Solutions Matrix for Potential Single Points of Failure on RS/6000 SP" on page 62 which discusses the identified single points of failure on the RISC/6000 SP.

The optional High Performance Switch in the RISC/6000 SP has built in recovery and redundancy to cope with component failure. This makes it a reliable part of the RISC/6000 SP system. However, it can still be seen as a single point of failure even if the probability of a network failure event occurring is minimal. To give a complete picture of High Availability on the RISC/6000 SP, the topic of a global HiPS switch failure must be discussed.

The basic concept used for the recovery of the High Performance Switch in this discussion is the use of an external network to recover the workload of the switch. The HiPS utilizes IP address aliasing within HACMP, and therefore we can utilize this concept to swap the IP addresses from the switch to the external network in the event of a failure.

For a more in-depth discussion of the High Performance Switch and other networks such as the internal Ethernet, refer to Chapter 4, "Network Considerations" on page 137.

3.7.1 Planning Considerations

When planning a highly available RISC/6000 SP system to recover from HiPS network failure, the following must be considered.

- Software requirements
- Hardware requirements
- Client requirements
- Application requirements

3.7.1.1 Software Requirements

The software used to implement the dual networking implementation of recovery of the HiPS in this redbook includes those requirements described in the basic configuration chapter, in addition to the following:

POST_network_up event script

POST_network_down event script

correct_routes

hps_swap_adapter

The additional scripts that enhance the High Availability Cluster Multi-Processing code to provide the HiPS network takeover functionality will be discussed later in this chapter.

3.7.1.2 Hardware Requirements

For the implementation of the HiPS network failure recovery, the following hardware was used:

- 2 RISC/6000 SP Thin nodes
- 2 HiPS TB2 Communications adapter (css0)
- 2 FDDI Adapters (fddi0)
- 2 Serial Link Disk Adapter (serdasda0)
- 2 Serial Link Eight Port Controller
- 2 8-Port Asynchronous Adapter EIA-232

Implementation of this HiPS dual networking requires that there is at least one external network adapter in the nodes for which High Availability Cluster Multi-Processing will be providing HiPS network failure recovery. For example, if you are to use FDDI for the backup network, then each node in the network should have an FDDI adapter defined to HACMP for this purpose.

With this in mind, the issue of adapter slots is an important one. Wide nodes are not as limited on slot availability as thin nodes. With a thin node there are only four adapter slots and one is already occupied by the switch adapter. This can be a limiting factor. With the apparent insufficient number of slots such as this, the non-IP serial network may not be used. The serial network is not required by HACMP, but is strongly recommended to deal with the scenario of TCP/IP as a single point of failure. This can be implemented through rs232 serial connection or Target mode SCSI. Owing to the fact that there are no serial ports on RISC/6000 SP nodes, there may be a requirement for serial adapter cards (for example, an 8-port asynchronous adapter may be needed to implement the serial network). If the configuration contains 7135 or SCSI, then tmscsi can provide the serial network, in conjunction with I/O pacing.

3.7.1.3 Other Considerations

The implementation of dual networking moves an IP address from one adapter to another so that it will now communicate over a different interface. In the event of such a failure, the applications running over the css0 interface and the clients connecting through the css0 interface need to be aware of the failover and to initiate the correct procedures in order to re-establish connections and continue communicating. Planning considerations must be made to enable clients and applications to recognize the failure and take the required actions. Clients can utilize the High Availability Cluster Multi-Processing clinfo API to detect the cluster changes and implement the customized actions to re-establish connections or to notify the user of the changes to the network topology. This will be discussed in more detail later in section 3.7.5, “Client Considerations” on page 127.

Considerations also have to be made for the application workloads. In this implementation of global network failure, there is no application implemented. Consideration needs to be given as to what to do with each of the workloads on the adapters. In this implementation the workload on the backup network would be disregarded, since the IP address of the backup adapter will be swapped to the css0 interface that is down. You may want to notify the users of the failure, and shut down any applications.

3.7.2 Installing Customized HiPS Dual Networking Scripts

The scripts that have been provided to perform additional functionality are a mixture of post-event scripts for High Availability Cluster Multi-Processing.

All the scripts need to be located in the directory:

/usr/local/cluster/events

3.7.2.1 Installation Procedure

In order to set up the scripts so that they can be implemented into the High Availability Cluster Multi-Processing environment, the scripts must be copied from the diskette to a temporary directory in the following manner.

```
tar -xvf /dev/fd0
```

This will put a tar file containing all the directories and scripts into the correct directory. The tar file is called **dualnet_inst.tar** All the files in the tar image have absolute pathnames. Execute the following command to create the correct directories and install the scripts in the appropriate directories:

```
tar -xvf dualnet_inst.tar
```

The following files are contained in dualnet_inst.tar:

```
drwxrwxrwx 0 0    0 Apr 11 17:13:40 1996 /usr/local/cluster/
-rwxr-xr-x 0 0 5696 Apr 09 17:07:40 1996 /usr/local/cluster/install_dual
drwxr-xr-x 0 0    0 Apr 11 17:14:16 1996 /usr/local/cluster/tmp/
drwxr-xr-x 0 0    0 Apr 09 14:39:06 1996 /usr/local/cluster/events/
-rwxr--r-- 0 0 1958 Apr 09 14:47:41 1996 /usr/local/cluster/events/
    POST_network_down
-rwxr--r-- 0 0 2814 Apr 09 14:48:45 1996 /usr/local/cluster/events/
    POST_network_up
-rwxr--r-- 0 0 1785 Apr 09 17:41:44 1996 /usr/local/cluster/events/
    correct_routes
-rwxr--r-- 0 0 3750 Apr 09 14:39:06 1996 /usr/local/cluster/events/
    hps_swap_adapter
```

Once the files are in place, then the cluster can be configured to implement the dual network takeover. The following section 3.7.3, “Configuration of Dual Network with HiPS and High Availability Cluster Multi-Processing” on page 108, will describe how to configure High Availability Cluster Multi-Processing to implement the HiPS dual networking scripts to cope with global HiPS failure. An install script has also been developed to verify the configuration of the HiPS global network failure which has been implemented in this redbook. It is contained on the diskette along with the scripts, and will be located in the /usr/local/cluster directory. The install script, which is provided with the HiPS scripts, removes some of the tasks in the implementation procedure and verifies the setup and required software level. The procedure that is outlined in section 3.7.3, “Configuration of Dual Network with HiPS and High Availability Cluster Multi-Processing” on page 108, is the complete procedure and does not assume the use of the install script. If you do use the diskette, then some of the processes are executed by the install script. It also verifies that High Availability Cluster Multi-Processing is configured in the correct way for the HiPS to failover correctly.

The install script **install_dual** verifies and implements the following:

It checks to verify that the level of High Availability Cluster Multi-Processing software is installed on the node is 4.1.1.

It verifies the existence of the HiPS network in High Availability Cluster Multi-Processing configuration.

It verifies the existence of the backup network containing the string BACKUP within the name in the High Availability Cluster Multi-Processing configuration.

It verifies if the different High Availability Cluster Multi-Processing nodes are up and ready for remote copies.

It creates the directories on the High Availability Cluster Multi-Processing server nodes.

It copies all of the scripts to all of the server nodes.

It adds the appropriate additional event scripts to the standard High Availability Cluster Multi-Processing events on all High Availability Cluster Multi-Processing server nodes.

It deinstalls, if required, and synchronizes definitions across all nodes.

Important

These scripts are in no way supported and are given in an advisory capacity for customization of your HACMP environment. Implementation of the scripts and any problems arising from them are the responsibility of the implementer.

3.7.3 Configuration of Dual Network with HiPS and High Availability Cluster Multi-Processing

This chapter describes how to set up an High Availability Cluster Multi-Processing cluster to cope with global failure of the switch. It discusses the panels that need to be completed within High Availability Cluster Multi-Processing, and naming conventions that need to be adhered to, such as IP addresses configuring for the various interfaces and resources that need to be defined.

For a more detailed view on basic High Availability Cluster Multi-Processing configuration, refer to section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68 or the *HACMP Cook Book*, SG24-4553.

To set up the cluster topology you must perform the following steps:

- Define a cluster definition.
- Define nodes to the cluster.
- Define adapters to the cluster.
- Synchronize the cluster definitions to all nodes in the cluster.
- Verify that the cluster definitions are correct.

It is best to work only on one node when defining cluster definitions to avoid overwriting good definitions from unsynchronized nodes.

3.7.3.1 Defining Cluster Topology

This section will describe the procedures taken to set up the cluster for dual networking using SMIT (Systems Management Interface Tool) screens. Alternatively the cluster topology could be set up using the High Availability Cluster Multi-Processing AIX Visual Systems Management (VSM) utility (xhacmpm) to perform many of the configuration tasks.

Add Cluster Definition: To add a cluster definition use the SMIT menu **Add a Cluster Definition**. You can use the smit fastpath `smit cm_config_cluster.add`. The following screen will be displayed. Enter a unique cluster ID. This must be a positive integer.

```
                          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                                        [9]
* Cluster Name                                      [DUAL_NET]




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Define Nodes: Define the nodes that will be included in the cluster topology. Use the SMIT menu **Add Cluster Nodes** and press Enter. The SMIT fastpath is obtained by entering `smit cm_config_nodes.add`. The node names are entered in a horizontal list with a space character as a delimeter. Node names cannot be greater than 31 characters.

```
                              Add Cluster Nodes

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

                                                   [Entry Fields]
* Node Names                                      [sp2n07 sp2n08]




F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Add Adapters: Add the switch adapter, backup adapter, and serial interfaces using the **Add an Adapter** SMIT screen. Type `smit cm_config_adapters.add` for the smit fastpath. (In this case the backup adapter is FDDI and the serial network is configured as /dev/tty1 on both nodes sp2n07 and sp2n08.) The following SMIT screen is showing the configuration of the switch service interface on node sp2n07. A discussion of the other interface setups follows.

```
                                Add an Adapter

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

                                                    [Entry Fields]
* Adapter IP Label                                  [sp2sw07_srvc]
* Network Type                                      [hps]
* Network Name                                      [HPSdual]
* Network Attribute                                    private
* Adapter Function                                     service
  Adapter Identifier                                [9.12.23.24]
  Adapter Hardware Address                          []
  Node Name                                         [sp2n07]


F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Adapter IP Label

This is the name of the adapter which you have chosen to be defined to High Availability Cluster Multi-Processing. The naming conventions used in our scenario are <interface>_srvc for the service adapters and <interface>_boot for the boot adapters.

Network Type

This determines the type of network to which the adapter is connected. On the RISC/6000 SP, the internal Ethernet is defined as type *ether*, and the High Performance Switch is defined as type *HiPS*. These are both preinstalled network modules and can be listed with the List option in the SMIT menu.

Network Name

For High Availability Cluster Multi-Processing Version 4.1.1. on the RISC/6000 SP, the High Performance Switch network must be defined with the string *HiPS* somewhere in the name of the network, for example, HPS_net. This allows the scripts to identify it as the switch network. For the purpose of the scripts provided for implementation of recovery from Global Switch failure, the backup network must include the string BACKUP within the network name, for example, BACKUP_net. Once the network adapter has been entered once, it can then be selected from a SMIT list.

Network Attribute

This can be either *public*, *private*, or *serial*. The HiPS must be defined as a private network and the RISC/6000 SP internal Ethernet

must be defined as a public network. The rs232 serial connection is defined as a serial network.

Adapter Function

This is defined as either *service*, *standby*, or *boot*. The HiPS has a service and boot adapter defined for it, but no standby. The internal Ethernet is defined to HACMP with only a service address so that it can be monitored by HACMP for AIX. It cannot be used for IP address takeover, because some RISC/6000 SP software such as the SDR associates the en0 internal Ethernet adapter with a specific node.

Adapter Identifier

Enter the IP address associated with the adapter. If this is the serial network, then the device name is entered. In this example, this would be /dev/tty1.

Note: The IP addresses specified for the HiPS service and boot addresses are alias addresses on the css0 interface. They are separate from the base IP address (the initial IP address configured on the css0 adapter). The base IP address must not be configured for IP address takeover in HACMP for compatibility reasons with the PSSP software, which associates the base IP address with a specific node. This address cannot be moved with IP address takeover. The boot and service addresses must also be defined on a different subnet to the base IP address, otherwise High Availability Cluster Multi-Processing clverify utility will fail when it is executed.

Adapter Hardware address

This is optional and contains the hardware address of the adapter. It is required if you are implementing hardware address swapping. This is not being implemented in this scenario.

Node Name

This associates the address with a particular node. The only cases where they are not associated with a node is when the service address is shared as part of a rotating resource. This is not the case in the demonstrated example. All service and boot interfaces are associated with a node, and those associated with resources are defined as part of a cascading resource.

Important

If you define a service adapter without an associated node name, and the service adapter interface will be used on either the HiPS network or the backup network in the HiPS dual network failover, then the scripts that implement this will fail. They *must* have an associated node name.

3.7.3.2 Node Configuration

The following is an overview of the High Availability Cluster Multi-Processing cluster topology which was used in the discussed scenario. This output can be obtained for any cluster topology by using the SMIT fastpath `smitty cm_show_menu` and selecting the **Show cluster Topology** screen, or you could alternatively use the command line option:

```
/usr/sbin/cluster/utilities/cllscf
```

```
Cluster Description of Cluster DUAL_NET
Cluster ID: 9
There were 4 networks defined : BACKUP_net, HPSdual, SPether, Serial_net
There are 2 nodes in this cluster.


NODE sp2n07:
        This node has 4 service interface(s):

        Service Interface spfddi07_srvc:
                IP address:     9.12.22.77
                Hardware Address:
                Network:        BACKUP_net
                Attribute:      public

        Service Interface spfddi07_srvc has a possible boot configuratio
                Boot (Alternate Service) Interface: spfddi07_boot
                IP address:     9.12.22.57
                Network:        BACKUP_net
                Attribute:      public

        Service Interface spfddi07_srvc has no standby interfaces.
        Service Interface sp2sw07_srvc:
                IP address:     9.12.23.24
                Hardware Address:
                Network:        HPSdual
                Attribute:      private

        Service Interface sp2sw07_srvc has a possible boot configuration
               Boot (Alternate Service) Interface: sp2sw07_boot
               IP address:     9.12.23.74
               Network:        HPSdual
               Attribute:      private

       Service Interface sp2sw07_srvc has no standby interfaces.


       Service Interface sp2n07:
               IP address:     9.12.20.57
               Hardware Address:
               Network:        SPether
               Attribute:      public

        Service Interface sp2n07 has no standby interfaces.
```

Figure 36. HACMP and HiPS Dual Network Topology Configuration (1 of 3)

```
        Service Interface sp2n07_serial:
                IP address:     /dev/tty1
                Hardware Address:
                Network:        Serial_net
                Attribute:      serial

        Service Interface sp2n07_serial has no standby interfaces.

NODE sp2n08:
        This node has 4 service interface(s):

        Service Interface spfddi08_srvc:
                IP address:     9.12.22.78
                Hardware Address:
                Network:        BACKUP_net
                Attribute:      public

        Service Interface spfddi08_srvc has a possible boot configuratio
                Boot (Alternate Service) Interface: spfddi08_boot
                IP address:     9.12.22.58
                Network:        BACKUP_net
                Attribute:      public

        Service Interface spfddi08_srvc has no standby interfaces.

        Service Interface sp2sw08_srvc:
                IP address:     9.12.23.25
                Hardware Address:
                Network:        HPSdual
                Attribute:      private

        Service Interface sp2sw08_srvc has a possible boot configuration
                Boot (Alternate Service) Interface: sp2sw08_boot
                IP address:     9.12.23.75
                Network:        HPSdual
                Attribute:      private
        Service Interface sp2sw08_srvc has no standby interfaces.

        Service Interface sp2n08:
                IP address:     9.12.20.58
                Hardware Address:
                Network:        SPether
                Attribute:      public

        Service Interface sp2n08 has no standby interfaces.
```

Figure 37. HACMP and HiPS Dual Network Topology Configuration (2 of 3)

```
Service Interface sp2n08_serial:
        IP address:     /dev/tty1
        Hardware Address:
        Network:        Serial_net
        Attribute:      serial

Service Interface sp2n08_serial has no standby interfaces.
```

Figure 38. HACMP and HiPS Dual Network Topology Configuration (3 of 3)

3.7.3.3 Synchronize the Cluster Topology

Once the definitions have been entered for the topology, they must be copied to the other nodes so that the definitions across all nodes are the same. This can be done from the SMIT option **Synchronize Cluster Topology**. This will update the ODM on all nodes defined in the cluster. To get to this menu, use the SMIT fastpath, `smitty cm_cfg_top_menu`, select the option, and press Enter.

```
                                Cluster Topology

Move cursor to desired item and press Enter.

  Configure Cluster
  Configure Nodes
  Configure Adapters
  Configure Network Modules
  Show Cluster Topology
  Synchronize Cluster Topology









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

3.7.3.4 Execute clverify

To ensure that the cluster configuration is correct with regards to the rules governed by High Availability Cluster Multi-Processing, use the */usr/sbin/cluster/diag/clverify* utility. This can be executed from the command line or through the SMIT option **Cluster Verification**. Use the SMIT fastpath `smit clverify.dialog`. Select whether you want to verify either the cluster resources, topology, or both, and then press Enter.

Another option that is available, but only from the command line, is the clverify **topology check** option which verifies that the High Availability Cluster Multi-Processing cluster definitions on each node are the same. You can also use the **sync** option to synchronize all the cluster nodes with the local node's definition.

```
/usr/sbin/cluster/diag/clverify cluster topology check
/usr/sbin/cluster/diag/clverify cluster topology sync
```

Note: Clverify may fail in SMIT and will display the following messages.

```
Verifying Cluster Topology...

ERROR: Node sp2n07 is not on its boot address.
ERROR: Node sp2n08 is not on its boot address.
```

The reason for the error messages is because the HiPS adapters, at the present time, have not been aliased with the boot addresses. Therefore, when HACMP performs the checking, it will notice that the HiPS adapter is not on the proper address, and issue the messages. However, these messages can be ignored, since HACMP will actually place the alias boot address on the HiPS adapter if it could not find one, and it will then change the boot address to the alias service address all during the start up process. However, if you want to eliminate the messages, you could either manually assign the alias boot address onto the HiPS adapter, or set an entry in the /etc/rc.net to alias the boot address when the system restarts. Again, even though these messages can be ignored, scan the entire clverify output for other ERROR messages before proceeding.

3.7.3.5 Defining Cluster Resources

In order to add resources to the High Availability Cluster Multi-Processing cluster, the following actions should be executed.

- Define a resource group to the cluster.
- Associate resources with the resource group.
- Synchronize the resource definitions across all nodes in the cluster.

This section will describe what resources need to be configured to implement dual networking with HiPS.

Four cluster resource groups were defined in the two node cluster that was set up, one for each service adapter on the Switch, and one for each service adapter on the backup net (FDDI). These were added using the **Define Resource Groups** SMIT option. This can be obtained by using the SMIT fastpath `smit cm_add_grp`.

Resource groups defined for the switch must be defined as cascading to implement the HiPS recovery in this scenario.

The following resources were defined for the HiPS dual network failure:

dual_grp - for sp2sw07_srvc

dual_grp1 - for sp2sw08_srvc

fddi07_grp - for spfddi07_srvc

fddi08_grp - for spfddi08_srvc

```
                              Add a Resource Group

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

                                                    [Enter Fields]
* Resource Group Name                               [dual_grp]
* Node Relationship                                   cascading
* Participating Node Names                          [sp2n07 sp2n08]









F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Once the resource groups have been defined, then the resources can be defined. Each of the HiPS service addresses must be defined to a resource group to be manipulated by High Availability Cluster Multi-Processing. They are defined in the SMIT option **Configure Resources for a Resource Group**. This can be obtained by using the SMIT fastpath `smit cm_cfg_res.select`. Select the resource group to which you want to define resources and then press Enter. This will then present you with the screen below.

```
                    Configure Resources for a Resource Group

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

[TOP]                                   [Enter Fields]
  Resource Group Name                    dual1_grp
  Node Relationship                   cascading
  Participating Node Names                        sp2n08 sp2n07

  Service IP label                                [sp2sw08_srvc]
  Filesystems                                     []
  Filesystems to Export                           []
  Filesystems to NFS mount                        []
  Volume Groups                                   []
  Concurrent Volume groups                        []
  Raw Disk PVIDs                                  []
  Application Servers                             []
  Miscellaneous Data                              []

[MORE...3]

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Synchronize Cluster Resources: After all the resource information has been entered, use the SMIT option to synchronize cluster resources. This will update the ODM on all the nodes in the cluster. Use the SMIT fastpath `smit`

cm_cfg_res_menu and then select the **Synchronize cluster resource** option. Alternatively, you can enter the following command:

```
/usr/sbin/cluster/diag/clconfig -s -r
```

```
                          Cluster Configuration

Move cursor to desired item and press Enter.

  Cluster Topology
  Cluster Resources
  Cluster Snapshots
  Cluster Verification









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

Figure 39 on page 118 shows the High Availability Cluster Multi-Processing server nodes configuration for sp2n07 and sp2n08.

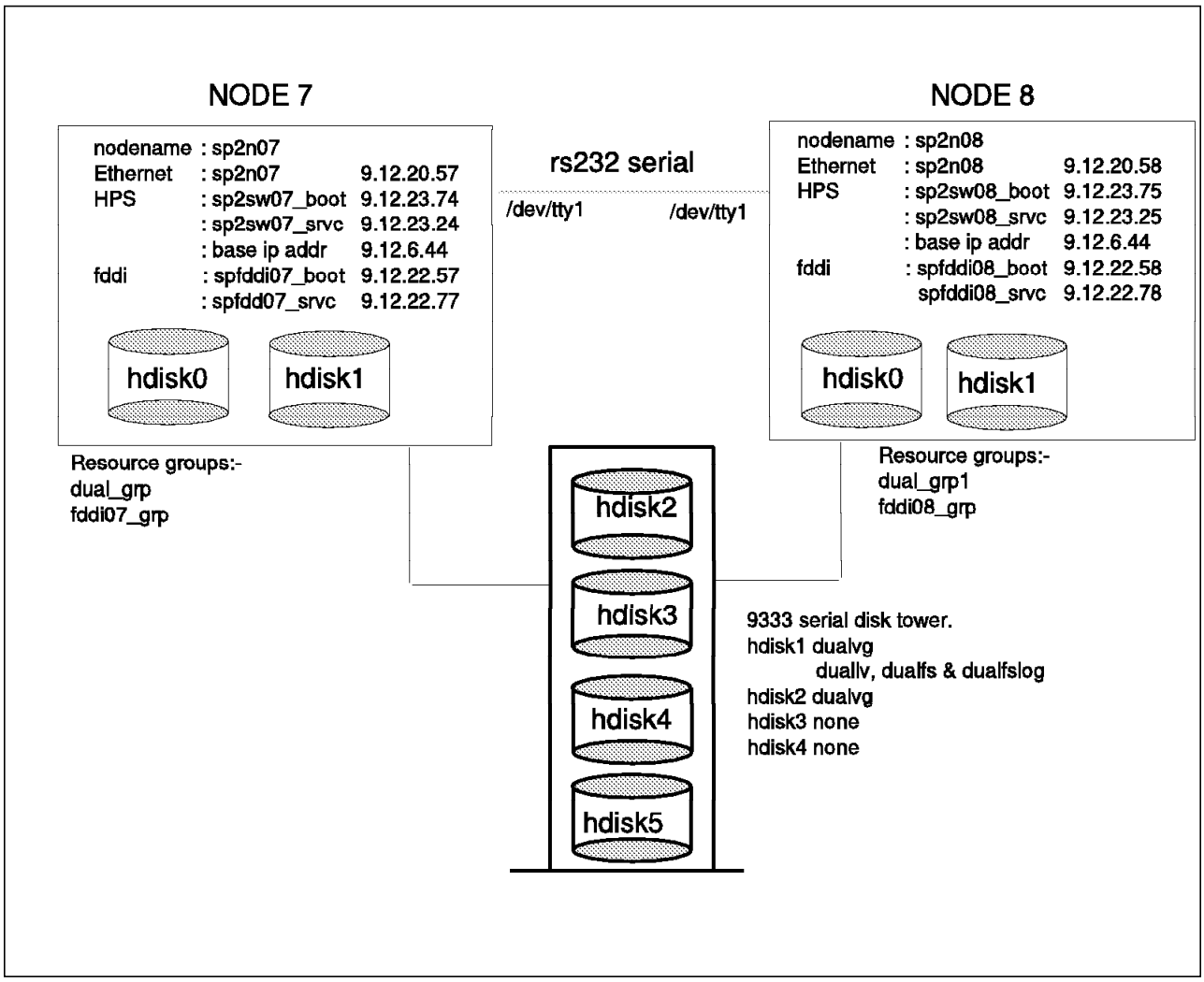


Figure 39. Setup of Cluster Servers for HiPS Dual Network Failure

3.7.3.6 Customizing Cluster Events

The scripts that have been created to enable the recovery from the global failure of the HiPS must be integrated into the High Availability Cluster Multi-Processing environment so that they are executed after the correct event. There are two events that need to have post-events added. The events are as follows:

Table 7. Post Events Added to High Availability Cluster Multi-Processing Event Scripts

High Availability Cluster Multi-Processing Event	Post event to be added
network_up	POST_network_up
network_down	POST_network_down

These scripts are executed with variables passed from the High Availability Cluster Multi-Processing scripts.

These events are added to High Availability Cluster Multi-Processing from the SMIT option **Change/Show Cluster Events**. The SMIT fastpath for this option is `smit clcsclev.select`.

```
                        Change/Show Cluster Events

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

                                            [Enter Fields]

  Event Name                                     network_down

  Description                                    Script run when a

* Event Command                               [/usr/sbin/cluster/]
  Notify Command                              []
  Pre-event Command                           []
  Post-event Command                          [/usr/local/cluster]
  Recovery Command                            []
* Recovery Counter                            [0]



F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The post events are added to each event by entering the full path name of the post event script into the Post-event Command field. The files will be in the path /usr/local/cluster/events. When Enter is pressed, the information is synchronized across all nodes in the cluster.

3.7.3.7 Other Configuration Issues

A working directory must be created for the scripts to implement the dual networking failover. Ensure that the directory has read and write permissions, and issue the following command:

```
mkdir /usr/local/cluster/tmp
chmod +w /usr/local/cluster/tmp
```

3.7.3.8 Starting High Availability Cluster Multi-Processing

Now that High Availability Cluster Multi-Processing has been configured, synchronized and verified across all nodes, it can be started. To start HACMP use the smit screen **Start Cluster Services**. You can get to this smit screen with the SMIT fastpath `smit clstart`. You will be presented with the following screen.

```
                              Start Cluster Services

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

                                                    [Enter Fields]

* Start now, on system restart or both                  now

  BROADCAST message at startup?                         true
  Startup Cluster Lock Services?                        true
  Startup Cluster Information Daemon?                   true

F1=Help             F2=Refresh          F3=Cancel           F4=List
Esc+5=Reset         F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

To verify that the cluster is started on the node use the following command:

```
lssrc -g cluster
```

The following output should be returned from the command:

```
sp2n07-root / -> lssrc -g cluster
Subsystem         Group            PID     Status
 clstrmgr         cluster          12148   active
 clsmuxpd         cluster          15776   active
 clinfo           cluster          18650   active
```

3.7.4 Implementation - What Actually Happens When the Switch Fails?

This section explains what has been implemented in the additional scripts to facilitate the failover of the High Performance Switch. With the basic concepts in mind, this could be tailored to meet specific requirements.

3.7.4.1 IP Address Aliasing

For IP addresses, the switch utilizes the IP address aliasing. Both the service and boot adapters are IP aliases created on the base IP address, which is the initial IP address defined on the css0 interface.

To create an alias on the switch interface, use the following command:

```
/usr/lpp/ssp/css/ifconfig css0 inet 9.12.23.24 netmask \
    255.255.255.0 alias up
```

Note: The netmask associated with the css0 base IP address will be used as the netmask value for all High Availability Cluster Multi-Processing for AIX HiPS network aliases. To delete an alias on the switch interface use the following command:

```
/usr/lpp/ssp/css/ifconfig css0 inet 9.12.23.24 delete
```

You can also bring an interface up and down using the following commands:

```
ifconfig <interface> up
ifconfig <interface> down
```

Important

Be careful not to use the standard ifconfig command for the css0 interface, especially when deleting aliases. The ifconfig command in /usr/lpp/ssp/css has been modified to handle the css0 interface. When deleting aliases with the standard command (/usr/sbin/ifconfig), all addresses for the css0 interface can be lost, including the base IP address.

3.7.4.2 Routes

Care must be exercised when manipulating IP addresses on network interfaces as a solution to the HiPS global network failure, because this may impact the routing tables that are defined. This could result in rendering some routes useless or missing so that communication across the manipulated interfaces is lost. In order to ensure that communication is maintained, the routing tables are adjusted using the route command.

In order to remove an incorrect route, enter:

```
route delete -net <network> <IP address>
```

for example:

```
route delete -net 9.12.22 9.12.22.57
```

In order to add a route, enter:

```
route add -net <network> <IP address>
```

for example:

```
route add -net 9.12.22 9.12.22.57
```

3.7.4.3 Scripts

The basic concept of the scripts is to provide extra functionality to the scripts that already exist with HACMP for IP address swapping and IP address aliasing to implement the recovery of the switch.

POST_network_down

This script determines whether it is actually the HiPS that has been involved in the network failure, and obtains the addresses of the HiPS and backup network adapters. It will then wait to ensure that it is not a temporary fault on the network, but a permanent network failure. It will then execute the hps_swap_adapter and correct_routes scripts (see below).

hps_swap_adapter

This script obtains the interfaces that will be involved in the swap. In this scenario it will be the HiPS and the FDDI interfaces. This script checks to see which way the swap is going. For example, has the switch just failed or has the interface come back up? In addition, the script will swaps the interfaces accordingly.

correct_routes

This script deals with the incorrect routing tables by deleting the incorrect routes. When swapping the routes between the HiPS and FDDI, attention had to be given to ensure that no spurious gateways existed in the routing tables. This is shown by the flags 'UG' in the output from the `netstat -rn` command. Communication was being impacted because of the bad route. This problem is dealt with in this script by removing the entry from the routing table and recreating it correctly.

POST_network_up

This script checks which interface is running, if there is a swap that has been initiated, and which interface the HiPS service IP alias is currently on.

Customization: In this redbook we have only looked at the scenario of the HiPS failing over to an external network, and we have basically disregarded the workload that existed on the backup network prior to the HiPS network failure. The scripts implemented are specific to this scenario. But this may not be the preferred action to take. Maybe the switch workload could be failed over to a standby adapter on an external network, or maybe the switch workload could co-exist with the external adapter workload (this could be extremely intensive and may be liable to hang the external network).

For any of the previously mentioned scenarios, the above scripts would need customization. For example, the scripts are specific about the type of adapter they will use for the takeover, in this case a service adapter. To use a standby adapter, the script would need to be modified to obtain the correct IP addresses. The High Availability Cluster Multi-Processing join_standby event would require attention to recognize the recovery of the css0 switch interface when it returns to the network, as at this point it would possess the IP address of the standby interface. This would initiate the IP address swapping between the two networks.

3.7.4.4 Verifying the HiPS Network Failover

In order to ensure that the HiPS network is failing over correctly you can execute several commands to monitor the network topology and ensure communication has been established.

Use the following commands:

Command	Function
netstat -i	Determines network attributes associated with interface
netstat -rn	Shows the current routes that are configured
ping	Verifies communication between interfaces

Figure 40 on page 123, Figure 41 on page 124, and Figure 42 on page 124 show the system environment, adapter interfaces, and routes before the global switch failure from the netstat command, respectively.

The Global switch failure was demonstrated by disconnecting one of the HiPS adapter connections from the back of one of the nodes.

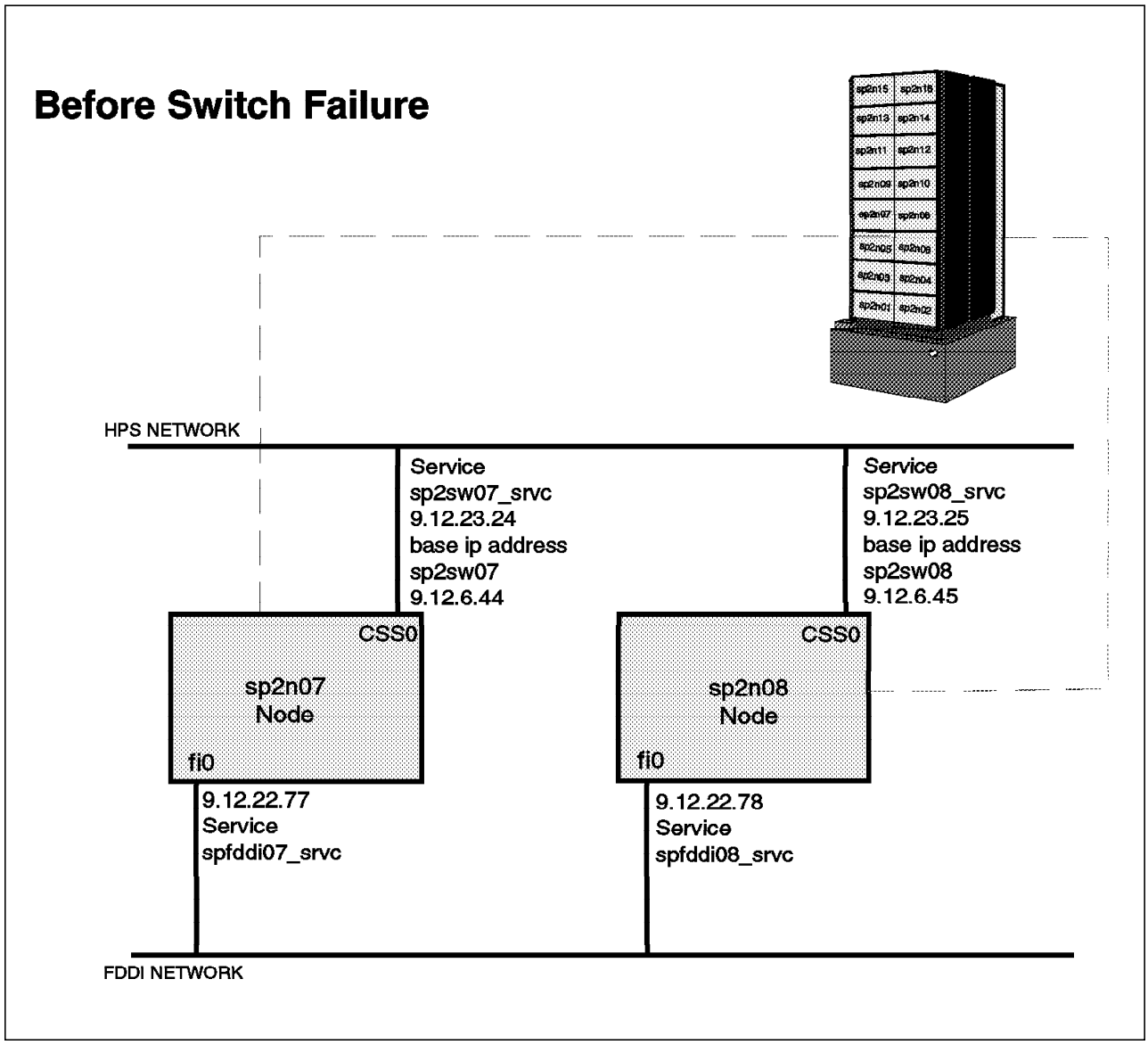


Figure 40. Network Topology before HiPS Network Failure

