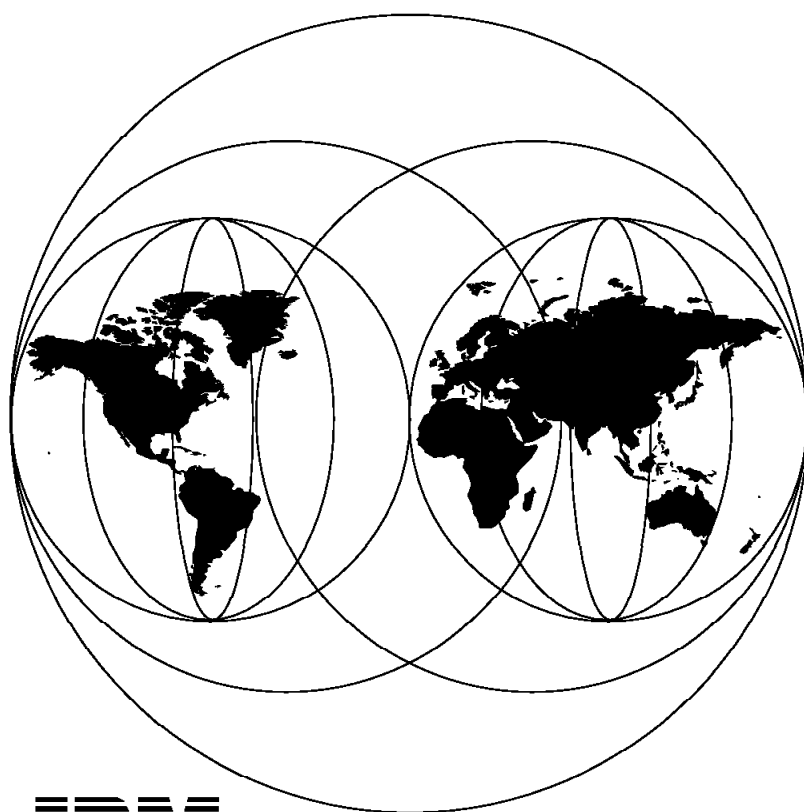


HACMP Enhanced Scalability

October 1997



IBM

**International Technical Support Organization
Poughkeepsie Center**



International Technical Support Organization

SG24-2081-00

HACMP Enhanced Scalability

October 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, "Special Notices" on page 171.

First Edition (October 1997)

This edition applies to HACMP Enhanced Scalability Version 4, Release 2, Modification 1 for use with PSSP Version 2, Release 3 and AIX Version 4, Release 2, Modification 1.

Warning

This book is based on a pre-GA version of a product and may not apply when the product becomes generally available. It is recommended that, when the product becomes generally available, you destroy all copies of this version of the book that you have in your possession.

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.

© **Copyright International Business Machines Corporation 1997. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	vii
Tables	ix
Preface	xi
The Team That Wrote This Redbook	xi
Comments Welcome	xii

Part 1. Introduction to High Availability and HACMP ES	1
Chapter 1. HACMP ES	5
1.1 What is HACMP ES?	5
1.2 Why HACMP ES?	6
1.3 IBM High Availability Product Family	7
1.4 High Availability Infrastructure	8
1.5 Benefits of HACMP ES	9
1.6 Single Points of Failure	12
1.7 Limitations of the Current Release	15
Chapter 2. Differences Between HACMP and HACMP ES	17
2.1 Look and Feel Compared to HACMP	17
2.2 Functional Similarities with HACMP	20
2.3 Functional Similarities with HACMP (cont'd)	22
2.4 Functional Similarities with HACMP	25
2.5 HACMP ES Event Processing	26
2.6 Functional Differences with HACMP	27
Chapter 3. Components and Their Relationships	29
3.1 HACMP ES Components	29
3.1.1 HACMP ES Subsystem Components	29
3.1.2 Other Related Components	30
3.2 HACMP ES Component Definitions	31
3.2.1 Group Services/ES	31
3.2.2 Topology Services/ES	31
3.2.3 HACMP ES Cluster Manager	32
3.3 Logical Connection to PSSP Components	33
3.4 PSSP Component Definitions	34
3.5 Multiple Clusters in a Partition	35
3.6 Logical Connections to Client Components	36
3.7 Client Interface Component Definitions	37
3.7.1 The Cluster SNMP Agent	37
3.7.2 Cluster Information Services	37
3.8 Topology Services Layer	38
3.9 Group Services Layer	41
3.10 Cluster Manager Layer	43
3.11 Start Sequence	44
Chapter 4. HACMP ES Event Management	47
4.1 Event Management Flow	47
4.2 Event Mapping	50
4.3 Event Mapping Files	51

4.3.1	Rules File	51
4.3.2	Event Mapping Example	53
4.4	Recovery Program Structure	54
4.4.1	Recovery Programs	54
4.4.2	Recovery Programs Example	57
4.4.3	Synchronization of Recovery Programs	58
4.5	User-Defined Event Detection	60
4.6	Event Priority	63
Chapter 5. HACMP ES Protocols		67
5.1	HACMP ES Protocols	68
5.1.1	Membership (join/leave)	68
5.1.2	Voting	68
5.1.3	Barrier	69
5.1.4	Cbarrier	69
5.1.5	Adapter Membership State	70
5.2	Node Joining	71
5.2.1	First Node Joining	71
5.2.2	Additional Node Joining	74
5.3	Node Departure and Rejoining	77
Chapter 6. Planning		79
6.1	Planning Hardware Configuration	79
6.2	Planning Software Configuration	82
6.2.1	Prerequisites	82
6.2.2	Planning Applications	83
6.2.3	Planning Clients	83
6.3	Planning Node Relationships	85
6.4	Planning Different Scenarios	86
6.5	Planning for Coexistence with Other High Availability Products	88
6.5.1	Coexistence with HACMP for AIX and HAGEO	88
6.5.2	Coexistence with HACWS	90
6.5.3	Coexistence with HANFS	90
6.5.4	Coexistence with RVSD	91
6.5.5	Coexistence with GPFS	92
Chapter 7. HACMP ES Installation and Customization		93
7.1	Installation	93
7.1.1	Installation Prerequisites	93
7.1.2	Installation Steps	98
7.2	HACMP ES Customization	100
7.2.1	Customization	100
7.2.2	Steps to Define Customized Events	102
7.2.3	Event Definition Examples	105
7.3	Migrating from Existing HACMP Environment to HACMP ES	107
Chapter 8. Configuration Examples		109
8.1	Takeover Scenarios	109
8.1.1	Rotating Failover Principles	109
8.1.2	Rotating Takeover Scenario	111
8.2	Cascading Configurations	114
8.2.1	Cascading Failover Principles	114
8.2.2	Cascading Configuration	115

Part 2. Technical Implementations	119
Chapter 9. Installation of HACMP ES	121
9.1 Prerequisites	121
9.1.1 Hardware Prerequisites	121
9.1.2 Software Prerequisites	121
9.2 Software Installation and Configuration	122
Chapter 10. Migration	123
10.1 New Installation by Using Scripts from Previous Installation	123
10.2 Using Snapshot	123
10.2.1 Using Snapshot for HACMP for AIX V4.2.1 and Older than V4.1.1	124
10.2.2 Using Snapshot for HACMP V4.1.1 or V4.2	125
Chapter 11. User-Defined Events	127
11.1 Prerequisites	127
11.2 Installation	127
11.3 Configuration	127
11.3.1 HACMP Scripts for an Application	127
11.3.2 The Recovery Program (rp File)	128
11.3.3 The Action Files	129
11.3.4 The rules.hacmprd File	129
Chapter 12. Using Kerberos	131
12.1 Kerberos Overview	131
12.2 Change HACMP to Use Kerberos	131
12.3 How to Kerberize the HACMP Interfaces	132
12.3.1 Using PSSP 2.3 Functions	132
12.3.2 Using Native Kerberos Functions	135
Chapter 13. Adding Additional (Unsupported) Interfaces to the SDR	139
Chapter 14. Cascading by Using One Network Adapter	143
14.1 Our Test Environment	143
14.2 Our Workaround	143
14.2.1 Technical Description	143
14.2.2 Advantages	143
14.2.3 Disadvantages	144
14.2.4 Installation	144
14.2.5 Configuration	144
14.2.6 System Requirements	145
14.2.7 Support	146
14.2.8 Using the Modified Event Scripts	147
Part 3. Appendices	149
Appendix A. AIX Scripts	151
A.1 Modified HACMP Scripts	151
A.1.1 The HACMP Script acquire_takeover_addr	151
A.1.2 The HACMP Script release_takeover_addr	156
A.1.3 The HACMP Script cl_alias_IP_address	159
A.1.4 The HACMP Script cl_unalias_IP_address	161
A.2 Scripts for User-Defined Events	164

A.2.1 The Start Script for the HACMP Application	164
A.2.2 The Stop Script for the HACMP Application	164
A.2.3 The rules.hacmprd File	165
A.2.4 The rwho.rp File	168
A.2.5 The rwho_msg_local File	168
A.2.6 The rwho_msg_remote File	168
A.2.7 The rwho_msg_complete File	168
A.2.8 The rwho_restart File	169
Appendix B. Special Notices	171
Appendix C. Related Publications	173
C.1 International Technical Support Organization Publications	173
C.2 Redbooks on CD-ROMs	173
C.3 Other Publications	173
How to Get ITSO Redbooks	175
How IBM Employees Can Get ITSO Redbooks	175
How Customers Can Get ITSO Redbooks	176
IBM Redbook Order Form	177
Glossary	179
List of Abbreviations	185
Index	187
ITSO Redbook Evaluation	189

Figures

1.	Topology Services/ES Status Example	39
2.	Group Services/ES Status Example	42
3.	Change/Show Run Time Parameters Screen	99
4.	Change Topology and Group Services Configuration Screen	100
5.	Upr6 Process	106
6.	HACMP Run Time Parameters	132
7.	Add Additional Adapter Information	133
8.	Set setup_server Information	134
9.	kadmin	136
10.	kadmin (help)	136
11.	kadmin (?)	137
12.	SMIT (SDR Additional Adapter Database Information)	140

Tables

1. Prerequisite Software	94
2. Structured Byte String (SBS):	106

Preface

This redbook describes the new HACMP Enhanced Scalability (ES) product (Version 4 Release 2.1). It discusses the product's design principles and how it fits into the HACMP family of products. HACMP ES 4.2.1 requires PSSP 2.3 and AIX 4.2.1.

The redbook consists of two parts. The first part is in a presentation format that consists of a set of foils and a detailed explanation for each foil. The second part discusses some implementation details, offers ideas for possible configuration scenarios, and provides guidelines for installation and customization of software detection events using the Event Management infrastructure.

This redbook is intended to help IBM customers, Business Partners, IBM System Engineers, and other RS/6000 SP specialists who are involved with HACMP Enhanced Scalability Version 4 Release 2.1 projects, including the education of RS/6000 professionals responsible for installing, configuring, and administering PSSP Version 2 Release 3 with HACMP ES Version 4 Release 2.1.

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.

Endy Chiakpo is a Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of RS/6000 SP. He holds a B.S. degree in Physics and a Master of Science degree in Electrical Engineering from Syracuse University, New York. Before joining the ITSO, Endy worked in the IBM Poughkeepsie Lab.

Bernhard Buehler is an HACMP Country Specialist in Germany. He has worked at IBM for 15 years, and has seven years of experience in the AIX field. His areas of expertise include HACMP, RS/6000 SP, and HAGEO. He is a co-author of the redbooks *DataJoiner Implementation and Usage Guide* and *Enterprise-Wide Security Architecture and Solutions Presentation Guide*.

Akihiro Sakuma is an I/T Specialist at IBM Japan Systems Engineering Co., Ltd. (an IBM Japan subsidiary) in Japan. He has six years of experience in the AIX field. He has worked at IBM for six years.

Teppo Seesto is a *Systeemineuvottelija* (System Engineer) in Finland. He has worked at IBM for nine years, and has seven years of experience in the AIX field. He holds a Master of Science degree in Computer Science from Helsinki University of Technology (HUT). His areas of expertise include HACMP and RS/6000 SP. He also works as a team leader for HACMP technical groups in Nordic countries. He is a co-author of the redbook *HACMP Customization Examples*.

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

Peter Kes
International Technical Support Organization, Poughkeepsie Center

IBM PPS Lab Poughkeepsie:

Deepak Advani
Mike Coffey
Dennis Jurgensen
Tim Race
Peter Badovinatz

EMEA HACMP Center of Competency:
John Easton

IBM Germany AIX Education & Training
Michael Mueller

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 189 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:
For Internet users <http://www.redbooks.ibm.com>
For IBM Intranet users <http://w3.itso.ibm.com>
- Send us a note at the following address:
redbook@vnet.ibm.com

Part 1. Introduction to High Availability and HACMP ES

Chapter 1. HACMP ES	5
1.1 What is HACMP ES?	5
1.2 Why HACMP ES?	6
To Protect Investments	6
To Support Large Clusters	6
To Enhance Event Detection of Software Problems	6
1.3 IBM High Availability Product Family	7
1.4 High Availability Infrastructure	8
1.5 Benefits of HACMP ES	9
Scalability	9
Protect Investments in HACMP for AIX	9
Reaction to Software Problems	10
Synchronized User Scripts	11
Future-Ready	11
1.6 Single Points of Failure	12
SP Switch Board and Network	13
Internal Ethernet	13
1.7 Limitations of the Current Release	15
Runs Only on RS/6000 SP	15
Partition-Bounded	15
Current Release of HACMP ES Supports 16 Nodes	16
Some Networks Are Not Supported	16
Some Software Is Not Supported	16
Limits In Clinfo	16
Chapter 2. Differences Between HACMP and HACMP ES	17
2.1 Look and Feel Compared to HACMP	17
Looks and Feels Like HACMP for AIX	17
Functions That Differ	17
2.2 Functional Similarities with HACMP	20
SMIT	20
VSM Interface	20
Clstat	20
HAView	21
Clinfo	21
2.3 Functional Similarities with HACMP (cont'd)	22
Dynamic Automatic Reconfiguration Events (DARE)	22
Cluster Single Point of Control (C-SPOC)	22
Command Execution Language (CEL)	23
Lazy Update	23
Global ODM	23
Snapshot	23
2.4 Functional Similarities with HACMP	25
Resource Group Definitions	25
Event Scripts	25
2.5 HACMP ES Event Processing	26
Event Notification	26
Pre- and Post-Event Scripts	26
Event Recovery	26
2.6 Functional Differences with HACMP	27
Heartbeat in Topology Services	27

Membership Protocol in Group Services	27
Event Detection	27
Naming	28
No forced stop of cluster	28
Chapter 3. Components and Their Relationships	29
3.1 HACMP ES Components	29
3.1.1 HACMP ES Subsystem Components	29
3.1.2 Other Related Components	30
3.2 HACMP ES Component Definitions	31
3.2.1 Group Services/ES	31
3.2.2 Topology Services/ES	31
3.2.3 HACMP ES Cluster Manager	32
3.3 Logical Connection to PSSP Components	33
3.4 PSSP Component Definitions	34
Event Management	34
3.5 Multiple Clusters in a Partition	35
3.6 Logical Connections to Client Components	36
3.7 Client Interface Component Definitions	37
3.7.1 The Cluster SNMP Agent	37
3.7.2 Cluster Information Services	37
3.8 Topology Services Layer	38
Non-IP Networks Are Used by HACMP ES through Topology Services/ES	40
Add adapter interface definition on HACMP ES	40
3.9 Group Services Layer	41
3.10 Cluster Manager Layer	43
HACMP ES Cluster Manager	43
3.11 Start Sequence	44
Topology Services/ES Builds a List of Nodes and Adapters	46
Chapter 4. HACMP ES Event Management	47
4.1 Event Management Flow	47
4.2 Event Mapping	50
4.3 Event Mapping Files	51
4.3.1 Rules File	51
4.3.2 Event Mapping Example	53
4.4 Recovery Program Structure	54
4.4.1 Recovery Programs	54
4.4.2 Recovery Programs Example	57
4.4.3 Synchronization of Recovery Programs	58
4.5 User-Defined Event Detection	60
User-defined Barrier Points	61
4.6 Event Priority	63
Chapter 5. HACMP ES Protocols	67
5.1 HACMP ES Protocols	68
5.1.1 Membership (join/leave)	68
5.1.2 Voting	68
5.1.3 Barrier	69
5.1.4 Cbarrier	69
5.1.5 Adapter Membership State	70
5.2 Node Joining	71
5.2.1 First Node Joining	71
5.2.2 Additional Node Joining	74
5.3 Node Departure and Rejoining	77

Chapter 6. Planning	79
6.1 Planning Hardware Configuration	79
Like an HACMP for AIX Configuration	79
Non-IP Serial Network	80
Network Adapters	80
SP Switch	80
6.2 Planning Software Configuration	82
6.2.1 Prerequisites	82
6.2.2 Planning Applications	83
6.2.3 Planning Clients	83
6.3 Planning Node Relationships	85
Supported Node Relationships	85
Node Relationships Not Supported	85
6.4 Planning Different Scenarios	86
Single or Multiple Frames?	86
How Many Clusters? How Big?	87
6.5 Planning for Coexistence with Other High Availability Products	88
6.5.1 Coexistence with HACMP for AIX and HAGEO	88
6.5.2 Coexistence with HACWS	90
6.5.3 Coexistence with HANFS	90
6.5.4 Coexistence with RVSD	91
6.5.5 Coexistence with GPFS	92
Chapter 7. HACMP ES Installation and Customization	93
7.1 Installation	93
7.1.1 Installation Prerequisites	93
Prerequisites	93
Overview.	95
HACMP ES Modules	96
7.1.2 Installation Steps	98
7.2 HACMP ES Customization	100
7.2.1 Customization	100
7.2.2 Steps to Define Customized Events	102
User Can Add Own Events	102
New Events Not Predefined in PSSP	104
7.2.3 Event Definition Examples	105
7.3 Migrating from Existing HACMP Environment to HACMP ES	107
Snapshot	107
Migration	107
HACMP Scripts	108
Node Names	108
Chapter 8. Configuration Examples	109
8.1 Takeover Scenarios	109
8.1.1 Rotating Failover Principles	109
8.1.2 Rotating Takeover Scenario	111
8.2 Cascading Configurations	114
8.2.1 Cascading Failover Principles	114
8.2.2 Cascading Configuration	115



Chapter 1. HACMP ES

This chapter gives you an overview of the new HACMP ES product and its relationship to other High Availability products.


1.1 What is HACMP ES?

RS/6000

What Is HACMP ES?




HACMP Enhanced Scalability




Scalable high availability cluster for RS/6000 SP

- ◆ Enhancement to HACMP
- ◆ Exploits new features of HACMP 4.2.1
- ◆ Up to 16 nodes (in current release)

 **POWERparallel Systems**

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



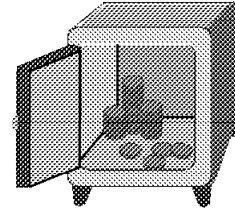
High Availability Cluster Multiprocessing Enhanced Scalability (HACMP ES) is an enhancement to RS/6000 SP. Its basic functions are equal to those of the other IBM high availability product, HACMP for AIX. HACMP ES is an enhancement to HACMP for AIX, but it uses IBM Parallel System Support Program (PSSP) Group Services for event detection and heartbeat, whereas HACMP for AIX has its own internal mechanisms. HACMP ES includes the new features of HACMP for AIX V4.2.1. As new functions are added to the HACMP for AIX product, they will be added to the HACMP ES product. The current release, 4.2.1, supports scalability for up to 16 nodes.

1.2 Why HACMP ES?

RS/6000

Why HACMP ES?

- To protect investments in growing environments
- To support large clusters
- To enhance event detection to software problems



ITSO Poughkeepsie Center

© Copyright 1997 IBM Corporation



To Protect Investments

HACMP ES uses the same hardware configuration, the same definitions, and the same event scripts as HACMP for AIX. The skills learned with HACMP are valid with HACMP ES, minimizing education cost and time. Skills used for tailoring event scripts are reused. Because HACMP ES exploits our strategy in high availability, it provides a protected path to the future, that is, it protects customers' investments to clusters in RS/6000 SP while their needs grow.

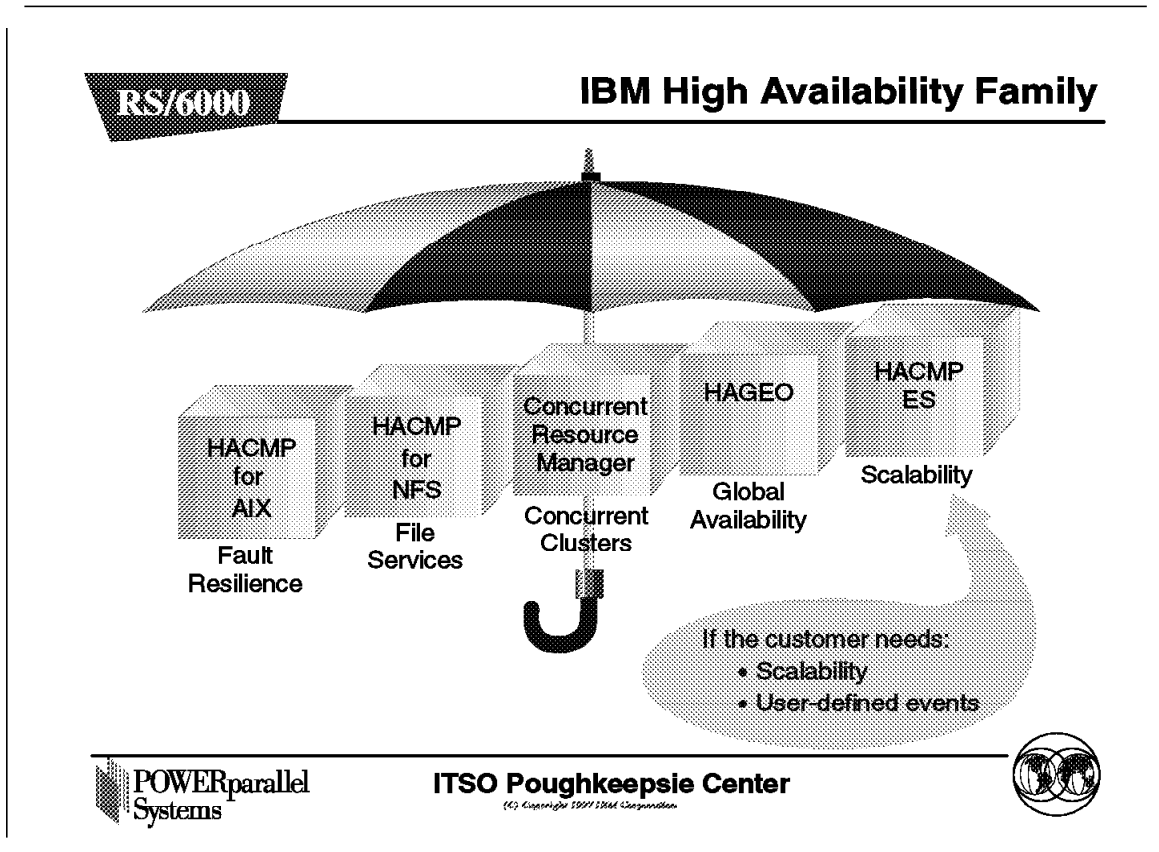
To Support Large Clusters

When a customer has a need for *large numbers of nodes* to be made highly available, HACMP ES provides the solution for this demand.

To Enhance Event Detection of Software Problems

HACMP ES uses IBM High Availability Infrastructure in IBM Parallel System Support Program (PSSP) to detect software problems. The user can define events to HACMP ES, that PSSP monitors any software or hardware component in the system, he is interested in. PSSP Event Management sends events to HACMP ES, which reacts to them in a predefined way. Thus, the user has an easier way to spread the cluster to cover problems in applications and middleware. For more information about event detection, see Chapter 4, "HACMP ES Event Management" on page 47, 7.2.2, "Steps to Define Customized Events" on page 102, 7.2.3, "Event Definition Examples" on page 105, and Chapter 11, "User-Defined Events" on page 127.

1.3 IBM High Availability Product Family

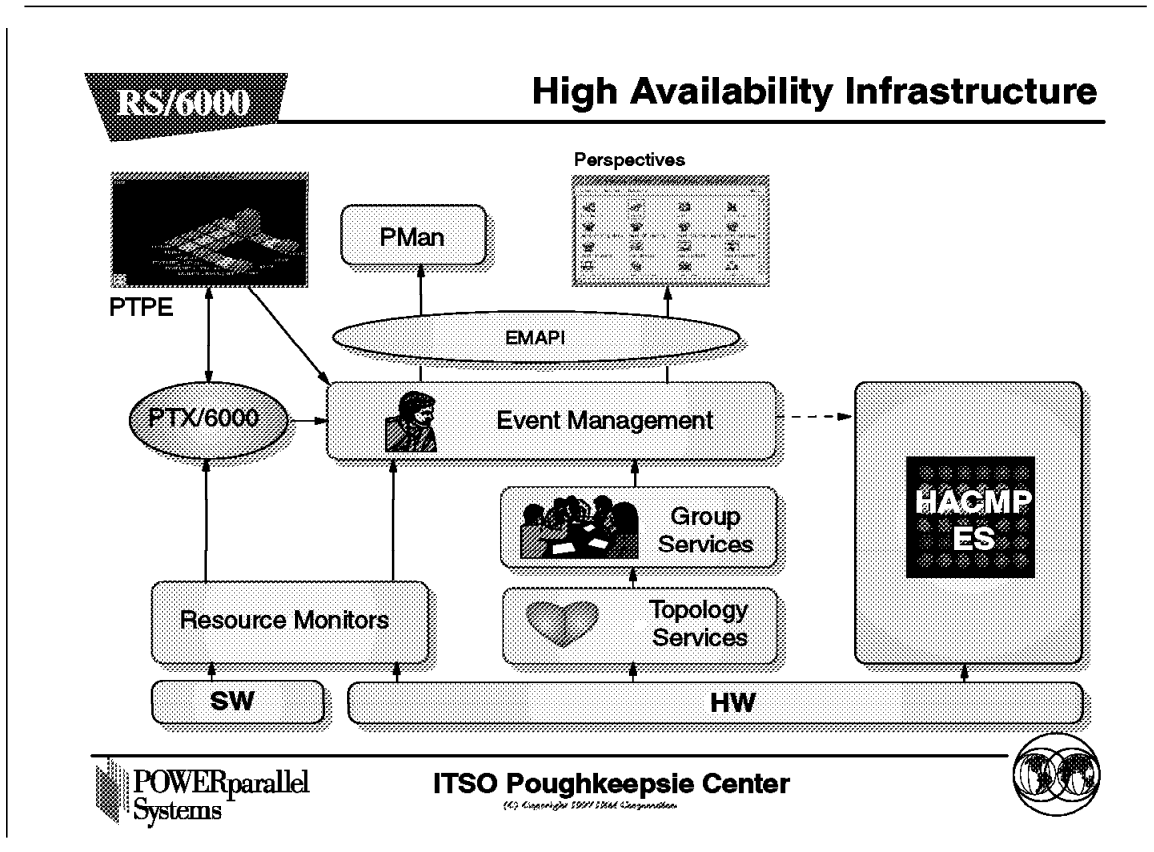


IBM has a wide range of high availability products for the RS/6000 platform. Together, these products offer a full solution for customers who need to maintain a highly available system.

HACMP for AIX is the main product in this family, providing a general purpose solution. In most cases, it fills customers' needs. If not, a customer can choose additional products from this family.

This release of HACMP ES is recommended for customers who have a need for user-defined events and scalability; especially new customers in the RS/6000 SP environment.

1.4 High Availability Infrastructure



High Availability Infrastructure is an architecture that was first implemented in the IBM Parallel System Support Program (PSSP) product. It includes necessary components for providing a basic high availability platform to its client applications. There are some applications that already use this infrastructure, for example, PTPE, VSD/RVSD, and now HACMP ES. This infrastructure is also planned to be used as a basis on other hardware platforms. High Availability Infrastructure is described in detail in the redbook *RS/6000 SP High Availability Infrastructure*, SG24-4838.

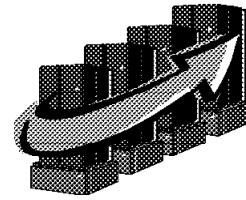
1.5 Benefits of HACMP ES

RS/6000

Benefits of HACMP ES

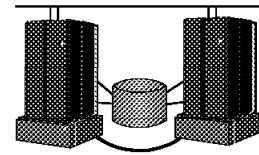
➤ Scalability over eight nodes

- ◆ Now up to 16 nodes
- ◆ Later up to 128 nodes



➤ Protect HACMP for AIX investments

- ◆ Same hardware configuration
- ◆ Same definitions and event scripts



ITSO Poughkeepsie Center

© Copyright 1997 IBM Corporation



Because HACMP ES is part of the IBM high availability product family, its nature is very close to that of HACMP for AIX. There are some differences, however. HACMP ES offers additional value to the user, as follows.

Scalability

The first release of HACMP ES scales up to 16 nodes, compared to eight by HACMP for AIX. Its implementation is designed to handle up to 128 nodes. This makes it possible to build large clusters for large RS/6000 SP environments.

For example, large SAP environments might have tens of nodes working as application servers. Some software has to know the status of each application server node at all times in order to provide high availability and good load balancing for this kind of environment. It has to be able to react to changes in the status of software and hardware components. HACMP ES does this.

The product supports both uni- and multi-processor nodes.

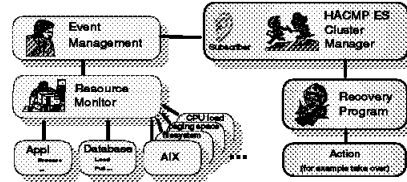
Protect Investments in HACMP for AIX

HACMP ES uses the same hardware configuration as HACMP for AIX. Thus, if a customer needs to move from HACMP for AIX to HACMP ES, the existing hardware can be used.

HACMP ES uses the same cluster definitions and event scripts as HACMP for AIX V4.2.1. Thus, a customer does not have to reinvent the wheel and invest in redefinitions or recustomizations.

➤ Reaction to software problems

User-defined event from applications to HACMP ES

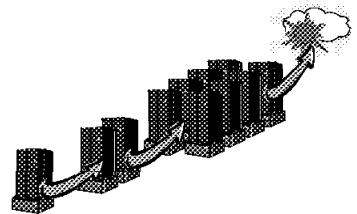


➤ Synchronized user scripts

Better way to synchronize user scripts with barriers

➤ Future ready

Already exploits future main clustering technology!



Reaction to Software Problems

With HACMP ES and PSSP, users can define events that they want HACMP ES to react to.

HACMP ES Cluster Manager gets events from Group Services and Event Management. Standard HACMP ES events come from Group Services; user-defined events come from Event Management. Group Services uses Topology Services to detect changes in hardware. Event Management gets events from the Resource Monitor, which monitors all kinds of parameters in hardware or software components. One Cluster Manager subscribes Event Management to get events and distributes them to the other nodes.

The user can define the Resource Monitor to monitor any component in the system. Event Management has a long list of pre-defined events that can easily be mapped to Cluster Manager to subscribe. Here are some examples of what the monitored parameters can be, and the conditions that can be set for them:

- File system /tmp in Node 4 is over 95% full
- CPU load on Node 3 is over 97%
- Count of process dbserve owned by user heidi on Node 2 is less than 3

Also see 7.2.2, "Steps to Define Customized Events" on page 102, 7.2.3, "Event Definition Examples" on page 105, and Chapter 11, "User-Defined Events" on page 127.

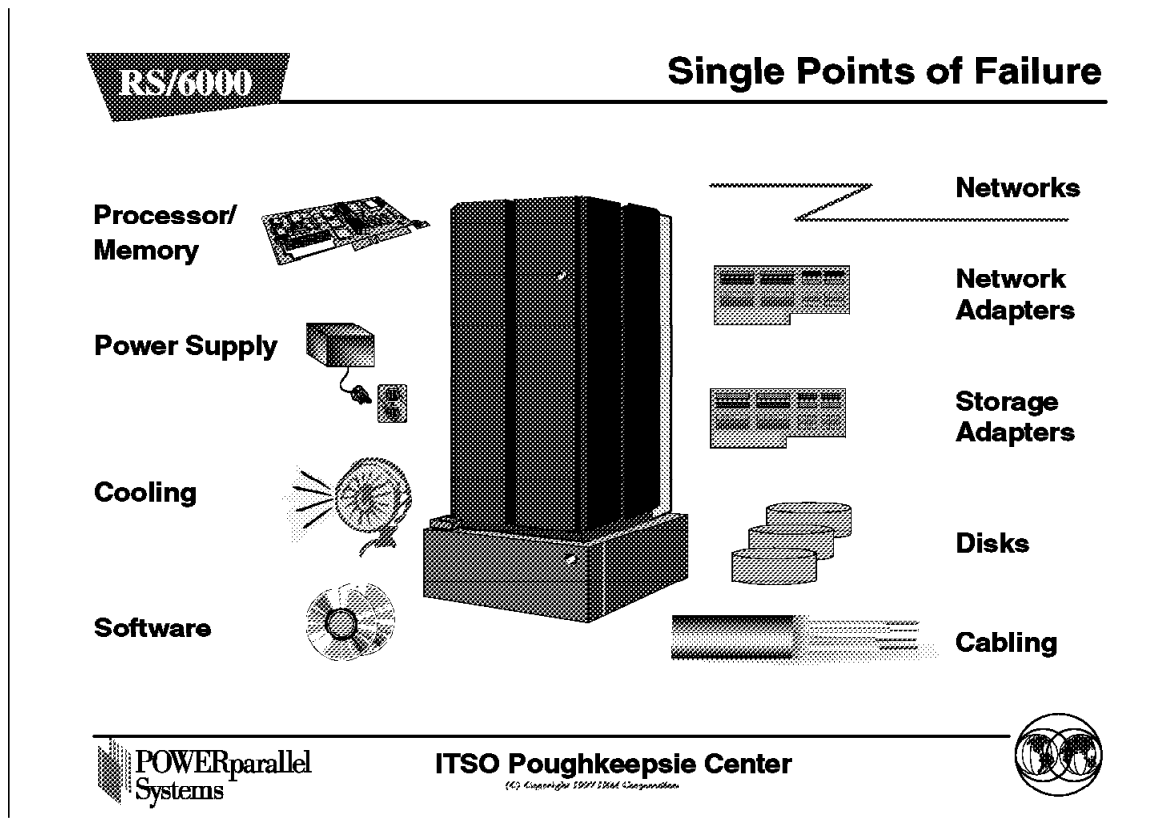
Synchronized User Scripts

HACMP ES provides an easy and reliable way for users to synchronize their own scripts: they can add barrier commands to their scripts as synchronization points. In these points, all nodes in a cluster wait until all nodes have reached the same stage. For more details, see Chapter 11, “User-Defined Events” on page 127.

Future-Ready

HACMP ES already uses the future technology to provide high availability clusters. This technology is planned to be available on platforms other than RS/6000 SP. If a customer selects HACMP ES today, he will not have to migrate his system later.

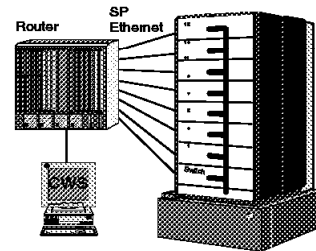
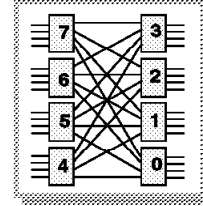
1.6 Single Points of Failure



In single systems there are many components that can prevent users from getting service from the server if one component fails. This component can be any kind of hardware or software. High availability clusters are designed to duplicate and manage most of the critical components. Normally this covers all hardware and part of the software. Most components are quite easy to cover with products such as HACMP ES, but some are more difficult. In some cases their failure probability is so low and the cost of coverage so high that customers are not willing to invest in this.

Sometimes customers want to try to eliminate all single points of failure. With HACMP ES, these points are the same as with HACMP.

- Same as with HACMP for AIX
- SP Switch board and network
 - ✦ Secondary network to take over (FDDI)
- Internal Ethernet
 - ✦ Needed to run VSD/RVSD
 - ✦ Solution: HACWS and routed Ethernet



SP Switch Board and Network

Because the SP Switch cannot be duplicated, it is clearly a potential single point of failure. It is designed to be very robust and reliable with some internal redundant components and paths.

An alternate path for the switch network can be used with HACMP ES. This path must be as fast as possible, and the take-over of this network has to be designed very carefully. In particular, routing issues for client nodes might be tricky. The FDDI network is a potential solution for this. Since ATM is not yet supported with HACMP ES, it cannot be considered.

Internal Ethernet

The internal Ethernet network in RS/6000 SP is mainly used for administration. In most cases it is not considered a critical component for production. In some cases, as with VSD/RVSD, it is needed to keep production going. In a standard thin Ethernet network, as in the basic configuration, one failed or loose connector can stop the whole network. This problem can be eliminated by using a routed Ethernet network where all nodes are on their own physical segment but in the same subnet.

Single Points of Failure Solution Matrix

Single Point of Failure	Solution
Node <ul style="list-style-type: none"> * Processor, memory etc. * Operating system, other SW * Internal disk or disk adapter * External data disk * External disk adapter * Network adapter * Switch adapter * Admin. Ethernet adapter 	HACMP ES: take over HACMP ES: take over AIX LVM: mirroring AIX LVM: mirroring AIX LVM: mirroring HACMP ES: adapter swap HACMP ES, PSSP: shut down node, take over HACMP ES with routed network: take over HACMP ES with multiple networks: routing
Network	
Switch <ul style="list-style-type: none"> * SP Switch board * "Eprimary" 	HACMP ES: dual network take over (FDDI) PSSP
Frame <ul style="list-style-type: none"> * Admin. Ethernet * RS232 serial link * Supervisor card * Power supply * Power cord 	HACMP ES: routed Ethernet and HACWS Hardware Hardware Hardware, N+1 Hardware with two frames
Control Workstation	HACWS



This matrix shows most of the basic methods for eliminating single points of failure component by component. More detailed descriptions can be found in the HACMP ES and HACMP for AIX manuals, in other chapters of this book, and in the redbook *Implementing High Availability on RS/6000 SP*, SG24-4742.

1.7 Limitations of the Current Release

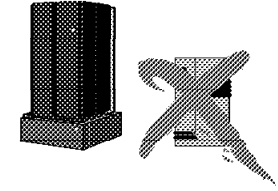
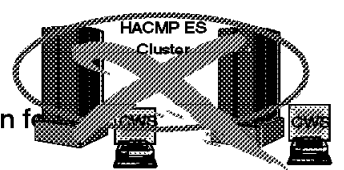
RS/6000


Limitations of Current Release

- **Runs only on RS/6000 SP**
Strategy to support other RS/6000 models

- **Partition bounded**
 - ❖ One SP system per cluster
 - ❖ One partition per cluster
 - ❖ Not integrated with HAGEO to provide full solution for disaster recovery


- **First implementation supports 16 nodes**
Max. 32 boot, 32 service and 32 standby interfaces per node (important only in large rotating clusters)



**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



The first release of HACMP ES has some limitations, as follows. Some of these are expected to change in time.

Runs Only on RS/6000 SP

HACMP ES uses PSSP Group Services for event detection and heartbeat. Therefore, it runs only on the platform on which Group Services runs. At the moment, RS/6000 SP is the only supported platform.

