

HP Serviceguard I

H6487S L.00



Student guide — Part 2

Use of this material to deliver training without prior written permission from HP is prohibited.

HP Serviceguard I

H6487S L.00



Student guide — Part 2

Use of this material to deliver training without prior written permission from HP is prohibited.

© Copyright 2007 Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

This is an HP copyrighted work that may not be reproduced without the written permission of HP. You may not use these materials to deliver training to any person outside of your organization without the written permission of HP.

Pentium®, Intel Itanium™ Logo: Intel, Intel Inside, and Itanium registered trademarks of Intel Corporation in the U.S. and other countries and are used under license.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft®, Windows®, and MS Windows® are U.S. registered trademarks of Microsoft Corp.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Export Compliance Agreement

Export Requirements. You may not export or re-export products subject to this agreement in violation of any applicable laws or regulations.

Without limiting the generality of the foregoing, products subject to this agreement may not be exported, re-exported, otherwise transferred to or within (or to a national or resident of) countries under U.S. economic embargo and/or sanction including the following countries:

Cuba, Iran, North Korea, Sudan and Syria.

This list is subject to change.

In addition, products subject to this agreement may not be exported, re-exported, or otherwise transferred to persons or entities listed on the U.S. Department of Commerce Denied Persons List; U.S. Department of Commerce Entity List (15 CFR 744, Supplement 4); U.S. Treasury Department Designated/Blocked Nationals exclusion list; or U.S. State Department Debarred Parties List; or to parties directly or indirectly involved in the development or production of nuclear, chemical, or biological weapons, missiles, rocket systems, or unmanned air vehicles as specified in the U.S. Export Administration Regulations (15 CFR 744); or to parties directly or indirectly involved in the financing, commission or support of terrorist activities.

By accepting this agreement you confirm that you are not located in (or a national or resident of) any country under U.S. embargo or sanction; not identified on any U.S. Department of Commerce Denied Persons List, Entity List, US State Department Debarred Parties List or Treasury Department Designated Nationals exclusion list; not directly or indirectly involved in the development or production of nuclear, chemical, biological weapons, missiles, rocket systems, or unmanned air vehicles as specified in the U.S. Export Administration Regulations (15 CFR 744), and not directly or indirectly involved in the financing, commission or support of terrorist activities.

Printed in the USA

HP Serviceguard I

Student guide — Part 2

April 2007

Contents

Module 11 — Cluster and Package Online Reconfiguration

11-1. SLIDE: Serviceguard Online Reconfiguration.....	11-2
11-2. SLIDE: Add a Node while a Cluster Is Running.....	11-3
11-3. SLIDE: Remove a Node while a Cluster Is Running.....	11-5
11-4. SLIDE: Summary of Cluster Configuration Changes	11-6
11-5. SLIDE: Add a Package while a Cluster Is Running.....	11-8
11-6. SLIDE: Remove a Package while a Cluster Is Running	11-9
11-7. SLIDE: Modify a Package while the Cluster and Package Are Running.....	11-10
11-8. SLIDE: Modify a Package while the Cluster Is Running, but the Package Is Down.....	11-11
11-9. SLIDE: Summary: Package Modifications	11-12
11-10. LAB: Online Package Addition.....	11-13
11-11. LAB: Testing Package FAILOVER_POLICY	11-21
11-12. LAB: Testing the Cluster-Wide ASCII File	11-23

Module 12 — Highly Available NFS

12-1. SLIDE: Highly Available NFS Package	12-2
12-2. SLIDE: NFS Toolkit Files	12-4
12-3. TEXT PAGE: The <code>nfs.flm</code> (“File lock migration”) Script.....	12-5
12-4. SLIDE: The <code>nfs.cntl</code> and <code>hanfs.sh</code> Scripts.....	12-9
12-5. SLIDE: Using the NFS Toolkit	12-10
12-6. SLIDE: NFS Script Variables.....	12-12
12-7. LAB: The <code>nfs</code> Package	12-13
12-8. LAB: NFS Package Performance	12-20

Module 13 — The Highly Available Oracle Database

13-1. SLIDE: Highly Available Oracle Package – Overview	13-2
13-2. SLIDE: Serviceguard Toolkits.....	13-4
13-3. SLIDE: Database Toolkit Contents.....	13-5
13-4. SLIDE: Example: Oracle Application.....	13-6
13-5. SLIDE: Oracle Control Script Modification.....	13-7
13-6. SLIDE: Oracle Monitor Script	13-8
13-7. SLIDE: Oracle Cookbook	13-9
13-8. LAB: Oracle Package Configuration.....	13-11
13-9. LAB: Test Oracle Package Failover.....	13-35

Module 14 — EMS Resources and Serviceguard Packages

14-1. SLIDE: Review of Client/Server Management Architecture.....	14-2
14-2. SLIDE: MIB Structure	14-3
14-3. SLIDE: EMS Overview	14-4
14-4. TEXT PAGE: EMS Resource Class Hierarchy	14-5
14-5. SLIDE: Setting Up a Package to Use an EMS Resource	14-7
14-6. LAB: Monitoring File Systems with EMS.....	14-9
14-7. LAB: Monitoring <code>numUsers</code> with EMS	14-12

Contents

Module 15 — High Availability Networking

15-1. SLIDE: Network Redundancy.....	15-2
15-2. SLIDE: Multi-Network Environment	15-3
15-3. SLIDE: Redundant LAN Cards.....	15-4
15-4. SLIDE: Redundant Hubs	15-5
15-5. SLIDE: Redundant Routers.....	15-6
15-6. SLIDE: Redundant Client Networks	15-7
15-7. SLIDE: Multiple IP Addresses	15-8
15-8. LAB: Losing Heartbeat Packets.....	15-9

Module 16 — Rolling Upgrade Issues

16-1. SLIDE: Minimizing Planned Downtime.....	16-2
16-2. SLIDE: Rules for Rolling Upgrade.....	16-4
16-3. SLIDE: Serviceguard Rolling Upgrades.....	16-6
16-4. SLIDE: Operating System Rolling Upgrades.....	16-7
16-5. SLIDE: Cluster before Rolling Upgrade	16-8
16-6. SLIDE: Example of a Rolling Upgrade — Running Cluster with Packages Moved.....	16-9
16-7. SLIDE: Example of Rolling Upgrade — Node1 Upgraded to HP-UX 11.23.....	16-10
16-8. SLIDE: Example of Rolling Upgrade — Install Serviceguard 11.17, Rejoin Cluster.....	16-11
16-9. SLIDE: Example of Rolling Upgrade — Run Cluster with all Packages on Node1.....	16-12
16-10. SLIDE: Example of Rolling Upgrade — Upgrade Node2	16-13
16-11. SLIDE: When a Rolling Upgrade Is Not Possible	16-14
16-12. LAB: Performing a Rolling Upgrade	16-15
16-13. LAB Solution: Performing a Rolling Upgrade.....	16-19

Module 17 — LVM Maintenance for Packages

17-1. SLIDE: Make a Logical Volume / File System Larger or Smaller	17-2
17-2. SLIDE: LVM Maintenance to a Package	17-3
17-3. SLIDE: Add Disk to Volume Group Owned by a Package	17-4
17-4. SLIDE: Add Logical Volume / File System to Volume Group Owned by a Package.....	17-6
17-5. SLIDE: Add a Volume Group to a Package	17-8
17-6. LAB: Increasing the Size of a Logical Volume / File System.....	17-10
17-7. LAB: Add and Remove an Unused Disk to Volume Group "vg01", Owned by the "test" Package.....	17-11
17-8. LAB: Add and Remove a Logical Volume/File System to the Volume Group Owned by the "test" Package.....	17-13
17-9. LAB: Add a New Volume Group to the "test" Package.	17-15
17-10. LAB Solution: Increasing the Size of a Logical Volume / File System	17-17
17-11. LAB Solution: Add and Remove an Unused Disk to Volume Group "vg01", Owned by the "test" Package.....	17-18
17-12. LAB Solution: Add and Remove a Logical Volume/File System to the Volume Group Owned by the "test" Package.....	17-22
17-13. LAB Solution: Add a New Volume Group to the "test" Package.....	17-25

Module 18 — Serviceguard Manager

18-1. SLIDE: HP Cluster Monitoring Tools	18-2
18-2. SLIDE: Serviceguard Manager	18-3
18-3. SLIDE: Welcome to Serviceguard Manager.....	18-10
18-4. SLIDE: Serviceguard Manager Login	18-12
18-5. SLIDE: Cluster / Node / Package Map	18-15
18-6. SLIDE: Initial Display.....	18-16
18-7. SLIDE: Cluster Properties	18-17
18-8. SLIDE: Node Properties.....	18-18
18-9. SLIDE: Node Properties.....	18-19
18-10. SLIDE: Package Properties	18-20
18-11. SLIDE: Package Properties	18-21
18-12. SLIDE: Modify Package – Nodes	18-23
18-13. SLIDE: Modify Package – Parameters	18-24
18-14. SLIDE: Modify Package – Networks	18-25
18-15. SLIDE: Modify Package – Services.....	18-26
18-16. SLIDE: Modify Package – EMS Resources.....	18-27
18-17. SLIDE: Modify Package – Control Script.....	18-28
18-18. SLIDE: Modify Package – Roles.....	18-29
18-19. SLIDE: Operation Log	18-30
18-20. SLIDE: Online Help	18-31
18-21. LAB: Exploring Serviceguard Manager.....	18-32
18-22. LAB: Creating a Package with Serviceguard Manager.....	18-36

Module 19 — Troubleshooting Scenarios and Final Review

19-1. SLIDE: Troubleshooting Scenarios	19-3
19-2. SLIDE: A Conflict Exists within Us.....	19-4
19-3. SLIDE: Missing the Point	19-5
19-4. SLIDE: Busy, Busy, Busy	19-7
19-5. SLIDE: The Import of This Problem Is Crucial.....	19-8
19-6. SLIDE: What Makes Johnny Run?	19-10
19-7. SLIDE: Fee, Fi, Foh, Fum	19-12
19-8. SLIDE: Care to Comment?.....	19-13
19-9. SLIDE: Thrown for a Loop?.....	19-15
19-10. SLIDE: What's in a Name?	19-17
19-11. SLIDE: Final Review	19-18
19-12. LAB Solution: Troubleshooting Scenarios	19-20

Module 20 — Using HP Serviceguard with Integrity Virtual Machines

20-1. SLIDE: HP Integrity Virtual Machines Overview	20-2
20-2. SLIDE: VM Concepts – Terminology	20-3
20-3. SLIDE: VM Concepts – Virtual Machine Resources	20-5
20-4. SLIDE: VM Concepts – Backing Store	20-7
20-5. SLIDE: VM Concepts – Virtual Switches (vswitch)	20-8
20-6. SLIDE: Integrity VM Commands.....	20-9
20-7. SLIDE: Serviceguard and Integrity VM Use Models.....	20-10
20-8. SLIDE: Availability Model for Integrity Virtual Machines as Serviceguard Nodes	20-12
20-9. SLIDE: VMs as nodes – Virtual/Physical Cluster	20-13
20-10. SLIDE: VMs as nodes – Virtual/Virtual Cluster – VM Guests on Separate Physical Nodes	20-14

Contents

20-11. SLIDE: VMs as Nodes – Virtual/Virtual "Cluster in a Box"	20-15
20-12. SLIDE: Cluster in a Box – Sharing Backing Store	20-16
20-13. SLIDE: VMs as Nodes – Standby Server Consolidation	20-17
20-14. SLIDE: VMs as Nodes – Multiple Cluster Standby Server Consolidation	20-18
20-15. SLIDE: Configuring VM Guests as Serviceguard Nodes – Considerations	20-19
20-16. SLIDE: Integrity VMs as Serviceguard Packages	20-21
20-17. SLIDE: VMs as Serviceguard Packages Use Case Examples	20-22
20-18. SLIDE: VMs as Packages Usage Considerations	20-23
20-19. SLIDE: Configuring VM Guests as Packages	20-25
20-20. SLIDE: Configuring an Integrity Multi-Server Environment	20-27
20-21. SLIDE: Configure VM Guest on All Nnodes	20-29
20-22. SLIDE: HPVM Toolkit – /opt/cmcluster/toolkit/hpvm	20-31
20-23. SLIDE: HPVM Toolkit – Updating the Cluster	20-36
20-24. SLIDE: Starting a Serviceguard Packaged VM	20-37
20-25. SLIDE: Shutting Down a Serviceguard Packaged VM	20-38
20-26. SLIDE: Serviceguard on VM Host – LAN Failover	20-39
20-27. SLIDE: Summary	20-41

Appendix A — Serviceguard Commands Summary

Appendix B — Managing Serviceguard with HP OpenView

B-1. SLIDE: OpenView Products	B-2
B-2. SLIDE: SNMP Notification	B-3
B-3. SLIDE: SNMP	B-5
B-4. SLIDE: What Is Network Node Manager?	B-6
B-5. SLIDE: Event Configuration Integration	B-7
B-6. SLIDE: OpenView Operations Integration	B-8
B-7. SLIDE: OpenView Operations Agents	B-9
B-8. SLIDE: OVO Unix GUI Windows	B-10
B-9. SLIDE: HP OpenView Unix Java GUI with OpenView Navigator	B-11
B-10. SLIDE: OVO Windows GUI	B-12
B-11. SLIDE: OVO Template / Policy Integration	B-13
B-12. SLIDE: Sources of Data ... OVO	B-14
B-13. SLIDE: OpenView Operations and Operators	B-15
B-14. SLIDE: Other OpenView Products	B-16

Appendix C —Serviceguard Worksheets

C-1. SLIDE: Serviceguard Worksheets	C-1
---	-----

Appendix D — Integrating VxVM and Serviceguard

D-1. SLIDE: Serviceguard Disk Space Management Overview	D-2
D-2. SLIDE: VxVM Terminology Overview	D-4
D-3. SLIDE: VxVM Command Overview	D-6
D-4. SLIDE: Serviceguard/VxVM Software Overview	D-7
D-5. SLIDE: Cookbook Overview	D-9
D-6. SLIDE: Configure the rootdg (no vg00)	D-10
D-7. SLIDE: Configure the rootdg (if vg00 already exists)	D-12
D-8. SLIDE: Configure VxVM Disks	D-14
D-9. SLIDE: Configure VxVM Disk Groups	D-17
D-10. SLIDE: Configure a VxVM Volume	D-19
D-11. SLIDE: Import / Deport the Disk Group on the Other Node	D-22
D-12. SLIDE: Manage VxVM Disk Groups	D-24

D-13. SLIDE: Manage VxVM Volumes	D-26
D-14. SLIDE: Configure a Package	D-28
D-15. SLIDE: For Further Study	D-30

Appendix E — Alternate Lab: New Package Configuration: xc lock Package

Appendix F — HP Serviceguard Storage Management Suite for HP-UX 11i

F-1. SLIDE: Oracle Is Pivotal to your Business	F-2
F-2. SLIDE: Common Customer Pain Points	F-3
F-3. SLIDE: What Is the Serviceguard Storage Management Suite?	F-4
F-4. SLIDE: HP Serviceguard Storage Management Suite	F-5
F-5. SLIDE: Oracle Disk Manager (ODM) RAW Performance and File System Manageability F-.....	F-6
F-6. SLIDE: Realize a Performance Gain in Oracle Environments with Serviceguard SMS.....	F-7
F-7. SLIDE: Dynamic Multi-Pathing (DMP) Protect Against I/O Path Failure	F-8
F-8. SLIDE: Off-Host Processing with Volume Snapshots.....	F-9
F-9. SLIDE: RAW versus File System	F-11
F-10. SLIDE: Common Management Activities	F-12
F-11. SLIDE: Cluster File System.....	F-13
F-12. SLIDE: Standard Recovery from Backup	F-14
F-13. SLIDE: Integrated, Instant Recovery: Storage Rollback	F-15
F-14. SLIDE: Dynamic Storage Tiers: Policy-based Tiered Storage	F-16
F-15. SLIDE: For More Information.....	F-17

Contents

Module 11 — Cluster and Package Online Reconfiguration

Objectives

Upon completion of this module, you will be able to do the following:

- Change the cluster configuration while the cluster is running.
- Add a node to the cluster while the cluster is running.
- Remove a node from the cluster while the cluster is running.
- Add a package to a running cluster.
- Delete a package from a running cluster.
- Modify specific package attributes while the package is running.
- Modify a down package while the cluster and other packages are running.
- Perform the above operations by using the HP-UX command line.

11-1. SLIDE: Serviceguard Online Reconfiguration



Serviceguard Online Reconfiguration

- Ability to change cluster configuration while the cluster is running
- Online Node Functionality
 - Add a new node.
 - Remove an existing node.
- Online Package Functionality
 - Add a package.
 - Delete a package.
 - Modify some package attributes while package is running.
 - Modify entire package while cluster and other packages continue to run, but modified package is down.
 - Change Package Access Control Policy

Student Notes

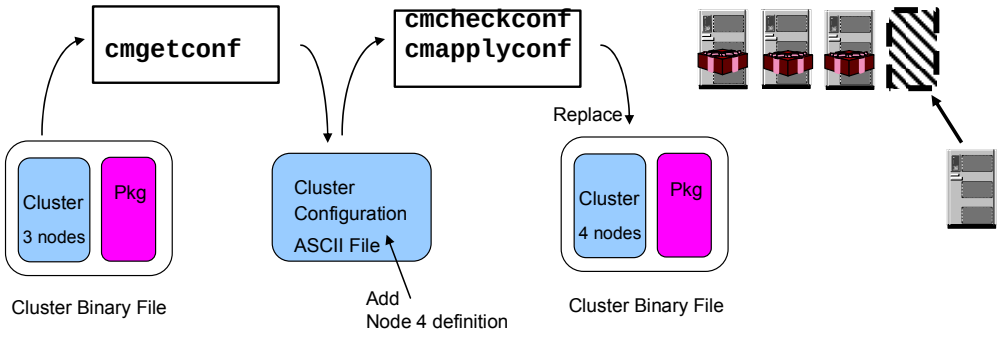
This module covers cluster and package maintenance tasks that can be performed while the cluster is up and running. Some of the tasks are related directly to cluster modifications, and other tasks focus more on package modifications. In some cases, a package can actually be reconfigured while the package itself is running. With "On-Line Modifications", only certain changes are permitted, and these modifications will be highlighted in the pages that follow.

11-2. SLIDE: Add a Node while a Cluster Is Running



Add a Node while a Cluster Is Running

- Update `cmclnodelist` on all nodes in the cluster (see notes below)
- `cmgetconf -v -c cluster1 temp.ascii`
- `cmquerycl -v -n n1 -n n2 -n n3 -n n4 -C cmclconfig.ascii`
- Combine the two configurations into one file. (`cmclconfig.ascii`)
- `cmcheckconf -v -C cmclconfig.ascii`
- `cmapplyconf -v -C cmclconfig.ascii`
- `cmrunnode -v n4`
- Modify package files to reflect new node
- If using a quorum server, update the `qs_authfile`



Student Notes

Remember to install Serviceguard on the new node before adding it to the cluster. Also, if the cluster is running a version of Serviceguard pre-A.11.16, create the `/etc/cmcluster/cmclnodelist` file on the new node, populating it with all nodes in the cluster. Be sure to add the new node to all of the existing nodes' `/etc/cmcluster/cmclnodelist` files, as well. If the cluster is running Serviceguard A.11.16 or later, this step should be skipped. Be sure to perform the above steps before doing the `cmquerycl` to add the new node to the cluster.

The above example demonstrates the Serviceguard commands and the order of such commands that are used to add a new node to an existing cluster. For the above example, nodes `n1`, `n2`, and `n3` are already configured in a running cluster named `mycluster`, and the desired action is to add node `n4`. This is all performed without interrupting a single event that is occurring on the existing cluster.

The first step probes the existing cluster and builds an ASCII file, `temp.ascii`, containing the configuration information for the existing cluster. The next step is used to generate a new template file, `cmclconfig.ascii`, for the desired cluster configuration. Remember to specify ALL nodes on the command line. The third step may be the most difficult. Compare `cmclconfig.ascii` with `temp.ascii` and make any necessary changes to

Cluster and Package Online Reconfiguration

`cmclconfig.ascii` so the desired configuration is captured. The next step is to verify the new configuration. The fifth step applies the changes to the configuration and sends the new binary configuration file to all cluster nodes. Next, start the new node. Remember to change any/all parameter values in the necessary start-up files.

If using a quorum server, be sure to update the quorum server `/etc/cmcluster/qs_authfile` to reflect the new node in the cluster.

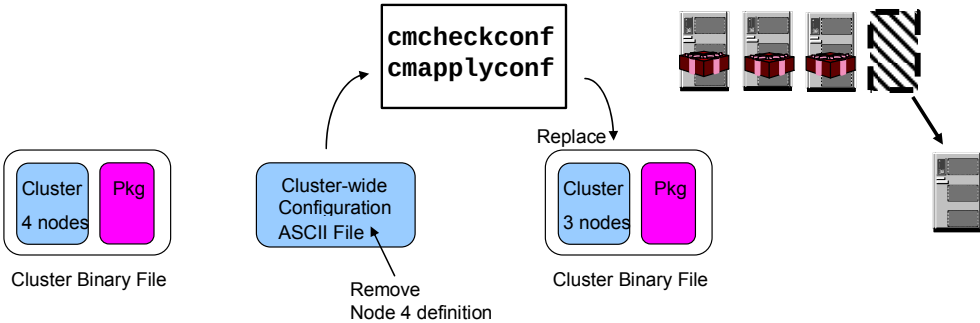
Also, be sure to run `/usr/sbin/qs -update` to force the quorum server to re-read the updated `qs_authfile`.

11-3. SLIDE: Remove a Node while a Cluster Is Running



Remove a Node while a Cluster Is Running

- `cmhaltpkg -v pkg_name`
- Remove any references to node 4 from all pkg configuration files
- `cmhaltnode -v node4`
- Edit the cluster configuration file and remove node 4's definition.
- `cmcheckconf -v -C cmclconfig.ascii`
- `cmapplyconf -v -C cmclconfig.ascii`
- `cmcheckconf -v` any altered package configuration files
- `cmapplyconf -v` any altered package configuration files



Student Notes

The above example demonstrates the Serviceguard commands and the order of such commands that are used to delete a node from an existing cluster. In the example, nodes n1, n2, n3, and n4 are all members of a running cluster. The desired action is to remove node n4 from the cluster.

The first step is to remove any package reference to node n4. Therefore, packages may need to be reconfigured. Next, halt the node to be removed. Edit the cluster ASCII file and remove all configuration information referring to node n4. Now verify the new configuration and apply the changes, which will also distribute the new binary configuration file to the remaining nodes.

If you are attempting to remove an unreachable node that has many packages dependent on it, especially if the dependent packages use a large number of EMS resources, you may see the following message:

The configuration change is too large to process while the cluster is running. Split the configuration change into multiple requests or halt the cluster.

In this situation, you must halt the cluster to remove the unreachable node.

11-4. SLIDE: Summary of Cluster Configuration Changes



Summary of Cluster Configuration Changes

As of SG version 11.17, the following summarizes available cluster changes:

Change to Cluster Configuration	Required Cluster State
Add a new node	All cluster nodes must be running.
Delete an existing node	A node can be deleted even if it is unavailable or unreachable.
Change Maximum Configured Packages	Cluster may be running
Change Timing Parameters	<i>Cluster must not be running.</i>
Change Cluster Lock or Quorum Server Configuration	<i>Cluster must not be running.</i>
Change serial device files	<i>Cluster must not be running.</i>
Change IP addresses for heartbeats or monitored subnets	<i>Cluster must not be running.</i>
Access control modifications	Cluster may be running

Student Notes

See the “Managing Serviceguard” document on <http://docs.hp.com> for the complete discussion of possible cluster configuration changes.

New nodes can be added to the cluster configuration or existing nodes can be deleted from the cluster configuration while the cluster is up and running. Note the following, however:

- You cannot change the lock disk configuration while the cluster is running.
- You cannot remove an active node from the cluster. You must halt the node first.
- You cannot delete an active volume group from the cluster configuration. You must halt any package that uses the volume group and ensure that the volume group is inactive before deleting it.
- You cannot change cluster timing parameters with the cluster running.

- The only configuration change allowed while a node is unreachable (for example, completely disconnected from the network) is to delete the unreachable node from the cluster configuration.

If there are also packages that depend upon that node, the package configuration must also be modified to delete that node.

Changes to the package configuration are covered in the next slide.

11-5. SLIDE: Add a Package while a Cluster Is Running



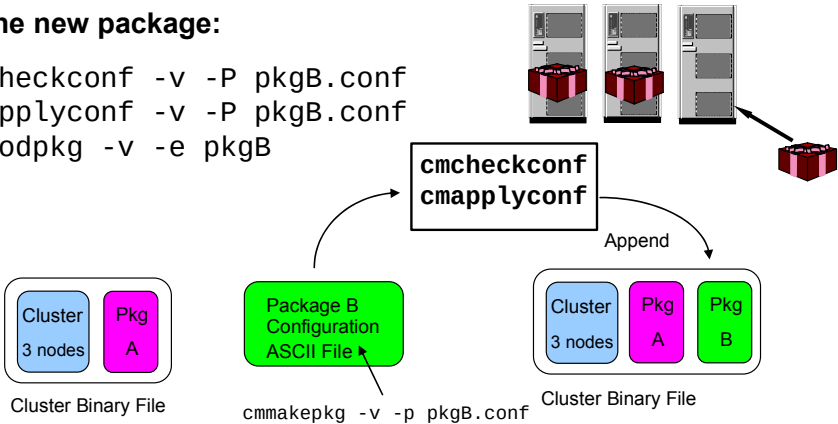
Add a Package while a Cluster Is Running

Normal package creation:

```
cmmakepkg -v -p pkgB.conf and edit as usual  
cmmakepkg -v -s pkgB.cntl and edit as usual  
chmod +x pkgB.cntl  
rcp pkgB.cntl node2:/etc/cmcluster/pkgB/pkgB.cntl
```

Apply the new package:

```
cmcheckconf -v -P pkgB.conf  
cmapplyconf -v -P pkgB.conf  
cmmodpkg -v -e pkgB
```



Cluster Binary File

Package B Configuration ASCII File

`cmmakepkg -v -p pkgB.conf`

Cluster Binary File

Append

Student Notes

It is possible to create a new package and add it to the cluster configuration while the cluster is up and while other packages are running. The number of packages you can add is subject to the value of `MAX_CONFIGURED_PACKAGES` as defined in the cluster-wide ASCII file.

To create the package, follow the steps for creating a package covered in the previous module with the following difference: do not specify the cluster name when verifying and distributing the configuration.

Remember to copy the control script to the `/etc/cmcluster/pkg_name` directory on all nodes that can run the package.

Adding a package to a running cluster is useful as your systems grow. Rather than purchasing a new set of systems, take advantage of the existing cluster configuration. This is especially useful when going from a 2-node active/standby configuration with an idle-node to a 2-node active/active configuration.

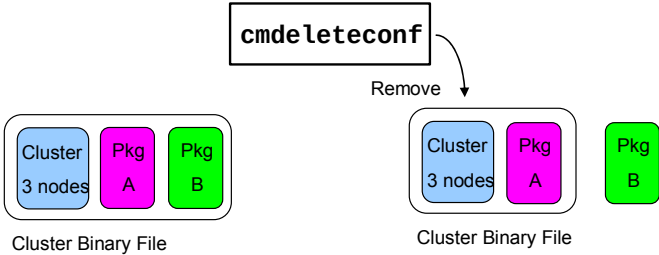
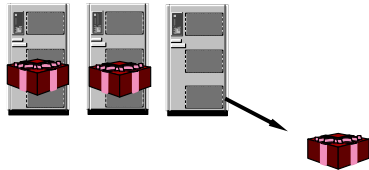
Note that any new hardware required for the newly added package must already be on all the systems for that package. For example, if adding a new package which requires a new FDDI network interface, the entire cluster would need to be brought down to install the FDDI network interface on each node in the cluster.