```
sp2n07-root / -> netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  C
lo0   16896 <Link>                        613049     0   650510     0
lo0   16896 127         loopback           613049     0   650510     0
en0   1500  <Link>10.0.5a.fa.13.af       1633245     0  1391531     0
en0   1500  9.12.20     sp2n07.itsc.pok  1633245     0  1391531     0
fi0   4352  <Link>10.0.5a.b8.2d.23       1025355     0   984629     0
fi0   4352  9.12.22     spfddi07_srvc    1025355     0   984629     0
css0  65520 <Link>0.0.0.0.0.0            1551480     0  1146465  2894
css0  65520 9.12.6      sp2sw07          1551480     0  1146465  2894
css0  65520 9.12.23     sp2sw07_srvc     1551480     0  1146465  2894
```

Figure 41. Adapter Interfaces before Global Switch Failure

```
sp2n07-root / -> netstat -rn
Routing tables
Destination      Gateway            Flags     Refs     Use  Interface
Netmasks:
255
255.255.255

Route Tree for Protocol Family 2:
default          9.12.20.37         UG           6    10784  en0
9.12.6           9.12.6.44          U            2      821  css0
9.12.20          9.12.20.57         U           16  1432998  en0
9.12.22          9.12.22.77         U            4    13945  fi0
9.12.23          9.12.23.24         U            3    15526  css0
127              127.0.0.1          U            9   454342  lo0
```

Figure 42. Routes before Global Switch Failure

Figure 43 on page 125, Figure 44 on page 125 and Figure 45 on page 126 show the system environment, adapter interfaces and routes after the global switch failure from the netstat command, respectively.

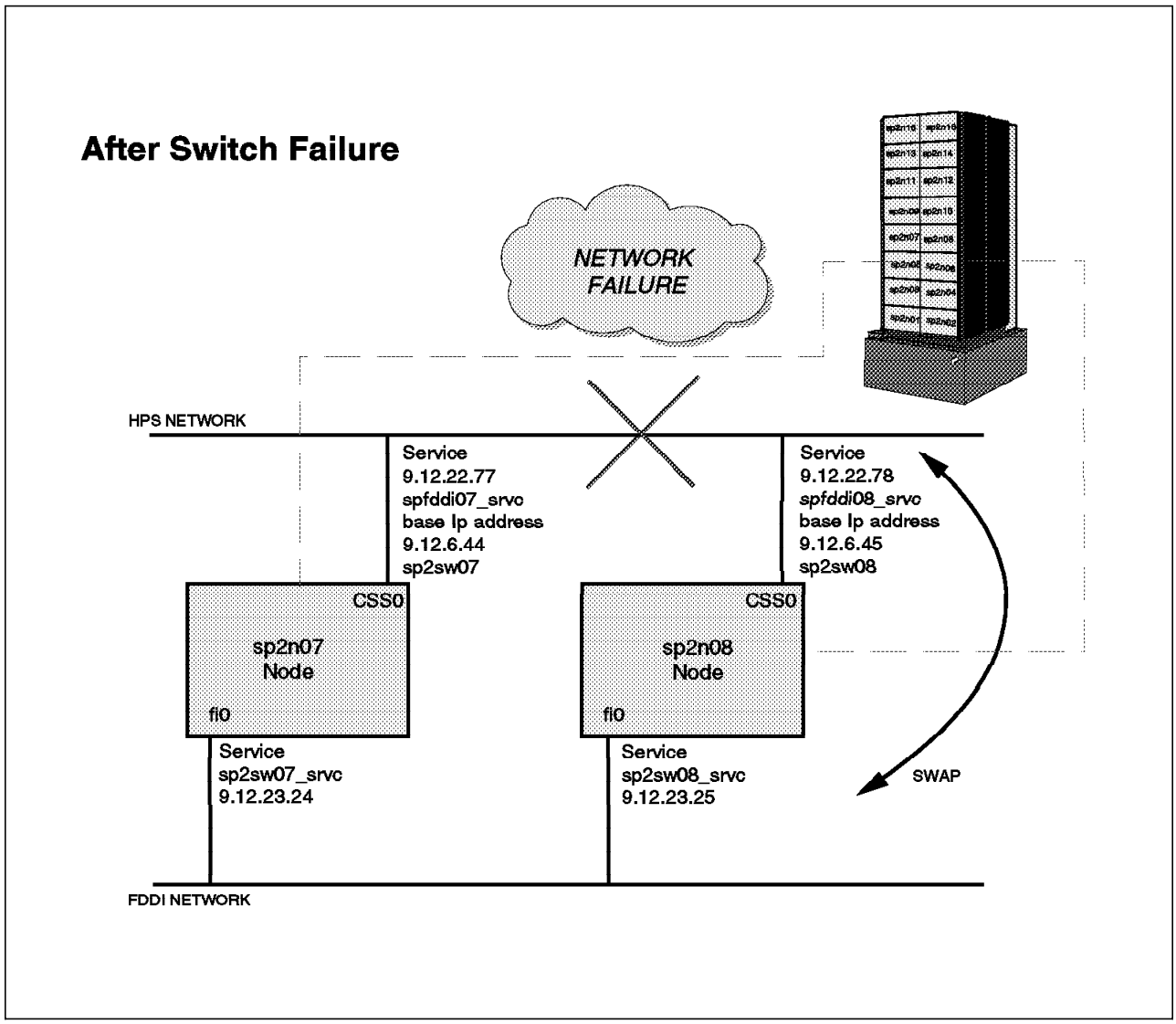


Figure 43. Network Topology after HiPS Network Failure

```
sp2n07-root / -> netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  C
lo0   16896 <Link>                        598696     0   635774     0
lo0   16896 127         loopback          598696     0   635774     0
en0   1500  <Link>10.0.5a.fa.13.af       1615211     0  1375824     0
en0   1500  9.12.20     sp2n07.itsc.pok   1615211     0  1375824     0
fi0   4352  <Link>10.0.5a.b8.2d.23       1015223     0   974876     0
fi0   4352  9.12.23     sp2sw07_srvc      1015223     0   974876     0
css0  65520 <Link>0.0.0.0.0.0            1535158     0  1137467  2848
css0  65520 9.12.6      sp2sw07           1535158     0  1137467  2848
css0  65520 9.12.22     spfddi07_srvc     1535158     0  1137467  2848
```

Figure 44. Adapter Interfaces after a Global Switch Failure

```
Routing tables
Destination      Gateway           Flags     Refs     Use  Interface
Netmasks:
255
255.255.255
Route Tree for Protocol Family 2:
default          9.12.20.37        UG           6    10837  en0
9.12.6           9.12.6.44         U            2      821  css0
9.12.20          9.12.20.57        U           16  1436809  en0
9.12.22          9.12.22.77        U            2      779  css0
9.12.23          9.12.23.24        U            2     1379  fi0
127              127.0.0.1         U            9   457407  lo0
```

Figure 45. Routes after the Global Switch Failure

Because there are only two High Availability Cluster Multi-Processing server nodes in the cluster, this constituted a network failure, because no communication could pass on the network. You could alternatively use the `ifconfig` command to bring the interface down, but this would result in a failure that would recover immediately, because the interface would be brought up again when the BACKUP adapter interface is swapped on to it.

Creating a network failure for more than a two node cluster is a little more complicated, as with generating any global network failure. With the HiPS switch, you can kill the worm of all the nodes simultaneously by using the `dsh` command. Or, more drastically, you can power off the switch.

The configuration was tested to see how tolerant it would be as a result of temporary switch faults. This was achieved by failing the Eprimary node, which is recovered by High Availability Cluster Multi-Processing in another scenario discussed in session 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86. The network_down event was generated by High Availability Cluster Multi-Processing, and the POST_network_down script was also called, but the IP addresses were not swapped over, because a network_up event was generated before the 2 minute grace period had elapsed. Therefore, the swap was abandoned and normal communications resumed.

The `ping` command can be executed to ensure that communication has been established on the backup interface for the service IP address of switch, in this case sp2sw07_srvc. To test the communications execute the following command from sp2n08:

```
ping sp2sw07_srvc
```

In our configuration after switch failure, this produced the following result:

```
sp2n08-root / -> ping sp2sw07_srvc
PING sp2sw07_srvc: (9.12.23.24): 56 data bytes
64 bytes from 9.12.23.24: icmp_seq=0 ttl=255 time=1 ms
64 bytes from 9.12.23.24: icmp_seq=1 ttl=255 time=1 ms
64 bytes from 9.12.23.24: icmp_seq=2 ttl=255 time=1 ms
64 bytes from 9.12.23.24: icmp_seq=3 ttl=255 time=1 ms
64 bytes from 9.12.23.24: icmp_seq=4 ttl=255 time=1 ms
```

The use of the ifconfig command at this point shows the following for the backup interface, the FDDI.

```
sp2n07-root / -> ifconfig fi0
fi0: flags=80a0843<UP,BROADCAST,RUNNING,SIMPLEX,ALLCAST,MULTICAST>
        inet 9.12.23.24 netmask 0xffffff00 broadcast 9.12.23.255
```

3.7.5 Client Considerations

In the event of a global switch failure, the client needs to be aware of the changes happening on the network in order to re-establish communications with the server. With the implementation of High Availability Cluster Multi-Processing, the clients can utilize the clinfo (Cluster Information Program) API or clinfo.rc file in order to implement this. Following is a discussion on the HACMP clinfo and how this could be customized to recognize the communication topology changes.

For a High Availability Cluster Multi-Processing client, the following filesets must be loaded on the client node to provide the utilities to monitor the cluster server nodes and network topology changes. These filesets contain among them the code for the clinfo and clstat utilities for use on the client.

cluster.base.lib
cluster.base.client.rte
cluster.base.client.utils
cluster.msg.en_US.client
cluster.adt.include
cluster.adt.clinet.demos
cluster.adt.client.samples.clinfo
cluster.adt.client.samples.clstat
cluster.adt.client.samples.demos
cluster.adt.samples.libcl
cluster.man.en_US.client.data

All clients accessing cluster services should be connected over a TCP/IP network and not a serial network. High Availability Cluster Multi-Processing does not detect failure of serial ports.

3.7.5.1 HACMP and Clients

High Availability Cluster Multi-Processing uses clinfo to provide information about the current state of the cluster. Clinfo is SNMP-based. Part of the SNMP protocol includes a Management Information Base (MIB) which contains data related to the network such as IP addresses and so on. The MIB database is manipulated by a standard SNMP agent, the snmpd daemon. Specific MIBs can be created for discrete environments by using the SNMP Multiplexing (SMUX)

protocol. A SMUX peer deamon (management agent) can manipulate information in the specialized MIB and can make it available.

An HACMP for AIX MIB database is provided by High Availability Cluster Multi-Processing which is maintained by the clsmuxpd daemon (management agent for HA). The information in the HACMP for AIX MIB can be read by clinfo through the clsmuxpd daemon.

Clinfo can run on the High Availability Cluster Multi-Processing server and client, and provides information regarding cluster topology changes. The clinfo clients and servers can then be customized to react to certain events. This enables the developer to use the /usr/sbin/cluster/etc/clinfo.rc file or its associated Application Programming Interface (API) to customize the clients to react to specific events in the cluster, such as adapter failure and IP address takeover. The default action taken executed by the clinfo.rc file is to flush the ARP cache to enable reconnection after adapter failover.

When the clsmuxpd daemon starts, it registers with snmpd and then constantly receives information from the cluster manager (clstrmgr). Clinfo normally only queries clsmuxpd for the status information every 15 seconds. This can be changed so that it is updated as soon as an event happens by initiating clinfo with the -a parameter.

Clinfo reads the /usr/sbin/cluster/etc/clhosts when it starts, and searches its contents for the IP address/IP label of the adapter in at least one node of the cluster that it wants to monitor. It will look for an active clsmuxpd for each IP address for the cluster, starting with the first until it finds one. If at any point the node that it is connected to through the clsmuxpd fails, then it will look for another active clsmuxpd on the other IP addresses in the clhosts file. Clinfo will not work if the clhosts file is not properly configured. By default this file will contain the loopback address. This needs to be edited and updated with IP addresses of the cluster servers.

Clinfo tracks the following events:

- Cluster state
- Cluster substate
- Failure of Service adapter and IP address swapping to standby
- Failure of Service adapter to standby completed
- Network failure
- Joining node
- Joined node
- Leaving node
- Left node

3.7.5.2 Client IP Addresses

Clients that wish to communicate with High Availability Cluster Multi-Processing cluster server nodes over the switch need to have alias IP addresses defined on their css0 interfaces. This is because the base IP address that they have defined is on a different subnet to that of the service interfaces of the server nodes. The `/usr/lpp/ssp/css/ifconfig` command can be used to generate an IP alias address. This is discussed in chapter 3.7.4.1, " IP Address Aliasing" on page 120.

If the command is executed from the command line, the alias will be lost when the machine is rebooted. To ensure that the alias is created on reboot, the /etc/rc.net file must be edited to include the ifconfig command for the IP address alias.

3.7.5.3 Updating Clients - Suggested Implementations

In order for the client to react to changes in the cluster that clinfo has provided to it, the developer can either update the clinfo.rc file or use the Clinfo Application Programming Interface to access the cluster status information. High Availability Cluster Multi-Processing provides Clinfo C API and Clinfo C++ API. More information can be obtained on the two APIs in *High Availability Cluster Multi-Processing 4.1 for AIX - Programming Client Applications*, SC23-2773-01.

With regards to updating the clinfo.rc file, standard shell programming can be used to capture the events that clinfo reports, and implement actions accordingly.

The following environment was set up with sp2n07 and sp2n08 as High Availability Cluster Multi-Processing servers and sp2n11 as a client. A global switch network failure was generated and the client was required to react to the change by swapping the IP address of its switch interface to the backup network - FDDI in order to continue to communicate with the High Availability Cluster Multi-Processing servers.

Figure 46 on page 130 shows the client/server network setup and cluster status information being propagated to the client.

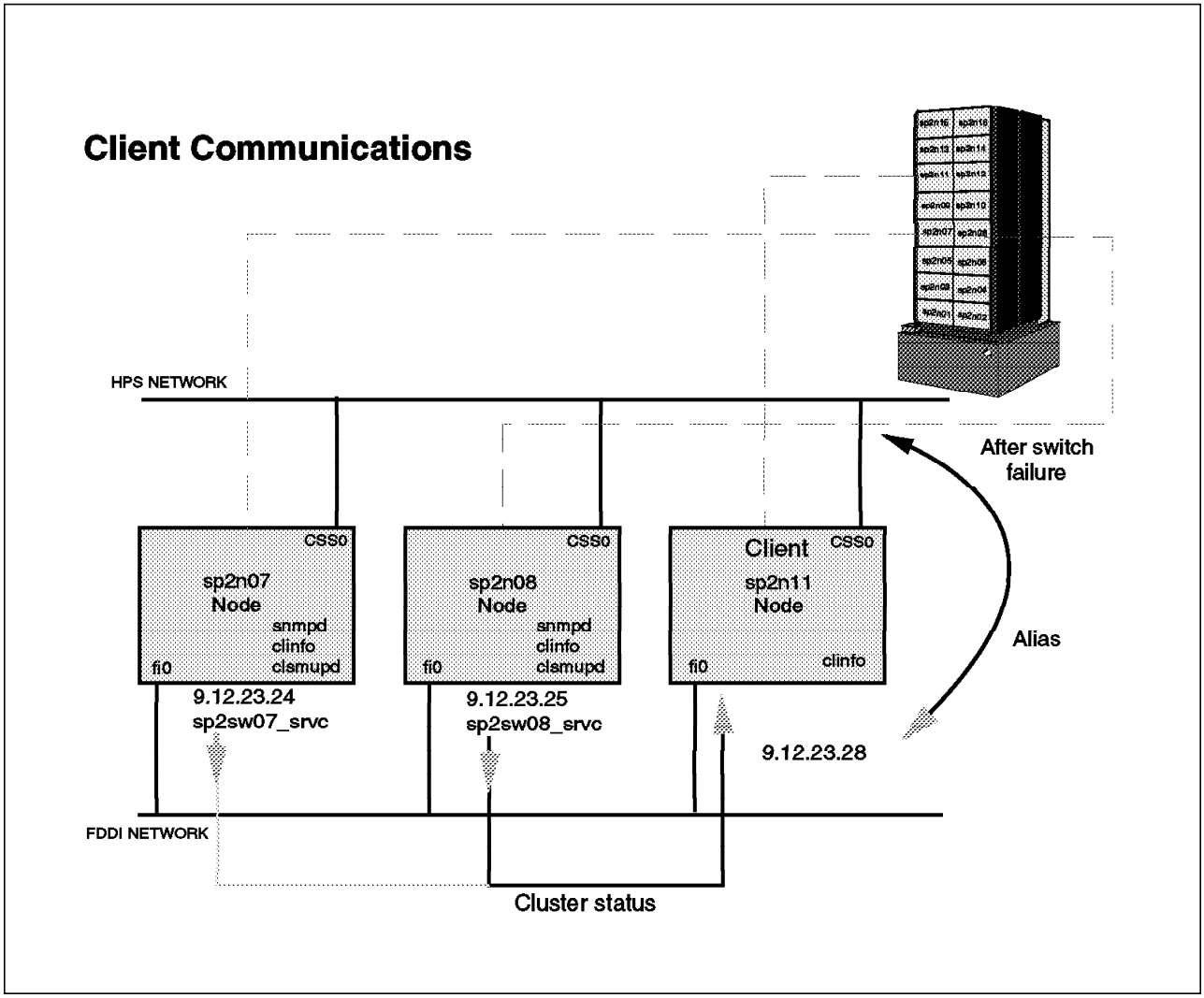


Figure 46. Client Server Communication of Cluster Status Information

For clinfo to continue to receive the cluster changes when the switch network goes down, another TCP/IP connection must exist for the clinfo daemons to communicate between server and client.

The following is intended to be an example of the client customization that can be performed. Every scenario will be different and will need customization for the specific environment in which the client is operating.

```
if [[ $EVENT = fail && $INTERFACE = sp2sw07_srvc ]]
then
        if [ -f /tmp/HPS_network_is_down ]
        then
                return 0
        fi
       touch /tmp/HPS_network_is_down
       /usr/lpp/ssp/css/ifconfig css0 inet sp2sw11_srvc delete
       sleep 1
       ifconfig fi0 inet sp2sw11_srvc netmask 255.255.255.0 alias up


fi
```

The above is intended to catch the failure of the sp2n07_srvc adapter interface and remove the IP address from the css0 (HiPS) interface and alias the IP address to the fi0 (FDDI) interface, so that the client can then communicate with the server nodes after the global failure of the switch. The same principle can be applied to deleting the interface from the backup adapter and aliasing it back onto the css0 interface. Be careful when writing the scripts, because several events of the same type can be received. For example, take the following scenario:

The HiPS network is returning after a global network failure. The css0 interface comes up, and generates a join event for the spfddi07_srvc interface, which currently exists on the css0 interface. IP addresses are then swapped and aliased to the appropriate interface, which will then create another join event for the spfddi07_srvc interface when it comes up on its original fi0 (FDDI) interface. If a simple ′if′ statement were included in the clinfo.rc file similar to the one above, then this code would be executed twice unless more checks are coded in.

3.7.5.4 Tips for Client Problems

When setting up a client to interact with the High Availability Cluster Multi-Processing cluster, ensure that the following have been completed.

/usr/sbin/cluster/etc/clhosts has been updated with IP addresses.

/etc/hosts contains the correct Node names and IP addresses.

Ensure clinfo is running.

Use the command:

```
lssrc -g cluster
```

Ensure that the subsystem exists and is active.

Ensure clsmuxpd and snmpd are running on the server.

Use the command:

```
lssrc -a | pg
```

Ensure that both subsystems exist and are active.

3.8 An Example of an Integrated Solution

As discussed in previous sections, High Availability Cluster Multi-Processing has been implemented to recover from several different single points of failure related to the HiPS switch. Each point of failure is discussed as a separate solution. The three points of failure identified for the HiPS are as follows:

- HiPS adapter
- Eprimary
- HiPS Global Network

For a complete solution to switch failure, these components can be combined in order to protect against all points of failure. This was implemented and tested by modifying the cluster scenario used in section 3.7, "Implementing High Availability for HiPS Network Failure" on page 105. The client node, sp2n11, was changed to a server node and set up in the same way as the original server nodes (node 7 and node 8). The cluster was then modified to deal with Eprimary and HiPS adapter failure using the procedures outlined in section 3.6, "Implementing High Availability for Eprimary and HiPS Adapter Failure" on page 86. The cluster was verified by running the `clverify` command.

Once the cluster had been set up, we simulated 3 possible failures to monitor the reaction of the cluster. The failures are as follows:

- HiPS adapter failure (not on Eprimary)
- HiPS adapter failure on Eprimary node
- Global Switch failure (implemented after an adapter failure)

HiPS adapter failure

In this scenario, node 7 was the Eprimary node and we simulated an HPS_FAULT9_ER error on the css0 adapter on node8. This had the effect of promoting the adapter failure to a node failure on node 8 as expected. The switch service address sp2sw08_srvc was swapped onto the css0 interface on node 7. This was checked by issuing `netstat -i` on node 7 as shown on the following screen.