There are plans to use the same technology on some other platforms as well. The other RS/6000 models are the most probable platforms for extending this technology.

Partition-Bounded

The lower layers HACMP ES uses, such as Group Services and Topology Services, are partition-dependent layers. For example, the Group Services daemon hagsd can work only on one partition. A HACMP ES cluster can use only one instance of Group Services, and thus can work only on one partition. This means that HACMP ES clusters can work *only on a single RS/6000 SP system*.

Because HACMP ES works only on a single RS/6000 SP, it cannot be used with HAGEO to provide a full solution for disaster recovery. HACMP for AIX and HAGEO have to be used.

Current Release of HACMP ES Supports 16 Nodes

The first release of HACMP ES supports up to 16 nodes.

The current release supports up to 32 boot, 32 service and 32 standby interfaces per node. This limitation is important only when running large rotating clusters, in which all interfaces have to be defined to all nodes. For example, a rotating resource group with 16 member nodes requires 1 boot and 1 service address on each node, and if you have more than 32 of these resource groups, you will exceed the capacity limitations.

Some Networks Are Not Supported

ATM networks are not supported in the current release.

Point-to-point networks can only be used to provide an alternate path for heartbeat traffic if the public network fails. The public network here is a Token-Ring, Ethernet, FDDI, SP Switch or HPS switch. This point-to-point network is a non-IP network to provide connection when a TCP/IP protocol stack fails. IP protocol is not supported on point-to-point networks.

Some Software Is Not Supported

HANFS is not supported with HACMP ES on the same node. This is true also for HACMP for AIX.

A concurrent access type of node relationship is not supported with the current release of HACMP ES. Thus, HACMP Cluster Lock Manager is not supported either, as it is used only for concurrent access.

In the current release, Dynamic Automatic Reconfiguration Events (DARE) does not support topology reconfiguration.

Limits In Clinfo

Client Information Program (Clinfo) also has some limits. They are the same as for HACMP. For the current release, they are:

- 16 clusters in one Clinfo
- 128 nodes per cluster
- 128 interfaces per node

Chapter 2. Differences Between HACMP and HACMP ES

This chapter contains foils showing the differences between HACMP for AIX V4.2.1 and HACMP ES.

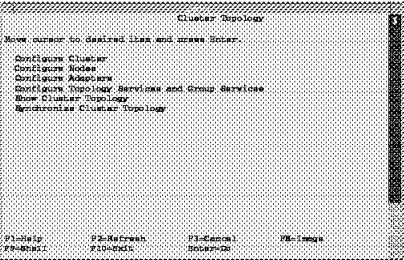
2.1 Look and Feel Compared to HACMP


RS/6000

Look and Feel Compared to HACMP

- **Basically looks and feels like HACMP for AIX**
 - ❖ Mostly same SMIT panels
 - ❖ Same xhacmpm


- **Some functions are in different places**
 - ❖ User-defined events in PSSP Event Management and rules.hacmprd file
 - ❖ Heartbeat parameters in Configure Topology Services and Group Services SMIT panel





POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



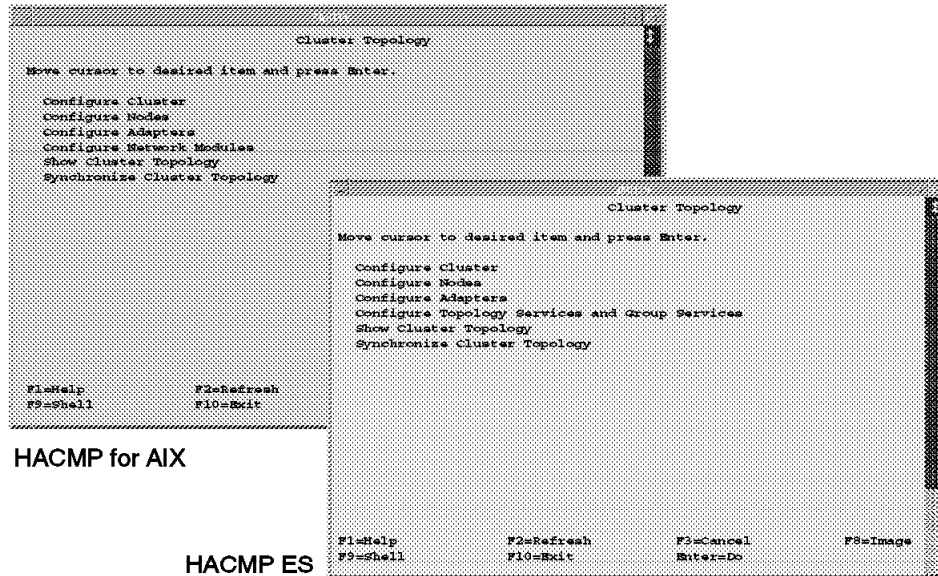
Looks and Feels Like HACMP for AIX

Basically, HACMP ES looks and feels like HACMP for AIX. The SMIT panels are mostly the same, and so is the xhacmpm management tool. The definitions and event scripts are identical. Some tools, such as C-SPOC, are equal. The scalability is, likewise, limited to eight.

Functions That Differ

Because HACMP ES uses PSSP Group Services and HACMP for AIX has its own internal function to detect events and its own heartbeat protocol, some functions are different. User-defined events to react to software problems, for example, can be implemented by using Event Management and Resource Monitor and the rules.hacmprd file.

The heartbeat is part of PSSP. Its rate can be tuned in a new HACMP ES SMIT panels called Configure Topology Services and Group Services. The default values for Topology Services are shown in 7.2.1, "Customization" on page 100.



SMIT panels that define cluster topology vary somewhat. The differences lie in how to define parameters related to heartbeat and networks. In HACMP for AIX these are defined in the network interface modules (NIM); in HACMP ES they are defined in Topology Services, and the chosen values take effect for all network types.

2.2 Functional Similarities with HACMP

RS/6000

Functional Similarities with HACMP

- **SMIT**
Mostly the same panels
- **VSM interface**
Object-oriented interface called xhacmpm
Up to eight nodes
- **Clstat**
Up to 32 nodes per cluster can be monitored
- **HAView**
- **Clinfo**
Same code



ITSO Poughkeepsie Center

© Copyright 1997, IBM Corporation



SMIT

The SMIT screens for HACMP are mostly the same. The structures in HACMP for AIX V4.2.1 and HACMP ES are identical. There are some detail differences. For example, the Configure Network Modules are replaced by Configure Topology Services and Group Services (this example is shown in the second picture of 2.1, "Look and Feel Compared to HACMP" on page 17).

VSM Interface

The VSM interface (xhacmpm) in this release of HACMP ES is the same as in HACMP for AIX V4.2.1. Therefore, it is limited to an 8-node cluster.

The VSM interface is sometimes called the Drag and Drop GUI.

Clstat

The clstat utility is basically the same as in HACMP for AIX V4.2.1. The HACMP ES version of this release supports up to 32 nodes (an example is shown in the picture Clstat Compared to HACMP on page 19) and is downward compatible. This means you can use it to get a view of all your HACMP installations.

HAView

The HACMP ES version of HAView is functionally the same as the version of HACMP for AIX V4.2.1.

HAView allows you to monitor HACMP clusters through the NetView network management platform, using SNMP. HACMP provides a management information base (MIB) that contains information about cluster topology and state. HAView displays the configuration and state of the clusters and cluster components through the NetView graphical user interface. HAView allows you to search through a series of nested submaps that reflect the state of all the nodes, networks, and network addresses configured in a particular cluster.

Clinfo

The clinfo daemon of HACMP ES is based on the same code as the HACMP for AIX V4.2.1 version.

2.3 Functional Similarities with HACMP (cont'd)

RS/6000 Functional Similarities with HACMP (cont'd)

- **DARE** (Dynamic Automatic Reconfiguration Events)
Except topology reconfiguration
- **C-SPOC** (Cluster Single Point of Control)
Same code
Up to eight nodes
- **CEL** (Command Execution Language)
- **Lazy update**
- **Global ODM**
Uses same ODM classes
- **Snapshot**



ITSO Poughkeepsie Center

© Copyright 1997, IBM Corporation



Dynamic Automatic Reconfiguration Events (DARE)

In this version, the topology-related definitions, that is, nodes, networks, and adapters, cannot be reconfigured dynamically as in HACMP.

Dynamic reconfiguration allows the user to change the configuration of a running cluster, that is, the definitions of cluster resources can be changed. These changes take effect immediately, without having to stop and restart the HACMP daemons, and without having to disrupt the applications running on the cluster.

Cluster Single Point of Control (C-SPOC)

The C-SPOC in HACMP ES is functionally the same as in HACMP for AIX V4.2.1 because HACMP ES uses the same code as HACMP for AIX V4.2.1. This is also the reason why it is limited to an 8-node cluster. C-SPOC uses the CEL to enable these functions to the user.

C-SPOC enables the user to perform certain common administrative operations across the cluster from a single SMIT session. These operations are:

- Starting and stopping HACMP
- Adding, changing, and deleting users and groups
- Operating on shared volume groups

For the last set of operations, the C-SPOC facility removes the need to manually synchronize the change across the cluster. In this version, this function is only available for a cluster with eight or less nodes.

Command Execution Language (CEL)

The CEL is part of the C-SPOC modules. It is also the same code that we mentioned in “Cluster Single Point of Control (C-SPOC)” on page 22. Therefore, it is also limited to an 8-node cluster.

CEL is a programming language that lets you integrate C-SPOC’s distributed functionality into each ksh script the CEL preprocessor (celpp) generates. These scripts are automatically performed on all cluster nodes when you invoke them to perform administrative tasks from a single node in the cluster. Without C-SPOC’s distributed functionality, you must execute each administrative task separately on each cluster node, which can lead to inconsistent node states within the cluster. More information about CEL can be found in the *HACMP Administration Guide*.

Lazy Update

The Lazy Update functionalities of HACMP ES are identical to those of HACMP for AIX V4.2.1. The function scales with HACMP ES but is limited to eight nodes by the disk (SSA) limitations.

Lazy Update is an attempt to fix certain HACMP data-related problems without the need to schedule cluster downtime.

Whenever a change is made to a volume group, like adding a new disk, the cluster nodes that do not have that volume group varied on do not know of this change. Lazy Update is a technique to resolve this problem. When a cluster node needs to take over a resource group, it checks to see if any time stamps differ between the VGDA’s and the saved time stamp (in `/usr/sbin/cluster/etc/vg`). If they are the same, the vary on occurs normally. If the time stamps differ, the volume group must have had a change applied to it and, therefore, the `exportvg`, `importvg`, and `chvg` commands are automatically executed to bring this cluster node’s definitions up-to-date.

In fact, the first time a cluster node takes ownership of a resource group it will always execute this sequence of commands, since the time stamp file needs to be initialized.

Global ODM

Global ODM is equal to the ODM classes used by HACMP. HACMP ES uses the same ODM classes as HACMP for AIX V4.2.1.

Snapshot

The snapshot function in HACMP ES is identical to that in HACMP for AIX V4.2.1. Snapshots created in HACMP for AIX V4.2.1 can be used in HACMP ES. This compatibility can be used for migration (for more information, see Chapter 10, “Migration” on page 123).

The Cluster Node Snapshot captures a cluster configuration, creating text files that contain all the information necessary to configure a similar cluster. Once captured, these snapshots - created in ASCII text format - can be applied to a new cluster. Cluster Node Snapshot can also be used in conjunction with the HACMP graphical user interface (VSM): any snapshot can be viewed as a

simple text file or as a graphical representation, enabling quick analysis and diagnosis.

2.4 Functional Similarities with HACMP

RS/6000

Functional Similarities with HACMP (cont'd)

➤ Resource Group definitions

- ✦ Nodes
- ✦ Disks
- ✦ File systems
- ✦ Volume groups
- ✦ Applications
- ✦ Network address

➤ Event scripts

- ✦ Same pre/post event scripts
- ✦ If script works with HACMP for AIX V 4.2.1, it works with HACMP ES



ITSO Poughkeepsie Center

© Copyright 1997, IBM Corporation



Resource Group Definitions

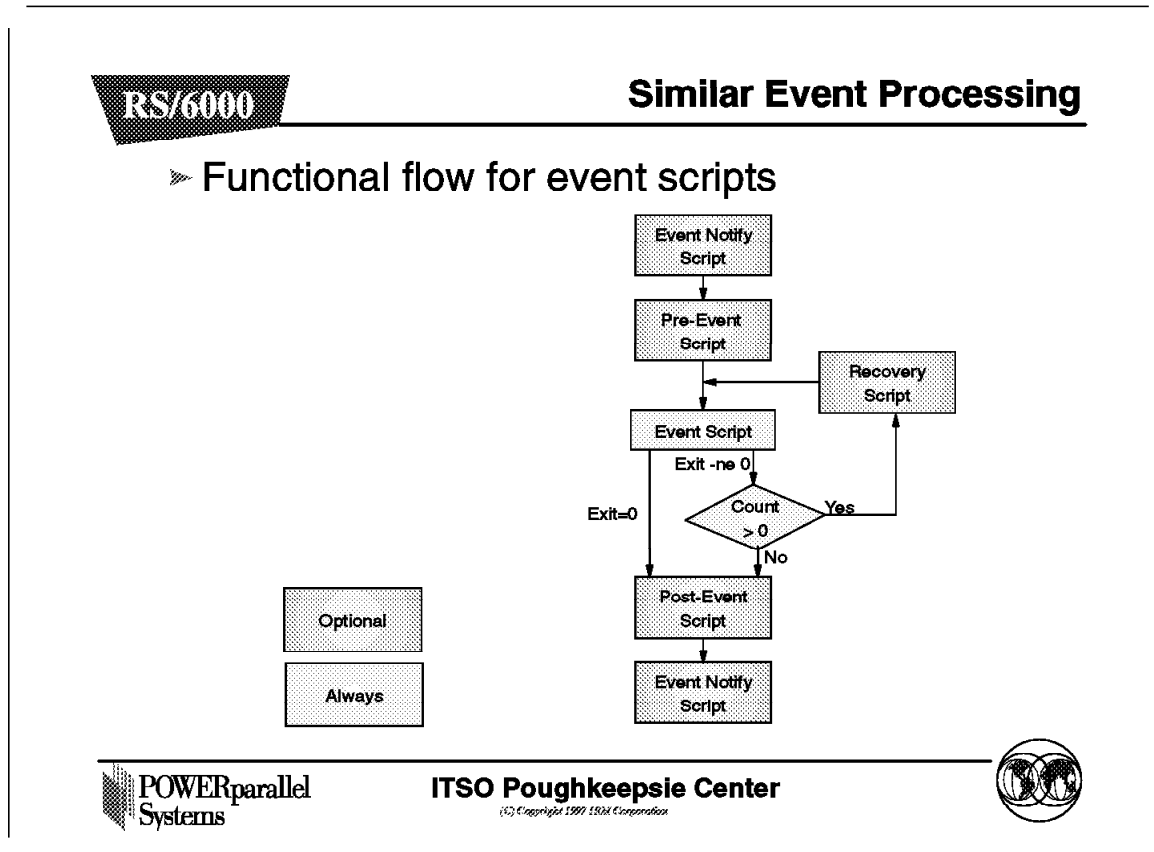
The resource group definitions are the same as in HACMP. They are as follows:

- Resource groups
- Application servers
- Resources for a resource group
 - A resource can be the service IP label, a file system, a volume group, an application, and so on.
- Run time parameters
- Cluster events
- Cluster Lock Manager resource allocation

Event Scripts

The event scripts delivered with HACMP ES are the same as in HACMP for AIX V4.2.1. All scripts (pre- or post-event or application scripts) that run in HACMP for AIX V4.2.1 also run in HACMP ES. Some scripts (pre- or post-event or application scripts) from an earlier HACMP for AIX installation may require modifications, but most will work without any modification.

2.5 HACMP ES Event Processing



HACMP ES events are processed using the flow shown in the figure. The notification, pre-event, post-event, and recovery scripts are optional and can be specified by the user during configuration. This is the same procedure as for HACMP. In HACMP ES, the utility program `clcallev` is used to execute the event scripts with this model.

Event Notification

The user can specify a notify command that sends mail to the system administrator to indicate that an event is about to happen or has just occurred. The message can contain success or failure information of the event script.

Pre- and Post-Event Scripts

These are user-defined scripts that execute specific commands before and after the Cluster Manager calls an event script.

Event Recovery

The user can specify a command that attempts to recover from an event script failure. If the recovery command succeeds and the retry count for the event script is greater than zero, the event script is rerun. The number of times to retry the recovery command can be specified.

RS/6000

Functional Differences with HACMP

- **Heartbeat in Topology Services**
 - ◆ One heartbeat rate for all network types
- **Membership protocol now in Group Services**
 - ◆ In HACMP for AIX inside Cluster Manager
- **Event detection**
 - ◆ Through Group Services and Event Management
- **Naming**
 - ◆ Node name must be in the /etc/hosts file or in a DNS server
- **No forced down option to stop cluster**
 - ◆ SMIT panel or clstop command



ITSO Poughkeepsie Center

© Copyright 1997, IBM Corporation



Heartbeat in Topology Services

In HACMP ES the heartbeat is now handled by Topology Services. The Network Interface Modules (NIM) that were part of HACMP for AIX are replaced by Topology Services, for which only one heartbeat error detection rate can be defined. A change of the values takes effect on all network types. Tuning for only one network type is no longer available. Examples of the values you can use are shown in 7.2.1, “Customization” on page 100.

Membership Protocol in Group Services

Membership and its protocols are now handled by Group Services. In HACMP for AIX this was done by Cluster Manager. Everything was hidden from the user. With Group Services you have more possibilities to add specific steps. For more information about the protocols, see Chapter 5, “HACMP ES Protocols” on page 67.

Event Detection

Event detection is now handled by Group Services and Event Management. In HACMP for AIX it was handled by Cluster Manager and was hidden to the user. For more information about event management, see Chapter 4, “HACMP ES Event Management” on page 47 This change allows you to add user-defined events. For more information about user events, see 4.5, “User-Defined Event Detection” on page 60 and Chapter 11, “User-Defined Events” on page 127.

Naming

In HACMP ES the HACMP node name must be resolvable to the hostname, that is, it must be defined as an alias to the hostname in the `/etc/hosts` file or in the Domain Name Server (DNS), or set equal to the hostname.

This is necessary because Topology Services uses information out of the SDR. These data can only be found if there is a way to figure out on which host HACMP is started. The information out of the SDR are used to build the *machines.lst* file. If this file can not be build HACMP will not start.

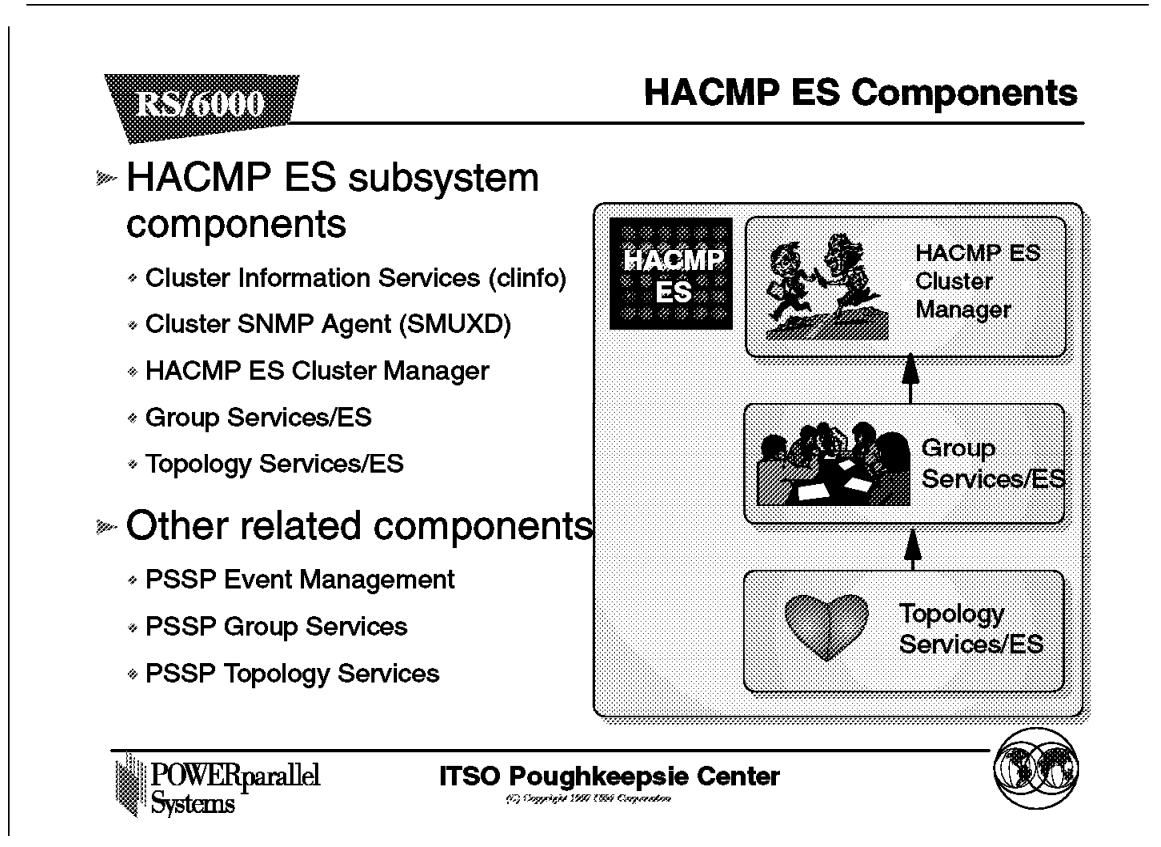
No forced stop of cluster

The forced shutdown is not available in this Version of HACMP ES. It can not be used by using the SMIT panels nor by using the command line. This function will be available in a following version.

Chapter 3. Components and Their Relationships

This chapter describes the HACMP ES components, such as Topology Services/ES, Group Services/ES, the HACMP ES Cluster Manager, and their relationship to the PSSP components and to each other.

3.1 HACMP ES Components



The above figure shows a model of an HACMP ES instance with the key subsystem components that run on the node shown as distinct layers.

3.1.1 HACMP ES Subsystem Components

The HACMP ES cluster is made up of the following components:

- Cluster Information Services

Cluster Information Services (clinfo) is a daemon that exploits SNMP and makes cluster status information available to applications using the API. For more information, see 3.7.2, "Cluster Information Services" on page 37.

- Cluster SNMP Agent

Cluster SNMP Agent (SMUXD) is the SNMP subagent for HACMP ES. It maintains information about HACMP ES status. For more information, see 3.7.1, "The Cluster SNMP Agent" on page 37.

- HACMP ES Cluster Manager

The HACMP ES Cluster Manager provides an infrastructure that drives events that execute the HACMP shell script. For more information, see 3.2.3, “HACMP ES Cluster Manager” on page 32.

- Group Services/ES

Group Services/ES is a new subsystem for HACMP ES that provides event detection to Cluster Manager. It runs separately from Group Services, which is provided by IBM Parallel System Support Program (PSSP). For more information, see 3.9, “Group Services Layer” on page 41.

- Topology Services/ES

Topology Services/ES is also a subsystem of HACMP ES. It provides adapter failure event detection to Group Services/ES by exchanging heartbeats. It runs separately from PSSP Topology Services. For more information, see 3.8, “Topology Services Layer” on page 38.

3.1.2 Other Related Components

PSSP provides high availability services. PSSP Event Management, PSSP Group Services and PSSP Topology Services (heartbeat) are the key components for the high availability services.

- PSSP Event Management

The Event Management subsystem allows the user to define events and notifies the HACMP ES Cluster Manager when these events occur. For more information, see “Event Management” on page 34.

- PSSP Group Services

The PSSP Group Services subsystem provides a distributed coordination and synchronization mechanism to other subsystems. The Group Services/ES subsystem is based on the PSSP Group Services subsystem.

- PSSP Topology Services (heartbeat)

PSSP Topology Services is the foundation of the entire infrastructure. It coordinates adapter membership and node membership in the system. This information is in turn provided to other subsystem components, such as PSSP Group Services. PSSP Topology Services supports up to 512 nodes, but it can monitor only the administrative Ethernet network and a switch network.

For more information about PSSP components, see *RS/6000 SP High Availability Infrastructure*, SG24-4838.

3.2 HACMP ES Component Definitions

RS/6000


HACMP ES Component Definitions

➤ **Group Services/ES**


- ◆ Provides event detection
- ◆ Provides all of the distributed protocols in the system
- ◆ Requires Topology Services

➤ **Topology Services/ES**


- ◆ Provides adapter failure event detection
- ◆ Sends Keep Alive packets
- ◆ Reliable message delivery services



Group Services/ES




Topology Services/ES



POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



This figure describes HACMP ES cluster components.

3.2.1 Group Services/ES

Group Services/ES is based on PSSP Group Services; it has enhancements for HACMP ES to include adapter membership groups. It provides event detection as well as all the distributed protocols that HACMP ES exploits. Group Services/ES requires the services of Topology Services/ES, which is provided with HACMP ES. For more information, see 3.9, “Group Services Layer” on page 41.

3.2.2 Topology Services/ES

Topology Services/ES is based on PSSP Topology Services. It has enhancements to HACMP ES to support IP address takeover and various kinds of networks, such as RS232. It is a reliable message delivery service to maintain availability information about the nodes and adapters, and to provide adapter failure event detection. Group Services/ES subscribes to Topology Services/ES for changes in the availability status of nodes and adapters.

For more information, see 3.8, “Topology Services Layer” on page 38.

3.2.3 HACMP ES Cluster Manager

RS/6000 HACMP ES Component Definitions (cont'd)

➤ HACMP ES Cluster Manager

- ◆ Runs recovery actions on all nodes of the cluster
- ◆ Uses services of Group Services to detect events
- ◆ Runs on each cluster node



ITSO Poughkeepsie Center

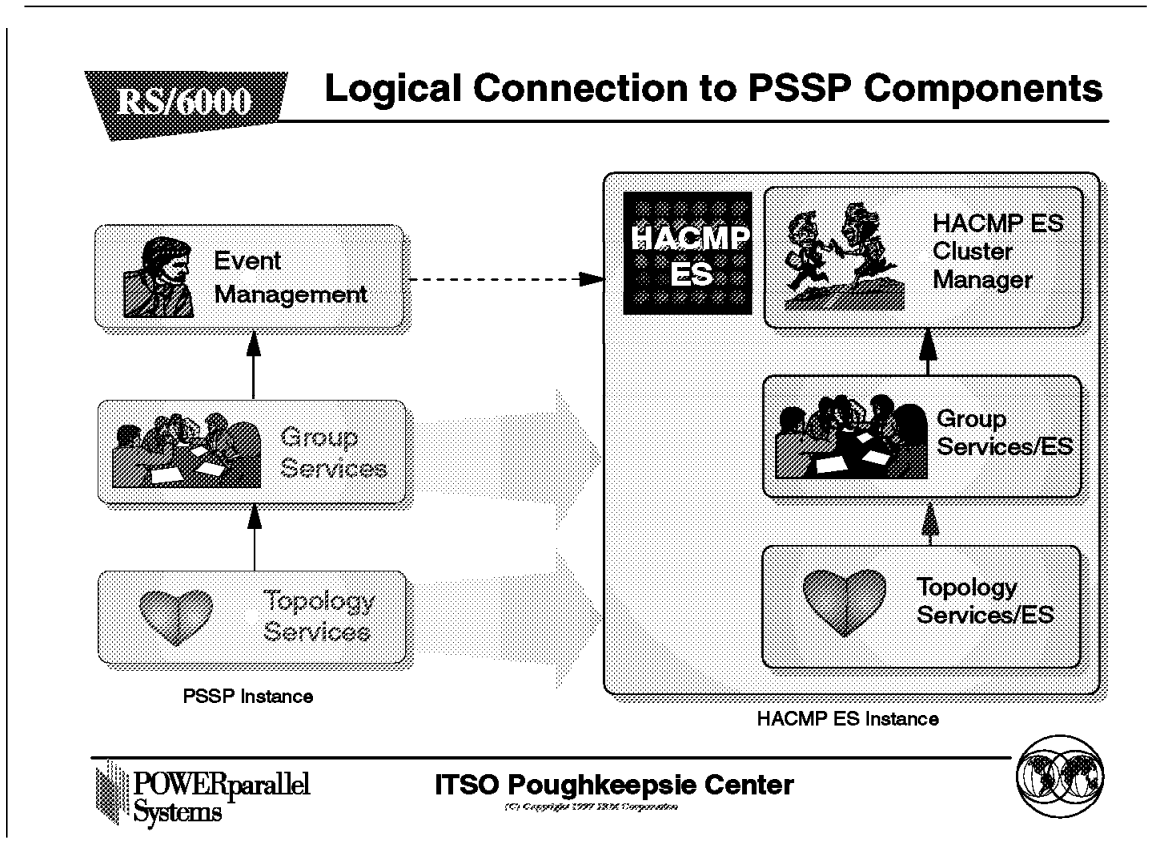
© Copyright 1999 IBM Corporation



HACMP ES Cluster Manager provides an infrastructure that drives user-defined events as well as integrated pre-defined events that drive the HACMP for AIX V4.2.1 shell script. It runs on each cluster node, and runs *recovery actions* on all nodes of the cluster. A recovery action is a recovery program that Cluster Manager runs when it detects an event. For example, when a “node up” event is detected, Cluster Manager runs the `node_up` recovery program.

HACMP ES Cluster Manager requires a mapping of an event with the associated recovery programs. The predefined events provided by HACMP ES, such as node up, are detected by Group Services/ES. All other events are triggered by an external agent, such as Event Management. For more information, see Chapter 4, “HACMP ES Event Management” on page 47.

3.3 Logical Connection to PSSP Components



To support IP address takeover and various kinds of networks, HACMP ES Instance uses two subsystems: Group Services/ES and Topology Services/ES, which are based on PSSP Group Services and Topology Services, which have enhancements for HACMP ES. Group Services/ES runs independently of PSSP Group Services, and the two do not communicate with each other. Topology Services/ES also runs independently of PSSP Topology Services, and they also do not have any way to communicate with each other. Only HACMP ES subscribes to Group Services/ES.

Event Management is only one PSSP component whose services HACMP ES exploits to detect user-defined events.


3.4 PSSP Component Definitions


RS/6000

PSSP Component Definitions

➤ **Event Management**


- ◆ Allows the user to define events
- ◆ Notifies the Cluster Manager of user-defined events





**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



This figure shows PSSP component definitions related to HACMP ES.

Event Management

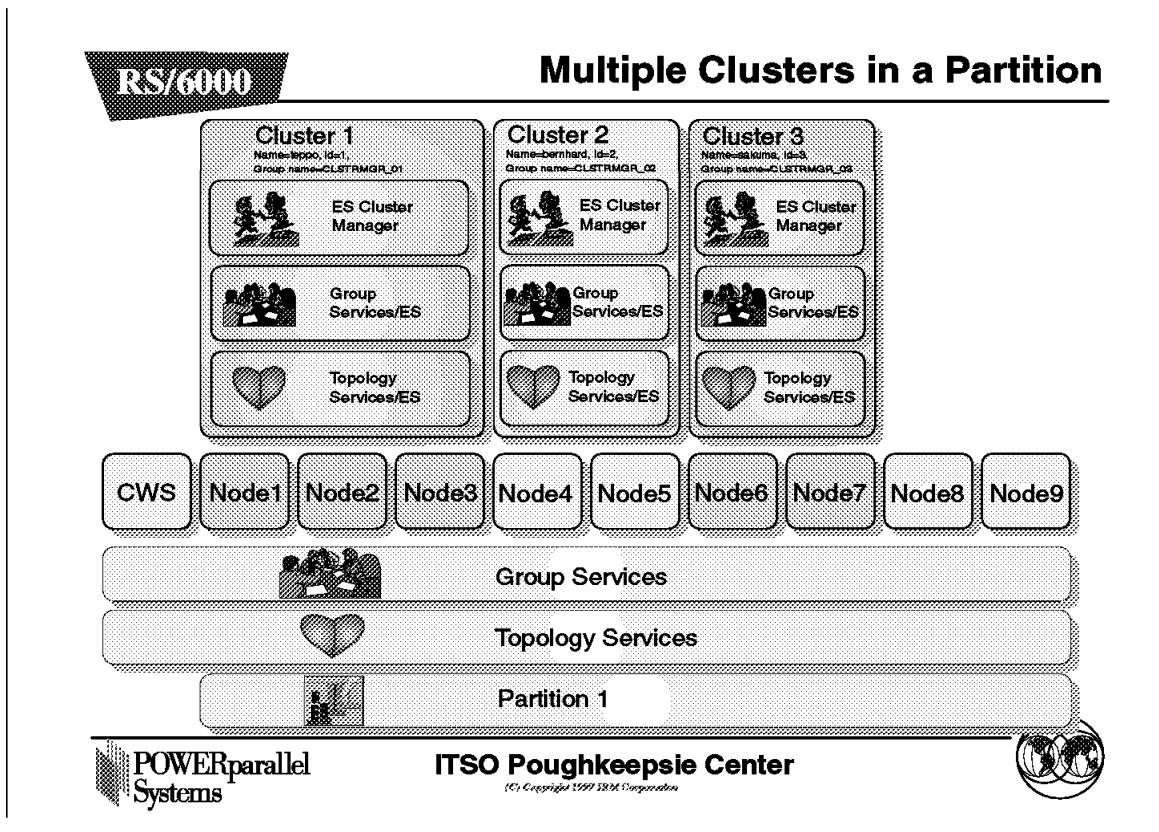
Event Management is a distributed subsystem of IBM Parallel System Support Program (PSSP) on the RS/6000 SP. It is one of several subsystems in PSSP that provide a set of high availability services.

The function of the Event Management subsystem is to match information about the state of system resources with information about resource conditions that are of interest to client programs, which may include applications, subsystems, and other programs. For Event Management, HACMP ES is a client.

The Event Management subsystem allows the user to define events, and it notifies the HACMP ES Cluster Manager when these events occur.

For more information about Event Management, see *RS/6000 SP High Availability Infrastructure*, SG24-4838. For more information about defining user events for Cluster Manager, see Chapter 11, “User-Defined Events” on page 127.

3.5 Multiple Clusters in a Partition



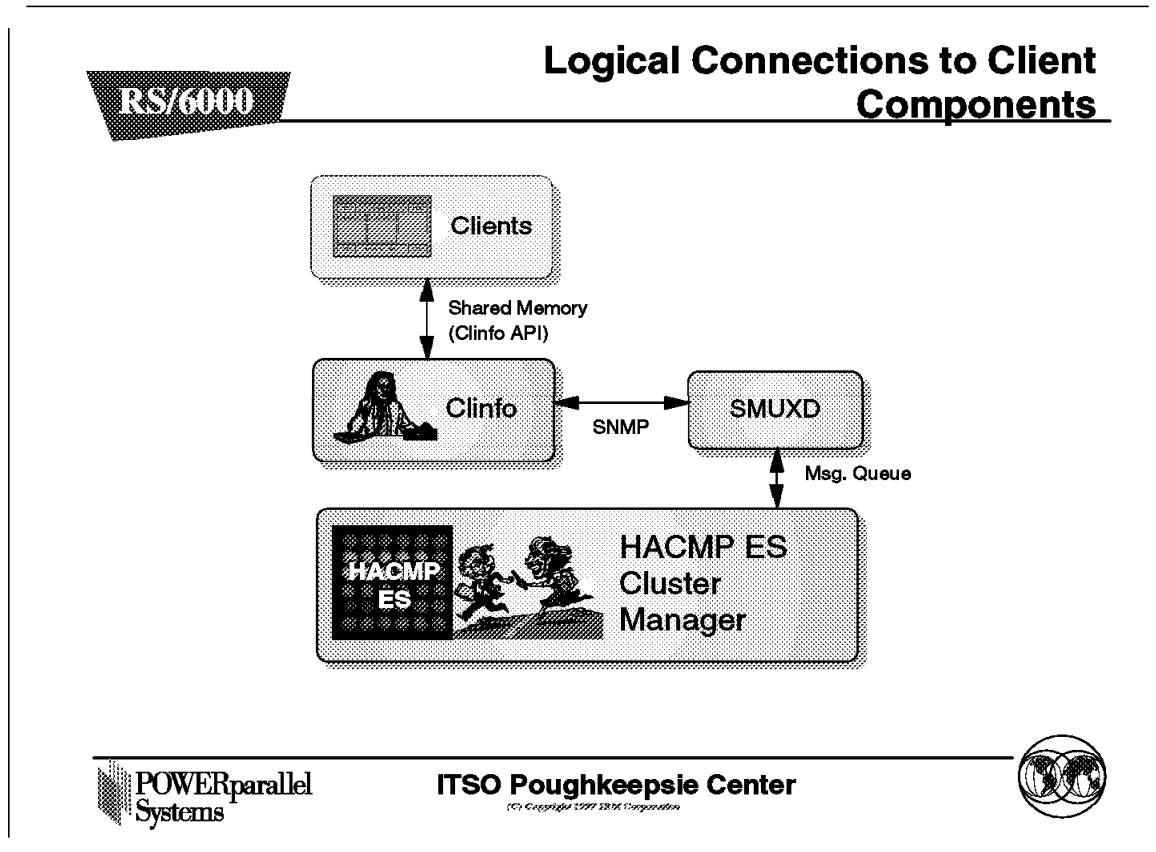
The above figure illustrates how multiple HACMP ES clusters exist in a PSSP partition.

The figure shows a PSSP partition with nine nodes. Group Services and Topology Services are running on these nodes and on the Control Workstation. These components get the partition information from the SDR.

On the other side, there are three HACMP ES clusters on the partition. Each cluster belongs to a different group. Every Group Services/ES within a cluster communicates with every other, but Group Services/ES does not communicate with the nodes outside the cluster.

HACMP ES uses ODM, which has the same structure as HACMP for AIX V4.2.1, to maintain the cluster information. When HACMP ES is started on a node, Group Services/ES is also started on the node. HACMP ES reads the ODM information and builds the HACMP ES provider group name (see 3.11, "Start Sequence" on page 44), which is "clstrmgr_" concatenated with the HACMP ES cluster-ID. As a result, multiple clusters can coexist in an SP partition, since each HACMP ES cluster has a unique provider group name, and a node can request Group Services/ES to join a proper group.

3.6 Logical Connections to Client Components



HACMP ES logical connections to HACMP ES client components are the same as for classic HACMP. Actually, Cluster Information Services (clinfo) is provided with HACMP ES, which uses the same code as classic HACMP.

HACMP ES Cluster Manager is the only subsystem that uses Group Services/ES to subscribe to the HACMP ES provider group. Other applications are not allowed to subscribe to the provider group. Clinfo is provided for these applications; it provides cluster status information with the API.

Clinfo communications are SNMP-based. Clinfo requests status information from the Cluster SNMP agent (SMUXD), which is the SNMP subagent for HACMP ES via the local SNMP daemon. SMUXD maintains information about HACMP ES status, and this status is updated by the Cluster Manager.