11-6. SLIDE: Remove a Package while a Cluster Is Running

Remove a Package while a Cluster Is Running



- `cmhaltpkg -v pkgB`
- `cmdeleteconf -v -p pkgB`
- `rm pkg.conf` (optional)
- `rm pkg.cntl` (optional)



Cluster Binary File

Student Notes

A package can be deleted from all cluster nodes by using the `cmdeleteconf` command. `cmdeleteconf` can only be executed when the package is not running; however, the cluster may be up. The command removes the package information from the binary configuration file on all the nodes in the cluster. The package configuration and control files are not removed.

Deleting a package is useful when you want to make major changes to a package. Rather than modifying the package, it might be easier to simply start with a new package by deleting the old one and adding the new one.

Caution: `cmdeleteconf`, without options, prompts to remove the entire cluster.

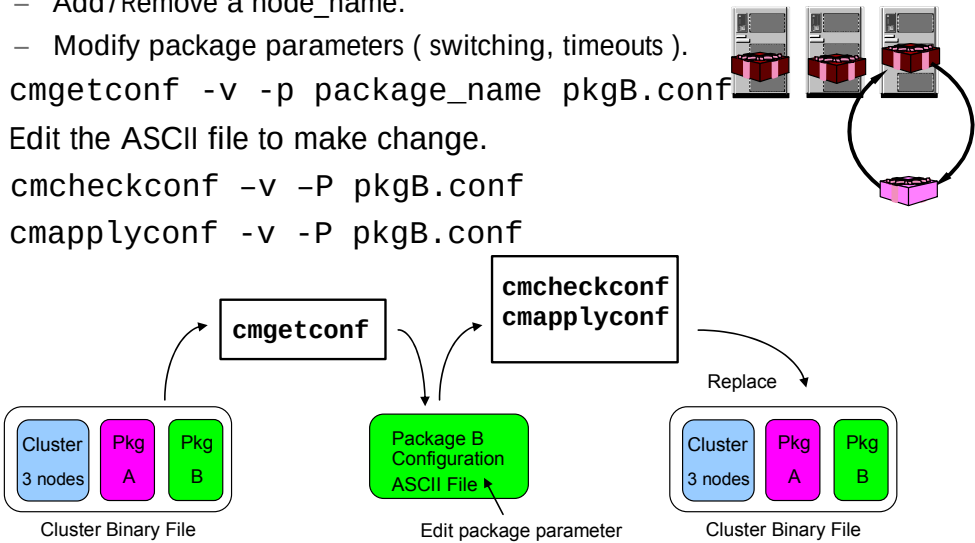
Syntax: `# cmdeleteconf -p pkg_name`

11-7. SLIDE: Modify a Package while the Cluster and Package Are Running

Modify a Package while the Cluster and Package Are Running



- What can be modified without bringing down the package?
 - Modify node failover order.
 - Add/Remove a node_name.
 - Modify package parameters (switching, timeouts).
- `cmgetconf -v -p package_name pkgB.conf`
- Edit the ASCII file to make change.
- `cmcheckconf -v -P pkgB.conf`
- `cmapplyconf -v -P pkgB.conf`



Student Notes

The items on the above slide can be performed on a package while the package is running. This implies that the cluster is also running. All nodes in the cluster must be powered up and accessible when making configuration changes.

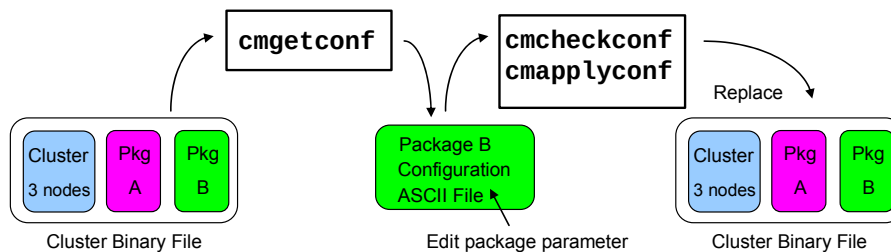
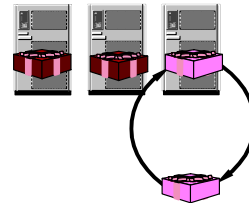
11-8. SLIDE: Modify a Package while the Cluster Is Running, but the Package Is Down

Modify a Package while the Cluster Is Running, but the Package Is Down



When do I need to bring the package down?

- Add or remove Service.
- Add or remove Subnet.
- Add or remove Resource (EMS).
- `cmhaltpkg -v -p pkgB`
- `cmgetconf -v -p pkgB pkgB.conf`
- Edit `pkgB.conf`
- `cmcheckconf -v -P pkgB.conf`
- `cmapplyconf -v -P pkgB.conf`



Student Notes

There are some instances when the package must be brought down. Fortunately, the cluster and remaining packages can remain up. To add a subnet or a resource to the package, the subnet or service must already be configured in the cluster. The next page provides a summary of the changes and the necessary state of a package and / or the cluster.

11-9. SLIDE: Summary: Package Modifications



Summary: Package Modifications

As of SG version 11.17, the following summarizes available package changes:

Package Modification	Required Package State
Add a new package	Other packages may be UP
Add a new node	Package(s) may be UP
Remove a node	Package(s) may be running on different nodes
Add / Remove a service	Package must be <i>halted</i>
Add / Remove a subnet	Package must be <i>halted</i>
Add / Remove a resource	Package must be <i>halted</i>
Change run / halt script contents	Recommended that package be <i>halted</i> . Timing problems may occur if the script is modified while the package is running
Script timeouts	Package may be UP
Service timeouts	Package must be <i>halted</i>
Failfast parameters	Package must be <i>halted</i>
AutoRun	Package may be UP
Local LAN Failover	Package may be UP
Change node adoption order	Package may be UP
Access control modifications	Package may be UP
Failover / Failback Policy	Package may be UP

Student Notes

All nodes in the cluster must be powered up and accessible when making configuration changes. The above table summarizes package maintenance task and package state requirements.

See the “Managing Serviceguard” document for the complete discussion of possible package configuration changes.

This document is available at no cost from <http://docs.hp.com>

11-10. LAB: Online Package Addition

Directions

Special Notes

- a. In this lab, we create an additional package, called "test", without bringing down the cluster.
- b. This new package will also be our first package to use disk resources. Specifically, we will use the vg01 volume group that was previously created in Module 4.
- c. The new package should execute the "test_program" script which will write to the shared volume group vg01.

Part 1: Create the Package Configuration File

1. Make a directory for the package files.

Answer:

```
# cd /etc/cmcluster  
# mkdir test
```

2. Change directory to the package directory that was just created.

Answer:

```
# cd test
```

3. Create the package configuration file template.

Answer:

```
# cmmakepkg -v -p test.conf
```

Module 11
Cluster and Package Online Reconfiguration

4. Edit the package configuration file template.

```
# vi test.conf
```

and modify the following parameters:

PACKAGE_NAME	test	
NODE_NAME	<node1>	
NODE_NAME	<node2>	
RUN_SCRIPT	/etc/cmcluster/test/test.cntl	
RUN_SCRIPT_TIMEOUT	NO_TIMEOUT	
HALT_SCRIPT	/etc/cmcluster/test/test.cntl	
HALT_SCRIPT_TIMEOUT	NO_TIMEOUT	
SERVICE_NAME	test_service	
SUBNET	X.X.X.X	← Use netstat -in to obtain cluster 10-net subnet (verify with instructor)

5. Save the changes, and exit the file.

Part 2: Create and Distribute the Package Control Script

1. Create the package control script template.

Answer:

```
# cmmakepkg -v -s test.cntl
```

2. Edit the package control script template.

```
# vi test.cntl
```

First, find the volume groups section, which now reads:

```
#VG[0]=""
```

Modify this value so that it now reads:

```
VG[0]="vg01" ← Don't forget to uncomment this line
```

Next, go to the section for logical volumes, which now reads:

```
#LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=""; FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""  
#FS_TYPE[0]=""
```

Break these lines up so that each variable is on a separate line (uncomment the lines, remove the semi-colons and the blank space, then make the lines look like the following):

```
LV[0]=""  
FS[0]=""  
FS_MOUNT_OPT[0]=""  
FS_UMOUNT_OPT[0]=""  
FS_FSCK_OPT[0]=""  
FS_TYPE[0]=""
```

Then, modify the following parameters:

```
LV[0]="/dev/vg01/sglvol1"  
FS[0]="/sgdata1"  
FS_MOUNT_OPT[0]=""  
FS_UMOUNT_OPT[0]=""  
FS_FSCK_OPT[0]=""  
FS_TYPE[0]="vxfs"
```

Next, go to the “Service Names and Commands” section, which now reads:

```
#SERVICE_NAME[0]=""  
#SERVICE_CMD[0]=""  
#SERVICE_RESTART[0]=""
```

Modify this section so that it now reads ...

```
SERVICE_NAME[0]="test_service"           ← Do not forget  
SERVICE_CMD[0]="/etc/cmcluster/test/test.mon" ← to uncomment  
SERVICE_RESTART[0]="-r 0"               ← these lines.
```

3. In the `customer_defined_run_cmds` function in the control script, replace the colon (:) with the following code. We use this code to start the `test_program` script:

```
/etc/cmcluster/test/test_program &
```

Also, in the `customer_defined_halt_cmds` function, replace the colon (:) with:

```
/etc/cmcluster/test/kill_test
```

where `kill_test` will contain the script for terminating the `test` package. We will write this script in Part 3 of this lab.

4. Save the changes, exit the file, and make it executable.

Answer:

```
# chmod +x test.cntl
```

5. Now, copy the `test.cntl` control script to other nodes within the cluster.

Answer:

```
# remsh <node2> "/usr/bin/mkdir /etc/cmcluster/test"
```

```
# rcp test.cntl <node2>:/etc/cmcluster/test/.
```

Part 3: Create and Distribute the Package Monitor and Kill Scripts

1. Create the test script monitor /etc/cmcluster/test/test.mon, as follows:

```
# vi /etc/cmcluster/test/test.mon
```

```
#!/usr/bin/sh

LOG="/etc/cmcluster/test/test.cntl.log"

echo Now entering the monitor on $(hostname) at $(date). >> $LOG
while true
do
    echo "Checking for the existence of the \"test_program\" script. "
    if ps -ef| grep -v grep | grep -q "/etc/cmcluster/test/test_program"
    then
        echo Found \"test_program\" on $(hostname) at $(date).
        sleep 30
    else
        echo "Could NOT find \"test_program\" on $(hostname) at $(date). "
        echo "The monitor is now exiting on $(hostname) at $(date). "
        exit
    fi
done >> $LOG
echo "We should never see this line of code. " >> $LOG
```

2. Create the script /etc/cmcluster/test/kill_test, as follows. This script will, as needed, kill the test_program script.

```
# vi /etc/cmcluster/test/kill_test
```

```
#!/usr/bin/sh

LOG="/etc/cmcluster/test/test.cntl.log"

echo Now entering \"kill_test\" on $(hostname) at $(date).      >> $LOG

echo "Now searching for all occurrences of \"c\"                >> $LOG
echo "the \"test_program\" script on $(hostname) at $(date).\" >> $LOG
if ps -ef | grep -v grep | grep -q test_program
then
    for pid in $(ps -ef | grep -v grep | grep \
                  "test_program" | cut -c10-14)
    do
        echo "Now trying to kill all occurrences of \"c\"      >> $LOG
        echo "the \"test_program\" script. \"                >> $LOG
        if ps -ef | cut -c10-14 | grep -q "$pid"
        then
            kill -9 $pid
        fi
    done
fi
echo "End of the \"kill_test\" program.  Now exiting ..."    >> $LOG
```

3. Make the test.mon and kill_test scripts executable.

Answer:

```
# cd /etc/cmcluster/test
# chmod 755 test.mon
# chmod 755 kill_test
```

4. Copy both the test.mon and kill_test scripts to the other system.

Answer:

```
# rcp test.mon <node2>:/etc/cmcluster/test/.
# rcp kill_test <node2>:/etc/cmcluster/test/.
```

Part 4: Create and Distribute the Test Program

1. Create the test_program script (in the /etc/cmcluster/test directory).

```
#!/usr/bin/sh

count=0
while true
do
    echo $(hostname) $(date) $count >> /sgdata1/test_data
    let count=count+1
    sleep 5
done
```

2. Make the test_program script executable, and copy it to the test package directory on the other node.

Answer:

```
# chmod 755 test_program
# rcp test_program <node2>:/etc/cmcluster/test/.
```

Part 5: Check the Package Configuration File, Distribute the Binary File and Start the Package

1. Check the configuration file for errors.

Answer:

```
# cd /etc/cmcluster/test
# cmcheckconf -v -P test.conf
```

2. Fix any errors. Once the configuration is OK, apply the new configuration.

Answer:

```
# cmapplyconf -v -P test.conf
```

3. Once the new configuration is applied and distributed, enable the package.

Answer:

```
# cmmodpkg -v -e test
```

4. Verify the /sgdata1/test_data file is growing by doing a
tail -f /sgdata1/test_data on the node which has the file system mounted.

Answer:

```
# tail -f /sgdata1/test_data
```

5. Execute /etc/cmcluster/test/kill_test to verify that the test package fails over correctly and mounts on the adoptive node. Execute tail -f /sgdata1/test_data on the adoptive node.

Answer:

Node 1:

```
# /etc/cmcluster/test/kill_test
```

Node 2:

```
# tail - f /sgdata1/test_data
```

11-11. LAB: Testing Package FAILOVER_POLICY

Directions

In this lab you will change the FAILOVER_POLICY of one of your packages from CONFIGURED_NODE to MIN_PACKAGE_NODE.

On the Configuration Node

1. While the following package policy changes would not require halting the cluster, the following exercises are better demonstrated if we do so. Therefore, halt the cluster now.

Answer:

```
# cmhaltcl -v -f
```

2. Before continuing, make a backup copy of the logprog.conf and the test.conf files.

Answer:

```
# cd /etc/cmcluster/logprog
# cp logprog.conf logprog.conf.save
# cd ../test
# cp test.conf test.conf.save
```

3. Examine the package configuration file for both the logprog and test packages. Notice the FAILOVER_POLICY is currently set to CONFIGURED_NODE. This policy means that the package will be run on the first node listed in the node list, unless that node is unavailable or the package is disabled on that node.

Configure both packages so that they have the same primary node. Use Node1 as the primary node (they may already be set up this way). If you modified the configuration, check and apply the modified files.

Answer:

```
# cd /etc/cmcluster/logprog
# vi logprog.conf

# cd ../test
# vi test.conf
```

Module 11
Cluster and Package Online Reconfiguration

4. Start the cluster. Verify that both packages are running on the same node (Node1).

Answer:

```
# cmruncl -v
```

5. Now, halt the cluster.

Answer:

```
# cmhaltcl -v -f
```

6. Modify the package configuration file for test. Change the FAILOVER_POLICY to MIN_PACKAGE_NODE. Check and apply the modified configuration.

Answer:

```
# cd /etc/cmcluster/test
# vi test.conf
# cmcheckconf -v -P test.conf
# cmapplyconf -v -P test.conf
```

7. Start the cluster. This time test should have started on the other node (Node2), since it had fewer packages running.

Answer:

```
# cmruncl -v
```

8. Restore the backup copy of the test configuration file. Reapply the restored configuration.

Answer:

```
# cd /etc/cmcluster/logprog
# cmhaltpkg -v logprog
# cmhaltpkg -v test
# cp logprog.conf.save logprog.conf
# cd ../test
# cp test.conf.save test.conf
# cmcheckconf -v -P test.conf -P ../logprog/logprog.conf
# cmapplyconf -v -P test.conf -P ../logprog/logprog.conf
# cmmodpkg -v -e logprog
# cmmodpkg -v -e test
```


11-12. LAB: Testing the Cluster-Wide ASCII File

Directions

1. Make a backup of the cluster ASCII file.

Answer:

```
# cd /etc/cmcluster
# cp cmclconfig.ascii cmclconfig.ascii.save
```

2. Modify the cluster ASCII file as follows:

```
# vi cmclconfig.ascii
```

Do step 2a only if you are using LVM disks for cluster lock functionality; otherwise, do steps 2b, 2c, and 2c, and proceed to step 4.

- a. Change the FIRST_CLUSTER_LOCK_VG to /dev/vglock (a non-existent volume group).
 - b. Modify the HEARTBEAT_INTERVAL to 15 seconds.
 - c. Modify MAX_CONFIGURED_PACKAGES to 200.
 - d. Make any other changes you would like to test.
3. With the cluster running, try to execute cmcheckconf on the cluster ASCII cluster configuration file. What happens?

Answer:

```
# cd /etc/cmcluster
# cmcheckconf -v -C cmclconfig.ascii
```

cmcheckconf reports "First cluster lock volume group /dev/vglock cannot be found in the cluster."

4. If cmcheckconf found an erroneous cluster lock volume group in the cluster ascii file fix it and attempt another cmcheckconf. What happens?

Answer:

```
# cmcheckconf -v -C cmclconfig.ascii
```

cmcheckconf reports "Modifying HEARTBEAT_INTERVAL value from 3000000 to 15000000 while cluster cluster_name is running is not supported.
cmcheckconf: Unable to verify cluster file: cmclconfig.ascii.

5. Halt the cluster. Reissue cmcheckconf. Will Serviceguard allow these changes to be made?

Answer:

```
# cmhaltcl -v -f  
# cd /etc/cmcluster  
# cmcheckconf -v -C cmclconfig.ascii
```

No.cmcheckconf reports "Node timeout (6.00 seconds) should be at least two times the heartbeat interval (15.00 seconds).
Maximum configured packages must be greater than or equal to 0, and less than or equal to 150.
cmcheckconf: Unable to verify cluster file: cmclconfig.ascii.

6. Restore the original copy of the cluster ASCII file and re-apply this configuration.

Answer:

```
# cd /etc/cmcluster  
# mv cmclconfig.ascii.save cmclconfig.ascii  
# cmcheckconf -v -C cmclconfig.ascii  
# cmapplyconf -v -C cmclconfig.ascii
```

7. Restart the cluster.

Answer:

```
# cmruncl -v
```

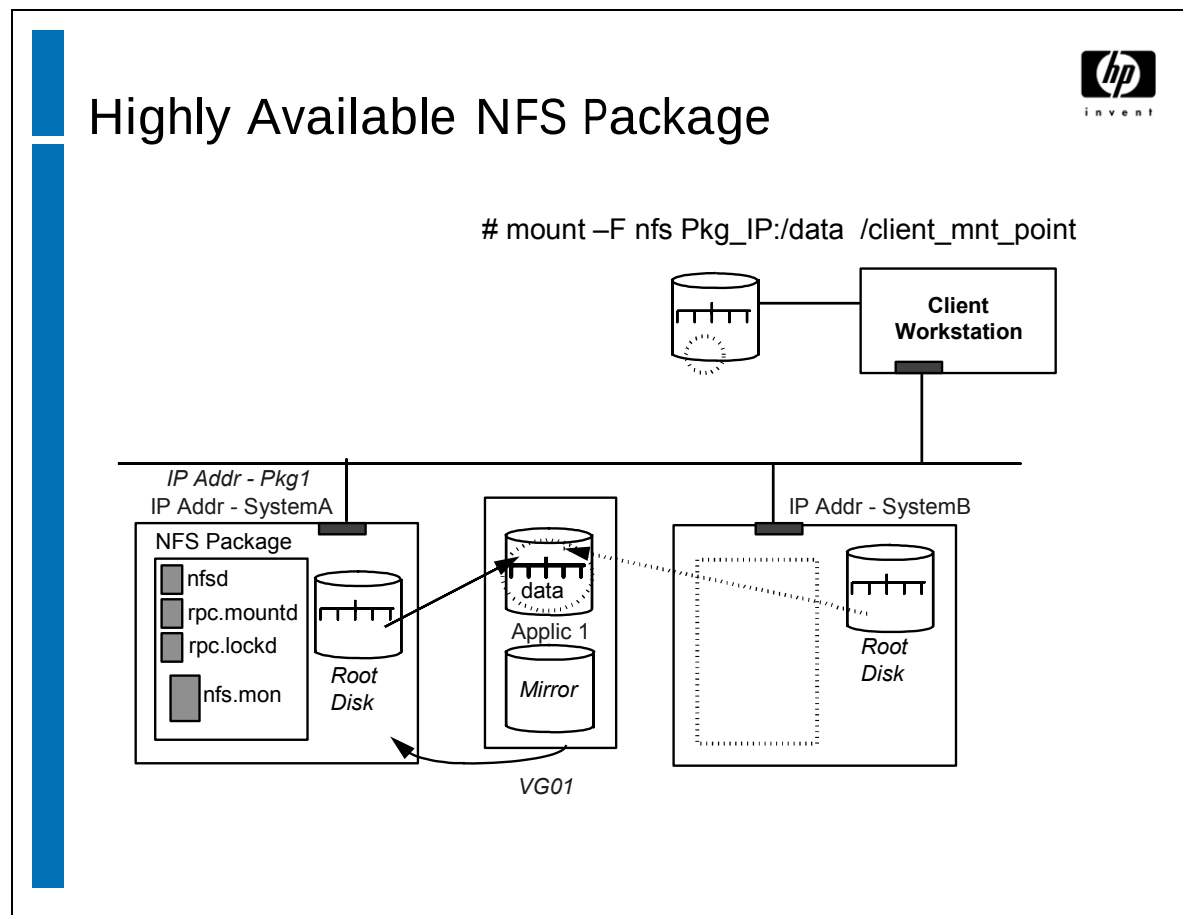
Module 12 — Highly Available NFS

Objectives

Upon completion of this module, you will be able to do the following:

- Use the Serviceguard NFS toolkit.
- List files in the NFS toolkit.
- Set up a Serviceguard package using the NFS toolkit.

12-1. SLIDE: Highly Available NFS Package



Student Notes

The purpose of highly available NFS servers is to ensure that clients will be able to access files and data located on a server file system. This is most beneficial to file server-based applications and data servers.

The concept is to group the daemons related to NFS into a package that can run on multiple hosts. The NFS file system is then placed in a shared volume group which any host in the cluster can access. Once configured, any host in the cluster is capable of providing NFS access to the data for the client workstations on the network.

The clients access the NFS file system through the package IP address, not the IP address of a host. The mount command issued by a client workstation would be in the following form:

```
# mount -F nfs NFS_Package_IP:/NFS_exported_fs /local_mnt_point
```

If the node providing NFS access to the file system fails, Serviceguard will move the NFS package to another node within the cluster. The new node will activate the volume group, mount the file systems, check that the NFS daemons are running, execute the `customer_defined_run_cmds` and start the NFS monitor script.

Client NFS packets will be automatically routed to the new host because that is where the package IP address now resides. Clients will be able to continue to access their data without remounting.

12-2. SLIDE: NFS Toolkit Files



NFS Toolkit Files

As of SG version 11.17, the NFS Toolkit (part number B5140BA) is available for HPUX 11.00, HP-UX 11.11, and HP-UX 11i v2.

The NFS Toolkit contains the following files:

- README
- hanfs.sh
- nfs.flm
- nfs.mon
- nfs_xmnt

The `nfs.conf` and `nfs.cntl` files are needed, as well, and are created with the `cmmakepkg` command.

Student Notes

The NFS toolkit is made up of five files. The files are located in the `/opt/cmcluster/nfs` directory and are typically copied from there to the `/etc/cmcluster/nfs` directory.

The following is a listing of these five files, and a very brief description of each file.

README	A general discussion of the toolkit.
hanfs.sh	This script was new at version 11.14 of Serviceguard and the NFS Toolkit. This script is where we will identify the file system which will be nfs-exported. In this file, we also identify the <code>NFS_SERVICE_NAME</code> , <code>NFS_SERVICE_CMD</code> and the <code>NFS_SERVICE_RESTART</code> values.
nfs.flm	This script handles the NFS file lock propagation on HA servers cluster-wide. It contains specific environment variables and functions for providing the NFS file lock migration feature.
nfs.mon	The NFS monitoring script. This will be the service process that monitors the health of the NFS daemons.
nfs_xmnt	A customized script used for cross-mounting the NFS Server package.