```
sp2n07-root / -> netstat -i

Name  Mtu   Network     Address       Ipkts Ierrs    Opkts Oerrs  Coll
lo0   16896 <Link>                    1415     0     1589     0     0
lo0   16896 127         localhost     1415     0     1589     0     0
en0   1500  <Link>10.0.5a.fa.12.1d    6008     0     5357     0     0
en0   1500  9.12.20     sp2n07        6008     0     5357     0     0
css0  65520 <Link>0.0.0.0.0.0         3763     0     2795     0     0
css0  65520 9.12.6      sp2sw07       3763     0     2795     0     0
css0  65520 9.12.23     sp2sw07_srvc  3763     0     2795     0     0
css0  65520 9.12.23     sp2sw08_srvc  3763     0     2795     0     0
```

There were now two switch service addresses (sp2sw07_srvc and sp2sw08_srvc) aliased to the css0 interface on node 7, as shown in Figure 47 on page 133. We executed a `ping` from node 11 to the sp2sw08_srvc interface, and communication was successful. We ran Estart on the RISC/6000 SP and then started HACMP on node 8. The sp2sw08_srvc interface was then swapped back to node 8. A `ping` test confirmed successful communication to the css interface on node 8.

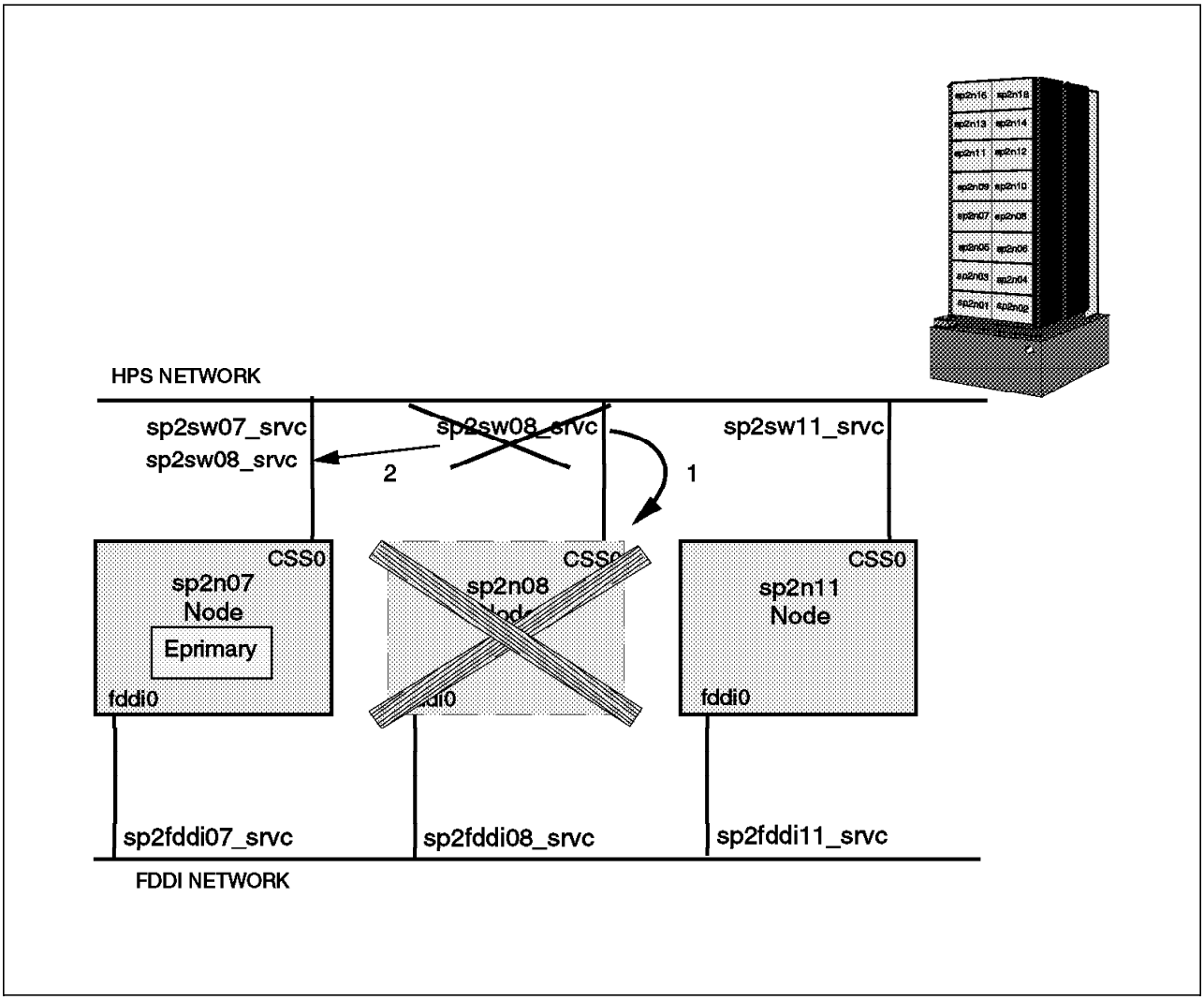


Figure 47. HiPS Adapter Failure

HiPS adapter failure on Eprimary node.

In this scenario, node 7 was the Eprimary node and we simulated an HPS_FAULT9_ER on node 7. This resulted in the adapter failure being promoted to a node failure as expected. The Eprimary node was failed over successfully and node 8 became the Eprimary node. The sp2sw07_srvc switch service interface was swapped and aliased on to node 8, so now both sp2sw07_srvc and sp2sw08_srvc service interfaces were on the css0 interface on node 8, as shown in Figure 48 on page 134. Communication to each interface was validated using the `ping` command and was successful for both service interfaces. The css0 interface on node 7 was unconfigured, as expected, in the Eprimary failover. To reintegrate node 7 back into the cluster, the node has to be rebooted. This is due to the unconfigured css0 interface and also because of the stale Eprimary Worm that exists on the node, as node 7 was the old Eprimary node.

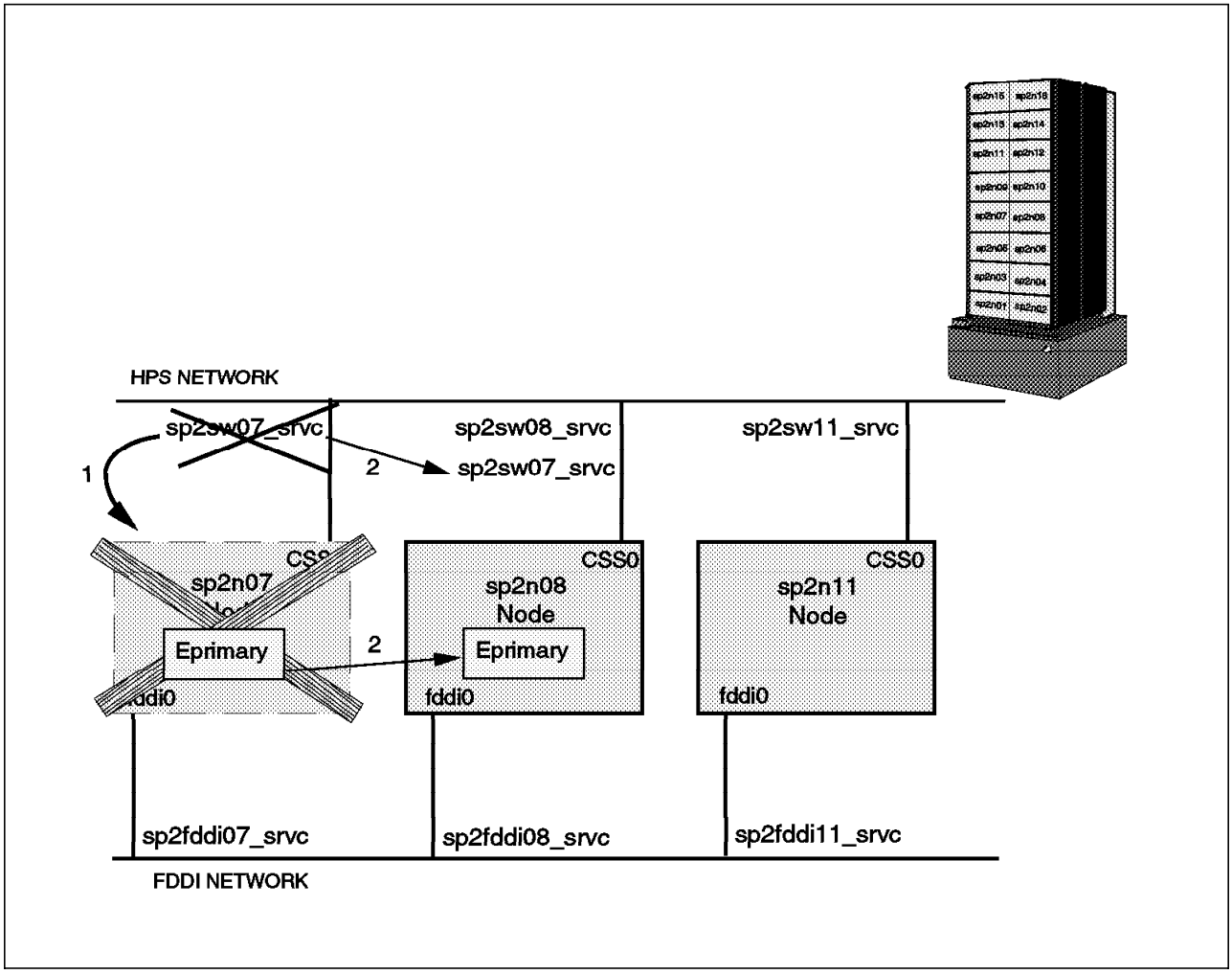


Figure 48. HiPS Adapter Failure on Eprimary Node

Global Network Failure

For this scenario, we used the cluster in the state it remained in after the HiPS adapter failure on the Eprimary node, before it was recovered. This means that sp2sw07_srvc and sp2sw08_srvc are both aliased on the css0 interface on node 8, and node 8 is the Eprimary. A global switch failure was then simulated by running off the switch at the base of the RISC/6000 SP system. The results observed were as follows: The sp2sw08_srvc address was swapped from the css0 interface on node 8, to the fi0 interface. The spfddi08_srvc service address was swapped from the fi0 interface to the css0 interface (which is down). The sp2sw07_srvc service address remained on the css0 interface; see Figure 49 on page 135. This was as expected, as this address is not the responsibility of node 8. The Eprimary remained as it was. The interfaces were then tested successfully with the `ping` command. The switch interfaces, for example sp2sw07_srvc, could communicate, as they were now using the fi0 interface. However, the backup interfaces (for example spfddi07_srvc) could not communicate, as they were now using the css0 interface.

To reinitialize the switch, the whole RISC/6000 SP system must be rebooted. This scenario deals with multiple points of failure which is outside the scope of this book. (We are only concerned with single points of failure.) This was implemented and documented purely for the interest of those who may be faced with similar kinds of environments.

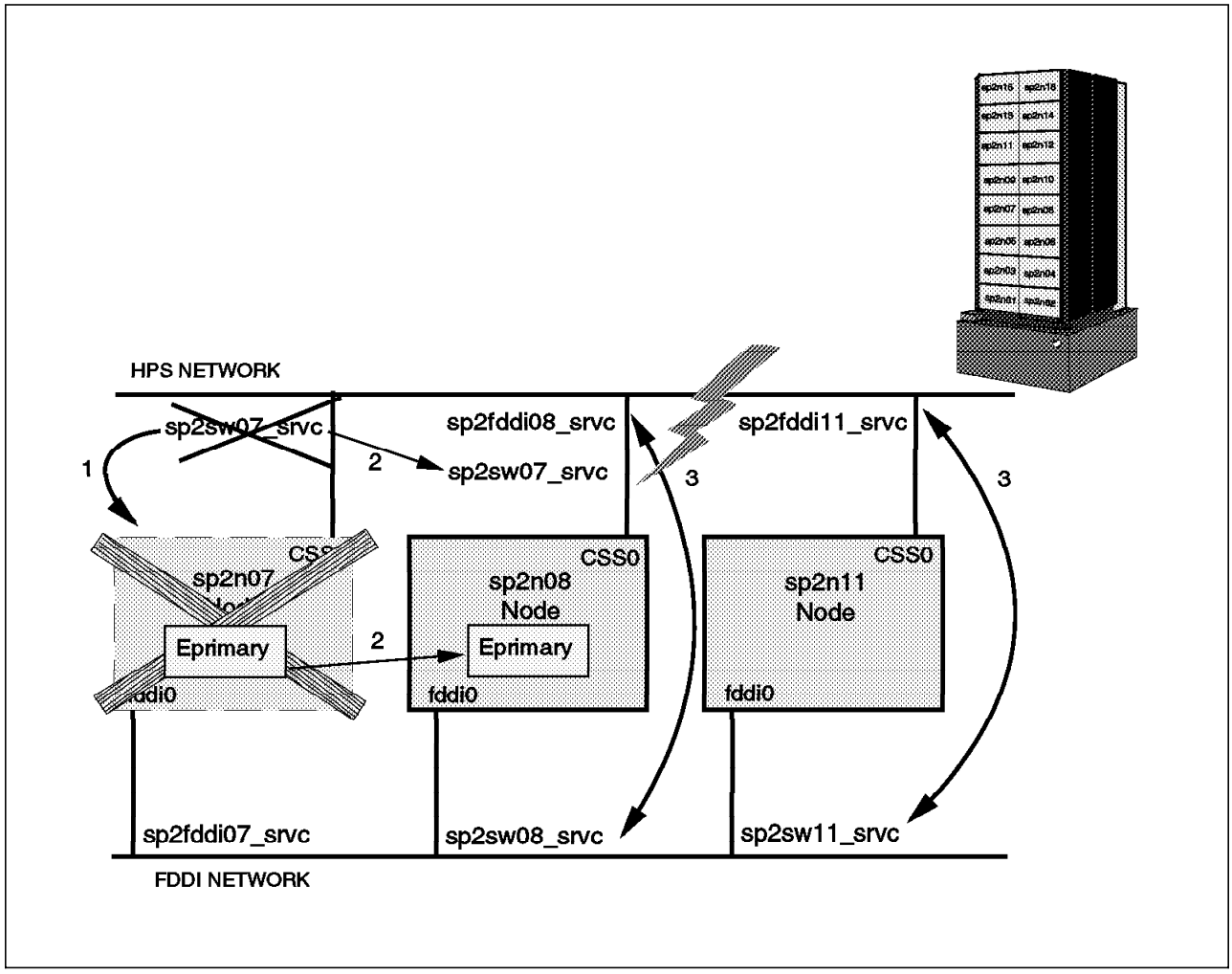


Figure 49. Global Network Failure

Chapter 4. Network Considerations

This chapter provides an insight into the considerations that need to be made concerning networks and communications when implementing High Availability on the RISC/6000 SP. It discusses the following topics:

- The switch communication network
- Resource availability considerations
- System partitioning
- External networks and adapters

4.1 High Performance Communication Network

The RISC/6000 SP switch communication network consists of two main pieces of hardware, namely, the switch board and the switch adapters. There are currently four types of switches available for the RISC/6000 SP. As the technology has progressed, the built-in redundancy and recovery provided with the switch software and hardware has also increased, creating a more fault-tolerant network that is eliminating more and more potential single points of failure.

The following is a list of the current switch technologies that exist and their respective feature code numbers:

- HiPS-8 (LC8) FC4007
- HiPS (HPS) FC4010
- SP Switch FC4011
- SP Switch-8 FC4008

The HiPS-8 and the SP switch-8 are the low-cost versions of the switch, and they only support up to eight nodes. The HiPS and the SP Switch support the connection of up to 16 thin nodes in a frame.

4.1.1 HiPS-8

The HiPS-8 switch was created as a low cost entry system alternative. It contained only a single switch chip and supports from 2-8 node configurations. This switch only contained one physical switch chip on the switch board for each of the nodes and external cable connections. System partitioning is not supported on this switch. This switch (LC8) has a different hardware than the HiPS switch. It does not allow scaling to larger systems beyond eight nodes.

4.1.2 HiPS

The HiPS has a different switch chip configuration than the HiPS-8 has. There are 16 physical switch chips on the HiPS node switch board. These are arranged into eight pairs. Four pairs of switch chips deal with the node communication and four pairs of switch chips deal with the external switch communication to other switches. The pairs are arranged such that there is a master chip and a slave chip. Both chips receive the inputs from the system, but the master chip provides the outputs to the rest of the system, while the slave uses the inputs it has received to check the data that the master has output, to

ensure that the two outputs match. This is an expensive but very reliable way of checking for errors. If the data is different, then a master/slave miscompare is generated.

When a switch fault occurs on the HiPS switch, the fault is propagated throughout the entire network. This means that all the links have to be reinitialized, which will temporarily bring the switch network down.

4.1.3 SP Switch

With this latest switch there is more redundancy and recoverability built into the technology. There are N+1 power supplies, and N+1 fans. The switch faults now only cause local faults rather than global faults, which was the case with the HiPS. The fault only affects the link where the fault occurred, and the rest of the switch network functions without interruption. This means that a node can be taken offline without causing any disruption to the rest of the network, while with the HiPS you would need to fence the node.

There is also a new feature of a primary and backup node for the Eprimary node. This is discussed in more detail in section 4.2.2, “Eprimary Considerations with the Switch” on page 146. This eliminates the Eprimary as a potential single point of failure.

The switch chip has also been redesigned so that there is now no master and slave relationship with two physical chips, but a single switch chip with enhanced error checking capabilities. This improves reliability in the switch board as it reduces the number of possible component failures.

A second oscillator has been added to the new switch clock, but the process of recovery will be a manual one of switching from the master oscillator to the backup.

This switch is not compatible with the HiPS switch as the two have different Recovery, Availability and Serviceability (RAS) characteristics, along with a different link technology used in each switch.

4.1.4 SP Switch-8

This is basically the same switch as the SP switch, but it only supports up to eight nodes. It does not allow scaling to larger systems beyond eight nodes, but it does support system partitioning. This is one of the advantages of the SP switch-8 over the HiPS-8, which does not support system partitioning.

4.2 Resource Considerations with the Switch

When configuring a RISC/6000 SP system to be highly available, there are several hardware aspects concerning the switch that need to be considered. The switch is designed in such a manner that each switch chip services four nodes, and therefore in a frame there are four switch chips handling connections to the nodes. If one of the chips fails, then communication through the switch for all four nodes serviced by that chip will be lost. Figure 50 on page 139 shows the layout of the switch chips on the switch board, and which chips are connected to which nodes and the external connections that go to other switch chips. These other switch chips reside on other switch boards that serves another frame.

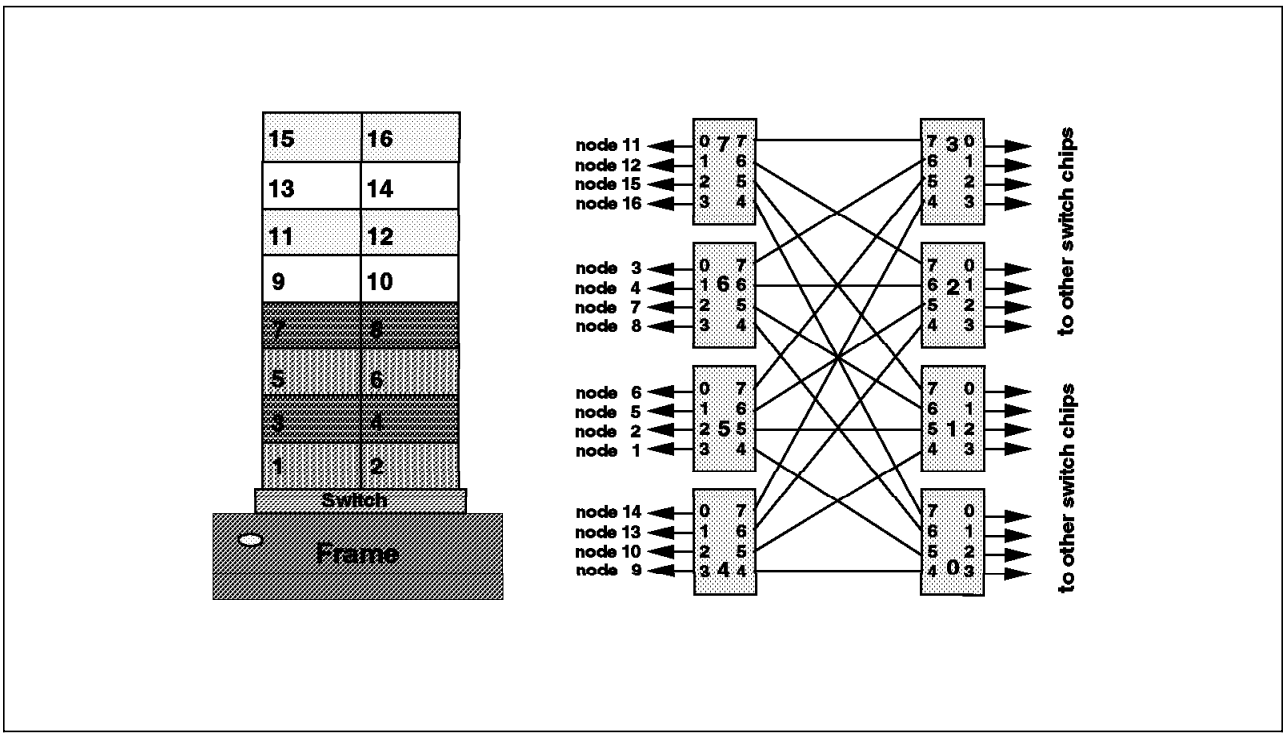


Figure 50. Switch Chip Layout

Figure 50 illustrates how the nodes and external cables are connected to the switch chips on the switch board. For disk subsystems to be highly available, they need to have at least two paths for communication, to protect against adapter failure and node failure. If the nodes that the disk subsystem are connected to are on the same switch chip, then there is a possibility that switch communication of data on the disk subsystem could be lost. Some chip failures such as a broken clock tree or central queue in the switch chip are failures that are capable of affecting all the nodes connected to a particular switch chip. However, the mean time between failure rate for the chips and other components is very high, so the probability of an actual failure of the chip is highly unlikely. When planning the layout of the RISC/6000 SP and which nodes are to perform specific functions, this should be taken into account so that perhaps the disk subsystem connections could be connected across switch chip boundaries. This not only applies for disks but for any applications and resources that you need to make highly available, such as VSD and databases.

The switch chips provide access to the following nodes.

Table 8. Switch Chip Nodes. Which chip connects to which node?

Switch Chip Number	**Node numbers that connect to the switch**
Switch Chip 4	Nodes 14, 13, 10 and 9
Switch Chip 5	Nodes 6, 5, 2 and 1
Switch Chip 6	Nodes 3, 4, 7 and 8
Switch Chip 7	Nodes 11, 12, 15 and 16

4.2.1 General Clock Path

The data path operation just described is the same for both the High Performance Switch (HiPS) and the SP-switch. Although for both switches the clock path distributions are similar at system level, they are different at the physical jack distribution level on the board. These differences are illustrated in Figure 51 on page 141 and in Figure 53 on page 143. The switch clock is used to ensure proper data reception, and it is also a mechanism that ensures that the "Time of Day" across the switch system is the same. There is one master switch board, so any additional switch boards (slave boards) in the system will obtain clock from the master or a redriven version of the clock. This maintains frequency synchronization, not phase synchronization. The phase synchronization is achieved during switch initialization. There are two kinds of phase initialization, which mainly depends on the type of switch in use, and they are:

- Software initialization
- Hardware initialization

The software initialization is performed during the Estart operation with the High Performance Switch (HiPS) while the hardware initialization is performed during Eclock operation with the SP-switch at power on time.

4.2.1.1 HiPS Clock Path

The HiPS switch clock card takes inputs from Jacks 3, 5, and 7 on the switch board, together with an onboard oscillator on the clock card in the HiPS switch. It selects one of these four inputs and then redrives it to the HiPS switch card. On the switch card the clock tree (source) is redriven to two major branches, one to the data cables and the other to the switch chips. It is then branched again so that six branches exist, four branches to the data cables and two branches to the switch chips. Each branch is driven by a different redrive chip. The clock tree branch is a potential point of failure on the switch board; for example, the redrive chip could fail.

Figure 51 on page 141 illustrates how the switch chips are fed by each branch of the clocking tree. You may also want to ensure that the connection of the subsystems takes into account the switch chip clocking tree as well. The following list shows how the switch chip clocking tree branch logic divides up the nodes. If you lose a clock tree branch, you can lose connection to the switch of eight nodes. You may want to ensure that resources are spread across not only switch chip boundaries, but clock tree branches (redrive chips) as well.

- Switch chip pairs 0 and 1
- Switch chip pairs 2 and 3
- Switch chip pairs 4 and 5
- Switch chip pairs 6 and 7

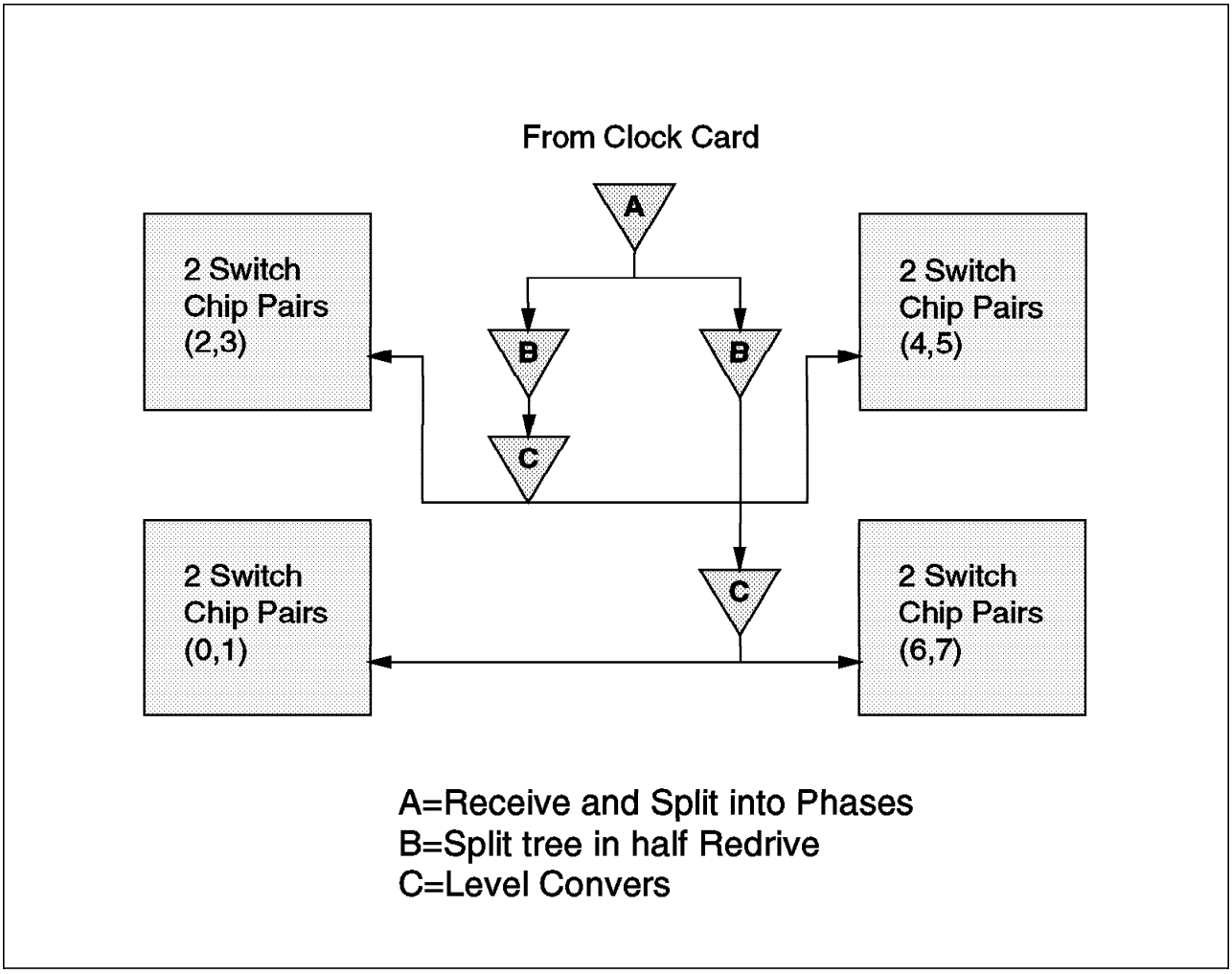


Figure 51. HiPS Switch Chip Clocking Tree

Planning is necessary to protect your resources against component failure. If you have an external disk containing data that you access through the switch, then you could set the system up so that disk access is through nodes 1 and 3. This would protect against clock tree branch failure and switch chip failure. Figure 52 on page 142 demonstrates this example.

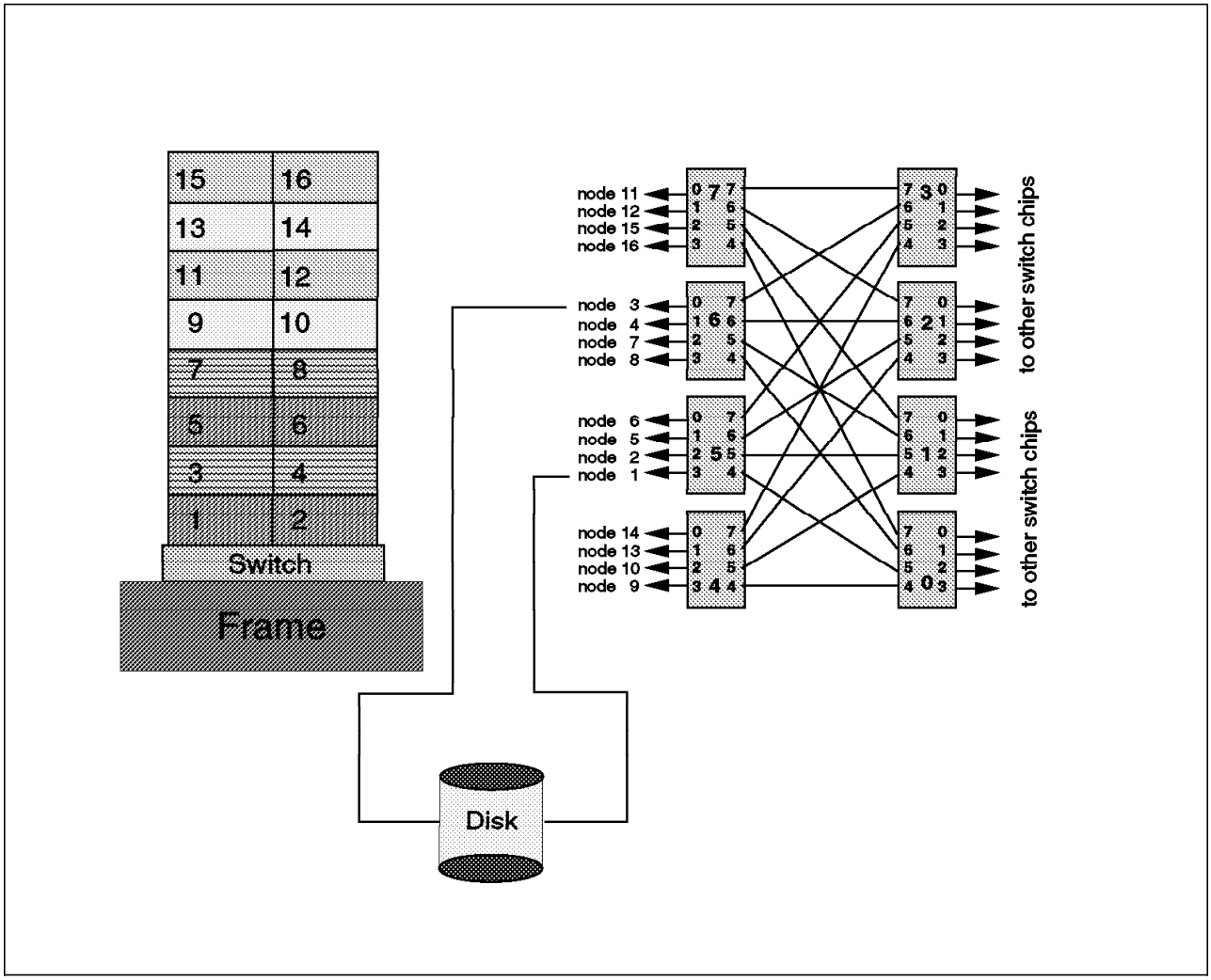


Figure 52. Resource Setup to Protect against Switch Component Failure

The whole switch board can fail due to a clock tree failure or a power failure. To protect resources against complete failure of the switch, you may spread them across several switches for fault tolerance. If this is not a valid option, then High Availability Cluster Multi-Processing could be used to recover the IP addresses so that the resources could still communicate. Implementation of HiPS recovery with High Availability Cluster Multi-Processing is discussed in 3.7, "Implementing High Availability for HiPS Network Failure" on page 105. This option makes use of an external network and should only be used as a temporary backup since communication will be much slower compared to the HiPS.

4.2.1.2 SP-switch Clock Path

Figure 53 on page 143 is an illustration of the SP-switch chip clocking tree. It consists of a single chip, which is chip 2 in this example. Chip 2 serves as the master chip on the master board. The input to chip 2 is through an oscillator called oscillator 2. This name is derived from the fact that the master chip is the chip 2. Within this master chip, there is a clock selector function, which is akin to the clock card in the HiPS, and the internal drivers to the other chips, which provide the connections to the nodes and to other switch boards. It should be noticed that the internal drivers to the other chips are all at the same level, unlike the HiPS in Figure 51 on page 141, which has four driver levels (A,B,C,C) to the chips. One of the reasons for the redesign of the SP-switch chip is to

ensure optimum availability by the apparent elimination of the different levels of drivers and interdependency among chips. With the reduction in the number of components, and with each of the chips having its own driver directly from the master chip, the mean time between failure rates is about four to ten times higher than the High Performance Switch (HiPS).

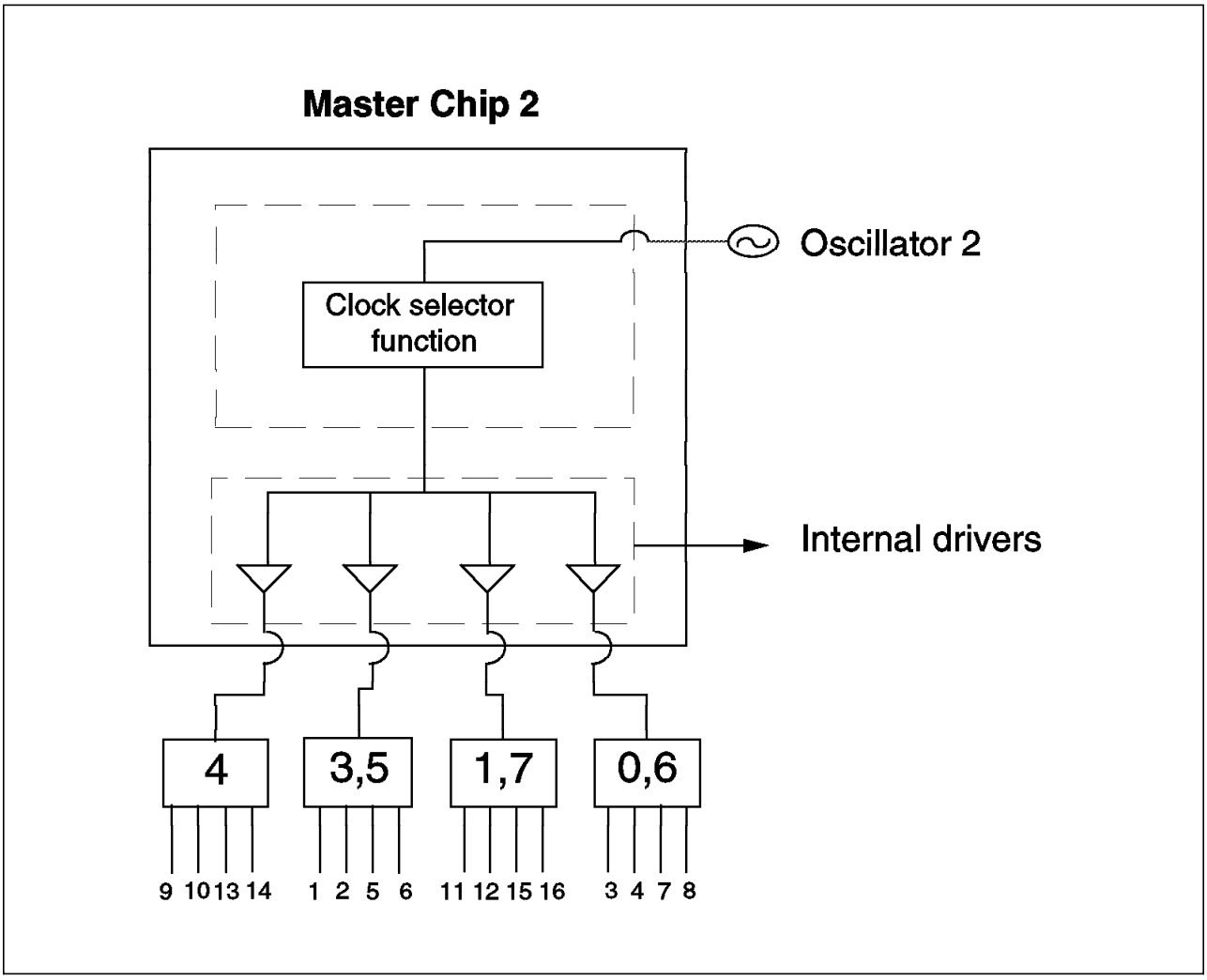


Figure 53. SP-switch Master Chip

The SP-switch is designed with a backup master chip to ensure optimum availability in the event of an unplanned outage of the oscillator 2. However, this recovery process is currently manual and the procedure is not covered in this document. Figure 54 on page 144 is an illustration of the SP-switch backup master chip. Chip 2 is the master chip, and the backup master is chip 4. The backup chip is akin to the master chip except that the inputs to chip 4 will be from a different oscillator called oscillator 4.

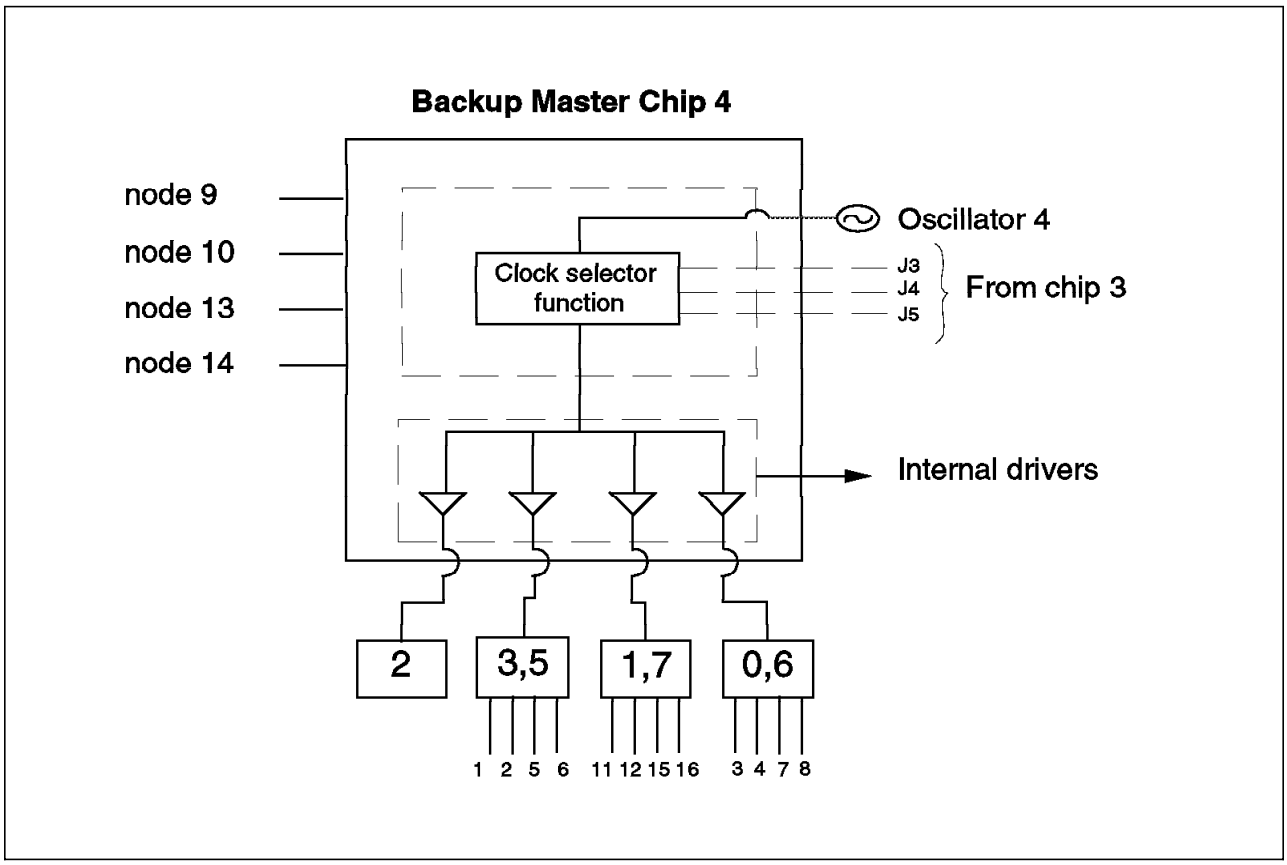


Figure 54. SP-switch Backup Master Chip

Chip 4 has a dual mode of operation. In the first mode of operation, it functions as the backup master with oscillator 4 as shown in Figure 54. The second mode of operation is when chip 4 is operating on the slave board with inputs from jacks J3, J4, J5, which are fed from chip 3 as shown in Figure 55 on page 145. Again, as with oscillator 2, the name oscillator 4 was derived from the fact that the backup master chip is the chip 4. One other apparent difference is that connections to nodes 9, 10, 13, and 14 are still on chip 4 and chip 2 is not connecting to any node.

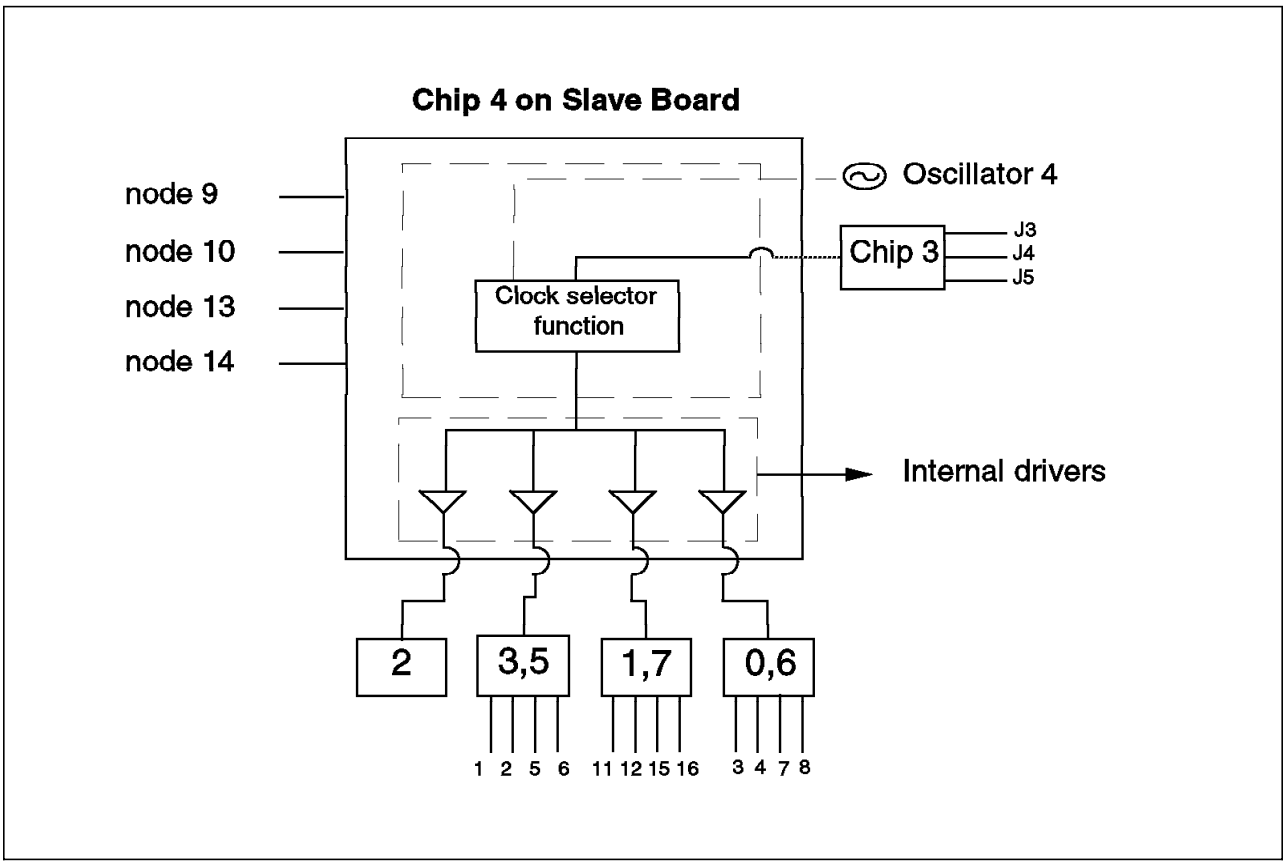


Figure 55. SP-switch Chip 4 Driven from Chip 3

Having illustrated the diagrams of the SP-switch chip layout, we are now armed with the knowledge of how we could plan our RISC/6000 SP system to be better protected against an unplanned system outage. To maximize on the Recoverability, Availability and Serviceability (RAS) characteristics of the SP-switch technology we will use Figure 56 on page 146 to show how the RISC/6000 SP resources could be connected across the different SP-switch chips to ensure that a failure on any of the chips or the internal driver does not impact the availability of the resources. From the picture, we could see that if our disk resources were to be connected to nodes 1 and 2 instead of nodes 1 and 3 or 11 then any impact on chip 5 or another component such as the internal driver to the chip will cause a complete outages of the disk and the data stored on the disk will be unavailable.

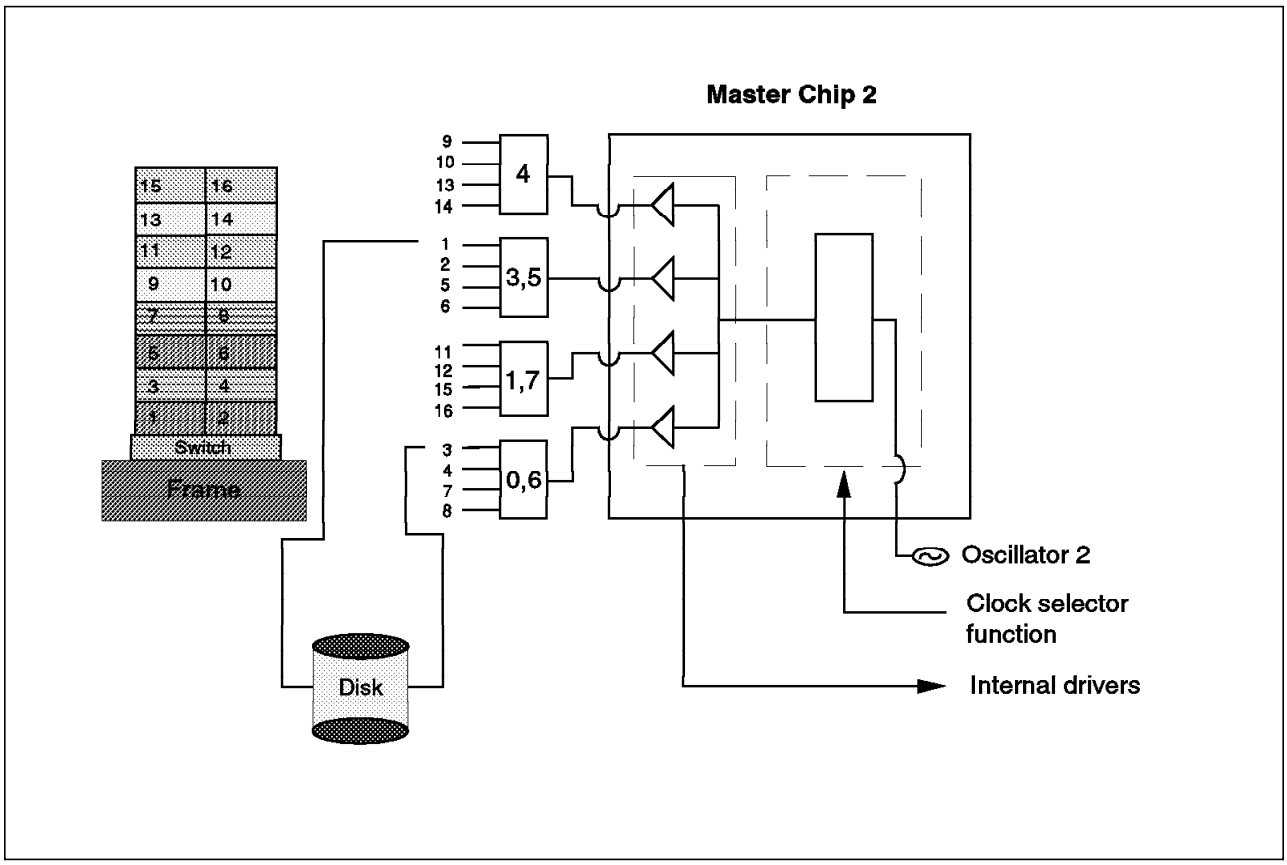


Figure 56. Resource Configuration for High Availability

4.2.2 Eprimary Considerations with the Switch

The Eprimary node is the node that initializes the entire switch network and recovers when switch faults are detected. The primary node uses available connections and information contained in the topology file to dynamically regenerate routes on the network. The Eprimary node can be set using the `Eprimary` command. This will also show the current designated primary node.

With the HiPS switch, the Eprimary has no backup. If the Eprimary node fails, then a new node needs to be manually selected and the `Estart` command needs to be run to restart the switch. The old Eprimary node must be rebooted in order for the node to be reintegrated into the switch mechanism as a normal node, not the Eprimary node. The reason for this is that the worm that runs on the Eprimary node is different than the worm that runs on the other nodes. The Eprimary node worm is a slightly modified worm that monitors the worms on the other nodes. When the new Eprimary node is initialized, it is running the modified worm. For the old Eprimary node to be monitored by the new Eprimary, it needs to be running the correct worm; therefore it must be rebooted to destroy the stale Eprimary worm, which it had previously been running, and to start the normal worm process. This is a single point of failure for the HiPS. A solution to this single point of failure has been given in 3.6, “Implementing High Availability for Eprimary and HiPS Adapter Failure” on page 86, which uses High Availability Cluster Multi-Processing to redefine a new Eprimary node and restart the switch.

The SP-switch has eliminated the Eprimary node as a single point of failure by introducing a new node type called the primary backup node. The primary backup node listens periodically to the primary node. If it does not hear from the primary node, it assumes the primary node role. This reinitializes the switch without disruption to the other nodes in the system. It updates the SDR (System Data Repository) and selects a new primary backup for itself. This primary node also listens for a backup node to ensure that it is alive. If this is not the case then it will select a new primary backup node. The primary node and the primary backup nodes can be selected using the `Eprimary` command and are updated with the `Estart` command. The primary and primary backup nodes should be selected using the switch considerations discussed in 4.2, "Resource Considerations with the Switch" on page 138. As mentioned above, care should be taken when selecting the nodes which you want to keep highly available, so switch hardware characteristics such as switch boards and switch chip boundaries must be taken into consideration.

The SP-switch has moved the error recovery of the switch from global recovery to local recovery. Instead of the entire switch network being affected when a switch fault occurs, the error detection and fault isolation is executed at the link level, therefore there is no disruption to the rest of the switch network. The node can also be brought back online with no disruption to the switch network by using the `Eunfence` command. The `Estart` command is used much less frequently than it is with the HiPS switch.

4.3 System Partitioning

With PSSP 2.1, the RISC/6000 SP system can be divided into partitions by switch chip boundaries. You can have a minimum of two wide nodes or four thin nodes in a partition. This basically splits the system logically into separate parts. This is useful for separating application environments such as production and development. This can provide the ability to isolate problems to a particular partition. An example of such a problem can be provided by the switch. With the HiPS adapter, when a switch fault occurs, it happens globally across a switch topology. When a system is partitioned into two or more logical partitions, each partition will have its own switch topology. They use the same physical switch, but the links that connect to the two partitions are logically removed so that each logical switch operates separately from other partitions. There is also a separate Eprimary node for each partition.

The heartbeat daemon (*hbd*) that runs on the administrative Ethernet has one instance running for each partition on the control workstation. You can determine which heartbeat daemon corresponds with which partition by running the following command:

```
lssrc -g hb
```

Each heartbeat daemon will communicate with those nodes in its partition. The nodes corresponding with the partition are stored in the SDR. Other daemons that have one instance per partition on the control workstation are the *sdrd* and the *hrd*. You can determine which instance corresponds to which partition by running the following commands:

For the sdrd: `ps -ef | grep sdrd`

The IP address associated with the partition is given as a parameter on the process.

For the hrd: `lssrc -g hr`

The current partition is governed by the *SP_NAME* variable. If nothing is set then the default partition is used. You must not set this variable to ″ ″ (nothing in quotes) or it will fail. You must use the command `unset SP_NAME` to unset the variable.

When partitioning a system there are certain things that you must bear in mind. High Availability Cluster Multi-Processing cannot be used across boundaries, so if there are nodes that you wish to have in the same cluster then they must be in the same partition. If you have twin-tailed disks, they must also not cross partition boundaries; the same applies to virtual shared disk pools.

Data about the partitions is held in the SDR. When a partition is created on a system, there are certain parts of the SDR that relate just to that partition.

4.4 External Networks and Adapters

Various adapters are used for network connectivity on the RISC/6000 SP, some optional, that can provide connectivity to mainframe networks, workstations and other RS6000s. Ethernet, token-ring, HiPPI, SCSI, FCS, and ATM can all be used on the RISC/6000 SP. Keeping these adapters highly available requires the same consideration taken with any RS6000 machine. High Availability Cluster Multi-Processing can be configured to recover adapters if there is a standby adapter on the machine. If a node failure occurs, HACMP can be configured to swap the service addresses to another machine so that access to resources can be maintained. This topic is covered extensively in *An HACMP Cook Book*, SC23-2773-01. Network failure is also covered in this manual. HACMP recognizes network failure and recovery but takes no actions. These need to be customized for the individual situation depending on the required actions, such as recreating routes and making use of another network.

Chapter 5. Implementing HACMP with RVSD on the RISC/6000 SP

This chapter describes the Virtual Shared Disk (VSD) and Recoverable VSD, and the steps used to integrate them in an High Availability Cluster Multi-Processing environment.

5.1 Virtual Shared Disk Overview

The Virtual Shared Disk lets application programs executing at different nodes of a cluster access a raw logical volume as if it were local at each of the nodes. In actuality, the logical volume is local at only one of the nodes, and this node is called a VSD server node. The nodes that are remotely accessing the logical volume are called VSD client nodes.

When a write or a read operation is issued to a VSD, the VSD device driver determines the server node and routes the request to that node, either locally or remotely.

5.1.1 Architecture

The I/O routing is done by the Virtual Shared Disk device driver layer that sits on top of the LVM (Logical Volume Manager). The Device Driver is loaded as a kernel extension on each node (client or server). Thus the logical volumes are globally accessible.

Figure 57 on page 150 illustrates the different layers passed by the data flow.

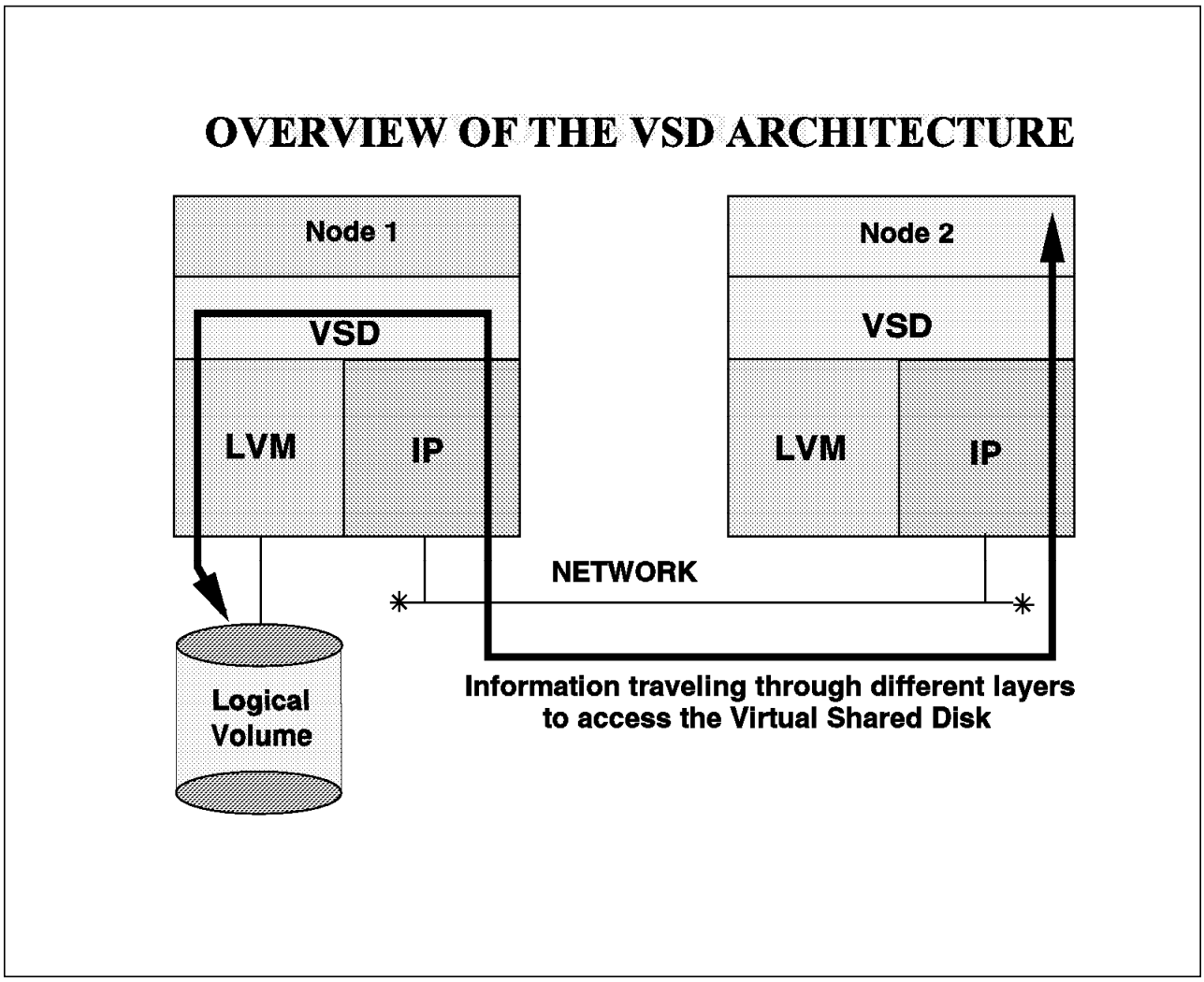


Figure 57. Different Layers Crossed by VSD

The Virtual Shared Disk is defined to the system through the SDR of the control workstation. There are three objects defined:

- VSD_GVG, is the table of volume groups.
- VSD_Table, is the table of VSD.
- All the Virtual Shared Disk attributes are defined in the SDR node object.

These objects are extracted from the SDR to local files which are located in /usr/lpp/csd/vsdfiles. Examples of these files are shown in topic 5.7.2, "Defining a Client Node on the Same Frame" on page 174.

5.1.2 Communication between Server and Clients

The Virtual Shared Disk uses its own protocol similar to TCP and UDP over IP. The requesting node implements an exponential back-off retransmission strategy. If the remote node does not service the request after about 20 minutes of retransmission, the Virtual Shared Disk fails the I/O request.

The network connection used between the server and the client nodes can be either a Local Area Network (LAN) for example, Ethernet or it can be the RISC/6000 SP High Performance Switch (HiPS). The communication protocol

supported for use with VSD is the IP network protocol such as the High Performance Switch and the performance is much better compared to other types of network. The *userspace* protocol is not currently supported with VSD.

It must be kept in mind, when implementing VSD and RVSD, that some hardware problems like switch chip failure and switch clock failure can be handled by choosing nodes that are on different switch chips and different switch clock tree branches. For more information, see 4.2, “Resource Considerations with the Switch” on page 138.

5.1.3 Data Integrity

A Virtual Shared Disk is seen like a device. The application using the VSD must supply a lock manager in order to allow several nodes to access the same data. If that single serving node should fail, access to the disk is lost until the serving node is rebooted by the administrator. The purpose of the Recoverable Virtual Shared Disk is to prevent this kind of failure.

Users with applications exploiting the Recoverable Virtual Shared Disk can recover more easily from node failures and have continuous access to the raw data on the disks which are twin-tailed. It is important to notice that Virtual Shared Disk manages only raw devices.

The AIX filesystem is not supported by Virtual Shared Disk.

On a single node, users must provide and use their own synchronization mechanisms for read and write to ensure data integrity. On a multiple node access configuration, a synchronization mechanism must be provided to ensure data integrity. Neither Virtual Shared Disk or the RISC/6000 SP provides this synchronization mechanism.

5.2 Recoverable Virtual Shared Disk Overview

This topic briefly describes the Recoverable Virtual Shared Disk mechanism and its processes.

5.2.1 Mechanisms

The Recoverable Virtual Shared Disk (RVSD) software provides availability by recovering the Virtual Shared Disk software on a backup node and by taking over the shared data. This data must be installed on twin-tailed disks. The installation of these devices is detailed in *High Availability Cluster Multi-Processing 4.1 for AIX Planning Guide*, SC23-2768, and in the *Installation Guide*, SC23-2769.

The purpose of RVSD is to detect a problem on the active node and to take over the VSD functions on a standby node. The RVSD must be able to do a certain number of tasks:

- Node failure detection
- Notification to other nodes
- Take corrective actions to allow access to data from other nodes
- Maintain an updated understanding of the status of VSDs

The Recoverable Virtual Shared Disk lets you use twin-tailed disks and configure nodes as primary and secondary VSD server nodes. It also provides a transparent switchover to a secondary server if the primary node fails.

When the node failure occurs, the backup node takes over the primary node. It varies on the volume group, and makes the VSD available and active again. No intervention is needed. If a node fails, any affected VSDs are suspended until takeover has occurred. Because I/O is pending while a Virtual Shared Disk is suspended, applications should only see a short delay and no I/O or data should be lost.

5.2.2 Processes

Recoverable Virtual Shared Disk processes are used to provide disk availability:

- The heartbeat daemon (hb) is running on each node in the RISC/6000 SP, including the control workstation, and is used to exchange heartbeats with its peers on the other nodes over the reliable network (SP Ethernet). If a node fails, then the other heartbeat deamons will declare it dead and Recoverable Virtual Shared Disk will react accordingly (if High Availability Cluster Multi-Processing is not installed).
- The recovery deamons (ha process) run on every Virtual Shared Disk node and receive information from the heartbeat daemon. If ha is notified by the heartbeat that a node is not available, ha invokes the recovery scripts and notifies application programs (through the hc process).
- The connection manager process (hc process) is responsible for providing membership information to the hc clients. The hc daemon also executes the hc.activate when it starts, and hc.deactivate when a node stops.

5.3 Hashed Shared Disks Overview

The Hashed Shared Disks (HSD) software is an additional software that can be installed over VSD in order to provide the ability to stripe a device over existing VSD devices. It is useful in helping distribute I/O over several nodes in a performance consideration.

The HSD is defined after defining VSD on each node. It takes place on two nodes or more. The HSD device driver takes place over the VSD device driver; it has the same limitations as VSD (it supports only raw devices).

5.4 Why Use RVSD and HACMP on the RISC/6000 SP?

This topic describes the interests of using the High Availability Cluster Multi-Processing on a Virtual Shared Disk node.

5.4.1 Combining RVSD and High Availability Cluster Multi-Processing

The use of Recoverable Virtual Shared Disk can be combined with High Availability Cluster Multi-Processing in order to provide the optimum availability for a given application on RISC/6000 SP. The combination of these two products can easily help build a strong automated solution against hardware and software failures. Adding High Availability Cluster Multi-Processing over RVSD has a meaning only if the aim is to protect other points of failure in combination with Virtual Shared Disk (See 3.1, “HACMP Solutions Matrix for Potential Single Points of Failure on RS/6000 SP” on page 62). It must be considered that RVSD

by itself provides security at the node level, and some protections at the software layer (only VSD) level.

The following are some of the functions that it does not protect:

- Access to the data from an external network point of view
- Software supervision
- Error notification
- System recovery (spool)

The only point of failure that High Availability Cluster Multi-Processing cannot secure on its own is the VSD software. This is one of the advantages of using HACMP and RVSD.

5.4.2 Advantages of High Availability Cluster Multi-Processing with RVSD

Using HACMP with RVSD can provide strong protection of the nodes:

- The heartbeat function provided by High Availability Cluster Multi-Processing is more flexible and secure than RISC/6000 SP′s heartbeat (used by RVSD), because it can be run on any network (with TCP/IP), on RS232, or Target Mode SCSI. High Availability Cluster Multi-Processing bypasses the RISC/6000 SP heartbeat to get its information from the other nodes defined in the same High Availability Cluster Multi-Processing cluster.
- HACMP provides granularity in eliminating the single points of failure that are not treated by RVSD. HACMP can protect against different network failures, and it can handle other types of logical volumes protection, like file systems. It can also secure spool and other system tasks. This list is not exhaustive.

It is important to note that the coexistence of RVSD controlled by HACMP for AIX and RVSD controlled by the heartbeat (hb) daemon within the same RISC/6000 SP or within the same partition is not supported.

If a set of RISC/6000 SP nodes run HACMP, and the rest of the nodes run RVSD with the ha/hc configuration, then the following rules must be observed:

- No RVSD can be managed from the HACMP nodes. It is recommended that these nodes be removed from the VSD configurations if they exist (for example, do not define a VSD_adapter in SDR for these nodes).
- HACMP can be used and configured normally, but must *not* manage VSDs in this system.

Following is a table comparing the two products:

Table 9 (Page 1 of 2). High Availability Cluster Multi-Processing and Recoverable Virtual Shared Disk Comparison

Function	**RVSD**	**HACMP**
Number of nodes	128	8
Configurations - Idle Standby, Mutual failover, Flexible failover, Cascading failover	Support limited configuration options	Yes
Selective application failover	No	Yes
Heartbeat over IP network	Ethernet	All

Table 9 (Page 2 of 2). High Availability Cluster Multi-Processing and Recoverable Virtual Shared Disk Comparison

Function	RVSD	HACMP
Heartbeat over SCSI and RS232	No	Yes
Hardware Error detection	Ethernet heartbeat is the only mechanism	Relies on heartbeat and errorlog entry
Hardware/Software Error Recovery	Recovery possible only for missed heartbeats over Ethernet	Recovery possible only for a fixed number of events
Ip address takeover	No	Yes
Client notification/reaction to cluster events	No	Yes (using clinfo)
Concurrent access support (DLM/CLVM)	No	Yes
Support for RISC/6000 SP and Risc/6000 clusters	RISC/6000 SP support only	Yes
Recovery scripts provided for key subsystems and applications	No	Yes
Coordination and synchronization services for distributed processes	For VSD only	No
SNMP Integration	No	SNMP Traps generated for a fixed set of events

5.4.3 Defining Resources to HACMP and RVSD

We have seen in topic 5.4.1, “Combining RVSD and High Availability Cluster Multi-Processing” on page 152 that RVSD by itself offers high availability. What is the interest of installing HACMP over RVSD? If we compare RVSD with HACMP, we can see that they have some similarities and some differences. The first advantage is that RVSD protects the VSD as a shared resource, backing up a VSD server node with an alternate server. This can be compared to the mutual take over configuration of HACMP.

The functions will be close to a rotating configuration, with some differences due to the interaction of RVSD and HACMP. The main point of the rotating resource in an HACMP point of view is that the first node which starts takes the available resource and keeps it whatever happens. The second node can start, but the resource will not be released. The first node that is started is the owner of the resource. On the RVSD the behavior is slightly different, as the first node which starts will take the VSD, even if it does not belong to itself. But when the “server” node starts, the other node will drop the VSD in order to give it back to the server.

We know that the interest of HACMP over RVSD is to protect resources other than the VSD itself. The customization of these resources will be slightly different than a normal configuration, due to the fact that it is HACMP that controls the resource, but RVSD still controls the VSD. This customization is different only if we want the resource to follow the VSD. It is VSD′s role to get back the shared disks and to activate the shared volume groups in case of a failure. Refer to topic 5.7.1, “Defining Resources in HACMP” on page 170 for an example of the configuration of a shared resource.

Another difference between these two products is the behavior of a VSD compared to the behavior of an HACMP resource while shutting down a server node. If a node holding an HACMP resource is also the owner of a VSD and it is shut down, then it will release both resources (HACMP and VSD), but the VSD will be activated on the backup node. This is because RVSD does not have a "stop graceful" option as does HACMP. It only knows the "stop graceful with takeover."

When shutting down a node, HACMP is stopped with the force option by the `shutdown` command, so it does not give the backup node the option to take over its resources.

This is not a important fact, as a shutdown is a programmed command, but it has to be kept in mind that the VSD will be activated automatically on the backup node.

Refer to topic 5.7.1, "Defining Resources in HACMP" on page 170 for an example of the behavior of a shared resource compared to a VSD.

5.5 Planning for HACMP and RVSD Installation

This section discusses the software and PTFs required to install and customize RVSD and HACMP on the RISC/6000 SP.

The processes of installing and customizing HACMP on RISC/6000 SP system are described in *High Availability Cluster Multi-Processing 4.1 for AIX Installation Guide*, SC23-2769 and *Planning Guide*, SC23-2768. Also refer to 3.3, "Installing HACMP on the RISC/6000 SP System" on page 64.

5.5.1 Software Prerequisites

The IBM Parallel System Support Programs for AIX (PSSP) is a prerequisite for all the options in the RCSD installp package.

This prerequisite software is described below for the different levels of the AIX operating system.

5.5.1.1 For AIX 3.2.5

On nodes and on the Control Workstation:

- Virtual Shared Disk:
  - 1.2.0.0 csd.vsd
  - 1.2.0.0 csd.cmi
  - 1.2.0.0 csd.hsd
  - rcsd.tool

On nodes:

- Recoverable Virtual Shared Disk:
  - rcvsd.vsd
  - rcsd.ha
  - rcsd.hc
  - rcsd.docs

- High Availability Cluster Multi-Processing Version 3.1.1:
  - cluster.server
  - cluster.client
- Required PTFS:
  - U440799
  - U440800
  - U440801
  - U440802

For SSA disks: install fix IX56768

5.5.1.2 For AIX 4.X

On nodes and on the Control Workstation:

- Virtual Shared Disk:
  - ssp.csd.vsd
  - ssp.csd.cmi
  - ssp.basic

On nodes:

- Recoverable Virtual Shared Disk:
  - rcsd.vsd
  - rcsd.ha
  - rcsd.hc
  - rcsd.tool
- High Availability Cluster Multi-Processing:
  - cluster.adt*
  - cluster.base*
  - cluster.msg*

    Note: The * indicates that all the filesets are recommended.
- Required PTFS:
  - ssp.csd.vsd U441822 2.1.0.3
  - ssp.csd.hsd U441618 2.1.0.1
  - rcsd.ha U441823 1.1.1.1
  - rcsd.hc U441825 1.1.1.1
  - rcsd.vsd U441824 1.1.1.1

Note: If you plan to install your data on RAID disks, the last level set for the scarray code is highly recommended:

devices.scsi.scarray.rte.4.1.4.3

5.6 Installing RVSD and HACMP

This section describes the installation of a VSD platform. It is a VSD environment with two servers backing up each other, and a client node on the same frame accessing both servers. ORACLE is not implemented in this installation.

This installation is done on a RISC/6000 SP with AIX 4.1.4, PSSP V2.1, RVSD 1.1.1 and HACMP 4.1.1.

5.6.1 Pre-Installation Procedure

The first step is to install the hardware parts for the shared disks and prepare them to receive the shared data. This is described in the *High Availability Cluster Multi-Processing 4.1 for AIX Planning Guide*, SC23-2768 and in *Installation Guide*, SC23-2769.

Important Note

Before installation, install all the previously mentioned APARS and PTFS. You may have to contact your support center in order to obtain the latest PTFs available.

The second step is to install the software for VSD, RVSD and HACMP on the nodes sharing VSD through, *smit install_latest,* with the latest PTFs.

5.6.2 VSD Installation

Figure 58 on page 158 illustrates how we will implement the RVSD/HACMP in our example.

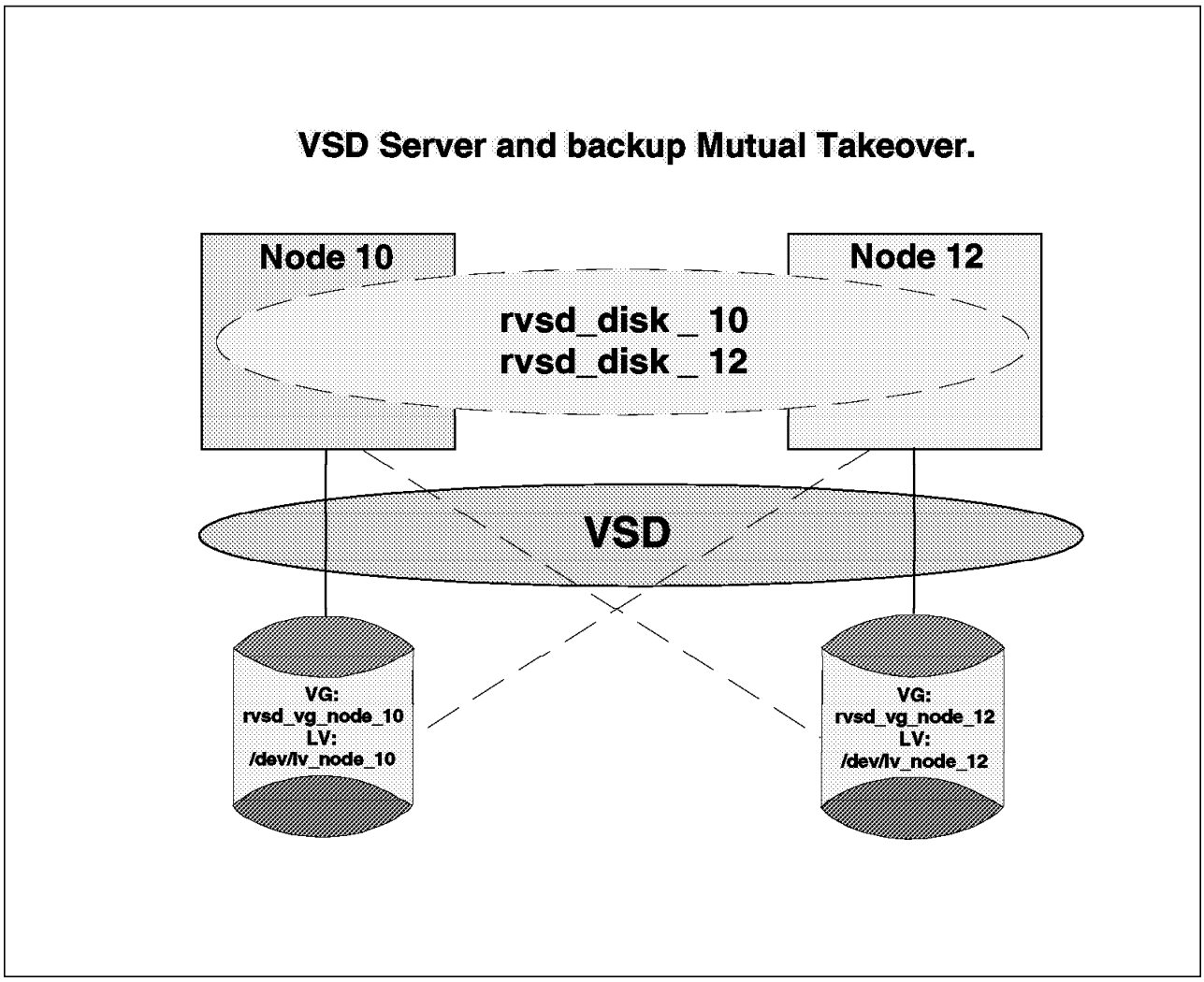


Figure 58. Configuration of VSD Cluster

5.6.2.1 System Setup

The different names given in this example are as follows:

- Node 10 and node 12 are the two VSD servers.
- Node 11 is a VSD client.
- Volume group rvsd_vg_node10 is created on node 10 and is also known on node 12.
- Volume group rvsd_vg_node12 is created on node 12 and is also known on node 10.
- Logical Volume lv_node_10, which is created on rvsd_vg_node10.
- Logical Volume lv_node_12, which is created on rvsd_vg_node12.
- The Virtual Shared Disk rvsd_disk_10 is created on lv_node_10 and is backed up by node 12.
- The Virtual Shared Disk rvsd_disk_12 is created on lv_node_12 and is backed up by node 10.

- Node 10 and node 12 use HACMP's heartbeat on three "networks": High Speed Switch, Ethernet and SCSI. sp2swxx, sp2nxx, and tmscsixx are the interfaces used to communicate.
- vg_node_10 is the volume group used as an HACMP resource, belonging to node 10.

5.6.2.2 Installation

The first step is to define the volume groups and logical volumes that will be used by VSD. This is done first by entering the following command on node 10:

```
# 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                     [rvsd_vg_node10]
  Physical partition SIZE in megabytes  4                              +
* PHYSICAL VOLUME names                 [hdisk2 hdisk3]       +
  Activate volume group AUTOMATICALLY   no                             +
    at system restart?
  Volume Group MAJOR NUMBER             []                             +#
```

The other volume group must be defined on node 12 in the same way (`smit mkvg`, for rvsd_vg_node_12 on node 12). As in a standard HACMP installation, the volume group *must* remain dormant at start up.

Once this is done, the logical volumes must be defined on each node with the command:

```
# smit  mklv
```

Then choose the external volume group **(rvsd_vg_node_10)**.