3.7 Client Interface Component Definitions

RS/6000

Client Interface Component Definitions


➤ **Cluster SNMP Agent**


- ◆ SNMP Multiplexor Daemon (SMUXD)
- ◆ Receives cluster state information from the HACMP ES Cluster Manager
- ◆ Provides cluster state information to clients using SNMP



➤ **Cluster Information Services**


- ◆ Provides an API that allows the development of "cluster aware" applications
- ◆ Runs on each cluster node and can run in each client outside the cluster





POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



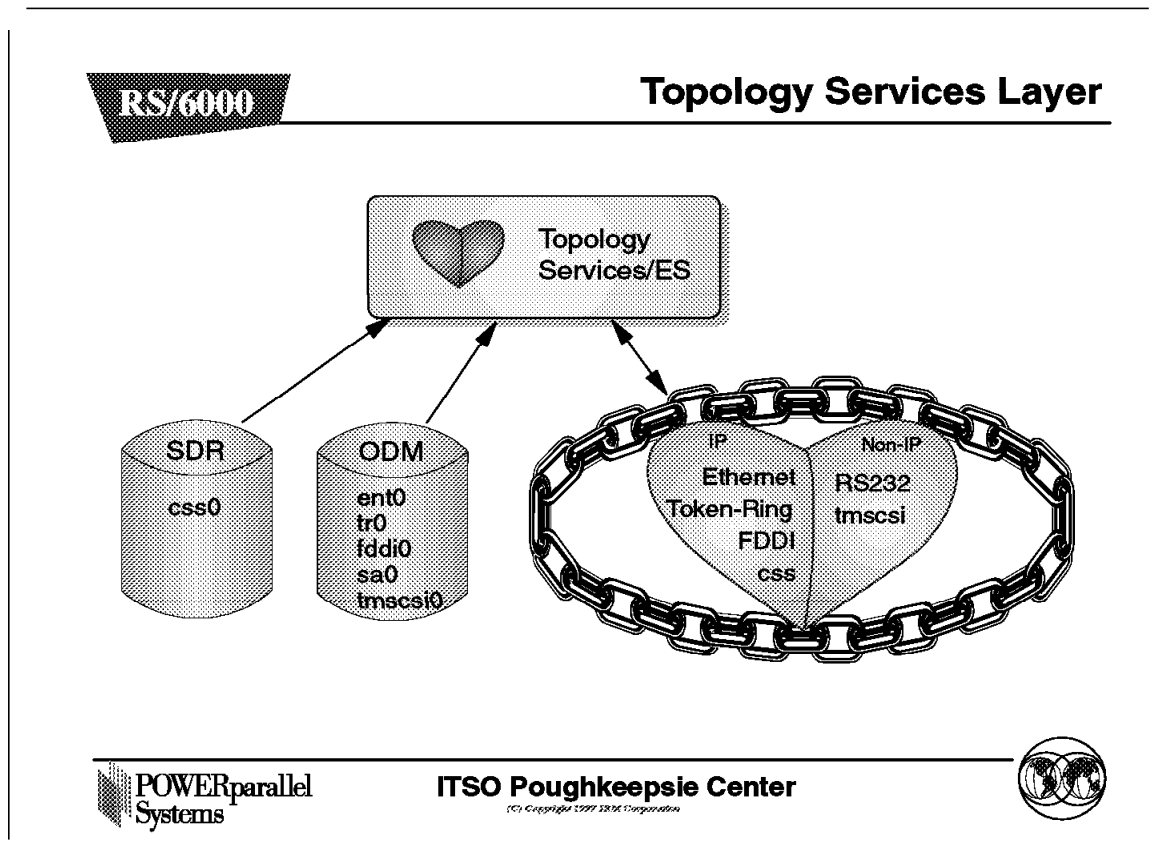
3.7.1 The Cluster SNMP Agent

The cluster SNMP agent, that is, the SNMP multiplexor daemon (SMUXD), receives cluster state information from HACMP ES Cluster Manager and provides it to clients using SNMP, as well as classic HACMP.

3.7.2 Cluster Information Services

Cluster Information Services (clinfo) is a daemon that maintains state information of nodes, networks and interfaces. Clinfo communications are SNMP-based. This daemon runs on each cluster node and can run on each client outside the cluster. It has the same code as HACMP for AIX V4.2.1 and provides an API that allows for the development of "cluster aware" applications, such as clstat.

3.8 Topology Services Layer



To provide heartbeat paths on the various types of networks, as well as supporting IP address takeover, Topology Services/ES was extended from PSSP Topology Services. The following networks are supported by Topology Services/ES:

- IP networks
 - Ethernet
 - Token-Ring
 - FDDI
 - SP Switch (css)
- Non-IP networks
 - RS232
 - Target Mode SCSI (tm SCSI)

The network configuration information for HACMP ES is stored in the ODM, and Topology Services/ES reads the information on the node to create heartbeat paths. For the Switch network, the SDR database is used to acquire the base address of the switch interface.

You can get the status with the following command:

```
# lssrc -ls topsvcs
Subsystem      Group          PID    Status
topsvcs        topsvcs        12436  active
eth1_0: NODES DEFINED/IN GROUP = 4/0, GROUP STATUS = Stable
spether_1: NODES DEFINED/IN GROUP = 4/1, GROUP STATUS = Stable
spether_1: Adapter/Group IDs = (192.168.3.7, 438b53aa)/(192.168.3.7, 438b53b4)
hps_sdr_0: NODES DEFINED/IN GROUP = 4/1, GROUP STATUS = Stable
hps_sdr_0: Adapter/Group IDs = (192.168.13.7, 438b53ab)/(192.168.13.7, 438b53b5)

HB Interval = 1 secs HB Sensitivity = 4 missed beats
CWS = 0.0.0.0
```

Figure 1. Topology Services/ES Status Example



- **Non-IP networks are used by HACMP ES through Topology Services/ES**
 - ◆ RS232 serial line or Target Mode SCSI
 - ◆ Provide heartbeat communications path in the event the TCP/IP subsystem fails
- **Add adapter interface definition on HACMP ES**
 - ◆ Topology Services/ES gets network information from HACMP ODM
 - ◆ Topology Services/ES gets CSS network information from the SDR



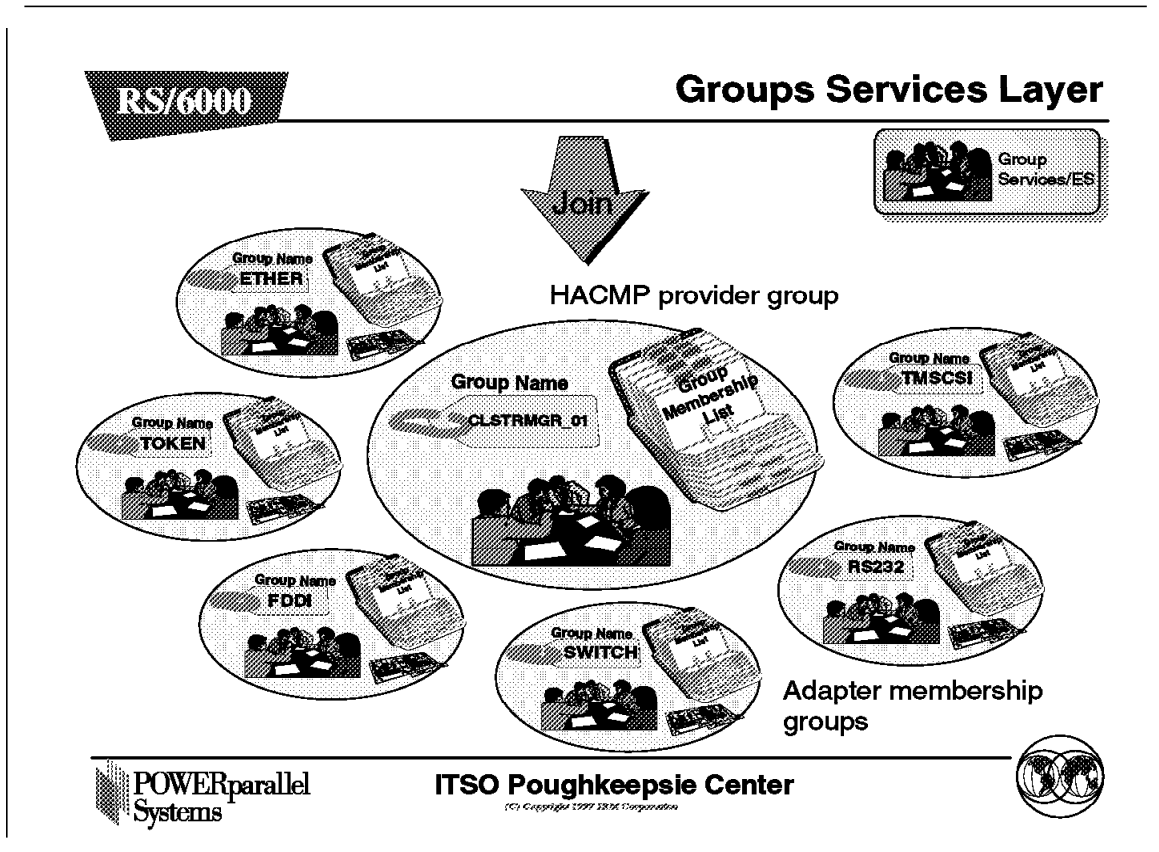
Non-IP Networks Are Used by HACMP ES through Topology Services/ES

A non-IP network is a point-to-point connection between two nodes. It is used by HACMP ES through Topology Services/ES, and provides a heartbeat communications path in the event the TCP/IP subsystem fails. Heartbeat communication is implemented in Topology Services/ES; HACMP ES only subscribes the adapter membership groups. A non-IP network can be an SCSI-2 differential bus using Target Mode SCSI, or an RS232 serial line.

Add adapter interface definition on HACMP ES

To have a heartbeat connection path on a network for Topology Services/ES, the network configuration must be defined on the HACMP ES ODM database, but no definition is needed in the SDR on the Control Workstation except for SP Ethernet and the SP Switch (base address). Topology Services/ES uses the information in the ODM to determine the heartbeat connection path. For the Switch network, Topology Services/ES uses not only the ODM but also the SDR to get base address information for each node.

3.9 Group Services Layer



A Group Services/ES daemon runs on each RS/6000 SP node that is a member of the HACMP ES cluster. HACMP ES consists of multiple processes running on multiple RS/6000 SP nodes and uses any service that the Group Services/ES subsystem provides by forming a group.

HACMP ES uses two kinds of groups. One is an HACMP ES provider group, the other an adapter membership group that is newly provided for HACMP ES.

Note: These groups are created in the Group Services/ES, and they are independent from PSSP Group Services groups.

1. HACMP ES Provider Group

A `clstrmgr` process in the Group Services/ES domain (which is an HACMP ES cluster in which Group Services/ES daemons run) can create a new group, or can ask to become a member of a group to use the functions that Group Services/ES provides. Such a member of a group is called a *provider*, and a group that consists of multiple providers is called a *provider group*. The provider group name is the string "clstrmgr_" concatenated with the HACMP cluster ID, for example, "clstrmgr_1."

A group is a collection of individual processes, which are also called members or providers. A group may have its members on multiple nodes, and each node may have multiple members. For each group, the Group Services/ES subsystem maintains the following group state data:

- A group name

This is a token that uniquely identifies each group in the system.

- A group membership list

This is a list of one or more providers. In a group, each provider is identified by its identifier, which consists of an *instance ID* and the node number on which the provider is running.

Note: Group Services/ES does not use the group state value.

2. The Adapter Membership Group

The Group Services/ES subsystem also keeps track of the status of network adapters, which is defined in the ODM. This status is reflected by the Group Services/ES. By subscribing to these groups, the Cluster Manager, but no other application, can obtain adapter membership information.

You can get the group name with the `lssrc` command, as in the following example:

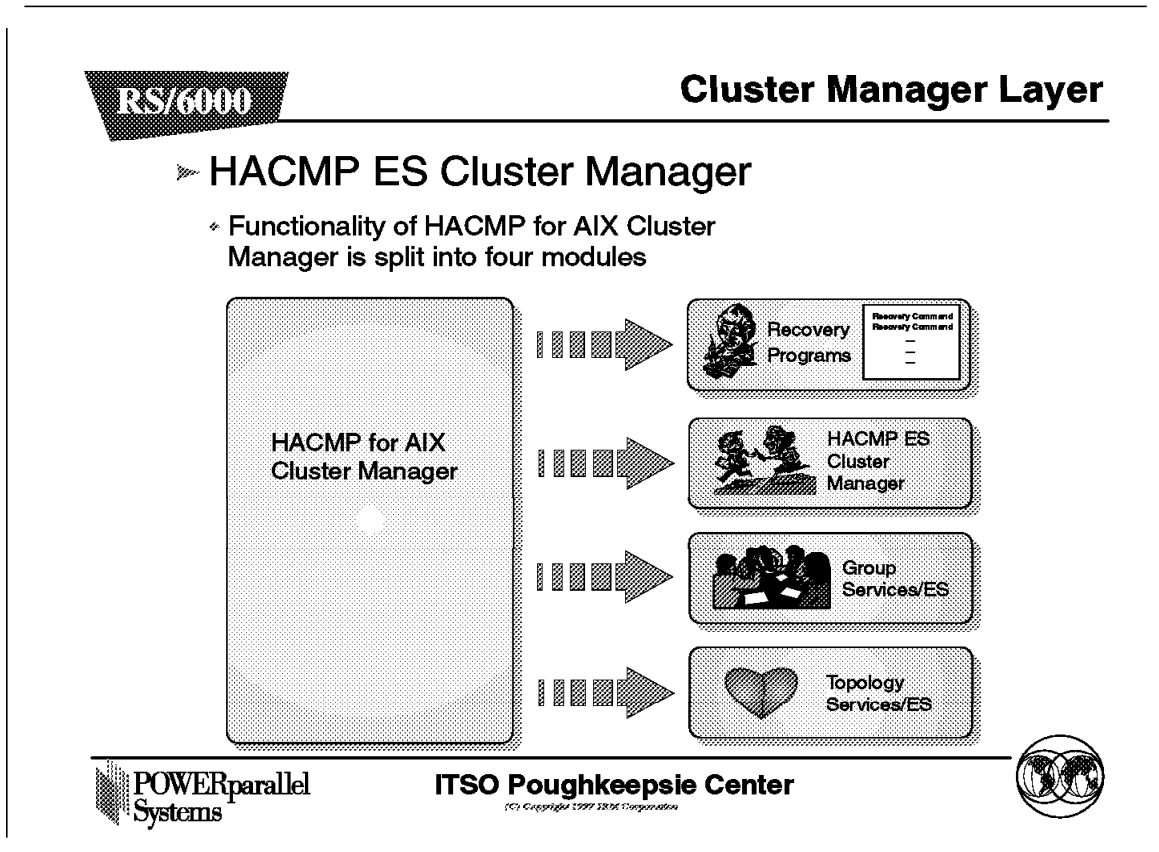
```
# lssrc -ls grpsvcs
Subsystem      Group          PID    Status
grpsvcs        grpsvcs        8188   active
2 locally-connected clients. Their PIDs:
11670 15848
HA Group Services domain information:
Domain established by node 7.
Number of groups known locally: 2

Group name      Number of providers  Number of local providers/subscribers
cssMembership   1                    1                    1
CLSTRMGR_3     1                    1                    0
```

Figure 2. Group Services/ES Status Example

For more information about Group Services, see *RS/6000 SP High Availability Infrastructure*, SG24-4838.

3.10 Cluster Manager Layer



HACMP ES Cluster Manager

The functionality of HACMP for AIX Cluster Manager is split into four modules:

- **Topology Services/ES**

The function of Topology Services/ES consists mainly of exchanging of heartbeats. For more information about Topology Services/ES see, 3.2.2, "Topology Services/ES" on page 31 and 3.8, "Topology Services Layer" on page 38.

- **Group Services/ES**

The function of Group Services/ES is to handle event detection. For more information about Group Services/ES, see 3.2.1, "Group Services/ES" on page 31.

- **HACMP ES Cluster Manager**

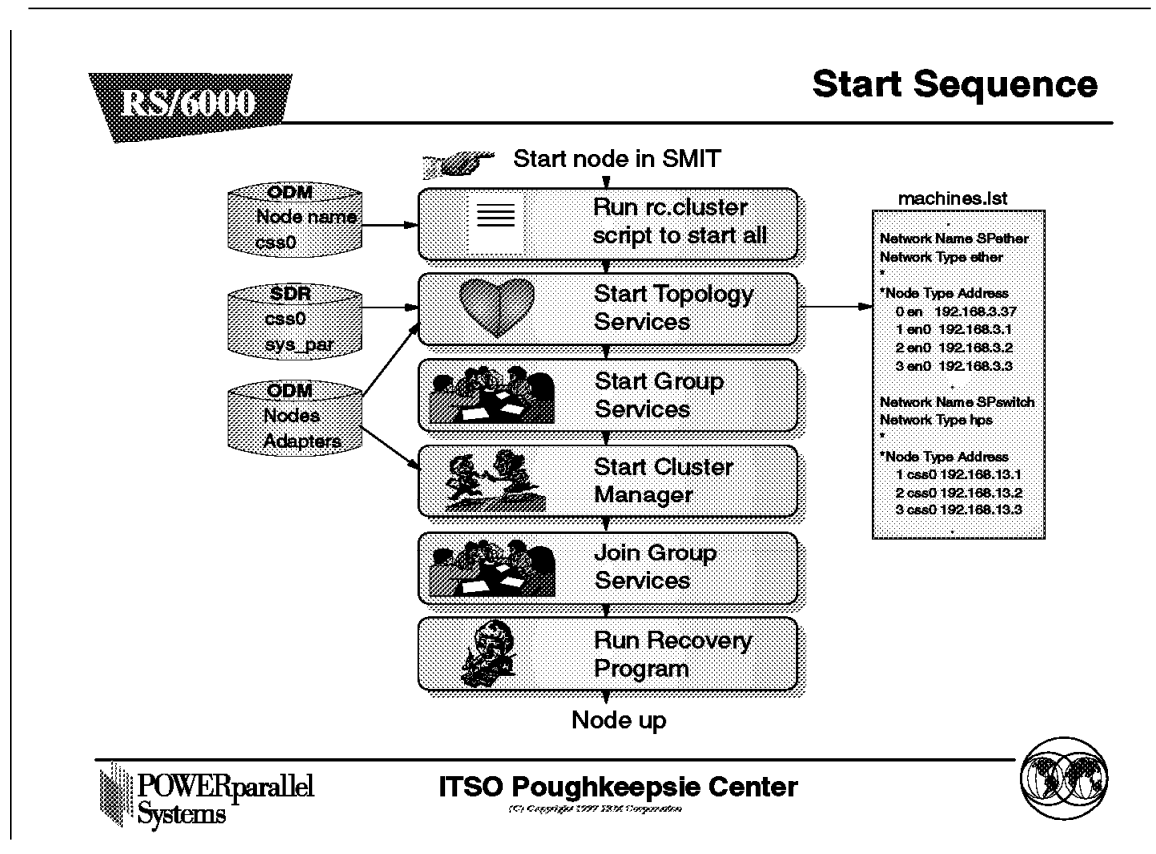
The function of HACMP ES Cluster Manager is to manage all the information coming from Group Services/ES and PSSP Event Management.

- **Recovery Program**

The function of the Recovery Program is to start the HACMP event scripts for a given event in the right order.

These four components together have the same functionality as the HACMP Cluster Manager.

3.11 Start Sequence



When HACMP ES is started, the following sequence takes place:

1. Run the `/usr/sbin/cluster/etc/rc.cluster` script to start all the subsystems. When `rc.cluster` is started, it reads the ODM to get the node name, switch network information, and so on.
2. Topology Services/ES is started by `rc.cluster`. Topology Services/ES first reads the ODM and if it finds switch network information, it then reads the SDR to get additional switch network information. It then creates the `machines.lst` file to create adapter groups (see next figure). At this time, if the Control Workstation is down, Topology Services/ES cannot start since it cannot read the SDR, and the node is not able to join the cluster. Lastly, Topology Services/ES creates all the adapter groups and starts to exchange heartbeats.
3. Group Services/ES is started by the `rc.cluster`. It creates all the adapter groups for itself using Topology Services/ES adapter state information.
4. HACMP ES Cluster Manager is started by the `rc.cluster`. It reads the ODM to get the cluster information and reads the rules file to put the map of events to memory.
5. Cluster Manager creates the HACMP ES provider group if Group Services/ES does not have the group, or joins the existing group and subscribes Event Management, if applicable.

6. Finally, HACMP ES Cluster Manager runs the node_up recovery program to process the node up event, and the node becomes a member of the cluster.

➤ Topology Services builds a list of nodes and adapters when starting

❖ /var/ha/run/topsvcs.<partition name>/machines.lst

Topology Services parameters

Networks in the cluster

Nodes and interfaces
(en1 does *not* refer to interface name)

```
machines.lst
*Timestamp(DDHHMMSS)=29200613
TS_Frequency=1
TS_Sensitivity=4
TS_FixedPriority=38
TS_LogLength=""
Network Name eth1_0
Network Type ether
*
*Node Type Address
5 en0 10.1.1.5
6 en0 10.1.1.6
*
Network Name spether_1
Network Type ether
*
*Node Type Address
5 en1 192.168.3.5
6 en1 192.168.3.6
*
Network Name hps_sdr_0
Network Type hps
*
*Node Type Address
5 cse0 192.168.13.5
6 cse0 192.168.13.6
*
```



ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



Topology Services/ES Builds a List of Nodes and Adapters

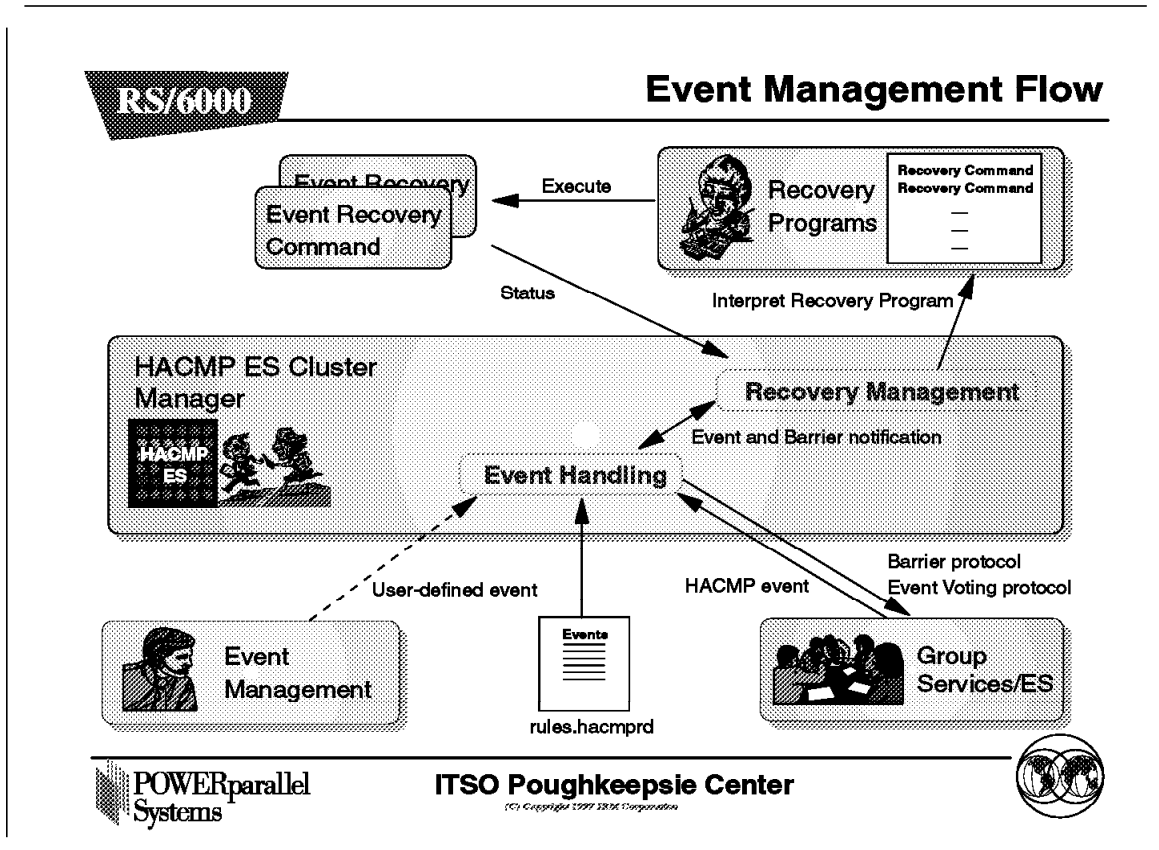
Topology Services/ES always builds a list of nodes and adapters when starting. This list is stored in the machines.lst file, which has an entry consisting of the interface name and IP address. Note that the interface name does not refer to the real interface name, because it is used only as the label for the internal process while creating groups.

Note: If the cluster name cannot be resolved to the hostname, the machines.lst file only contains the first four lines and the Topology Services daemon dies immediately.

Chapter 4. HACMP ES Event Management

This chapter gives you more information about how the events are handled in HACMP ES and how you can add your own events to HACMP ES

4.1 Event Management Flow



This figure shows the relationship between the components. HACMP ES Cluster Manager is the central component that coordinates the takeover and release of cluster resources in response to changes in the cluster topology. It is a daemon that runs on each node configured in the HACMP cluster. It is responsible for monitoring local hardware and software subsystems, tracking the state of the cluster peers, and triggering cluster events when there is a change in the status of the cluster.

For example, when a node joins a running cluster, HACMP ES Cluster Manager sends a request to Group Services/ES with the membership protocol. Group Services/ES then determines the next event with the voting protocol and notifies each Cluster Manager of the cluster member nodes, including the node that is trying to join the cluster.

When an event happens it is mapped by the *rules files* to a recovery program (xxx.rp file). The member nodes run the recovery program for the event. Recovery programs that contain recovery commands are mapped to HACMP ES

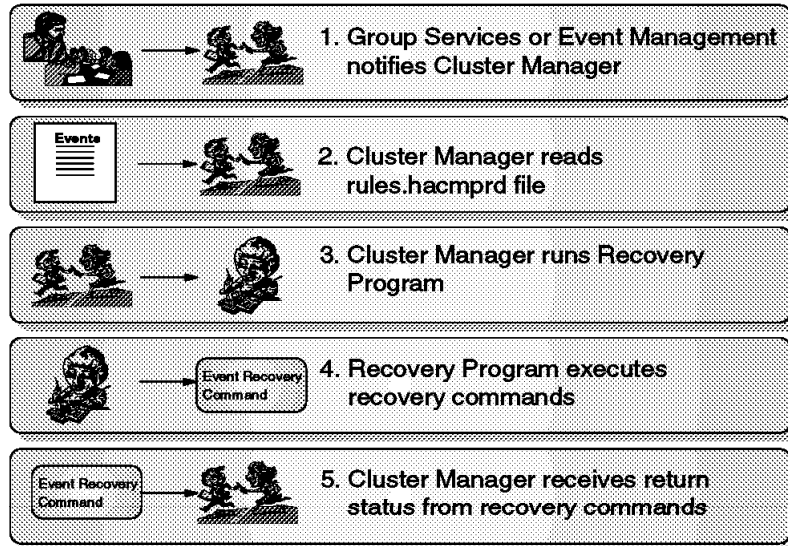
events and execute these HACMP ES scripts. The recovery program (HACMP event script) may release any resources that the joining node is configured to take over. The joining node also runs a recovery program to take over these resources. These processes are synchronized by the *barrier protocol*.

Finally, the joining node is made a member of the cluster.

There are two cases when nodes leave the cluster: one is a planned transition (a node shutdown, or stopping HACMP ES Cluster Manager on a node), the other a failure. In the former case, HACMP ES Cluster Manager controls the release of resources held by the leaving node and the acquisition of these resources by other nodes. The failing node is removed from the membership and its resources are taken over by the nodes configured to do so. User options are provided to override the resource release and the acquisition for tasks such as system maintenance.

Note: Event Recovery commands are also known as HACMP event scripts.

➤ When an event happens:



When an event happens, the following actions are taken:

1. Group Services/ES or Event Management notifies Cluster Manager.

Group Services/ES handles predefined events, while Event Management handles user-defined events.

2. HACMP ES Cluster Manager reads the rules.hacmprd file.

The rules file rules.hacmprd is used by Cluster Manager to determine the recovery program mapped to the event. For more information, see 4.3.1, "Rules File" on page 51.

3. HACMP ES Cluster Manager runs the recovery program.

The recovery program consists of a sequence of recovery command specifications. For more information, see 4.4.1, "Recovery Programs" on page 54.

4. The recovery program executes recovery commands.

A recovery command is an executable program such as a shell script or a binary program. For example, the node_up script is a recovery command.

Note: The recovery commands are the same as the HACMP event scripts in HACMP for AIX.

5. HACMP ES Cluster Manager receives return status from the recovery commands.

If an unexpected status is returned, the cluster hangs. To recover, manual intervention with the `smit cm_rec_aids` or `/usr/sbin/cluster/utilities/clruncmd` command is required.

4.2 Event Mapping

RS/6000

Event Mapping

- The mapping between events and recovery programs is specified in the rules file
 - ✦ Pre-defined HACMP ES events
 - ✦ Manually edited user-defined events

The diagram illustrates the flow of information from a configuration file to a management component. At the bottom, a document icon labeled 'rules.hacmprd' is shown. An arrow points upwards from this icon to a box labeled 'Event Handling' inside a larger box titled 'HACMP ES Cluster Manager'. The 'HACMP ES Cluster Manager' box also contains a small graphic of two figures shaking hands.

POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation

The actions to be taken in response to a particular event are specified in the recovery programs. All nodes must have a rules file with the same name and contents in the same directory. The mappings between events and recovery programs are specified in the rules file. User-defined events can be added by editing this file.

4.3 Event Mapping Files

An event is mapped to a recovery program in the rules file.

4.3.1 Rules File

RS/6000


Event Mapping Files

Events

rules.hacmprd


► Rules File

- ◆ /usr/sbin/cluster/events/rules.hacmprd
- ◆ Nine lines per event:
 1. Event name
 2. State (qualifier)
 3. Resource program path
 4. Recovery type (reserved for future use)
 5. Recovery level (reserved for future use)
 6. Resource variable name (used for Event Manager events)
 7. Instance vector (used for Event Manager events)
 8. Predicate (used for Event Manager events)
 9. Rearm predicate (used for Event Manager events)



POWERparallel
Systems

ITSO Poughkeepsie Center
IBM Corporation



The rules file rules.hacmprd is in the /usr/sbin/cluster/events directory. Each event in the file consists of the following nine objects:

1. Event name

Each event must have a unique name.

2. State (qualifier)

The event name and state are the rule triggers. HACMP ES Cluster Manager will initiate recovery only if it finds a rule with a trigger corresponding to the event name and state.

3. Resource program path

This is a full-path specification of the file containing the recovery program (xxx.rp file).

4. Recovery type (reserved for future use)

This is not used in the current release, but you have to specify some value (as in the following example).

5. Recovery level (reserved for future use)

Recovery level is also not used in the current release, but you have to specify some value.

6. Resource variable name (used for Event Manager events)

7. Instance vector (used for Event Manager events)

In Event Management, this is a set of elements, where each element is a name/value pair of the form name=value, and whose values uniquely identify the copy of the resource (and, by extension, the copy of the resource variable) in the system.

8. Predicate (used for Event Manager events)

In Event Management, this is the relational expression between a resource variable and other elements (such as constants or the previous value of an instance of the variable) that, when true, generates an event. An example of a predicate is $X < 10$, where X represents the resource variable IBM.PSSP.aixos.PagSp.%totalfree (the percentage of total free paging space). When the predicate is true, that is, when the total free paging space is observed to be less than 10%, the Event Management subsystem generates an event to notify the appropriate application (Cluster Manager).

9. Rearm predicate (used for Event Manager events)

In Event Management, this is a predicate used to generate an event that alternates the status of the primary predicate. The rearm predicate is commonly the inverse of the primary predicate (for example, a resource variable is on or off). It can also be used with the event predicate to define an upper and lower boundary for a condition of interest.


Each object needs one line in the file even if you do not specify the value (in this case, you have to enter a blank line). If these lines are removed, HACMP ES Cluster Manager cannot parse the event definition properly, which may cause the system to hang. The line that starts with “#” is treated as a comment line.

Note: The rules file requires exactly nine lines for each event definition (not counting the comment lines) since each field is separated by the NEWLINE. When you add a user-defined event at the end of the file, it is important to remove the unnecessary empty line at the end of the file, or the node will hang.

4.3.2 Event Mapping Example


RS/6000

Event Mapping Example



rules.hacmprd

➤ /usr/sbin/cluster/events/rules.hacmprd file:

```
      :
      :
##### Beginning of Event Definition Node Up #####
#
TE_JOIN_NODE
0
/usr/sbin/cluster/events/node_up.rp
2
0
# 6) Resource variable only used for event manager events
# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events
# 9) Rearm predicate, only used for event manager events
##### End of Event Definition Node Up #####
      :
      :
```


POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



This example shows the definition for the node_up event, which is part of the rules.hacmprd file. When the node_up event occurs, the recovery program /usr/sbin/cluster/events/node_up.rp is executed. According to the rules, the proper value is specified in the state, recovery type, and recovery level lines. There are four empty lines for resource variable name, instance vector, predicate and rearm predicate, since a system-defined event does not use Event Management.

4.4 Recovery Program Structure

This section describes the recovery program structure.

4.4.1 Recovery Programs

RS/6000

Recovery Programs

- The action flow for events is specified in recovery program
- Sequence of recovery command specifications
 - ❖ May have barrier commands in the sequence for synchronization
 - ❖ Barrier command
 - Synchronizes all nodes within the cluster using barrier protocol
 - ❖ Recovery command
 - Quote-delimited string specifying a path to an executable program

```
graph TD; RP[Recovery Programs] --> H[Recovery Management];
```

The diagram illustrates the relationship between Recovery Programs and the HACMP ES Cluster Manager. A box labeled 'Recovery Programs' contains a list of 'Recovery Command' entries. An arrow points from this box to a larger box labeled 'HACMP ES Cluster Manager' which contains 'Recovery Management'.

POWERparallel
Systems

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation

A recovery program is a definition of the actions to be taken given a particular HACMP ES event. It consists of a sequence of recovery command specifications. This sequence consists of two kinds of commands:

- Barrier command

This command is used to synchronize all nodes within the cluster using the barrier protocol. For more details, see 5.1.3, “Barrier” on page 69.

- Recovery command

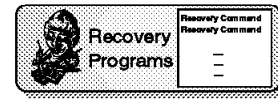
This command is an executable command or a shell script.

Note: The recovery commands are the same as the HACMP event scripts in HACMP for AIX. In HACMP ES you can add your own event scripts.

Each recovery program must have the same contents and be placed in the same directory with the same name on every node.

➤ Format:

```
"node_set recovery_command expected_status"
```



Example: node_up Recovery Program

Node Set	Recovery Command	Expected Status
other	"node_up"	0
	barrier	
event	"node_up"	0
	barrier	
all	"node_up_complete"	X

Node Set

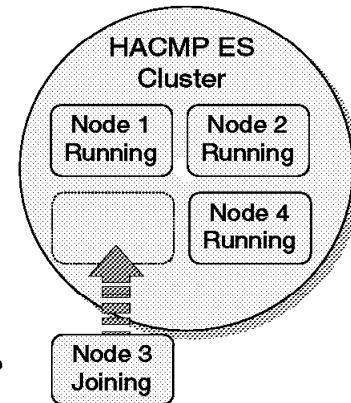
other: 1, 2, 4

event: 3

all: 1, 2, 3, 4

Expected Status

an integer status or
'X' - CLM ignores
the return code



The format of a recovery program is:

```
node_set recovery_command expected_status
```

where

node_set

This is a value that determines on which set of nodes the recovery command will run. The following three sets are supported:

- All - the recovery command is executed on all nodes in the current membership.
- Event - the node on which the event occurred.
- Other - all nodes except the node on which the event occurred.

recovery_command

In this case it is a quote-delimited string specifying a path to an executable program. For a predefined event, only the recovery command name is specified because ODM contains the recovery command information used by the predefined event. For other recovery commands, such as user-defined ones, the information must be specified with the path. For the barrier command, just specify the barrier without quotes.

expected_status

This is the return code of the recovery command. It is an integer status or X. If X is specified, HACMP ES Cluster Manager ignores the return code. If there is a node on which the return code is other than

the expected one (in the shown example it is 0) HACMP ES Cluster Manager on that node detects a failure and does not execute the next barrier command. In this case, the user has to solve the problem with manual intervention in order to recover.

Null

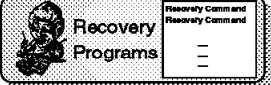
This is reserved for future releases. At the end of each line the recovery command has the word "NULL."

Multiple recovery commands always executed in parallel, except when separated by a barrier command.

4.4.2 Recovery Programs Example

RS/6000


Recovery Programs Example



Recovery Command
Recovery Command
Recovery Command

➤ `/usr/sbin/cluster/events/node_up.rp` file:


```
#
#
# *****
# This file contains the HACMP/PE recovery program for node_up events
# format:
# relationship      command to run      expected status NULL
#
# other "node_up"  0 NULL
# barrier
# event "node_up"  0 NULL
# barrier
# all "node_up_complete" X NULL
#
#
# *****
#
```



POWERparallel
Systems

ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



This figure shows a core part of the `node_up.rp` file, and the next figure describes how the cluster member nodes are synchronized with this definition.