12-3. TEXT PAGE: The nfs.flm (“File lock migration”) Script

The following is a copy of the nfs.flm “File lock migration” script for HP-UX 11i v2.

```
#!/usr/bin/sh
# A.11.23.02
# *****
# *
# *      SERVICEGUARD NFS TOOLKIT FILE LOCK MIGRATION SCRIPT      *
# *
# *      Note: This is a template NFS file lock migration script.   *
# *      It MUST be edited before it can be used.                  *
# *
# *****
#####
#
# This script handles the NFS file lock propagation on HA servers
# cluster wide. It contains specific environment variables and functions
# for providing the NFS file lock migration feature.
#
# Each HA/NFS package requiring NFS File Lock Migration after a failover
# should run an invocation of this script and specify a unique holding
# directory via the NFS_FLM_HOLDING_DIR variable.
#
# Implementation:
#
#   Use cp(1) to copy the contents of the /var/statmon/sm directory to
#   a holding directory in one of the disk volumes associated with
#   this specific HA/NFS package.
#
#   During package start-up, the contents of the holding directory
#   will be merged into the /var/statmon/sm directory on the server
#   running the package.
#
#####
#
# The NFS_FLM_HOLDING_DIR variable defines the holding directory in a
# disk volume associated with the HA/NFS package. This directory
# will hold copies of the files from the /var/statmon/sm directory
# and will migrate with the HA/NFS package during a failover. The
# contents of this directory will be used to populate the
# /var/statmon/sm directory when recovering from a failover event.
#
# Select a directory in a file system that is part of the
# HA/NFS package.
#
# Example: NFS_FLM_HOLDING_DIR="/export/sm"
#
# where "/export/sm" is a directory residing in a disk volume
# associated with this HA/NFS package.
#
NFS_FLM_HOLDING_DIR=""

#
# Define the interval in seconds to copy the file lock state
# (i.e. monitoring entries) files into the holding directory.
# The default is 5 seconds.
#
```

Module 12

Highly Available NFS

PROPAGATE_INTERVAL=5

```
function validate_holding_dir {
#
# Make sure the configured holding directory (configured via the
# NFS_FLM_HOLDING_DIR variable) exists and has the proper ownership
# and permissions.
#
# First check to make sure the NFS_FLM_HOLDING_DIR variable has
# been configured.  If not, print an error and exit.
#
if [[ -z $NFS_FLM_HOLDING_DIR ]]
then
    print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
    print "\"$(hostname)\": The NFS File Lock Migration holding \c"
    print "directory has not been configured."
    print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
    print "\"$(hostname)\": Set the NFS_FLM_HOLDING_DIR variable \c"
    print "in the ${0##*/} script.  Exiting."
    exit 1
fi

if [[ ! -d $NFS_FLM_HOLDING_DIR ]]
then
    print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
    print "\"$(hostname)\": The NFS File Lock Migration holding \c"
    print "directory \"$NFS_FLM_HOLDING_DIR\" does not exist."
    print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
    print "\"$(hostname)\": Creating directory \"$NFS_FLM_HOLDING_DIR\""

    mkdir -p -m 644 $NFS_FLM_HOLDING_DIR 2>&1
    if [[ $? -ne 0 ]]
    then
        print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
        print "\"$(hostname)\": Failed to create \c"
        print "\"$NFS_FLM_HOLDING_DIR\".  Exiting."
        exit 1
    fi

    chown root:sys $NFS_FLM_HOLDING_DIR
else
    chmod 644 $NFS_FLM_HOLDING_DIR
    chown root:sys $NFS_FLM_HOLDING_DIR
fi
}
```

```
function copy_sm_files {
#
# This function is used to copy Status Monitor files between
# the package holding directory (configured via the
# NFS_FLM_HOLDING_DIR variable) and the Status Monitor
# directory (/var/statmon/sm).  This function takes two input
# parameters - the first is the source directory to copy SM
# files from and the second is the target directory to copy
# SM files to.
#
# The server's Status Monitor directory (/var/statmon/sm) gets
# populated just prior to killing and restarting the
# rpc.statd and rpc.lockd daemons during package startup.
# The holding directory get populated with current SM entries
# at a defined interval (PROPAGATE_INTERVAL) while the package
```



```
# runs. This allows the holding directory to only maintain
# the list of NFS clients that have performed file locks since
# the lockd/statd daemons were re-started at package start time.
#
# Before copying files, check to see if there are any files in
# the source directory to copy. If there are then copy the
# files and check for errors that occurred during the copy.
# Only print notification of successful file copy in the case
# where we are populating the /var/statmon/sm directory.
#
SOURCE_DIR=$1
TARGET_DIR=$2

if [ -f $SOURCE_DIR/* ]
then
    cp $SOURCE_DIR/* $TARGET_DIR/. 2>&1
    if [[ $? -ne 0 ]]
    then
        print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
        print "\"$(hostname)\": FAILED to copy NFS File Lock \c"
        print "entries from \"$SOURCE_DIR\" to \"$TARGET_DIR\""
        exit 1
    else
        if [[ $SOURCE_DIR = $NFS_FLM_HOLDING_DIR ]]
        then
            print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
            print "\"$(hostname)\": Populated \c"
            print "\"$TARGET_DIR\" with contents of \c"
            print "\"$SOURCE_DIR\": \c"
            ls $SOURCE_DIR 2>&1
        fi
    fi
fi
}

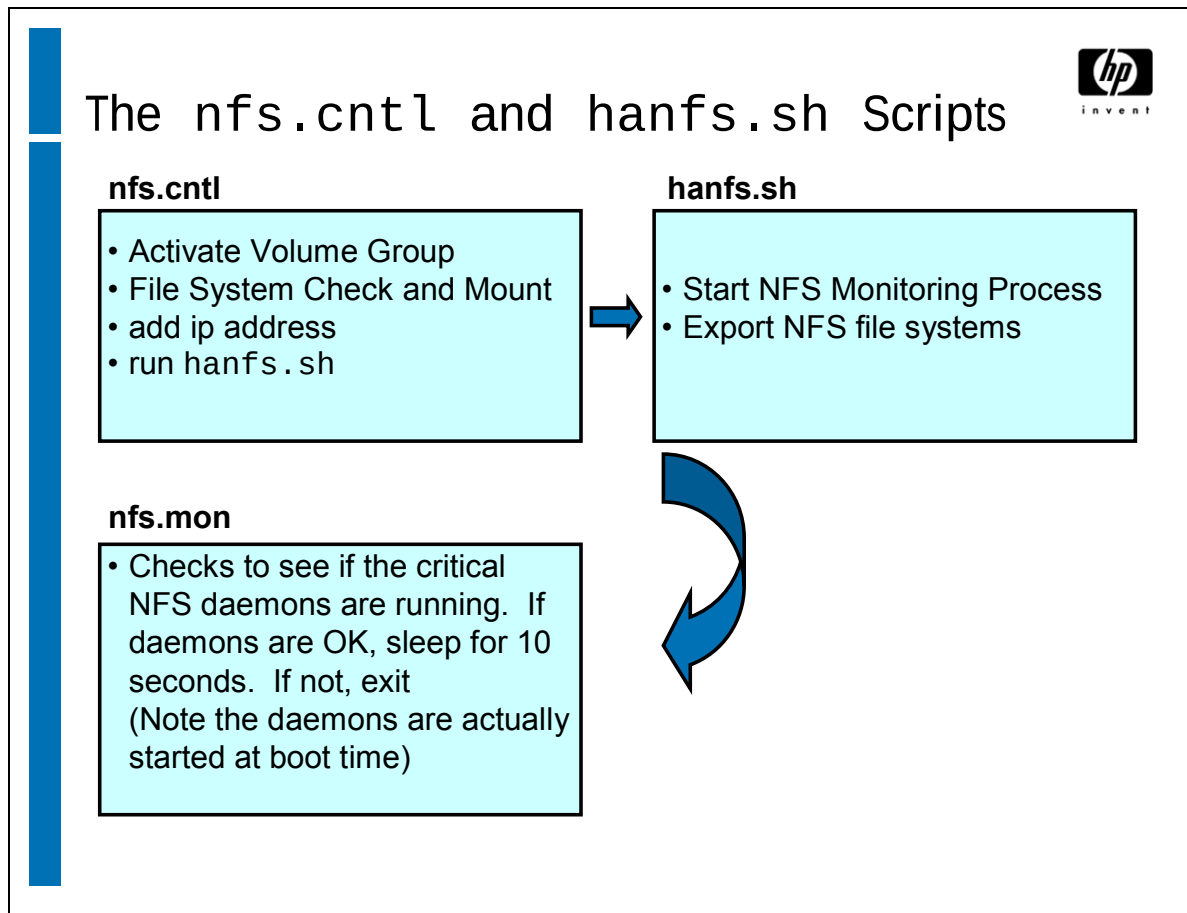
function remove_stale_sm_files {
#
# This function simply removes any files residing in the holding
# directory. It is called only after the holding directory contents
# have been copied to the SM directory during package startup.
#
if [ -f $NFS_FLM_HOLDING_DIR/* ]
then
    rm -f $NFS_FLM_HOLDING_DIR/* 2>&1
    if [[ $? -ne 0 ]]
    then
        print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
        print "\"$(hostname)\": Failed to remove stale FLM \c"
        print "entries from holding directory"
    else
        print "HANFS FLM -- $(date '+%b %e %X') - Node \c"
        print "\"$(hostname)\": Removed stale FLM entries \c"
        print "from \"$NFS_FLM_HOLDING_DIR\" directory"
    fi
fi
}
```

Module 12
Highly Available NFS

```
#-----MAINLINE NFS Code Starts Here-----  
#  
# Make sure the holding directory exists and has the proper permissions  
#  
validate_holding_dir  
  
#  
# Test to see if we are being called to populate the sm directory  
# or populate the holding directory.  
#  
if [[ $1 = "populate_sm_dir" ]]  
then  
    copy_sm_files $NFS_FLM_HOLDING_DIR /var/statmon/sm  
    remove_stale_sm_files  
elif [[ $1 = "populate_holding_dir" ]]  
then  
    while true  
    do  
        copy_sm_files /var/statmon/sm $NFS_FLM_HOLDING_DIR  
        sleep $PROPAGATE_INTERVAL  
    done  
fi
```

*** End of nfs.flm script ***

12-4. SLIDE: The nfs.cntl and hanfs.sh Scripts



Student Notes

The slide shows the major activities performed by the NFS package.

- Activates the volume group or volume groups associated with the package.
- Mounts each file system associated with the package.
- Initiates the NFS monitor script to check periodically on the health of NFS services, if you have configured your NFS package to use the monitor script.
- Exports each file system associated with the package so that it can later be NFS-mounted by clients.
- Assigns a package IP address to the LAN card on the current node.

After this sequence, the NFS server is active, and clients can NFS-mount the exported file systems associated with the package.

12-5. SLIDE: Using the NFS Toolkit



Using the NFS Toolkit

1. Ensure the `/etc/rc.config.d/nfsconf` file on all nodes is configured to start the NFS daemons
2. Create a directory for the new nfs package
3. Copy the files from the toolkit to the nfs package directory
4. Run `cmmakepkg` to create the `nfs.conf` and `nfs.cntl` files
5. Edit `nfs.conf`
 - use `nfs_service` as the `SERVICE_NAME`
6. Edit `nfs.cntl`
 - specify volume group, file system names
 - supply package IP/subnet
7. Edit the `hanfs.sh` file
 - Specify which file systems are to be exported
8. Run `cmcheckconf` and `cmapplyconf` to add the package

Student Notes

Follow these procedures to configure Highly Available NFS

1. On the primary node and all adoptive nodes for the NFS package, set the `NFS_SERVER` and `START_MOUNTD` variables to 1 in the `/etc/rc.config.d/nfsconf` file.

Do not configure the exported directories in the `/etc/exports` file.

2. Use the `mkdir` command to create a directory for the nfs package. In our example we'll call the directory `/etc/cmcluster/nfs`:

```
# mkdir /etc/cmcluster/nfs
```

3. Copy the files from the NFS toolkit directory to the new NFS package directory:

```
# cp /opt/cmcluster/nfs/* /etc/cmcluster/nfs
```

4. Use `cmmakepkg` to create the package control file and run/halt script:

```
# cd /etc/cmcluster/nfs

# cmmakepkg -p nfs.conf

# cmmakepkg -s nfs.cntl
```

5. Make the following changes to the `nfs.conf` file created by `cmmakepkg`:

```
PACKAGE_NAME      <name of your nfs package>
NODE_NAME          <primary node name>
NODE_NAME          <alternate node name>
RUN_SCRIPT         /etc/cmcluster/nfs/nfs.cntl
HALT_SCRIPT        /etc/cmcluster/nfs/nfs.cntl
SERVICE_NAME      nfs_service
SUBNET             <subnet to be monitored>
```

6. Make the following changes to the `nfs.cntl` file created by `cmmakepkg`:

```
VG[0]=<vgname>
LV[0]=<lvname>; FS[0]=<mount_point>; FS_MOUNT_OPT[0]=<options>
IP[0]=<pkg ip address>
SUBNET[0]=<subnet address>
```

Note, if you wish to have more than one highly available nfs package you can use the variable `HA_NFS_SCRIPT_EXTENSION=<suffix to append to hanfs.sh>`. For example, in your first nfs package you might set the variable to `pkg1.sh`. The `hanfs.sh` script for this package would then be called `hanfs.pkg1.sh`.

7. Edit the `hanfs.sh` script:

```
XFS[0]=<mount point of file system to export>
NFS_SERVICE_NAME[0]="nfs_service"
NFS_SERVICE_CMD[0]="/etc/cmcluster/nfs/nfs.mon"
NFS_SERVICE_RESTART[0]="-r 0"
```

You do not have to run the NFS monitor script. If your NFS package configuration file specifies `AUTO_RUN YES` and `LOCAL_LAN_FAILOVER_ALLOWED YES` (the defaults), the package switches to the next adoptive node or to a standby network interface in the event of a node or network failure. The NFS monitor script causes the package failover if any of the monitored NFS services fails. If you do not want to run the NFS monitor script, comment out the `NFS_SERVICE_NAME` and `NFS_SERVICE_CMD` variables

8. Copy the files in the `/etc/cmcluster/nfs` directory to the other nodes in the cluster. This can be done with `rcp` or `ftp`. Make sure all the scripts are executable after they have been copied.
9. Run `cmcheckconf` and `cmapplyconf` to update the cluster binary.

12-6. SLIDE: NFS Script Variables



NFS Script Variables

The “hanfs.sh” script contains several important variables:

- `XFS[ ]` These variables allow exported file systems to be defined and managed through the package control script instead of using the `/etc/exports` file.

Example: `XFS[0]="/data"`

Example: `XFS[1]="/tools"`
- `NFS_SERVICE_NAME` These variables define the service `nfs.mon`.
- `NFS_SERVICE_CMD` Because of these variables, no other
- `NFS_SERVICE_RESTART` services need to be defined through the standard `SERVICE` variables.

Note: Remember to uncomment the above lines.

Student Notes

A number of shell variables tailored specifically for the NFS application are included in the `hanfs.sh` package script.

`XFS[0]` These variables specify the names of the highly available NFS file systems. These file systems are exported when the control script starts the package, and unexported when the control script halts the package. If NFS export options are needed, you may want to add an array for those options. For example if you wish to export with the read only option use `XFS[0]=" -o ro /sgdata1"`.

`NFS_SERVICE_NAME` This variable, along with `NFS_SERVICE_CMD` and `NFS_SERVICE_RESTART`, defines the `nfs.mon` monitoring script as a service process. If an additional service process is needed (not expected), it would be defined using the standard `SERVICE_NAME` variable.

12-7. LAB: The nfs Package

Directions

Configure the cluster to contain a highly available NFS package. Use the NFS toolkit when configuring the package.

NOTE: The volume group names, disk device files, and volume group minor numbers must be localized for your system. This example uses an IP address and subnet. Your instructor will supply you with the correct ones to use.

Part 1. Create a New Volume Group

1. First, on your cluster's primary node, and using two disks, create a new shared volume group named `vg02`, with a 200-MB, single-mirrored, logical volume named `sglvol2`, and build a `vxfs` file system in the logical volume. Then mount the file system at mount point `/sgdata2` and copy some files into it.

Answer:

```
# pvcreate [ -f ] /dev/rdisk/c_t_d_  
# pvcreate [ -f ] /dev/rdisk/c_t_d_  
# mkdir /dev/vg02  
# mkknod /dev/vg02/group c 64 0x020000  
# vgcreate vg02 /dev/dsk/c_t_d_ /dev/dsk/c_t_d_  
# lvcreate -L 200 -m 1 -n sglvol2 vg02  
# newfs -F vxfs /dev/vg02/rsglvol2  
# mkdir /sgdata2  
# mount -F vxfs /dev/vg02/sglvol2 /sgdata2  
# cp /sbin/lv* /sgdata2
```

2. Now, import this volume group to the other node in your cluster.

Answer:

On the primary node:

```
# vexport -p -s -m /tmp/vg02.map vg02
```

On the adoptive node:

```
# cd /tmp  
# rcp <primary_node>:/tmp/vg02.map .  
# mkdir /dev/vg02  
# mkknod /dev/vg02/group c 64 0x020000  
# vgimport -s -m /tmp/vg02.map vg02  
# mkdir /sgdata2  
# vgchange -a r vg02  
# vgcfgbackup vg02  
# vgchange -a n vg02
```

3. Back on the primary node, unmount the file system and deactivate the volume group.

Answer:

```
# umount /sgdata2  
# vgchange -a n vg02
```

4. While still on the primary node, mark the vg02 volume group as belonging to the Serviceguard cluster.

Answer:

```
# vgchange -c y vg02
```

Part 2. Configure the NFS Package

1. Ask your instructor for an IP address. Assign your NFS package that assigned IP address. Update the /etc/hosts files on **both systems** with the package name and IP address.

Answer:

```
# vi /etc/hosts
```

2. On your primary node, in the /etc/cmcluster directory, create a directory called nfs for the NFS package named nfs. Copy the NFS toolkit files from /opt/cmcluster/nfs to your new directory.

Answer:

```
# mkdir /etc/cmcluster/nfs  
# cd /etc/cmcluster/nfs  
# cp /opt/cmcluster/nfs/* .
```


3. Use `cmmakepkg` to create the package configuration file.

Answer:

```
# cd /etc/cmcluster/nfs
# cmmakepkg -v -p nfs.conf
```

4. Edit `nfs.conf` as follows:

```
# vi nfs.conf
```

PACKAGE_NAME	nfs
NODE_NAME	<node1>
NODE_NAME	<node2>
RUN_SCRIPT	/etc/cmcluster/nfs/nfs.cntl
HALT_SCRIPT	/etc/cmcluster/nfs/nfs.cntl
SERVICE_NAME	nfs_service
SUBNET	10.x.x.x ← Use <code>netstat -in</code> to obtain cluster 10-net subnet

5. Using `cmmakepkg` , create the package control script named `nfs.cntl`.

Answer:

```
# cmmakepkg -v -s nfs.cntl
```

6. Edit the `nfs.cntl` control script to customize it for your volume group, mount point, and the IP address you were assigned for this package.

```
# vi nfs.cntl
```

and make the following changes:

```
VG[0]="vg02"
```

In the logical volumes and file systems section, break up this line into separate lines, as we did in Module 11, and make the following changes:

LV[0]="/dev/vg02/sglvol2"	← Note this value
FS[0]="/sgdata2"	← Note this value
FS_MOUNT_OPT[0]=""	← Leave this value undefined
FS_UMOUNT_OPT[0]=""	← Leave this value undefined
FS_FSCK_OPT[0]=""	← Leave this value undefined
FS_TYPE[0]="vxfs"	← Note this value

Finally, in the IP addresses section, make the following changes:

IP[0]=10.x.x.x	← Use your assigned 10-net IP address here
SUBNET[0]=10.x.x.0	← Use netstat -in to obtain cluster 10-net subnet

7. The last file to edit is the `hanfs.sh` file. Edit this file and make the following changes:
- a. First, please go to line 67, which currently reads ...

```
#XFS[0]=""
```

and change this line to now read ...

```
XFS[0]="/sgdata2" ← Do not forget to uncomment this line
```

- b. Next, go to line 99 and following, which now reads ...

```
#NFS_SERVICE_NAME[0]=""  
#NFS_SERVICE_CMD[0]=""  
#NFS_SERVICE_RESTART[0]=""
```

and change these lines to read ...

```
NFS_SERVICE_NAME[0]="nfs_service" ← Do not forget  
NFS_SERVICE_CMD[0]="/etc/cmcluster/nfs/nfs.mon" ← to uncomment  
NFS_SERVICE_RESTART[0]="-r 0" ← these lines.
```

- c. Save your work and exit `vi`.
8. Ensure that the scripts in `/etc/cmcluster/nfs` are executable.

Answer:

```
# chmod 744 nfs.*
```

9. Next, create the `/etc/cmcluster/nfs` directory on the other system. Then, copy all these NFS files to the `nfs` subdirectory on the other system.

Answer:

```
# remsh <node2> "/usr/bin/mkdir /etc/cmcluster/nfs"  
# rcp * <node2>:/etc/cmcluster/nfs/.
```

10. Now, back on your cluster's primary node, add your newly created volume group to your cluster-wide ASCII file.

Answer:

```
# cd /etc/cmcluster
# vi cmclconfig.ascii
```

11. Execute `cmcheckconf` against both your NFS package conf file **and** the modified cluster-wide ASCII configuration file.

Answer:

```
# cd /etc/cmcluster/nfs
# cmcheckconf -v -P nfs.conf -C ../cmclconfig.ascii
```

12. Update the cluster binary. You must do a `cmapplyconf` to set the volume group for Serviceguard exclusive activation.

Answer:

```
# cmapplyconf -v -P nfs.conf -C ../cmclconfig.ascii
```

13. Start the NFS package.

Answer:

```
# cmmodpkg -v -e nfs
```

14. Verify that the NFS package volume group is active, and the `/sgdata2` file system is mounted and exported on the node where the `nfs` package is running. If not, look at the package log file, correct any errors, and try again.

Answer:

```
# vgdisplay vg02
# bdf
# exportfs -v
```

15. At this point, it is very helpful to perform several failovers. We want to make **absolutely sure** that the NFS package is behaving properly. Use a series of commands to test your failover such as ...
- Kill the `nfs.mon` process. The package should fail over. Re-enable the node.
 - On the other node (where the `nfs` package should now be running), kill the `rpc.mountd` daemon. Do a `tail -f` on the `/etc/cmcluster/nfs/nfs.cntl.log` file on the node where the package was running. Eventually the package should failover. Restart `rpc.mountd` by issuing the command: `/usr/sbin/rpc.mountd`. Re-enable the node
 - Halt, then restart the package

When you are convinced that the package is running and failing over properly, continue with step 16.

16. Move to a spare machine (*or use whatever machine your instructor suggests - this machine must be on the 10.x.x.x network*) and create a directory called `/your_name_nfstest` to be used as an NFS mount point.

Answer:

```
# mkdir /"your_name"_nfstest
```

17. Now, on this same spare machine, mount the file system from the NFS package

Answer:

```
# mount -F nfs <ip_address_of_package>:/sgdata2 /"your_name"_nfstest
```

18. Continuing on the spare machine, open a terminal window and write the following short "while true" loop to simulate access to the NFS package file system.

```
# while true
> do
> clear
> sleep 1
> ll /your_name_nfstest
> sleep 1
> done
```

19. Next, return to your Serviceguard cluster, and on the node on which your `nfs` package is running, and while the above `"while true"` loop is still running, kill the `nfs.mon` monitoring script. However, to understand this simulation, read the following paragraph before killing the `nfs.mon` process.

After you kill the `nfs.mon` process, watch carefully what happens to the `"while"` loop on the spare machine. You should see the `"ll"` command hang (in the `"while true"` function in step 18 above), and then, depending on hardware, the failover should normally require less than 10 seconds. After the package fails over, you should see the `"while true"` loop pick up again with continuing access to the NFS-mounted file system!

Answer:

```
# ps -ef | grep nfs.mon    ← On the machine on which the package is running
# kill <pid>              ← On the machine on which the package is running
```

20. Re-enable the switches for failover, and then try the package failover a few more times. Each time, please watch what happens to the `"while true"` loop on the spare machine.

Answer:

```
# cmmmodpkg -v -e -n <failed_node> nfs
# ps -ef | grep nfs.mon    ← On the machine on which the package is running
# kill <pid>              ← On the machine on which the package is running
```

21. Congratulations on being successful with your first package that uses all three of the following major components of a package:

- Disk resources
- Network resources
- Service Processes

12-8. LAB: NFS Package Performance

Directions

In this lab, we modify the NFS package so that the package is able to utilize as much of various system resources as possible. You can do this with two approaches:

- **Approach 1:** Shut down some other noncritical processes running on the system, or
- **Approach 2:** Shut down another package and move the specified package to another system automatically.

Approach 1: Shut Down Noncritical Processes (sendmail)

Special Note

In the following example, we kill the sendmail process. Of course, sendmail has nothing to do with Serviceguard, but we use it here to demonstrate how non-critical processes on a node can be shut down when a package fails over to that node. This would be useful where the NFS package is mission critical, and must have as much CPU time as it needs.

1. Halt the nfs package and edit its control script as follows:

```
# cmhaltpkg -v nfs
# vi /etc/cmcluster/nfs/nfs.cntl
```

```
function customer_defined_run_cmds
{
    LOG="/etc/cmcluster/nfs/nfs.cntl.log"
    #
    # shutdown sendmail using the rc script
    #
    /sbin/init.d/sendmail stop
    echo "Sendmail daemon stopped by nfs_server \c"          >> $LOG
    echo "package on $(hostname) at $(date)."                 >> $LOG
    test_return 51
}
```

```
function customer_defined_halt_cmds
{
    LOG="/etc/cmcluster/nfs/nfs.cntl.log"
    #
    # startup sendmail using the rc script
    #
    /sbin/init.d/sendmail start
    echo "Sendmail daemon started by nfs_server \c"          >> $LOG
    echo "package on $(hostname) at $(date)."                 >> $LOG
    test_return 52
}
```

2. Next, perform the steps to get the modified nfs package running again.

```
# rcp /etc/cmcluster/nfs/nfs.cntl <node2>:/etc/cmcluster/nfs
# cmmodpkg -v -e nfs
# cmviewcl -v
```

3. On the node where the nfs package is running, is sendmail also running? Remember that if our package is running properly, sendmail should not be running.

Answer:

```
# ps -ef | grep sendmail
```

4. Move the `nfs` package to the other node.

Answer:

```
# cmhaltpkg -v nfs  
# cmrunpkg -v -n <other_node> nfs  
# cmmodpkg -v -e nfs
```

5. Is `sendmail` running on the node where the `nfs` package is running? Did `sendmail` come back on the other system? Check the NFS log file for the messages sent by the `echo` commands from our **run/halt** script.

Answer:

The `sendmail` package should not be running on the same node as the `nfs` package.
The `sendmail` package should have returned to the other node.

Approach 2: Shut Down the "test" Package When the "nfs" Package Moves to a Node Where the "test" Package is Currently Running

Note

In the following example, we deliberately fail over the test package. We do this to free up as much CPU time as possible for the nfs package. This would be useful where the nfs package is mission critical but the test package is not.

This approach will require you to put some additional logic in the function `customer_defined_run_cmds` function to stop the test package.

The sample solution will reflect only the information needed to move the test package. If you have other commands in the function, that is acceptable also. The sample solution will use `node1` and `node2` as your node names. Of course, you would use your specific hostnames.

1. Halt the nfs package and edit the `nfs.cnt1` script. Remember to insert the correct hostnames where the script calls for `node1` and `node2`.

```
# cmhaltpkg -v nfs
# vi /etc/cmcluster/nfs/nfs.cnt1
```

```
function customer_defined_run_cmds
{
    # move test package to other system
    cmhaltpkg -v test
    case $(hostname) in
        node1) cmmodpkg -v -e -n <node2> test ;;
        node2) cmmodpkg -v -e -n <node1> test ;;
    esac
    cmmodpkg -v -d -n $(hostname) test
    cmmodpkg -v -e test
    test_return 51
}
```

2. Be sure to copy the modified control script to the other node in your cluster.

Answer:

```
# rcp /etc/cmcluster/nfs/nfs.cnt1 <node2>:/etc/cmcluster/nfs
```

3. Start the nfs package again.

Answer:

```
# cmmodpkg -v -e nfs
```

4. On the node where the nfs package is running, is the test package also running?

Answer:

```
# cmviewcl -v -p test
```

5. Move the nfs package to the other node.

Answer:

```
# cmhaltpkg -v nfs  
# cmrunpkg -v -n <other_node> nfs  
# cmmodpkg -v -e nfs
```

6. Is the test package running on the node where the nfs package is running? Did the test package come back on the other system?

Answer:

```
# cmviewcl -v -p test
```

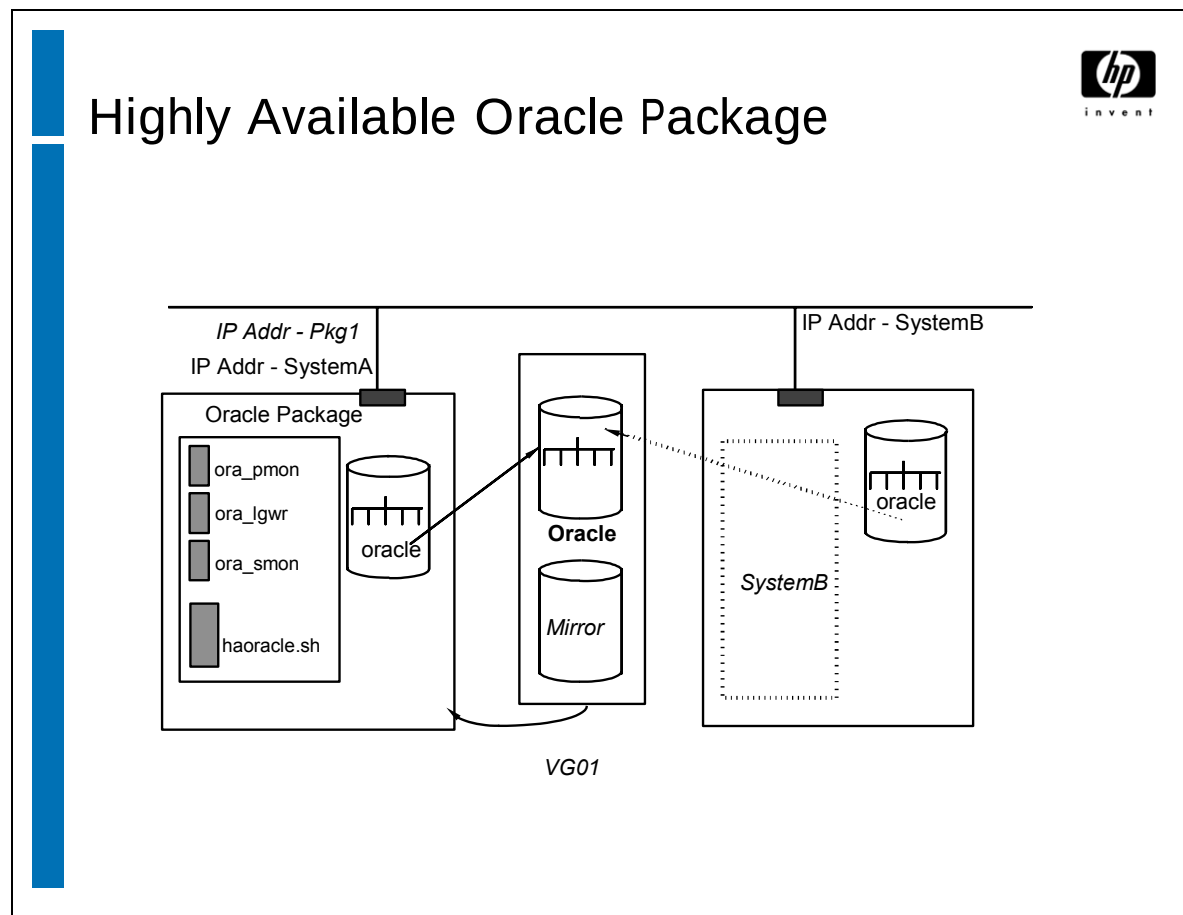
Module 13 — The Highly Available Oracle Database

Objectives

Upon completion of this module, you will be able to do the following:

- List different Serviceguard toolkits.
- Use Oracle toolkit and cookbook.
- Configure the Oracle 10g Database to run in a Serviceguard environment.