```
                              Add a Logical Volume

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

[TOP]                                                 [Entry Fields]

  Logical volume NAME                              [lv_node_10]
* VOLUME GROUP name                                rvsd_vg_node10
* Number of LOGICAL PARTITIONS                     [10]
  PHYSICAL VOLUME names                            [ ]
  Logical volume TYPE                              [ ]
  POSITION on physical volume                      middle
  RANGE of physical volumes                        minimum
  MAXIMUM NUMBER of PHYSICAL VOLUMES               [ ]
    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?
  RELOCATE the logical volume during               yes
    reorganization?
  Logical volume LABEL  middle                     [ ]
  MAXIMUM NUMBER of LOGICAL PARTITIONS             [128]
  Enable BAD BLOCK relocation?                     yes
  SCHEDULING POLICY for writing logical            parallel
    partition copies
  Enable WRITE VERIFY?                             no
  File containing ALLOCATION MAP                   [ ]
  Stripe Size?                                     [Not Striped]
[BOTTOM]
```

The volume group belonging to node 10 (rvsd_vg_node10) is now defined and so is the logical volume lv_node_10.

The next step is to define the logical volume (lv_node_12) belonging to node 12. Do this by repeating the last step on node 12.

The next step is to define the volume groups on the backup nodes. Node 10 is the backup node of node 12 for rvsd_vg_node12, and node 12 is the backup of node 10 for rvsd_node_10. So we now have to define rvsd_vg_node10 on node 12, and rvsd_vg_node12 on node 10. Following are the steps to do the definition:

From node 10, list the physical volumes to get the IDs of the Physical Volumes (PV IDs):

```
# lspv
hdisk0          000029080001bc80    rootvg
hdisk1          000029087b46165d    rootvg
hdisk2          00020321d519c0fc    rvsd_vg_node10
hdisk3          00020321d60b49fd    rvsd_vg_node10
hdisk4          00020321d60b4d42    None
hdisk5          00020321d5196f01    None
```

The next step is to vary off the volume group:

```
# varyoffvg rvsd_vg_node_10
```

We list the physical volumes on node 12 to see the corresponding PV IDs. The highlighted line is the hdisk we are going to import from.

```
# lspv

hdisk0          000027990001acc3    rootvg
hdisk1          00002799638938c3    rootvg
hdisk2          00020321d519c0fc    None
hdisk3          00020321d60b49fd    None
hdisk4          00020321d60b4d42    rvsd_vg_node12
hdisk5          00020321d5196f01    rvsd_vg_node12
```

The following step is to import the rvsd_vg_10 volume group on node 12, from the hdisk2:

```
# importvg -y rvsd_vg_node_10 hdisk2
rvsd_vg_node_10
```

The result can be listed by the following command:

```
# lspv

hdisk0          000027990001acc3    rootvg
hdisk1          00002799638938c3    rootvg
hdisk2          00020321d519c0fc    rvsd_vg_node10
hdisk3          00020321d60b49fd    rvsd_vg_node10
hdisk4          00020321d60b4d42    rvsd_vg_node12
hdisk5          00020321d5196f01    rvsd_vg_node12
```

The volume group rvsd_vg_node10 is known on both sides. The next step is to make it remain dormant at start up, and then to vary it off:

```
# chvg -a n -Qy rvsd_vg_node10

# varyoffvg  rvsd_vg_node10
```

Once this is done, the same steps must be executed for rvsd_vg_node12 for it to be defined on node 10.

5.6.3 Creating and Defining the VSD and RVSD

Once the logical volumes are defined, the next step is to create the VSD and RVSD.

The first thing to do is to define the volume groups and logical volumes into the SDR of the control workstation. This is done through the SMIT panels by issuing the following command:

```
# smit vsd_data
```

```
                              VSD Database
Information

Move cursor to desired item and press Enter.

  VSD Node Information
  VSD Global Volume Group Information
  Define a Virtual Shared Disk
  Define a Hashed Shared Disk
```

Use the highlighted option to define the VSD nodes into the SDR:

```
                         VSD Node Database Information

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

                                            [Entry Fields]
* Nodes                                     [10 12]
* Adapter Name for VSD Communications       css0
* Initial Cache Buffer Count                [64]
* Maximum Cache Buffer Count                [256]
* VSD Request Count                         [256]
* Read/Write Request Count                  [48]
* VSD Minimum Buddy Buffer Size             [4096]
* VSD Maximum Buddy Buffer Size             [131072]
* VSD Number of Max-sized Buddy Buffers     [9]
```

It is recommended to leave the default parameters at installation time for performance purposes.

For more information about the parameters on this panel, see 5.7.2, “Defining a Client Node on the Same Frame” on page 174.

The next step is to define the volume group into the SDR:

```
# smit vsd_data
```

Select the highlighted line:

```
                              VSD Database
Information

Move cursor to desired item and press Enter.

  VSD Node Information
  VSD Global Volume Group Information
  Define a Virtual Shared Disk
  Define a Hashed Shared Disk
```

```
                    VSD Global Volume Group Information

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

                                                    [Entry Fields]
* Local Volume Group Name                           [rvsd_vg_node10]
* Primary Node on which Volume Group is Resident    [sp2n10]
  Secondary Node on which Volume Group is Resident  [sp2n12]
  Global Volume Group Name                          [  ]
```

Note: The underscored line is the definition of the RVSD backup.

The last step is to define the VSD itself by selecting the highlighted line:

```
                            VSD Database
Information

Move cursor to desired item and press Enter.

  VSD Node Information
  VSD Global Volume Group Information
  Define a Virtual Shared Disk
  Define a Hashed Shared Disk
```

```
                        Define a Virtual Shared Disk

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

                                          [Entry Fields]
  Logical Volume Name                     [lv_node_10]
  Global Volume Group Name                [rvsd_vg_node10]
  Virtual Shared Disk Name                [rvsd_disk_10]
  Virtual Shared Disk Option              [nocache]
```

These steps should be repeated for node 12.

In this installation we did not work with Hashed Shared Disks. Implementing HSD over VSD is not difficult; it is only a supplementary step in SMIT (see 5.3, “Hashed Shared Disks Overview” on page 152 for more information on HSD).

At this point the VSDs are defined and may be started.

In order to start the VSDs, first look to see if the VSD device driver is loaded:

```
# cd /usr/lpp/csd/bin/
# lsvsd

lsvsd: 0034-002 Error opening vsd /dev/VSD0.
 : No such device or address
```

The result of the command shows that the driver is not loaded. The load will be done by the following command:

```
# cfgvsd -a

Initializing VSD driver with Kernel-to-Kernel Interface.
load_iptbl: loading 9.12.6.47 for node 10.
load_iptbl: loading 9.12.6.49 for node 12.
```

Repeat the command to verify if the driver is loaded:

```
# lsvsd

rvsd_disk_12
rvsd_disk_10
```

The result of the previous command shows that the VSDs are now configured. The next step will be starting the VSDs.

Attention

These steps will not be valid once HACMP is customized on the system.

Use the following command to verify the state of the VSDs on node 12:

```
# lsvsd -l

minor   state server lv_major lv_minor   vsd-name     option
 1       STP   -1     34        1       rvsd_disk_12  K-to-K
 2       STP   -1      0        0       rvsd_disk_10  K-to-K
```

The result of the previous command shows that the VSDs are in a stopped state.

The VSDs can be started by the following command:

```
# startvsd -a
# lsvsd -l
minor   state server lv_major lv_minor   vsd-name     option
 1       ACT   12      34        1       rvsd_disk_12  K-to-K
 2       ACT   10       0        0       rvsd_disk_10  K-to-K
```

The result of the `lsvsd` command shows that the VSDs are active.

The status of VSD rvsd_disk_10 seems to be active, but in fact it is not. It is only the client part on node 12 which is active; the server part (which is on node 10) is not started. The same steps must be executed on node 10 to activate the VSD on node 10.

When all the VSDs are in an active state, they may be tested by the following command, which should give a successful message:

```
# vsdvts rvsd_disk_10
```

```
vsdvts: Step 1: Writing file /unix to VSD rvsd
dd if=/unix of=/dev/rrvsd_disk_10 count=256 bs
256+0 records in.
256+0 records out.
vsdvts: Step 1 Successful|
vsdvts: Step 2: Reading data back from the VSD
dd of=/tmp/vsdvts.17104 if=/dev/rrvsd_disk_10
256+0 records in.
256+0 records out.
vsdvts: Step 2 Successful|
vsdvts: Step 3: Verifying data read from the V
dd if=/unix count=256 bs=4096 | cmp -s - /tmp/vsdvts.17104
256+0 records in.
256+0 records out.
vsdvts: Step 3 Successful|

VSD Verification Test Suite Successful|
```

The VSDs are successfully installed when the above messages are displayed.

5.6.4 Configuration of RVSD and HACMP

The implementation of HACMP over RVSD is quite simple and does not require a complicated customization of HACMP for the integration of the RVSD. It is important to keep in mind that installing HACMP over RVSD is only useful if you want to do some IP takeover on a secondary network, restart some application, or implement specific developments. See 5.7.1, “Defining Resources in HACMP” on page 170 for an example of resources associated with VSD.

To install HACMP, the only step is to define the “base” switch address as a service address on HACMP. This means that in a configuration with RVSD and HACMP, it will not be possible to do IP takeover over the switch (for more information on doing IP takeover on the switch, refer to 3.7.4.1, “ IP Address Aliasing” on page 120).

To define this address, you first must define a new HACMP cluster with its nodes (for more information, refer to 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68). When this is done, you must define the base IP address of the switch as a service IP address.

This is done by the following command:

```
# smit cm_config_adapters.add
```

```
                          Add an Adapter

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

                                                  [Entry Fields]
* Adapter IP Label                                    [sp2sw10]
* Network Type                                        [hps]
* Network Name                                        [HPSrvsd]
* Network Attribute                                  private
* Adapter Function                                   service
  Adapter Identifier                                 [ ]
  Adapter Hardware Address                           [ ]
  Node Name                                          [sp2n10]
```

It is important to keep the single points of failure in mind. This is why we will define in our cluster two interfaces for each node. The first one will be the switch and the other one will be the Ethernet interface. As TCP/IP will be another single point of failure, it will be useful to define another interface not based on TCP/IP protocol. In this example we worked with 7135 Raid array disks based on SCSI. We defined a Target Mode SCSI in order to activate the heartbeat.

The first thing to do is to see what target modes are already defined on the node. In our case, we will define the tmscsi only on nodes 10 and 12, as these are the only nodes to be connected together through a SCSI cable.

The target mode devices are automatically defined at IPL time, or when the SCSI devices are defined. If they are not defined already, see *High Availability Cluster Multi-Processing 4.1 for AIX Installation Guide*, SC23-2769.

Issue the following command on node 10:

```
# ls /dev/tms*

4150 crw-rw-rw-   1 root     system    20,  0 Mar 28 23:50 /dev/tmscsi0.im
4151 crw-rw-rw-   1 root     system    20,  1 Mar 28 23:50 /dev/tmscsi0.tm
4152 crw-rw-rw-   1 root     system    20,  2 Mar 28 23:50 /dev/tmscsi1.im
4153 crw-rw-rw-   1 root     system    20,  3 Mar 28 23:50 /dev/tmscsi1.tm
```

We have on both systems the initiator (.im) and the target (.tm). We can now define tmscsi0 on both nodes, by using the SMIT fast path command as follows:

```
# smit cm_config_adapters.add
```

```
                                Add an Adapter

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

                                                        [Entry Fields]
* Adapter IP Label                      [rvsd_10_tmsci0]
* Network Type                          [tmscsi]
* Network Name                          [tmscsi]
* Network Attribute                     serial
* Adapter Function                      service
  Adapter Identifier                    [/dev/tmscsi0]
  Adapter Hardware Address              []
  Node Name                             [sp2n10]
```

On node 12 repeat the proceeding commands that was done on node 10.

```
                                Add an Adapter

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

                                        [Entry Fields]
* Adapter IP Label                      [rvsd_12_tmsci0 ]
* Network Type                          [tmscsi]
* Network Name                          [tmscsi]
* Network Attribute                     serial
* Adapter Function                      service
  Adapter Identifier                    [/dev/tmscsi0]
  Adapter Hardware Address              []
  Node Name                             [sp2n12 ]
```

When all commands are completed, the cluster must be synchronized.

To synchronize the cluster, please use the following command:

```
# smit cm_cfg_top_menu
```

The following screen will appear:

```
                                Cluster Topology

Move cursor to desired item and press Enter.

  Configure Cluster
  Configure Nodes
  Configure Adapters
  Configure Network Modules
  Show Cluster Topology
  Synchronize Cluster Topology
```

Choose the highlighted option and press Enter. This will synchronize the cluster configuration on all the nodes.

All the additional work will be standard HACMP development. At this point you have a working RVSD/HACMP configuration.

5.6.5 Cluster Configuration

The following command will give you the actual configuration of the cluster. It will give all the details concerning the network.

```
/usr/sbin/cluster/utilities/cllscf
```

The result will be the following:

```
Cluster Description of Cluster hacmp_rvsd
Cluster ID: 11
There were 3 networks defined : HPSrvsd, ethernet, tmscsi
There are 3 nodes in this cluster.

NODE sp2n10:
        This node has 3 service interface(s):

        Service Interface sp2sw10:
                IP address:     9.12.6.47
                Hardware Address:
                Network:        HPSrvsd
                Attribute:      private

        Service Interface sp2sw10 has no standby interfaces.


        Service Interface sp2n10:
                IP address:     9.12.20.60
                Hardware Address:
                Network:        Ethernet
                Attribute:      public

        Service Interface sp2n10 has no standby interfaces.

        Service Interface rvsd_tmscsi0:
                IP address:     /dev/tmscsi0
                Hardware Address:
                Network:        tmscsi
                Attribute:      serial

        Service Interface rvsd_tmscsi0 has no standby interfaces.
```

Figure 59. Cluster Configuration (1 of 3)

```
NODE sp2n11:
        This node has 2 service interface(s):

        Service Interface sp2sw11:
                IP address:     9.12.6.48
                Hardware Address:
                Network:        HPSrvsd
                Attribute:      private

        Service Interface sp2sw11 has no standby interfaces.


        Service Interface sp2n11:
                IP address:     9.12.20.61
                Hardware Address:
                Network:        Ethernet
                Attribute:      public

        Service Interface sp2n11 has no standby interfaces.

NODE sp2n12:
        This node has 3 service interface(s):

        Service Interface sp2sw12:
                IP address:     9.12.6.49
                Hardware Address:
                Network:        HPSrvsd
                Attribute:      private

        Service Interface sp2sw12 has no standby interfaces.


        Service Interface sp2n12:
                IP address:     9.12.20.62
                Hardware Address:
                Network:        Ethernet
                Attribute:      public


                Attribute:      public
        Service Interface sp2n12 has no standby interfaces.


        Service Interface rvsd12_tmscsi0:
                IP address:     /dev/tmscsi0
                Hardware Address:
                Network:        tmscsi
                Attribute:      serial

        Service Interface rvsd12_tmscsi0 has no standby interfaces.
```

Figure 60. Cluster Configuration (2 of 3)

```
Breakdown of network connections:

Connections to network HPSrvsd
        Node sp2n10 is connected to network HPSrvsd by these interfaces:
                sp2sw10

        Node sp2n11 is connected to network HPSrvsd by these interfaces:
                sp2sw11

        Node sp2n12 is connected to network HPSrvsd by these interfaces:
                sp2sw12

Connections to network Ethernet
        Node sp2n10 is connected to network Ethernet by these interfaces
                sp2n10

        Node sp2n11 is connected to network Ethernet by these interfaces
                sp2n11

        Node sp2n12 is connected to network Ethernet by these interfaces
                sp2n12


Connections to network tmscsi
        Node sp2n10 is connected to network tmscsi by these interfaces:
                rvsd_tmscsi0

        Node sp2n12 is connected to network tmscsi by these interfaces:
                rvsd12_tmscsi0
```

Figure 61. Cluster Configuration (3 of 3)

5.7 Integrating the Environment

This step describes how to define resources in HACMP related to the VSD, and also how to create a VSD client node in the same frame.

5.7.1 Defining Resources in HACMP

In order to define a resource following a VSD, HACMP must be configured differently from an ordinary HACMP resource definition.

Its functions will be close to a rotating configuration, with some differences due to the interaction of RVSD and HACMP. In our example, when node 10 is started, it will activate both volume groups (rvsd_vg_node10 and rvsd_vg_node_12) and corresponding VSDs (rvsd_disk_10 and rvsd_disk_12) if node 12 is not started yet. When node 12 is activated, node 10 will then release node 12′s resources, and node 12 will activate its own resources. If we want to define a resource that will follow the VSD, we have to take care of the following points.

First, it must be a cascading resource defined on the same node as the VSD itself. Then it must be set in order to be activated automatically through the first

starting node (as the VSD resource is taken by the first starting node whether it is the owner or not).

The following example will illustrate an HACMP application server following rvsd_disk_10, and an external volume group following rvsd_disk12. The first step is to define a resource group. This is done by issuing the following command:

```
# smit cm_add_grp
```

The name of the resource group will be rvsd_node10.

```
                              Add a Resource Group

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

                                                      [Entry Fields]
* Resource Group Name                                 [rvsd_node10]
* Node Relationship                                   cascading
* Participating Node Names                           [sp2n10 sp2n12]
```

The next step is to define the contents of the resource group. The resources we will add are a volume group and an application server.

HACMP Customization

All the customization must be done with a perfect knowledge of the HACMP chapter of this book. Every step concerning HACMP *will not* be detailed in this part of the book. For more information concerning the definition of the resources, see 3.3, “Installing HACMP on the RISC/6000 SP System” on page 64 and also *High Availability Cluster Multi-Processing 4.1 for AIX Installation Guide*, SC23-2769.

To define the contents of the resource group, type the following command :

```
# smit cm_cfg_res_menu
```

and choose the following highlighted line:

```
                                Cluster Resources

Move cursor to desired item and press Enter.

  Define Resource Groups
  Define Application Servers
  Change/Show Resources for a Resource Group
  Change/Show Run-Time Parameters
  Change/Show Cluster Events
  Change/Show Cluster Lock Manager Resource Allocation
  Show Cluster Resources
  Synchronize Cluster Resources
```

Select the resource group **rvsd_node10**.

```
                    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                        rvsd_node10
  Node Relationship                          cascading
  Participating Node Name                    sp2n10 sp2n12

  Service IP label                          []
  Filesystems                               []
  Filesystems to Export                     []
  Filesystems to NFS mount                  []
  Volume Groups                             [vg_node_10]
  Concurrent Volume groups                  []
  Raw Disk PVIDs                            []
  Application Servers                       [start_vsd_appli]
  Miscellaneous Data                        []


 Inactive Takeover Activated                true
 9333 Disk Fencing Activated                false
 SSA Disk Fencing Activated                 false
```

The volume group defined into the resource must be a shared volume group between node 10 and 12. It must rely on a twin-tailed disk, as a standard HACMP resource does. It must be defined on both nodes 10 and 12, and it must be accessible on both nodes.

Note: vg_node10 has nothing to do with the volume group that holds the VSD (rvsd_vg_node10).

The application server must be defined as a standard HACMP application server (see *High Availability Cluster Multi-Processing 4.1 for AIX Installation Guide*, SC23-2769 for more information). In our example, it is only a dummy file which does not do anything. It could be, for example, the shell used to start or recover the VSD application, as it will be executed in the start process after the activation of the VSD and in the stop process before the deactivation of the VSD. In the case of an ORACLE installation, the application server would execute all the steps to start and stop the database.

For the resource to follow the VSD, the *Inactive Takeover Activated* must be set to true.

This shared volume group can be on the same shared drawer as the VSDs or it can be on a separate rack.

The following picture illustrates the behavior of the HACMP resource and of the RVSD resource.

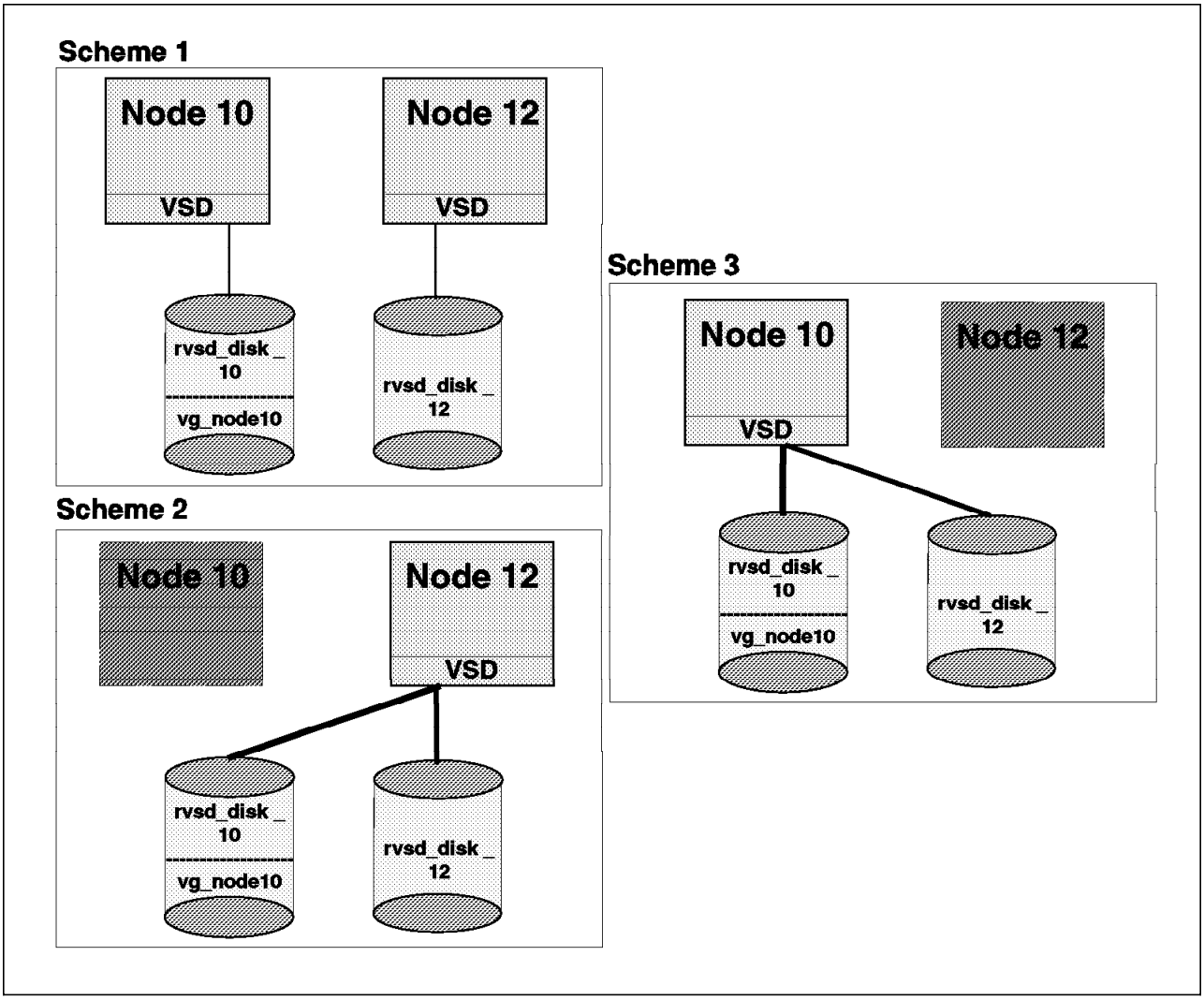


Figure 62. Client and Server Configuration

Explanation of the 3 schemes:

scheme 1: This scheme represents the cluster when both nodes are started. The HACMP resource and the VSD are activated on both nodes, as usual.

scheme 2: This scheme represents the same cluster when node 10 is down. All the resources are accessed through node 12.

scheme 3: This scheme represents the same cluster when node 12 is down. All the resources are accessed through node 10.

Important

It is important to note the behavior of the VSD compared to the HACMP resource. In our example, if we shut down node 10 without stopping HACMP, after a while the VSD is activated on node 12, and not the vg_node_10. This is due to to the fact that the RVSD is not aware of the difference between a node failure and a node shutdown. In order to have the HACMP resource with the VSD, we have to stop HACMP on node 10 with the “takeover” option before the shutdown.

The last thing to do before testing is to define a client in order to test the behavior of VSD client in a failing environment.

The cluster is ready to be tested; refer to 5.7.4, “Testing the VSD” on page 181 for more information.

5.7.2 Defining a Client Node on the Same Frame

This step describes how to define a client on the same frame. This client will be able to access data situated on the two servers nodes as if they were local to this node.

Figure 63 on page 175 illustrates the configuration in our environment.

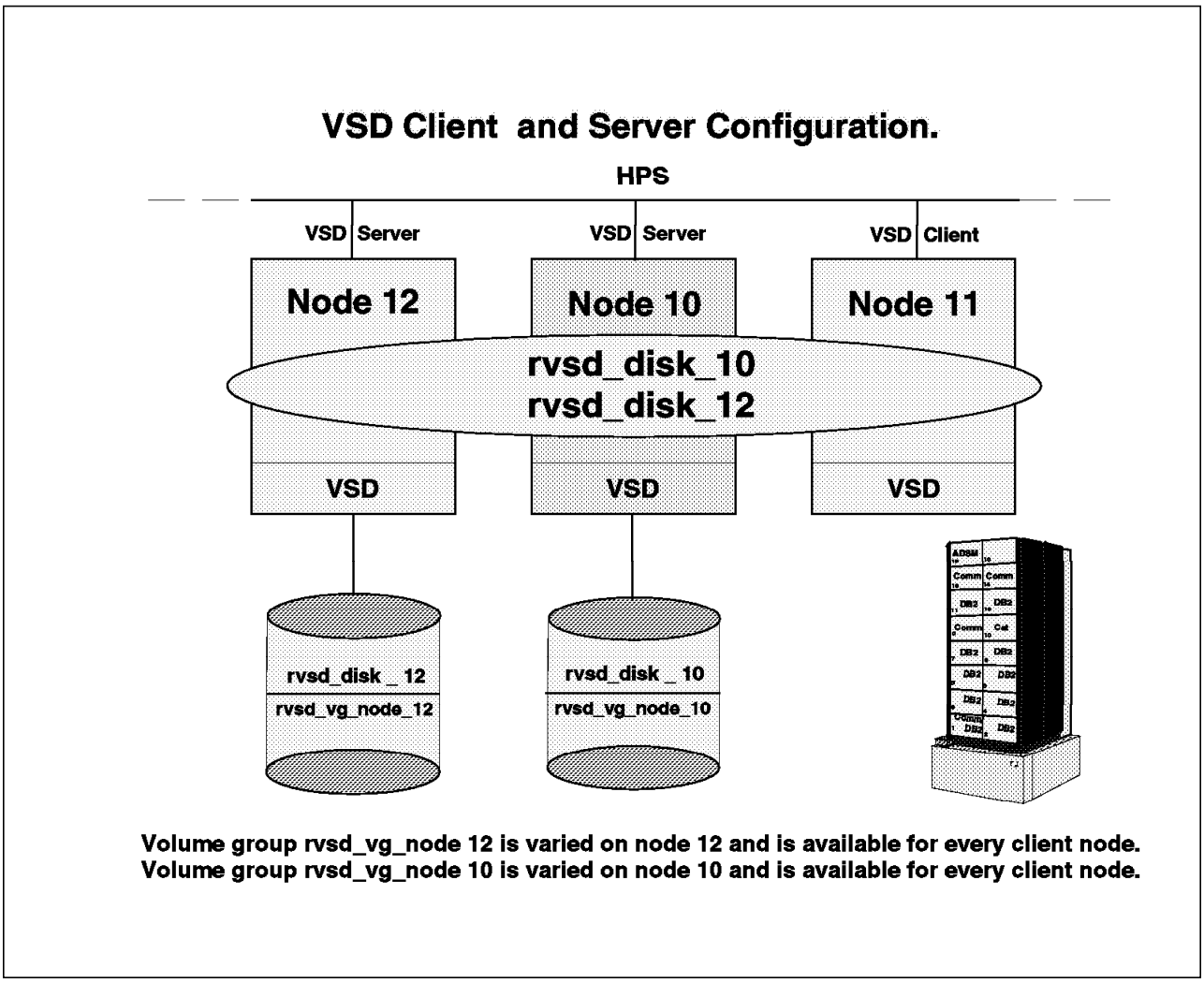


Figure 63. Client and Server Configuration

5.7.2.1 Software Installation

This step is the same as for the server nodes. Refer to 5.5, “Planning for HACMP and RVSD Installation” on page 155.

The simplest way of implementing the client is to install VSD, RVSD, and HACMP to integrate the client node in the cluster.

Once this is done, this node must be defined into the SDR of the control workstation as a VSD node. Use the following procedure to include the node into the SDR.

From the control workstation, enter:

```
# smit vsdnode_dialog
```

```
                        VSD Node Database Information

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

                                                    [Entry Fields]
* Nodes                                              [11]
* Adapter Name for VSD Communications                css0
* Initial Cache Buffer Count                         [64]
* Maximum Cache Buffer Count                         [256]
* VSD Request Count                                  [256]
* Read/Write Request Count                           [48]
* VSD Minimum Buddy Buffer Size                      [4096]
* VSD Maximum Buddy Buffer Size                      [131072]
* VSD Number of Max-sized Buddy Buffers              [9]
```

Node 11 is now included in the VSD. It is necessary to reboot all the VSD nodes to include the modification.

When configuring the VSD, several files are updated on the VSD nodes. They are located under /usr/lpp/csd/vsdfiles:

- Node
- VSD_GVG
- VSD_Table
- VSD_ipaddr

The following is the description of the contents of these files:

Node File

```
10    64    256    256    48    4096    131072  9

11    64    256    256    48    4096    131072  9

12    64    256    256    48    4096    131072  9
```

This file describes all the VSD nodes (client and servers) and the different parameters that may be found on the definition SMIT panel. These parameters are described below in detail.

Explanation of Node File:

10,11,12 These are the node numbers.

64 This is the initial cache buffer count. The VSD device driver implements an optional write-through cache of pinned kernel memory with a block size of 4 KB. The value of 64 results in a 256 KB cache, which is the recommended value.

256 This is the maximum cache buffer value. The number of buffer cache can be increased to this value (using the `ctlvsd` command).

256 This is the number of outstanding VSD requests on the node. The VSD device driver allocates this number of request blocks in pinned kernel memory the first time a VSD is configured. If the number is

too small, the local requests will queue up waiting for a request block to become available. The recommended value is 256 (a block is approximately 76 bytes).

48 This is the maximum amount of pbufs. The read/write requests to the LVM structure will not allocate more than this number of pbufs. The recommended value is 48 (a block is 104 bytes).

4096 This is the minimum buddy buffer size.

131072 This is the maximum buddy buffer size.

9 Maximum buddy buffers. The buddy buffer is the pinned kernel memory allocated when the VSD device driver is loaded for the first time. This is done when the first VSD is configured. This memory is freed when the last VSD is unconfigured. The size of the buddy buffer will affect the number of remote requests the VSD server can handle at one time. Remote requests can queue waiting for a buddy buffer. The statvsd command reports this queuing as buddy buffer shortages. Use this to select the buddy buffer size for your environment. The recommended value is 4 which results in a 256 KB buddy buffer size.

VSD_GVG:

```
rvsd_vg_node10 rvsd_vg_node10        10      12

rvsd_vg_node12 rvsd_vg_node12        12      10
```

This file describes the different volume groups that are shared between the VSD server nodes. The first column is the global group name, and the second column is the volume group name which holds the virtual shared disk (unfortunately the name is the same because we did not define it at creation time). The third column defines the node this VSD belongs to, and the last column is the backup node (when RVSD is defined).

VSD_Table

```
rvsd_disk_10 rvsd_vg_node10  lv_node_10    1     1

rvsd_disk_12 rvsd_vg_node12  lv_node_12    2     1
```

This file details the contents of the shared volume groups (logical volumes and VSD).

VSD_ipaddr

```
10  9.12.6.47

11  9.12.6.48

12  9.12.6.49
```

For each node, this file gives the node's corresponding IP address.

The next step is to define node 11 in the same HACMP cluster as node 10 and node 12.

5.7.2.2 Client Definition in HACMP

Node 11 must now be integrated into the HACMP cluster. We assume the node is node 11, its switch address is sp2sw11, and its Ethernet address is sp2n11.

The first step is to define the new node into HACMP:

```
# smit cm_config_nodes.add
```

```
                              Add Cluster Nodes

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

                                                      [Entry Fields]
* Node Names                                         [sp2n11]
```

The following command will configure node 11's adapters into the HACMP cluster.