4.4.3 Synchronization of Recovery Programs

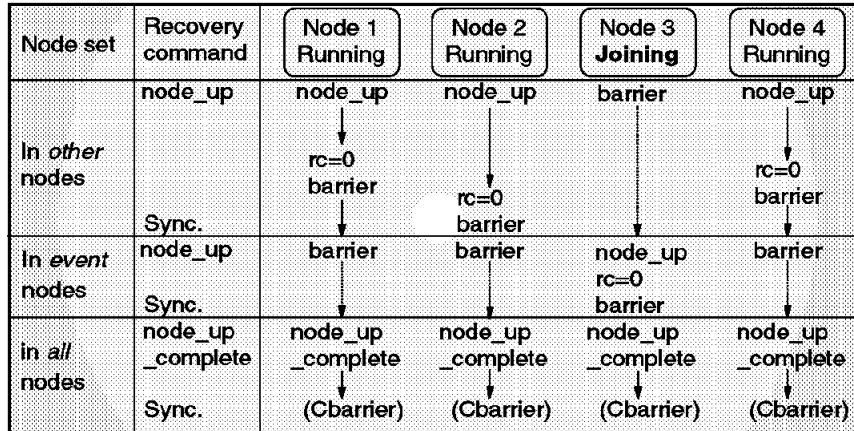
RS/6000

Synchronization of Recovery Programs

➤ Each Cluster Manager controls nodes synchronization with barrier command



Example: Node 3 joining



ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation

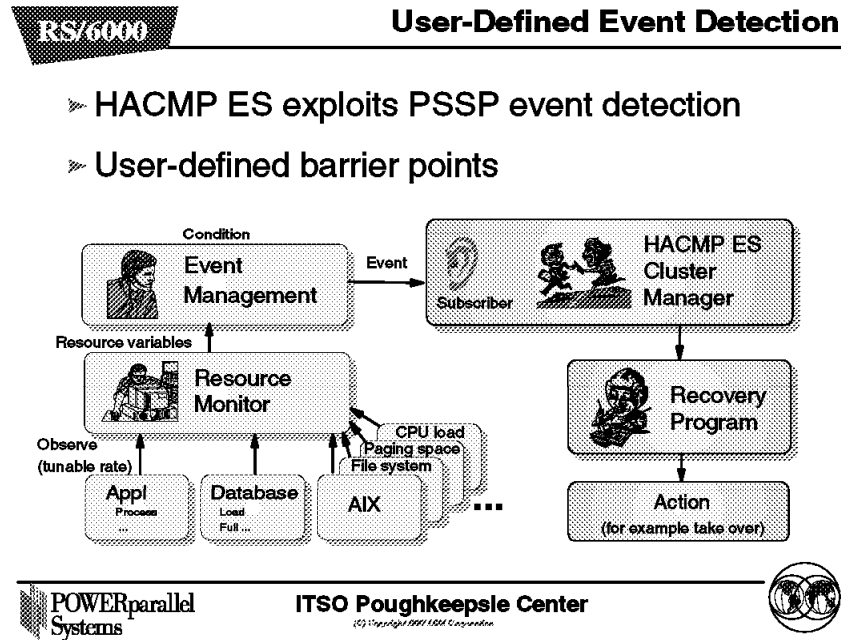


This figure explains how to synchronize each node when the node_up recovery program is executed in the cluster. When Node 3 joins the HACMP ES provider group, which currently consists of three members, Nodes 1, 2, and 4, the recovery program is executed as the following steps:

1. Each HACMP ES Cluster Manager on the nodes, except Node 3, executes the node_up script because node set "other" is specified in the rules file. In this case, "other" means Nodes 1, 2, and 4. At this time, Node 3 encounters the barrier command since the barrier command is followed by first node_up script. HACMP ES Cluster Manager on Node 3 initiates the barrier protocol, and it waits until the other nodes finish the node_up script with return code 0 and reach the first barrier command. If there is a node on which the return code is other than 0, HACMP ES Cluster Manager on that node detects a failure and does not execute the barrier command. Then the cluster hangs because the other nodes wait until all the nodes encounter the barrier command.
2. Once all the nodes encounter the first barrier command, Node 3 executes the node_up script followed by the second barrier command. The other nodes encounter the barrier command, and then each Cluster Manager on the other nodes waits until Node 3 encounters the barrier command.
3. After Node 3 encounters the barrier command, all the nodes execute the node_up_complete script. The return code of the node_up_complete script is ignored by HACMP ES Cluster Manager because X is specified as the expected status in the recovery program. Since the node_up_complete

script is the last command in the node_up recovery program, all nodes are synchronized by HACMP ES Cluster Manager using the Cbarrier protocol internally. The Cbarrier protocol ensures that all nodes have the same status and prevents the execution of another event before the node_up event has completed. For more information about Cbarrier, see 5.1.4, “Cbarrier” on page 69.

4.5 User-Defined Event Detection



HACMP ES exploits IBM Parallel System Support Program (PSSP) event detection to treat user-defined events. It subscribes to the PSSP Event Management subsystem, which provides comprehensive event detection by monitoring various hardware and software resources.

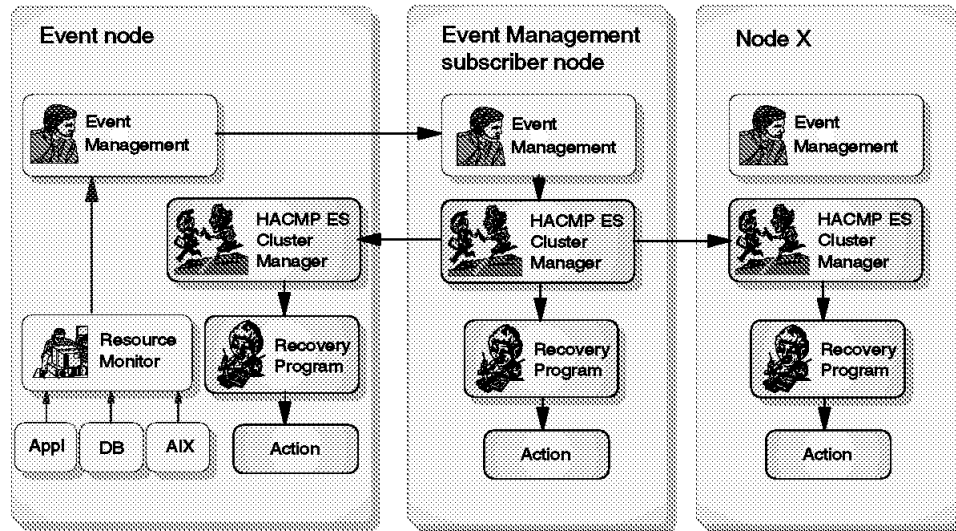
Resource states are represented by resource variables. Resource conditions are represented as expressions called predicates, which have a syntax that is a subset of the expression syntax of the C programming language.

Event Management receives resource variables from Resource Monitor, which observes the state of specific system resources and transforms this state into several resource variables. Resource Monitor periodically passes these variables to the Event Management daemon. This daemon applies predicates that have been specified by HACMP ES Cluster Manager to each resource variable. If the predicate is true, an event is generated and sent to the Cluster Manager. The Cluster Manager processes the event from Event Management by executing a recovery program that is associated with the event.

User-defined Barrier Points

User-defined events can have the synchronization points by specifying barrier commands in the recovery program that is associated with the event.

User-Defined Event Detection (cont'd)



Subscribing to Event Management is done by HACMP ES Cluster Manager on a specific node. This node is called the Event Management subscriber node.

When a user event is notified by Resource Monitor on a node, Event Management on that node notifies Event Management on the subscriber node. Then Event Management on the subscriber node notifies the event to Cluster Manager on the node, Cluster Manager initiates the voting protocol, and the recovery program is executed on a set of nodes specified by "node sets" in the recovery program, according to event priority.

4.6 Event Priority

RS/6000

Event Priority

➤ Event priority consideration for voting

1. Node joins
2. Node fails
3. Swap adapter
4. Network up
5. Network down
6. User-defined events

➤ One protocol at a time



ITSO Poughkeepsie Center
IBM Corporation 1999 IBM Corporation



Event priority is considered among the participating nodes in the following order while the voting protocol is processed:

1. Node joins
Node joins are processed by all of the member nodes plus the joining node.
2. Node fails
Node fails are processed by all of the member nodes minus the failed node, unless this is a graceful down, in which case the failed node runs its event script.
3. Swap adapter
Swap adapters are processed by all of the member nodes.
4. Network up
Processed by all of the member nodes.
5. Network down
Processed by all of the member nodes.
6. User-defined events
Processed by all of the member nodes.

The queue sorting algorithm ensures that events are added to the queue such that:

- Low-numbered nodes are processed before high-numbered nodes.
- Low-numbered networks are processed before high-numbered networks.
- User-defined events are processed after HACMP events.

Notes:

1. Node numbers are equal to the SP node IDs.
2. This differs from HACMP because HACMP uses node names for sorting.

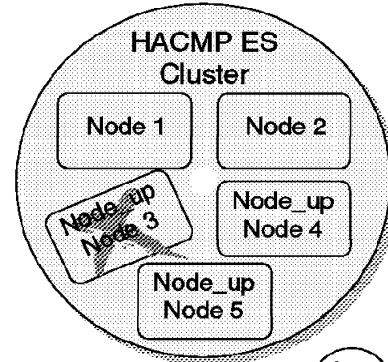
Numbered nodes and networks amount to a canonical sorting order of the names of these resources.

The events are processed serially, one event at a time.

➤ Example

Node 3 failure occurred while starting HACMP ES on Nodes 3, 4 and 5 (Nodes 1 and 2 already up)

1. Process event `node_up` for Node 3 (but failure occurred)
2. Process event `node_up` for Node 4
3. Process event `node_up` for Node 5
4. Process event `node_down` for Node 3, and takeover the resources for Node 3 to other nodes.



This figure describes the flow of the processing events.

Example: a Node 3 failure occurred while starting HACMP ES on Nodes 3, 4 and 5 (Nodes 1 and 2 already up).

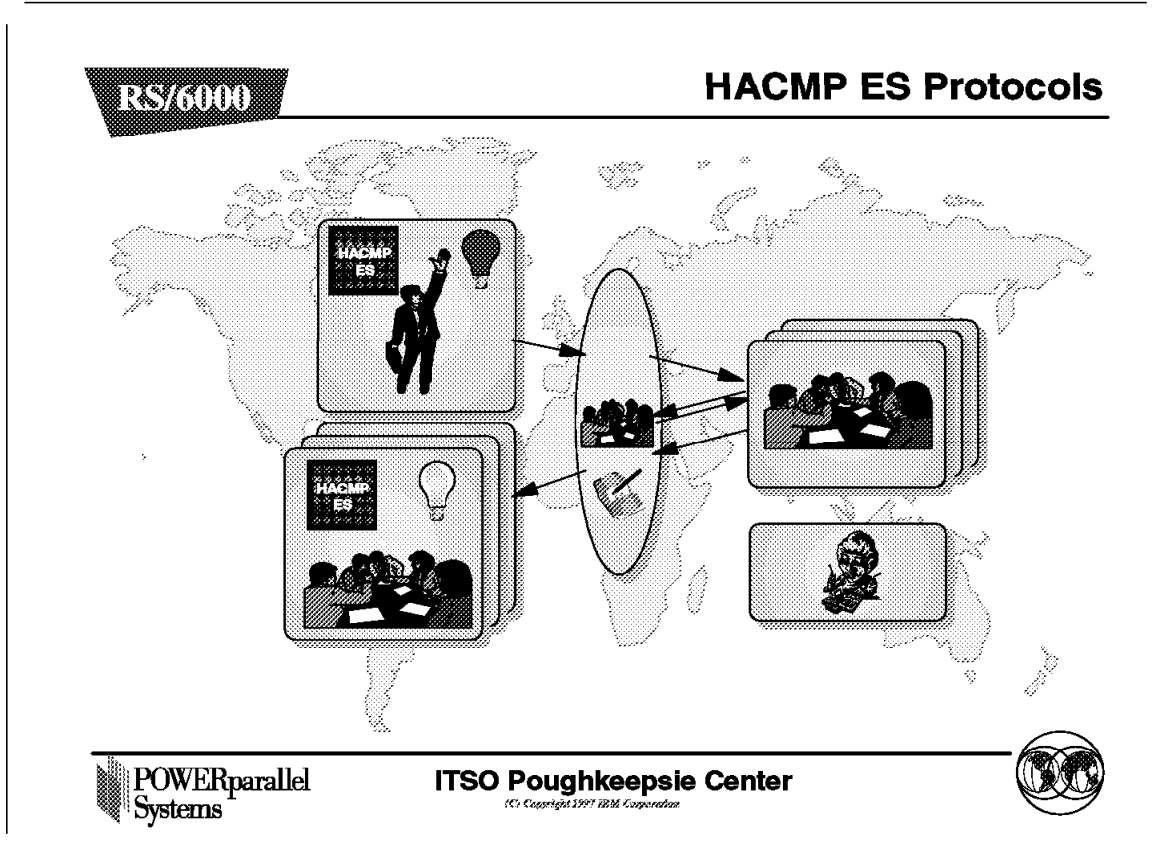
1. Process the event `node_up` for Node 3 (but failure occurred)

The *node joins* event for Node 3 is processed first because node joins have the highest priority and Node 3 is the lowest number node of the joining nodes.

2. Process the event `node_up` for Node 4
3. Process the event `node_up` for Node 5
4. Process the event `node_down` for Node 3, and take over the resources of Node 3 to other nodes.

The *node fails* event for Node 3 is processed last because its priority is lower than that of node joins events.

Chapter 5. HACMP ES Protocols



During normal operation, HACMP ES Cluster Manager exploits Group Services/ES to monitor nodes and networks in the cluster for possible failures. If such failures occur, it acts appropriately to maintain the availability of system resources.

When a node joins or leaves the cluster, membership changes occur. Using Group Services/ES, HACMP ES Cluster Managers on each node coordinate the management of the membership and the release and takeover of resources in response to these events.

RS/6000

HACMP ES Protocols (cont'd)

➤ Membership (join/leave)

- ◆ Protocol initiated when any change is made due to either node up/down or death of Cluster Manager, Group Services/ES, or Topology Services/ES process
- ◆ One-phase protocol initiated by Group Services/ES

➤ Voting

- ◆ Mechanism to decide next event to execute
- ◆ Initiated by a node whose event queue has stabilized (two seconds)
- ◆ Two-phase protocol - to ensure that all of the nodes in the cluster are processing the same event



ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



HACMP ES Cluster Manager uses Group Services/ES to drive protocols, and establishes a provider group of HACMP ES Cluster Manager peers to participate in the protocols. The following protocols are used in the HACMP ES Cluster Manager:

5.1.1 Membership (join/leave)

Membership protocol is not driven using the Group Services/ES Provider Broadcast facility, but is initiated by Group Services/ES when a membership change is made due to either node up/down or a HACMP ES Cluster Manager process death. This is a one-phase protocol that results in an event being put on the event queue.

5.1.2 Voting

Voting is a two-phase protocol initiated by a node whose event queue has stabilized. All nodes verify that the proposed next event is the highest priority event. If any node does not have this event on its queue, it adds it. If any node has a higher priority event, it rejects the protocol and initiates a vote for the highest priority event (priority of events is discussed in 4.6, "Event Priority" on page 63).

➤ Barrier

- ◆ Is used to synchronize recovery steps on all nodes
- ◆ Is initiated by Cluster Manager when a barrier statement is encountered in recovery program
- ◆ All nodes go into the barrier state
- ◆ Two-phase protocol used to implement the barrier command

➤ Cbarrier

- ◆ Synchronize all nodes at the end of an HACMP ES event
- ◆ Internal two-phase protocol implemented in the recovery programs

**5.1.3 Barrier**

Barrier is a two-phase protocol used to implement barrier commands in the recovery programs. Once a node encounters a barrier command in the recovery program, it initiates this protocol, causing all nodes to go into the barrier state. As each node encounters the barrier command in the recovery program, it votes to approve the protocol. When all nodes have done this, Group Services/ES notifies all nodes that the protocol has completed.

5.1.4 Cbarrier

The Cbarrier protocol is intended to synchronize all nodes at the end of an HACMP ES event. It is an internal two-phase protocol implemented in the recovery programs.

➤ Adapter Membership State

- ◆ Distribute adapter group notifications since they are not provided in a consistent manner from adapter membership
- ◆ Internal n-phase protocol



ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



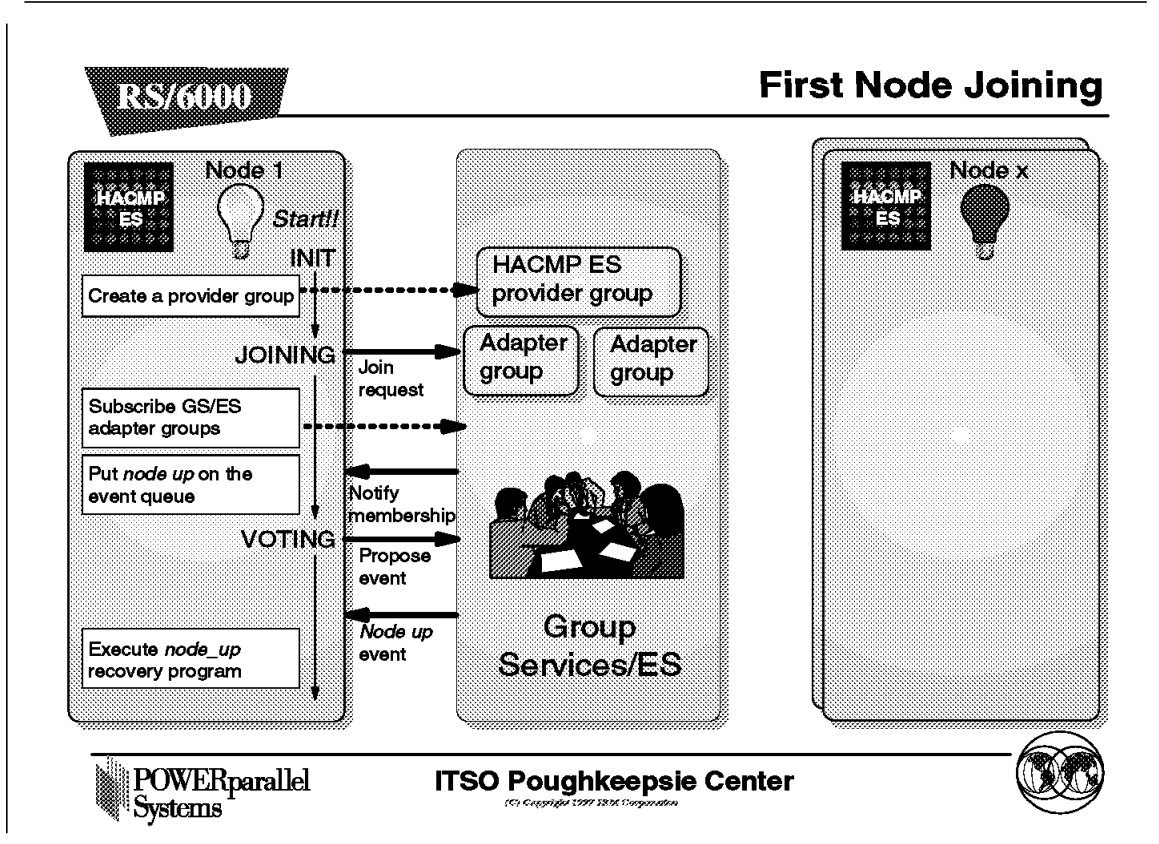
5.1.5 Adapter Membership State

This is an internal n-phase protocol to distribute adapter group notifications since they are not provided in a consistent manner from adapter membership or the Group Services/ES shadow groups in the presence of disjoint networks.

5.2 Node Joining

This chapter describes the steps taken by HACMP ES Cluster Manager when the user starts the cluster. It shows how HACMP ES Cluster Manager interacts with Group Services/ES and how resources are distributed as the cluster grows.

5.2.1 First Node Joining



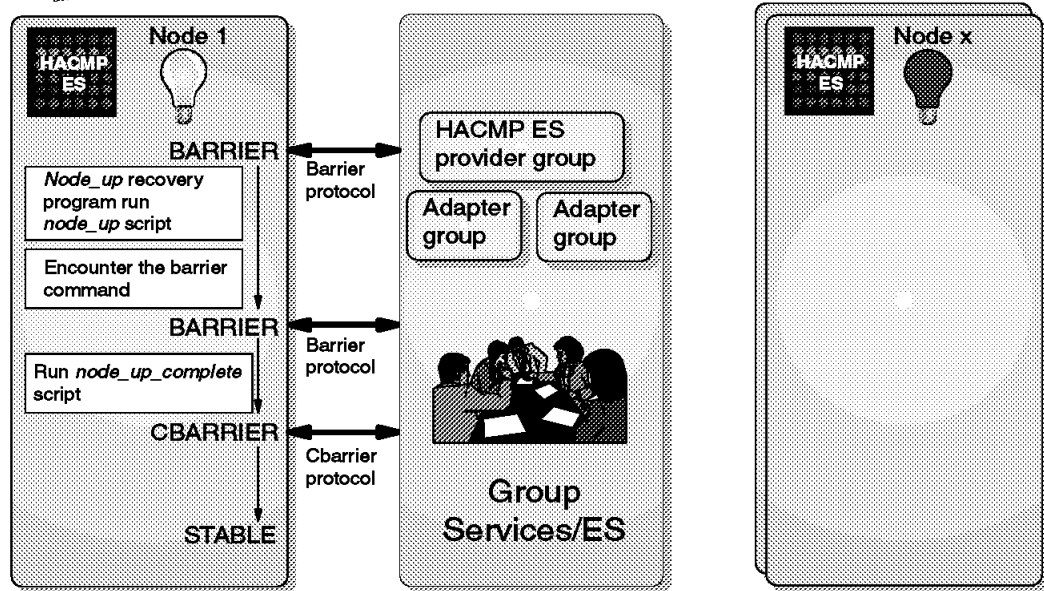
- **Step 1**

HACMP ES is started on Node 1. It registers with Group Services/ES and creates an "clstrmgr_xx" provider group. The provider group name will be the string "clstrmgr_" concatenated with the HACMP ES cluster-ID. If no HACMP ES configuration is found on a node, it joins the clstrmgr provider group.

Node 1 is ready to join the cluster and sends a join request to Group Services/ES. When HACMP ES Cluster Manager starts on a node, it starts in the *init* state. If there is an error in reading the configuration on allocating memory for data structures, HACMP ES Cluster Manager enters the *done* state and terminates. Once the join request is sent to Group Services/ES, HACMP ES Cluster Manager enters the *joining* state. If the ODM configuration indicates that HACMP ES has network adapters to monitor, Cluster Manager subscribes to those Group Services/ES adapter groups.

- **Step 2**

When Group Services/ES notifies HACMP ES Cluster Manager of its membership in the group, HACMP ES Cluster Manager enters the *stable* state and puts its join node on the event queue. Once an event goes on the event queue, HACMP ES Cluster Manager enters the *unstable* state and HACMP ES Cluster Manager waits for a set time interval for the queue to stabilize, then enters the *voting* state. A two-phase event voting protocol is then initiated to reach consensus on the next event to process. The node up event for Node 1 is voted to be the next event to process; Node 1 enters the *rp_running* state and executes its *node_up* recovery program.



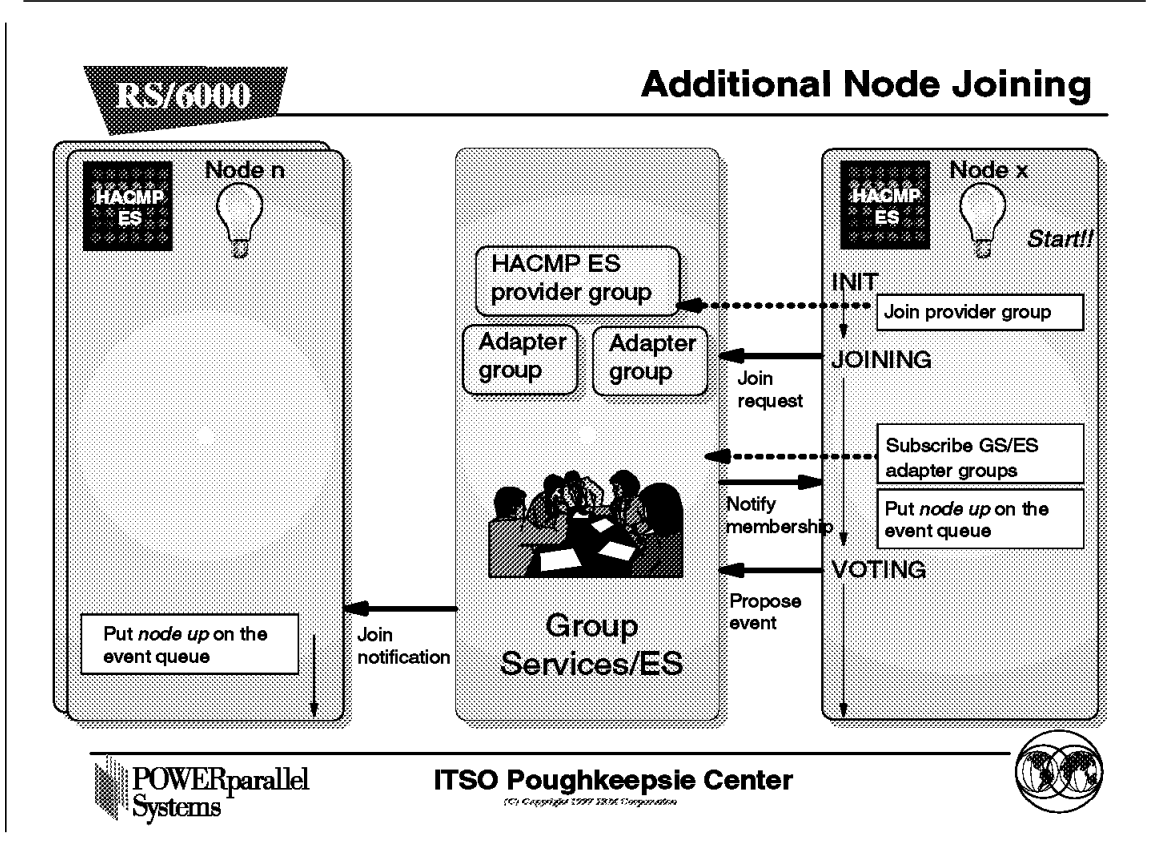
- Step 3

There are two barrier commands in the `node_up` recovery program (see 4.4.3, “Synchronization of Recovery Programs” on page 58). This recovery program encounters the first barrier command before the `node_up` script is executed, since this node is an event node. The barrier command causes HACMP ES Cluster Manager to enter the *barrier* state and a two-phase barrier protocol is initiated. When Group Services/ES indicates the barrier protocol is complete, HACMP ES Cluster Manager enters the *rp_running* state, runs the `node_up` script, and then encounters the second barrier command.

When Group Services/ES indicates the barrier protocol is complete, HACMP ES Cluster Manager enters the *rp_running* state and runs the `node_up_complete` script. When the end-of-file is encountered in the recovery program, HACMP ES Cluster Manager enters the *cbarrier* state and runs the two-phase *cbarrier* protocol.

When Group Services/ES indicates the *cbarrier* protocol is complete, HACMP ES Cluster Manager enters the *stable* state if the event queue is empty, otherwise it enters the *unstable* state. The shell scripts executed from the recovery program check the node resource configuration and take control of the first resource group per network listed in the ODM.

5.2.2 Additional Node Joining

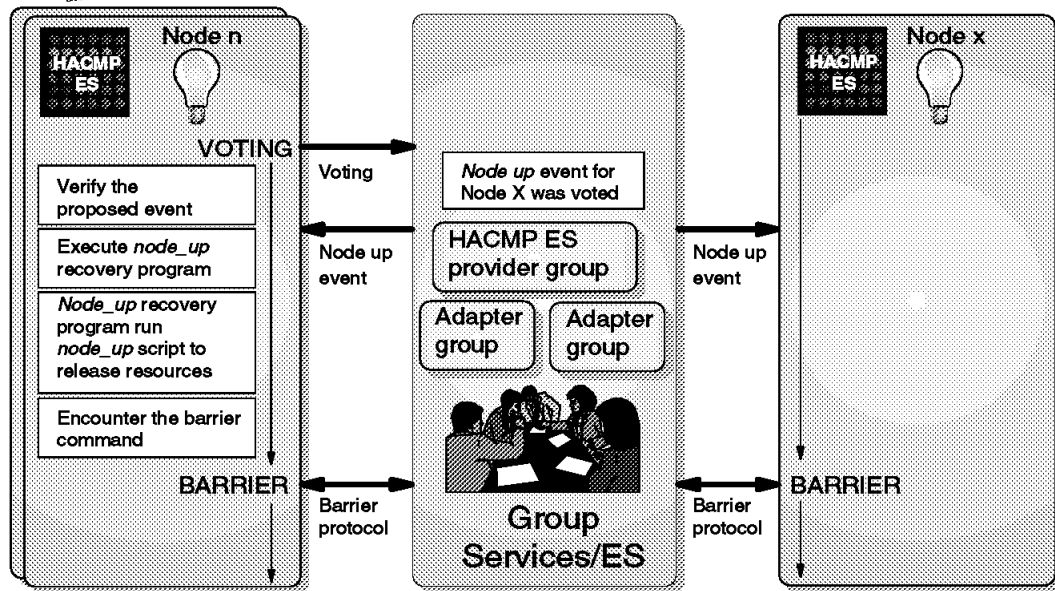


- **Step 1**

HACMP ES is started on Node x. It registers with Group Services/ES and joins the clstrmgr_xxx provider group. If there is an error in reading the configuration or allocating memory for data structures, HACMP ES Cluster Manager enters the *done* state and terminates. Once the join request is sent to Group Services/ES, HACMP ES Cluster Manager enters the *joining* state. If the ODM configuration indicates that HACMP ES has network adapters to monitor, it subscribes to those Group Services/ES adapter groups.

- **Step 2**

When HACMP ES Cluster Manager on Node x joined the HACMP group it went from the *joining* state to the *voting* state, as described for Node 1 in Step 2 (see 5.2.1, "First Node Joining" on page 71). When HACMP ES Cluster Manager on Node n receives Node x's join notification from Group Services/ES, it adds a node up event for node x to its event queue and enters the *unstable* state.



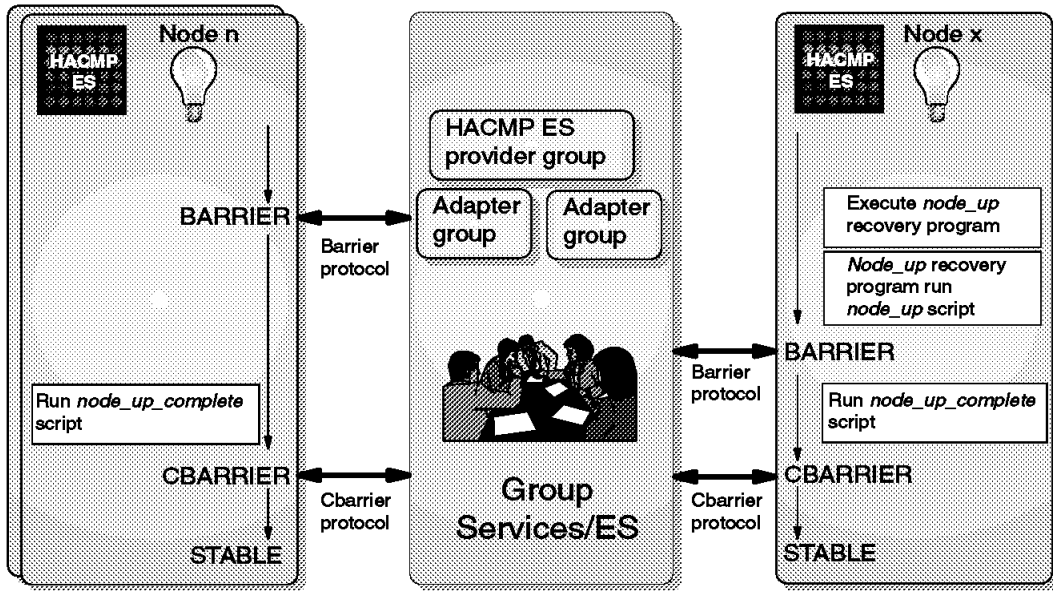
- **Step 3**

After the event queue stabilizes, HACMP ES Cluster Manager on Node n enters the *voting* state and initiates a two-phase voting protocol among the nodes joining the HACMP ES cluster. Note that the first node to enter the *voting* state initiates the event voting protocol. Other nodes in the *stable* or *unstable* state that find themselves in a voting protocol enter the *voting* state. The node up event for Node x is voted the next event to process and HACMP ES Cluster Managers on all nodes enter the *rp_running* state and execute the *node_up* recovery program.

- **Step 4**

The *node_up* recovery program runs the *node_up* script on all nodes in the membership before Node x joins (Node 1 to n in this case) and encounters a first barrier command. The shell scripts run by HACMP ES Cluster Manager may release resources currently held by Node n, if both are in the resource chain for one or more resources and Node x has a higher priority for one of the resources. The barrier command causes HACMP ES Cluster Manager to enter the *barrier* state and initiate a two-phase barrier protocol with all nodes, including node x, which in turn causes the nodes to wait until every node reaches the barrier command in the recovery program. When Group Services/ES indicates that the barrier protocol is complete, HACMP ES Cluster Manager enters the *rp_running* state.

Additional Node Joining (cont'd)



- **Step 5**

HACMP ES Cluster Manager on Node x also executes the node_up recovery program, which runs the node_up script after the first barrier protocol is done. This node_up script causes Node x to claim all of the resources for which it is configured.

After the node_up script is executed, HACMP ES Cluster Manager encounters the second barrier command, resulting in another two-phase barrier protocol and state transitions as previously described, leaving all nodes in the *rp_running* state.

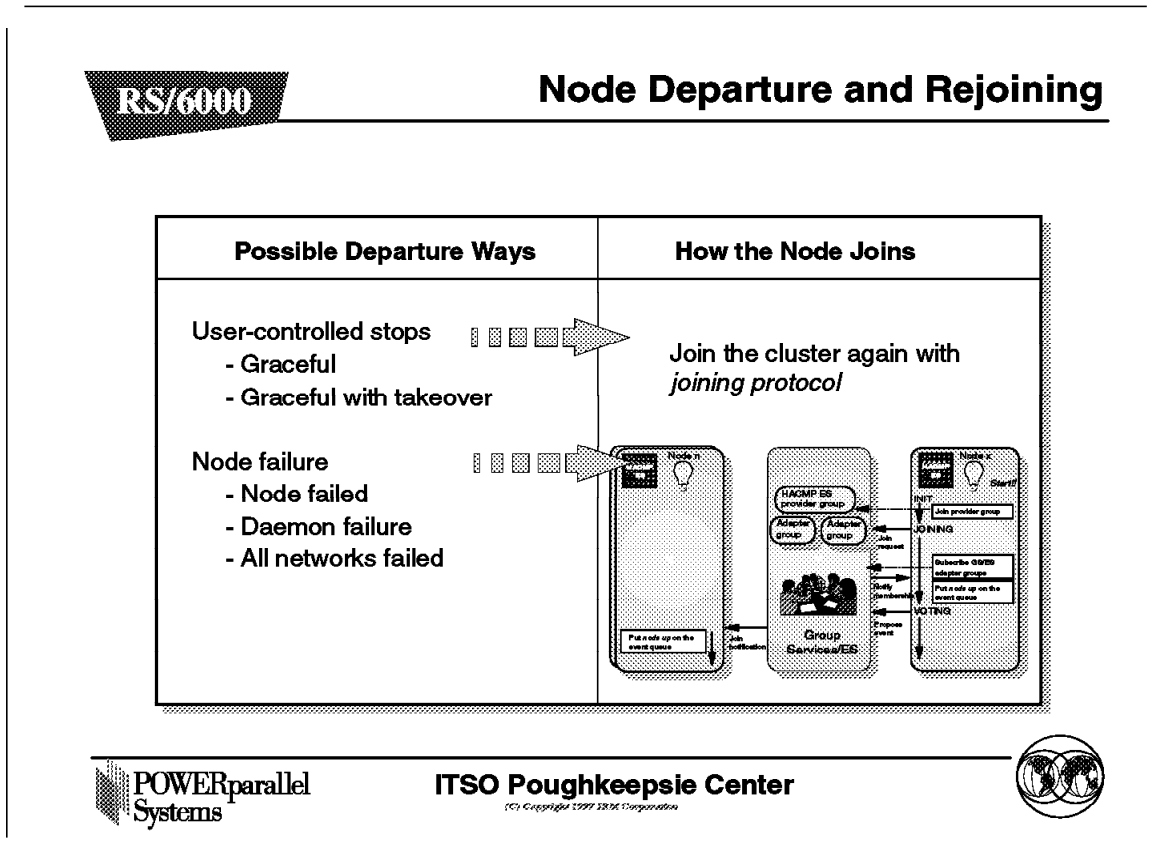
- **Step 6**

The node_up recovery program executes the node_up_complete script on all nodes and end-of-file is reached in the recovery program. HACMP ES Cluster Manager enters the *cbarrier* state and runs a two-phase cbarrier protocol. When Group Services/ES indicates the cbarrier protocol is complete, HACMP ES Cluster Manager enters the *stable* state if the event queue is empty, else it enters the *unstable* state. At this point all resources configured for each node are available to cluster clients.

- **Step 7**

HACMP ES is started on Node x+1. Repeat Steps 1-6.

5.3 Node Departure and Rejoining



HACMP ES Cluster Manager uses Group Services/ES to keep track of the status of nodes within the cluster. If a node fails or HACMP ES Cluster Manager on the node is stopped on purpose, Group Services/ES detects this and takes the necessary actions to get critical applications up and running, and to ensure that data remains available. The possible ways of a node departure are as follows:

- User-controlled stops

You can stop HACMP ES Cluster Manager in the following ways:

- Graceful - HACMP ES Cluster Manager sends a message to the other nodes indicating this is a graceful down. It shuts down after the last `node_down_complete` script has run and the node has released its resources. The surviving nodes do *not* take over these resources.
- Graceful with takeover - HACMP ES Cluster Manager shuts down after the last `node_down_complete` script has run and the node has released its resources. The surviving nodes take over these resources.
- Forced down - HACMP ES does not support the force option, because the other nodes do not have a way to distinguish a forced down from a node failure caused by a HACMP ES daemon's death. Therefore, the standby node takes over the resources from the forced down node.

- Node failure

When a node fails, HACMP ES Cluster Manager on that node does not have time to generate a `node_down` event. In this case, HACMP ES Cluster

Managers on the surviving nodes recognize that the node_down has occurred when Group Services/ES initiates a membership protocol; they then execute the node_down recovery program.

When the failed node joins the cluster again, HACMP ES Cluster Managers running on the nodes recognize a node_up event when Group Services/ES initiates a membership protocol. All nodes, including the rejoining node, execute their node_up recovery programs as described in 5.2.2, “Additional Node Joining” on page 74. The node_up recovery program runs the node_up script on all nodes except the rejoining node, to acknowledge that the returning node is up and to release any resources belonging to it. The returning node then runs its node_up script so that it can resume providing cluster resources. Whether or not resources are actually released in this situation depends on how the resources are configured for takeover.

Chapter 6. Planning

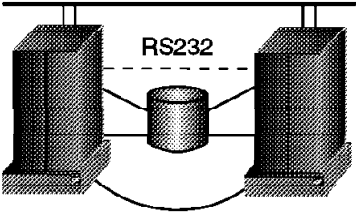
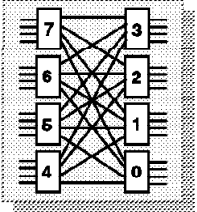
This chapter discusses planning issues, from hardware and software configurations to scenarios and coexistence with other high availability products.


6.1 Planning Hardware Configuration

RS/6000

Planning Hardware Configuration


- Like an HACMP for AIX configuration
- Non-IP serial network
 - Only in case of TCP/IP failure (max. two per node)
 - Daisy-chaining of non-IP networks (no star topology)
- Network adapters
- SP Switch
 - IPAT for switch interface with IP aliasing
 - Dual networks to provide alternative path for switch



POWERparallel
Systems

ITSO Poughkeepsie Center
(C) Copyright 1999 IBM Corporation



Like an HACMP for AIX Configuration

Basically, an HACMP ES hardware configuration is the same as an HACMP for AIX configuration. It needs at least two nodes connected via a LAN. All shared disks must be external disks. A non-IP serial network between nodes is recommended but not mandatory. Two network adapters per network segment are recommended but not mandatory. And so on.

Note: If you do not use the recommended configurations you will reduce the overall availability of your system.

Because of enhancements and limitations of HACMP ES, there are some differences in the configurations. HACMP ES can use more nodes, but it cannot, for example, use an ATM network or multiple RS/6000 SP systems.

Non-IP Serial Network

In an HACMP ES configuration, a non-IP RS232 serial network is recommended to provide an alternate path for heartbeat traffic if other networks fail or if the TCP/IP software protocol stack fails. This serial network is not mandatory.

One limitation is that there can only be two serial networks per node. Therefore, it is not possible to build a star topology with these serial networks, as in HACMP V3.x and earlier versions. To connect multiple nodes to non-IP serial networks, use daisy chaining, as in HACMP V4.x.