13-1. SLIDE: Highly Available Oracle Package – Overview



Student Notes

One of the most common applications for a high availability environment is Oracle.

The following are recommendations for configuring Oracle for a Serviceguard environment:

- Install the Oracle product (libraries, binaries, commands) on all nodes of the cluster, as opposed to a single installation on a shared file system. By keeping the Oracle executable on each system rather than having one copy of the executables on a shared file system, there is less file system recovery (fsck) during a failover. This will help speed up the amount of time needed to failover the application.
- By having the executables on each system, you also can either patch or update the executable on one system and move the package to the updated system to test the new version of Oracle. However applying a major patch to the Oracle Software implies a 'data dictionary' change, which causes a full outage due to the fact that several scripts need to run against the database without user connected.
- Create a shared file system to hold the Oracle configuration files (init.ora, spfile, listener.ora, tnsnames.ora, sqlnet.ora).

- Create a shared file system to hold the archive log files.
- Create the Oracle database (including control files and redo logs) on a shared volume group. All clustered nodes will potentially need access to the database depending on where the package is located. A shared database with locally installed binaries is the recommended configuration.
- Oracle clients connect to the database via the package IP address, not the IP address of a host. Upon application failover, Oracle clients will need to reconnect using the same IP address. This is because Oracle uses TCP (as opposed to UDP), which is a connection-based protocol.
- An Oracle monitoring script will be needed to monitor the health of the Oracle application. HP offers an Oracle monitoring script as part of the database toolkit product.

13-2. SLIDE: Serviceguard Toolkits



Serviceguard Toolkits

- Enterprise Cluster Master Toolkit (B5139DA)

Custom monitoring scripts for:

- Oracle
- Apache
- Samba
- Tomcat

Student Notes

The toolkits shown on the slide have been developed to simplify the set up of packages related to NFS and a few other applications. Some applications cannot be started as service processes; therefore, they require an application monitoring script.

The primary value of the toolkits is the inclusion of an application monitoring script for each application. This eliminates the need for the customer to code a monitoring script from scratch.

The monitoring scripts included with the toolkits require minor modifications to customize the script for the customer's application.

The Enterprise Cluster Master Toolkit is intended to contain all future scripts for ease of ordering.

This toolkit is already loaded onto the system when one installs HP-UX 11i operating system with the "Mission Critical" operating environment. To see the toolkit files, look in the `/opt/cmcluster/toolkit/oracle` directory.

13-3. SLIDE: Database Toolkit Contents



Database Toolkit Contents

Oracle Toolkit Files

- README
- haoracle.mon and more
- General information and setup procedure
- Monitoring script for Oracle

Apache Toolkit Files

- README
- hahttp.mon and more
- General information and setup procedure
- Monitoring script for Apache

Samba Toolkit Files

- README
- hasmb.sh and more
- General information and setup procedure
- Monitoring script for Samba

Tomcat Toolkit Files

- README
- hatomcat.sh and more
- General information and setup procedure
- Monitoring script for Tomcat

Student Notes

There are four different database products for which toolkits currently exist.

The content of each toolkit is:

README file

This file contains configuration information for monitoring in a Serviceguard environment.

Monitoring script

This script is designed to monitor critical processes related to the specific application. This script must be edited for your specific application before it can be used. Other scripts are also included in the toolkit for each of the above applications.

13-4. SLIDE: Example: Oracle Application



Example: Oracle Application

1. Activate Volume Group
2. File system Check and Mount
3. Add IP Address
4. Customer Defined Run Commands
Start oracle database here

ora_dbwr

ora_smon

ora_lgwr

ora_pmon
5. Start Oracle Monitoring Process (aka a service process)

haoracle.mon

-checks to see if critical oracle daemons are running
If daemons are OK, sleep for 10 seconds and check again.
If daemons are NOT OK, try to fix. If cannot be fixed, exit.

Student Notes

The slide shows the five major activities performed by the package control script when starting an Oracle database application.

1. Activate the volume group containing the Oracle database.
2. Check and mount any file systems related to the database.
3. Add the IP address for the Oracle package.
4. Execute any commands in the `customer_defined_run_cmds` function. This function is where commands to start the Oracle application are added.
5. Start the Oracle monitoring script to monitor the Oracle application started in step 4.

Before attempting to write your own monitoring script, check to see if a monitoring script already exists for your application.

13-5. SLIDE: Oracle Control Script Modification



Oracle Control Script Modification

- `SERVICE_NAME[0]=` Use these variables to call the
- `SERVICE_CMD[0]=` `haoracle.sh` script with the
- `SERVICE_RESTART[0]=` monitor parameter.

- `customer_defined_run_cmds{` Use this function to start the Oracle DB
 `haoracle.sh start` instance. Use the `haoracle.sh` script
 `}` with the start parameter.
- `customer_defined_halt_cmds{` Use this function to stop the Oracle DB
 `haoracle.sh shutdown` instance. Use the `haoracle.sh` script
 `}` with the shutdown parameter.

Student Notes

The `haoracle.sh` script is used for *three* purposes. It is used to:

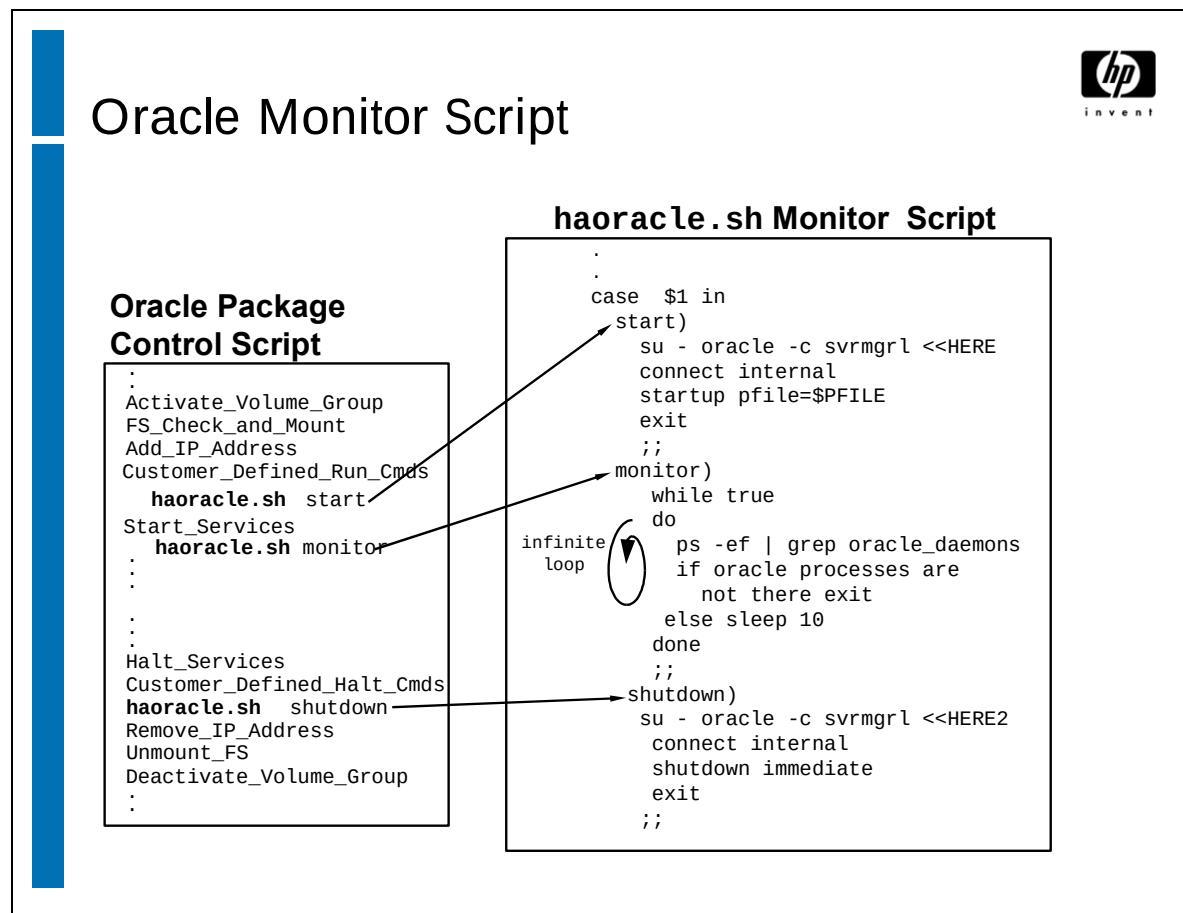
- Monitor the database (defined in the `SERVICE_CMD` variable).
- Start the database (defined in the `customer_defined_run_cmds` function).
- Bring down the database (defined in the `customer_defined_halt_cmds` function).

```
SERVICE_NAME[0]=oracle_class
SERVICE_CMD[0]="/etc/cmcluster/oracle/haoracle.sh monitor"
```

```
function customer_defined_run_cmds {
    /etc/cmcluster/oracle/haoracle.sh start
    test_return 51
}
```

```
function customer_defined_halt_cmds {
    /etc/cmcluster/oracle/haoracle.sh shutdown
    test_return 52
}
```

13-6. SLIDE: Oracle Monitor Script



Student Notes

The Oracle start/stop/monitoring script (`haoracle.sh`) provided in the database toolkit calls other scripts provided. With this toolkit version the start/stop/abort commands are located in `haoracle_sql.sh` and the monitoring is performed by `haoracle.mon`.

Moving the start up and shutdown of Oracle to the toolkit script guarantees the correct Oracle instance will be started/stopped and monitored.

This means the Oracle **control** script does the following:

- Activates the volume group(s)
- Checks and mounts the file systems
- Adds the package IP address
- Spawns the Oracle monitoring script as the service process

The Oracle **monitor** script monitors the Oracle database daemons.

13-7. SLIDE: Oracle Cookbook



Oracle Cookbook

1. Create directory and shared LV for Oracle database specific files.
2. `su - oracle`
3. `svrmgrl` (Set up datafiles ,logfiles, database, tablespace)
4. `mkdir /etc/cmcluster/oracle/`
5. `cd /etc/cmcluster/oracle`
6. `cmmakepkg -v -p oracle.conf` and `vi oracle.conf`
7. `cmmakepkg -v -s oracle.cntl` and `vi oracle.cntl`
8. `cp /opt/cmcluster/toolkit/oracle/haoracle.sh .`
9. `vi haoracle.sh`
10. `cmcheckconf` and `cmapplyconf` and `cmrunc`
11. Start the Oracle package and test database failover.

Student Notes

The slide provides a high-level overview of the steps for configuring Oracle as a package within the Serviceguard environment.

Create the Subdirectory for the Oracle Instance (Step1)

Each Oracle instance must have its own subdirectory **structure** that contains all the configuration files, binaries, libraries, commands, and any other files that the **instance** might require.

Rather than reinstall these files for each instance, symbolically link to an already existing installation.

Create the Oracle Database, redo Logs, and Tablespace (Steps 2 and 3)

Once the instance files have been created (or linked), the database and other related files and tables need to be created. These items must be created by the Oracle account.

Create the Package Configuration File and Control Script (Steps 4 - 7)

To package the Oracle application for Serviceguard, a package configuration file and a package control script will need to be created using the standard tool, **cmmakepkg**. Both files will require customization related to the specific Oracle instance.

Create the Oracle Start / Stop and Monitoring Script (Steps 8 and 9)

The Oracle monitoring script that is included with the database toolkit product will need to be customized to be consistent with the specific Oracle instance.

The package control script and the Oracle monitoring script need to be copied to all nodes within the cluster.

Compile and Distribute the Binary File (Step 10)

The final steps are to rebuild the binary file using the standard **cmcheckconf** and **cmapplyconf** commands and then to bring up the cluster.

Verify Cluster and Package Status (Step 11)

The **cmviewcl** command verifies the Oracle package is up and running properly. If the Oracle package did not come up, the package control script log file, the monitor script log file, and the **syslog.log** files should be checked for errors.

13–8. LAB: Oracle Package Configuration

Introduction

1. This lab will setup an Oracle database to work with Serviceguard.
2. The volume group names, disk device files, and volume group minor numbers must be localized for your system. This example uses IP addresses and subnets. Your instructor will supply you with the correct ones to use.
3. This lab is structured to minimize the Oracle knowledge you need to know. The lab creates a database using some shell scripts that have been placed on your system in the /labs/mod_13 subdirectory.
4. This lab reinforces where the 3 components of setup are needed.
 - Basic system administration setup and network configuration
 - Creation of the Oracle database
 - Packaging the Oracle database to work with Serviceguard
5. These systems were setup with a user named "oracle". The Oracle executables are loaded in the /ora1/oracle/product/10.1.0.db directory on **both systems**.
6. One objective of the lab is to create an environment where several instances of Oracle could run concurrently. This would allow one instance to move while leaving the other Oracle instances alone.
7. A second objective is to share the Oracle executables among all instances on a system. Each instance will be symbolically linked to the binaries under the subdirectory ...

/opt/oracle/product/10.1.0.db

A third and final objective is to allow each instance to run on any node in the cluster. To do this, each instance will need to be configured on its own volume group and file system.

NOTE:	This course creates <i>one</i> Oracle instance. Creating multiple Oracle instances with Serviceguard is covered in “Disaster Tolerant Architectures with Serviceguard, Oracle 9i and Oracle 10gRAC” course, HP course number U8601S.
--------------	--

NOTE:	In this course, due to the manner in which the lab setup was done, and the hard-coding of some variables, only <i>one</i> Oracle package is possible per Serviceguard cluster.
--------------	--

Special Notes for the Whole Lab

1. In the following steps, Node1 will always refer to the primary node for your Oracle package, and Node2 will always refer to the adoptive or alternate node.
2. It is **vitaly important** that you execute each step in this lab on the correct node!
3. Please complete all parts of this lab entirely on your cluster's primary node unless otherwise specifically indicated.
4. It is also **vitaly important** that you execute each step as the correct user, either the root user or the oracle user. Pay special attention to this!

Part 1: Prepare Oracle For an Oracle 10g Environment

Important Special Note for Part 1 ONLY

When executing the script below, it is **critically important** to do so using an ASCII terminal. Remote users should establish a Putty (or similar) session to their system. Do not use ReflectionX (or any other graphical terminal emulator as the execution time of the following script will be much longer. When you use an ASCII terminal, the script should execute in as little as 4 minutes. Using a GUI such as ReflectionX will mean that the execution time will be MUCH greater!

1. To ensure success later in this lab, execute the following script on **both nodes**. In an ASCII terminal, this script should require about 4 or 5 minutes to execute.

```
<Node1> # /labs/mod_13/oracle_10g.sh
```

```
<Node2> # /labs/mod_13/oracle_10g.sh
```

While this script is executing, and in another window, continue with Part 2 of the lab.

Part 2: Set Up the LVM and File System Structures for the Oracle Database

1. **[Node1/root]** We need to have two disks still "unused". Therefore, it may be necessary for you to first `cmhaltpkg` one or more packages and/or `vgexport` one or more volume groups in order to accomplish this. So, after freeing up two disks (*if needed*), create a shared volume group named `vg03`.

```
# mkdir /dev/vg03
# mkknod /dev/vg03/group c 64 0x030000
# pvcreate [ -f ] /dev/rdisk/c_t_d_
# pvcreate [ -f ] /dev/rdisk/c_t_d_
# vgcreate vg03 /dev/dsk/c_t_d_ /dev/dsk/c_t_d_
# vgchange -a n vg03
# vgchange -c y vg03
# vgchange -a e vg03
```

2. **[Node1/root]** Next, create and mirror the Oracle logical volume named `oracle1`. Also, note that the `oracle1` logical volume ends with a numeral one, not the letter "ell".

```
# lvcreate -L 3000 -m 1 -n oracle1 vg03
```

3. **[Node1/root]** In the `/dev/vg03/oracle1` logical volume, create a **JFS** file system. Also, create a mount point at `/oracle1`. Mount the `oracle1` logical volume at the `/oracle1` mount point. Again, note that the `oracle1` mount point ends with a numeral *one*, not the letter "ell".

```
# newfs -F vxfs /dev/vg03/oracle1
# mkdir /oracle1
# mount -F vxfs /dev/vg03/oracle1 /oracle1
```

4. **[Node1/root]** Create the following directory structure to hold the Oracle database by executing the following script:

```
# /labs/mod_13/dirbuild.sh
```

This script contains the following commands:

```
# mkdir /oracle1/admin
# mkdir /oracle1/admin/bdump
# mkdir /oracle1/admin/cdump
# mkdir /oracle1/admin/udump
# mkdir /oracle1/admin/pfile
# mkdir /oracle1/admin/network
# mkdir /oracle1/oraredo
# mkdir /oracle1/oradata
# mkdir /oracle1/oraarch
# chown -R oracle:dba /oracle1
# chmod -R 755 /oracle1
```

5. **[Node1/root]** Next, create a map file for your new vg03 volume group and transfer the map file to <node2>:

```
# vgexport -v -p -s -m /tmp/vg03.map vg03
# rcp /tmp/vg03.map <node2>:/tmp/vg03.map
```

6. **[Node2/root]** Now, on <node2>, import the new vg03 definition.

```
# mkdir /dev/vg03
# mknod /dev/vg03/group c 64 0x030000
# vgimport -v -s -m /tmp/vg03.map /dev/vg03
```

Note: If warnings appear regarding the volume group belonging to a different CPU ID, you can ignore them.

```
# vgchange -a r vg03      ← Activate vg03 in read-only mode
# vgcfgbackup vg03        ← Create a backup copy of the LVM structures
# vgchange -a n vg03      ← Deactivate volume group vg03
```


Part 3: Create the Oracle Database

1. **[Node1/root]** Copy Oracle's .profile, the Oracle network file (listener.ora) and create the ORACLE_BASE directory. Set the appropriate ownership.

```
# cp /labs/mod_13/profile.oracle /home/oracle/.profile
# cp /labs/mod_13/listener.ora /oracle1/admin/network
# mkdir -p /opt/oracle/product
# chown oracle:dba /opt/oracle/product
# chown oracle:dba /oracle1/admin/network
# chown oracle:dba /home/oracle/.profile
```

2. **[Node2/root]** Copy Oracle's .profile and create the ORACLE_BASE directory.

```
# cp /labs/mod_13/profile.oracle /home/oracle/.profile
# mkdir -p /opt/oracle/product
# chown oracle:dba /opt/oracle/product
# chown oracle:dba /home/oracle/.profile
```

3. **[Node1/oracle]** Copy the default configuration file to the oracle1 instance directory and link the pre-installed Oracle software to the ORACLE_HOME used by the database.

```
# su - oracle
$ cp /labs/mod_13/initORACLE1.ascii /oracle1/admin/pfile
$ ln -sf /ora1/oracle/product/10.1.0.db /opt/oracle/product
$ exit
```

4. **[Node2/oracle]** Link the pre-installed Oracle software to ORACLE_HOME used by the database.

```
# su - oracle
$ ln -sf /ora1/oracle/product/10.1.0.db /opt/oracle/product
$ exit
```

5. **[Node1/oracle]** Back on **<node1>**, su to the oracle account, and verify that profile changes have taken effect. You can do this by checking the value of ORACLE_SID.

```
# su - oracle
$ echo $ORACLE_SID          ← Verify the ORACLE_SID variable is ORA1SID
```

Important Special Notes for Step 6

1. Note that we do not exit the oracle user account. Therefore, while still logged in as the oracle user, please continue with step 6 below.
2. Most importantly, when executing the create_database.sh script (see step 6 below), it is **critically important** to do so using an ASCII terminal. Remote users should use Putty to system (or a similar terminal emulator). Do NOT use ReflectionX (or any other graphical terminal emulator). When you use an ASCII terminal, the script should execute in a little as 22 minutes. Using a graphical terminal emulator will mean that the execution time will be measured in hours!

6. **[Node1/oracle]** Next, **still on the primary node**, and **still as the oracle user**, execute the /labs/mod_13/create/create_database.sh script. Due to the way Oracle creates views, by trying to remove the view before creating it, many errors occur because the view name does not exist. These error messages are to be expected and should be ignored. Allow 30-40 minutes for this script to complete its execution.

```
$ /labs/mod_13/create/create_database.sh
```

This create_database.sh script contains the following code:

```
#####
#
# Create Database
#
# Configuration parameters are set
# in DB.conf
#
# Last Modification: 05/05/2006
#
#####
cd /labs/mod_13/create
sqlplus "/" as sysdba" @00_Main.sql
```

The script simply calls an sql script. All configuration is made in the DB.conf file.

The 00_Main.sql script contains the following code:

```

-----
--
-- Main script to create an Oracle 10g database
--
-- Last modification: 27/05/2005
--
-----
@DB.conf
-----
-- Create spool directory (if not already created) and delete logfiles of
-- previous database creation
host mkdir -p &&SPoolDIR
host rm &&SPoolDIR/* 2>/dev/null
--
host echo "Start database creation &&DBNAME at " `date` > &&SPoolDIR/time.log
--
spool &&SPoolDIR/sql.log append
--
-- Create password file to allow remote access as sysdba
host $ORACLE_HOME/bin/orapwd file=&&ORACLE_BASE/admin/pfile/orapw&&ORACLE_SID
password=&&sysPWD force=y
host ls -l &&ORACLE_BASE/admin/pfile/orapw&&ORACLE_SID
--
-- Link password file to $ORACLE_HOME/dbs
host ln -sf &&ORACLE_BASE/admin/pfile/orapw&&ORACLE_SID $ORACLE_HOME/dbs
--
-- Create init.ora
host echo "spfile='&&SPFILE'" > &&PFILE
host cat &&PFILE
--
-- Link init.ora to $ORACLE_HOME/dbs
host ln -sf &&PFILE $ORACLE_HOME/dbs/init&&ORACLE_SID..ora
host ls -l $ORACLE_HOME/dbs
--
-----
create
  spfile='&&SPFILE'
  from
  pfile='&&PFILE_ASCII'
  ;
-----
host ./10_CreatedB.sh 3 2 2 2 &&REDOLOGS &&ASM
@@10_CreatedB.sql
--
@?/rdbms/admin/catalog.sql;
@?/rdbms/admin/catblock.sql;
@?/rdbms/admin/catproc.sql;
@?/rdbms/admin/catotck.sql;
@?/rdbms/admin/owminst.plb;
@?/rdbms/admin/dbmspool.sql;
connect SYSTEM/&&systemPWD
@?/sqlplus/admin/pupbld.sql;
@?/sqlplus/admin/help/hlpbld.sql helpus.sql;
create table test (num number, name varchar2(15), city varchar2(20));
truncate table test;
insert into test values( 0, 'MJ','Chicago; IL');
insert into test values( 1, 'Magic','Los Angeles, CA');
insert into test values( 2, 'Bird','Boston, MA');
insert into test values( 3, 'Gary','Detroit, MI');
insert into test values( 4, 'Martin','Frankfurt, Germany');

```

```
insert into test values( 5, 'Vince','Toronto, Canada');
commit;
--
connect / as sysdba
@$ORACLE_HOME/javavm/install/initjvm.sql;
@$ORACLE_HOME/xdk/admin/initxml.sql;
@$ORACLE_HOME/xdk/admin/xmlja.sql;
@$ORACLE_HOME/rdbms/admin/catjava.sql;
@$ORACLE_HOME/rdbms/admin/catexf.sql;
--
host echo "End database creation &&DBNAME at " `date` >> &&SPoolDIR/time.log
select * from system.test;
shutdown immediate
spool off
exit;
```

The DB.conf script contains the following code:

```
--
#####
--
-- Name: DB.conf
--
-- Author: ML
--
-- Creation date: 01/05/2004
--
-- Description: Oracle Environment & DB creation parameters
--
-- This file contains all parameters to run the DB-create scripts.
-- It needs to be included in each script.
--
-- Directories:
-- -----
-- SPOOLDIR          : Script Output Directory
-- ORACLE_BASE       : e.g. /opt/oracle
-- ORACLE_HOME       : e.g. /oracle1/product/10.1.0
--
-- -----
--
-- Controfile Parameters:
-- -----
-- DBNAME            : Database Name (default service name)
-- INSTANCES         : No. of instances, that will be setup for this
--                   : database.
-- REDOLOGS          : No. of redolog files per thread
-- REDOLOG_SIZE      : Size of each redolog
-- MAXLOGFILES       : minimum: INSTANCES*REDOLOGS
-- MAXLOGHISTORY     : minimum: MAXLOGFILES*8
-- MAXLOGMEMBERS     : Max. No. of redo memebers in a single group
-- MAXDATAFILES      : Max. No. of datafiles,that can be used for the
--                   : database
-- CHARSET           : database default character set
-- NATIONAL_CHARSET  : national database character set
--
-- PFILE             : init${ORACLE_SID}.ora
-- PFILE_ASCII       : init&&DB_NAME..ascii
-- SPFILE            : spfile&&DB_NAME..ora
--
-- The database character set will be derived from NLS_LANG.
--
--
#####
--
set echo on
set termout on
--
define ORACLE_SID=$ORACLE_SID
--
-- Directories:
-- -----
define SPOOLDIR=/home/oracle/create_db/logs
define ORACLE_BASE=/oracle1
define ORACLE_HOME=/opt/oracle/product/10.1.0.db
--
-- Controfile Parameters:
-- -----
define DBNAME=ORACLE1
```

Module 13
The Highly Available Oracle Database

```
define MAXINSTANCES=2
define REDOLOGS=5
define MAXLOGFILES=32
define MAXLOGHISTORY=1024
define MAXLOGMEMBERS=2
define MAXDATAFILES=256
define CHARSET=WE8ISO8859P15
define NATIONAL_CHARSET=UTF8
--
-- File locations:
-- -----
define DATADIR=/oracle1/oradata
define REDODIR=/oracle1/oraredo
define ARCHDIR=/oracle1/oraarch
define TMPDIR=/oracle1/oradata
define UNDDIR=/oracle1/oradata
--
-- File sizes:
-- -----
define EXAMPLE_SIZE=256M
define LOGMNRTS_SIZE=256M
define REDO_SIZE=64M
--
--
define SYSTEM_SIZE=192M
define SYSAUX_SIZE=128M
define TEMP_SIZE=128M
define UNDO_SIZE=128M
--
-- Password definitions
-- -----
define sysPWD=change_on_install
define systemPWD=manager
define sysmanPWD=manager
--
-- ASM definitions
-- -----
-- To create a database using ASM define the following parameters.
-- If you DO NOT use ASM leave the ASM variable empty, otherwise put ASM.
define ASM=''
define DG_ASM_SYSTEM=DG1
define DG_ASM_SYSAUX=DG1
define DG_ASM_TEMP=DG1
define DG_ASM_UNDO=DG1
define DG_ASM_REDO=DG1
define DG_TOOLS=DG1
define DG_SPFILE=DG1
--
-- Parameterfile locations:
-- -----
define PFILE=&&ORACLE_BASE/admin/pfile/init${ORACLE_SID}.ora
define PFILE_ASCII=&&ORACLE_BASE/admin/pfile/init&&DBNAME..ascii
--
define SPFILE=&&ORACLE_BASE/admin/pfile/spfile&&DBNAME..ora
-- define SPFILE=+&&DG_SPFILE/spfile&&DBNAME..ora
-- -----
--
define ULTRASEARCH_HOST=$NODE1
define ULTRASEARCH_PORT=30005
define ULTRASEARCH_SERVICE=ULTRA
-- -----
--
set serveroutput on
```

```
set linesize 200
set trimspool on
set timing on
set time on
define _EDITOR=vi
-----
---
```