```
# smit cm_config_adapters.add
```

```
                                Add an Adapter

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

                                                      [Entry Fields]
* Adapter IP Label                                   [sp2sw11]
* Network Type                                       [hps]
* Network Name                                       [HPSrvsd]
* Network Attribute                                   private
* Adapter Function                                    service
  Adapter Identifier                                 []
  Adapter Hardware Address                           []
  Node Name                                          [sp2n11]
```

The same step must be repeated for the adapter sp2n11. The cluster can now be started and VSD can be tested.

5.7.3 Starting, Testing and Validating the Installation

This chapter is the final test to validate all of the installation. It provides the simplest way of testing the complete installation. At this point, the customer should not have any data in his VSD, as the different test steps will destroy the VSD's contents.

5.7.3.1 Starting the Cluster

The first step is to verify the installation for HACMP.

```
# smit clverify.dialog
```

This command will give you the verification panel for HACMP.

```
                              Verify Environment

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

                                                      [Entry Fields]
  Verify Cluster Topology, Resources, or Both         both
  Error Count                                        []
```

```
                                COMMAND STATUS

Command: OK            stdout: yes           stderr: no

Before command completion, additional instructions may appear below.

[TOP]
Contacting node sp2n10 ...
HACMPcluster ODM on node sp2n10 verified.

Contacting node sp2n12 ...
HACMPcluster ODM on node sp2n12 verified.

Contacting node sp2n10 ...
HACMPnode ODM on node sp2n10 verified.

Contacting node sp2n12 ...

Contacting node sp2n10 ...
[MORE...70]

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

If the result fails, see 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68 to verify that the entire customization was successful.

The result of this command must be OK. If it is, the next step is to start HACMP on all of the three nodes.

```
# smit clstart
```

```
                          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?                         true
  Startup Cluster Lock Services?                        false
  Startup Cluster Information Daemon?                   true
```

This command should be executed on every node.

After HACMP has been launched on all the nodes, it is possible to verify that each node took its own resource after a few seconds:

From node 10, issue this command:

```
# lssrc -g cluster

Subsystem         Group            PID     Status
 clstrmgr         cluster          8004    active
 clsmuxpd         cluster          9310    active
 clinfo           cluster          9578    active
```

The HACMP processes are started. Then issue the following command:

```
# lsvg -o

vg_node_10
rvsd_vg_node10
rootvg
```

We can see that the volume group vg_node_10 has been varied on by HACMP, and the rvsd_vg_node10 has been varied on by RVSD.

Issue the following commands from node 12:

```
# lssrc -g cluster

Subsystem         Group            PID     Status
 clstrmgr         cluster          8004    active
 clsmuxpd         cluster          9310    active
 clinfo           cluster          9578    active

# lsvg -o

rvsd_vg_node12
rootvg
```

We can see on node 12 that the volume group rvsd_vg_node12 has been varied on by RVSD. Issue the following command on node 11:

```
# lssrc -g cluster

Subsystem         Group            PID     Status
 clstrmgr         cluster          8004    active
 clsmuxpd         cluster          9310    active
 clinfo           cluster          9578    active
```

We can see that HACMP processes are running on node 11.

Starting HACMP launches all the processes for VSD recovery and loads the driver for VSD.

5.7.4 Testing the VSD

To test the VSD, we will use the test routine provided with VSD (vsdvts) located under /usr/lpp/csd/bin.

The first step is to test on node 11 that the VSD works.

```
# cd /usr/lpp/csd/bin
# vsdvts rvsd_disk_10

vsdvts: Step 1: Writing file /unix to VSD rvsd_10.
dd if=/unix of=/dev/rrvsd_disk_10 count=256 bs=4096 seek=1
256+0 records in.
256+0 records out.
vsdvts: Step 1 Successful!
vsdvts: Step 2: Reading data back from the VSD.
dd of=/tmp/vsdvts.13864 if=/dev/rrvsd_disk_10 count=256 bs=4096 skip=1
256+0 records in.
256+0 records out.
vsdvts: Step 2 Successful!
vsdvts: Step 3: Verifying data read from the VSD.
dd if=/unix count=256 bs=4096 | cmp -s - /tmp/vsdvts.13864
256+0 records in.
256+0 records out.
vsdvts: Step 3 Successful!


VSD Verification Test Suite Successful!

# vsdvts rvsd_disk_12

vsdvts: Step 1: Writing file /unix to VSD rvsd_12.
dd if=/unix of=/dev/rrvsd_disk_12 count=256 bs=4096 seek=1
256+0 records in.
256+0 records out.
vsdvts: Step 1 Successful!
vsdvts: Step 2: Reading data back from the VSD.
dd of=/tmp/vsdvts.13938 if=/dev/rrvsd_disk_12 count=256 bs=4096 skip=
256+0 records in.
256+0 records out.
```

```
vsdvts: Step 2 Successful!
vsdvts: Step 3: Verifying data read from the VSD.
dd if=/unix count=256 bs=4096 ] cmp -s - /tmp/vsdvts.13938
256+0 records in.
256+0 records out.
vsdvts: Step 3 Successful!

   VSD Verification Test Suite Successful!
```

The VSD works correctly.

The next step will be to validate that RVSD and HACMP work correctly.

5.7.5 Validating the Solution

This step will simulate a crash on node 10 to see if node 12 takes over the resources of node 10, as illustrated in Figure 64.

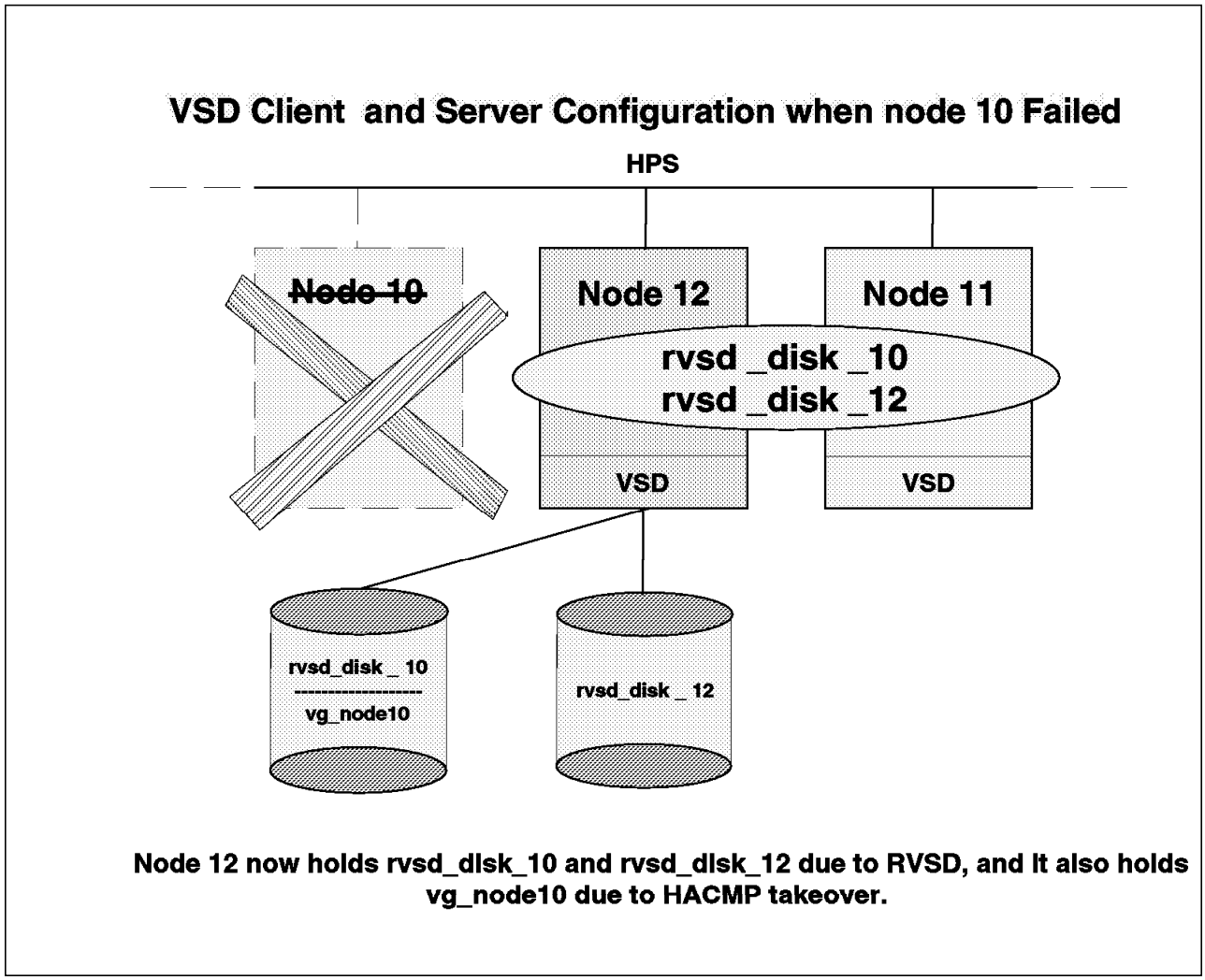


Figure 64. Takeover Configuration

There are several methods for doing this test. One of the simplest ways is to stop HACMP with the takeover option, in order to tell the backup node (node 12)

to activate the resources. However, we will turn off the node to simulate a power failure on the node.

The first thing to do is to initiate read and write on the client node. This can be done on the command line of node 10:

```
# cd  /usr/lpp/csd/bin
# for i in 1 2 3 4 5 6 7 8 9
> do
> vsdvts rvsd_disk_10
> done
```

This command will process several tests on rvsd_disk_10. The next step is to go on the control workstation and turn off node 10. This will result in all the reads and writes hanging on node 11 for a few seconds while node 12 is taking over node 10. After a while everything should come back to normal, and the tests should complete successfully.

Validation will be performed on node 12 by using the following command:

```
# lsvg -o

vg_node_10
rvsd_vg_node10
rvsd_vg_node12
rootvg
```

We can see that all resources are now allocated to node 12. Node 12 is now the VSD server for rvsd_disk_10 and rvsd_disk_12. It also has the shared volume group vg_node_10, which is an HACMP resource. The resource follows the VSD.

The resource integrated into HACMP in the previous step could have been any kind of resource usually defined into a cluster. For example, it could have been a complex IP takeover or a filesystem takeover.

The next step is to recover node 10. This means rebooting it and launching HACMP. When node 10 starts, HACMP on node 12 will release node 10′s resources so node 10 can take them back.

Chapter 6. Implementing LoadLeveler for High Availability

This chapter describes some of the built-in capabilities of LoadLeveler for high availability. It also discusses how LoadLeveler's overall system availability can be further enhanced with products such as:

High Availability Cluster Multi-Processing

High Availability Control Workstation

LoadLeveler is a workload management system that allows users to submit, schedule, and execute jobs to a pool of resources.

The primary objective of LoadLeveler is to support users quickly and efficiently using all available resources. LoadLeveler is based on an extremely flexible design which dynamically matches job requirements with available machine resources. These resources may come from dedicated or spare capacity of various platforms and architectures, including UNIX, user workstations and enterprise systems.

LoadLeveler runs as a set of daemons on each of its client machines. A group of client machines reporting to a central manager machine is referred to as a LoadLeveler cluster. A Loadleveler cluster is defined by a configuration file. Client machines, or nodes, perform one or more functions for the Loadleveler cluster:

- Scheduling jobs
- Executing jobs
- Managing jobs

The role a client machine plays depends on which LoadLeveler daemons are configured on it. As a whole, the cluster provides the capability to build, submit, execute, and manage serial and parallel batch jobs.

6.1 Roles of LoadLeveler Machines

This section describes the various roles of LoadLeveler machines. Multiple scheduling and executing machines can be defined but only one central manager can be named in the configuration. However, it is possible to have several alternate central managers.

6.1.1 Central Manager

There is only one active central manager in a LoadLeveler cluster. The central manager is the resource coordinator for the cluster. It knows about all the machines in its cluster: their memory sizes, disk sizes, architectures, and when they are available for work. It also routinely receives status from all the nodes on what jobs are still running or are completed. The central manager finds matching resources for the end user's job requests and dispatches jobs to the appropriate machines to balance the workload or maximize throughput.

6.1.2 Scheduling Machine

In the LoadLeveler machine pool, any node can be designated to function as the scheduling node for itself or for other nodes. The scheduling function is handled by the LoadLeveler daemon called *schedd*. This schedd is responsible for a job until the job completes.

The scheduling machine notifies the central manager that it has a job to be considered for dispatch. The central manager decides on the scheduling request and sends its decision back to the *schedd* where the job was submitted. When the schedd receives the dispatch, it connects with another LoadLeveler daemon called *startd* on the target execution node, which can be local or remote to the scheduling node.

6.1.3 Execution Machine

In the Loadleveler cluster, any node can be designated to function as an execution node. One of the LoadLeveler daemons that run on the execution machines is called *startd*. The startd's function is to receive jobs and spawn a *starter* for each job. The *starter* process loads and monitors the job execution.

During the job execution time, the starter process sends information to the local startd, which reports to the schedd responsible for the job, which in turn reports to the central manager. The central manager then updates its in-memory machine table with the latest status.

6.2 LoadLeveler Functional Flow

Figure 65 on page 187 illustrates how LoadLeveler manages and controls a job submitted by a user through the sched to both the central manager and the execution machine.

1. A user submits a job to a scheduling machine in the LoadLeveler cluster.
2. The scheduling machine maintains a copy of the job description information, logs the submission and spools it into the job list on the local disk. The scheduling node also forwards the job description information to the central manager.
3. The central manager decides on the disposition of the job. If the job is able to be dispatched, the central manager sends authorization to the scheduling machine to begin taking steps to run the job. The central manager maintains knowledge of machine, job, and user states.
4. After receiving authorization from the central manager, the scheduling machine forwards the job to the execution machine that the central manager has directed it to use (The execution node can be local or remote). After the execution machine is contacted by the scheduling machine, the execution machine spawns a process called *starter*. The scheduling machine passes the job information and the executable to the starter. The execution machine then runs the job.
5. The scheduling machine notifies the central manager that the job has been started and the central manager marks it as Running.
6. When the job completes, the execution machine notifies the scheduling machine. The scheduling machine updates its job list, and notifies the central manager, which updates its machine and job status.

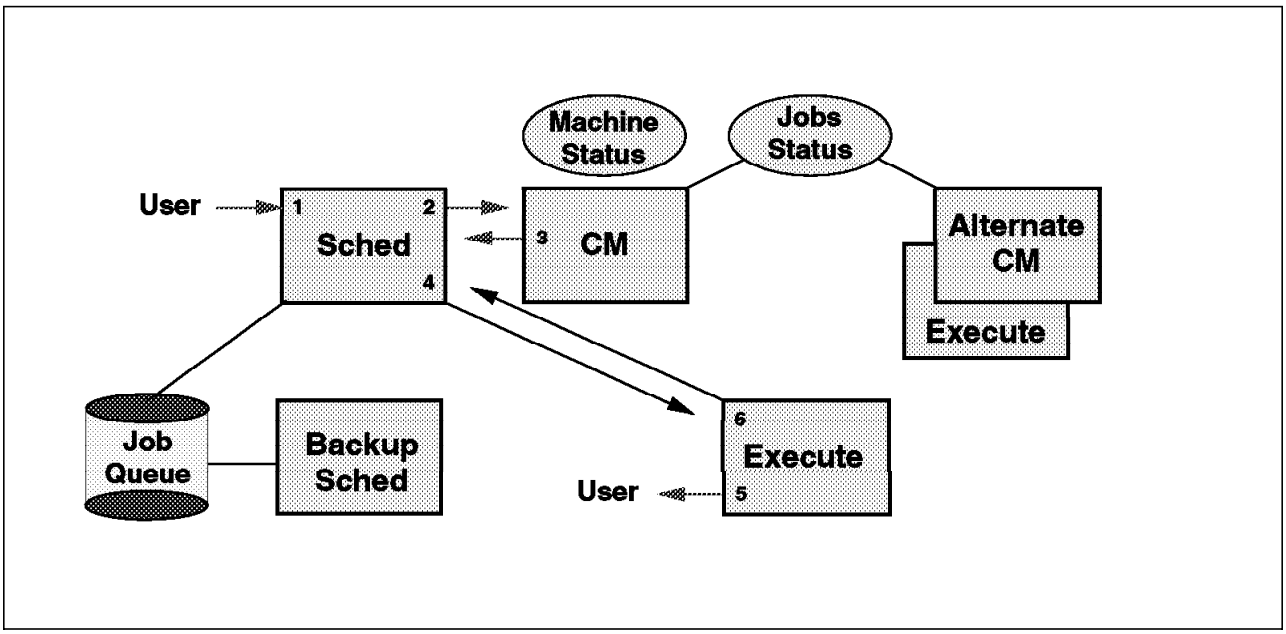


Figure 65. Functional Flow of LoadLeveler

6.3 Potential Single Points of Failure

To implement high availability for LoadLeveler, we need to identify which components and subsystems may be potential single points of failure.

By definition, a single point of failure exists when a critical cluster function is provided by a single component. If that component fails, the cluster has no other way to provide that function, and essential services become unavailable.

6.3.1 What Happens When Network Goes Down

Access to the global configuration and the administration files are of critical importance to LoadLeveler. Without these files, LoadLeveler cannot start up any of the nodes. As such, the network, which provides the access to this resource, is a potential single point of failure during the start up, or when new nodes or late-comers join the LoadLeveler cluster.

Access to the nameserver, home directories, log, spool and execution directories are of critical importance during operations as well. In our test configuration, these critical files are shared and remotely mounted from the control workstation. For performance considerations, SP nodes may be used as the LoadLeveler AMD file server instead of using the control workstation. In either case, the network is of critical importance and is identified here as a potential single point of failure.

6.3.2 What Happens When the Central Manager Goes Down

The central manager is the central point of control for LoadLeveler. The central manager makes all the decisions regarding which jobs are scheduled on which nodes. LoadLeveler provides the capability to define an alternate central manager and have it take over the role of the primary central manager in case of failure.

In the event of a central manager failure without an alternate central manager defined, jobs which have started run to completion without loss of information but their status (running, pending, or completed) are not reportable to the users. New jobs will be accepted by scheduling nodes and later forwarded to the central manager when it returns, or when an alternate takes over.

Only one central manager can be running in the cluster at any one time. The active central manager needs to have the following key subsystems or daemons running:

- Master daemon
- Negotiator daemon

Beginning with LoadLeveler Version 1 Release 2, the central manager is no longer a potential single point of failure.

6.3.3 What Happens When Scheduling Node Goes Down

The scheduling machine maintains information about all jobs that have been submitted to that machine. The job information on one scheduling machine is not normally shared or accessed by other scheduling machines in the LoadLeveler cluster. The scheduling machines operate independently and in the event of a scheduling machine failure, the job information that resides on the failed scheduling machine will be temporarily unavailable (but not lost).

Jobs waiting to be scheduled will not be considered for execution during this time. For this reason, it may be of critical importance to have the scheduling machine re-established as quickly as possible.

The scheduling machine maintains the job list and stores it on disk. In our example of high availability configuration, the necessary local files and directories are placed on external shared disk storage, which makes them available to a backup scheduling machine in the event of scheduler node failure.

6.3.4 What Happens When Execution Node Fails

If an execution node fails, jobs running on the node fail and require restart when the node is restored. Jobs will start from the beginning or from the last checkpoint, if checkpointing was selected or coded in the application. The establishment of a backup execution node would provide immediate the capability to restart the job in a more timely fashion. With or without a backup node, jobs and their job information and checkpoints are not lost in the event of node failure, as long as their disks are still accessible using the appropriate cabling and disk techniques.

Both the scheduling node and the execution nodes maintain persistent information about the state of their jobs. Persistence, in this case, means that both nodes use a protocol that ensures that the state information is kept on disk by at least one of them. In the event of a failure, the state can be recovered from somebody′s disk. Neither the scheduling node nor the execution node discard the job information until it is passed onto and accepted by the other node. In our configuration, the information is stored in shared disks of the control workstation.

6.3.5 Recap of Failure Scenarios

Based on the above analysis, the network and the control workstation are the key points of failure which need to be addressed for high availability in a LoadLeveler environment.

6.4 LoadLeveler High Availability Solution Matrix

Table 10. How LoadLeveler and HACMP Address Potential Single Points of Failure

Critical LoadLeveler Resource	Potential Single Point of Failure	High Availability Solution
• /u/loadl Filesystem • Configuration Files • Host Name Server (if used)	Network /u/loadl Network File Server	HACMP HACWS
Central Manager	Network /u/loadl Network File Server Node	HACMP HACWS LoadLeveler
Scheduler	Network /u/loadl Network File Server Node	HACMP (+Custom Scripts) HACWS HACMP
Execution Node	Network /u/loadl Network File Server Node	HACMP (+Custom Scripts) HACWS HACMP

6.4.1 Configure LoadLeveler for High Availability

LoadLeveler has some built-in features which can be utilized to gain some level of high availability:

- LoadLeveler's architecture allows for the placement of LoadLeveler's home and local directories and file placements in a shared high availability storage filesystem and network file server.
- LoadLeveler has the capability to assign an alternate central manager to back up the primary central manager. Use of this alternate central manager is illustrated in 6.6, "Enhancing LoadLeveler's System Availability with HACMP" on page 199.
- LoadLeveler daemons maintain consistent states of all jobs. They use a protocol to ensure that the state of all jobs is consistent across LoadLeveler.
- LoadLeveler can be configured to use the High Performance Switch for IP network communications on the SP.

6.4.2 HACMP to Address Network Availability

On the RISC System/6000 SP, the High Performance Switch network could be considered a single point of failure if no other connectivity to the outside is available on the SP nodes.

The High Performance Switch has built in recovery and redundancy to minimize component failure. This makes it a reliable part of the RISC System/6000 SP System. In the rare occurrence that the entire switch fails, an external network could be utilized to recover the workload of the switch.

Section 3.7, “Implementing High Availability for HiPS Network Failure” on page 105 describes how High Availability Cluster Multi-Processing can be implemented to provide such a backup network for the HiPS switch network. This solution illustrates how the elimination of this network as a single point of failure provides higher availability for LoadLeveler.

The High Performance Switch uses IP address aliasing and HACMP IP address takeover to swap the IP addresses from the switch to the external network in the event of failure. This requires at least another network and adapter.

In our sample configuration, an FDDI backup network was implemented with HACMP to provide this network with a high availability environment. Sample HACMP custom scripts are provided for reference and customization.

6.4.3 HACMP/HACWS to Address File Server Availability

In our sample configuration, the file server is protected by using HACMP and the High Availability Control Workstation (HACWS). There are some advantages to using the control workstation as the file server. The main advantage is the use of AMD. It provides for central administration of the loadl user ID and passwords. The other advantage for using AMD with NFS is the its automounting feature. We used AMD for the /u/loadl filesystem.

The /u/loadl shared filesystem was made highly available using the High Availability Control Workstation (HACWS). Automount Daemon (AMD) and NFS provided the access to the shared filesystem. As discussed earlier placement of the home and LoadLeveler filesystems in a shared filesystem was necessary to achieve maximum flexibility in the initialization and start of any LoadLeveler configuration on any backup node.

6.4.4 HACMP to Address SP Node Availability

Immediate restoration of LoadLeveler scheduling and execution nodes is made possible using High Availability Cluster Multi-Processing for performing IP address takeover, invoking scripts for hostnaming, and making ARP table updates to client machines. Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68, describes how High Availability Cluster Multi-Processing was implemented to provide a backup node.

6.5 Configuring LoadLeveler for High Availability

Our first objective is to take advantage of LoadLeveler′s client/server architecture and configuration file system to achieve high availability.

Configuring LoadLeveler to use a shared filesystem is perhaps the most important part of this strategy. By placing all the home and local directories under one shared filesystem, we are able to access all the configuration files and the log and spool directories necessary to recover and restart a failed LoadLeveler node anywhere in the pool.

The following discussion is not to revisit the standard LoadLeveler install process, but to highlight the strategies taken by this redbook in configuring LoadLeveler for high availability.

The steps for installing Loadleveler are:

Step 1. Plan the LoadLeveler installation.

Step 2. Create LoadLeveler user ID(s) and loadl group.

Step 3. Install the LoadLeveler software.

Step 4. Run the installation script.

Step 5. Update configuration files.

Step 6. Repeat the installation process for each node in cluster.

6.5.1 Step 1: Plan the LoadLeveler Installation

Planning the installation for high availability requires making the following configuration decisions:

- Which machine will act as the central manager
- Which machines will be backup central managers
- Where to locate home and local directories for the LoadLeveler user ID
- Which directories to share

As shown in Figure 66 on page 192, Nodes 4, 5 and 6 were selected to illustrate how LoadLeveler can be configured for high availability and how High Availability Cluster Multi-Processing and High Availability Control Workstation were used to enhance its availability.

For LoadLeveler SP Clusters Only

Note that this high availability solution is for SP environments only, which means it is not for mixed platform or mixed architecture LoadLeveler clusters. It may not always be desirable to have the LoadLeveler daemons communicate over the switch. You need to evaluate the network traffic in your system to determine if your network can accommodate LoadLeveler daemons communication over the switch. If a switch is not used in your SP, you can still implement this HACMP solution over ethernet, token ring or FDDI networks.

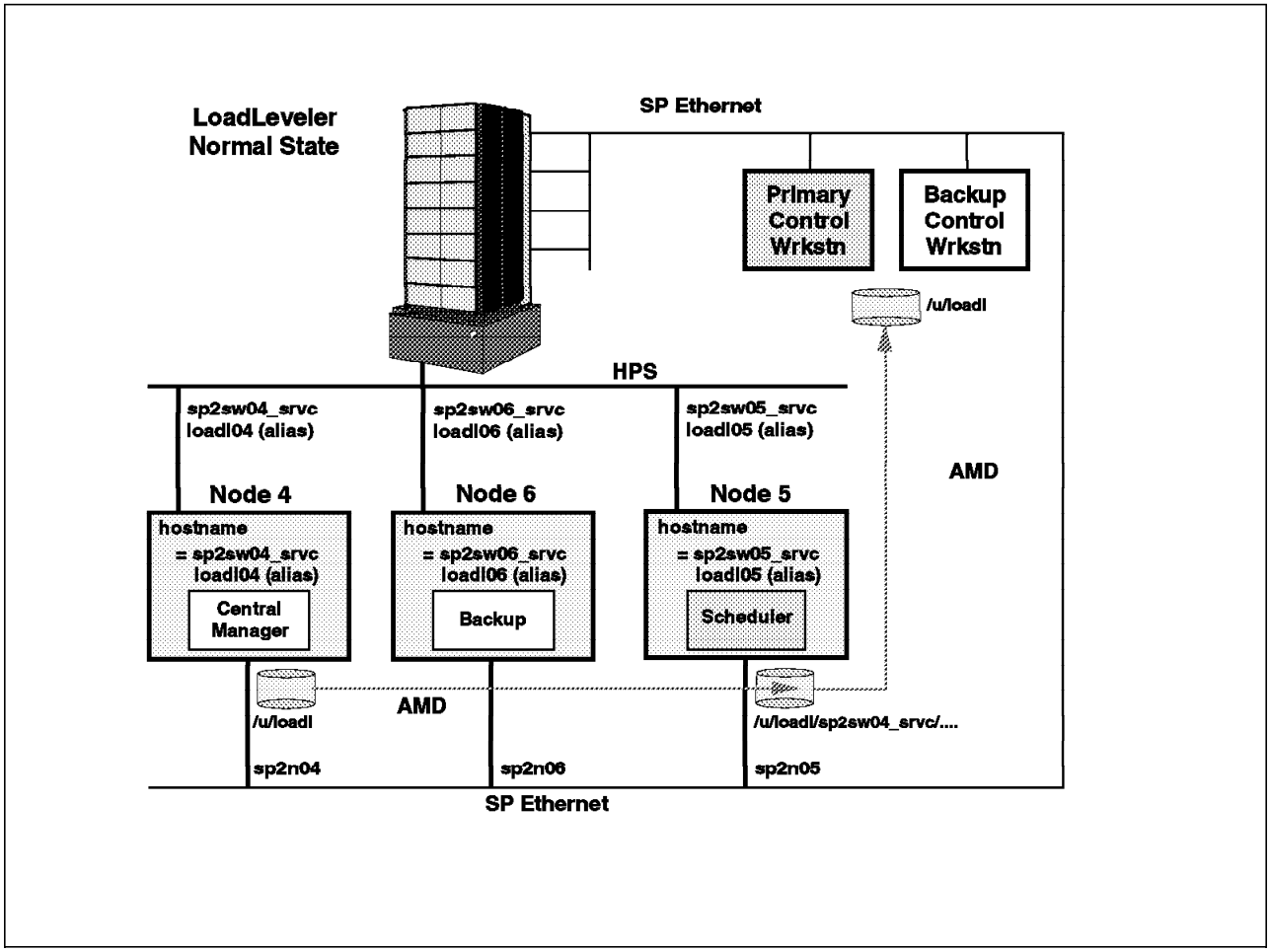


Figure 66. LoadLeveler Cluster Configuration

In this example, Node 4 was designated as the central manager. Node 5 was the scheduler node and Node 6 was the backup node. The RISC/6000 SP control workstation was used as the network file server for the LoadLeveler files and home directories.