Network Adapters

It is strongly recommended to have two network adapters per network segment or subnet that the node is connected to. That is the only officially supported way to build HACMP ES clusters. If there is only one network adapter to a subnet, TCP/IP aliasing can be used to mask IP addresses, but this is not supported (see also Chapter 14, “Cascading by Using One Network Adapter” on page 143), except for the SP Switch. There can only be one SP Switch adapter in a node, therefore IP aliasing is used, but this one is supported and it is different from the one described in Chapter 14, “Cascading by Using One Network Adapter” on page 143.

It is recommended that all interfaces defined in HACMP ES be defined in PSSP also. Conversely, all network interfaces defined in PSSP should be defined in HACMP ES, also. But not all network interfaces defined in HACMP ES have to be defined as takeover resources. For example, the internal administrative Ethernet cannot be defined as a takeover resource.

For a full implementation of HACMP ES, thin nodes do not have enough microchannel slots. This is especially true with a switched RS/6000 SP. There are only three slots available after internal SCSI, internal Ethernet, and SP Switch adapter. It is recommended to have at least two network and two disk adapters in the node. All these adapters do not fit onto a thin node. In cases where there is no need for disk adapters or external network adapters, thin nodes might fit well.

Normally, when running HACMP for AIX or HACMP ES on a switched RS/6000 SP, the switch adapter is made highly available. It protects against node and switch adapter failures.

SP Switch

The SP Switch interface can be a takeover resource. In that case, if a node fails, the IP address of the SP Switch interface is taken over to the other SP Switch adapter interface on the other node. The surviving adapter still has its own IP address, but it also has an IP alias that actually is the IP address of the failed node’s SP Switch interface.

To protect against total SP Switch failure, an alternate path has to be defined, such as FDDI. In case of failure, traffic is moved to that network. This scenario should work nicely if the critical traffic across the SP Switch is between nodes inside the HACMP ES cluster. This is the case, for example, if we make an RS/6000 SP into a DB/2 PE database server, and clients are connecting through external networks. If clients are connecting to the cluster directly through the SP Switch, they should be able to change to using external networks instead of the SP Switch. We recommended that if the SP Switch is evaluated as a single point of failure that must be eliminated, all nodes participating in critical traffic across the SP Switch should be inside the HACMP ES cluster.

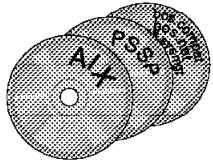
Implementations of the SP Switch are described in detail in the redbook *Implementing High Availability on RS/6000 SP*, SG24-4742).


6.2 Planning Software Configuration

RS/6000


Planning Software Configuration

- **Prerequisites**
 - AIX 4.2.1 and PSSP 2.3 on each node
- **Applications as in HACMP for AIX**
 - ◆ Data layout on disks
 - ◆ Application-specific scripts
 - ◆ Processor-specific licenses
- **Clients as in HACMP for AIX**
 - ◆ Cluster-aware clients
 - ◆ Naive clients





ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



6.2.1 Prerequisites

To install HACMP ES you have to have the following software installed on all the nodes that will be part of the cluster:

- AIX V 4.2.1
- PSSP V 2.3
- bos.compat
- bos.net
- bos.sysmgt
- License for HACMP ES

For more details, see 7.1, "Installation" on page 93 and Chapter 9, "Installation of HACMP ES" on page 121.

6.2.2 Planning Applications

Planning applications in the HACMP ES environment involves the same issues as in the HACMP for AIX environment.

It is not enough to make hardware highly available in order to provide highly available applications. To have a totally highly available application, it needs to be designed for that. Usually the customer does not have the opportunity to influence the design. But there are some things that should be taken care of to make applications as highly available as possible in each situation.

To provide available data for applications requires some design of how to lay out data on disks. When using LVM, mirroring copies should be placed so that there is no single point of failure that can prevent access to the data. Typically this means to have mirror copies for disk adapters, disk busses, power supplies, fan units, and so on. It also means that disks are twintailed to at least two nodes. To protect against frame or SP Switch board failure, nodes should be distributed on different frames.

Typically, performance issues have a strong impact on disk layout, but are not usually in conflict with availability issues.

The main issue is to manage applications from a cluster. Normally it is enough to provide scripts to start and stop the application. If the application itself is not capable to recover from abnormal termination, a start script should be able to do it. Sometimes applications need different parameters to run well in the other node (these parameters should be defined in the planning phase). This is the case when nodes are not equally configured. For example, the primary node for this application is a high node with a lot of memory; the secondary backup node is a wide node with less memory. For example, some database management systems might need parameters that differ according to the memory available.

Some applications use unique license keys for each processor number. In these cases you have to have license keys for each node where the application might be running. Especially in cascading configurations, the application might move to more than just one other node.

6.2.3 Planning Clients

Planning applications in an HACMP ES environment involves the same issues as in an HACMP environment.

To provide a total high availability solution, clients should be included. They can be divided into two categories: naive and intelligent.

An intelligent client is cluster-aware, that is, it is able to read cluster status from Clinfo and react accordingly. It might be able to change the server in case of node failure.

Usually it is too big an investment to make standard clients cluster-aware. Therefore, most customers use naive clients. These can be divided further into two categories: clients using connection-oriented protocol, and clients using connectionless protocol. For example, the TCP protocol is a typical connection-oriented protocol, while UDP is a connectionless protocol. If a node fails, TCP connections are always lost, but UDP connections just hang for a while. In a real client/server solution, clients should just send a package to the server and wait for a response. If no response is received during a timeout

period, they try again. Such a client will simply see a delay in case of node failure.

6.3 Planning Node Relationships

RS/6000




Planning Node Relationships


➤ **Supported node relationships:**

- ✦ Cascading
- ✦ Rotating

➤ **Node relationship not supported in current release:**


- ✦ Concurrent access



**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



Supported Node Relationships

HACMP ES supports two basic node relationships: *cascading* and *rotating*. A cascading relationship means that for some resource group that is defined to take over, there is a priority list for where the resource group should move in case of node failure. If the node that has taken over also fails, there might be another node that takes over that resource group. When the original home node rejoins the cluster, the resource group will move back there.

A rotating relationship means that in case of node failure the resource group that has been taken over stays on the destination node until that node fails. Therefore, its home node changes every time the node it is on fails. For examples of cascading and rotating configurations, see Chapter 8, “Configuration Examples” on page 109.

Node Relationships Not Supported

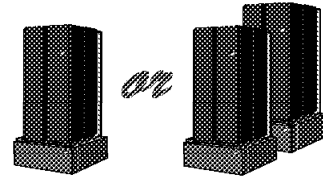
In the current version of HACMP ES, *concurrent access* is not supported. A concurrent access relationship means that each node has direct access to shared disks. Other components in the resource group are handled in either the cascading or rotating manner. Such environments are usually Oracle Parallel Server environments.

RS/6000

Planning Different Scenarios

➤ Single or multiple frames?

- ◆ Multiple frames are a solution for some single points of failure on a single frame:
 - ✎ Supervisor card and power cord
- ◆ For SP Switch board:
 - ✎ All critical nodes have backup nodes behind another SP Switch board
- ◆ Spread the cluster over at least two frames



➤ How many clusters? How big?

- ◆ One 16-node cluster or eight 2-node clusters?
- ◆ Many different applications inside one cluster?



POWERparallel
Systems

ITSO Poughkeepsie Center

© Copyright 1997 IBM Corporation



Single or Multiple Frames?

When planning a large cluster with HACMP ES, the question arises how to divide nodes between the frames. Potential single points of failure on a single node are SP Switch board, supervisor card, and power cord. Therefore, for ultimate high availability, nodes in a relationship chain should include nodes from two frames. If this level of high availability is needed, multiple frames should be considered.

An SP Switch board failure will be seen as a loss of switch network on all nodes connected to the switch board, and as an error event in PSSP. Of course, if we have two or more frames, then we also have to have two or more SP Switch boards to be able to recover. For such a recovery we need customized event scripts, which should be able to move all critical components to the SP Switch board in the other frame. This should also include possible clients.

Recovery from a power cord failure will be seen as a total failure of a single frame. All critical components should have a counterpart in the other frame to be able to recover.

How Many Clusters? How Big?

Another question that arises is how big clusters can be while still being easy to manage. It is easier to manage a small cluster than a big one. On the other hand, one big cluster is easier to manage than many small ones. Some tools, such as C-SPOC, are limited to eight nodes and some are not.

There is no general answer to these questions. They have to be dealt with on a case-by-case basis. If there is one big homogeneous application running on, for example, 16 nodes, it is probably easier to manage as a single cluster than as eight 2-node clusters. PSSP provides tools to manage the nodes as a single node group. If the 16 nodes contain a large variety of different applications with different networks, disks and node relationships, it is probably better to split the cluster into smaller pieces. *Keep it simple!* is still the best advice.

6.5 Planning for Coexistence with Other High Availability Products

RS/6000

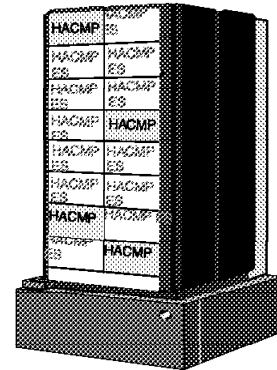
Planning for Coexistence

➤ Coexistence with HACMP for AIX

- ◆ Can coexist in the same partition
- ◆ Some reasons to have both in the same partition:
 - Application needs older version
 - Migration
 - Geographically distributed cluster
- ◆ Not both in the same node

➤ Coexistence with HAGEO

- ◆ No geographically distributed cluster with automatic recovery with HACMP ES



4-node HACMP for AIX Cluster

12-node HACMP ES Cluster



ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation

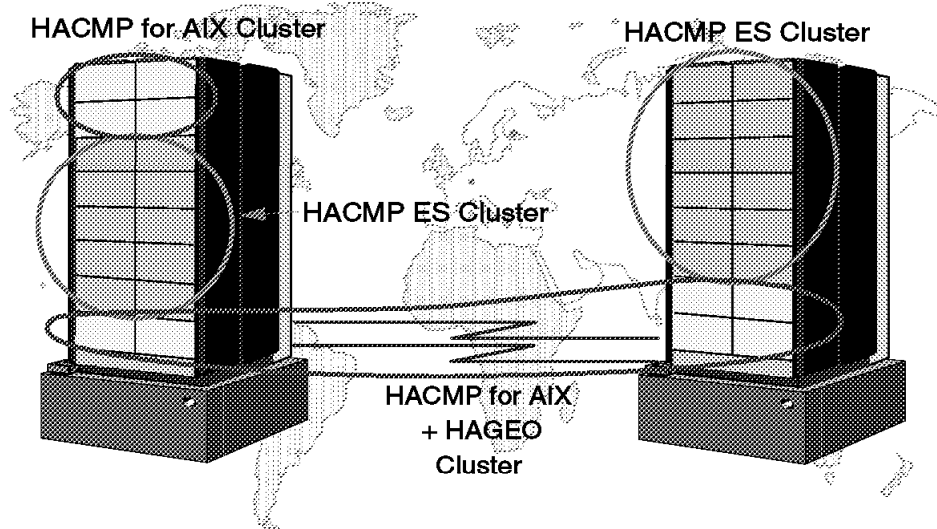


6.5.1 Coexistence with HACMP for AIX and HAGEO

HACMP ES and HACMP for AIX can operate simultaneously in the same RS/6000 SP system, but they cannot be on the same node. If HACMP ES is installed on a node where HACMP for AIX is already installed, HACMP for AIX must be deinstalled first.

HACMP ES and HACMP for AIX nodes cannot be in the same cluster, either. There is no heartbeat between HACMP ES and HACMP for AIX nodes. For example, the joining protocol is different because HACMP for AIX uses its own internal heartbeat and protocols while HACMP ES uses Group Services. Thus, HACMP for AIX clusters cannot be migrated node by node to HACMP ES clusters; migration needs to take the whole HACMP cluster down.


Planning Coexistence with Other High Availability Products (cont'd)



One reason to have both HACMP ES and HACMP for AIX in the same RS/6000 SP system is to use HAGEO. Because HACMP ES can operate only in a single RS/6000 SP system, it cannot provide full disaster recovery. One solution for this is shown in this figure.

Let us take an SAP environment as example. There are many application server nodes in a single HACMP ES cluster on one site. This makes it easier to balance the load in case of node failures. Full disaster recovery is needed. The application servers do not have data on their disks, so there is no need to mirror them to the other site. Database servers have all the data, which should be mirrored to the RS/6000 SP on the other site. Because HACMP ES does not support geographically distributed clusters, we use HACMP for AIX with database server. Normally there are not too many database servers. This database server cluster includes a node from the other site running HACMP for AIX and HAGEO. In that RS/6000 SP system might also be its own, possibly smaller, cluster of application servers. There might also be a cluster running HACMP for AIX for some reason. For instance, the application running may need an earlier version of AIX.

➤ Coexistence with HACWS

- ◆ HACWS only in Control Workstation
- ◆ HACMP ES only in SP nodes
-  No real coexistence

➤ Coexistence with HANFS

- ◆ Not both in the same node

**6.5.2 Coexistence with HACWS**

HACWS is based on HACMP for AIX and always runs on a Control Workstation. HACMP ES does not run on the Control Workstation and so there is no real coexistence between these two products. Both can be used in the same RS/6000 SP environment.

6.5.3 Coexistence with HANFS

HANFS is an HACMP-based product that provides highly available NFS services as a 2-node cluster. But as is the case with HACMP for AIX, it does not work together with HACMP ES on the same node or in the same cluster.

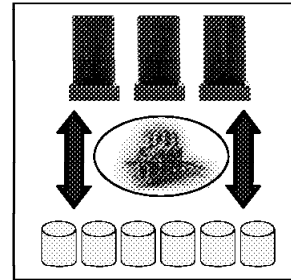
➤ Coexistence with VSD/RVSD

- ◆ VSD/RVSD is a transparent subsystem which is independent from HACMP ES
- ◆ VSD/RVSD takes care of disk resources, HACMP ES of other resources



➤ Coexistence with GPFS

- ◆ Runs on top of VSD/RVSD and has its own recovery mechanisms
- ◆ GPFS and VSD/ RVSD takes care of disk resources, HACMP ES of other resources



6.5.4 Coexistence with RVSD

Recoverable Virtual Shared Disks (RVSD) is used to provide highly available disk services for products like Oracle Parallel Server, which exploits it. VSD/RVSD provides logical volume services, not file systems. An RVSD configuration has multiple nodes with twintailed disks. For heartbeat and event detection it uses PSSP Group Services.

HACMP ES can run in the same node as VSD/RVSD. VSD/RVSD is a transparent subsystem that is independent from HACMP ES. It is like an encapsulated recoverable global device that provides disk services entirely on its own. HACMP ES takes care of other resources, such as networks and applications. Both are using the same type of heartbeat, but HACMP ES also uses other networks that VSD/RVSD does not use. With the new version of VSD/RVSD (Version 2.1), it is recommended to use HACMP ES instead of HACMP for AIX.

HACMP ES can execute RVSD 1.2 recovery scripts just as HACMP for AIX does. HACMP ES and RVSD 2.1 may see different topology information at any given point in time since RVSD gets topology information from PSSP Topology Services based on two network interfaces. HACMP ES gets topology information from Topology Services/ES, which may be based on many more network interfaces than PSSP Topology Services. RVSD 2.1 provides an option to bring down the Cluster Manager when it does a takeover, thus causing the HACMP and RVSD failovers to be synchronized. For an example, see 8.1.2, "Rotating Takeover Scenario" on page 111.

6.5.5 Coexistence with GPFS

General Parallel File Server (GPFS) is used to provide file services. It provides the same high availability features as RVSD, but on the file system level.

HACMP ES can run on the same node as VSD/RVSD and GPFS. They have no communication to each other and they are totally independent.

Chapter 7. HACMP ES Installation and Customization

This chapter gives you an overview of installing, customization and migrating of HACMP ES

7.1 Installation

This chapter discusses the HACMP ES installation steps and related topics that demand special attention.

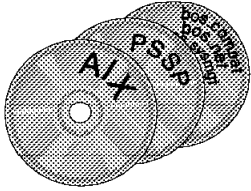
7.1.1 Installation Prerequisites


RS/6000

Installation

➤ **Prerequisite**


- ✦ AIX 4.2.1 on each node
- ✦ PSSP 2.3 with ssp.ha modules on each node
- ✦ bos.compat
- ✦ bos.net
- ✦ bos.sysmgt
- ✦ Root authority
- ✦ 15MB free space on /usr
- ✦ 1MB free space on /
- ✦ License for each cluster node
- ✦ NetView installed if HAView is used





**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



Prerequisites

The following components are mandatory for HACMP ES to work:

- AIX 4.2.1 on each node
- PSSP 2.3 with ssp.ha modules on each node
- bos.compat
- bos.net
- bos.sysmgt

Table 1. Prerequisite Software

Component name	File name
4.2.1.0 LAN COMIO Compatibility Software	bos.compat.lan 4.2.1.0
4.2.1.0 AIX 3.2 to 4 Compatibility Links	bos.compat.links 4.2.1.0
4.2.1.0 Network File System Client	bos.net.nfs.client 4.2.1.0
4.2.1.0 Network File System Server	bos.net.tcp.server 4.2.1.0
4.2.1.0 TCP/IP Client Support	bos.net.tcp.client 4.2.1.0
4.2.1.0 TCP/IP Server	bos.net.tcp.server 4.2.1.0
4.2.1.0 Software Trace Service Aids	bos.sysmgt.trace 4.2.1.0


The following resources are required for installing HACMP ES:

- Root authority - is needed to install HACMP ES
- 15MB of free space on /usr
- 1MB of free space on /
- License for each cluster node
- NetView installed if HAView is used

HAView requires NetView.

For more information about installation, see Chapter 9, "Installation of HACMP ES" on page 121.

➤ Overview

- ◆ If HACMP is installed and you want to keep definitions, save them
  See the Migration chapter
- ◆ If HACMP is installed, remove it
 Run `smit install_remove` on each cluster before installing HACMP ES
- ◆ HACMP ES is installed in exactly the same manner as the HACMP for AIX LPP



This figure shows an overview of the installation steps.

- If HACMP is installed and you want to keep the definitions, save them.
 For more information about migration, see Chapter 10, “Migration” on page 123.
- If HACMP is installed, remove it.
 Run `smit install_remove` on each cluster node before installing HACMP ES.
- HACMP ES is installed in exactly the same manner as HACMP for AIX LPP.

For more information about installation, see Chapter 9, “Installation of HACMP ES” on page 121.

➤ HACMP ES Modules

- ◆ Base
- ◆ VSM
- ◆ C-SPOC
- ◆ HAView
- ◆ man
- ◆ messages



HACMP ES consists of the following modules:

- Base

The following base modules are mandatory for HACMP ES:

- cluster.es
- cluster.adt.es

cluster.es includes HACMP ES Base Client, Saver, and so on. cluster.adt.es includes some samples, such as clinfo and clstat.

- VSM

- cluster.vsm

This module includes the HACMP Visual System Management Configuration utility.

- C-SPOC

- cluster.cspoc

This module includes cspoc runtime and related commands.

- HAView

- cluster.haview

This module includes HACMP HAView.

- man

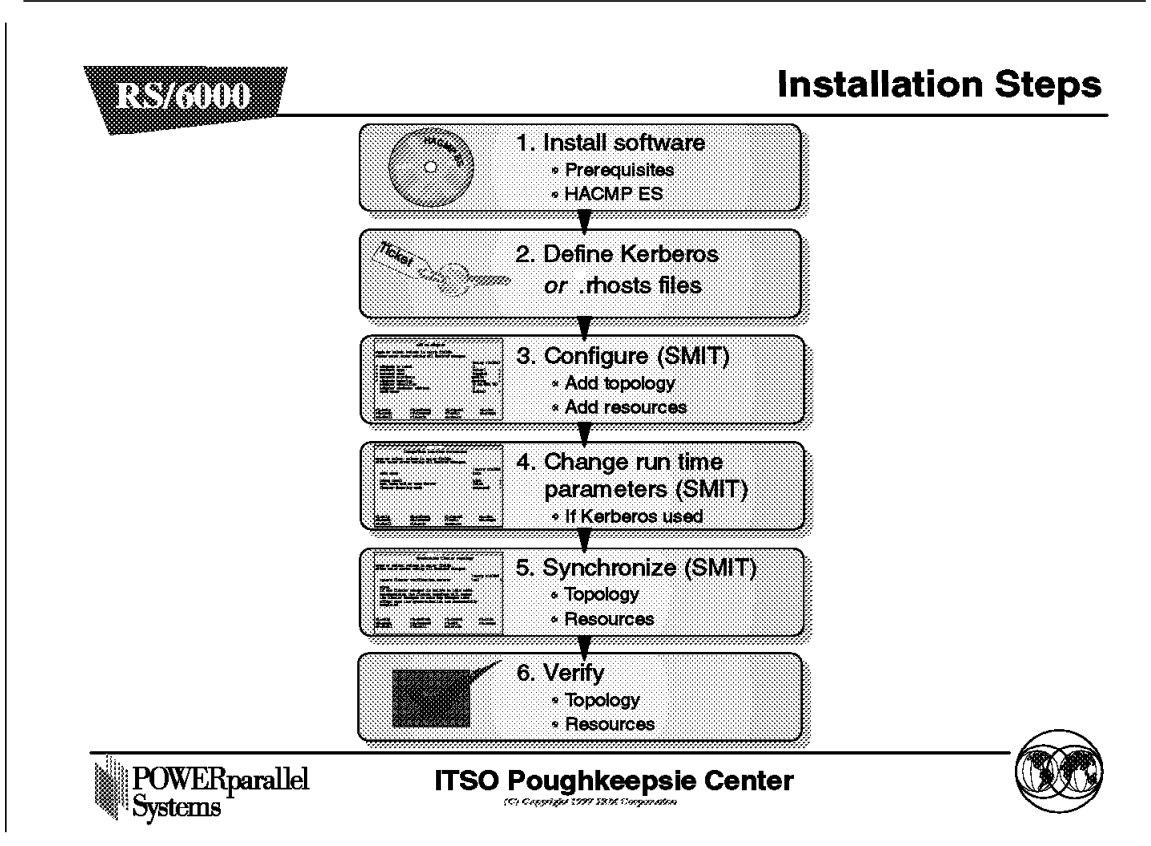
- cluster.man.en_US.data
- cluster.man.en_US.haview.data

These modules include man pages for HACMP ES.

- messages
 - cluster.msg.en_US
 - cluster.msg.en_US.es
 - cluster.msg.en_US.haview

These modules include HACMP ES messages.

7.1.2 Installation Steps



This figure describes the HACMP ES installation steps.

1. Install software

If the previous HACMP is installed on the system, back up the user scripts and take a snapshot of the cluster. Then remove HACMP and install the components required by HACMP ES.

2. Define Kerberos or the .rhosts file

If using the .rhosts file, add all boot addresses and service addresses to the .rhosts file. You can also add standby addresses to this file, but this is not mandatory.

If using Kerberos, all boot addresses and service addresses have to be kerberized. You can also kerberize the standby addresses, but this is not mandatory. For details about using Kerberos, see Chapter 12, "Using Kerberos" on page 131.

3. Configure (SMIT)

Configure the topology and resources as follows:

Topology:

- Cluster
- Nodes
- Adapters

Resource:

- Resource groups
- Application servers

Make sure that all nodes defined in HACMP ES are in the same PSSP partition, because HACMP ES does not check this.

4. Change run time parameters (SMIT)

If you want to use Kerberos, change a parameter with the following SMIT screen. The fast path for this screen is `cm_run_time.select`.

A screen like the one shown in Figure 3 is displayed:

```

Change/Show Run Time Parameters

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

Node Name                [Entry Fields]      sp2n05
Debug Level              high                +
Host uses NIS or Name Server  false              +
Cluster Security Mode     Enhanced           +

```

Figure 3. Change/Show Run Time Parameters Screen

Set the Cluster Security Mode field to Enhanced to use Kerberos-authenticated security.

Note: The `/.rhosts` file must be removed from *all* cluster nodes when the security mode is set to Enhanced. Failure to do so allows for the opportunity to compromise the authentication server. Once compromised, all authentication passwords must be changed.

5. Synchronize (SMIT)

After configuring the cluster, the configuration information should be synchronized on all the nodes.

The fast path is:

- For the topology - `configchk.dialog`
- For the resource - `clsyncnode.dialog`

If Cluster Manager is active on this node, synchronizing the Cluster Topology will cause Cluster Manager to make any changes take effect once the synchronization has successfully completed.


6. Verify

Before starting the cluster, make sure that the configuration is appropriate by verifying the topology and resources. The fast path for this is `clverify.dialog`.

7.2 HACMP ES Customization

This chapter offers an outline of HACMP ES customization.


7.2.1 Customization

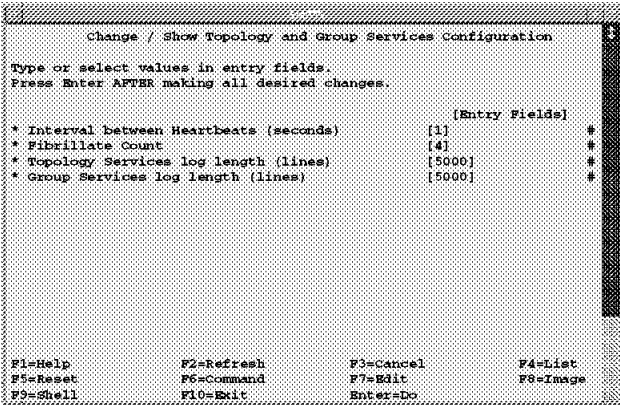

RS/6000


Customization

➤ **Heartbeat rate tunable in HACMP ES SMIT panels**


Fast path: `smitty change_show_ts_gs`


Topology Services




**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



The heartbeat for Topology Services/ES is tunable in the HACMP ES SMIT panel. The fast path for this panel is `change_show_ts_gs`.

Your screen will look like the example in Figure 4:

```
Change / Show Topology and Group Services Configuration

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

* Interval between Heartbeats (seconds)      [Entry Fields]
* Fibrillate Count                           [1]          #
* Topology Services log length (lines)       [4]          #
* Group Services log length (lines)         [5000]       #
* Group Services log length (lines)         [5000]       #

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

Figure 4. Change Topology and Group Services Configuration Screen

- Interval between heartbeats

This field specifies the time interval, in seconds, between heartbeat messages. The heartbeat interval and the fibrillate count determine how soon a failure can be detected. The time needed to detect a failure can be calculated using the following formula:

$(\text{heartbeat interval}) * (\text{fibrillate count}) * 2 \text{ seconds}$

The default value for this field is 1.

- Fibrillate count

The default value is 4.

- Topology Services log length (lines)

This field indicates the maximum length of the Topology Services/ES log file, that is, the maximum number of entries, or lines, that can be recorded to the log file. When the log file reaches this limit, it is copied to another file; then it is cleared, and subsequent entries are recorded at the start of the file.

The default value is 5000.

- Group Services log length (lines)

This field indicates the maximum length of the Group Services/ES log file. The default value is 5000.

Notes:

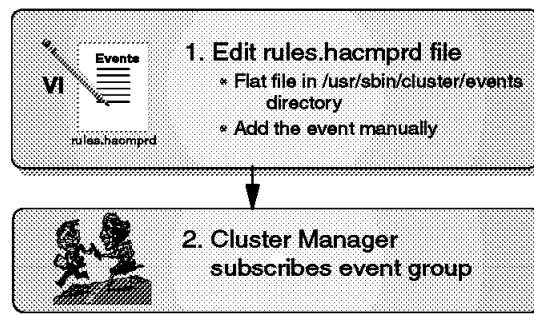
1. These values take effect for all networks.
2. Changing the failure detection time is not method to speed up takeover times.

7.2.2 Steps to Define Customized Events

RS/6000 Steps to Define Customized Events

➤ User can add own events

- ◆ To react to non-standard HACMP ES events
- ◆ For example:
 - /tmp filesystem is over 90% full
- ◆ When event occurs, Cluster Manager runs the recovery program defined in rules.hacmprd file
- ◆ Steps for pre-defined PSSP events:



ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



User Can Add Own Events

Users can add their own events to react to nonstandard HACMP ES events. For more details, see 4.5, "User-Defined Event Detection" on page 60 and Chapter 11, "User-Defined Events" on page 127.

For example, to define the event that the /tmp file system is over 90% full, HACMP ES executes the recovery program that is defined in the rules.hacmprd file when the event occurs.

Many events are predefined in PSSP. You can exploit these as user-defined events. To subscribe a predefined PSSP event by Cluster Manager, the following steps are necessary:

1. Stop the cluster
2. Edit the rules.hacmprd file

This is a file in the /usr/sbin/cluster/events directory. You have to add the predefined PSSP event manually. We recommend that you back up the file before modifying it. Then create a recovery program that contains a sequence of recovery commands, such as a shell script or executable commands, and place the same directory on all nodes. All nodes must have the same rules.file in the same place, and recovery programs in the same directory. If you need synchronizing points while the recovery action is taking place, you can use the barrier command in your recovery program.

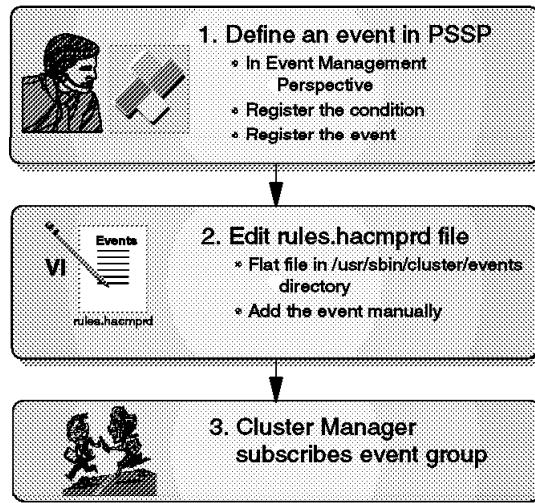
3. Restart the cluster

The rules.hacmprd file is stored in memory when Cluster Manager is started. To reflect the changes, you have to restart all clusters. There should not be any inconsistent rules in a cluster.

4. Cluster Manager subscribes event groups

➤ **New events that are not pre defined to PSSP**

- ◆ For example:
Return code of database roll-back



New Events Not Predefined in PSSP

To define an event in the SDR, Cluster Manager can subscribe events that are not predefined in PSSP, as well as predefined events.

To define an event not predefined in PSSP, Event Management Perspectives is prepared. Perspectives is stored in the /usr/lpp/ssp/bin directory on the Control Workstation. You can register for an event of interest from Perspectives.

For more information about Perspectives and adding events, see Chapter 11, "User-Defined Events" on page 127.

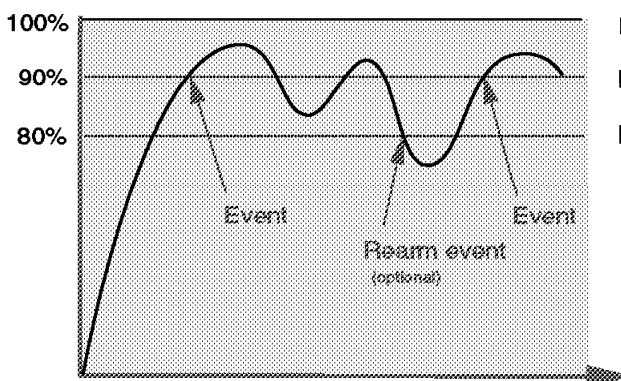
7.2.3 Event Definition Examples

RS/6000

Event Definition Examples

➤ **Event example:**


- ◆ File system /tmp full
- ◆ Resource variable name: IBM.PSSP.aixos.FS.%totused



Instance vector: NodeNum=6

Predicate: X>90


Rearm predicate: X<80



**POWERparallel
Systems**

ITSO Poughkeepsie Center

© Copyright 1999 IBM Corporation



System resources can be monitored by Event Management and events defined in the rules file. Cluster Manager can react to an event and run the recovery program created by the user.

You can specify a user-defined event in the rules file, for example, the file system /tmp is full, as follows:

- Resource variable name: IBM.PSSP.aixos.FS.%totused
- Instance vector: NodeNum=6
- Predicate: X>90
- Rearm predicate: X<80

With these definitions, Cluster Manager is notified of this event when the utilization of the /tmp file system is over 90 percent. After an event happens, the next event will not be notified until the utilization matches the rearm predicate. So if you want to have the event whenever the utilization exceeds 90 percent, you should specify the predicate and rearm predicate as follows:

- Predicate: X>90
- Rearm predicate: X<90

➤ **Event example:**

- ✦ Application process died
- ✦ Resource variable name: IBM.PSSP.Prog.xpcount

Structured Byte String (SBS):

X@0	X@1	X@2
Current number of processes	Previous number of processes	List of process IDs

Instance vector: NodeNum=6;ProgName=upr6;UserName=root

Predicate: X@0==0

Rearm predicate: X@0>0

```
# ps -ef | grep upr
root 7218 5624 0 Jul 18 - 0:00 /usr/sbin/upr6
root 39276 20298 1 10:52:01 pts/8 0:00 grep upr
```



ITSO Poughkeepsie Center
© Copyright 1999 IBM Corporation



Structured Byte String is also available to compare the value of the current resource and previous resource. For example, to get an event when the application process upr6 dies, specify the following:

- Resource variable name: IBM.PSSP.Prog.xpcount
- Instance vector: NodeNum=6;ProgName=upr6;UserName=root
- Predicate: X@0==0
- Rearm predicate: X@0>0

Table 2. Structured Byte String (SBS):

X@0	X@1	X@2
Current number of processes	Previous number of processes	List of process IDs