The 00_Main.sql script calls one other script, 10_CreateDB.sh.
The 10_CreateDB.sh script contains the following code:

```
# /usr/bin/ksh
# -----
#
# Script: 10_CreateDB.sh
#
# Last Modification:29/03/2005
#
# Generate DB creaesript based on variables passed:
#
# $1: No. of system datafiles
# $2: No. of sysaux datafiles
# $3: No. of temp datafiles
# $4: No. of undo datafiles
# $5: No. of redo logfiles
# $6: ASM (empty)
#
# -----
SYS=$1
AUX=$2
TMP=$3
UND=$4
RDO=$5
ASM=$6
#
echo "shutdown abort" >> 10_CreateDB.sql
echo "set echo on" >> 10_CreateDB.sql
echo "startup nomount " >> 10_CreateDB.sql
echo "create database \"&&DBNAME\" " >> 10_CreateDB.sql
echo "controlfile reuse" >> 10_CreateDB.sql
echo "maxinstances &&MAXINSTANCES" >> 10_CreateDB.sql
echo "maxloghistory &&MAXLOGHISTORY" >> 10_CreateDB.sql
echo "maxlogfiles &&MAXLOGFILES" >> 10_CreateDB.sql
echo "maxlogmembers &&MAXLOGMEMBERS" >> 10_CreateDB.sql
echo "maxdatafiles &&MAXDATAFILES" >> 10_CreateDB.sql
echo "datafile " >> 10_CreateDB.sql
#
counter=0
i=${SYS}
k=','
while [ "${i}" != "${counter}" ]; do
    counter=`expr ${counter} + 1`
    if [ "${i}" = "${counter}" ]; then
        k=''
    fi
    if [ "${ASM}x" != "ASMx" ]; then # Filesystem or raw devices
        echo " '&&DATADIR/system0${counter}.dbf' size && \
            SYSTEM_SIZE reuse${k}" >>10_CreateDB.sql
    else # ASM
        echo " '+&&DG_ASM_SYSTEM' size &&SYSTEM_SIZE reuse${k}" \
```

```
>>10_CreateDB.sql

fi
done
#
echo " extent management local" >> 10_CreateDB.sql
echo "sysaux" >> 10_CreateDB.sql
echo "datafile " >> 10_CreateDB.sql
counter=0
i=${AUX}
k=', '
while [ "${i}" != "${counter}" ]; do
    counter=`expr ${counter} + 1`
    if [ "${i}" = "${counter}" ]; then
        k=''
    fi
    if [ "${ASM}x" != "ASMx" ]; then # Filesystem or raw devices
        echo " '&&DATADIR/sysaux${counter}.dbf' size &&SYSAUX_SIZE reuse${k}"
    >>10_CreateDB.sql
    else # ASM
        echo " '+&&DG_ASM_SYSAUX' size &&SYSAUX_SIZE reuse${k}"
    >>10_CreateDB.sql
    fi
done
#
echo "default temporary tablespace temp" >> 10_CreateDB.sql
echo "tempfile" >> 10_CreateDB.sql
counter=0
i=${TMP}
k=', '
while [ "${i}" != "${counter}" ]; do
    counter=`expr ${counter} + 1`
    if [ "${i}" = "${counter}" ]; then
        k=''
    fi
    if [ "${ASM}x" != "ASMx" ]; then # Filesystem or raw devices
        echo " '&&TEMPDIR/temp0${counter}.dbf' size &&TEMP_SIZE reuse${k}"
    >>10_CreateDB.sql
    else # ASM
        echo " '+&&DG_ASM_TEMP' size &&TEMP_SIZE reuse${k}" >>10_CreateDB.sql
    fi
done
#
echo 'undo tablespace "UNDOTBS1"' >> 10_CreateDB.sql
echo "datafile " >> 10_CreateDB.sql
counter=0
i=${UND}
k=', '
while [ "${i}" != "${counter}" ]; do
    counter=`expr ${counter} + 1`
    if [ "${i}" = "${counter}" ]; then
        k=''
    fi
    if [ "${ASM}x" != "ASMx" ]; then # Filesystem or raw devices
        echo " '&&UNDODIR/undotbs10${counter}.dbf' size &&UNDO_SIZE reuse${k}"
    >>10_CreateDB.sql
    else # ASM
        echo " '+&&DG_ASM_UNDO' size &&UNDO_SIZE reuse${k}" >>10_CreateDB.sql
    fi
done
echo "CHARACTER SET &&CHARSET" >> 10_CreateDB.sql
echo "NATIONAL CHARACTER SET &&NATIONAL_CHARSET" >> 10_CreateDB.sql
echo "LOGFILE" >> 10_CreateDB.sql
counter=0
```



```
i=${RDO}
k=' '
while [ "${i}" != "${counter}" ]; do
  counter=`expr ${counter} + 1`
  if [ "${i}" = "${counter}" ]; then
    k=' '
  fi
  if [ "${ASM}x" != "ASMx" ]; then # Filesystem or raw devices
    echo "  GROUP ${counter} ('&&REDODIR/redo0${counter}.log') SIZE
&&REDO_SIZE reuse${k}" >>10_Crea
teDB.sql
  else # ASM
    echo "  GROUP ${counter} ('+&&DG_ASM_REDO') SIZE &&REDO_SIZE reuse${k}"
>>10_CreatedB.sql
  fi
done
echo 'USER SYS IDENTIFIED BY "&&SYSPWD"'
>>10_CreatedB.sql
echo 'USER SYSTEM IDENTIFIED BY "&&SYSTEMPWD";'
>>10_CreatedB.sql
```

The 10_CreateDB.sh script creates a script 10_CreatedB.sql depending on various parameters in DB.conf.

The following is an example of the output of 10_CreatedB.sql:

```
shutdown abort
set echo on
startup nomount
create database "&&DBNAME"
controlfile reuse
maxinstances &&MAXINSTANCES
maxloghistory &&MAXLOGHISTORY
maxlogfiles &&MAXLOGFILES
maxlogmembers &&MAXLOGMEMBERS
maxdatafiles &&MAXDATAFILES
datafile
'&&DATADIR/system01.dbf' size &&SYSTEM_SIZE reuse,
'&&DATADIR/system02.dbf' size &&SYSTEM_SIZE reuse,
'&&DATADIR/system03.dbf' size &&SYSTEM_SIZE reuse
extent management local
sysaux
datafile
'&&DATADIR/sysaux1.dbf' size &&SYSAUX_SIZE reuse,
'&&DATADIR/sysaux2.dbf' size &&SYSAUX_SIZE reuse
default temporary tablespace temp
tempfile
'&&TEMPDIR/temp01.dbf' size &&TEMP_SIZE reuse,
'&&TEMPDIR/temp02.dbf' size &&TEMP_SIZE reuse
undo tablespace "UNDOTBS1"
datafile
'&&UNDODIR/undotbs101.dbf' size &&UNDO_SIZE reuse,
'&&UNDODIR/undotbs102.dbf' size &&UNDO_SIZE reuse
CHARACTER SET &&CHARSET
NATIONAL CHARACTER SET &&NATIONAL_CHARSET
LOGFILE
GROUP 1 ('&&REDODIR/redo01.log') SIZE &&REDO_SIZE reuse,
```

The Highly Available Oracle Database

```
GROUP 2 ('&&REDODIR/redo02.log') SIZE &&REDO_SIZE reuse,  
GROUP 3 ('&&REDODIR/redo03.log') SIZE &&REDO_SIZE reuse,  
GROUP 4 ('&&REDODIR/redo04.log') SIZE &&REDO_SIZE reuse,  
GROUP 5 ('&&REDODIR/redo05.log') SIZE &&REDO_SIZE reuse  
USER SYS IDENTIFIED BY "&&SYSPWD"
```

7. It is **absolutely** critical that your create_database script runs correctly. The last several lines of the script output should be the same as the listing below. If the last portion of your screen output does not look like the following, ask your instructor for assistance.

```
PL/SQL procedure successfully completed.

Elapsed: 00:00:07.41
11:00:38 SQL>
11:00:38 SQL> EXECUTE dbms_expfil_reg.validate_expfil;

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.09
11:00:38 SQL>
11:00:38 SQL> ALTER SESSION SET CURRENT_SCHEMA = SYS;

Session altered.

Elapsed: 00:00:00.00
11:00:38 SQL> --
11:00:38 SQL> host echo "End database creation &&DBNAME at " `date` >>
&&SP00LDIR/time.log

11:00:38 SQL> select * from system.test;

      NUM NAME          CITY
-----
        0 MJ            Chicago; IL
        1 Magic          Los Angeles, CA
        2 Bird            Boston, MA
        3 Gary            Detroit, MI
        4 Martin          Frankfurt, Germany
        5 Vince           Toronto, Canada

6 rows selected.

Elapsed: 00:00:00.07
11:00:38 SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
11:01:08 SQL> spool off
11:01:08 SQL> exit;
Disconnected from Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 -
64bit Production
With the Partitioning, Oracle Label Security, OLAP and Data Mining Scoring
Engine options

Fri Jun  3 11:01:09 EDT 2005
$
```

8. **[Node1/oracle]** Once the script completes, exit the oracle user account and un-mount the Oracle file system. Then, de-activate the volume group vg03.

```
$ exit                                ← Exit the oracle account
# umount /oracle1
# vgchange -a n vg03
```

9. **[Node2/root]** Again on **<node2>**, create a mount point directory for the oracle1 package, and make it writable. Then activate the volume group and mount the file system /oracle1.

```
# mkdir /oracle1
# vgchange -a e vg03
# mount -F vxfs /dev/vg03/oracle1 /oracle1
# chmod 777 /oracle1
```

10. **[Node2/oracle]** Create two required links from shared device to the local Oracle_HOME/dbs directory.

```
# su - oracle
$ ln -sf /oracle1/admin/pfile/orapwORA1SID \
      /opt/oracle/product/10.1.0.db/dbs
$ ln -sf /oracle1/admin/pfile/initORA1SID.ora \
      /opt/oracle/product/10.1.0.db/dbs
$ exit                                ← Exit the Oracle account
```

11. **[Node2/root]** Un-mount the oracle file system and de-activate volume group vg03.

```
# umount /oracle1
# vgchange -a n vg03
```

12. **[Node1/root]** Create an entry in the `/etc/hosts` file to refer to a 10-net non-routable IP address that will be used by the Oracle Listener.

```
# vi /etc/hosts
```

and add the following line, where the `10.x.x.x` is replaced by a 10-net non-routable IP address provided by your instructor.

<code>10.x.x.x</code>	<code>ora1host</code>
-----------------------	-----------------------

When finished, exit the file and save your changes.

13. **[Node2/root]** Create an entry in the `/etc/hosts` file to refer to a 10-net non-routable IP address that will be used by the Oracle Listener.

```
# vi /etc/hosts
```

and add the following line, where the `10.x.x.x` is replaced by a 10-net non-routable IP address provided by your instructor.

<code>10.x.x.x</code>	<code>ora1host</code>
-----------------------	-----------------------

When finished, exit the file and save your changes.

Part 4: Create the Package Configuration File and Control Script

1. **[Node1/root]** While still on <node1>, make a directory for the Oracle package files and cd into the new directory. Note again that the "oracle1" directory ends with a one, not an "ell".

```
# cd /etc/cmcluster
# mkdir oracle1
# cd oracle1
```

2. **[Node1/root]** Next, create the package template file for the oracle package, named oracle1.

```
# cmmakepkg -v -p oracle1.conf
```

3. **[Node1/root]** Edit this package configuration file as described below.

```
# vi oracle1.conf
```

and modify the following parameters:

PACKAGE_NAME	oracle1
NODE_NAME	<node1>
NODE_NAME	<node2>
RUN_SCRIPT	/etc/cmcluster/oracle1/oracle1.cntl
RUN_SCRIPT_TIMEOUT	NO_TIMEOUT
HALT_SCRIPT	/etc/cmcluster/oracle1/oracle1.cntl
HALT_SCRIPT_TIMEOUT	NO_TIMEOUT
SERVICE_NAME	ORACLE1
SERVICE_FAIL_FAST_ENABLED	NO
SERVICE_HALT_TIMEOUT	300
SUBNET	10.x.x.x ← Use netstat -in to obtain the cluster 10-net subnet

When finished, save your changes and exit the file.

4. **[Node1/root]** Create the **run/halt** script template.

```
# cmmakepkg -v -s oracle1.cntl
```

5. **[Node1/root]** Edit the package control script template as described below.

```
# vi oracle1.cntl
```

and modify the following parameters:

```
VG[0]="vg03"

LV[0]="/dev/vg03/oracle1"
FS[0]="/oracle1"
FS_MOUNT_OPT[0]=""
FS_UMOUNT_OPT[0]=""
FS_FSCK_OPT[0]=""
FS_TYPE[0]="vxfs"

IP[0]="10.x.x.x"      → get a 10-net IP address from your instructor
SUBNET="10.y.y.y"     → use netstat -in to obtain the cluster 10-net
                        subnet

SERVICE_NAME[0]="ORACLE1"
SERVICE_CMD[0]="/etc/cmcluster/oracle1/haoracle.sh monitor"
SERVICE_RESTART[0]="-r 0"
```

```
function customer_defined_run_cmds {
# ADD customer definedrun commands

    /etc/cmcluster/oracle1/haoracle.sh start
    test_return 51
}
```

```
function customer_defined_halt_cmds {
# ADD customer defined run commands

    /etc/cmcluster/oracle1/haoracle.sh shutdown
    test_return 52
}
```

When finished, save your changes and exit the file.

Part 5: Create the Oracle Monitoring Script

1. **[Node1/root]** Place a copy of the default monitoring script in the `oracle1` directory.

```
# cd /etc/cmcluster/oracle1
# cp /opt/cmcluster/toolkit/oracle/haoracle* .
```

2. **[Node1/root]** Update the `haoracle.conf` file using the values in the table below:

ORACLE_HOME=/opt/oracle/product/10.1.0.db
SID_NAME=ORA1SID
LISTENER_NAME=LISTENERORA1

Save your changes and exit the file.

Part 6: Copy the Control and Monitor Scripts to Second System

1. **[Node1/root]** On your primary system, ensure that the `haoracle.sh` and `oracle1.cntl` scripts are both executable.

```
# cd /etc/cmcluster/oracle1
# chmod 755 oracle1.cntl
# chmod 755 haoracle.sh
```

2. **[Node2/root]** On <node2>, create the oracle package directory and copy the appropriate files from <node1>.

```
# cd /etc/cmcluster
# mkdir oracle1
# cd oracle1
# rcp <node1>:/etc/cmcluster/oracle1/* .
```


Part 7: Finish the Configuration of the Oracle Package

1. **[Node1/root]** Back on <node1>, modify the cluster-wide ASCII configuration file to include our new **vg03** volume group.

```
# cd /etc/cmcluster
# vi cmclconfig.ascii          ← add vg03 at the end of the file
```

2. **[Node1/root]** Check both the cluster-wide configuration file and the Oracle package configuration file for errors.

```
# cd oracle1
# cmcheckconf -v -P oracle1.conf -C ../cmclconfig.ascii
```

3. **[Node1/root]** Fix any errors. Once configuration is OK, apply the new configuration file.

```
# cmapplyconf -v -P oracle1.conf -C ../cmclconfig.ascii
```

4. **[Node1/root]** Once the new configuration is applied, start the `oracle1` package.

```
# vgchange -a n vg03          ← Be sure vg03 is not activated on either node
# cmmodpkg -v -e oracle1
```

Module 13
The Highly Available Oracle Database

5. **[Node1/oracle]** On whichever node the oracle1 package is running (it should be <node1>), check the database status and contents.

```
# su - oracle
$ /labs/mod_13/test_oracle.sh
```

You should see output that looks like the following:

```
$ /labs/mod_13/test_oracle.sh

SQL*Plus: Release 10.1.0.2.0 - Production on Wed Jun 8 08:15:26
2005

Copyright (c) 1982, 2004, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - 64bit
Production
With the Partitioning, Oracle Label Security, OLAP and Data Mining
Scoring Engine options

SQL>
      NUM NAME          CITY
-----
      0 MJ             Chicago; IL
      1 Magic          Los Angeles, CA
      2 Bird            Boston, MA
      3 Gary            Detroit, MI
      4 Martin          Frankfurt, Germany
      5 Vince           Toronto, Canada

6 rows selected.

SQL> Disconnected from Oracle Database 10g Enterprise Edition
Release 10.1.0.2.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP and Data Mining
Scoring Engine options

$
```

Be sure to exit from the oracle login before continuing

```
$ exit
```

Part 8: Manually Test Serviceguard Failover of the Oracle Package

1. **[Node1/root]** First, verify that the oracle1 package can start on the alternate node correctly.

```
# cmhaltpkg -v oracle1
# cmrunpkg -v -n <node2> oracle1
```

2. **[Node2/oracle]** Now, on <node2> (where the oracle1 package should be running, check the database status and contents.

```
# su - oracle
$ /labs/mod_13/test_oracle.sh
```

As before, you should see output that looks like the following:

```
$ /labs/mod_13/test_oracle.sh

SQL*Plus: Release 10.1.0.2.0 - Production on Wed Jun 8 08:15:26
2005

Copyright (c) 1982, 2004, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - 64bit
Production
With the Partitioning, Oracle Label Security, OLAP and Data Mining
Scoring Engine options

SQL>
      NUM NAME                CITY
-----
      0 MJ                    Chicago; IL
      1 Magic                 Los Angeles, CA
      2 Bird                  Boston, MA
      3 Gary                  Detroit, MI
      4 Martin                Frankfurt, Germany
      5 Vince                 Toronto, Canada

6 rows selected.

SQL> Disconnected from Oracle Database 10g Enterprise Edition
Release 10.1.0.2.0 - 64bit Production
With the Partitioning, Oracle Label Security, OLAP and Data Mining
Scoring Engine options

$
```

Be sure to exit from the oracle login before continuing

```
$ exit
```

3. **[Node1/root]** Verify the oracle1 package as it starts up on the alternate node in the cluster. Also, re-enable AUTO_RUN switch for your oracle1 package.

```
# cmviewcl -v -p oracle1  
# cmmodpkg -v -e oracle1
```

If the package is not running correctly on either node, troubleshoot the problem.

4. Congratulations on successfully configuring your first Oracle package!

13-9. LAB: Test Oracle Package Failover

1. From the node on which your `oracle1` package is currently running, and as the `oracle` user, execute the `loop_oracle.sh` script, which will write to the database once a second.

```
# cmviewcl -vp oracle1          ← To determine on which node oracle1 is running
# su - oracle
$ /labs/mod_13/loop_oracle.sh
```

Allow the `loop_oracle.sh` script to continue running as you view the script contents. Then, while the `loop_oracle.sh` script is still running, continue to step 2 in this lab.

The `loop_oracle.sh` script contains the following code:

```
#####
#
# This script inserts rows into table test.
#
i=150
while true
do
    sqlplus -S -s system/manager <<-! >/dev/null 2>&1
    set echo off
    set termout off
    whenever sqlerror exit 5;
    insert into test
        values ($i, $i || ' Any Name', $i || ' Any City');
    commit;
    exit;
    !
    if [[ $? -ne 0 ]]
    then
        echo "Oracle has exited"
        exit
    else
        echo "$i occurrence in loop"
        i=$(expr $i + 1)
        sleep 1
    fi
    if [[ $i -gt 300 ]]
    then
        exit
    fi
done
```

The Highly Available Oracle Database

2. Allow the `loop_oracle.sh` script to continue to run. Now, from the node on which your `oracle1` package is currently running, open a new terminal window and prepare to kill the `haoracle.sh` service process. In the window in which the `loop_oracle.sh` script is running, note the last line number written to the database at the moment you execute the following `kill` command. Write that number below.

```
# ps -ef | grep haoracle.sh
# kill <pid>
```

Enter here the line number of the last line written to the database → _____

3. Verify that the `oracle1` package fails over to the other node in the cluster.

```
# cmviewcl -vp oracle1
```

4. From the node where the `oracle1` package is **now** running, verify that data in the database table experienced no data loss as follows: The last number written to the database (as seen in the output of the following script) should match the last number, as logged in step 2 above, thus proving that there was no loss of data during the failover. When you are finished viewing the Oracle data, exit the `oracle` account.

```
# su - oracle
$ /labs/mod_13/test_oracle.sh
$ exit
```

5. Finally, to allow for future failovers, re-enable your node switching parameter for the node where the `oracle1` package was previously running.

```
# cmmodpkg -v -e -n <nodeX> oracle1
```

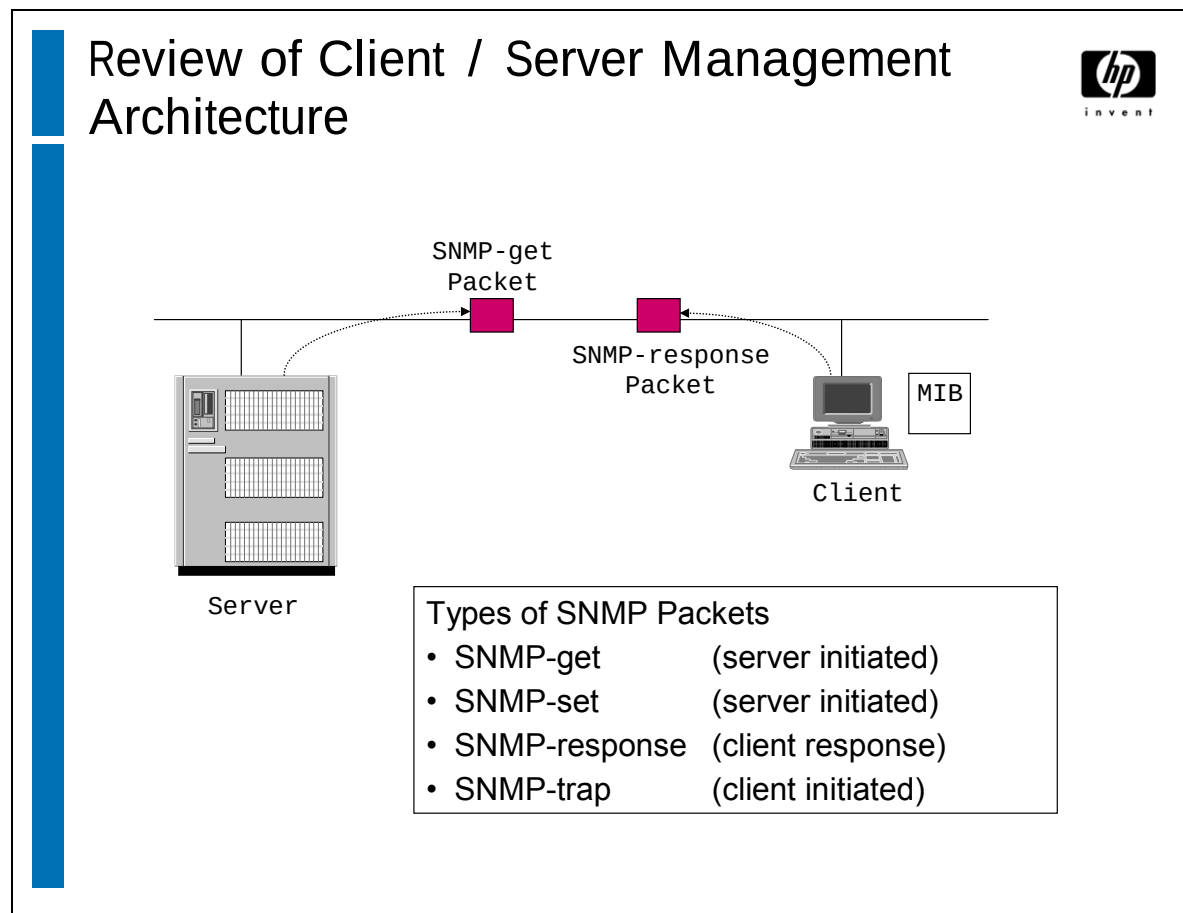
Module 14 — EMS Resources and Serviceguard Packages

Objectives

Upon completion of this module, you will be able to do the following:

- List the five EMS monitoring daemons.
- Use the `res ls` command to list the available EMS resources.
- Set up a Serviceguard package to use EMS resources.

14-1. SLIDE: Review of Client/Server Management Architecture



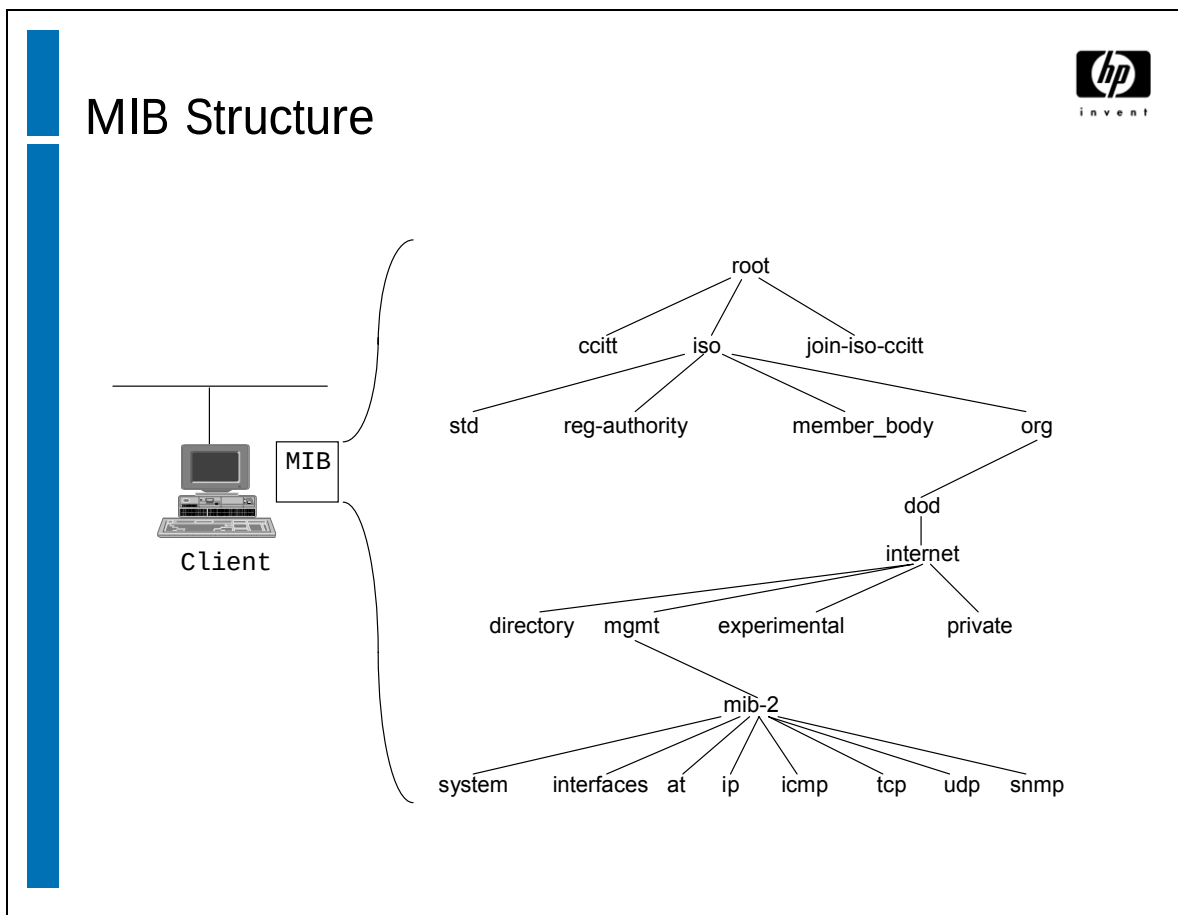
Student Notes

A client/server monitoring protocol, which has been around for a long time, is the SNMP design. This is the design in which EMS (Event Monitoring Service) monitoring is based upon.