In general, the choice of which machines to use as central manager and backup nodes, from a high availability point of view, depends on your environment.

The central manager can be any machine in the pool. In selecting one, consider the machine's current workload and network access. Remember that no new work can be performed while the central manager is down, and no queries can be made about any of the running jobs without the central manager. Fortunately, LoadLeveler Version 1 Release 2 provides for the designation of alternate central managers. The selected alternate central managers should be members of the LoadLeveler pool.

If node isolation is a potential problem with IP network failure, consider using HACMP instead of the built-in alternate central manager. HACMP provides a non-IP network path for establishing node status, which can protect against having two central managers configured (one on each network-partitioned segment). Using HACMP, only one central manager can be activated.

Note that in our sample configuration, we set the local directory to be the same as the home directory. We manually created the hostname-specific log and spool directories.

This approach provided an open strategy: that of being able to initialize a newly joining node to whatever configuration is necessary for the situation.

A set of very simple scripts are necessary. On scheduler takeover, HACMP invokes a script which renames the hostname of the node to the failing scheduler's hostname. This enables LoadLeveler to restart the failing scheduling node and provide end users with a status of their jobs.

Administratively, there is one requirement. System administrators need to keep HACMP synchronized with the LoadLeveler node groupings. Each backup node needs to know which set of eight nodes it will back up. This relationship is kept in the form of HACMP resource groups. Each backup node will have up to eight resource groups defined, and each resource group defines which primary node the backup is paired with for take over.

Remember that the basic design of HACMP allows for only single failover events -- meaning occurrences of two or more node failures cannot be accommodated until the first failure is repaired and the back up node is ready.

The following LoadLeveler file placements were used in our configuration:

Table 11. LoadLeveler File Placement Used

LoadLeveler Element	**Central Manager**	**Scheduler Node**	**Execution Node**
Home Directory	/u/loadl (shared)	/u/loadl (shared)	/u/loadl(shared)
Local Directory	/u/loadl (same as home)	/u/loadl (same as home)	/u/loadl (same as home)
Release Directory	/usr/lpp/LoadL/nfs (shared)	/usr/lpp/LoadL/nfs (shared)	/usr/lpp/LoadL/nfs (shared)
Global Configuration File	/u/loadl/LoadL_config	/u/loadl/LoadL_config	/u/loadl/LoadL_config
Local Configuration File	/u/loadl/${hostname} ./LoadL_config.local	/u/loadl/${hostname} ./LoadL_config.local	/u/loadl/${hostname} ./LoadL_config.local
Administration File	/u/loadl/LoadL_admin (shared)	/u/loadl/LoadL_admin (shared)	/u/loadl/LoadL_admin (shared)
Log Directory	/u/loadl/${hostname}/log	/u/loadl/${hostname}/log	/u/loadl/${hostname}/log
Spool Directory	/u/loadl/${hostname}/spool	/u/loadl/${hostname}/spool	/u/loadl/${hostname}/spool
Execute Directory	/u/loadl/${hostname}/execute	/u/loadl/${hostname}/execute	/u/loadl/${hostname}/execute

Note: In the above table under the column captioned Local Configuration File, the *./LoadL_config.local* is relative to the /u/load/${hostname} directory. For example the first column in the role under the heading of central manager should be /u/load/${hostname}/LoadL_config.local and this should be the same for the rest of this column.

6.5.2 Step 2: Set Up loadl User and Group IDs

LoadLeveler uses a specific user and group named *loadl* for the access and administration of files and directories. To ensure the same user and group ID across the pool, the automount daemon (AMD) was used together with the network file system (NFS) and the Software Update Protocol (SUPPER) on the control workstation to administer these user and group IDs. From the control workstation, setup the loadl user and group IDs as follows:

1. Verify that AMD is configured for your system, using the following command:

   ```
   splstdata -e | grep amd_config
   ```

   The expected response is:

   ```
   amd_config=true
   ```

   If this response is not received, issue the following command:

   ```
   spsitenv amd_config=true
   ```

   Then retry the initial command,

   ```
   splstdata -e | grep amd_config
   ```

   for a true response.

2. Create an external AIX shared filesystem called /home/<CWS_hostname> where CWS_hostname is *sp2cw0*. In our configuration, an external shared volume group named *spadatavg* was created to allow the RISC/6000 SP system management */spdata* and the */home/sp2cw0* filesystems to be accessible from either the primary or backup control workstation.

3. Create AIX loadl group name and ID. This is a standard AIX system management procedure involving the use of the SMIT panels.

   From SMIT main menu select the following options:

   => Security and Users
   => Groups
   => Add Group

```
                                Add a Group

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

                                              Entry Fields
* Group NAME                                   loadl
  ADMINISTRATIVE group?                        false
  Group ID                                     300
  USER list                                    loadl
  ADMINISTRATOR list
```

4. Create a 9076 User ID for loadl by using the following SMIT options:

   => 9076 SP System Management
   => 9076 SP Users
   => Add User

```
                    Add 9076 User

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

                                          Entry Fields
* User NAME                               loadl
  User ID                                 300
  LOGIN user?
  PRIMARY group                           300
  Secondary GROUPS                        loadl
  HOME directory
  Initial PROGRAM                         /bin/ksh
  User INFORMATION
```

Leave the HOME directory blank. The system will create it under the /home/<cws_hostname> directory for AMD to mount.

5. Add /home/<cws hostname> to the NFS export file list.

```
                  Add Directory to Export List

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

                                          Entry Fields
* PATHNAME of Directory to Export          /home/sp2cw0
* MODE to export directory                 read-write
  HOSTS & NETGROUPS allowed client access
  Anonymous UID                           -2
  HOSTS allowed root access               sp2n05,sp2n06,sp2n04
  HOSTNAME list. If exported read-mostly
  Use SECURE OPTION?                       no
* CHANGE export now, system restart or both  both
  PATHNAME of alternate Exports file
```

6. Verify that /etc/amd/amd-maps/amd.u now includes loadl filesystem. Find the following stanza in the amd.u file. This stanza was automatically created by SMIT during step three.

```
loadl   host==sp2cw0;type:=link;fs:=/home/sp2cw0 \
                host!=sp2cw0;type:=nfs;rhost:=sp2cw0;rfs:=/home/sp2cw0
```

7. Create a new AMD map if necessary. See *RISC/6000 SP IBM RISC System /6000 Scalable POWERparallel System, Administration Guide*, GC23-3897 for creating new AMD maps.

8. If new maps were created, refresh AMD and update the file collection using *dsh supper update user.admin* and then */etc/amd/refresh_amd* from the control workstation.

6.5.3 Step 3: Install LoadLeveler Software

The installation was performed as *root* from the */usr/lpp/LoadL* directory. The entire LoadLeveler product was installed on each node. LoadLeveler provides the option to configure *Submit-Only* workstations and not have the entire product installed. The standard AIX installp process was used.

The LoadLeveler product is comprised of the following components:

- LoadLeveler (approximate disk space requirement is 13 MB)

- Submit-only LoadLeveler (2 MB)
- Interactive session support (512 KB)
- InfoExplorer documentation (3 MB)
- PostScript documentation (2.5 MB)
- Allocation for local directories (typically 15MB)

6.5.4 Step 4: Run Installation Script

LoadLeveler provides an installation script called *llinit* to initialize a new machine as a member of the LoadLeveler hardware resource pool.

Before running this script, log on as root user and create subdirectories for each LoadLeveler machine in your SP cluster. (In our configuration, we logged on as root on our control workstation which is our /u/loadl file server.)

```
cd /u/loadl
mkdir /u/loadl/sp2sw04_srvc
mkdir /u/loadl/sp2sw05_srvc
mkdir ......
```

From each machine subdirectory execute the llinit script. This script will perform the following:

1. Creates the log, execute and spool directories under that subdirectory (/u/loadl/$(hostname)) and the administration and local configuration files under loadl's home directory (/u/loadl)
2. Creates the LoadLeveler spool and execute directories
3. Creates the LoadLeveler log subdirectory in that machine subdirectory
4. Copies the LoadL_admin and LoadL_config files from the release directory to the home directory (/u/loadl)
5. Copies the LoadL_config.local file from the release directory into the local directory (/u/loadl)
6. Creates the symbolic links from the loadl home directory to the spool, execute and log subdirectories and the LoadL_config.local file in the local directory (if home and local directories are not identical)
7. Creates symbolic links from the home directory to the bin, lib, man, samples, and include subdirectories in the release directory
8. Creates symbolic links for the llcc and llxlf commands in the release directory
9. Updates the LoadL_config file with the release directory name
10. Updates the LoadL_admin file with the central manager hostname

Manually delete the symbolic links created by the llinit script for the loadl home directory and spool and log directories and the LoadL_config.local file. We will hardcode and direct LoadLeveler to use our new log, spool and execute directories, and local files in the global configuration file LoadL_config, as shown in the next step.

6.5.5 Step 5: Customizing the Configuration Files

The global: */u/loadl/LoadL_config* file required the following changes to reflect our high availability strategy. Note the use of the LoadLeveler variable $(hostname) to define the path to the log and spool directories. And note the location of the local configuration file.

```
/u/loadl/LoadL_config file

RELEASEDIR      = /usr/lpp/LoadL/nfs
LOCAL_CONFIG    = /u/loadl/$(hostname)/LoadL_config.local
ADMIN_FILE      = /u/loadl/LoadL_admin
LOG             = /u/loadl/$(hostname)/log
SPOOL           = /u/loadl/$(hostname)/spool
EXECUTE         = /u/loadl/$(hostname)/execute
HISTORY         = /u/loadl/$(hostname)/spool/history
BIN             = /usr/lpp/LoadL/nfs/bin
LIB             = /usr/lpp/LoadL/nfs/lib
```

Part of the high availability strategy would be to control the number of scheduler nodes per HACMP back up node. It is necessary to configure a new HACMP resource group for each pair of backup relationships. With HACMP we also need to avoid exceeding the current maximum cluster size of eight nodes.

The following keyword should be included on each scheduler node's local configuration file.

`/u/loadl/$(hostname)/LoadL_config.local`

```
SCHEDD_RUNS_HERE        =       True
```

For our LoadLeveler sample configuration, Node 5's local configuration file

`/u/loadl/sp2sw05_srvc/LoadL_config.local`

should have:

```
SCHEDD_RUNS_HERE        =       True
```

6.5.6 Step 6: Verifying the Configuration

The LoadLeveler installation was verified by:

1. Starting LoadLeveler from Node 4, using the global start command:

   ```
   login: loadl
   llctl -g start
   ```

2. Verify that Node 4 is running LoadLeveler as the central manager. The negotiator and collector daemons should be running on Node 4. Use the *ps* command to verify this, as shown below:

   ```
   ps -efa | grep loadl
   ```

   The ps response on Node 4 should include the following line items:

   ```
   sp2n04_srvc-root /u/loadl -> ps -efa ] grep loadl
   loadl 16766     1   0 21:05:25      -  0:00 /usr/lpp/LoadL/nfs/bin/Loader
   loadl 17282 16766   0 21:05:25      -  0:00 LoadL_negotiator -f
   +)
   ```

3. Verify that sp2sw04_srvc (Node4) and sp2sw05_srvc (Node5) are running LoadLeveler. The sp2sw04_srvc should be the central manager and sp2sw05_srvc should be the scheduler (should have schedd available).

Since these are the only LoadLeveler nodes defined, the status should show all nodes being present.

`llstatus`

```
sp2sw0-loadl /u/loadl -> llstatus
Name                      Schedd  InQ Act Startd Run LdAvg Idle Arch

sp2sw04_srvc.itsc.pok.ibm Down      0   0 Down     0 0.00     0 R6000

sp2sw05_srvc.itsc.pok.ibm Avail    10   0 Idle     0 0.00    22 R6000


R6000/AIX41            2 machines  10 jobs   0 running
                       2 machines  10 jobs   0 running

The Central Manager is defined on sp2sw04_srvc.itsc.pok.ibm.com

All machines on the machine_list are present
```

4. Verify that LoadLeveler's backup system for the central manager works by shutting down LoadLeveler on Node 4 and confirming that Node 5 (sp2sw05_srvc) takes over as the central manager.

   To shutdown the central manager use:

   `llctl -h <hostname of primary central manager> stop`

   In our configuration, Node 4 is the central manager and the primary central manager shutdown was done as follows (on Node 4, logged on as loadl):

   `llctl -h sp2sw04_srvc stop`

   After approximately 2-3 minutes node 5 (sp2sw05_srvc) should take over as the central manager (actual time will depend on what you have set the central manager timeout and interval periods to be in the LoadL_config file).

   Re-issue the status command:

   `llstatus`

   The response screen should look as follows:

```
sp2sw0-root /u/loadl -> llstatus
Name                      Schedd  InQ Act Startd Run LdAvg Idle Arch

sp2sw05_srvc.itsc.pok.ibm Avail    10   0 Idle     0 0.12    27 R6000


R6000/AIX41            1 machines  10 jobs   0 running
                       1 machines  10 jobs   0 running

The Central Manager is defined on sp2sw04_srvc.itsc.pok.ibm.com, but is
unusable.  Alternate Central Manager is serving from
sp2sw05_srvc.itsc.pok.ibm.com

The following machine is absent...
sp2sw04_srvc.itsc.pok.ibm.com
```

6.6 Enhancing LoadLeveler′s System Availability with HACMP

Implementing HACMP/HACWS in conjunction with LoadLeveler provides recoverability in the event of network failure, scheduling and execution node failures, or control workstation failure. It also allows LoadLeveler job information and status to become immediately available to users even when failures occur.

6.6.1 Overview of HACMP Cluster Design

The sample LoadLeveler HACMP cluster consists of three RISC/6000 SP nodes as shown in Figure 67 on page 200. Node 4 is the central manager. Node 5 is the scheduler node to be made highly available and Node 6 is the backup node. In this configuration, Node 6 is shown backing up only one primary, Node 5. When implemented, up to six other scheduler nodes can be backed up by this single node. If more than seven scheduler nodes are to be backed up, multiple HACMP clusters of seven schedulers and one backup can be configured in similar fashion.

The following considerations were made in the design of our sample LoadLeveler high availability configuration. Detailed steps to configure the High Availability Cluster Multi-Processing software are described and illustrated in 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68.

LoadLeveler High Availability Design Objectives

Restart the failed LoadLeveler Schedule node on a backup node with the same hostname and IP address as defined in the LoadLeveler configuration files.

Configuring IP Address Takeover

IP address takeover will be necessary in order for the backup node to assume the role of the failed scheduler. Job notification messages for updating job queues will be routed to the IP address associated with the scheduler name (hostname) in the /etc/hosts table.

Flushing ARP cache of other LoadLeveler nodes

Since hardware address takeover is not possible on the HiPS, it will be necessary to flush the ARP cache of the original scheduling node′s execution nodes, central manager, and submit-only nodes. For a detailed procedure on flushing the ARP cache of client systems, see 3.7.5, “Client Considerations” on page 127. An *arp -d <failing LoadLeveler hostname>* is executed on every client LoadLeveler machine to accomplish this requirement.

Setting Hostname

The hostname should match the name given to the scheduler in the LoadL_admin configuration file. This will initialize the backup and allow it to access the specific hostname subdirectory where the log and spool files reside. The log and spool directories contain the job queues and checkpoint files for both pending and running jobs.

Start/Stop LoadLeveler on the Nodes

It will be necessary for HACMP to reset the hostname of the backup node to the failing node′s hostname. It will also be required to start LoadLeveler when the failover occurs and to stop LoadLeveler on

the backup once the original scheduler host is ready. This will avoid having two similarly named schedulers on the domain.

6.6.2 Implementing HACMP on RISC/6000 SP Nodes

Section 3.5, “Implementing High Availability for RISC/6000 SP Nodes” on page 68 describes in detail how the HACMP software was configured to implement the LoadLeveler cluster design.

6.6.3 Configuration BEFORE HACMP Failover of LoadLeveler Node

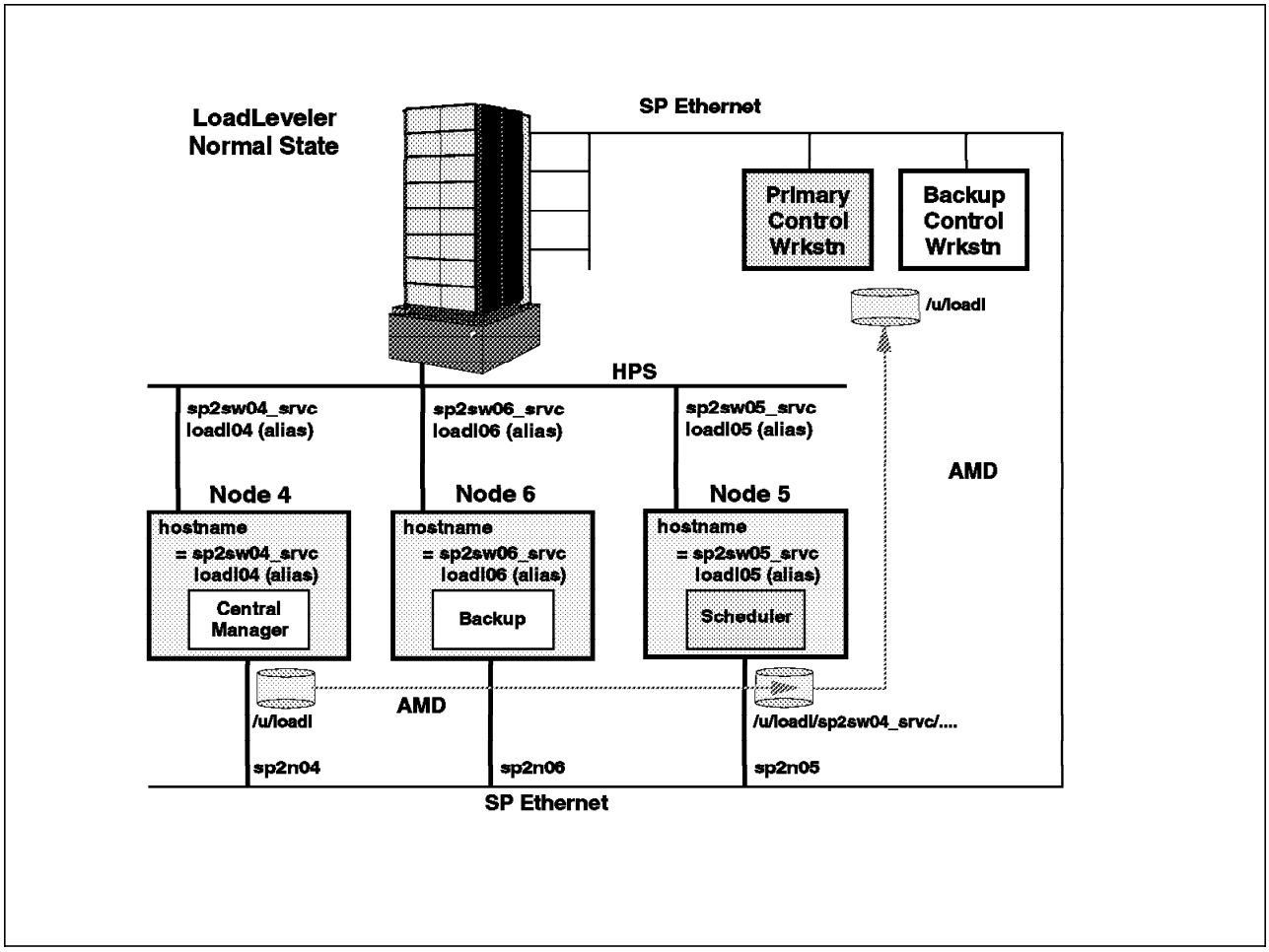


Figure 67. LoadLeveler High Availability Cluster Design for LoadLeveler

Note that the hostname of the scheduler is *sp2sw05_srvc*, and that the scheduling node is using *sp2sw05_srvc* as the IP service address. Note that Node 5 is the primary host for this scheduler entity. Node 4 is the central manager. Note that Node 6 is not defined in the LoadLeveler pool or configuration files as a LoadLeveler resource.

6.6.4 Configuration AFTER HACMP Failover of LoadLeveler Node

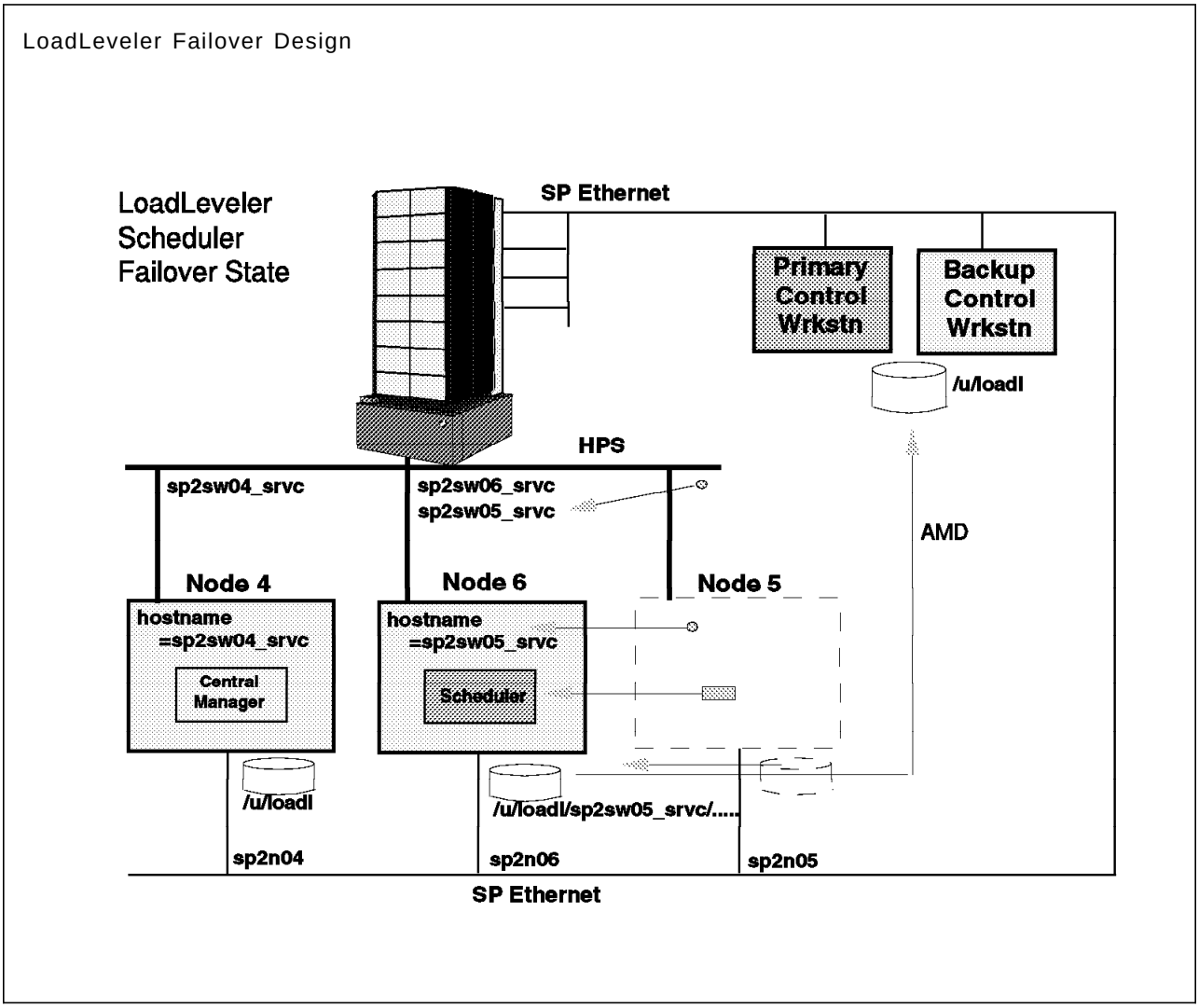


Figure 68. LoadLeveler High Availability Cluster AFTER Node Failover

Note that the scheduler hostname and service IP address of have both moved to the backup node, Node 6. At this point LoadLeveler sees the scheduler node as being back on service, but in reality, Node 6 has taken its place and is acting as the original scheduling node, having acquired its hostname and IP address.

6.7 Verifying LoadLeveler′s High Availability Implementation

The verification process involves failing over the scheduling node from Node 5 to Node 6 successfully, and reintegrating Node 5 back as the scheduler and Node 6 as the backup node.

We begin by establishing a normal steady state of the cluster where Node 5 is the scheduler and Node 6 is a backup node. As the backup for any scheduler node, Node 6 remains unconfigured until a failover occurs.

Once the normal state is achieved, several jobs are submitted to the scheduling node, some with a user hold on them.

Once the job is posted, a scheduling node failure is simulated using HACMP cluster shutdown with takeover facilities, then later by powering off the node.

The cluster is expected to have node 6 take over the scheduler identity and job queues.

Once the failover is completed and a steady state condition is achieved, Node 6 returns control of the scheduling function to Node 5 and becomes available once again as a backup.

6.7.1 Step 1. Start Up LoadLeveler on Nodes 4 and 5

From Node 4, log on as loadl, then start LoadLeveler by entering:

```
llctl -g start
```

```
sp2sw04_srvc loadl /u/loadl -> llctl -g start
4/17 10:47:42 Attempting to start sp2sw04_srvc.itsc.pok.ibm.com
4/17 10:47:42 CentralManager = sp2sw04_srvc.itsc.pok.ibm.com
4/17 10:47:42 Version 1.2.1  PTF 0  96/03/05
4/17 10:47:42 ************************************************
4/17 10:47:42 ***          LOADL_MASTER STARTING UP         ***
4/17 10:47:42 ***               STARTED BY loadl            ***
4/17 10:47:42 ***               PID = 13268                 ***
4/17 10:47:42 ************************************************
4/17 10:47:42 Real      uid: 300
4/17 10:47:42 Effective uid: 0
```

Verify LoadLeveler startup by entering the following command on Node 4 while logged on as *loadl*:

```
llstatus
```

```
sp2sw04_srvc-loadl /u/loadl -> llstatus
Name                      Schedd  InQ Act Startd Run LdAvg Idle Arch

sp2sw04_srvc.itsc.pok.ibm Down      0   0 Idle     0 0.15    56 R6000

sp2sw05_srvc.itsc.pok.ibm Avail    10   0 Idle     0 0.19    37 R6000


R6000/AIX41            2 machines  10 jobs   0 running
                       2 machines  10 jobs   0 running

The Central Manager is defined on sp2sw04_srvc.itsc.pok.ibm.com

All machines on the machine_list are present
```

Jobs which were running when the failure occurred should continue to run to completion on their execution node, but will be reported as running even after they complete. The job status when the failure occurred remains stale until the scheduling node returns to accept the new status and update the central manager. Without the HACMP failover to a backup scheduler, this status information will remain stale.

Jobs which were dispatched by the central manager to the scheduling node but not yet running on an execution node will remain in the pending state until the scheduling node returns.

6.7.2 Step 2. Establish Normal States for Nodes 4, 5 and 6

Verify that nodes 4, 5, and 6 are at their normal operating states, by issuing the commands shown underlined in the following sample screens.

- Node 4 should have the hostname *sp2sw04_srvc* and network interface css0 *spsw04_srvc*. As the central manager, the LoadLeveler daemon and *LoadL_negotiator* should be running on this node.
- Node 5 should have the hostname *sp2sw05_srvc* and network interface css0 *spsw05_srvc*. As the scheduler node, the LoadLeveler daemon *LoadL_schedd* should be running on this node.
- Node 6 should have the hostname *sp2sw06_srvc* and network interface css0 *spsw06_srvc*. There should be no LoadLeveler daemons running on the node.
- The HACMP clstrmgr daemon should be running on Node 5 and Node 6.

On Node 4:

```
sp2sw04_srvc-root / -> hostname
sp2sw04_srvc

sp2sw04_srvc-root / -> netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Co
lo0   16896 <Link>                          2054     0     2214     0
lo0   16896 127         localhost           2054     0     2214     0
en0   1500  <Link>10.0.5a.fa.1d.b8         31777     0    24205     0
en0   1500  9.12.20     sp2n04.itsc.pok    31777     0    24205     0
css0  65520 <Link>0.0.0.0.0.0               2757     0      221     2
css0  65520 9.12.6      sp2sw04.itsc.po     2757     0      221     2
css0  65520 9.12.23     sp2sw04_srvc.it     2757     0      221     2


sp2sw04_srvc-root / -> ps -efa ] grep loadl
   loadl 10942 14774   0 10:09:11      -  0:00 LoadL_kbdd -f
   loadl 13498 14774   0 10:09:11      -  0:00 LoadL_negotiator -f
   loadl 14774     1   0 10:09:11      -  0:00 /usr/lpp/LoadL/nfs/bin/Loa
   loadl 15036 14774   0 10:09:11      -  0:00 LoadL_startd -f
    root 15300 10156   1 10:18:21  pts/0  0:00 grep loadl


sp2sw04_srvc-root / -> lssrc -g cluster
Subsystem         Group            PID     Status
-                 -                -       -
```

On Node 5:

```
sp2sw05_srvc-root / -> hostname
sp2sw05_srvc

sp2sw05_srvc-root / -> netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Co
lo0   16896 <Link>                          728     0      776     0
lo0   16896 127         localhost           728     0      776     0
en0   1500  <Link>10.0.5a.fa.1a.92         6734     0     6316     0
en0   1500  9.12.20     sp2n05.itsc.pok    6734     0     6316     0
css0  65520 <Link>0.0.0.0.0.0               1890     0     1544     0
css0  65520 9.12.6      sp2sw05.itsc.po     1890     0     1544     0
css0  65520 9.12.23     sp2sw05_srvc.it     1890     0     1544     0

sp2sw05-srvc-root / -> ps -efa ] grep loadl
   loadl 11290 13336   0 10:09:14      -  0:00 LoadL_schedd -f
   loadl 13336     1   0 10:09:14      -  0:00 /usr/lpp/LoadL/nfs/bin/Loa
    root 14140 16678   1 10:14:42  pts/0  0:00 grep loadl
   loadl 14620 13336   0 10:09:14      -  0:00 LoadL_startd -f
   loadl 22302 13336   0 10:09:14      -  0:00 LoadL_kbdd -f

sp2sw05_srvc-root / -> lssrc -g cluster
Subsystem         Group            PID     Status
 clstrmgr         cluster          14512   active
 clsmuxpd         cluster          17352   active
```

On Node 6:

```
sp2sw06-loadl /u/loadl -> hostname
sp2sw06_srvc

sp2sw06_srvc-loadl /u/loadl -> netstat -i
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Co
lo0   16896 <Link>                          987     0     1052     0
lo0   16896 127         localhost            987     0     1052     0
en0   1500  <Link>10.0.5a.fa.3.33          5763     0     4853     0
en0   1500  9.12.20     sp2n06.itsc.pok     5763     0     4853     0
css0  65520 <Link>0.0.0.0.0.0               1546     0     1396     0
css0  65520 9.12.6      sp2sw06.itsc.po     1546     0     1396     0
css0  65520 9.12.23     sp2sw06_srvc.it     1546     0     1396     0

sp2sw06_srvc-loadl /u/loadl -> ps -efa ]grep loadl
   loadl 13632 10302   0 10:10:16  pts/0  0:00 -ksh
    root 20532 13398   1 10:12:43  pts/0  0:00 grep loadl

sp2sw06_srvc-loadl /u/loadl -> lssrc -g cluster
Subsystem         Group            PID     Status
 clstrmgr         cluster          14512   active
 clsmuxpd         cluster          17352   active
```

If Nodes 5 and 6 are not running the HACMP clstrmgr daemons, restart HACMP on both nodes, using (as root):

```
smitty clstart
```

Then repeat this step.