```
# ps -ef | grep upr
root 7218 5624 0 Jul 18 - 0:00 /usr/sbin/upr6
root 39276 20298 1 10:52:01 pts/8 0:00 grep upr
```

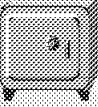
Figure 5. Upr6 Process

7.3 Migrating from Existing HACMP Environment to HACMP ES

RS/6000


Migrating from HACMP to HACMP ES

- **Snapshot**
 - ◆ Uses same ODM classes
- **Migration only from HACMP for AIX V 4.2.1**
- **Scripts work as with HACMP for AIX V 4.2.1**
- **Node name**
 - ◆ Must be in the /etc/hosts file or in a DNS server before starting HACMP ES




1. Save


- ◆ Definitions manually or use snapshot
- ◆ Scripts manually




2. Deinstall HACMP for AIX



3. Install HACMP ES




4. Import saved data



**POWERparallel
Systems**

ITSO Poughkeepsie Center
© Copyright 1997 IBM Corporation



Snapshot

The *snapshot* functionality in HACMP ES is identical to that in HACMP for AIX V4.2.1; it uses the same ODM classes. (Also see “Snapshot” on page 23 and “Global ODM” on page 23.) Snapshots created in HACMP for AIX V4.2.1 can be used in HACMP ES. This compatibility can be used for migration. For more information, see Chapter 10, “Migration” on page 123.

Migration

We recommend that you migrate from HACMP for AIX V4.2.1 to HACMP ES. To perform this migration, you must:

1. Save your cluster information.
2. De-install HACMP for AIX.
3. Install HACMP ES.
4. Restore your cluster information.

You can also migrate from an earlier version of HACMP for AIX to HACMP ES, but this requires additional steps and tests. For more details about migration see Chapter 10, “Migration” on page 123.

HACMP Scripts

As mentioned in “Event Scripts” on page 25, scripts that work in HACMP for AIX V4.2.1 work in HACMP ES. Scripts from an earlier version of HACMP for AIX may require a modification.

Node Names

As mentioned in “Naming” on page 28, the HACMP node name must be resolvable to the hostname. Therefore, the HACMP node name must be equal to the hostname or an alias of the hostname. The alias can be defined in `/etc/hosts` or in Domain Name Server (DNS).

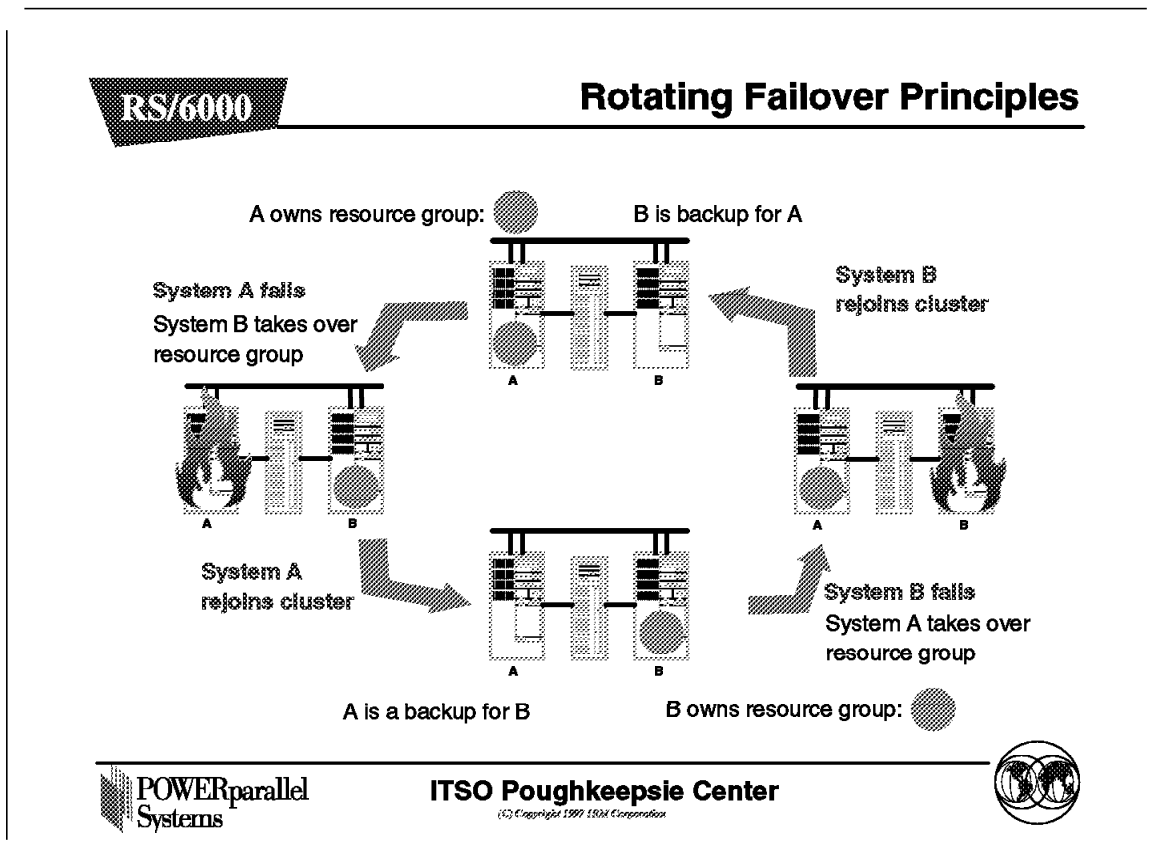
Chapter 8. Configuration Examples

This chapter some general possibilities they are not based on an existing or tested configuration.

8.1 Takeover Scenarios

This section contains takeover scenarios that illustrate the principles of rotating and cascading functions.

8.1.1 Rotating Failover Principles



This picture provides an overview of rotating functionality and shows how it works. It illustrates the basic relationship that makes up a rotating cluster.

The functional flow is as follows:

The bullet (circle) represents the application. The figure in the top middle of this picture represents the initial configuration of the cluster.

- If node A (System A in the picture) fails, node B (System B in the picture) takes over the application (resource) from node A.
- If the problem on node A is fixed and node A rejoins the cluster, nothing happens--the application will still reside on node B.

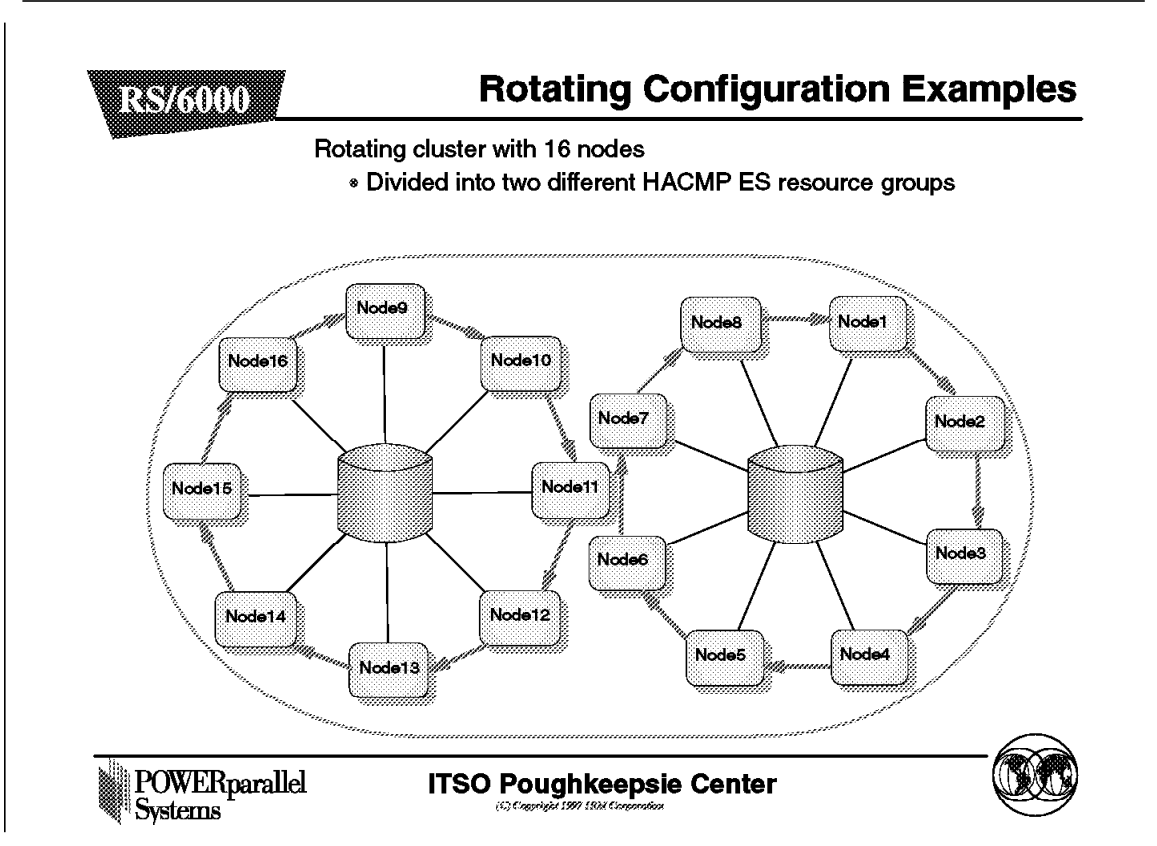
Now we have a fully functional cluster again (see the figure in the lower middle of the picture).

- If node B fails, node A takes over the application (resource) from node B.
- If the problem on node B is fixed and node B rejoins the cluster, nothing happens--the application will still resist on node A.

Once again we have a fully functional cluster (see the figure in the top middle of the picture), which is identical to the initial configuration.

8.1.2 Rotating Takeover Scenario

This section presents rotating takeover scenarios for HACMP ES.



This picture shows a 16 node cluster with two independent rotating resource groups.

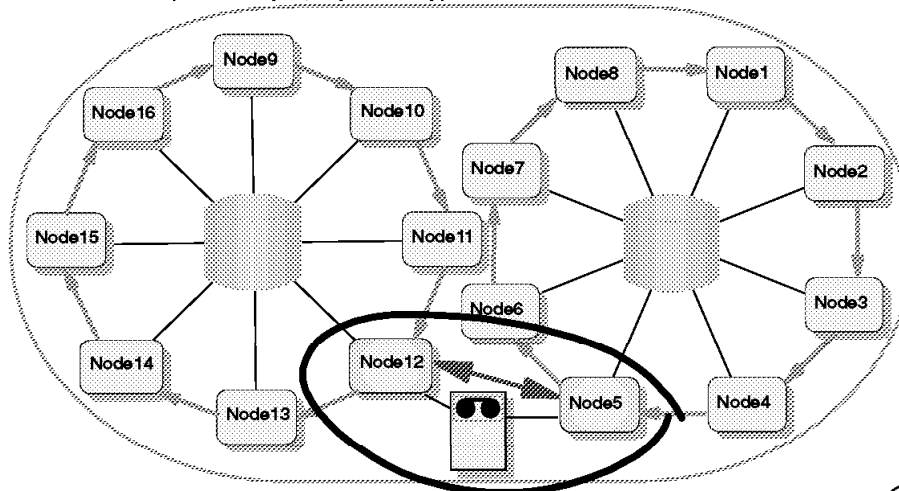
The base configuration is similar to a normal HACMP installation with two 8-node clusters. The main difference is that the two resources now belong to one cluster instead of two clusters.

While the benefits of creating this new configuration rather than using a normal HACMP installation may not be obvious, we will show that this rotating cluster configuration has an advantage in regard to scalability. This advantage is shown in the next picture.

Rotating Configuration Examples (cont'd)

Rotating cluster with 16 nodes

- Divided into two different rotating resource groups
- Two nodes also have a cascading resource group (for example, tape library)



This picture shows the same base configuration as the previous picture: two rotating resources in a 16-node cluster. The main difference here is that node 12 and node 5 share an additional rotating, or cascading, resource.

Such a configuration is not possible with a normal HACMP for AIX installation, because you would need two clusters instead of one.

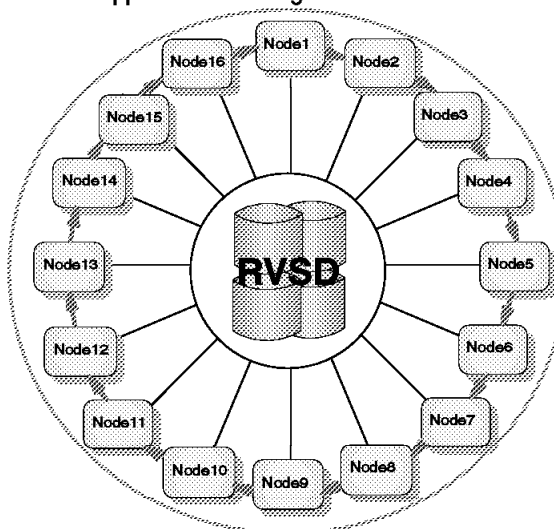
Note: In an HACMP installation, two clusters cannot share a resource.

Now, however, because of the scalability of HACMP ES, configurations like the one shown are possible.

Rotating Configuration Examples (cont'd)

Rotating cluster with 16 nodes

- One VSD/RVSD cluster with multiple disks
- Application running on nodes



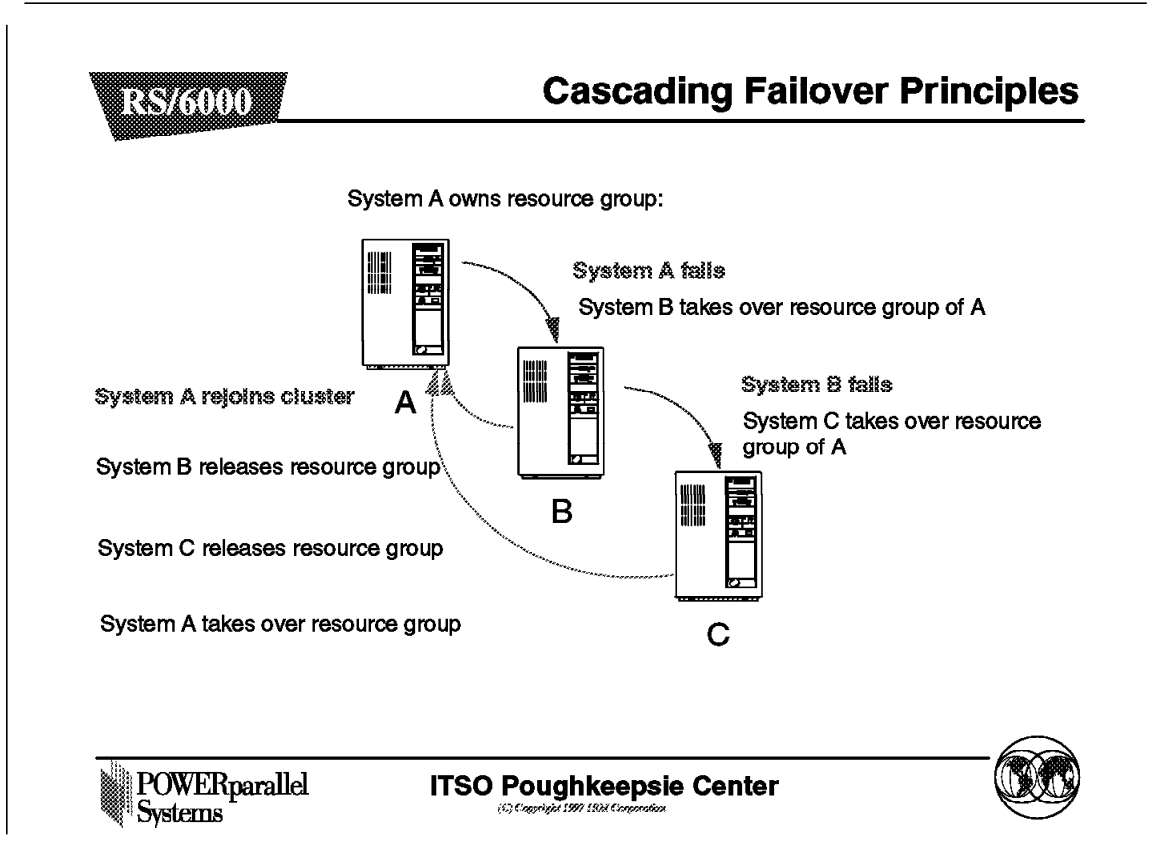
This picture shows how to build a 16-node cluster to share the same logical disk. The combination of the new VSD/RVSD and HACMP ES now allows you to create configurations like this one.

Note: HACMP ES can execute RVSD 1.2 recovery scripts just as HACMP for AIX does. HACMP ES and RVSD 2.1 may see different topology information at any given point in time since RVSD gets topology information from PSSP Topology Services based on two network interfaces. HACMP ES gets topology information from Topology Services/ES, which may be based on many more network interfaces than PSSP Topology Services. RVSD 2.1 provides an option to bring down the cluster manager when it does a takeover, thus causing the HACMP and RVSD failovers to be synchronized.

8.2 Cascading Configurations

This chapter contains cascading configuration examples. The first section illustrates the principles of the cascading function.

8.2.1 Cascading Failover Principles



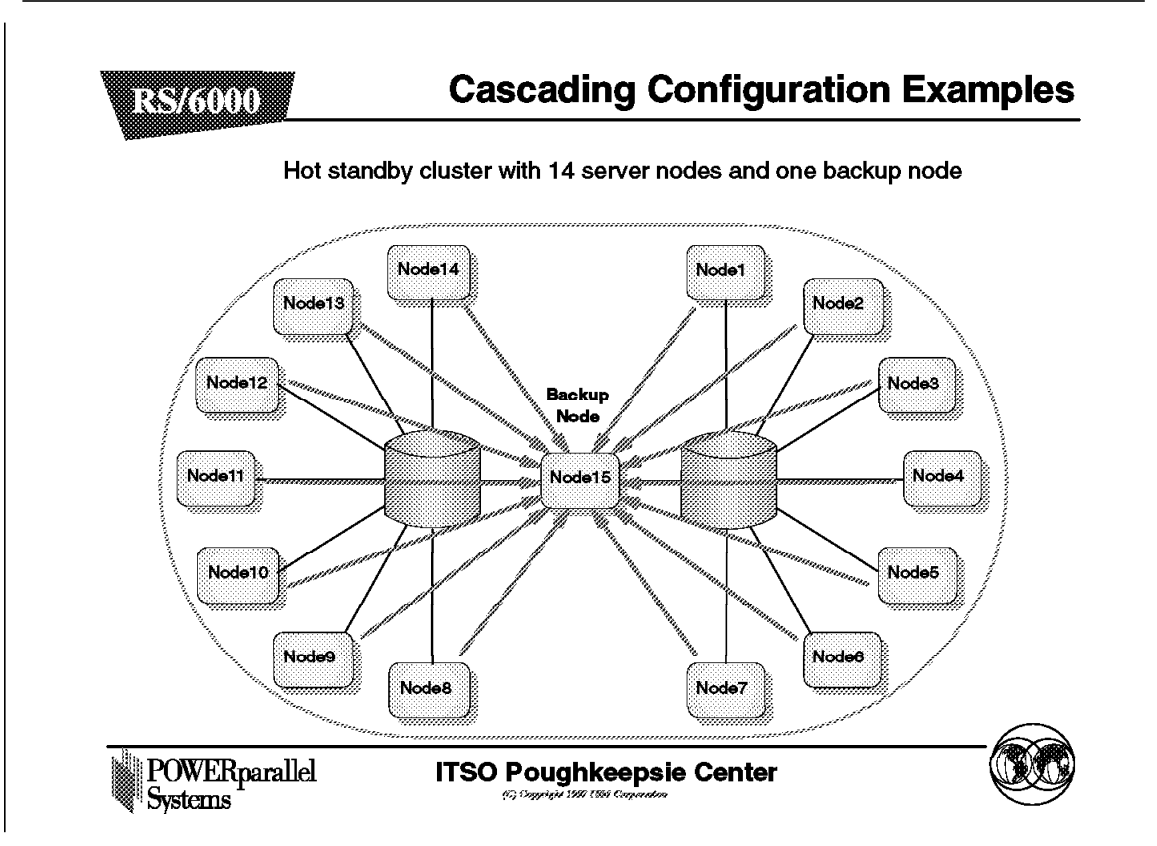
This picture shows a three-node cluster with a cascading relationship. After the initial start of all cluster nodes, there is an application (resource) running on System A. System A is the primary host for the application.

- If System A fails, System B takes over the application of A.
- When the problem is fixed on System A and HACMP is started, the application will be taken over by System A because System A is the primary node.

The following describes what happens if A and B fail sequentially.

- If System A fails, System B takes over the application of A.
- System A is still down, but now System B fails, too.
- The application of System A (which ran on System B) is taken over by System C.
- If System B rejoins the cluster but System A is still down, no takeover happens. The application is still running on System C because A is the primary node for the application.
- When System A rejoins the cluster, the application goes back to System A.

8.2.2 Cascading Configuration



This picture shows a hot standby configuration with 15 nodes in one HACMP ES cluster. One node is the backup node for the 14 other nodes. This configuration is based on SSA limitations.

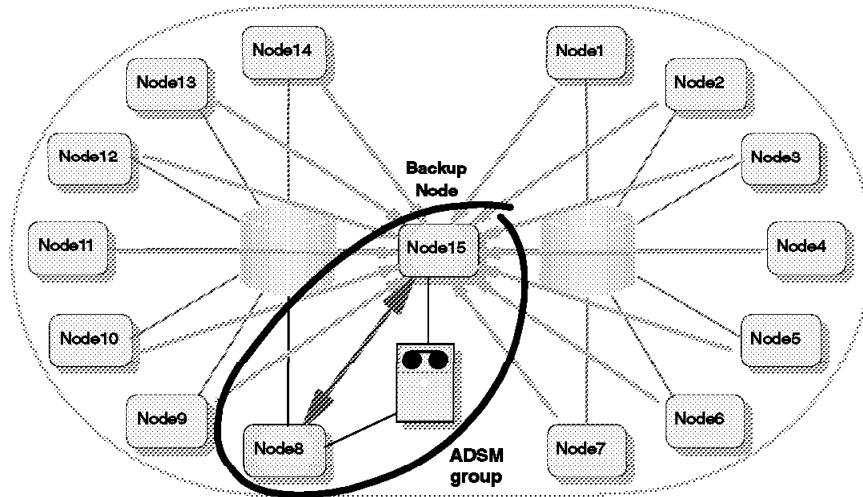
The picture illustrates that now, with HACMP ES, we can build a 1-to-14 relationship between nodes, instead of a 1-to-7 relationship.

Note: Related to the HACMP recommendations, this configuration is not a fully equipped one because you cannot add 14 standby adapters to the other required adapters on the backup node.

Cascading Configuration Examples (cont'd)

Hot standby cluster with 14 server nodes and one backup node

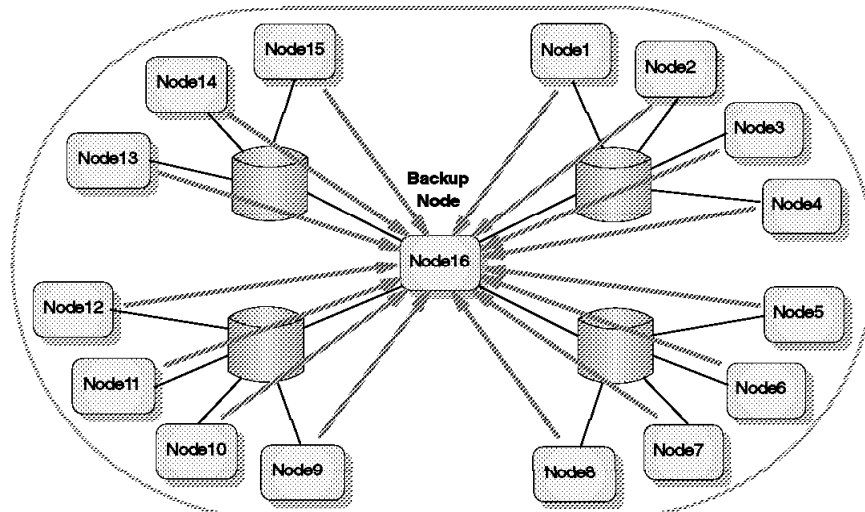
- Two nodes are in mutual takeover resource group (for example, ADSM server group)



This picture shows basically the same cluster as the previous one, but in this case we have an additional resource defined. This configuration is using the hot standby system for additional workload as long as the other nodes are up. It is similar to the configuration shown in 8.1.2, Rotating Takeover Scenario on page 112.

Hot-standby cluster with 15 server nodes and one backup node

- Nodes are using four different shared disks



This picture also shows a hot standby configuration, but it is a 16-node cluster. To achieve better performance and to be able to use the maximum number of nodes (16 for this release), this configuration is split into four resources (this is an SSA-related consideration).

Note: Related to the HACMP recommendations, this configuration is not a fully equipped one because you cannot add 14 standby adapters to the other required adapters on the backup node.

Part 2. Technical Implementations

This part does not contain any presentation pictures. Its aim is to give technical details of the new or changed parts of HACMP ES.

Chapter 14, "Cascading by Using One Network Adapter" on page 143 describes a special case, which is not part of any HACMP product, and can be used as is without any support. It details both a problem we encountered and a workaround we used to resolve it.

Chapter 9. Installation of HACMP ES	121
9.1 Prerequisites	121
9.1.1 Hardware Prerequisites	121
9.1.2 Software Prerequisites	121
9.2 Software Installation and Configuration	122
Chapter 10. Migration	123
10.1 New Installation by Using Scripts from Previous Installation	123
10.2 Using Snapshot	123
10.2.1 Using Snapshot for HACMP for AIX V4.2.1 and Older than V4.1.1	124
10.2.2 Using Snapshot for HACMP V4.1.1 or V4.2	125
Chapter 11. User-Defined Events	127
11.1 Prerequisites	127
11.2 Installation	127
11.3 Configuration	127
11.3.1 HACMP Scripts for an Application	127
11.3.2 The Recovery Program (rp File)	128
Recovery Program Format	128
Barrier	129
11.3.3 The Action Files	129
11.3.4 The rules.hacmprd File	129
Chapter 12. Using Kerberos	131
12.1 Kerberos Overview	131
12.2 Change HACMP to Use Kerberos	131
12.3 How to Kerberize the HACMP Interfaces	132
12.3.1 Using PSSP 2.3 Functions	132
12.3.2 Using Native Kerberos Functions	135
Using the SDR	135
The Switch (css0)	135
Other Network Interfaces	136
Chapter 13. Adding Additional (Unsupported) Interfaces to the SDR	139
Chapter 14. Cascading by Using One Network Adapter	143
14.1 Our Test Environment	143
14.2 Our Workaround	143
14.2.1 Technical Description	143
14.2.2 Advantages	143
14.2.3 Disadvantages	144
14.2.4 Installation	144
14.2.5 Configuration	144
14.2.6 System Requirements	145

14.2.7 Support 146
14.2.8 Using the Modified Event Scripts 147

Chapter 9. Installation of HACMP ES

This chapter offers a brief description of the installation steps for HACMP ES.

- For a new installation, the steps are the same as for HACMP for AIX V4.2.1.
- To migrate from an existing HACMP cluster to HACMP ES, refer to Chapter 10, “Migration” on page 123.

9.1 Prerequisites

The following sections provide a brief overview of the hardware and software prerequisites. These prerequisites relate to the first version of HACMP ES. Requirements may change for future versions.

9.1.1 Hardware Prerequisites

The hardware requirements are basically the same as those for HACMP for AIX, such as having a non-IP network, having more than one adapter per network, and so on.

The following list shows the *differences* in hardware requirements between HACMP ES and HACMP for AIX:

- HACMP ES only runs on an RS/6000 SP
- HACMP ES is partition-bounded. All nodes of an HACMP ES cluster have to be in the same SP partition.

9.1.2 Software Prerequisites

HACMP ES has the following known prerequisites.

For the latest list of software prerequisites, refer to the HACMP ES *Installation and Administration Guide*, as well as the release notes.

- Each cluster node must have AIX 4.2.1 installed.
- The fix for AIX APAR #IX52597 is required.
- IBM Parallel System Support Program (PSSP) Version 2.3 must be installed on the Control Workstation and the SP nodes.
- The following AIX optional bos components are mandatory for HACMP ES:
 - bos.compat.lan
 - bos.compat.links
 - bos.net.nfs.client
 - bos.net.nfs.server
 - bos.net.tcp.client
 - bos.net.tcp.server
 - bos.sysmgmt.trace
- Each cluster node requires its own HACMP ES software license.
- The /usr file system must have a minimum of 15MB free space (or the volume group must have enough space to extend it by 15MB).
- The ./ (root) file system must have a minimum of 1MB free space (or the volume group must have enough space to extend it).
- The installation process must be performed by the root user.

- The latest service level of PSSP 2.3 should be installed.

9.2 Software Installation and Configuration

Before you start the installation and configuration of your HACMP ES cluster, the following conditions should be met:

- The planning should be finished.
- The necessary documentation (planning worksheets) should be available.
- The hardware should be connected together.

The following lists the main steps for installation and configuration of HACMP ES. These steps are the same as for the installation of HACMP for AIX V4.2.1.

For a detailed description of these steps, refer to HACMP ES *Installation and Administration Guide*.

1. Make the installp images available on the nodes (nfs or ftp).
2. Install the HACMP ES code.
3. Define the .rhosts file or Kerberos (this can be done before you start the installation, but it has to be done before you issue the first HACMP synchronization command).
4. Define the cluster topology:
 - a. Add the cluster ID and name.
 - b. Add the cluster node names.

Note: In HACMP ES, the node name must be resolvable. It can be an alias to the hostname, or the hostname itself.

- c. Add the adapters.
 - d. If Kerberos will be used, go to resource definitions and change the runtime parameter from Standard to Enhanced.
 - e. Synchronize the topology with the other nodes.
5. Define the cluster resources:

The following list contains only the steps you must do before you are able to start HACMP ES. There may be more steps required for your installation, like defining an application.

- a. Add a resource group.
 - b. Configure resources for the resource group.
 - c. Synchronize the resources with the other nodes.
6. Verify the cluster nodes.

Now you can start the first node of your cluster. We recommend you start the nodes separately for the first time, because this makes it easier to determine if everything is configured correctly.

Chapter 10. Migration

There is no real “migration” from HACMP to HACMP ES, because the HACMP code has to be de-installed before HACMP ES can be installed.

However, there are several ways to move to HACMP ES, as follows:

- Perform a completely new installation of HACMP ES, (in which case, you don’t care about the old code).
- Perform a new installation of HACMP ES by saving the modified or added scripts.
- Use snapshot.

We do not describe the first item because it is the same as installation. The only additional step you have to do is to remove the HACMP code before you start the installation.

The following sections describe what you have to do for the other two possibilities.

10.1 New Installation by Using Scripts from Previous Installation

This installation is similar to a new installation. Follow these steps before you start the procedure described in 9.2, “Software Installation and Configuration” on page 122.

1. Back up your system.
2. Copy your scripts to /tmp or to any other directory that is not under /usr/sbin/cluster.

Note: The /usr/sbin/cluster directory and everything in it will be removed by the de-installation of HACMP for AIX.

3. Save the output of the configuration listings in a file or as a printout.

Note: These listings may be used as input for the retyping of the configuration in HACMP ES.

4. De-install the HACMP for AIX code by using SMIT or command line input.
5. Now you are ready to follow the procedures in 9.2, “Software Installation and Configuration” on page 122.

When you have restored your script files, you have to test to see if these files will work as expected in the new environment.

Note: If you migrate from HACMP for AIX V4.2.1, every script should work fine. If you migrate from an earlier HACMP for AIX version, some scripts may not work properly.

10.2 Using Snapshot

This section is split into two parts. The differences between these parts are related to your existing HACMP version that you want to migrate from.

- Using HACMP for AIX V4.2.1, or an HACMP version lower than HACMP for AIX V4.1.1.

In this case, continue with 10.2.1, “Using Snapshot for HACMP for AIX V4.2.1 and Older than V4.1.1” on page 124.

- Using HACMP for AIX V4.1.1 or HACMP for AIX V4.2.

In this case continue with 10.2.2, “Using Snapshot for HACMP V4.1.1 or V4.2” on page 125.

10.2.1 Using Snapshot for HACMP for AIX V4.2.1 and Older than V4.1.1

Using snapshot to migrate assumes that many requirements have been met before you start installing HACMP ES.

Once these requirements are met, the following steps will help you to migrate from HACMP for AIX to HACMP ES. We also provide hints and tips to perform a successful migration.

Note: Snapshot can only be used for migration from HACMP for AIX V4.2.1 to HACMP ES.

1. Migrate to HACMP for AIX V4.2.1.

If you already use HACMP for AIX V4.2.1, continue with 3.

If you are using an earlier HACMP for AIX version, you first have to migrate to HACMP for AIX V4.2.1. This can be done by the utilities delivered with HACMP.

2. Test the HACMP for AIX V4.2.1 installation.

Before you continue, you must test your installation. Most user scripts will work properly, but if you have modified HACMP scripts, or scripts that work very closely with HACMP scripts, they may no longer work. These scripts will have to be modified to work with HACMP for AIX V4.2.1.

Note: We could not find any script which did not work in HACMP ES if it worked in HACMP for AIX V4.2.1 .

3. Create the snapshot.

Before you create the snapshot, you may change the node names in HACMP, or add the node names to the /etc/hosts file or to the nameserver in your DNS environment.

Note: In HACMP ES, the node name must be resolvable. It can be an alias to the hostname, or the hostname itself.

4. Save the snapshot file.

The snapshot files are stored in the /usr/sbin/cluster/snapshots directory. Copy these files to /tmp or to any other directory that is not under /usr/sbin/cluster.

Note: The /usr/sbin/cluster directory and everything in it will be removed by the de-installation of HACMP for AIX

5. Back up the system.

To backup the system is highly recommended in case something goes wrong.

6. Save user scripts.

Save all the script files you have added or modified, because the de-install of HACMP may remove them.

Note: We couldn't find any script which did not work in HACMP ES if it worked in HACMP for AIX V4.2.1 .

7. **Stop the HACMP cluster.**

We recommend you stop the complete HACMP cluster, because HACMP is going to synchronize all cluster nodes even if you install the snapshot on only one system.

Note: The synchronization will work even if only one system has HACMP ES installed and all the other systems still have HACMP for AIX V4.2.1 installed. You may experience problems on the HACMP for AIX V4.2.1 systems, however, because the ODM classes are the same on both HACMP versions.

8. **De-install HACMP for AIX V4.2.1.**

De-install HACMP for AIX V4.2.1 on all nodes of this HACMP cluster. By using SMIT or the command line.

9. **Install the HACMP ES code.**

Install HACMP ES on all nodes of this HACMP cluster. To install HACMP ES, use SMIT or command line input.

10. **Restore the saved files.**

Restore the saved script files and the snapshot files. The script files have to be restored to all nodes in this HACMP cluster. The snapshot files have to be at least on one of the nodes.

11. **Apply the saved snapshot.**

To apply the snapshot, all nodes of the HACMP cluster have to be up and reachable through the network.

Note: The .rhosts file or Kerberos have to be set up properly.

12. **Test the installation.**

Test your installation to make sure that everything works as expected.

10.2.2 Using Snapshot for HACMP V4.1.1 or V4.2

Using snapshot to migrate assumes that many requirements have been met before you start installing HACMP ES.

Once these requirements are met, the following steps will help you to migrate from HACMP for AIX to HACMP ES. We also provide hints and tips to perform a successful migration.

Note: Snapshot can only be used for migration from HACMP for AIX V4.2.1 to HACMP ES.

1. **Create a snapshot of the 4.1.1 or 4.2 cluster.**

Before you create the snapshot, you may change the node names in HACMP, or add the node names to the /etc/hosts file or to the nameserver in your DNS environment.

Note: In HACMP ES, the node name must be resolvable. It can be an alias to the hostname, or the hostname itself.

2. **Save the snapshot file.**

The snapshot files are stored in the /usr/sbin/cluster/snapshots directory. Copy these files to /tmp or to any other directory that is not under /usr/sbin/cluster.

Note: The `/usr/sbin/cluster` directory and everything in it are removed by the de-installation of HACMP for AIX.

3. **Back up the system.**

It is highly recommended to backup the system.

4. **Stop the HACMP cluster.**

To stop the cluster, use SMIT or the command line.

5. **Save user scripts.**

Save all the script files you have added or modified, because the de-install of HACMP may remove them.

6. **De-install HACMP.**

De-install HACMP on all nodes of this HACMP cluster, by using SMIT or the command line.

7. **Install the HACMP ES code.**

Install HACMP ES on all nodes of this HACMP cluster. To install HACMP ES, use SMIT or command line input.

Note: Make sure `ssp.topsvcs` is installed; install it if it is not.

8. **Restore the snapshot file.**

Restore the saved snapshot file to `/usr/sbin/cluster/snapshots`.

9. **Run `clconvert_snapshot`.**

Run the `clconvert_snapshot` program against the 4.1 snapshot.

10. **Restore the saved user scripts.**

Restore the saved script files to all nodes in the cluster.

Note: These scripts may require modifications to work with the new code. We noticed that most of them will work without any modification.

11. **Apply the saved snapshot.**

To apply the snapshot, all nodes of the HACMP cluster have to be up and reachable through the network.

Note: The `.rhosts` file or Kerberos have to be set up properly.

12. **Synchronize the cluster.**

Make sure that all your nodes are synchronized.

13. **Test the installation.**

Test your installation to make sure that everything works as expected.

14. **Start the cluster.**

Chapter 11. User-Defined Events

In this chapter we describe the steps to take if you want to use *user-defined events*.

11.1 Prerequisites

The prerequisites related to hardware and software are the same as the requirements for the PSSP software and the HACMP ES software. There is no additional hardware and software prerequisite for the user-defined events function.

To create or manage user-defined events, the system administrator has to have a good understanding of event handling (PSSP 2.2+), HACMP, and AIX. Familiarity with Chapter 7, “HACMP ES Installation and Customization” on page 93 is also required.

11.2 Installation

This is the easiest part because everything you need was installed when you installed PSSP 2.3 and HACMP ES.

To test your event definitions, you should have ssp.pman installed on your systems. We recommend that you install it on at least two of your test systems.

11.3 Configuration

This section guides you in defining your user events for HACMP ES, based on the example we chose.

Before starting with any modification, you should back up your system first. We highly recommend making a copy of the `/usr/sbin/cluster/events/rules.hacmprd` file.

Note: The `rules.hacmprd` file must be the same on all nodes in a cluster.

In our example, we used the `rwhod` daemon like an application. We called it the “`rwho`” application. All the script files we used for this application can be found in A.2, “Scripts for User-Defined Events” on page 164.

11.3.1 HACMP Scripts for an Application

For our `rwho` application, we created a start script (see A.2.1, “The Start Script for the HACMP Application” on page 164) and a stop script (see A.2.2, “The Stop Script for the HACMP Application” on page 164). These are very simple scripts that use the `startsrc` and `stopsrc` commands. For event detection to work properly, it was also necessary to have a test file, which was required to determine whether HACMP stopped the application or whether the application died unexpectedly.

You should test your application’s start and stop scripts with and without HACMP before you do any modification for event detection.

11.3.2 The Recovery Program (rp File)

The recovery program, or xxx.rp file, is where you define the programs that have to run in case of an event. You can specify on what kind of nodes the programs will run. For example, on all nodes, or only where the event happens. The following section explains the format of the xxx.rp file. The recovery program we used for our installation can be found in A.2.4, “The rwho.rp File” on page 168.

Recovery Program Format

The format of the recovery program is:

```
relationship command_to_run expected_status NULL
```

There has to be at least one blank between the values.

- **Relationship**

Relationship is a value that is used to decide which program should run on which kind of node. The following three types are supported:

- All

The specified command/program is executed on all nodes of the current HACMP cluster.

- Event

The specified command/program is executed on the node where the event occurred, only.

- Other

The specified command/program is executed on all the nodes where the event did not occur.

- **command_to_run**

This is a quote-delimited string with or without a full path definition. to an executable program. A none full path definition is used for the HACMP delivered event scripts only. For using other scripts or programs you have to use the full path definition. This is true even if these programs are located in the same directory as the HACMP event scripts.

- **expected_status**

This is the return code of the specified command/program. It is an integer value or an X. If you use an X, Cluster Manager does not care about the return code.

For all other values:

The return code has to be equal to the expected one. If it is not, Cluster Manager detects the event failure. The handling of this event will hang until the problem is solved via manual intervention to recover. The reason for this is that this node does not hit the barrier, and all the other nodes are waiting till the node with the failure hits the barrier.

- **Null**

Null is a reserved field for a future release. The word NULL must appear at the end of each of these lines, except the barrier line.

If you specify multiple recovery commands between two barrier commands, or before the first one, the recovery commands are executed in parallel, both on the node itself and between the nodes.

Barrier

The barrier is intended to be the synchronization point for all the specified commands before it. When a node hits the barrier statement in the recovery program, Cluster Manager initiates the barrier protocol on this node. When all nodes have met the barrier in the recovery program and voted to approve the protocol, Group Services notifies all nodes that this protocol has completed. The next part of the recovery program is executed. The barrier itself is a two-phase protocol.

11.3.3 The Action Files

These are the files we used for our rwho application. Their function is very simple: to send out messages and restart the application via the `startsrc` command. The following lists the file names and where you can find them:

- `rwho_msg_local` (see A.2.5, “The `rwho_msg_local` File” on page 168)
- `rwho_msg_remote` (see A.2.6, “The `rwho_msg_remote` File” on page 168)
- `rwho_msg_complete` (see A.2.7, “The `rwho_msg_complete` File” on page 168)
- `rwho_msg_restart` (see A.2.8, “The `rwho_restart` File” on page 169)

11.3.4 The `rules.hacmprd` File

In this version of HACMP ES, the user event definition has to be done to a flat file by using a normal editor like `vi`. This may change in following releases.

Note: Before you start editing this file, make sure that you have a backup copy.

The `rules.hacmprd` file is a very sensitive file. Every user event you add *must have exactly nine lines*. The comment lines are not counted. If you have one line more or less, the system will hang. If this happens, change the file or copy your backup to it and then reboot your system.

Note: Do not add or remove any blank line.

The `rules.hacmprd` file contains the following items for each event. Some of these items can be a blank line. For additional information regarding this list, see 4.3.1, “Rules File” on page 51.

1. Name
2. State (qualifier)
3. Recovery program path
4. Recovery type (reserved for future use)
5. Recovery level (reserved for future use)
6. Resource variable name (used for Event Manager events)
7. Instance vector (used for Event Manager events)
8. Predicate (used for Event Manager events)
9. Rearm predicate (used for Event Manager events)

The following figure shows the definitions we added to the existing `rules.hacmprd` file. The listing of the complete file we used can be found in A.2.3, “The `rules.hacmprd` File” on page 165.