The SNMP protocol was designed to allow one system (a server) to monitor many other systems (clients) remotely, easily, and with as little overhead as possible for the server.

With the SNMP protocol, every client maintains a data structure called a Management Information Base (MIB), which contains many variables related to the state and status of the client. If a server wants to retrieve a specific MIB variable from a client, the server initiates an SNMP-get packet to the client. The client then returns the MIB value through an SNMP-response packet.

14-2. SLIDE: MIB Structure

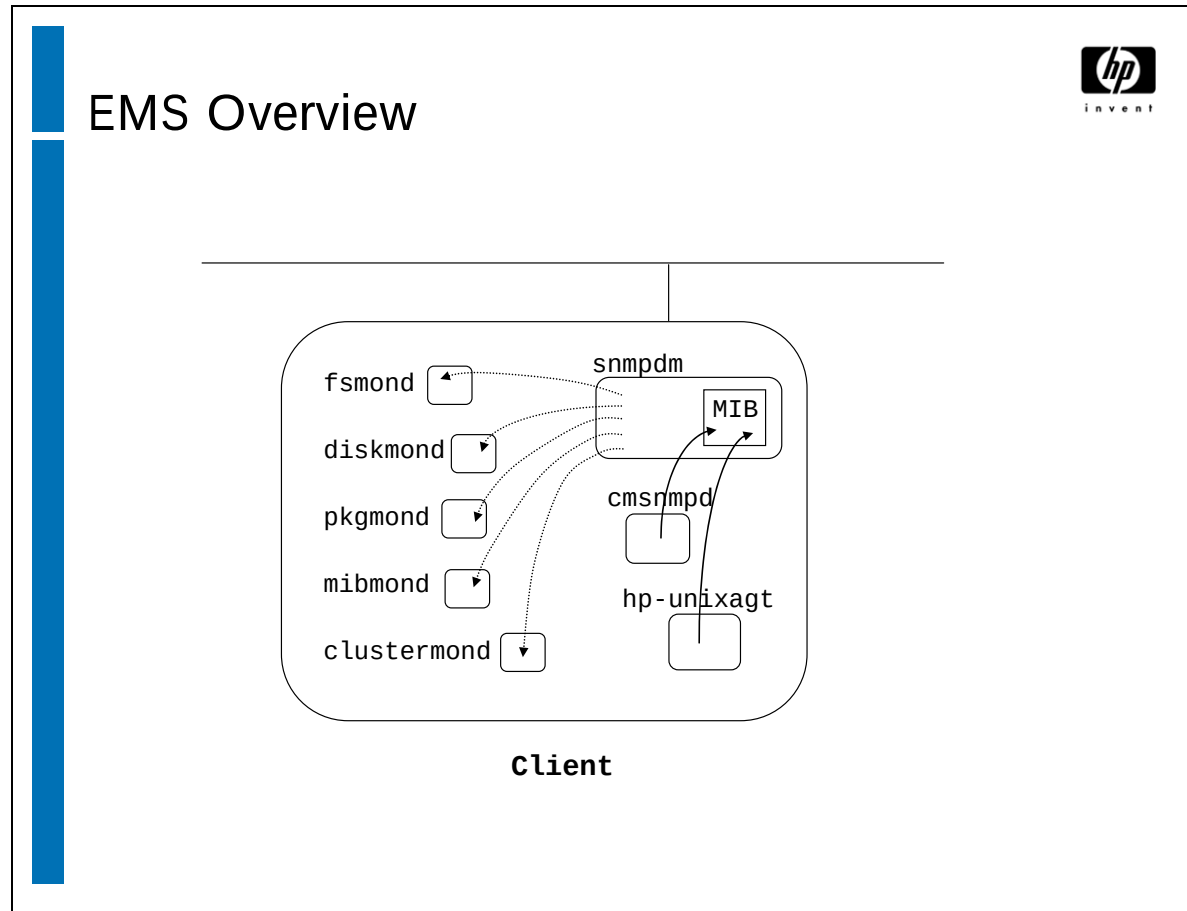


Student Notes

The slide contains a general, high-level diagram of the MIB data tree structure maintained on each client system. There are hundreds of different variables maintained in this structure, including system information, NIC status information, and statistics related to the IP, TCP, UDP, and ICMP protocols.

The MIB structure is maintained in the data area of the snmpdm process (also known as the SNMP master daemon).

14-3. SLIDE: EMS Overview



Student Notes

The EMS product provides local monitoring services for applications running on the local system.

The EMS product works by starting five daemons, which monitor variables in the local MIB, and are kept and maintained by the SNMP master daemon.

The five EMS daemons are:

- fsmond Monitors file system variables like percent full and MB available.
- diskmond Monitors disk variables like driver status and disk errors.
- mibmond Monitors general network variables like NIC status and packet counts.
- pkgmond Monitors Serviceguard package variables.
- clustermond Monitors Serviceguard cluster variables.

14-4. TEXT PAGE: EMS Resource Class Hierarchy

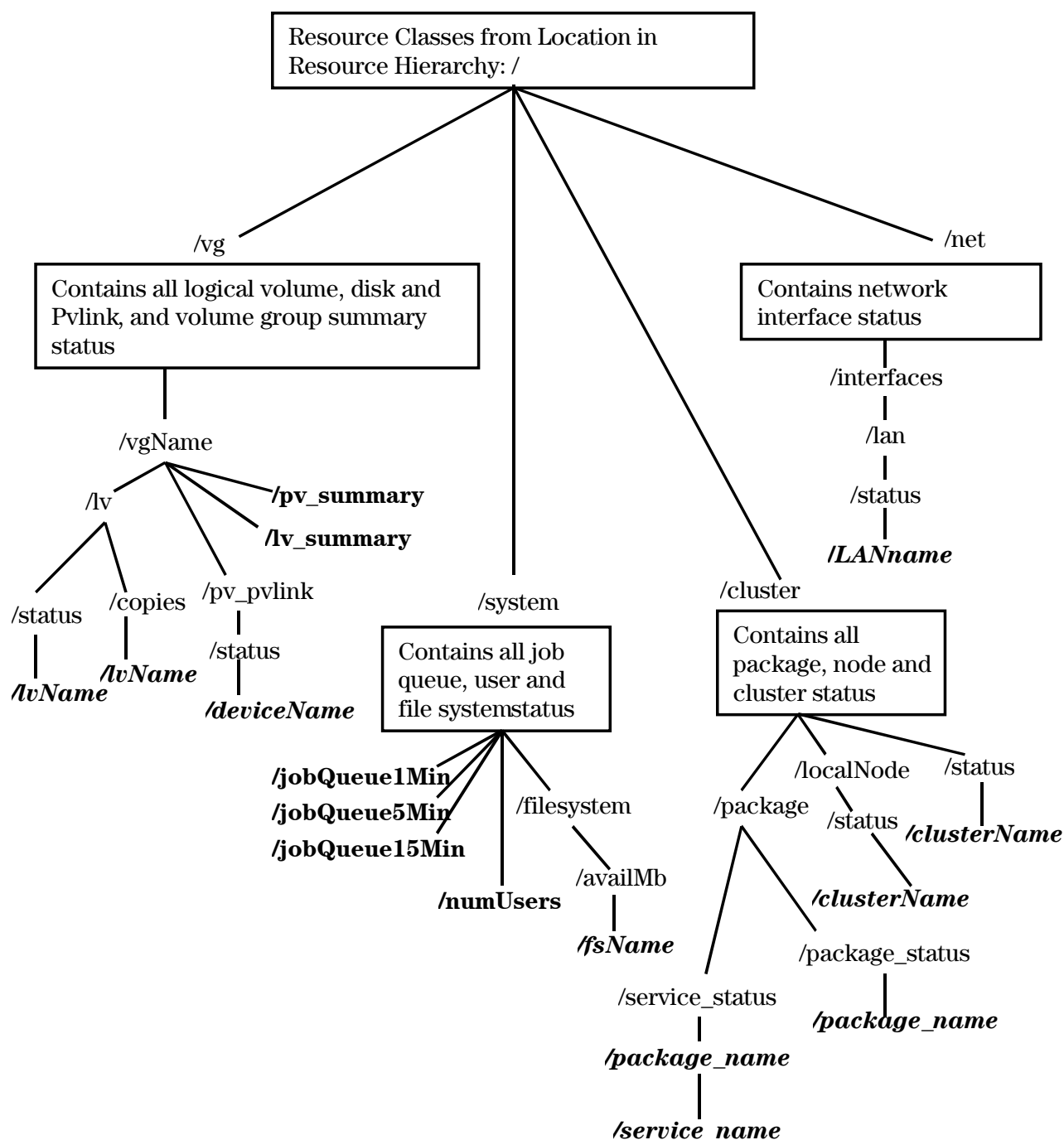
Online documentation about EMS is available in the following entries:

man 5 ems, man 1 resls, man 1m diskmond, and man 1m mibmond

The full path of a resource includes the resource class hierarchy and instance. The figure shows an example of a full resource path for the physical volume status of the device `/dev/dsk/c0t1d2` belonging to volume group `vgDataBase` is `/vg/vgDataBase/pv_pvlink/status/c0t1d2`

The `resls` command takes a resource pathname as an argument and displays elements under that pathname that are being monitored. `"/` is the top of the resource path tree.

The following chart shows the resource hierarchy.



14-5. SLIDE: Setting Up a Package to Use an EMS Resource

Setting Up a Package to Use an EMS Resource



1. Use the `resls` command to identify the EMS resource name.
2. Edit the package configuration file and add the resource definition.

```
vi /etc/cmcluster/package/pkg.conf
```

```
RESOURCE_NAME           /system/filesystem/availMb/home
RESOURCE_POLLING_INTERVAL 60
RESOURCE_START           AUTOMATIC
RESOURCE_UP_VALUE        >= 5
```

3. Halt the package.
4. Reapply the package configuration file.
5. Restart the package.

Student Notes

The above slide shows the procedure to setup a package to use an EMS resource.

The `resls` command can be used to list the different resource variables.

To list an EMS resource, the full name of the resource must be specified. For example, to list the available megabytes currently free in the `/home` file system, the command is:

```
# resls /system/filesystem/availMb
```

NOTE: The default `RESOURCE_POLLING_INTERVAL` is 60 seconds. (The minimum is 30 seconds).

A feature first introduced with Serviceguard version 10.10 is the ability to make a package dependent upon an EMS resource (a MIB variable).

EMS Resources and Serviceguard Packages

In order to do this, the `cmcl` daemon needs to know if any packages are defined containing EMS resources. The package variables are:

```
RESOURCE_NAME  
RESOURCE_POLLING_INTERVAL  
RESOURCE_START  
RESOURCE_UP_VALUE
```

If a package does have an EMS resource defined, the `cmcl` must register with the appropriate EMS monitoring daemons during its initialization. By registering, the EMS daemons return the current value of the variable every `RESOURCE_POLLING_INTERVAL` seconds.

If the `RESOURCE_NAME` variable ever returns with a value outside the range of `RESOURCE_UP_VALUE`, then the `cmcl` daemon will failover the package to a node where the `RESOURCE_UP_VALUE` is within specified parameters.

The `RESOURCE_POLLING_INTERVAL` indicates how often, in seconds, the resource is to be monitored. The default `RESOURCE_POLLING_INTERVAL` is 60 seconds if not specified. The minimum `RESOURCE_POLLING_INTERVAL` is 30 seconds.

The `RESOURCE_START` value may be set to either `AUTOMATIC` or `DEFERRED`. The default setting is `AUTOMATIC`. A setting of `AUTOMATIC` will mean that Serviceguard will automatically start up resource monitoring when the node starts up. If `DEFERRED` is selected, Serviceguard will not attempt to start resource monitoring during node start up. In this case, the Serviceguard administrator should specify all `DEFERRED` resources in the package run script so that these `DEFERRED` resources will be started up from the package run script when the package starts up.

The `RESOURCE_UP_VALUE` requires an operator and a value. The combination of these two items defines the resource "UP" condition. The operators are `=`, `!=`, `>`, `<`, `>=`, and `<=`, depending on the type of value. These values can be string or numeric. If the type is string, then only `=` and `!=` are valid operators. If the string contains whitespace, it must be enclosed in quotes. The following are some examples of the correct use of the `RESOURCE_UP_VALUE` ...

<code>RESOURCE_UP_VALUE</code>	<code>= UP</code>
<code>RESOURCE_UP_VALUE</code>	<code>!= DOWN</code>
<code>RESOURCE_UP_VALUE</code>	<code>= 5</code>
<code>RESOURCE_UP_VALUE</code>	<code>> 5.1</code>
<code>RESOURCE_UP_VALUE</code>	<code>> -5 and < 10</code>

14-6. LAB: Monitoring File Systems with EMS

Directions

This lab uses EMS (Event Monitoring Service) to monitor the free disk space available in a new file system. This Resource Monitor will be added to the existing package logprog.

1. First, ensure you have all EMS products loaded on all nodes in your cluster. Check and verify that you have the following five products already installed:
 - EMS-Config,
 - EMS-Core,
 - EMS-DBMon,
 - EMS-DskMon, and
 - EMS-MIBMon.

Answer:

```
# swlist -l product | grep -i ems
```

2. If your systems are missing any of the above products, swinstall them now.

Answer:

```
# swinstall
```

3. Perform the following steps on both nodes in your cluster.
In volume group vg00, create a new 64MB logical named ems_test. Build a vxfs file system in the ems_test logical volume. Make a /ems_test mount point directory and mount your new file system.

Answer:

```
<node1># lvcreate -L 64 -n ems_test vg00
<node1># newfs -F vxfs /dev/vg00/ems_test
<node1># mkdir /ems_test
<node1># mount /dev/vg00/ems_test /ems_test

<node2># lvcreate -L 64 -n ems_test vg00
<node2># newfs -F vxfs /dev/vg00/ems_test
<node2># mkdir /ems_test
<node2># mount /dev/vg00/ems_test /ems_test
```

4. Next, halt the logprog package.

Answer:

```
# cmhaltpkg -v logprog
```

5. Edit the logprog.conf file to add a resource to the logprog package. Afterwards, check and apply your logprog.conf file

```
# cd /etc/cmcluster/logprog
# vi logprog.conf
```

RESOURCE_NAME	/system/filesystem/availMb/ems_test
RESOURCE_POLLING_INTERVAL	30
RESOURCE_START	AUTOMATIC
RESOURCE_UP_VALUE	>= 48

```
# cmcheckconf -v -P logprog.conf
# cmapplyconf -v -P logprog.conf
```

6. To test our new EMS resource, bring up the logprog package.

Answer:

```
# cmmodpkg -v -e logprog
```

7. Let us now confirm the resource monitoring.

Answer:

```
# cmviewcl -v -p logprog
```

8. To test the EMS failover capability, copy a fairly large file (eg. /stand/vmunix) into the /ems_test file system. This file system is 64 MB in size. An HP-UX 11.23 kernel file is about 53 MB in size. An HP-UX 11.00 kernel file is about 25 MB in size. Either way, after copying, the available free space in the /ems_test file system should be well under the 48 MB limit, as specified by the RESOURCE_UP_VALUE. This will then cause the logprog package to failover.

Answer:

```
# cp /stand/vmunix /ems_test/.
```


9. After the failover, execute a `cmviewcl` on the `logprog` package. Note the condition of the `logprog` package node switches.

Answer:

```
# cmviewcl -v -p logprog
```

10. As you have time and interest, continue to experiment with the EMS resources. For example, what happens to the `logprog` package if the `RESOURCE_UP_VALUE` is violated on both nodes simultaneously?

Answer:

Run various experiments.

11. To return the cluster to its previous state, remove the resource from your `logprog` package. Also, remove all copies of `vmunix` from the `/ems_test` file system(s).

Answer:

Re-comment out the following lines in `logprog.conf`:

```
#RESOURCE_NAME  
#RESOURCE_POLLING_INTERVAL  
#RESOURCE_START  
#RESOURCE_UP_VALUE
```

```
Then:    # cmhaltpkg -v logprog  
         # cmcheckconf -P /etc/cmcluster/logprog/logprog.conf  
         # cmapplyconf -P /etc/cmcluster/logprog/logprog.conf  
         # cmmodpkg -e logprog
```

14-7. LAB: Monitoring *numUsers* with EMS

Directions

Special Note

- a. This lab demonstrates the use of EMS and Serviceguard to monitor "numUsers."
- b. **VERY IMPORTANT:** If you are using remote equipment, this lab will perform correctly only when using an `hpterm` to log into Unix. The lab will not work correctly using "**XDMCP DIRECT or Putty**". To simulate multiple logins, use ReflectionX.

1. First, halt the logprog package.

Answer:

```
# cmhaltpkg -v logprog
```

2. Modify the `logprog.conf` file as follows:

```
# cd /etc/cmcluster/logprog
# vi logprog.conf
```

Set the following values:

RESOURCE_NAME	/system/numUsers
RESOURCE_POLLING_INTERVAL	60
RESOURCE_START	AUTOMATIC
RESOURCE_UP_VALUE	< 4

Note that the above lines must be *uncommented*, and they must have values added to the end of each line.

Also note that, if you have several windows open already, the `RESOURCE_UP_VALUE` may need to be modified upward. Choose a value that is 2 or 3 greater than the number of windows currently open.

3. Now, check the `logprog.conf` file for errors, apply it, and bring up the `logprog` package.

Answer:

```
# cmcheckconf -v -P logprog.conf
# cmapplyconf -v -P logprog.conf
# cmmodpkg -v -e logprog
```

4. To test this new resource, do several logons to the node running the `logprog` package. Create enough separate logins (or, open enough additional `hpterm` windows) to violate the `RESOURCE_UP_VALUE`. This should cause the package to move over to the adoptive node. You should also notice the state of this resource in `cmviewcl -v`.

Answer:

Create enough separate logins to violate the `RESOURCE_UP_VALUE`. Remember the “Special Note” at the beginning of this lab concerning the use of `hpterm`s only. Afterward, ...

```
# cmviewcl -v -p logprog
```

5. To return the cluster to its previous state, remove the four “`RESOURCE`” lines from the `logprog` package configuration file. Then `cmcheckconf` and `cmapplyconf` your `logprog` package. Finally, start your `logprog` package.

Answer:

```
# vi logprog.conf
# cmcheckconf -v -P logprog.conf
# cmapplyconf -v -P logprog.conf
# cmmodpkg -v -e logprog
```

Some Final Notes

- a. Package switches are unaffected, even after the `logprog` package fails over to an adoptive node due to resource failure.
- b. No logging information is found in the package logs; all logging is in `syslog.log`.
- c. A package cannot even start on a node in which the resource is not met.

Module 14
EMS Resources and Serviceguard Packages

Module 15 — High Availability Networking

Objectives

Upon completion of this module, you will be able to do the following:

- Connect a second network to a Serviceguard cluster with multiple levels of redundancy.
- Set up a Serviceguard node to provide LAN card redundancy.
- Configure a highly available network using redundant hubs.
- Configure a highly available network using redundant routers.
- Configure a highly available network using redundant networks.

15-1. SLIDE: Network Redundancy



Network Redundancy

- Redundant LAN cards
- Redundant hubs
- Redundant routers
- Redundant networks

Student Notes

In order to provide a high level of availability, a typical cluster should remain accessible to client systems even after a failure of any network component.

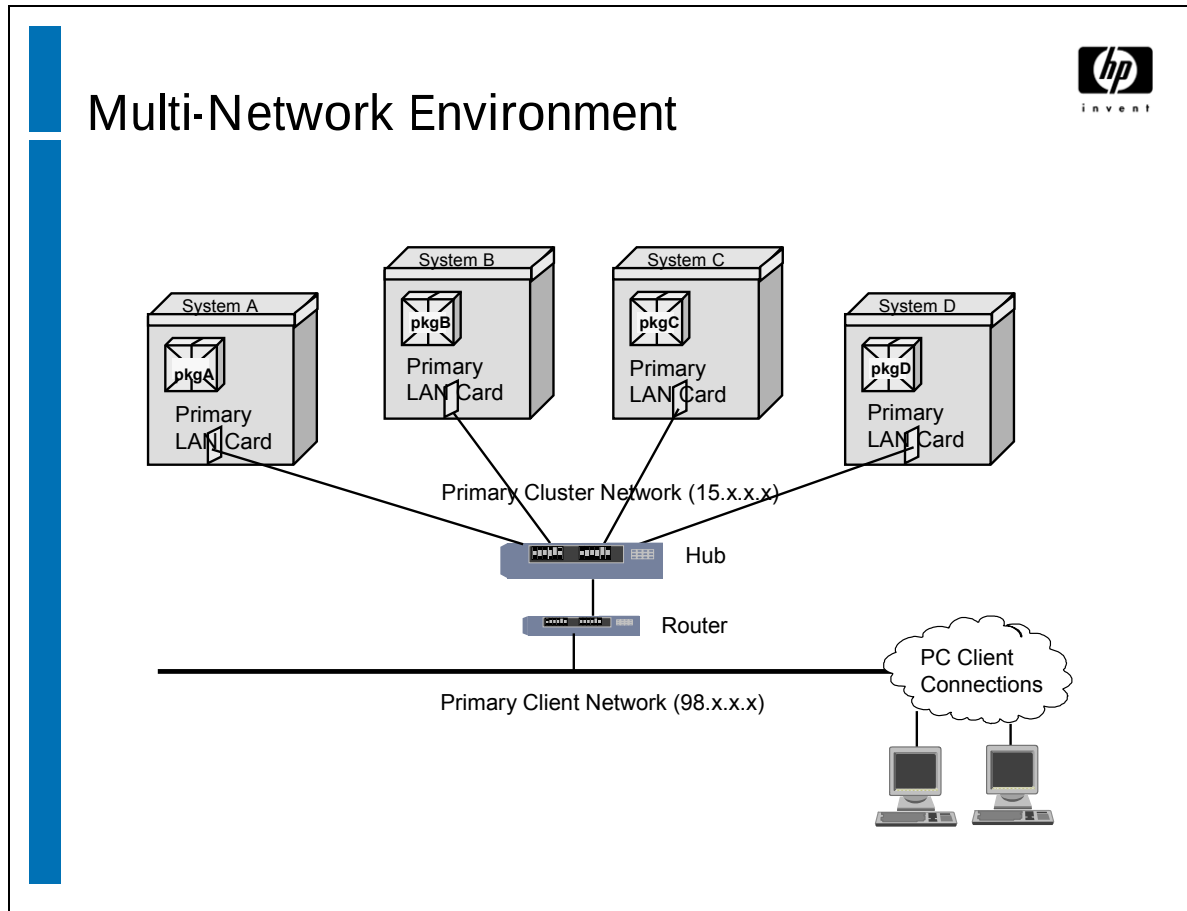
The key to eliminating SPOF (within a network) is understanding the topologies and the failure points within those topologies.

For most customers, it is not cost effective to eliminate every single possible point of failure within their network for every single node. In a Serviceguard environment, the nodes within the cluster (at the very least) should be made highly available from a networking standpoint. Any LAN card, LAN segment, hub, or router failure, should not cause communication among nodes within the cluster to fail.

For other nodes within the network, it will depend on the type of node and how that node is used as to whether the node should be highly available.

This module covers how SPOF in the network can be eliminated from the server side and how to segment the clients such that a network failure only affects a small portion of the users.

15-2. SLIDE: Multi-Network Environment



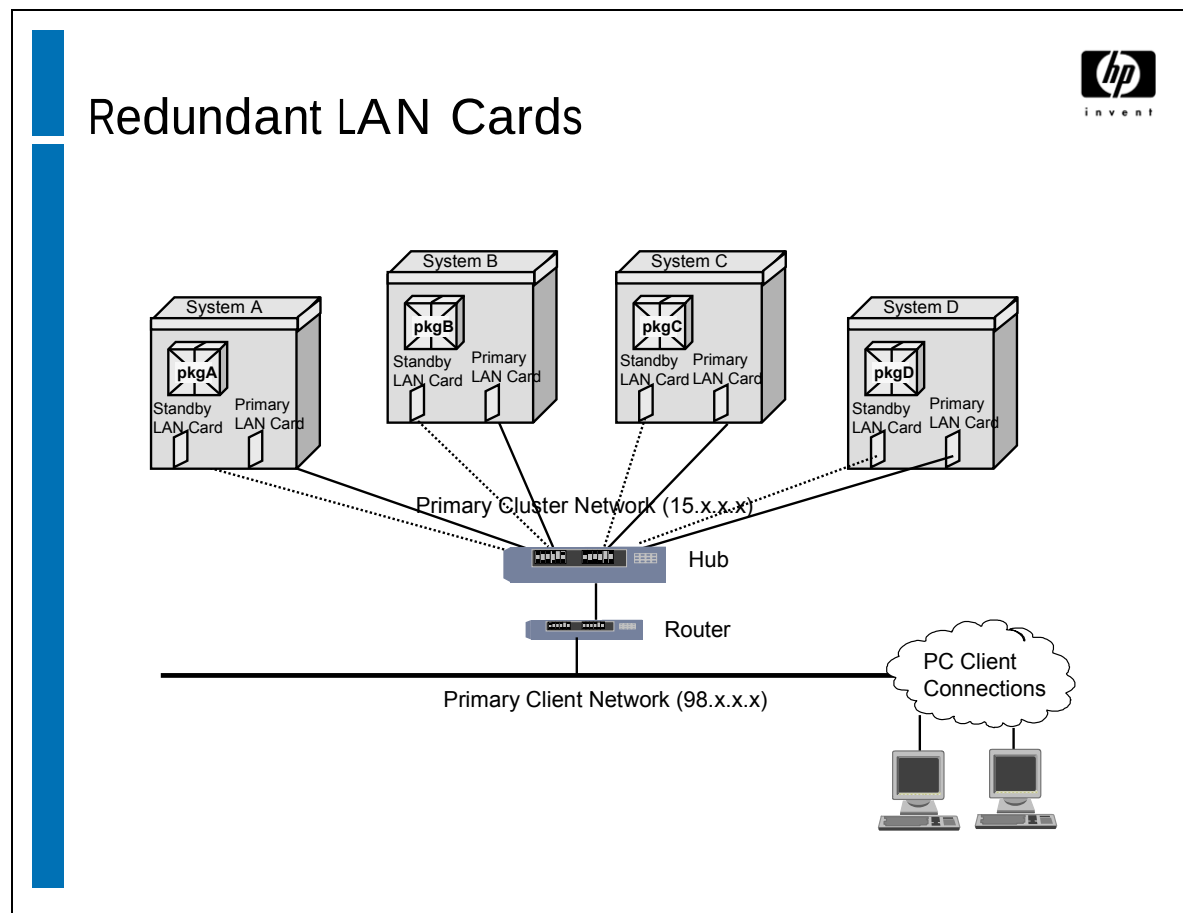
Student Notes

Typically, a Serviceguard network is not on the same network as the client systems. In many cases, a firewall separates clients from the cluster to prevent unauthorized access to the cluster.