6.7.3 Step 3. Query the Job Queue

List the jobs queued by scheduler, using the command:

```
llq > /tmp/step3_results
```

.

```
sp2sw0-loadl /u/loadl -> llq
Id                       Owner      Submitted   ST PRI Class        Runni
------------------------ ---------- ----------- -- --- ------------ -----
sp2n05.11.0              loadl       3/27 00:06 I  50  No_Class
sp2n05.12.0              loadl       3/27 00:07 I  50  No_Class
sp2n05.4.0               loadl       3/27 00:05 H  50  No_Class
sp2n05.5.0               loadl       3/27 00:05 H  50  No_Class
sp2n05.13.0              loadl       3/27 00:07 H  50  No_Class
sp2n05.14.0              loadl       3/27 00:07 H  50  No_Class
sp2n05.15.0              loadl       3/27 00:07 H  50  No_Class
sp2sw05_srvc.18.0        loadl       3/29 12:15 H  50  No_Class
sp2sw05_srvc.19.0        loadl       3/29 12:15 H  50  No_Class
sp2sw05_srvc.25.0        loadl       4/11 19:18 H  50  No_Class
sp2sw05_srvc.27.0        loadl       4/18 18:26 H  50  No_Class

11 jobs in queue 2 waiting, 0 pending, 0 running, 9 held.
```

6.7.4 Step 4. Simulate Failure of Scheduler Node

Simulate a scheduler node failure by invoking an HACMP *clstop* command with the shutdown with takeover option, as follows. Log on as root on Node 5, then enter *smitty clstop*. Execute the panel as follows:

```
                              Stop Cluster Services

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

                                                      Entry Fields
* Stop now, on system restart or both                  now

  BROADCAST cluster shutdown?                          true
* Shutdown mode                                        takeover
          (graceful, graceful with takeover, forced)
```

6.7.5 Step 5. Verify Successful Takeover by Backup Node

On Node 6, log on as root and verify that the following have occurred:

- Hostname has been set to *sp2sw05_srvc* even though you are on Node 6.
- css0 network interface has a new alias IP label and address named *sp2sw05_srvc* added to its css addresses.
- The LoadLeveler scheduler daemon *LoadL_schedd* is running on this node.
- The LoadLeveler status shows the *sp2sw05_srvc* scheduler still running.
- The LoadLeveler job submitted prior to the scheduler failure is still in the queue and can be released.

```
sp2sw06_srvc root / -> hostname
sp2sw05_srvc

sp2sw0-root / -> netstat -i
lo0   16896 <Link>                          1344     0     1460     0
lo0   16896 127         localhost           1344     0     1460     0
en0   1500  <Link>10.0.5a.fa.3.33         11372     0    11207     0
en0   1500  9.12.20     sp2n06.itsc.pok   11372     0    11207     0
css0  65520 <Link>0.0.0.0.0.0               5154     0     5779     0
css0  65520 9.12.6      sp2sw06.itsc.po     5154     0     5779     0
css0  65520 9.12.23     sp2sw06_srvc.it     5154     0     5779     0
css0  65520 9.12.23     sp2sw05_srvc.it     5154     0     5779     0

sp2sw0-root / -> ps -efa ] grep loadl
   loadl 10426 13496   0 10:50:53      -  0:00 LoadL_schedd -f
   loadl 13496     1   0 10:50:53      -  0:00 /usr/lpp/LoadL/nfs/bin/Loa
    root 20952 15300   1 10:53:05  pts/0  0:00 grep loadl
   loadl 21180 13496   0 10:50:54      -  0:00 LoadL_startd -f
   loadl 22206 13496   0 10:50:54      -  0:00 LoadL_kbdd -f

sp2sw0-root / -> su - loadl

sp2sw0-root /u/loadl -> llstatus
Name                      Schedd  InQ Act Startd Run LdAvg Idle Arch
sp2sw04_srvc.itsc.pok.ibm Down      0   0 Idle     0 0.01   386 R6000
sp2sw05_srvc.itsc.pok.ibm Avail    10   0 Idle     0 0.00   278 R6000
The Central Manager is defined on sp2sw04_srvc.itsc.pok.ibm.com

All machines on the machine_list are present
```

As seen from Node 4, the central manager node, nothing has changed. From Node 4, log on as loadl and enter the following command:

```
llstatus
```

```
sp2sw0-loadl /u/loadl -> llstatus
Name                      Schedd  InQ Act Startd Run LdAvg Idle Arch
sp2sw04_srvc.itsc.pok.ibm Down      0   0 Idle     0 0.00   506 R6000
sp2sw05_srvc.itsc.pok.ibm Avail    10   0 Idle     0 0.00     0 R6000

R6000/AIX41            2 machines  10 jobs   0 running
                       2 machines  10 jobs   0 running

The Central Manager is defined on sp2sw04_srvc.itsc.pok.ibm.com

All machines on the machine_list are present
```

Using xloadl verifies that the job submitted in Step 3 is still available and can be released. Log on as loadl and issue the command:

```
llq
```

```
sp2sw0-loadl /u/loadl -> llq
Id                       Owner      Submitted   ST PRI Class        Runni
------------------------ ---------- ----------- -- --- ------------ -----
sp2n05.11.0              loadl       3/27 00:06 I  50  No_Class
sp2n05.12.0              loadl       3/27 00:07 I  50  No_Class
sp2n05.4.0               loadl       3/27 00:05 H  50  No_Class
sp2n05.5.0               loadl       3/27 00:05 H  50  No_Class
sp2n05.13.0              loadl       3/27 00:07 H  50  No_Class
sp2n05.14.0              loadl       3/27 00:07 H  50  No_Class
sp2n05.15.0              loadl       3/27 00:07 H  50  No_Class
sp2sw05_srvc.18.0        loadl       3/29 12:15 H  50  No_Class
sp2sw05_srvc.19.0        loadl       3/29 12:15 H  50  No_Class
sp2sw05_srvc.25.0        loadl       4/11 19:18 H  50  No_Class
sp2sw05_srvc.27.0        loadl       4/18 18:26 H  50  No_Class

11 jobs in queue 2 waiting, 0 pending, 0 running, 9 held.
```

Compare this output with the /tmp/step3_results file. They should match.

6.7.6 Step 6. Reintegrate Node 5 As the Scheduler

Restart HACMP on Node 5 by issuing the smitty clstart command on Node 5. HACMP should automatically return the scheduler to Node 5 and place Node 6 on backup status once again. Verify successful completion of this step by checking for the following normal state conditions:

- Node 4 should have the hostname *sp2sw04_srvc* and network interface css0 *spsw04_srvc*. As the central manager, the LoadLeveler daemon and *LoadL_negotiator* should be running on this node.
- Node 5 should have the hostname *sp2sw05_srvc* and network interface css0 *spsw05_srvc*. As the scheduler node, the LoadLeveler daemon *LoadL_schedd* should be running on this node.
- Node 6 should have the hostname *sp2sw06_srvc* and network interface css0 *spsw06_srvc*. There should be no LoadLeveler daemons running on this node.
- The HACMP clstrmgr daemon should be running on Node 5 and Node 6.

The commands and screen responses shown in step 2 should be used to verify the successful completion of this step as well. There should be no loss of job queues or submission.

6.7.7 Step 7. Repeat Steps 3 Thru 6 (Power-Off Node 5)

Repeat the verification steps 3 through 6. This time simulate the node failure by powering down Node 5 through the hardware switch on the node. The cluster should behave and reach the same failover configuration already tested.

6.7.8 Step 8. Failover the Primary CWS to the Backup CWS

Return the cluster to the normal operating state as described in Step 1, then failover the Primary Control Workstation to the Backup Control Workstation. Perform steps 2 through 6 once again to verify that LoadLeveler will operate while the RISC/6000 SP uses the backup control workstation as the active control workstation.

Appendix A. HACWS_Supplied Scripts Functional Flow

HACWS provides some scripts for the installation and verification of HACWS, and for run-time HACMP event scripts. Attached are the functional flows of the following three scripts:

- /usr/sbin/hacws/install_hacws

  The script used to install and configure HACWS on the primary control workstation, the backup control workstation, or both.

- /usr/sbin/hacws/spcw_apps

  This is the HACMP application server start script for HACWS.

- /usr/sbin/hacws/events/network_down.post_event

  The network_down post event script used to promote the local network failure to the node failure. This script is called by */usr/sbin/hacws/hacws_post_event* script which is the actual post event script set by HACWS.

A.1 ″install_hacws″ Script

1. Command syntax check and Prerequisite confirmation for HACMP configuration

   Check Syntax of command
   - $PROGNAME -p primary_hostname -b backup_hostanme [-s]

   Check if the hostnames are valid
   - Can be resolved?
   - Remote command can be issued?
   - Primary & backup are different?

   Pre-requisite PSSP filesets checking
   - ssp.client
   - ssp.basic
   - ssp.sysman
   - ssp.gui
   - ssp.hacws

   If “-s” flag is specified
   then this shellscript (without “-s” flag) runs on both primary and backup CWSs
   else NOP

   Make sure this is not a non-CWS SP node

   Check if Kerberos is configured properly
   - /etc/krb.conf
   - /etc/krb.realms
   - /etc/krb-srvtab

2. Check of HACWS configuration and status

   Check of /spdata/sys1
   - Check if primary CWS can get to the /spdata/sys1 filesystem
   - Check if /spdata/sys1 filesystem resides on an external volume group
   - Check If there are no SDR files on the backup (If there are, it means install_cw was mistakenly run on the backup)

   Check of the HACWS configuration state
   - The primary must be the currently active CWS

3. Setup of HACWS run-time-environment

   Set the HACWS_STATE to the appropriate value
   - Active primary : 2
   - Inactive backup : 16

   Create needed directories if they do not exist
   - /var/adm/SPlogs
   - /var/adm/SPlogs/SPconfig
   - /var/adm/SPlogs/spmon
   - /var/adm/SPlogs/spmon/hardmon
   - /var/adm/SPlogs/spmon/splogd
   - /var/adm/SPlogs/spmon/nc

   Add invocation of CWS log cleanup to cron
   Update /etc/services
   - updservices -s hardmon -p 8435 -t tcp
   - updservices -s sdr -p 5712 -t tcp
   - updservicds -s heartbeat -p 4893 -5 udp

   Do an mkssys for the sdr daemon that runs in the default partition
   if it hasn′t already been done, and start the sdr daemon
   Add hardmon to SRC if it has not already been done
   Create destination information file and copy the file to the backup
   Do a quick check of the SDR
   - init_ssp_envs (?)

   Copy switch expected files into /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.1nsb_8.0isb.0 /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.1nsb.0isb.0 /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.2nsb.0isb.0 /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.3nsb.0isb.0 /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.4nsb.0isb.0 /etc/SP
   - cp -p /usr/lpp/ssp/config/Eclock.top.1nsb.0isb.0 /etc/SP

   Create ssp dir in /usr/sys/inst.images if it does not exists
   Setup the spmon logging daemon
   - setup_logd

   Stop the daemon on the backup CWS
   - stopsrc -splogd

   Edit /etc/inittab file
   - Create a backup copy of /etc/inittab
   - Remove the “hb,”

   “sp,”
   “sdrd,”
   “hr,”
   “splogd,”
   “hardmon,” and
   “hmon”
   entries from inittab
   - Create the “/etc/rc.hacws” file
   - Add /etc/rc.hacws to inittab

   Add HACWS data to the SDR
   - SP class: “SDRChangeAttrValues SP backup_cw=$short_backup_HNAME active_cw=control_workstation”
   - NET_ATTR=cw_ipaddrs
   - NET_ATTR=ipaddrs_bucw
   - SP class: “SDRChangeAttrValues SP ‘${NET_ATTR}=$ipaddrs’”

   Remove the contents of the /spdata/sys1 directory on the backup CWS

4. Set I/O pacing parameters for HACMP

Set the disk I/O pacing parameters
 - minpout : 24
 - maxpout : 33
    If already set up, then exit
exit 0

A.2 ″spcw_apps″ Script

[Main]

Check command syntax
 - $PROGNAME -u/-d -i/-a
If “-u” flag is specified
 then if “-i” flag is specified
  then /etc/rc.hacws
  else if “-a” is specified
   then /etc/rc.hacws -active
If the CWS is inactive
 then if “-u” flag is specified
  then start_on_inactive
  else stop_on_inactive
 else if “-u” flag is specified
  then start_on_active
  else stop_on_active
If “-d” flag is specified
 then if “-i” flag is specified
  then /etc/rc.hacws
  else if “-a” flag is specified
   then /etc/rc.hacws -active
exit

[start_on_active()]

Run the /spdata/sys1/hacws/rc.syspar_aliases script to add any required IP address alias
 - /spdata/sys1/hacws/rc.syspar_aliases -add
Start the SDR daemon for the default partition
 - sdr -spname vhostname -s start
Update the SRC and start the daemon groups for sdr, hb and hr using the latest partition configuration obtained from the SDR
 - sdr restore
 - hr restore
Update active_cw field of the SP class in the SDR
 - SDRChangeAttrValues SP active_cw=$ACTIVE_CW
 ($ACTIVE_CW=backup_cw or control_workstation)
Disable amd_config script before running /etc/rc.sp, and re-enable adm_config script
Start and restart sysctld
Start hardmon
Start syslogd
Run setup_server
Return

[start_on_inactive()]

Check if the SDR is available

If not, then quit
Disable amd_config script before running /etc/rc.sp, and re-enable adm_config script
If file collections are specified, then configure the inactive CWS as a file collections client
- stopsrc -s supfilesrv
- /var/sysman/supper install sup.admin
- /var/sysmansupper install user.admin
- /var/sysman/supper install power_system
- crontab -l]grep -q "/usr/sbin/hacws/spcw_filec_update"
Determines whether the inactive CWS needs to act as an NTP client of the active CWS, and if necessary, configures the inactive CWS as an NTP client
- spcw_defer_ntp
Stop and restart sysctld
Return

[stop_on_active()]

Stop splogd
Stop hardmon
Stop sysctld
Stop supfilesrv
Stop the sdr, hb and hr groups
Unexport the directories that were previously exported by setup_server
Run the /spdata/sys1/hacws/rc.syspar_aliases script to delete any required IP address aliases
- /spdata/sys1/hacws/rc.syspare_aliases -delete
Return

[stop_on_inactive()]

Delete the spcw_filec_update crontab entry, if it exists
Return

A.3 "network_down.post_event" Script

Check if this is a global network_down or this event occurred on a remote node
  If so, then exit
Check if the HACWS resource group (hacws_group1) exited
  If not, then exit
Check if this node participate in HACMP cluster and backup node exist
  If not, then exit
Check if the failed network contain one of the CWS's service addresses
  If not, then exit
Check if this node control the shared volume group
  if not, then exit
Run failover
- exec clstop -N -gr -y -s

Appendix B. Enabling Address Resolution Protocol on HiPS

DISCLAIMER

Be careful and complete all steps presented. The following sequence requires that CuAt is backed up. If CuAt is not backed up on the nodes and user error corrupts CuAt and reboots, the SP nodes may be corrupted and have to be re-installed. If you corrupt CuAt on the nodes, be sure that you copy your backup (CuAt.save) back to /etc/objrepos/CuAt prior to any reboot. Be careful! If you feel that this process is too risky, please see the *SP Administration Guide*, GC23-3897-01 for the procedure in enabling ARP through "customized" operation.

1. On the control workstation, enter the command:

   ```
   dsh -av "/usr/lpp/ssp/css/ifconfig css0"
   ```

   If

   "NOARP"

   appears on output from any of the nodes, then proceed to the rest of the steps for enabling ARP to be used for IP takeover on the High Performance Switch.

   Note: ARP must be enabled on all SP nodes connected to the HiPS.

2. On the control workstation, enter:

   ```
   dsh -av "cp /etc/objrepos/CuAt /etc/objrepos/CuAt.save"
   ```

3. On the control workstation, enter:

   ```
   dsh -av "odmget -q'name=css and attribute=arp_enabled' CuAt | sed
   s/no/yes/ > /tmp/arpon.data"
   ```

4. On the control workstation, enter:

   ```
   dsh -av "odmchange -o CuAt -q'name=css and attribute=arp_enabled'
   /tmp/arpon.data"
   ```

5. Verify that the previous commands worked by issuing

   ```
   dsh -av "odmget -q'name=css and attribute=arp_enabled' CuAt | grep
   value"
   ```

   You should see an entry from every node reporting

   "value=yes"

6. On the control workstation, enter:

   ```
   dsh -av rm /tmp/arpon.data
   ```

7. On the control workstation, enter:

   ```
   dsh -av "bosboot -a -d /dev/hdisk0"
   ```

8. Shutdown and reboot all nodes.

Appendix C. HiPS Global Network Failure Scripts

The following is the code contained in the scripts used for the HiPS global network failure using High Availability Cluster Multi-Processing These scripts are given on an example basis and are not supported in any form. They are given as an example of a specific implementation of recovery of the Switch and require certain criteria to function correctly. Please refer to chapter 3.7, “Implementing High Availability for HiPS Network Failure” on page 105 for the specified criteria.

POST_network_up

```
#!/bin/ksh
#
# POST_network_up script
#
set -x
PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster
/events/utils:/usr/sbin/cluster/utilities
export PATH

PROGNAME=$0

if [ $# -ne 4 ]
then
      echo Usage: $PROGNAME nodename network_name
      exit 2
fi
#
LOCALNODENAME=odmget HACMPcluster | grep nodename | cut -d′″′
 -2

#
#
# Set the Run-Time Parameter values and export them
# to all successive scripts.
#
set -a
eval /usr/sbin/cluster/utilities/cllsparam -n $LOCALNODENAME
set +a

if [ ″$VERBOSE_LOGGING″ = ″high″ ]
then
      set -x
fi
set -u

#
#  Define the directory where to look for .HPS_swap script
#  (must be the same as in network_down).
#
LOCALDIR=/usr/local/cluster/tmp
#
```

```
#   Define a variable for the HiPS networkname.
#
HPSNETWORKNAME=$(/usr/sbin/cluster/utilities/cllsnw | grep HPS | awk
'{ print $1 }')
#
#  Only if this is the local network_up for HiPS network do
#  we want to do something.
#
if [ "$4" = "$HPSNETWORKNAME" ]
then
#
#       if $LOCALDIR/.HPS_swap exists (i.e. swap_adapter has
#       been scheduled but not been called yet)
#       remove it (for example, cancel the swap_adapter).
#
       if [ -x $LOCALDIR/.HPS_swap ]
       then
             rm $LOCALDIR/.HPS_swap
             return 0
       fi
fi
#
# Detemine which network is coming up and define variables for the
# IP addresses of the swapping interfaces.
#
NET_UP=$4
LOCALNODENAME=odmget HACMPcluster | grep nodename | cut
-d"" -f2
BACKUP_ADAPTER=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME
| grep BACKUP | grep service | awk '{ print $7 }')
BACKUP_NETWORK=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME
| grep BACKUP_ADAPTER | grep service | awk '{ print $3 }')
#
#
# If it is the backup adapter interface that is coming up then exit,
#  otherwise, determine that the switch interface is up.
#
#
if [ $BACKUP_NETWORK != $NET_UP ]
then
       return 0
fi
HPS_ADAPTER=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME |
 grep service | grep HPS | awk '{ print $7 }')
ADAPTER_NAME=$(netstat -in | grep $HPS_ADAPTER | awk ' { print $1 }')
ADAPTER_STATE=$(/usr/sbin/ifconfig $ADAPTER_NAME | grep
 $ADAPTER_NAME | awk -F "<" '{print $2 }'| cut -f1 -d",")
if [ "$ADAPTER_STATE" != "UP" ]
then
       return 0
fi
if [ $( netstat -in | grep $HPS_ADAPTER | grep $ADAPTER_NAME |
 awk '{ print $1') = css0  ]
then
       return 0
```

```
fi
#
#
# If the switch service address exists on the backup adapter interface
# and the switch interface returns, bring the backup interface down.
#
#
ifconfig $ADAPTER_NAME down

------------------------------------------------------------------------
```

POST_network_down

```
#!/bin/ksh
#
# POST_network_down event script
#
#

######################################################################
PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster
/events/utils:/usr/sbin/cluster/utilities
export PATH
set -x
PROGNAME=$0

if [ $# -ne 4 ]
then
      echo Usage: $PROGNAME nodename network_name
      exit 2
fi

if [ "$VERBOSE_LOGGING" = "high" ]
then
      set -x
fi

set -u
#
#
#   Define directory in which to put .HPS_swap script
#   (must be the same as in network_up and post_join_standby).
#
LOCALDIR=/usr/local/cluster/tmp
#
#   We want to do something only if this is a global network
#   failure of HPS network.
#
 HPSNETWORKNAME=$(/usr/sbin/cluster/utilities/cllsnw | grep HPS |
  awk '{ print $1 }')
LOCALNODENAME=odmget HACMPcluster | grep nodename |
 cut -d"" -f2
#
if [ "$3" = "-1" -a "$4" = "$HPSNETWORKNAME" ]
then
```

```
#       Get IP addresses of HPS service and BACKUP service.
#
        HPS_IP=cllsif -i $LOCALNODENAME]grep hps|grep service|
        awk '{ print $7}'
        BACKUP_IP=cllsif -i $LOCALNODENAME|grep BACKUP|grep
        service|awk '{ print $7}'
#
#       Create script $LOCALDIR/.HPS_swap which will later
#       call the swap_adapter event.
#
        echo "/usr/local/cluster/events/hps_swap_adapter $LOCALNODENAME
 dummy HPS_IP $BACKUP_IP" > $LOCALDIR/.HPS_swap &&
        echo "wait " >>  $LOCALDIR/.HPS_swap &&
        echo "/usr/local/cluster/events/correct_routes network_down -1
         $HPSNETWORKNAME " >>  $LOCALDIR/.HPS_swap &&
        chmod u+x $LOCALDIR/.HPS_swap || exit 1
#
#       In the background, do the following:
#       Wait for 120 seconds
#       (since we want only "real" failures to start the swap).
#       If $LOCALDIR/.HPS_swap is still there (for example no
#       network_up occurred), execute it and rename it to
#       $LOCAL DIR/.HPS_swap_complete
#
        ( sleep 120
        if [ -x $LOCALDIR/.HPS_swap ]
        then
          mv $LOCALDIR/.HPS_swap $LOCALDIR/.HPS_swapping &&
          $LOCALDIR/.HPS_swapping  &&
          mv $LOCALDIR/.HPS_swapping $LOCALDIR/.HPS_swap_complete||
            exit 1
        fi ) &
fi

exit 0
```

---

correct_routes

```
#!/bin/ksh
#
# correct_routes script
#
# This script is called from POST_network_down.
#
set -x
echo " EXECUTING CORRECT_ROUTES "  >>  /tmp/hacmp.out
#
# Assign variables for failed network, local node and HPS network.
#
LOCALDIR=/usr/local/cluster/tmp
NET_DOWN=$3
LOCALNODENAME=odmget HACMPcluster | grep nodename |
 cut -d"" -f2
```

```
HPS_NETWORK=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME |
 grep hps |grep service | awk '{ print $3 }')
#
# Check to see if the failed network is the HiPS network.
#
if [ $HPS_NETWORK != $NET_DOWN ]
then
       return 0
fi
#
#  Get the IP service address of the HiPS and BACKUP adapters.
#
BACKUP_ADAPTER=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME |
 grep BACKUP | grep service | awk ' { print $7 }')
HPS_ADAPTER=$(/usr/sbin/cluster/utilities/cllsif -i $LOCALNODENAME |
 grep service | grep HPS | awk '{ print $7 }')
#
# Check to see if there are any incorrect routes and delete them.
#
netstat -rn | fgrep " UG " | grep $BACKUP_ADAPTER | awk '{
 print $1" "$2" "&3 }' | read NETWORK junk TYPE
if [ X$TYPE = "XUG" ]
then
    route delete -net $NETWORK $BACKUP_ADAPTER
fi
sleep 2
#
# Obtain the netmask of the HiPS interface and check to see if the
# backup IP address is currently on the css0 interface. If it is,
# delete the IP address alias from the interface and recreate it to
# ensure that the correct alias exists.
#
SET_NETMASK=$(ifconfig css0 | grep 0x | awk '{ print $4 }')
sleep 2
if [ X$(netstat -in | grep css0 | grep $BACKUP_ADAPTER |
 awk '{ print $4 }') != X$BACKUP_ADAPTER ]
then
       return 0
fi
/usr/lpp/ssp/css/ifconfig css0 inet $BACKUP_ADAPTER delete
sleep 2
/usr/lpp/ssp/css/ifconfig css0 inet $BACKUP_ADAPTER netmask
 $SET_NETMASK alias up
return
```

---

hps_swap_adapter

```
#!/bin/sh
#
# hps_swap_adapter
#
# This script is a modified version of the swap_adapter event script to
```

```
# implement HiPS network failover.  Scripts from a white paper 'HACMP/6
# on RS/6000 SP' have been also been used when creating the script.
#
PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster
/events/utils:/usr/sbin/cluster/utilities
export PATH

PROGNAME=$0
STATUS=0


#############################################################################
# Name: flush_arp
#
#      Flushes entire arp cache
#
# Returns: None.

#############################################################################
flush_arp () {
      for addr in /etc/arp -a | /bin/sed -e 's/^.*
       (\([0-9].*[0-9]\)).*/$/\1/'-e
       /incomplete/d
      do
            /etc/arp -d $addr >/dev/null 2>&1
      done
      return 0
}

#
# This routine maps a label_address to an internet_dot_address.
# Thus, given a label_address, we do not require the name server
# to get its corresponding dot address.
#
name_to_addr () {
      /bin/echo cllsif -cSn $1 | cut -d: -f7 | uniq
      exit $?
}

if [ $# -ne 4 ]
then
      echo Usage: $PROGNAME nodename network ip_address1 ip_address2
      exit 2
fi
if [ $1 != $LOCALNODENAME ]
then
      flush_arp
      exit 0
fi

LOCALNODENAME=odmget HACMPcluster | grep nodename |
 cut -d'"' -f2

#
# Set the Run-Time Parameter values and export them
```

```
# to all successive scripts.
#
set -a
eval /usr/sbin/cluster/utilities/cllsparam -n
 $LOCALNODENAME
set +a

if [ "$VERBOSE_LOGGING" = "high" ]
then
      set -x
fi

set -u

if [ "$NAME_SERVER" = "true" ]
then
      #
      # Turn off name server before swapping.
      #
      /usr/sbin/cluster/events/utils/cl_nm_nis_off
      if [ $? -ne 0 ]
      then
            STATUS=1
      fi
fi

#
# If address is given in IP label format, convert to dot address.
# This is required by cl_swap_IP_addr.
#
ADDR1=name_to_addr $3

#
# Get the interface associated with the future service address.
#
SERVICE_INTERFACE=/usr/sbin/cluster/utilities/clgetif -a
 $ADDR1
if [ "$SERVICE_INTERFACE" = "" ]
then
      cl_log 315 "Interface for $3 is not found." $3
      exit 1
fi

NETMASK1=/usr/sbin/cluster/utilities/clgetif -n
 $ADDR1
#
# Do the same for ADDR2.
#
ADDR2=name_to_addr $4

STANDBY_INTERFACE=/usr/sbin/cluster/utilities/clgetif -a
 $ADDR2
if [ "$STANDBY_INTERFACE" = "" ]
then
      cl_log 315 "Interface for $4 is not found." $4
```

```
        exit 1
fi

NETMASK2=/usr/sbin/cluster/utilities/clgetif -n
 $ADDR2
#
# This section of the script deals specifically with the css0 interface
# and IP address swapping whether the interface is failing or being
# reintergrated into the cluster.
#
if [ "$SERVICE_INTERFACE" = "css0 " ]
then
        ifconfig $STANDBY_INTERFACE down && \
        /usr/sbin/cluster/events/utils/cl_swap_HPS_IP_address css0
          $ADDR NETMASK1 delete && \
        /usr/sbin/cluster/events/utils/cl_swap_HPS_IP_address css0
          $ADDR NETMASK2 && \
        /usr/sbin/cluster/events/utils/cl_swap_IP_address
         $STANDBY_INTERFACE $ADDR1 $NETMASK1
else
        if [ "$STANDBY_INTERFACE" = "css0 " ]
        then
                /usr/sbin/cluster/events/utils/cl_swap_HPS_IP_address
                 css0 &ADDR2 $NETMASK2 delete && \
                /usr/sbin/cluster/events/utils/cl_swap_IP_address
                 $SERVICE_INTERFACE $ADDR2 $NETMASK2 && \
                /usr/sbin/cluster/events/utils/cl_swap_HPS_IP_address
                 css0 $ADDR1 $NETMASK1
        else

        /usr/sbin/cluster/events/utils/cl_swap_IP_address
         $SERVICE_INTERFACE $ADR2 \
                $STANDBY_INTERFACE $ADDR1 $NETMASK1
        fi
fi
if [ $? -ne 0 ]
then
        STATUS=1
        cl_log 316 "Can not perform adapter swap on $SERVICE_INTERFACE
 and STANDBY_INTERFACE" $SERVICE_INTERFACE $STANDBY_INTERFACE
fi

if [ "$NAME_SERVER" = "true" ]
then
        #
        # After swapping, turn on name server.
        #
        /usr/sbin/cluster/events/utils/cl_nm_nis_on
        if [ $? -ne 0 ]
        then
                STATUS=1
        fi
fi
exit $STATUS
```

Appendix D. Special Notices

This publication is intended to help IBM customers and system engineers who will be implementing and configuring their RISC/6000 SP systems to be highly available. The information in this publication is not intended as the specification of any programming interfaces that are provided by PSSP, HACMP and LoadLeveler packages. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP, HACMP and LoadLeveler 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:

IBM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

Appendix E. Related Publications

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

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 227.

- *High Availability Cluster Multi-Processing 4.1 for AIX*, SC23-2773
- *IBM RISC System/6000 Scalable POWERparallel Systems Administration Guide*, GC23-3897
- *IBM RISC System/6000 Scalable POWERparallel Systems Planning*, GC23-3902
- *IBM RISC System/6000 Scalable POWERparallel Systems Installation Guide*, GC23-3898
- *HACMP for AIX Planning Guide*, SC23-2768
- *AIX Installation Guide*, SC23-2550
- *HACMP for AIX Installation Guide*, SC23-2769
- *RS/6000 SP System Management Easy, Lean and Mean*, GG24-2563
- *An HACMP Cookbook*, SG24-4553
- *High Availability on the RISC System/6000 Family*, SG24-4551
- *IBM LoadLeveler Technical Presentation Support*, ZZ81-0348
- *IBM PSSP Technical Presentation Support*, SG24-4542

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.

E.2 Other Publications

These white paper publications are also relevant as further information sources:

- HACMP/6000 on RS/6000 SP (available from marketing tools)
- HACMP for AIX Version 4.1.1 on SP (available from marketing tools)
- Powerparallel Systems Availability and Recoverability for commercial systems (available from marketing tools)

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com/redbooks.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks/redbooks.html

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	bookshop at dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29554	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada (toll free)	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com/redbooks
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number _____
- Credit card number _____

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

ADDR	address
AIX	advanced interactive executive (IBM′s flavor of UNIX)
AMD	automount deamon
APAR	authorized program analysis report
API	application program interface
ARP	address resolution protocol
BIN	binary
CDE	Common Desktop Environment (from X/Open)
CPU	central processing unit
CWS	control workstation
ECHO	electronic controlled high output (IBM)
EOF	end of file
FAX	facsimile
FDDI	fiber distributed data interface (100 Mbit/s fiber optic LAN)
HA	high availability
HACMP	high availability cluster multi-processing (AIX)
HISTORY	history file for event calendar
HPS	high performance switch
HSD	hashed shared disk
RVSD	recoverable virtual shared disk
I/O	input/output
IBM	International Business Machines Corporation
IBMLINK	IBMLink, single framework offering including, interactive applications, electronic mail, electronic data interchange and file transfer (IBM)
ICMP	internet control message protocol
ID	identification number
IP	internet protocol (ISO)
IPL	initial program load
IS	Information Systems
ITSC	International Technical Support Center (IBM)
ITSO	International Technical Support Organization
KB	kilobyte, 1000 bytes (1024 bytes memory)
LAN	local area network
LED	light emitting diode
LIB	library
LL	load leveler
LOG	logarithm
LOG	logon
LV	local volume
LVM	logical volume manager
MIB	management information base (OSI management)
NFS	network file system (USA, Sun Microsystems Inc)
NIM	network installation management
NIS	network information system (Sun Microsystems Inc.)
ODM	object data manager (AIX)
ORACLE	Oak Ridge automatic computer and logic engine
PC	Personal Computer (IBM)
PE	parallel edition
PING	packet internet groper
PIO	parallel input output
POK	Poughkeepsie, NY
PPS	Power Parallel Systems
PSSP	AIX Parallel System Support Programs (IBM program product for SP1 and SP2)
PTF	program temporary fix
PTFS	program temporary fixes
PV	physical volume
PVM	Parallel Virtual Machine (public domain software)
RAID	redundant array of independent disks
RAS	reliability, availability, serviceability
RISC	reduced instruction set computer/cycles
SCSI	small computer system interface

SDR system data repository

SMIT System Management Interface Tool (see also DSMIT)

SMUX single multiplexor

SNMP simple network management protocol (a TCP/IP protocol)

SP IBM RS/6000 Scalable POWERparallel Systems (RS/6000 SP)

SPF single point of failure

SPOOL simultaneous peripheral operation on-line

SSA serial storage architecture

TCP/IP Transmission Control Protocol/Internet Protocol (USA, DoD, ARPANET; TCP=layer 4, IP=layer 3, UNIX-ish/Ethernet-based system-interconnect protocol)

TTY teletypewriter

TXT text

U micro (10**-6)

UG users group

UNIX an operating system developed at Bell Laboratories (trademark of UNIX System Laboratories, licensed exclusively by X/Open Company, Ltd.)

UPS uninterruptible power supply/system

US microsecond

USER user application

VG volume group

VSD virtual shared disk

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

Printed in U.S.A.