```
#
# Definition of
# Monitor death of rwhod daemon on node 5
#
UE_PGMRWHO_DOWN
0
/usr/sbin/cluster/local/rwho.rp
2
0
IBM.PSSP.Prog.xpcount
NodeNum=*;ProgName=rwhod;UserName=root
X@0==0&&X@1!=0
X@0>0
#
# End of definition.
```

Chapter 12. Using Kerberos

HACMP for AIX V4.2.1 and HACMP ES now allow you to use the Kerberos function instead of using the .rhost file. This chapter discusses this possibility.

12.1 Kerberos Overview

The RS/6000 SP currently uses MIT Kerberos version 4. Kerberos functions as a third party to authenticate the identities of clients and servers. Kerberos on the RS/6000 SP is used to initially authenticate the identity of a user and to provide information through which the server can authenticate the identity of a client in a distributed environment. The underlying mechanism for authentication of users is a ticket scheme.

Kerberos provides an authenticated (kerberized) version of rsh and rcp (HACMP for AIX V4.2.1 and HACMP ES are now able to use this rsh). Using Kerberos avoids the need to have a .rhosts file to control access to network services such as rsh and rcp. On a RS/6000 SP, dsh, pcp and sysctl are kerberized, too.

For a brief description of Kerberos and its components, see Chapter 3, "Kerberos," in the *RS/6000SP: Problem Determination Guide*, SG24-4778.

And for a more detailed discussion of how Kerberos works, see Chapter 14, "Understanding Secure Authentication," in *RS/6000 Scalable POWERparallel Systems: PSSP Version 2 Technical Presentation*, SG24-4542.

Or see *IBM Parallel System Support Program for AIX Administration Guide*, GC23-3897.

12.2 Change HACMP to Use Kerberos

HACMP for AIX V4.2.1 and HACMP ES, by default, use the .rhost functionality. To use the Kerberos-based functions, you have to change the run-time parameters in HACMP. You can do this with the command:

```
/usr/sbin/cluster/utilities/clchparam
```

or via SMIT:

```
# smit hacmp
=> Cluster Config
=> Cluster Resources
=> Change/Show Run Time Parameters
```

After selecting the node, you see the following screen: (Figure 6 on page 132)

```

Change/Show Run Time Parameters

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

Node Name                               [Entry Fields]
                                         sp2n05
Debug Level                             high          +
Host uses NIS or Name Server            false         +
Cluster Security Mode                   Standard      +

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

```

Figure 6. HACMP Run Time Parameters

Now go to the Cluster Security Mode line and change Standard to Enhanced. This can be done by using the Tab key or F4.

Note: You have to do this for all nodes in this cluster.

Before you can use the Kerberos functions, your boot and service interfaces must be kerberized. In the next section we describe how to do this.

12.3 How to Kerberize the HACMP Interfaces

This section is split into two parts. The first describes the steps you should do for HACMP ES and HACMP for AIX V4.2.1 when using PSSP 2.3. The second describes how you can do this for earlier PSSP versions.

Using PSSP 2.3 means you have it installed on your Control Workstation (CWS). You can use the description in 12.3.1, “Using PSSP 2.3 Functions” even if you have PSSP 2.3 installed on the CWS only. It does not matter which PSSP version is used on the nodes.

12.3.1 Using PSSP 2.3 Functions

The new PSSP 2.3 version enables you to define more than one IP address to an interface in the SDR. The additional addresses are used for Kerberos only.

In this release we are still restricted to the same number of interfaces as before:

- css0
- en1
- tr0
- tr1
- fi0
- fi1

The following description can also be used as workaround to get all other interfaces kerberized.

You can add the addresses for the en2 adapter (and so on) to the other_addr list for en1 (or tr0, tr1, fi0, fi1) since the adapter type is not used by Kerberos. Another possible workaround is described in Chapter 13, “Adding Additional (Unsupported) Interfaces to the SDR” on page 139.

The enhancement to add additional addresses to the SDR is not imbedded in SMIT yet. But it will be added to SMIT in a following release. At this time this can only be done by using the command line. The following steps describe how to do it.

1. Define the initial TCP/IP address in the SDR:

```
/usr/lpp/ssp/bin/spadaptrs
```

or via SMIT:

```
# smit enter_data
=> Node Database Information
=> Additional Adapter Information
```

After typing in some values, your screen may look like as follows:

```

Additional Adapter Database Information

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

                                     [Entry Fields]
Start Frame                          [1]                               #
Start Slot                            [1]                               #
Node Count                            [16]                              #

OR

Node Group                            []                               +

OR

Node List                              []

* Adapter Name                        [en1]
* Starting Node's IP Address or Hostname [10.15.1.1]
* Netmask                             [255.255.255.0]
Ethernet Adapter Type                  +
Token Ring Data Rate                   +
Skip IP Addresses for Unused Slots?    no                            +
Enable ARP for the css0 Adapter?       no                            +
Use Switch Node Numbers for css0 IP Addresses? yes                       +

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

```

Figure 7. Add Additional Adapter Information

Note: We recommend that you use the boot address here.

2. Define the additional TCP/IP address to this interface. In this case the service address. To do this, use the following command:

```
SDRChangeAttrValues Adapter node_number==<NO> adapter_type==<Network> \
other_addr==<Address>
```

Where:

<NO> is the number of the node.

<Network> is the network interface to which the IP address(es) should be added, for instance en1 or tr0.

<Address> is the address or addresses to be added.

Note: The addresses are separated by a comma.

The following example shows how to add two TCP/IP addresses to en1 on Node 9:

```
SDRChangeAttrValues Adapter node_number==9 adapter_type==en1 \
other_addrs=129.40.162.67,129.40.162.68
```

Notes:

- a. There is only one "=" for the other_addrs attribute.
 - b. If there is already an address defined to other_addrs you have to add it to the list of other_addrs otherwise this address will be deleted.
3. If you have more interfaces, repeat the step until you are finished.
 4. Run setup_server.

The following example uses SMIT:

```
# smit enter_data
=> Node Database Information
=> Boot/Install/usr Server Information
```

After typing in some values, your screen should look similar to the following:

```

                                     Boot/Install/usr Server Information
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                     [Entry Fields]
Start Frame                         [1]                               #
Start Slot                           [1]                               #
Node Count                           [16]                              #

OR

Node Group                           []                               +

OR

Node List                             []

Boot/Install Server Node Identifier  []
Network Install Image Name           []
Destination Hard Disk(s)              []
Response from Server to bootp Request customize                       +
LPP Source Name                       []
PSSP Level                             []                               +

/usr Server's Hostname or IP Address  []
Gateway to /usr Server                []
/usr Client Adapter Name               []                               +

Run setup_server on the Control Workstation?  yes                       +

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

Figure 8. Set setup_server Information

5. Reboot the nodes on which you changed the interfaces.

12.3.2 Using Native Kerberos Functions

This description is for all who are not able to install PSSP 2.3 on the Control Workstation (CWS).

Note: PSSP 2.3 is not required to be installed on the nodes to use the function described in 12.3.1, “Using PSSP 2.3 Functions” on page 132.

In our first beta code of the new PSSP 2.3, we could not use the enhanced IP definition function. Therefore, we used the following description to kerberize our boot and service addresses. We also used it for our reference installation (HACMP for AIX V4.2.1 on PSSP 2.2), which we used for internal tests only.

We are including this description here for those of you who have to use an earlier PSSP version with HACMP for AIX.

The `css0` and `en0` interfaces are automatically kerberized during installation by using the `setup_server` command and booting the node. We can use these functions for some of the interfaces, but not for all. The following sections describe:

- Using the existing functions (SDR)
- Using `css0` together with HACMP
- All the other interfaces

Using the SDR

This is the easiest one to use, relative to Kerberos. If an interface is defined to the SDR, the `setup_server` program automatically creates all the necessary Kerberos information for this interface. In this release and the earlier one, only the following interfaces can be used with this method (for a workaround, see Chapter 13, “Adding Additional (Unsupported) Interfaces to the SDR” on page 139):

- `en1`
- `fi0`
- `fi1`
- `tr0`
- `tr1`
- `css0`

Another restriction in this area is that we can define only one IP address to these interfaces, but HACMP requires two (boot and service).

We recommend that you define the boot interface to the SDR, and that the hostname points to `en0` or `css0`. We know that there are some customers who want to have the hostname of the nodes pointing to the service address of a user interface like `tr0` or `fi0`. Normally, however, there are easy workarounds by setting up the DNS and/or the `/etc/hosts` file correctly.

The Switch (`css0`)

If you want to use HACMP to do the IP take over functionality on the switch, you cannot use the address that is defined in the SDR for the `css0` interface. In this case you have to define the HACMP boot and service interfaces as aliases to the existing `css0` interface. This can easily be done by adding the IP addresses of the boot and service interfaces to the `/etc/hosts` file or to the DNS system, and defining the boot and service interfaces to HACMP. All additional steps are done by HACMP, except for the Kerberos part.

To kerberize the boot and service addresses of the switch, you have to do the steps described in “Other Network Interfaces” on page 136. These steps are the same as those required for all the network interfaces that are normally not definable to the SDR at this time.

Other Network Interfaces

This section describes how to kerberize all the network interfaces that can normally not be defined to the SDR. For the network interfaces en2, en3, and so on, you may want to use the workaround described in Chapter 13, “Adding Additional (Unsupported) Interfaces to the SDR” on page 139. But if you have more than one TCP/IP address for the same interface, we recommend that you use the following description.

The method we describe is not the only possible one, but it is one of the safest. For this method the following steps are needed to define the interfaces to Kerberos:

1. Add the IP addresses of all the interfaces you need to the /etc/hosts file or to the DNS system. If you use /etc/hosts, you have to do this on all nodes and on the Control Workstation (CWS).
2. Define the network interfaces (en2, en3, and so on) to AIX by using standard AIX commands. Do this on all nodes.
3. Define the interfaces you want to be kerberized to the Kerberos database on the CWS. This is done by adding each interface (boot/service) as a principal to Kerberos via the kadmin command, as follows (before starting, add the /usr/lpp/ssp/kerberos/etc path to your path variable, if not already done):

```
kadmin -m
```

Note that -m allows multiple requests without reauthentication (reentry of your administrative password). Your screen will look as follows:

```
sp2cw0/> kadmin -m
Welcome to the Kerberos Administration Program, version 2
Type "help" if you need it.
admin:
```

Figure 9. kadmin

By typing help you get the following:

```
admin: help
Welcome to the Kerberos administration program.Type "?" to get
a list of requests that are available. You can get help on each of
the commands by typing "help command_name". Some functions of this
program will require an "admin" password from you. This is a password
private to you, that is used to authenticate requests from this
program. You can change this password with the "change_admin_password"
(or short form "cap") command. Good Luck|
admin:
```

Figure 10. kadmin (help)

By using ?, you get the following:

```

admin: ?
Available admin requests:

change_password, cpw      Change a user's password
change_admin_password, cap
                           Change your admin password
add_new_key, ank          Add new user to kerberos database
get_entry, get           Get entry from kerberos database
destroy_tickets, dest     Destroy admin tickets
help                     Request help with this program
list_requests, lr, ?     List available requests.
quit, exit, q           Exit program.
admin:

```

Figure 11. kadmin (?)

Now you have to enter the request you want. Your command should look like this:

```
ank rcmd.<interface>
```

Where:

<interface> is the name you use for this IP interface in the /etc/hosts file or in the DNS.

Note: It can also be an alias name of this IP address.

For example:

```
ank rcmd.node2_svc
```

After typing in this command, you will be prompted for the admin password. This password is required to build the Kerberos database. Now you have to enter a password for this new kerberos principals, you will never need it again. Your screen may now look like the following:

```

admin: ank rcmd.node2_svc
Admin password:
Password for rcmd.node2_svc:
Verifying, please re-enter Password for rcmd.node2_svc:
rcmd.node2_svc added to database.
admin:

```

If you used the -m flag, you can now continue defining all the other IP interfaces without retyping the administrator password. When you are finished, type a "q" to terminate this action.

4. Now set Response from Server to bootp Request to **customize** or **install** (if it is a new installation). and run setup server. This will create files like the following in the /tftpboot directory:

```
<nodename>-new-srvtab
```

Where:

<nodename> is the hostname of your nodes, for example, node02.

Note:

This will only create the keys for the interfaces defined in the SDR.

By using SMIT the parameter Run setup_server on the Control Workstation? must be on **YES** otherwise setup_server will not run.

5. Now add all the other IP interfaces you want to have kerberized to the srvtab file, using the following procedure:

- a. To create the key for this interface, use:

```
ext_srvtab <interface>
```

Where:

<interface> is the name you use for this IP interface in the /etc/hosts file or in the DNS.

Note: It can also be the alias name to this IP address, for example, node2_svc.

This command will prompt you for the administrator password. Your screen may now look like the following:

```
sp2cw0/> ext_srvtab node2_svc
Enter the Kerberos master key:
Generating 'node2_svc-new-srvtab'....
sp2cw0/>
```

Repeat this for each IP interface and for each node.

Note: If you use `ext_srvtab -n <interface>`, you will not be prompted for the Kerberos master key. The meaning of `-n` is “no prompt for master key.”

- b. Now you have to append the created srvtab files to the one in the /tftpboot directory. This can be done with the cat command. For example:

```
cat node2_svc-new-srvtab >> /tftpboot/node02-new-srvtab
```

Repeat this for each IP interface and for each node.

6. Before you boot or reboot the nodes, you have to copy (save) your /tftpboot/<nodename>-new-srvtab files to another directory. This is necessary for later use.

Where:

<nodename> is the name of the IP interface you used in the previous step. In our example it is node02.

Note: If you run `setup_server` with the options “customize” or “install” (if it is a new installation), the /tftpboot/<nodename>-new-srvtab files will be recreated and the content of an existing one will be destroyed.

7. Now you can boot or reboot your nodes.

Chapter 13. Adding Additional (Unsupported) Interfaces to the SDR

For our HACMP tests we could not add a second Ethernet adapter (standby) to the SDR, because PSSP 2.3 is still limited to the following interfaces:

- en1
- fi0
- fi1
- tr0
- tr1
- css0

We used the following description to add en2 to the SDR anyway, **but remember this is only a workaround we used; it is not supported.**

The benefit of this workaround is that the additional interface is defined to the node and is automatically kerberized.

You can modify the `/usr/lpp/ssp/bin/spadaptrs` perl script or use a modified copy of it. Before you change this script, make sure you have a backup copy available. You can change the first part as in the following example.

Note: If you modify the `/usr/lpp/ssp/bin/spadaptrs` script, the installation of a PTF may destroy your modifications.

We modified `/usr/lpp/ssp/bin/spadaptrs` and kept the original in a separate file. We did this to be able to use SMIT panels for defining adapters to the SDR.

The following screen shows the original part that will be modified:

```
...
%VALID_ADAPTERS = ('css0', '1', 'en1', '1', 'fi0', '1', 'fi1', '1',
'tr0', '1', 'tr1', '1');
...
```

The following screen shows the changes to this part. Here we made en2 and tr2 available to be used by this script.

```
...
%VALID_ADAPTERS = ('css0', '1', 'en1', '1', 'fi0', '1', 'fi1', '1',
'tr0', '1', 'tr1', '1', 'en2', '1', 'tr2', '1');
...
```

You can still use SMIT, but the `spadaptrs` script will fail if you use some additional flags. In our case we can not use the `-t` flag. As soon as you specify the value `dix` or `bnc` for Ethernet Adapter Type in SMIT, the `-t` is used and the script will fail.

Figure 12 on page 140 shows the screen we used.

```

Additional Adapter Database Information

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

                                [Entry Fields]
Start Frame                      [1]                      #
Start Slot                       [1]                      #
Node Count                       [16]                     #

OR

Node Group                       []                        +

OR

Node List                        []

* Adapter Name                   [en2]
* Starting Node's IP Address or Hostname [10.12.11.1]
* Netmask                        [255.255.255.0]
Ethernet Adapter Type            +
Token Ring Data Rate             +
Skip IP Addresses for Unused Slots? no          +
Enable ARP for the css0 Adapter? no           +
Use Switch Node Numbers for css0 IP Addresses? yes +

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

```

Figure 12. SMIT (SDR Additional Adapter Database Information)

You can also use the command line if you like. In our example, the command line looks as follows:

```
/usr/lpp/ssp/bin/spadptrs 1 1 16 en2 10.12.11.1 255.255.255.0
```

Disadvantages

The following is a list of disadvantages we noticed. There may be more.

- The changes to the splstdata file may be overridden by a PTF installation.
- One big disadvantage of this is that the added interface cannot be seen by the splstdata -a command.
- The output of the splstdata -a command may show other values as they are really defined in the SDR. You may get confused.

If this problem appears delete the css0 definition and define it again, because the css0 definition must be the last one in the SDR, in this case only.

Note: These disadvantages only apply to the browsing of the data in the SDR by using splstdata. The values in the SDR are still valid and usable. If you use SDRGetObjects Adapter, you see all the values as they are defined.

Conclusion

The restrictions we have at this time will be solved a future release. We cannot recommend a workaround that requires modifications of the existing code. The only safe way we can recommend is to configure the additional networks after

the installation of the node by using standard AIX commands and using the methods described in Chapter 12, “Using Kerberos” on page 131 for kerberizing these interfaces.

Chapter 14. Cascading by Using One Network Adapter

In this chapter we describe a problem we encountered and the solution we adapted. This problem is not unique to HACMP ES; it can appear on all HACMP installations. **Remember this is only a workaround we used; it is not supported.**

14.1 Our Test Environment

For our test environment, we had one RS/6000 SP frame with 16 thin nodes. Each node was equipped with

- One switch adapter
- One SSA adapter
- One 8-port adapter
- One Ethernet adapter for the user

This hardware configuration normally does not allow a cascading relationship between the nodes via the user Ethernet network. The Rotating relationship is the only supported configuration for this hardware layout.

14.2 Our Workaround

The workaround we chose was originally designed by Simon Marchese at IBM UK. We changed the scripts so that they would work in HACMP for AIX V4.2.1. The following sections give you more information about the advantages and disadvantages of this solution.

14.2.1 Technical Description

IP Address Takeover (IPAT) is one of the major functions of HACMP for AIX. The technique used to support IPAT in HACMP is IP address swapping. When a service address needs to be taken by a node, either through node failure or maintenance, that address is swapped onto a spare network adapter, known as a *standby interface*, whose own address is first discarded. By using an alternative technique known as IP address aliasing, the requirements for network adapters can be reduced to one adapter per node by avoiding discarding addresses on takeover.

14.2.2 Advantages

This solution has the following advantages:

- The enhancement reduces the minimum number of network adapters required to support a cascading resource group to one per node.
- In configurations of more than two nodes, multiple standby adapters are not required if multiple takeovers need to be supported. Takeover can be caused both by node failure and graceful shutdown for maintenance.
- This enhancement will especially help large HACMP configurations, such as RS/6000 SP. RS/6000 SP provides economies of scale in administration and maintenance and is a particularly attractive offering for multi-node configurations. However, they are sometimes limited in the available number of adapter slots available.

14.2.3 Disadvantages

This solution has the following disadvantages:

- By utilizing IP address aliasing, we are relying on a single network adapter to support multiple IP addresses. Currently, a single network adapter can only support one hardware address (or MAC address, as it is also known). That means that IP address aliasing cannot support Hardware Address Takeover (HWAT).
- Because we now have only one network adapter configured on a given network, HACMP cannot determine whether a heartbeat failure on this network is due to failure of the adapter or the network itself. This is only a problem in clusters where there are only two nodes active. This may be because there are only two nodes in the cluster or because the other nodes are not currently active, either through failure or graceful shutdown for maintenance.
- The system is more likely to suffer a performance bottleneck at the network adapter because we are now supporting multiple IP addresses on a single network adapter.

14.2.4 Installation

Installation should be performed while the cluster is down. The enhancement consists of enhanced versions of the `acquire_takeover_addr` and `release_takeover_addr` events and two new utilities, `cl_alias_IP_address` and `cl_unalias_IP_address`. These may be copied over the existing versions. However, any subsequent fix to HACMP may replace the new events, so a check should be made before reintegrating an upgraded node into an active cluster. HACMP usually renames any replaced events as `<eventname>.ORIG` when installing a fix, so the enhanced events will not be lost, but their function will be lost and they may not be compatible with other functions contained in the fix.

An alternative method is to install the new events elsewhere, for example, into a new directory such as `/usr/local/cluster/events`. The HACMP events can then be changed through SMIT to point to the new event scripts. As mentioned in the previous paragraph, any subsequent fix to HACMP may overwrite the path name of the new events, so a check should be made before reintegrating an upgraded node into an active cluster.

For simplicity, the enhancement is packaged as a tar format archive, stored with absolute path names from `/usr/local/cluster/events` down. This one is stored on the diskette as `ip_alias_fullp.tar`. For all who do not like to have it in this path, we stored the same package with relative path names from `./` down. This one is stored on the diskette as `ip_alias_relp.tar`.

14.2.5 Configuration

Configuration of the enhancement is performed through SMIT, using the standard HACMP SMIT panels.

The enhancement takes effect when no standby adapters are configured on a given node on the HACMP network concerned. Therefore, only service and boot adapter labels should be configured to enable the enhancement.

14.2.6 System Requirements

The enhancement has been partially tested with HACMP for AIX V4.2.1 and HACMP ES, all on AIX V4.2.1. However, the usual HACMP implementation testing should be performed.

Note: We did not have enough time to do all the necessary tests. Therefore, it may contain some bugs, but for all the tests we did it worked well.

In order to test whether your cluster configuration will support IP address aliasing, the enhancement may be simulated by using the `ifconfig` and `netstat` commands. Perform the test on a cluster node that has been taken out of the cluster for maintenance, or on a non-clustered RS/6000.

Make sure that the correct address is removed from the `netstat` output by the `ifconfig` commands. If the results are as expected, the enhancement is likely to work, as that is pretty much what it does in the code. If the results are not as expected, there is a problem.

For an example of what you should see when testing your system, refer to the following:

```

# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 100 0 100 0 0
lo0 1536 127 localhost 100 0 100 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.1 0 0 0 0 0
# ifconfig en0 detach
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 100 0 100 0 0
lo0 1536 127 localhost 100 0 100 0 0
# ifconfig en0 192.9.200.1 up
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 101 0 101 0 0
lo0 1536 127 localhost 101 0 101 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.1 0 0 0 0 0
# ifconfig en0 alias 192.9.200.2 up
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 101 0 101 0 0
lo0 1536 127 localhost 101 0 101 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.1 0 0 0 0 0
en0 1500 192.9.200 192.9.200.2 0 0 0 0 0
# ifconfig en0 delete 192.9.200.2
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 101 0 101 0 0
lo0 1536 127 localhost 101 0 101 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.1 0 0 0 0 0
# ifconfig en0 alias 192.9.200.2 up
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 101 0 101 0 0
lo0 1536 127 localhost 101 0 101 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.1 0 0 0 0 0
en0 1500 192.9.200 192.9.200.2 0 0 0 0 0
# ifconfig en0 delete 192.9.200.1
# netstat -i
Name Mtu Network Address Ipkts Ierrs Opkts Oerrs Coll
lo0 1536 <Link> 101 0 101 0 0
lo0 1536 127 localhost 101 0 101 0 0
en0 1500 <Link> 0 0 0 0 0
en0 1500 192.9.200 192.9.200.2 0 0 0 0 0
#

```

14.2.7 Support

The enhancement is supplied “as is.” No commitment to support the enhancement is implied. Customers should not contact their local support organization for help with this enhancement unless a prior agreement has been made.

14.2.8 Using the Modified Event Scripts

As previously mentioned, there are several possible ways to install the modified scripts. You can overwrite the current scripts or copy them to a local directory. To preserve any changes even if HACMP is going to be updated, we recommend that you use local directories for modifications like this one.

By using the `ip_alias_fullp.tar` file, you may do the installation as follows:

Copy the `ip_alias_fullp.tar` file from the disk to `/tmp/ip_alias_fullp.tar`, as follows:

```
# cd /tmp
# tar -xf /dev/rfd0 ip_alias_fullp.tar
```

The following command will automatically place the files into the `/usr/local/cluster/events` and `/usr/local/cluster/events/utls` directories:

```
tar -xf /tmp/ip_alias_fullp.tar
```

Note: The `/usr/local/cluster/events` directory will be automatically created if it does not exist.

Now you need to modify the cluster events for `acquire_takeover_addr` and `release_takeover_addr`, either by using the command:

```
/usr/sbin/cluster/utilities/clchevent
```

or via SMIT:

```
# smit hacmp
=> Cluster Config
=> Cluster Resources
=> Change/Show Cluster Events
```

Modify the path for the event command so that `/usr/local/cluster/events` is used.

What if you do not want to have the files in the `/usr/local/cluster/events` directory? Perhaps you would like to overwrite the original ones. To do this, you have to modify the `acquire_takeover_addr` and `release_takeover_addr` files to invoke the scripts `cl_alias_IP_address` and `cl_unalias_IP_address` stored in the original `utls` directory, as follows:

Edit `acquire_takeover_addr`.

Replace the line (299),

```
/usr/local/cluster/events/utls/cl_alias_IP_address $INTERFACE $addr...
```

with

```
/usr/sbin/cluster/events/utls/cl_alias_IP_address $INTERFACE $addr..
```

Edit `release_takeover_addr`.

Replace the line (189),

```
/usr/local/cluster/events/utls/cl_unalias_IP_address $STBY_INTERFACE $addr...
```

with

```
/usr/sbin/cluster/events/utls/cl_unalias_IP_address $STBY_INTERFACE $addr...
```

Part 3. Appendices

This part contains our appendices.

Appendix A. AIX Scripts	151
A.1 Modified HACMP Scripts	151
A.1.1 The HACMP Script acquire_takeover_addr	151
A.1.2 The HACMP Script release_takeover_addr	156
A.1.3 The HACMP Script cl_alias_IP_address	159
A.1.4 The HACMP Script cl_unalias_IP_address	161
A.2 Scripts for User-Defined Events	164
A.2.1 The Start Script for the HACMP Application	164
A.2.2 The Stop Script for the HACMP Application	164
A.2.3 The rules.hacmprd File	165
A.2.4 The rwho.rp File	168
A.2.5 The rwho_msg_local File	168
A.2.6 The rwho_msg_remote File	168
A.2.7 The rwho_msg_complete File	168
A.2.8 The rwho_restart File	169
Appendix B. Special Notices	171
Appendix C. Related Publications	173
C.1 International Technical Support Organization Publications	173
C.2 Redbooks on CD-ROMs	173
C.3 Other Publications	173

Appendix A. AIX Scripts

This appendix contains the scripts we used in our HACMP ES installation.

A.1 Modified HACMP Scripts

This section contains the modified scripts for HACMP for AIX V4.2.1 and HACMP ES we used for test installations.

We had only one free Ethernet adapter available on each node we used. To be able to test the cascading relationship between these nodes, we had to use these modified scripts.

A.1.1 The HACMP Script `acquire_takeover_addr`

```
#!/bin/ksh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 41hacmp421 src/41hacmp/usr/sbin/cluster/events/acquire_takeover_addr.sh 1.6.1.2
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1997
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# @(#)66 1.6.1.2 src/41hacmp/usr/sbin/cluster/events/acquire_takeover_addr.sh, hacmp.event,
# 41hacmp421, 9715A_41ha421 4/4/97 16:36: 01
# $Id: acquire_takeover_addr.sh,v 7.3 1996/07/08 02:56:54 bobbyg Exp $
# $Id: acquire_takeover_addr.sh,v 7.3.1 1997/05/07 15:15:30 bobbyg Exp $
#
# COMPONENT_NAME: EVENTS
#
# FUNCTIONS: name_to_addr
#
# ORIGINS: 27
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1994
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#####
#
# Name:          acquire_takeover_addr
#
# Description:   This script is called when a remote node
#               leaves the cluster.
#               The script first checks to see if a
#               configured standby address exists and
#               is considered 'up' by clstrmgr, then does
#               a standby_address -> takeover_address swap.
#
#
#               For an SP-switch, the script aliases the
#               takeover address on the same adapter as the
#               local service address.
#
# Called by:     node_down_remote, node_up_local
#
# Calls to:      cl_swap_IP_address
#               cl_alias_IP_address
#
#
```

```

# Arguments: takeover_address... #
# Returns: 0 success #
# 1 failure #
# 2 bad argument #
# #####
PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster/events/utlis:/usr/
sbin/cluster/utilities
export PATH

PROGNAME=$0
TELINIT=false
DELAY=5
STATUS=0

if [ $# -eq 0 ]
then
    cl_echo 1029 "Usage: $PROGNAME takeover_address...\n" $PROGNAME
    exit 2
fi

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

# Routine to turn NIS on.
turn_on_DNS_NIS() {
if [ "$NAME_SERVER" = "true" ]
then
    /usr/sbin/cluster/events/utlis/cl_nm_nis_on
    if [ $? -ne 0 ]
    then
        STATUS=$?
    fi
fi
}

#####
# Name: addback_route
#
# When two or more standbys are on the same subnet, only one of the
# standbys is in the routing table as the route. If this standby is
# used to takeover the remote address, the route also gets destroyed
# in routing table. This routine is used to restore the route for
# the remaining standbys on the subnet.
#
# Arguments: standby_IP_address
#
# Returns: None
#
#####
addback_route () {

NETWORK=/usr/sbin/cluster/utilities/cllsif -cSn $1 ] cut -d':' -f3 ] uniq

standby_list=/usr/sbin/cluster/utilities/cllsif -cS ] grep "standby" ] cut -d':' -f7

for standby in $standby_list
do
    #
    # Make sure the standby is not the same one
    #
    if [ "$standby" = "$1" ]
    then
        continue
    fi

    #
    # Make sure two standbys are on the same network
    #
    network=/usr/sbin/cluster/utilities/cllsif -cSn $standby ] cut -d':' -f3 ] uniq
    if [ "$network" != "$NETWORK" ]

```

```

then
    continue
fi

#
# Make sure the standby is defined on local node
#
/usr/sbin/cluster/utilities/clgetif -n $standby >/dev/null 2>&1
if . $? != 0 ]
then
    continue
fi

NETMASK=/usr/sbin/cluster/utilities/clgetif -n $standby
INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $standby

#
# Make sure the standby is up on local node
#
addr=i"$standby"_"$LOCALNODENAME"
addr=/bin/echo $addr ] /bin/sed -e s/././x/g
VAR=\ "$addr"
set +u
VAL="eval echo $VAR"
set -u

if . "$VAL" != "UP" ]
then
    continue
fi

#
# Do ifconfig to add the route in. Will be a no-op if already in
#
ifconfig $INTERFACE $standby netmask $NETMASK up

done
}

#
# This routine maps a label_address to the 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 /usr/sbin/cluster/utilities/cllsif -cSn $1 ] cut -d: -f7 ] uniq
    exit $?
}

#####
#
# main routine
#
#####
# Turn NIS off.
if . "$NAME_SERVER" = "true" ]
then
    /usr/sbin/cluster/events/utills/cl_nm_nis_off
    if . $? -ne 0 ]
    then
        exit 1
    fi
fi

set -u

for addr in $*
do

    #
    # Determine if address is already configured. If not, try to
    # acquire it.
    #
    clgetif -a $addr 2>/dev/null
    if . $? -ne 0 ]

```

```

then

#
# Get dot address of takeover_address, network and configured standby
# addresses for later use.
#
STATUS=1
addr_dot_addr=name_to_addr $addr
NETWORK=/usr/sbin/cluster/utilities/cllsif -cSn $addr_dot_addr ] /bin/cut -d':' -f3 ] uniq

# Get the service address associated with this network
SERVICE_ADDR=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
] grep $NETWORK ] grep service ] /bin/cut -d':' -f7 ] uniq

# Determine the interface associated with the service address
INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $SERVICE_ADDR

if . -z "$INTERFACE" ]
then
# Get the boot address associated with this network
BOOT_ADDR=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
] grep $NETWORK ] grep boot ] /bin/cut -d':' -f7 ] uniq

# Determine the interface associated with the boot address
INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $BOOT_ADDR
fi

# Unable to determine local boot/service interface. We should never be here
if . -z "$INTERFACE" ]
then
MSG=dspmsg scripts.cat 342 "Unable to determine local boot/service interface.\n"
STATUS=1
fi

# Determine if this is an SP switch interface. If so,
# execute appropriate script.
if . $INTERFACE = "css0" ]
then

# Determine the netmask
for interface in $SERVICE_ADDR
do
SP_SWITCH_NETMASK=/usr/sbin/cluster/utilities/clgetif -n $interface
if . -n "$SP_SWITCH_NETMASK" ]
then
break
fi
done

if . -n "$SP_SWITCH_NETMASK" ]
then
/usr/sbin/cluster/events/utl/c1_swap_HPS_IP_address css0 $addr $SP_SWITCH_NETMASK
STATUS=$?
else
STATUS=1
fi

save="placeholderjunk"

else

STDBYS=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME ] grep :standby: \
] cut -d':' -f1,3 ] grep -w $NETWORK ] /bin/cut -d':' -f1

#####
# Start of modified part (aliasing on other networks)
#####
print "STDBYS='$STDBYS'"
if . -z "$STDBYS" ]
then
# run aliasing
save="placeholderjunk"
SERVS=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
] grep :service: ] grep :$NETWORK: ] /bin/cut -d':' -f1
for service in $SERVS

```



```

do
# Check if netmon thinks service/boot is up or down
service_dot_addr=name_to_addr $service
addr=i"$service_dot_addr" "$LOCALNODENAME"
addr=/bin/echo $addr ] /bin/sed -e s/./]/x/g
VAR=\$"$addr"
set +u
SERVICE_STATE="eval echo $VAR"
set -u
boot=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME ] grep \
:boot: ] grep :$NETWORK: ] /bin/cut -d:' -f1
boot_dot_addr=name_to_addr $boot
addr=i"$boot_dot_addr" "$LOCALNODENAME"
addr=/bin/echo $addr ] /bin/sed -e s/./]/x/g
VAR=\$"$addr"
set +u
BOOT_STATE="eval echo $VAR"
set -u

if . "$SERVICE_STATE" = "UP" -o "$BOOT_STATE" = "UP" ]
then
INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $service_dot_addr
NETMASK=/usr/sbin/cluster/utilities/clgetif -n $service_dot_addr
/usr/local/cluster/events/utls/cl_alias_IP_address $INTERFACE
$addr_dot_addr $NETMASK
STATUS=$?
fi
if . $STATUS -eq 0 ]
then
break
fi
done # for service in $SERVS
else # . -n $STDBYS ]
# as normal
#####
# End of modified part (aliasing on other networks)
#####

for standby in $STDBYS
do
#
# Get dot address of standby_label and its associated interface
# for later use.
#
standby_dot_addr=name_to_addr $standby
INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $standby_dot_addr

if . -n "$INTERFACE" ]
then
#
# If standby address in the local node is 'up',
# swap the standby_address to the takeover_address
# (cl_swap_IP_address).
#
NETMASK=/usr/sbin/cluster/utilities/clgetif -n $standby_dot_addr
save=i"$standby_dot_addr" "$LOCALNODENAME"
save=/bin/echo $save ] /bin/sed -e s/./]/x/g
VAR=\$"$save"
set +u
VAL="eval echo $VAR"
set -u
if . -z "$VAL" -o "$VAL" = "UP" ]
then
/usr/sbin/cluster/events/utls/cl_swap_IP_address \
$INTERFACE $addr_dot_addr $NETMASK
STATUS=$?
addback_route $standby_dot_addr
break
fi
fi
done
#####
# fi from modified part
fi # if . -n "$STDBYS" ]
#####

```

```

fi

if . $STATUS -ne 0 ]
then
    MSG=dspmsg scripts.cat 340 "IP Address Takeover of $addr_dot_addr failed.\n"
    $addr_dot_addr
    /bin/echo $MSG >/dev/console

    # Turn Name Service back on
    turn_on_DNS_NIS
    exit 1
else
    #
    # Mark this standby adapter 'DOWN', so it will not be
    # used again in the next iteration
    #
    export $save=DOWN
    TELINIT=true
fi
fi
done

# Turn on Name Service
turn_on_DNS_NIS

#
# Start tcp/ip servers and network daemons via 'telinit a'.
#
if . "$TELINIT" = "true" ]
then
    #
    # Set hostname to first public service address
    #
    # FIRST_SERVS=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME \
    #                ] grep :service: \
    #                ] grep :public: ] cut -d':' -f1
    # FIRST_SERV=echo $FIRST_SERVS ]cut -d' ' -f1
    # if . -n "$FIRST_SERV" ]
    # then
    #     hostname $FIRST_SERV
    # fi

if . ! -f /usr/sbin/cluster/.telinit ]
then
    #
    # In /etc/inittab, there is an entry to touch /usr/sbin/cluster/.telinit
    # after tcp/ip is functionally up.
    #

telinit a

while . ! -f /usr/sbin/cluster/.telinit ]
do
    sleep $DELAY
done
fi
fi

exit $STATUS

```

A.1.2 The HACMP Script `release_takeover_addr`

```

#!/bin/sh
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 41hacmp421 src/41hacmp/usr/sbin/cluster/events/release_takeover_addr.sh 1.6
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1997
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or

```

```

# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# @(#)91 1.6 src/41hacmp/usr/sbin/cluster/events/release_takeover_addr.sh, hacmp.events,
41hacmp421, 9715A_41ha421 4/1/97 17:51:24
# $Id: release_takeover_addr.sh,v 7.3 1996/07/08 02:57:00 bobbyg Exp $
#
# COMPONENT_NAME: EVENTS
#
# FUNCTIONS: none
#
# ORIGINS: 27
#
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1994
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
#####
# Name: release_takeover_addr #
# Description: This script is called if the local node has #
# the remote node's service address on its #
# standby adapter, and either the remote node #
# re-joins the cluster or the local node #
# leaves the cluster gracefully. #
# Called by: node_down_local, node_up_remote #
# Calls to: cl_swap_IP_address #
# Arguments: takeover-address... #
# Returns: 0 success #
# 1 failure #
# 2 bad argument #
#####

PATH=$PATH:/usr/sbin/cluster:/usr/sbin/cluster/events:/usr/sbin/cluster/
events/utills:/usr/sbin/cluster/utilities
export PATH

PROGRAM=$0
STATUS=0

if [ $# -eq 0 '
then
    cl_echo 1029 "Usage: $PROGRAM takeover-address..\n" $PROGRAM
    exit 2
fi

if [ "$VERBOSE_LOGGING" = "high" '
then
    set -x
fi

# Routine to turn NIS on.
turn_on_DNS_NIS () {
if [ "$NAME_SERVER" = "true" '
then
    /usr/sbin/cluster/events/utills/cl_nm_nis_on
    if [ $? -ne 0 '
    then
        STATUS=$?
    fi
fi
}

#
# This routine maps a label_address to the 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 /usr/sbin/cluster/utilities/cllsif -cSn $1 ] cut -d: -f7 ] uniq
    exit $?
}

#####
#
# main routine
#
#####

# Turn NIS off.
if [ "$NAME_SERVER" = "true" '
then
    /usr/sbin/cluster/events/utlils/cl_nm_nis_off
    if [ $? -ne 0 '
    then
        exit 1
    fi
fi

set -u

for addr in $*
do
    #
    # Determine if address is already unconfigured. If not, try to
    # release it.
    #
    clgetif -a $addr 2>/dev/null
    if [ $? -eq 0 '
    then

        STBY_IP_ADDR=""
        addr_dot_addr=name_to_addr $addr

        #
        # Get the standby interface to which the remote service address is mapped.
        #
        STBY_INTERFACE=/usr/sbin/cluster/utilities/clgetif -a $addr_dot_addr

        if [ "$STBY_INTERFACE" = "" '
        then
            cl_echo 318 "No service address $addr was taken by this node." $addr
            continue
        fi

        #
        # Get the netmask and network name for later use.
        #
        NETMASK=/usr/sbin/cluster/utilities/clgetif -n $addr_dot_addr
        NETWORK=/usr/sbin/cluster/utilities/cllsif -cSn $addr_dot_addr ] cut -d:' -f3 ] uniq

        #
        # Get this node's original standby address from the configuration.
        #
        STBYS=/usr/sbin/cluster/utilities/cllsif -cSi $LOCALNODENAME]grep :standby: \
            ] cut -d:' -f3,7 ] grep -w $NETWORK ] cut -d:' -f2

        for s in $STBYS
        do
            if [ "/usr/sbin/cluster/utilities/clgetif -a $s" = "" '
            then
                #
                # This standby is not configured, it is the missing standby.
                # Record it and exit loop.
                #
                STBY_IP_ADDR="$s"
                break
            fi
        done

        if [ -n "$STBY_IP_ADDR" '

```

```

then
#
# Reconfigure the standby.
#
/usr/sbin/cluster/events/utlils/cl_swap_IP_address \
    $STBY_INTERFACE $STBY_IP_ADDR $NETMASK
if [ $? != 0 '
then
    STATUS=1
fi
else
# Determine if the interface belongs to an SP switch. If so,
# call the SP Switch-related script.
if [ -n "$STBY_INTERFACE" -a $STBY_INTERFACE = "css0" '
then
    /usr/sbin/cluster/events/utlils/cl_swap_HPS_IP_address \
        css0 $addr $NETMASK delete
    if [ $? != 0 '
    then
        STATUS=1
    fi
fi
else
#####
# Start of modified part (aliasing on other networks)
#####

#
    cl_log 319 "No missing standby found for service address $addr." $addr
#
    STATUS=1
    /usr/local/cluster/events/utlils/cl_unalias_IP_address $STBY_INTERFACE $addr $NETMASK
    if [ $? != 0 '
    then
        cl_log 319 "No missing standby found for service address $addr." $addr
        STATUS=1
    fi
fi
#####
# End of modified part (aliasing on other networks)
#####

fi
fi
done

turn_on_DNS_NIS
exit $STATUS

```

A.1.3 The HACMP Script cl_alias_IP_address

```

#!/bin/sh
PROGRAMME="$0"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

DELETE_ROUTES=/usr/sbin/cluster/.delete_routes
ADD_ROUTES=/usr/sbin/cluster/.add_routes
ROUTE_ADD=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
}