The next few slides look at different issues/concerns when connecting a second network (i.e. a client network) to a Serviceguard network.

In the above slide, identify four potential SPOF that exist in the network connection between the Serviceguard servers and the PC Client systems.

15-3. SLIDE: Redundant LAN Cards



Student Notes

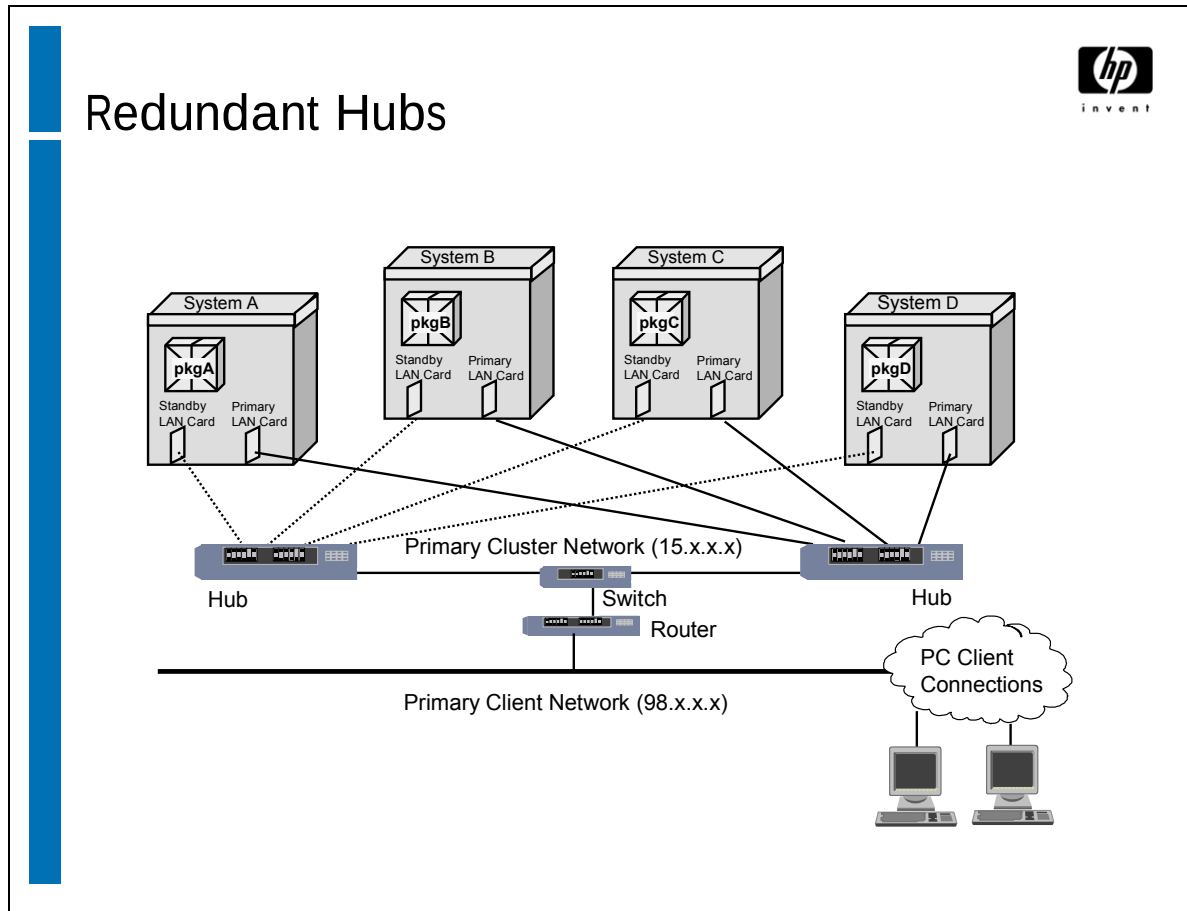
The first step towards eliminating single points of failure for networking is to ensure each node in the cluster has redundant network interface cards. A second network interface card also protects against a cable failure between the node and the hub.

The IP addresses for the packages and the node are assigned to the primary network interface card. The second interface card initially does not have any IP addresses assigned to it, and it is referred to as the "standby" interface card.

Serviceguard uses the standby card if it detects a failure with network communications on the primary card. The network communications failure could be a failed interface card, or a bad cable, or a down device connecting the node to the client network, like a hub or router.

Whenever Serviceguard detects no communications is possible through the primary interface card, it switches all IP addresses over to the standby interface card.

15-4. SLIDE: Redundant Hubs



Student Notes

If the cluster is cabled using hubs (as shown on the slide), a minimum of two hubs should be used to avoid a SPOF in the hub.

Since a router is not allowed to have two connections to the same network, the router in the above slide is not allowed to have connections to two hubs. Hence, a switch is used to connect both hubs to the router.

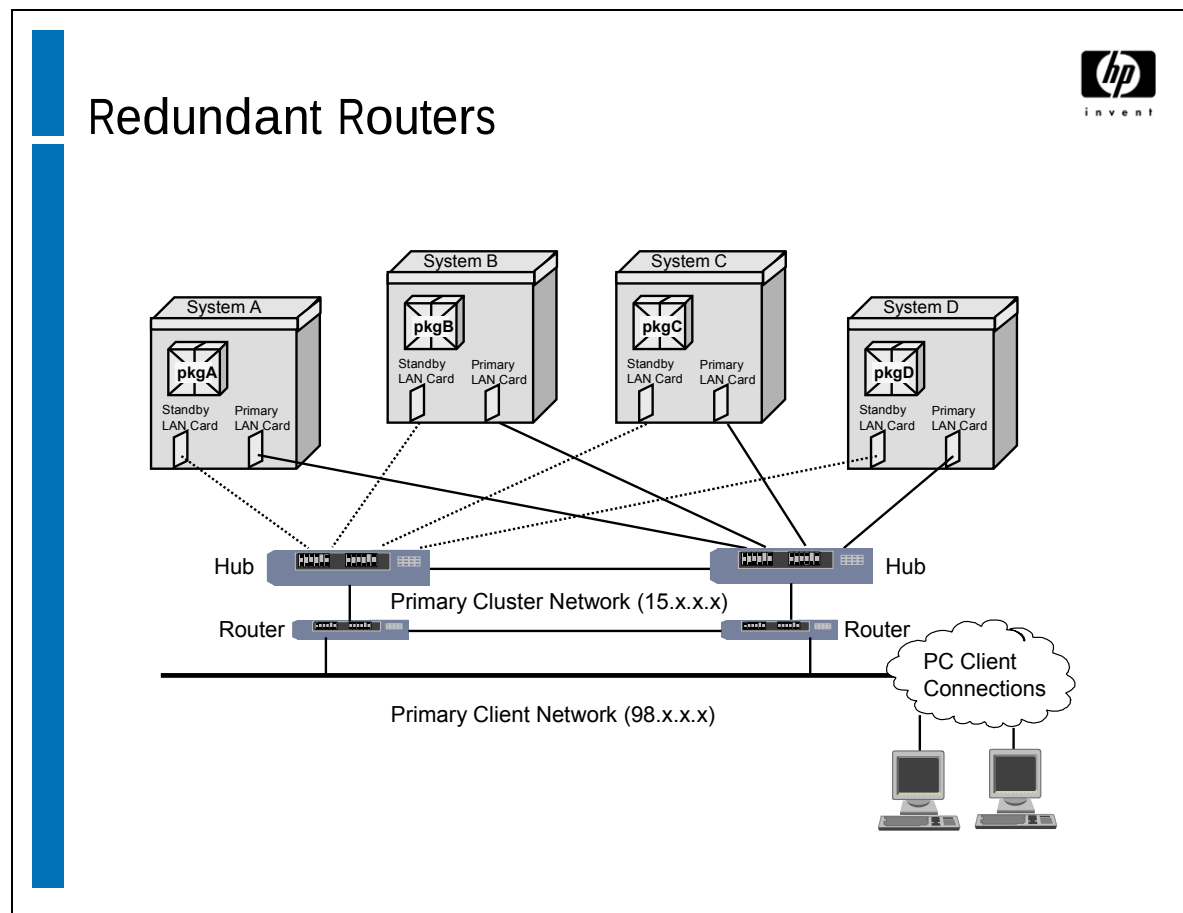
In the slide, communications among cluster nodes will survive any networking related failure, including LAN cards, twisted pair cables, hubs, or network backbone.

NOTE:

If the network backbone breaks, communication among the cluster nodes continues because the cluster nodes are attached to the same hubs. However, from the perspective of clients connecting to the cluster via the backbone, the backbone failure would cause all communications with the cluster to cease.

In this situation, a second LAN segment should be added, as shown in an upcoming slide.

15-5. SLIDE: Redundant Routers

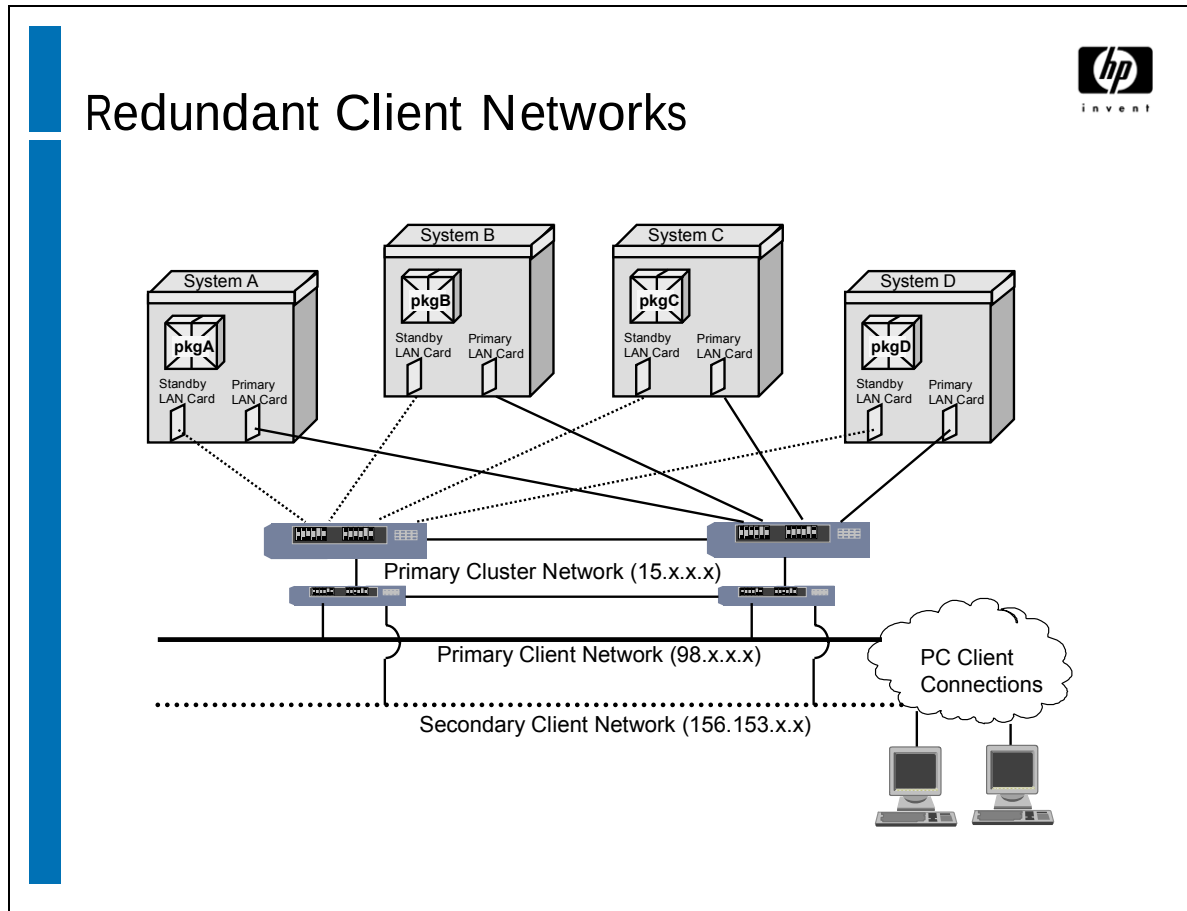


Student Notes

If a single router is used, the client / cluster communications will have a SPOF at the router. To eliminate the SPOF, a second router between the clients and the cluster is added as shown on the slide.

Each client has its own route table entry to get to the cluster. Some clients may have entries for the first router; other clients could have entries for second router. Should a router fail, routing information protocol (RIP) packets would inform any clients using the failed router to now use the other router for their cluster communications.

15-6. SLIDE: Redundant Client Networks



Student Notes

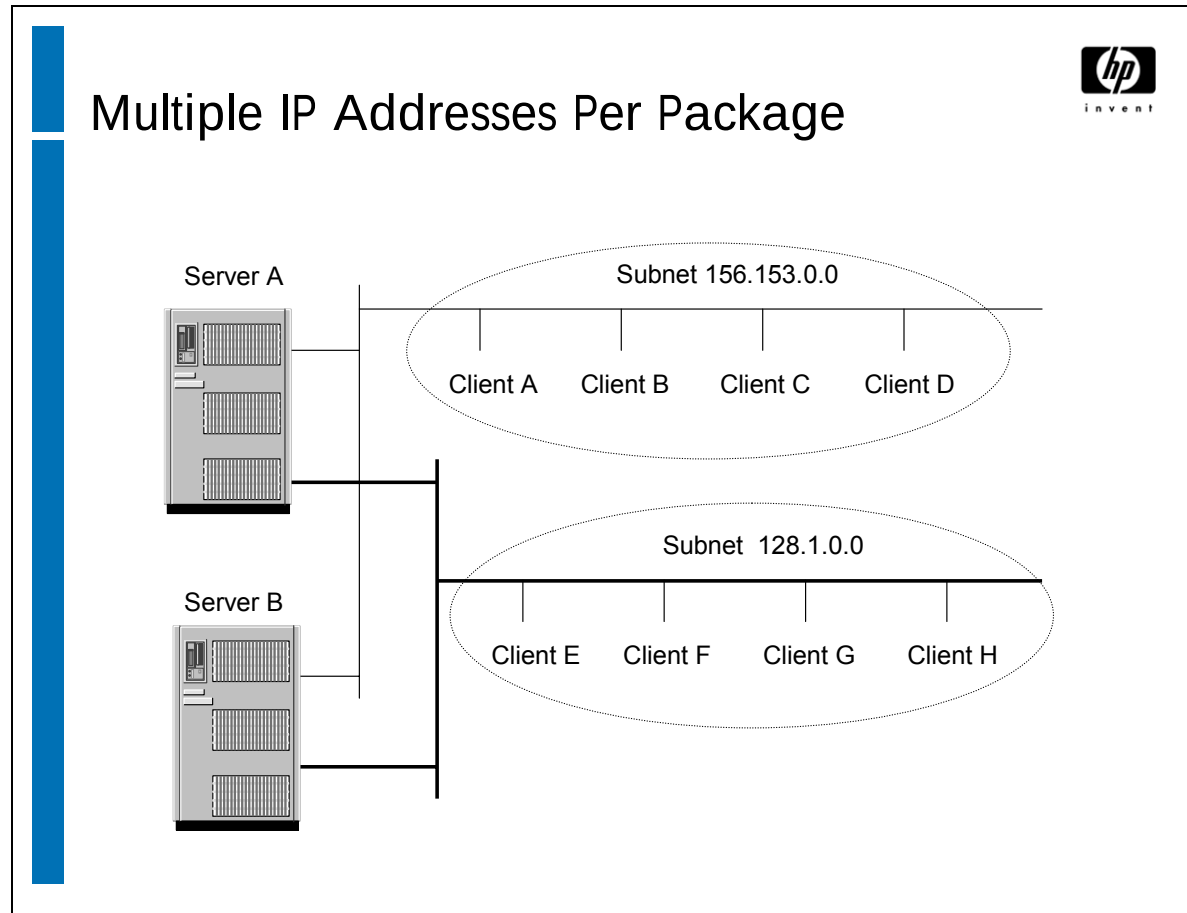
The final component that needs to be made highly available is the client network itself. This is accomplished by connecting both routers to a second network that belongs to the group of network associated with client networks.

By having multiple client networks being able to connect to the Serviceguard clustered network, the communications to the cluster no longer has a single-point-of-failure (SPOF) with the client network itself.

The above slide shows a network configuration where the four main SPOFs related to client-cluster communications have been eliminated. The four SPOFs which have been eliminated are:

- LAN Card failures
- Hub failures
- Router failures
- Client network failures

15-7. SLIDE: Multiple IP Addresses



Student Notes

A package can have multiple floating IP addresses assigned to it (on different subnets) to better balance traffic across multiple subnets. The package would have one IP address for each subnet on which clients reside. Each client would point to the floating IP address on its own subnet.

15-8. LAB: Losing Heartbeat Packets

Directions

1. With the cluster up, both nodes as members, and
tail -f /var/adm/syslog/syslog.log running on both servers ...
 - a. **If you are using remote equipment**, cd into the /labs/mod_15 and then run the "simulate_lan_card_failure3.sh" script **on one node** of your cluster. This is similar to the script that you have executed in Module 6. Here, however, this script will disable all LAN cards on one node.
 - b. **If you are using local equipment**, then disconnect **all** LAN card cables on one node from the network.

Note that both systems are up, Serviceguard daemons are running on both nodes, but neither node is receiving heartbeats (both nodes think the other node has failed). In this situation, if the cluster splits into two, only one subcluster will continue and the other subcluster will **TOC** (panic). If one set of nodes consists of more than half the cluster, these nodes adopt all applications configured for them. All other nodes (with less than half) **TOC** to protect against the same applications running on two different systems.

If the pieces are exactly 50% of the cluster (as is the case here) the lock disk is needed to break the tie. Both servers will attempt to access the lock disk. One server obtains the lock disk and continues to run the cluster as a single node cluster, the other node should **TOC**.

Answer:

```
# cd /labs/mod_15
# ./simulate_lan_card_failure3.sh -or- disconnect ALL LAN cables on one node.
```

2. After one of the nodes TOCs, ...
 - a. **If you are using remote equipment**, hit <CNTRL-C> (if needed) to exit out of the "simulate_lan_card_failure3.sh" script on the surviving node.
 - b. **If you are using local equipment**, then reconnect the LAN cables on the same node.

Answer:

<CNTRL-C> --or-- reconnect LAN cables on ServerA.

Module 16 — Rolling Upgrade Issues

Objectives

Upon completion of this module, you will be able to do the following:

- Explain why rolling upgrades are carried out.
- List provisions for rolling upgrades.
- Know when rolling upgrades will fail.
- Carry out a rolling upgrade.

16-1. SLIDE: Minimizing Planned Downtime



Minimizing Planned Downtime

- Faster system upgrades/patches
- Faster application updates/patches
- On-line application reconfiguration
- More accurate package configuration documentation
- More detailed package sign-off and testing

Student Notes

Planned downtime is scheduled. As a system administrator you strive to keep this time to an absolute minimum. The most common reasons for planned downtime are for disk management, operating system upgrades, package updates and hardware upgrades. All of these points can be addressed by various means. File system management can be done using OnLine JFS. Operating system, hardware and package updates/upgrades/maintenance can be done using rolling upgrades.

Reducing Time for Operating System/Application Upgrades

A rolling upgrade is the method used to minimize the amount of planned downtime. Rolling upgrades can take many forms. In most cases it involves transferring all packages from the node which requires maintenance to another node in the cluster. For example, in a client server model, the database server could be upgraded on one day, causing one hour of downtime. Other application servers could be upgraded one at a time throughout the week, maintaining the highly available environment.

In your initial cluster specification, you must allow for large outages caused when packages from a failed node add extra load to their adoptive nodes. Having a "spare node" in the cluster allows you to eliminate this additional load when you have to remove a node from the

cluster for upgrade. The spare node is configured as an adoptive node for all packages, so, when a node upgrade is required, you transfer its packages to the spare, and remove the node to be upgraded from the cluster. You are now free to upgrade the node and its software, do final testing and get project signoff without significantly impacting the highly available environment. When completed, bring the upgraded node back into the cluster, either as the new spare, or as the returned active node.

Documentation Package Sign Off and Testing

As the packages and cluster are left fully configured during the upgrades, it is especially important that proper package sign off and testing takes place on the operating system and the application. This should never be carried out by the department who developed the software. Since package failure in a running cluster must be detected, testing procedures may need to be modified to test all possible failure causes and detection behaviors. This is time consuming, but if not carried out, could cause more outage than would be caused by making the modifications using a rolling upgrade.

If rolling upgrades are not performed the scenario would be to bring down the entire system, upgrade every node to the new version of software and then restart the application on all affected nodes.

To minimize planned downtime requires application designers to understand how Serviceguard works so they can design their applications to answer the questions:

- Can a new version of an application be installed without scheduled downtime? In most cases this should be possible. The most common obstacle is changing the layout of data between releases.
- Can different revisions of an application operate within the system simultaneously?
- If there are problems with an application, can it roll back to a previous release?

<i>NOTE:</i>	When the data layout changes between releases, a rolling upgrade may require planned downtime to convert the data to the new format. Once the upgraded software has been brought up on the converted data, make sure the package can only run on the node that has been upgraded, until the application has been upgraded on all adoptive nodes.
--------------	--

16-2. SLIDE: Rules for Rolling Upgrade



Rules for Rolling Upgrade

- Cluster configuration files cannot be updated until all nodes are at the same version of operating system.
- All Serviceguard commands must be issued from the node with latest version of Serviceguard.
- Only two versions of Serviceguard can be implemented during a rolling upgrade.
- Only two versions of the operating system can be implemented during a rolling upgrade.
- Binary configuration files may be incompatible.
- Rolling upgrade can only be carried out on configurations that have not been modified since the last time cluster was started.
- Serviceguard cannot be removed from a node while cluster is being upgraded.
- Any new features of Serviceguard cannot be utilized until all nodes are running that version of software.
- Keep kernels consistent.
- Hardware configurations cannot be modified.

Student Notes

The following limitations apply to rolling upgrades:

- During the rolling upgrade of your Serviceguard cluster, you should only issue Serviceguard commands (other than `cmrunnode` and `cmhaltnode`) on those nodes with the latest revision of Serviceguard. *Performing any other tasks on another node will cause inconsistent results.*
- Cluster and Package configuration cannot be modified in any way until all nodes have been upgraded.
- Hardware cannot be modified.
- None of the new features of the new version of Serviceguard can be used until all nodes have been upgraded
- Binary configuration files may be incompatible between releases of Serviceguard. Manually copy these files only where a cold install of the operating system has taken place. ***Read the release notes for any scripts that must be executed with a cold install to modify package files.***

- No more than two versions of Serviceguard may be running in a cluster at any one time.
- Rolling upgrades were not designed to allow multiple versions of Serviceguard to be operating on a long term basis. It is important that all nodes in the cluster are upgraded as quickly as possible.
- Serviceguard software must not be removed (via `swremove`) from a node while the cluster is in the process of a rolling upgrade.
- Keep kernels consistent and modify any with guidance from the operating system / application provider.

NOTE: If you upgrade a cluster to A.11.16 or later, the `cmclnodelist` entries are automatically updated into Access Control Policies in the cluster configuration file. All non-root user-hostname pairs will be given the role of Monitor (view only).

16-3. SLIDE: Serviceguard Rolling Upgrades



Serviceguard Rolling Upgrades

- Use `cmhaltnode` to halt the node you wish to upgrade.
- Turn off auto cluster configuration.
- Upgrade node.
- Apply patches.
- Turn on auto cluster configuration.
- Rejoin cluster using `cmrunnode`.
- Repeat for all nodes in cluster.

Student Notes

Be sure to plan sufficient system capacity / availability to allow packages to be moved between nodes without unacceptable loss of performance.

The only nodes which can issue configuration commands within a mixed cluster (eg. Serviceguard 11.15 and Serviceguard 11.17) are those with the latest version of Serviceguard.

16-4. SLIDE: Operating System Rolling Upgrades



Operating System Rolling Upgrades

- Use `cmhaltnode` to halt node you wish to upgrade.
- Perform operating system upgrade.
- Patch operating system.
- Create all device files and reconfigure kernel.
- Install Serviceguard / add patches.
- Copy `/etc/cmcluster/cmclconfig` file from a node still running in cluster.
- Edit `/etc/rc.config.d/cmcluster` so `AUTOSTART_CMCLD=1`.
- Rejoin cluster with `cmrunnode`.
- Repeat for all other nodes in cluster.

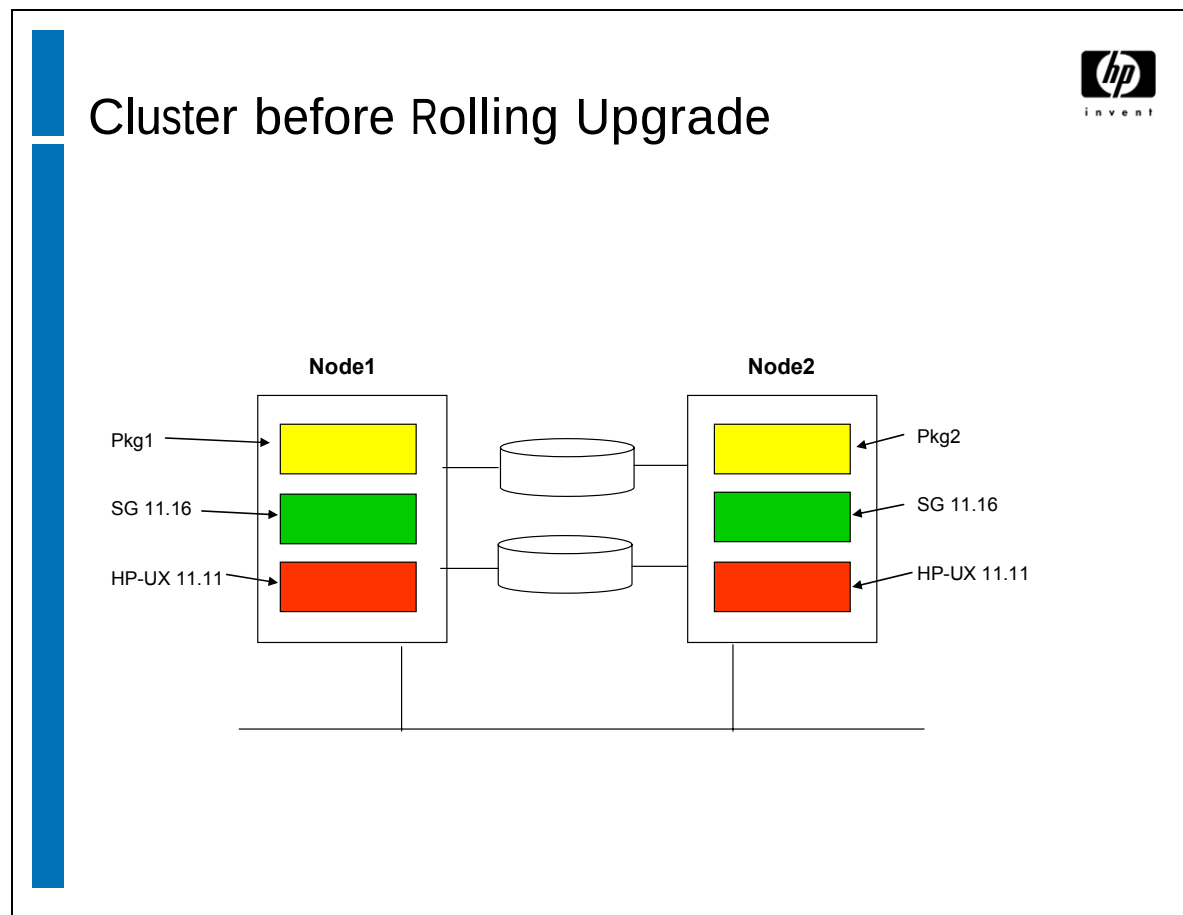
Student Notes

Operating system upgrades are supported in a cluster (via the rolling upgrade method). However, per the Serviceguard Release Notes, a cold install of the operating system is not supported.