```

```

#####
# Name: add_routes
#
#     Echos route add commands nessary to restore routing table after
#     adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#####
add_routes() {
    /bin/echo "#!/bin/sh -x"
    /bin/echo "PATH=$PATH"

    for interface in "$@"
    do
        netstat -rn | fgrep $interface | fgrep " UG " | \
            awk '{print "route add -net " $1" "$2}'

        netstat -rn | fgrep $interface | fgrep -v " UG " | \
            fgrep -v " U " | awk '{print "route add " $1" "$2}'

    done

    /bin/echo "exit 0"
    return 0
}

#####
# Name: delete_routes
#
#     Echos route delete commands nessary to clear routing table
#     before adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#####
delete_routes () {
    /bin/echo "#!/bin/sh -x"
    /bin/echo "PATH=$PATH"

    for interface in "$@"
    do
        netstat -rn | fgrep $interface | fgrep -v "H" | \
            awk '{print "route delete -net " $1" "$2}'

    done

    for interface in "$@"
    do
        netstat -rn | fgrep $interface | fgrep "H" | \
            awk '{print "route delete " $1" "$2}'

    done

    /bin/echo "exit 0"
    return 0
}

#####
#
# Main entry point
#
#####
c1_echo 33 "Starting execution of $0 with parameters $* " $0 "$*"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

set -u

# this form for single interface
if [ $# -eq 3 ]
then

```

```

IF=$1
ADDR=$2
NETMASK=$3

# Get routes bound to adapter and create script file
# to re-add routes later.
add_routes $IF ] tee $ADD_ROUTES
# add_routes $IF > $ADD_ROUTES
chmod +x $ADD_ROUTES

# Prevent 'no routes to dest' errors by adding default
# route to loopback. The packets will get dropped but
# TCP will endure
route add default 127.0.0.1 >/dev/null 2>&1
ROUTE_ADD=$?

# down old interfaces
cl_echo 60 "$PROGNAME: Configuring adapter $IF at IP address $ADDR" $PROGNAME $IF $ADDR
# ifconfig $IF down

# Must delete routes because of ifconfig down.
delete_routes $IF ] tee $DELETE_ROUTES
#delete_routes $IF > $DELETE_ROUTES
chmod +x $DELETE_ROUTES
$DELETE_ROUTES

#set the specified interface to specified address
ifconfig $IF alias $ADDR netmask $NETMASK up
if [ $? -ne 0 ]
then
    ifconfig $IF alias $ADDR netmask $NETMASK up
    if [ $? -ne 0 ]
    then
        cl_log 59 "$PROGNAME: Failed ifconfig \
$IF inet $ADDR netmask $NETMASK up." $PROGNAME $IF $ADDR $NETMASK
        exit 1
    fi
fi

# flush arp table
flush_arp

# Add back pre-existing routes
$ADD_ROUTES

# Delete default route only if we succeed before
if [ $ROUTE_ADD -eq 0 ]
then
    route delete default 127.0.0.1
fi

else
# else bad arg count
cl_echo 62 "usage: $PROGNAME interface address netmask" $PROGNAME
cl_echo 63 "    or $PROGNAME interface1 address1 interface2 address2 netmask" $PROGNAME
exit 2
fi

cl_echo 32 "Completed execution of $0 with parameters $*. Exit status = $?" $0 "$*" $?

exit 0

```

A.1.4 The HACMP Script `cl_unalias_IP_address`

```

#!/bin/sh -x
#
# Returns:    0 - success
#            1 - ifconfig failure
#            2 - bad number of arguments
#            3 - Hardware swap failure
#
# Environment: VERBOSE_LOGGING,PATH
#####
NEW_ADDRESSES=
PATH=$PATH:/usr/sbin/cluster/events/utlis

```

```

PROGRAMME="$0"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

DELETE_ROUTES=/usr/sbin/cluster/.delete_routes
ADD_ROUTES=/usr/sbin/cluster/.add_routes
ROUTE_ADD=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
}

#####
# Name: add_routes
#
#     Echos route add commands ncessary to restore routing table after
#     adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#####
add_routes() {
    /bin/echo "#!/bin/sh -x"
    /bin/echo "PATH=$PATH"

    for interface in "$@"
    do
        netstat -rn ] fgrep $interface ] fgrep " UG " ] \
            awk '{print "route add -net " $1" "$2}'

        netstat -rn ] fgrep $interface ] fgrep -v " UG " ] \
            fgrep -v " U " ] awk '{print "route add " $1" "$2}'

    done

    /bin/echo "exit 0"
    return 0
}

#####
# Name: delete_routes
#
#     Echos route delete commands ncessary to clear routing table
#     before adapter reconfiguration.
#
# Arguments: list of interfaces
#
# Returns: None
#####
delete_routes () {
    /bin/echo "#!/bin/sh -x"
    /bin/echo "PATH=$PATH"

    for interface in "$@"
    do
        netstat -rn ] fgrep $interface ] fgrep -v "H" ] \
            awk '{print "route delete -net " $1" "$2}'

    done

    for interface in "$@"

```



```

do
    netstat -rn ] fgrep $interface ] fgrep "H" ] \
    awk '{print "route delete " $1" "$2}'
done

/bin/echo "exit 0"
return 0
}
#
# This routine maps a label_address to the 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 /usr/sbin/cluster/utilities/cllsif -cSn $1 ] cut -d: -f7 ] uniq
    exit $?
}
#####
#
# Main entry point
#
#####
cl_echo 33 "Starting execution of $0 with parameters $*" $0 "$*"

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

set -u

# this form for single interface
if [ $# -eq 3 ]
then
    IF=$1
    NAME=$2
    NETMASK=$3

    ADDR=name_to_addr $NAME

    # Get routes bound to adapter and create script file
    # to re-add routes later.
    add_routes $IF > $ADD_ROUTES
    chmod +x $ADD_ROUTES

    # Prevent 'no routes to dest' errors by adding default
    # route to loopback. The packets will get dropped but
    # TCP will endure
    route add default 127.0.0.1 >/dev/null 2>&1
    ROUTE_ADD=$?

    # down old interfaces
    cl_echo 60 "$PROGNAME: Configuring adapter $IF at IP address $ADDR" $PROGNAME $IF $ADDR
    # ifconfig $IF down

    # Must delete routes because of ifconfig down.
    delete_routes $IF > $DELETE_ROUTES
    chmod +x $DELETE_ROUTES
    $DELETE_ROUTES

    ifconfig $IF $ADDR delete

    # flush arp table
    flush_arp

    # Add back pre-existing routes
    $ADD_ROUTES

    # Replace automated route - but use alias in case > 1 addresses
    ADDRESS=ifconfig $IF ] awk 'FNR==2 {print $2}'
    NETMASK=/usr/sbin/cluster/utilities/clgetif -n $ADDRESS

    ifconfig $IF alias $ADDRESS netmask $NETMASK up

```

```

        # Delete default route only if we succeed before
        if [ $ROUTE_ADD -eq 0 ]
        then
            route delete default 127.0.0.1
        fi

    else
        # else bad arg count
        cl_echo 62 "usage: $PROGNAME interface address netmask" $PROGNAME
        exit 2
    fi

    cl_echo 32 "Completed execution of $0 with parameters $*. Exit status = $" $0 "$*" $?

    exit 0

```

A.2 Scripts for User-Defined Events

This section contains the scripts we used for testing user-defined events. It also contains the start and stop scripts for the HACMP application we used.

A.2.1 The Start Script for the HACMP Application

```

#!/bin/ksh
#
#

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

# creating test file for a User Event
if [ ! -d /usr/sbin/cluster/tmp ]
then
    mkdir -p /usr/sbin/cluster/tmp
fi

touch /usr/sbin/cluster/tmp/rwhod.on

#
# Starting the rwhod daemon
#

print "$(date) Starting the \"rwhod\" daemon"
startsrc -s rwhod

exit 0

```

A.2.2 The Stop Script for the HACMP Application

```

#!/bin/ksh
#
#

if [ "$VERBOSE_LOGGING" = "high" ]
then
    set -x
fi

rm /usr/sbin/cluster/tmp/rwhod.on > /dev/null

print "$(date) Stopping the \"rwhod\" daemon"
stopsrc -s rwhod

exit 0

```

A.2.3 The rules.hacmprd File

```
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 41hape421 src/41hape/usr/sbin/cluster/events/rules.hacmprd 1.1
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1996,1997
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# #####
# "@(#)29 1.1 src/41hape/usr/sbin/cluster/events/rules.hacmprd, hacmp.pe,
# 41hape421 11/7/96 13:41
#
# COMPONENT_NAME: EVENTS
#
# FUNCTIONS: none
#
# ORIGINS: 27
#
#
# (C) COPYRIGHT International Business Machines Corp. 1990,1994
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# #####
# This file contains the HACMP/PE recovery program to event mapping
#
# format: (1) name
#         (2) state (qualifier)
#         (3) recovery program path
#         (4) recovery type (Reserved for future use)
#         (5) recovery level (Reserved for future use)
#         (6) resource variable name (Used for Event Manager events)
#         (7) instance vector (Used for Event Manager events)
#         (8) predicate (Used for Event Manager events)
#         (9) rearm predicate (Used for Event Manager events)
#
##### Beginning of Event Definition Node Up #####
#
TE_JOIN_NODE
0
/usr/sbin/cluster/events/node_up.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition Node Up #####
#
##### Beginning of Event Definition Node Down #####
#
TE_FAIL_NODE
0
/usr/sbin/cluster/events/node_down.rp
2
0
# 6) Resource variable only used for event manager events
```

```

# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events
# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Node Down      #####
#
#
##### Beginning of Event Definition      Network Up #####
#
TE_JOIN_NETWORK
0
/usr/sbin/cluster/events/network_up.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events
# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Network Up      #####
#
#
##### Beginning of Event Definition      Network Down #####
#
TE_FAIL_NETWORK
0
/usr/sbin/cluster/events/network_down.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events
# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Network Down      #####
#
#
##### Beginning of Event Definition      Swap Adapter #####
#
TE_SWAP_ADAPTER
0
/usr/sbin/cluster/events/swap_adapter.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events
# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Swap Adapter      #####
#
#
##### Beginning of Event Definition      Join Standby #####
#
TE_JOIN_STANDBY
0
/usr/sbin/cluster/events/join_standby.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events
# 8) Predicate, only used for event manager events

```

```

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Join Standby      #####
#
#
##### Beginning of Event Definition      Fail Standby      #####
#
TE_FAIL_STANDBY
0
/usr/sbin/cluster/events/fail_standby.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      Fail Standby      #####
#
#
##### Beginning of Event Definition      DARE Topology      #####
#
TE_DARE_TOPOLOGY
0
/usr/sbin/cluster/events/reconfig_topology.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      DARE Topology      #####
#
#
##### Beginning of Event Definition      DARE Resource      #####
#
TE_DARE_RESOURCE
0
/usr/sbin/cluster/events/reconfig_resource.rp
2
0
# 6) Resource variable only used for event manager events

# 7) Instance vector, only used for event manager events

# 8) Predicate, only used for event manager events

# 9) Rearm predicate, only used for event manager events

##### End of Event Definition      DARE Resource      #####
#
# Definition of
# Monitor death of rwhod daemon on node 5
#
UE_PGMWHO_DOWN
0
/usr/sbin/cluster/local/rwho.rp
2
0
IBM.PSSP.Prog.xpcount
NodeNum=*;ProgName=rwhod;UserName=root
X@0==0&&X@1!=0
X@0>0
#
# End of definition.

```

A.2.4 The rwho.rp File

```
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# 41hape421 src/41hape/usr/sbin/cluster/events/node_up.rp 1.2
#
# Licensed Materials - Property of IBM
#
# (C) COPYRIGHT International Business Machines Corp. 1996,1997
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
# #####
# COMPONENT_NAME: EVENTS
#
# FUNCTIONS: none
#
# ORIGINS: 27
#
# #####
# This file contains the HACMP/PE recovery program for node_up events
#
# format:
# relationship    command to run    expected status NULL
#
other "/usr/sbin/cluster/local/rwho_msg_remote" 0 NULL
event "/usr/sbin/cluster/local/rwho_msg_local" 0 NULL
#
barrier
#
event "/usr/sbin/cluster/local/rwho_restart" 0 NULL
#
barrier
#
all "/usr/sbin/cluster/local/rwho_msg_complete" X NULL
#
```

A.2.5 The rwho_msg_local File

```
#!/bin/ksh
#
print "$TIMESTAMP $EVNAME detected a failure of \"rwhod\" program"
print "The failure occurred on the local SP-Node $EVLOCATION"

exit 0
#
```

A.2.6 The rwho_msg_remote File

```
#!/bin/ksh
#
print "$TIMESTAMP $EVNAME detected a failure of \"rwhod\" program"
print "The failure occurred on the remote SP-Node $EVLOCATION"

exit 0
#
```

A.2.7 The rwho_msg_complete File

```
#!/bin/ksh
#
print "$(date): recovery from \"rwhod\" daemon failure completed on Node $EVLOCAT
exit 0
```

A.2.8 The rwho_restart File

```
#!/bin/ksh
#
#set -x
#
if [ -f /usr/sbin/cluster/tmp/rwhod.on ]
then
    print "$(date) Restarting the \"rwhod\" daemon"
    startsrc -s rwhod
else
    print "$(date) Recovery terminated because of HACMP takeover"
fi

exit 0
```

Appendix B. Special Notices

This publication is intended to help IBM customers, Business Partners, IBM System Engineers, and other RS/6000 SP specialists who are involved with HACMP Enhanced Scalability Version 4 Release 2.1 projects, including the education of RS/6000 SP professionals responsible for installing, configuring, and administering PSSP Version 2 Release 3 with HACMP ES Version 4 Release 2.1. The information in this publication is not intended as the specification of any programming interfaces that are provided by Parallel System Support Programs and HACMP ES. See the PUBLICATIONS section of the IBM Programming Announcement for PSSP Version 2 Release 3 and HACMP ES Version 4 Release 2.1. 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.

Licenseses of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

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.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

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

ADSTAR	AIX
AS/400	BookManager
Current	DataJoiner
IBM	NetView
POWERparallel	RS/6000
Scalable POWERparallel Systems	SP
System/390	400

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Appendix C. Related Publications

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

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 175.

- *RS/6000 SP High Availability Infrastructure*, SG24-4838
- *Implementing High Availability on RISC/6000 SP*, SG24-4742
- *RS/6000 SP System Management: Easy, Lean and Mean*, GG24-2563
- *RS/6000 SP: Problem Determination Guide*, SG24-4778
- *RS/6000 Scalable POWERparallel Systems PSSP Version 2 Technical Presentation*, SG24-4542

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

C.3 Other Publications

This publication is also relevant as an information source:

- *IBM Parallel System Support Programs for AIX Administration Guide*, GC23-3897

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 <http://www.redbooks.ibm.com>.

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 BookManager BOOKs of redbooks, type the following command:

```
TOOLCAT REDBOOKS
```

To get lists of redbooks, type one of the following commands:

```
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, type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
```

For a list of product area specialists in the ITSO: type the following command:

```
TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
```

- **Redbooks Web Site on the World Wide Web**

<http://w3.itso.ibm.com/redbooks>

- **IBM Direct Publications Catalog on the World Wide Web**

<http://www.elink.ibm.link.ibm.com/pb1/pb1>

IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS 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.ibm.link.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.

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.htm>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

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** — send orders to:

In United States:
In Canada:
Outside North America:

IBMMAIL
usib6fpl at ibmmail
caibmbkz at ibmmail
dkibmbsh at ibmmail

Internet
usib6fpl@ibmmail.com
lmannix@vnet.ibm.com
bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)
Canada (toll free)

1-800-879-2755
1-800-IBM-4YOU

Outside North America
(+45) 4810-1320 - Danish
(+45) 4810-1420 - Dutch
(+45) 4810-1540 - English
(+45) 4810-1670 - Finnish
(+45) 4810-1220 - French

(long distance charges apply)
(+45) 4810-1020 - German
(+45) 4810-1620 - Italian
(+45) 4810-1270 - Norwegian
(+45) 4810-1120 - Spanish
(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications
Publications Customer Support
P.O. Box 29570
Raleigh, NC 27626-0570
USA

IBM Publications
144-4th Avenue, S.W.
Calgary, Alberta T2P 3N5
Canada

IBM Direct Services
Sortemosevej 21
DK-3450 Allerød
Denmark

- **Fax** — send orders to:

United States (toll free)
Canada
Outside North America

1-800-445-9269
1-403-267-4455
(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States) or (+1)001-408-256-5422 (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 Web Site <http://www.redbooks.ibm.com>
IBM Direct Publications Catalog <http://www.elink.ibm.link.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.ibm.link.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (<http://www.redbooks.ibm.com/redpieces.htm>). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

Glossary

A

Adapter Membership State.

This is an internal n-phase protocol to distribute adapter group notifications since they are not provided in a consistent manner from adapter membership or the Group Services/ES shadow groups in the presence of disjoint networks.

Address Resolution Protocol.

The Internet communication protocol used to dynamically map Internet addresses to hardware (physical) addresses on local area networks (LAN). Limited to networks that support hardware broadcast.

Alias.

In the AIX operating system, an alternative name used for a network name, or other network entity. Synonymous with nickname.

Aliasing.

- In the AIX terminology. The `ifconfig ... alias` command used for IP address aliasing on the network interfaces. Also see Alias
- In the HACMP terminology. The `ifconfig ... alias` command used for IP address aliasing on the High Performance Switch network on the SP machine. This permits IP address takeover.

Application Server.

An application that runs on a cluster node. When queried by client applications, the application server may access a database on the shared external disk and then respond to client requests. Application servers are cluster resources guaranteed to be highly available.

ARP.

See Address Resolution Protocol

B

Barrier.

Barrier is a two-phase protocol used to implement barrier commands in the recovery programs. Once a node encounters a barrier command in the recovery program, it initiates this protocol, causing all nodes to go into the barrier state. As each node encounters the barrier command in the recovery program, it votes to “approve” the protocol. When all nodes have encountered the barrier command in the recovery program and voted to “approve” the protocol, Group Services/ES notifies all nodes that the protocol has completed.

Boot Address.

Address for a node to use before HACMP assigns a service address. If you want to use IP address takeover (with or without hardware address swapping) in an HACMP cluster, you must define a boot address (associated with a service adapter) so when a failed node comes back up, it can use the boot address until the process of node reintegration reassigns IP addresses.

Broadcast.

A packet delivery system where a copy of a given packet can be sent to all hosts attached to the network.

C

C-SPOC (Cluster Single Point of Control).

C-SPOC enables the user to perform certain common administrative operations across the cluster from a single SMIT session. These operations are: starting and stopping HACMP; adding, changing, and deleting users and groups; and operating on shared volume groups. For the last set of operations, the C-SPOC facility removes the need to manually synchronize the change across the cluster.

Cascading Resources.

Resources that may be taken over by more than one node. A takeover priority is assigned to each configured cluster resource group on a per-node basis. In the event of a takeover, the node with the highest priority acquires the resource group. If that node is unavailable, the node with the next highest priority acquires the resource group, and so on.

Cascading Takeover.

HACMP functionality that allows a particular resource or set of resources to be taken over by more than one node in a cluster.

Cbarrier.

The cbarrier protocol is intended to synchronize all nodes at the end of an HACMP ES event. It is an internal two-phase protocol implemented in the recovery programs.

CEL.

See Command Execution Language

CIconvert.

A utility that converts the configuration of an earlier cluster version of HACMP to the current version. The utility creates new data structures and objects,

redefines field names and values within data structures, and ensures data integrity between versions of the HACMP software.

Client.

Machine connected to the cluster network so it can access data or services maintained on the cluster.

Cluster.

Loosely coupled collection of independent systems (nodes) organized into a network for the purpose of sharing resources and communicating with each other. HACMP defines relationships among cooperating systems where peer cluster nodes provide the service offered by a cluster node should that node be unable to do so.

Cluster Event.

Represents a change in a cluster's composition that the Cluster Manager recognizes and can respond to. Major cluster events include:

- node_down
- node_up
- network_down
- network_up
- swap_adapter

Cluster Manager or HACMP ES Cluster Manager.

Cluster Manager is the component of HACMP for AIX V4.2.1 that monitors the state of the nodes, interfaces, and networks comprising a cluster. It also provides highly available access to these resources and to critical disk data and software resources running on the cluster.

HACMP ES Cluster Manager is the same for HACMP ES as Cluster Manager for HACMP for AIX V4.2.1.

Cluster Node.

RS/6000 system unit that participate in an HACMP cluster as a server.

Cluster information Services (clinfo).

Cluster Information Services is a daemon that exploits SNMP and makes cluster status information available to applications using the API.

Cluster SNMP Agent.

Cluster SNMP Agent (SMUXD) is the SNMP subagent for HACMP ES and maintains information about HACMP ES status.

Command Execution Language.

The language used to create commands that work across cluster nodes in a eight-node cluster. C-SPOC commands were created using Command Execution Language (CEL).

Concurrent Access.

In this configuration, two nodes are active simultaneously, sharing the same physical disk resources. The disk resources are defined as concurrent. Any other resources are divided between the two nodes, each owning some of them; the resources not owned by each node are designated as cascading. If either node fails, the other node takes over all of the resources. When the failed node rejoins the cluster, the resources are returned to the original owning node.

D

DARE (Dynamic Automatic Reconfiguration Events).

Dynamic Reconfiguration allows the user to change the configuration of a running cluster. That is, the definitions of cluster resources can be changed. These changes take effect immediately, without having to stop and restart the HACMP daemons, and without having to disrupt the applications running on the cluster.

Disk Mirroring.

Method of minimizing the effect of a disk failure by duplicating the contents of the disk. If a disk fails, the node can access the mirrored disk and continue to work.

Domain Name Server.

In TCP/IP, a server program that supplies name-to-address translation by mapping domain names to internet address. Synonymous with name server.

Dynamic Reconfiguration.

The process where changes made to the cluster configuration on one node are synchronized across all cluster nodes and the changed configuration becomes the currently active configuration.

E

Event.

The term *event* has several meanings:

- A predefined or system-defined event with HACMP ES that is included in the ODM like node_up event. See Cluster Event
- A user-defined event for HACMP ES that is the additional event that is added to the rules file by the user.
- A PSSP predefined event that is the ready-made event that is available in the SDR so that it can be displayed with the perspectives command. It notifies HACMP ES Cluster Manager when you add the event to the rules file.
- A PSSP user-defined event that is the event which is not a predefined event. Users have the flexibility to define an event of interest by using the perspectives command. HACMP ES Cluster

Manager can also subscribe this event if you add the event to the rules file.

Event Notification.

The user can specify a notify command that may send mail to the system administrator. This can be done to indicate that an event is about to happen and has just occurred. The message can contain success or failure information of the event script.

Event queue.

The event queue exists in Cluster Manager and stores events by priority.

Event Recovery.

The user can specify a command that attempts to recover from an event script failure. If the recovery command succeeds and the retry count for the event script is greater than zero, the event script is rerun. The number of times to retry the recovery command can be specified.

Event Recovery Command.

This is a HACMP ES phrasing, its function is the same as the HACMP Event Script. See HACMP Event Script and Cluster Event for further information.

Event Script.

See HACMP Event Script.

F

Failure Detection Rate.

The amount of time the cluster takes to detect a failure.

H

HACMP Event Script.

HACMP Event Script are the main scripts called by the Cluster Manager (HACMP for AIX) or by the HACMP ES Cluster Manager (HACMP ES). These scripts are normally located in the /usr/sbin/cluster/events directory and they are defined in the ODM. By using SMIT or the /usr/sbin/cluster/utilities/clchevent command you can add Pre and Post events, a Notify Command and a Recovery Command.

Hardware Address Takeover (HWAT).

Works in conjunction with IP address takeover. When an IP address is taken over by a surviving node's standby adapter, its hardware address is also taken over by that adapter. It can then continue to provide service to those clients that depend on the hardware address, rather than only the IP address.

HAView.

HAView allows you to monitor HACMP clusters through the NetView network management platform. The HAView application monitors the clusters using SNMP. HACMP provides a management information base (MIB) that contains information about cluster topology and state. HAView displays the configuration and state of the clusters and cluster components through the NetView graphical user interface. HAView allows you to search through a series of nested submaps that reflect the state of all the nodes, networks, and network addresses configured in a particular cluster.

Heartbeat.

The heartbeat is used to monitor adapter availability. Heartbeats are sent from one adapter to the adapter in the group with the next lower IP address. The adapter with the lowest IP address sends its heartbeat to the Group Leader, that is, the adapter with the highest IP address.

When an adapter failure occurs, the next lower adapter notifies the death of the adapter to the Group Leader after the time which is specified in the ODM has expired. Then Topology Services/ES detects the adapter failure event and notifies Group Services/ES of the event.

Hot Standby.

In this configuration, all resources are cascading with a single node having the highest priority for them all. If the owning node fails, the standby node takes over the resources. When the failed node rejoins the cluster, the resources are returned to the original owning node. That is, one processor is normally idle, waiting to recover should the other fail.

I

Internet Protocol (IP).

A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

IP Address.

The 32-bit address defined by the Internet Protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

IP Address Takeover (IPAT).

A networking capability that allows one node to assume the networking address of a node that has left the cluster. This assures the cluster will continue providing network service to clients.

J

Journalled File System (JFS).

AIX facility that uses database journaling techniques to protect the integrity of the file system meta data. This cannot be used in HACMP for concurrent access.

K

Keepalive.

Heartbeat or state of health message exchanges between network modules.

L

Lazy Update.

Process where the ODM definition of LVM components stored on cluster nodes that do not currently have a LVM component activated is not updated until a failover occurs. Alternatively, a user can deactivate the volume group on the local node and export and import the volume group on all the other cluster nodes. Lazy Update is only an option for LVM components under the control of HACMP for AIX.

Logical Volume Manager (LVM).

AIX facility that manages disks at the logical level. HACMP uses AIX LVM facilities to provide high availability.

M

MAC.

See Medium Access Control

Management Information Base.

See Simple Network Management Protocol

Medium Access Control.

Medium Access Control is a LAN, bottom sublayer of layer 2, IEEE 802.3 - 802.6 . Each adapter has a burned in medium access control (MAC) address (hardware address) this address is build out of 12 hex digests for example such a address my look like "10005ab1cf4e."

Membership.

The term *membership* has several meanings:

- Membership states the nodes or adapters which are currently joining.
- Membership is a one-phase protocol that results in an event being put on the event queue, and is initiated by Group Services/ES when a membership change occurs.

MIB.

See Simple Network Management Protocol

Mirroring.

AIX facility for maintaining more than one copy of stored data, to prevent loss of data.

Mutual Takeover.

In this configuration, resources are cascading and divided among the nodes; some are defined as owned by each node. If either node fails, the other node takes over all of the resources. When the failed node rejoins the cluster, the resources are returned to the original owning node. That is, each processor backs up the other.

N

Name Server.

- In TCP/IP, synonym for domain name server.
- In Internet communications, the station that translates host names into their respective internet addresses when requested by the stations on the network.

Network Interface.

Connects a node to a network. Synonym is interface

Notify Command.

See Event Notification

O

Object Data Manager (ODM).

AIX facility that stores objects describing AIX and HACMP entities.

P

Pre- and Post-Event Scripts.

These are user-defined scripts that execute specific commands before and after the HACMP ES Cluster Manager calls an event script.

Q

Quorum.

Quorum is an LVM facility that must be considered when configuring logical volume components in an HACMP environment. Quorum determines whether a volume group can be placed online, using the varyonvg command or whether it can remain online after a failure of one or more of the physical volumes (disks) in a volume group. Quorum checking is enabled by default.

R

Rotating Takeover or Rotating Standby.

This configuration is identical to a Hot Standby, except that when the failed node rejoins the cluster, the resources are not returned to the node until the standby node fails.

RS232 serial line.

See Serial Network

Run Time Parameters.

HACMP environmental conditions set per node.

S

Serial Network. An RS232 serial line that may be used to connect pairs of nodes in a cluster. It does not use TCP/IP for communication. In HACMP it can also be a SCSI bus using Target Mode SCSI. The purpose of the serial network is to prevent node isolation.

Service Adapter.

The primary connection between the node and the network. A node has one service adapter of each physical network to which it connects. The service adapter is used for AIX network connections and is the address published by the Cluster Information Program (Cinfo) to application programs that want to use cluster services.

Service Address.

See Service Adapter.

Shared disks.

Disks configured to serve more than one node. In the HACMP system, shared disks are physically connected to multiple nodes.

Single Point of Failure.

A Single Point of Failure exists when a critical cluster function is provided by a single component. If that component fails, there has no alternative way to provide that function and essential services become unavailable.

Snapshot.

The Cluster Node Snapshot captures a cluster configuration, creating text files that contain all the information necessary to configure a similar cluster. Once captured, these snapshots - created in ASCII text format - can be applied to a new cluster. Cluster Node Snapshot can also be used in conjunction with the HACMP graphical user interface: any snapshot can be viewed as a simple text file or as a graphical representation, enabling quick analysis and diagnosis.

Simple Network Management Protocol (SNMP).

In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application's Management Information Base (MIB).

Standby.

Idle resource available to replace another equal resource currently in use. For example, an adapter or a processor.

Synchronization.

The term *synchronization* has several meanings:

- To use barrier commands in a recovery command, HACMP ES Cluster Manager is able to act in line with the recovery action in the node that is joining a cluster while an event is processing.
- This is a prerequisite task to have the same ODM data within every cluster node using HACMP ES. Also, all nodes must have the same rules file in the same place, and the same recovery commands in the same place.

T

Takeover.

A process of an active node acquiring resources previously owned by another node, in order to maintain availability of those resources.

Target Mode SCSI.

A serial network that is using the SCSI connection between two nodes. It may be used to connect pairs of nodes in a HACMP cluster.

Tmcscli. See Target Mode SCSI

Transmission Control Protocol/Internet Protocol (TCP/IP).

A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

V

Voting.

Voting is a two-phase protocol initiated by a node whose event queue has stabilized. All nodes verify that the proposed next event is the highest priority event. If any node does not have this event on its queue, it adds it. If any node has a higher priority event, it rejects the protocol and initiates a vote for the highest priority event.

List of Abbreviations

ARP	Address Resolution Protocol	PTPE	Performance Toolbox Parallel Extension
MIB	Management Information Base	VSD	Virtual Shared Disks
MAC	Medium Access Control	RVSD	Recoverable Virtual Shared Disks
IP	Interface Protocol	AIX	Advanced Interactive Executive
IPAT	IP Adress Takeover	NFS	Network File System (USA, Sun Microsystems Inc.)
HWAT	Hardware Adress Takeover	GPFS	General Parallel File System
CWS	Control Workstation	C-SPOC	Cluster Single Point of Control
IBM	International Business Machines Corporation	SMIT	System Management Interface Tool
ITSO	International Technical Support Organization	NIM	Network Interface Module
ATM	Asynchronous Transfer Mode	DARE	Dynamic Automatic Reconfiguration Events
FDDI	Fiber Distributed Data Interface (100Mbit/s fiber optic LAN)	VSM	Visual System Management
SP	IBM RS/6000 Scalable POWERparallel System (RS/6000 SP)	HPS	High Performance Switch
TCP	Transmission Control Protocol	ODM	Object Data Manager
TCP/IP	Transmission Control Protocol/Internet Protocol	SDR	System Data Repository
Clinfo	Client Information Program	SNMP	Simple Network Management Protocol
IPAT	IP-address takeover	CPU	Central Processing Unit
UDP	User Datagram Protocol	EMAPI	Event Management Application Programming Interface
LAN	Local Area Network	PTX/6000	Performance Toolbox/6000
LVM	Logical Volume Manager	LPP	Licensed Program Product
HACMP	High Availability Cluster Multi-Processing	SMUXD	SNMP Multiplexor Daemon
HANFS	High Availability Network File System	SBS	Structured Byte String
HACMP ES	High Availability Cluster Multi-Processing Enhanced Scalability	DNS	Domain Name Server
PTF	Program Temporary FIX	ADSM	ADSTAR Distributed Storage Manager
PSSP	Parallel System Support Program	SCSI	Small Computer System Interface

Index

Special Characters

.rhosts 131
.rhosts 98

A

abbreviations 185
acronyms 185
adapter failure 31
adapter groups 44
adapter membership 30
adapter membership group 41, 42
adapter membership state 70, 179
Address Resolution Protocol 179
alias 179
aliasing 80, 179
all 55
alternate path 13, 80
API 36
application server 179
ARP 179
ATM 16
availability status 31

B

barrier 11, 69, 129, 179
barrier command 54, 58, 73
barrier protocol 47, 58, 73
base 96
base address 40
bibliography 173
boot address 98, 133, 135, 179
bos.compat 93
bos.net 93
bos.sysmgmt 93
broadcast 179

C

C-SPOC 22, 96, 179
cascading 85
cascading resource 179
cascading takeover 179
cbarrier 69, 179
CEL 22, 179
clconvert 126, 179
clconvert_snapshot 126
client 180
clinfo 16, 21, 29, 36, 37, 180
clruncmd 49
clstat 20, 37
CLSTRMGR 41, 71

cluster 180
cluster aware 37
cluster event 180
Cluster Information Services 29, 36, 37, 180
Cluster Manager 180
cluster node 180
cluster resources 47
cluster security mode 99
cluster SNMP agent 29, 37, 180
cluster state information 37
cluster-aware 83
cluster-ID 35
Command Execution Language 180
components 47
concurrent access 16, 85, 180
Concurrent Resource Manager 7
Control Workstation 35, 132, 135
css 38

D

DARE 16, 22, 180
define events 34
disk mirroring 180
domain 41
domain name server 180
Drag and Drop GUI 20
dsh 131
Dynamic Automatic Reconfiguration Events 16
dynamic reconfiguration 180

E

Ethernet 38
event 55, 73, 128, 180
event definition 129
event detection 31, 60, 127
Event Management 30, 33, 34, 49, 52, 60, 105
Event Management subscriber node 62
event name 51
event notification 26, 181
event priority 63
event queue 72, 74, 181
event recovery 181
event recovery command 181
event script 181
event scripts 25, 26, 147
expected status 55
ext_srvtab 137

F

failure 48
failure detection rate 181

FDDI 38
fibrillate count 101
forced down 77

G

global ODM 23
glossary 179
GPFS 92
graceful 77
graceful with takeover 77
group 41
group name 42
Group Services 30
Group Services log length 101
Group Services/ES 30, 31, 33, 41, 49
grpsvs 42

H

HACMP ES 7
 Cluster Manager 47
 installation 121
 migration 123
 status 36
HACMP ES Cluster Manager 30, 180
HACMP ES event 54
HACMP event script 181
HACMP for AIX 7
HACMP for NFS 7
HACWS 90
HAGEO 7, 15, 88
HANFS 16, 90
hardware address takeover 144, 181
HAView 20, 96, 181
heartbeat 30, 40, 181
heartbeat for Topology Services/ES 100
High Availability Infrastructure 8
hot standby 181
HWAT 144, 181

I

IBM Parallel System Support Program (PSSP) 8
installation 93
installation overview 95
installation steps 98
instance vector 52, 105
intelligent client 83
interfaces 132
internal Ethernet 13
internet protocol 181
interval between heartbeats 100
IP 181
IP address 132, 181
IP address aliasing 143, 144
IP address take-over 33, 38
IP address takeover 143, 181

IP alias 80
IPAT 143, 181

J

JFS 182
join request 71, 74
Journaled File System 182

K

kadmin 136
keepalive 182
kerberos 98, 131

L

layers 29
Lazy Update 23, 182
leaving the cluster 48
license 94
logical connections 36
Logical Volume Manager 182
Issrc 39, 42
LVM 182

M

MAC 144, 182
machines.lst 44
man pages 97
Management Information Base 182
manual intervention recover 49
mapping 32
medium access control 182
membership 48, 67, 182
membership list 42
membership protocol 47, 68
messages 97
MIB 182
mirroring 182
modular approach 43
modules 43
multiple HACMP ES clusters 35
multiple recovery commands 56
mutual takeover 182

N

naive clients 83
name server 182
network down 63
network interface 182
network up 63
NIM 18
node fails 63
node failure 77
node join 47

- node joins 63
- node membership 30
- node_set 55
- node_up 73, 75
- node_up_complete 73
- non-IP network 40
- notify command 182
- null 56

O

- Object Data Manager 182
- ODM 23, 182
- other 55, 58

P

- partition 35
- partition-bounded 15
- pcp 131
- Perspectives 104
- point-to-point connection 40
- point-to-point networks 16
- post event scripts 182
- pre event scripts 182
- pre-defined event 102
- pre-defined PSSP event 102
- predefined 55
- predefined events 49
- predicate 52, 60, 105
- prerequisite 93
- Provider Broadcast facility 68
- provider group 41, 68, 71
- provider group name 35
- PSSP 8
- PSSP Event Management 30
- PSSP Group Services 30
- PSSP Topology Services 30

Q

- qualifier 51
- quorum 182

R

- rc.cluster 44
- rcp 131
- rearm predicate 52, 105
- recovery actions 32
- recovery command 47, 49, 54
- recovery level 51
- recovery program 32, 45, 47, 49, 54, 62, 69, 128
- recovery type 51
- recovery_command 55
- relationship 47
- resource 98
- resource conditions 34, 60

- resource group 25
- resource program path 51
- resource state 60
- resource variable name 52, 105
- resource variables 60
- resources 48
- rotating 85
- rotating standby 183
- rotating takeover 183
- RS232 31, 38, 40, 80, 183
- RS232 serial line 183
- rsh 131
- rules file 47, 49, 51, 58
- rules.hacmprd 49, 51, 53, 102, 127, 129
- run time parameters 183
- run-time parameter 131
- RVSD 91

S

- scalability 9
- SCSI target mode 183
- SDR 35, 132
- sequence 44
- serial network 80, 183
- service adapter 183
- service address 98, 133, 135, 183
- shared disks 183
- Simple Network Management Protocol 183
- Single Point of Failure 183
- SMIT 18
- SMUXD 29, 36, 37, 180
- snapshot 23, 107, 123, 183
- SNMP 29, 36, 183
- SNMP Agent 180
- SNMP multiplexor daemon 37
- sorting algorithm 63
- SP Switch 38, 80
- SP Switch Board 13, 86
- splstdata 140
- srvtab 137
- standby 183
- standby address 98
- state 51
- structured byte string 106
- subscribing Event Management 62
- subsystem components 29
- swap adapter 63
- synchronization 183
- synchronize 58
- synchronizing point 102

T

- takeover 183
- Target Mode SCSI 38, 40, 183
- TCP/IP 183
- TCP/IP aliasing 80

tftpboot 137
tmscsi 38, 183
Token-Ring 38
topology 98
Topology Services 30
Topology Services log length 101
Topology Services/ES 30, 31, 33, 38
topsvcs 39
Transmission Control Protocol/Internet Protocol 183

U

unique license keys 83
user-controlled stops 77
user-defined event 10, 102
user-defined events 49, 60, 63, 127

V

verify 99
voting 68, 183
voting protocol 47, 62, 72, 75
VSD 91
VSM 20, 96

X

xhacmpm 17

ITSO Redbook Evaluation

HACMP Enhanced Scalability &titleline2.
SG24-2081-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at <http://www.redbooks.com>
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

What other redbooks would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)



Printed in U.S.A.

SG24-2081-00