On a cold install (as an alternate method of updating the operating system on a node), instance numbers can (and probably will) change, which will likely cause shared disk device file consistency issues.

Note: If a cold install is necessary, be sure to print a full `ioscan` of your system prior to re-installing. Armed with this `ioscan` listing, and referring back to the High Availability Planning module (module 3), you can alter instance numbers after the re-install as necessary.

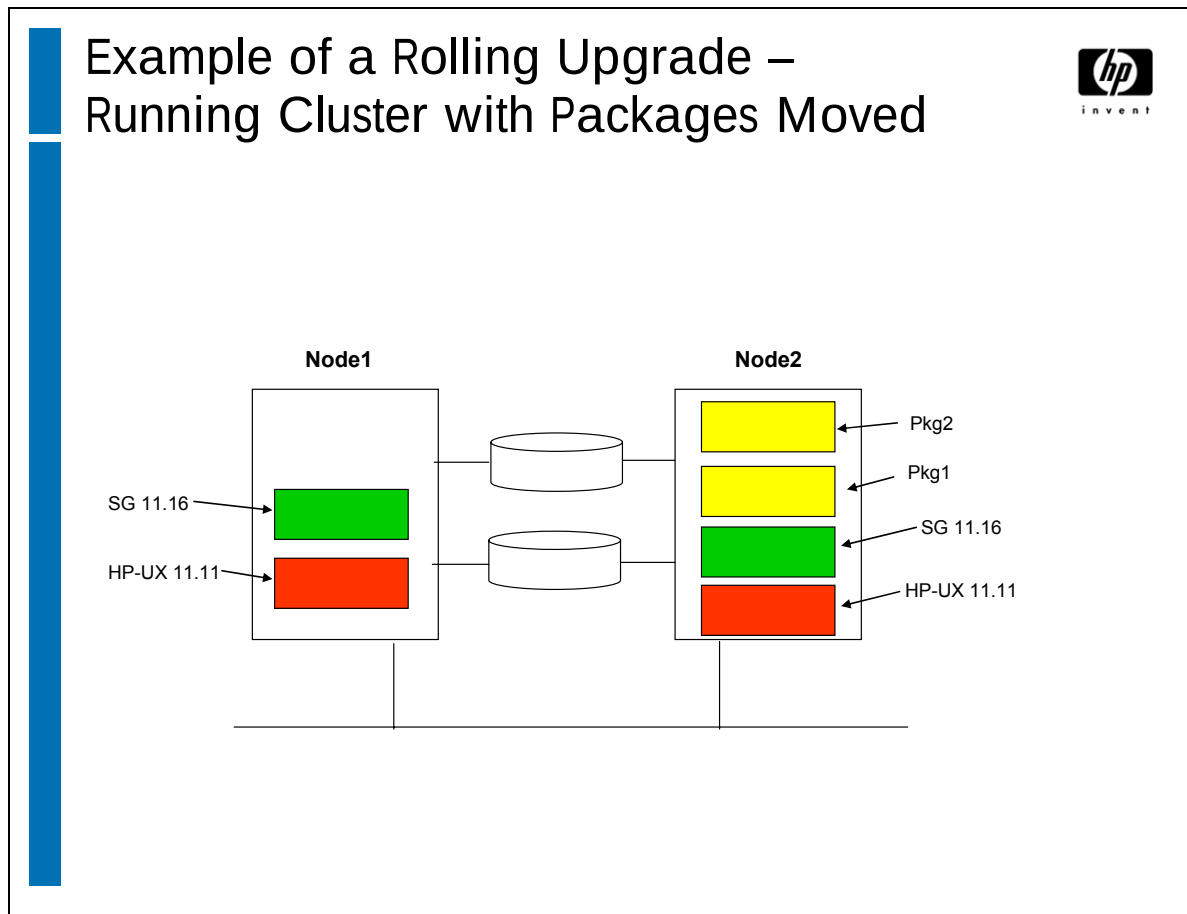
16-5. SLIDE: Cluster before Rolling Upgrade



Student Notes

- Get all kernel parameters.
sysdef | lp
kctune | lp
- Get all volume groups and logical volume information.
vgdisplay -v | lp
lvdisplay /dev/vgXX/lvolYY
- Halt the node.

16-6. SLIDE: Example of a Rolling Upgrade — Running Cluster with Packages Moved

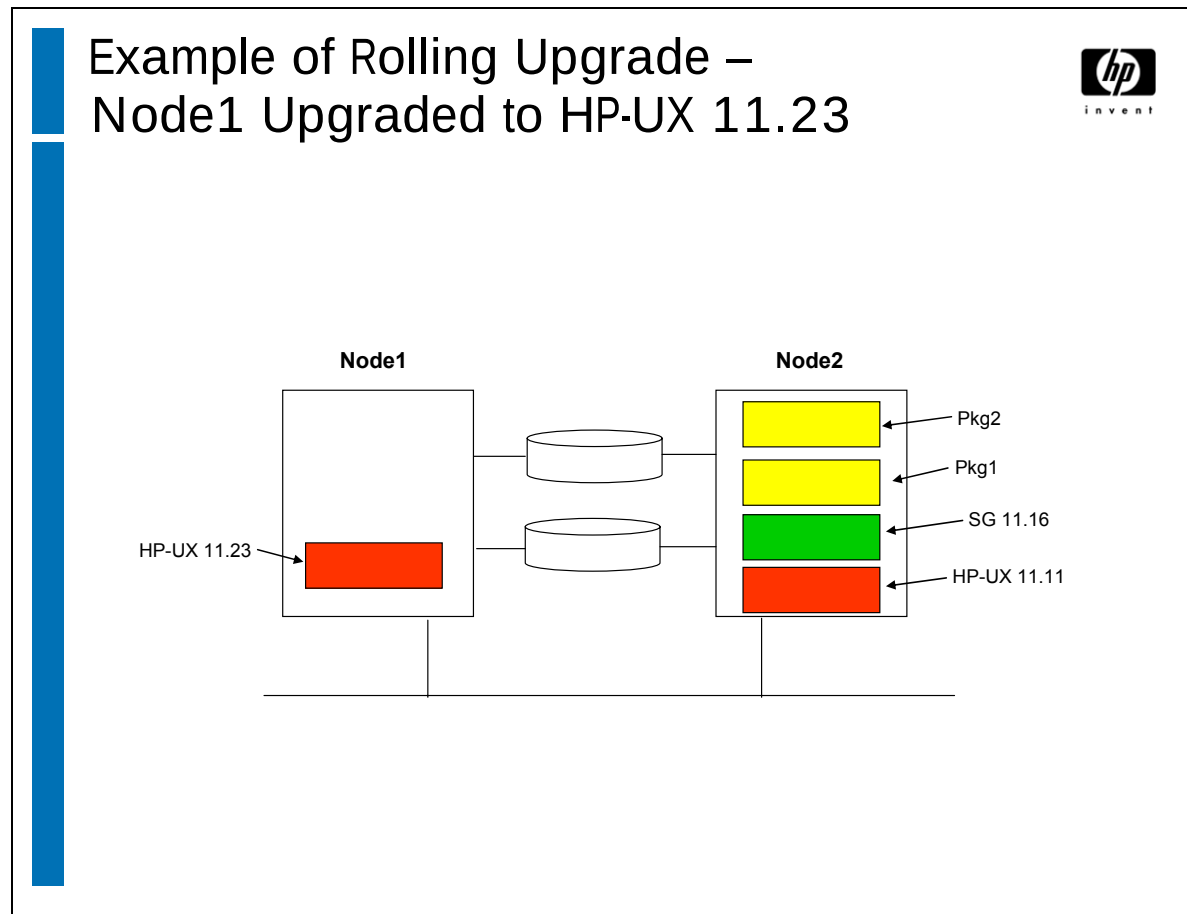


Student Notes

All packages are up on node2.

- Backup node1.
- If this is a cold install, shutdown the machine.
- If this is an upgrade, read the release notes to identify special actions required.

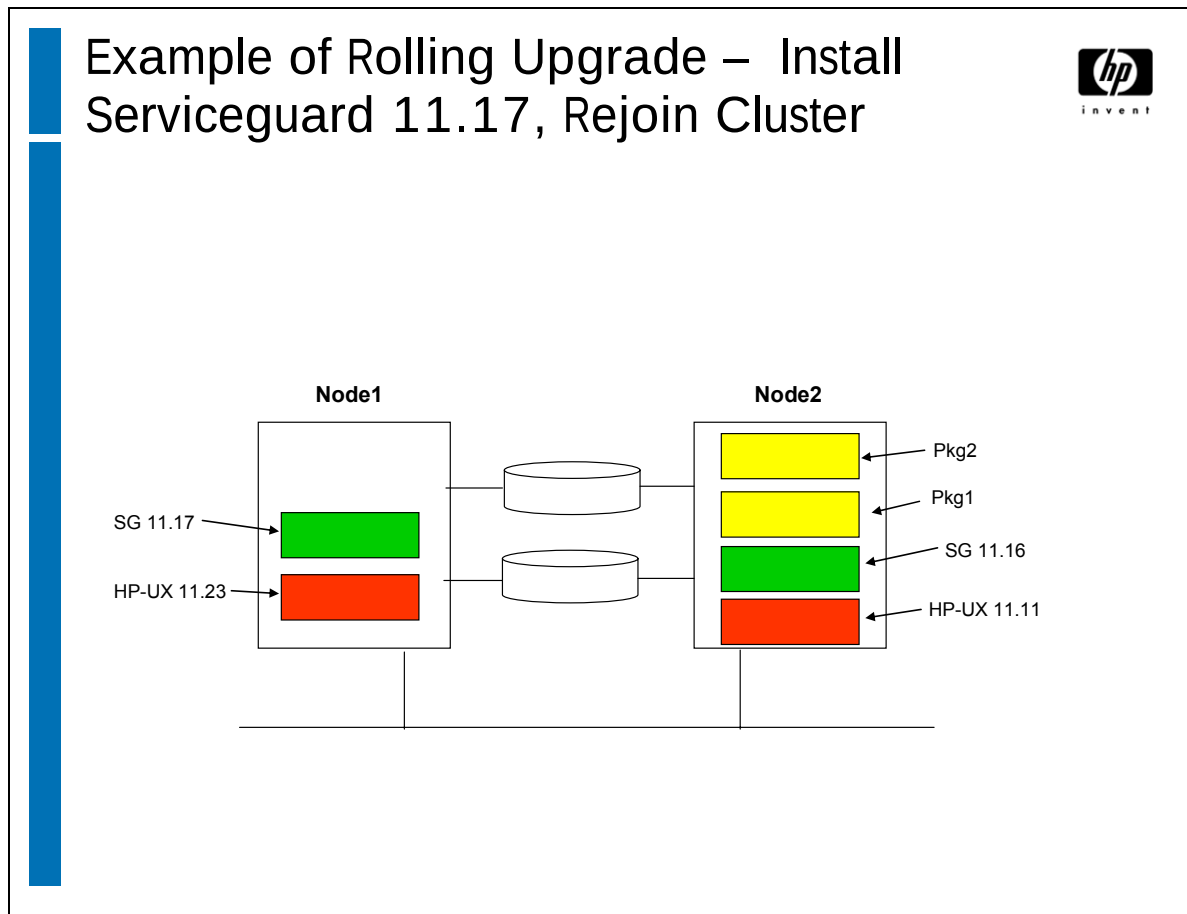
16-7. SLIDE: Example of Rolling Upgrade — Node1 Upgraded to HP-UX 11.23



Student Notes

- Update operating system.
- Patch system.
- Install correct version of Serviceguard.
- Set all kernel parameters.
- Add any drivers/subsystems to the kernel.

16-8. SLIDE: Example of Rolling Upgrade — Install Serviceguard 11.17, Rejoin Cluster



Student Notes

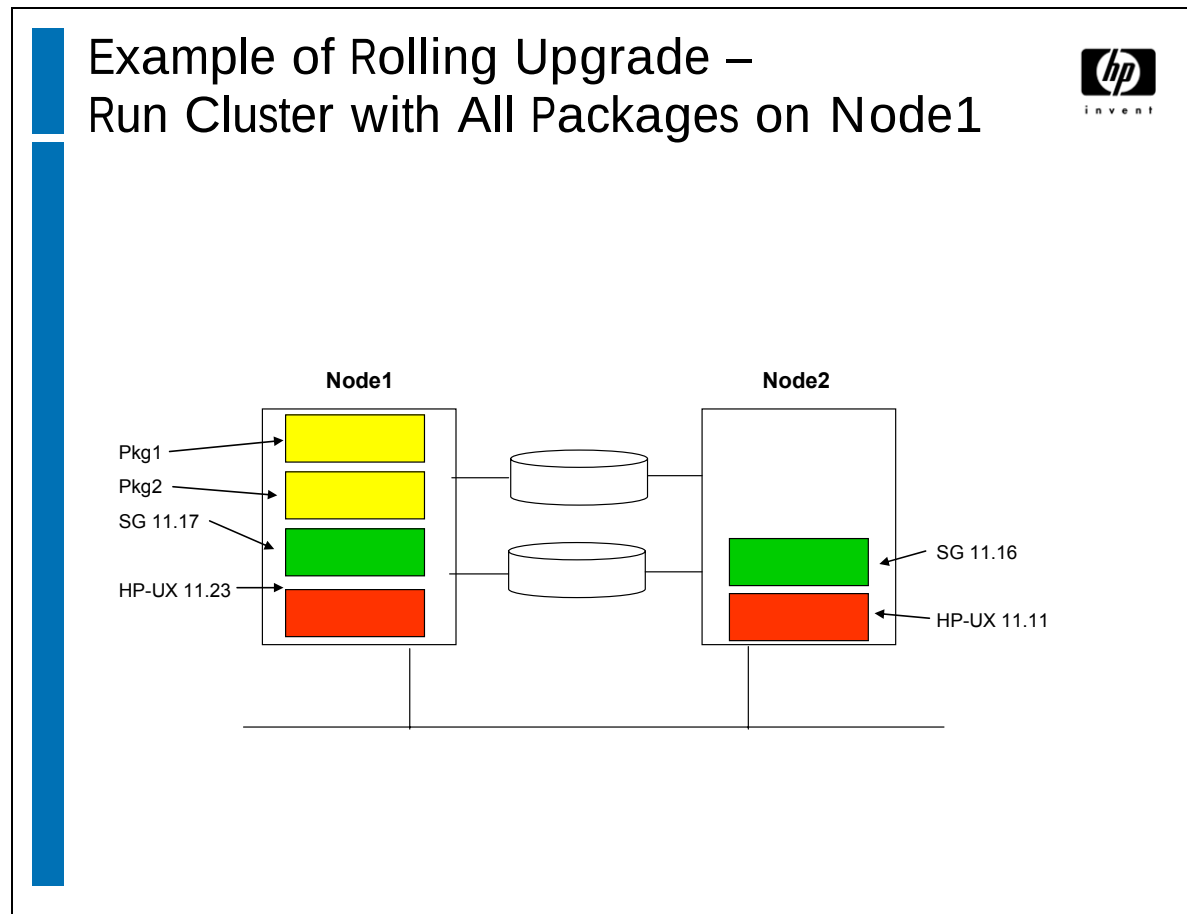
- Upgrade any applications that require modification.

NOTE: All static data should be in `/opt/<app>` and all node specific data should be in `/etc/opt/<app>`.

You should be able to copy `/opt/<app>` to any node, as this should be node-independent if designed correctly.

- Use standard sign off procedures with test data to ensure the node is working correctly. You should have standard test configurations that can be applied to allow sign off of operating system and application modifications.
- Backup node.

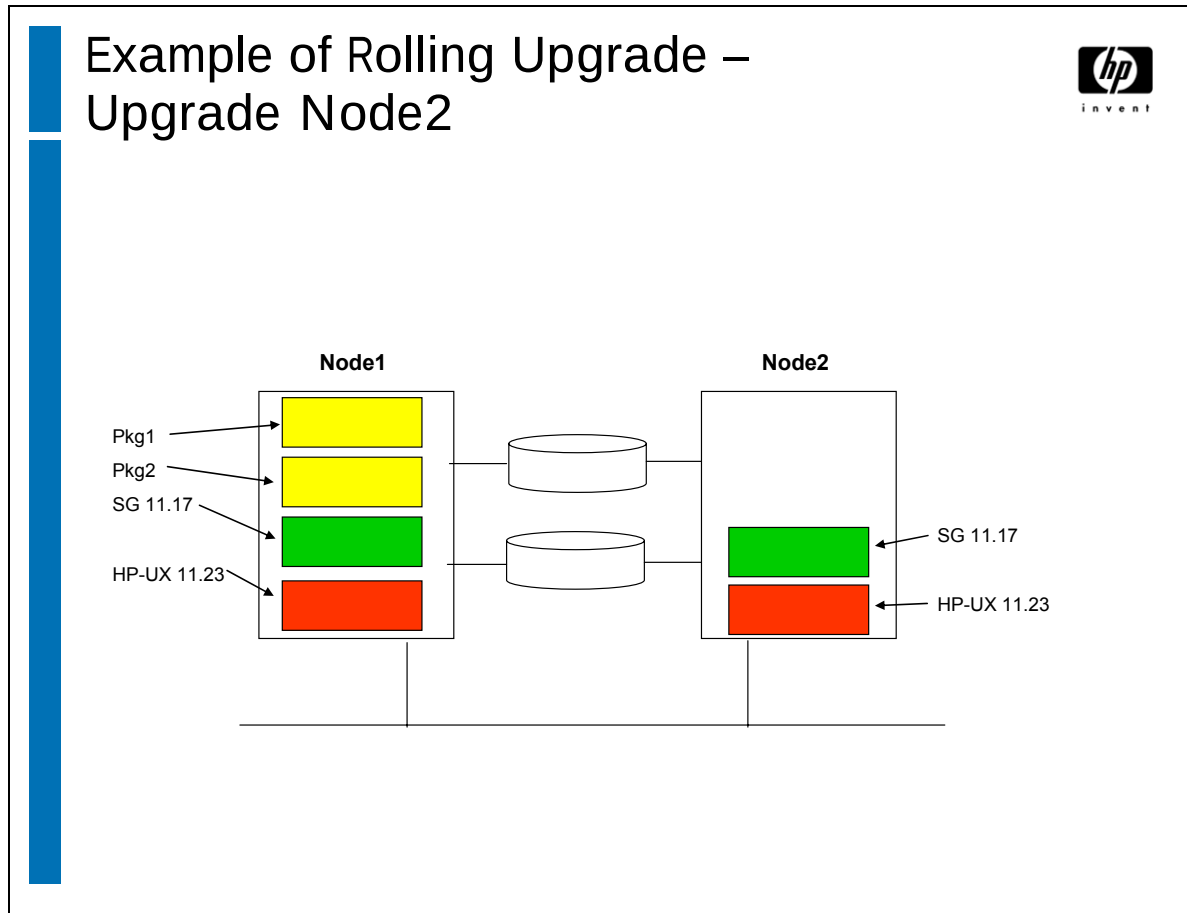
16-9. SLIDE: Example of Rolling Upgrade — Run Cluster with all Packages on Node1



Student Notes

- Add node to cluster.
cmrunnode -v node1
- Observe log files and check cluster.
- Move all packages to node1.
cmhaltnode -v -n node2

16-10. SLIDE: Example of Rolling Upgrade — Upgrade Node2



Student Notes

- Upgrade operating system on node2 to HP-UX 11.23.
- Apply operating system patches as necessary.
- Install Serviceguard 11.17.
- Apply Serviceguard patches as necessary.
- Return node2 to the running cluster.

```
# cmrunnode -v <node2>
```

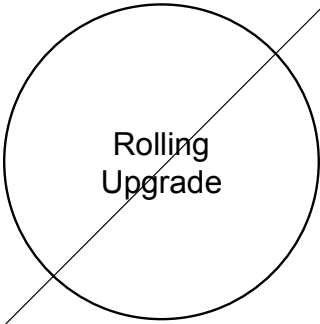
- Move Pkg2 back to its original node.

```
# cmhaltpkg -v Pkg2  
# cmrunpkg -v -n <node2> Pkg2
```

16-11. SLIDE: When a Rolling Upgrade Is Not Possible



When a Rolling Upgrade Is Not Possible

Major database conversion required = 

Student Notes

If a major database conversion is required to update to the new version, a rolling upgrade is not possible because the application will need to be down for an extended period to perform the database conversion.

16-12. LAB: Performing a Rolling Upgrade

Directions: Install the Auto-Port Aggregation Software

Special Note

In this lab, we install the Auto-Port Aggregation software, part number J4240AA, version B.11.23.05 or later. We will not actually use this software, however. We are swinstall'ing it only for the purpose of learning the "rolling upgrade" technique.

You may find that there is insufficient disk space in the /var file system when installing the Auto-Port Aggregation software. This may be the result of crash dumps created on /var/adm/crash from earlier modules. You may simply remove these crash dump files and then try installing the Auto-Port Aggregation software again.

Additionally, because of the dependency built in to the "nfs" package in Module 12, the "test" package cannot run on the same node with the "nfs" package during the rolling upgrade unless the "nfs" package starts up before the "test" package. You may choose to use the steps learned in Module 12 to undo this dependency and properly perform the rolling upgrade.

Preparation for the Rolling Upgrade and Assumptions

In this lab, we assume that the cluster is up and running, and that all packages are up and running on their respective primary nodes.

1. Write the hostname of all nodes in your cluster in the order that you wish to upgrade them.

<Node1>

<Node2>

Upgrade the First Node in Your Cluster

2. Halt each package currently running on <Node1> (as noted above in step 1).

3. Next, verify that <Node1> is now up and running in the cluster, but with no packages.
4. Now, restart each of <Node1>'s packages on its respective adoptive node.
5. Now that each package (previously halted in step 2 above) is running on its first adoptive node (step 4 above), re-enable the global package switching parameter for each of these packages. We do this so that the package can still failover if necessary.
6. Also, for each of the above packages, turn off the "local" switch for the node being upgraded (i.e., the primary node for each of the above packages). We do this to ensure that the package does not attempt to failback to the node being upgraded while we are in the middle of the upgrade.
7. Again, verify that all packages are now running on the first adoptive nodes, AND that the node to be upgraded is running with zero packages.
8. Next, halt cluster services on the node to be upgraded (thus removing it from the cluster).
9. Again, verify that this node has been removed from the cluster.
10. Set AUTOSTART_CMCLD to 0 in the /etc/rc.config.d/cmcluster file. This will ensure that, if the node is rebooted as part of the upgrade, the node will **not** rejoin the cluster after the reboot.

AUTOSTART_CMCLD=0

11. Now, do **ONE** of the following steps: either 11.a OR 11.b (but not both).

- a. Perform the upgrade of this node by using NON-INTERACTIVE "swinstall".

```
# swinstall -s /var/spool/sw -x autoreboot=true J4240AA
```

where J4240AA is the part number for the Auto-Port Aggregation software.

- b. OR, perform the upgrade of this node by using INTERACTIVE "swinstall".

```
# swinstall
```

- (1) At the "Specify Source" window, accept the default hostname and default depot name (/var/spool/sw).
- (2) Then, at the "Software Selection" window, select the "Auto-Port Aggregation J4240AA, version B.11.23.05 (or later) product.
- (3) From this point, proceed as you would normally for any "swinstall". Note that, after the analysis phase, and as the installation has begun, you will see a message that says "The system will be rebooted as soon as installation is complete." When you see this, proceed with the installation, including the reboot.

12. After the system has rebooted, and after logging in, check to be sure that this node has NOT rejoined the cluster.

13. Now that <Node1> has been successfully upgraded, add it back into the cluster. Verify that it is now part of the cluster.

14. Re-enable AUTOSTART_CMCLD on this node.

```
AUTOSTART_CMCLD=1
```

Rolling Upgrade Issues

15. Return those packages currently running on the <node2> back to this node, and reset the AUTO_RUN and local package switching parameters (if necessary). Use `cmhaltpkg`, `cmrunpkg` and `cmmodpkg` to do this.

16. Congratulations on successfully upgrading the first node in your cluster!

17. To see if the Auto-Port Aggregation software has indeed been installed, execute the following:

```
# swlist -l bundle | grep -i auto
```

Upgrade <Node2>

18. Repeat steps 2 through 17 on <node2>.

19. Congratulations on successfully upgrading your entire cluster using the "rolling upgrade" technique!!

16-13. LAB Solution: Performing a Rolling Upgrade

Directions: Install the Auto-Port Aggregation Software

Special Note

In this lab, we install the Auto-Port Aggregation software, part number J4240AA, version B.11.23.05 or later. We will not actually use this software, however. We are swinstall'ing it only for the purpose of learning the "rolling upgrade" technique.

You may find that there is insufficient disk space in the /var file system when installing the Auto-Port Aggregation software. This may be the result of crash dumps created on /var/adm/crash from earlier modules. You may simply remove these crash dump files and then try installing the Auto-Port Aggregation software again.

Preparation for the Rolling Upgrade and Assumptions

In this lab, we assume that the cluster is up and running, and that all packages are up and running on their respective primary nodes.

1. Write the hostname of all nodes in your cluster in the order that you wish to upgrade them.

<Node1>

<Node2>

Upgrade the First Node in Your Cluster

2. Halt each package currently running on <Node1> (as noted above in step 1).

Answer:

```
# cmhaltpkg -v package1
# cmhaltpkg -v package2
# cmhaltpkg -v package3
```

< etc >

3. Next, verify that <Node1> is now up and running in the cluster, but with no packages.

Answer:

```
# cmviewcl -v
```

4. Now, restart each of <Node1>'s packages on its respective adoptive node.

Answer:

```
# cmrunpkg -v -n <first_adoptiv_node> package1
# cmrunpkg -v -n <first_adoptiv_node> package2
# cmrunpkg -v -n <first_adoptiv_node> package3

< etc >
```

5. Now that each package (previously halted in step 2 above) is running on its first adoptive node (step 4 above), re-enable the global package switching parameter for each of these packages. We do this so that the package can still failover if necessary.

Answer:

```
# cmmodpkg -v -e package1
# cmmodpkg -v -e package2
# cmmodpkg -v -e package3

< etc >
```

6. Also, for each of the above packages, turn off the "local" switch for the node being upgraded (i.e., the primary node for each of the above packages). We do this to ensure that the package does not attempt to failback to the node being upgraded while we are in the middle of the upgrade.

Answer:

```
# cmmodpkg -v -d -n < primary_node > package1
# cmmodpkg -v -d -n < primary_node > package2
# cmmodpkg -v -d -n < primary_node > package3

< etc >
```

7. Again, verify that all packages are now running on the first adoptive nodes, AND that the node to be upgraded is running with zero packages.

Answer:

```
# cmviewcl -v
```

8. Next, halt cluster services on the node to be upgraded (thus removing it from the cluster).

Answer:

```
# cmhaltnode -v node_to_be_upgraded
```

9. Again, verify that this node has been removed from the cluster.

Answer:

```
# cmviewcl -v
```

10. Set AUTOSTART_CMCLD to 0 in the /etc/rc.config.d/cmcluster file. This will ensure that, if the node is rebooted as part of the upgrade, the node will **not** rejoin the cluster after the reboot.

```
AUTOSTART_CMCLD=0
```

11. Now, do **ONE** of the following steps: either 11.a OR 11.b (**but not both**).

- a. Perform the upgrade of this node by using NON-INTERACTIVE "swinstall".

```
# swinstall -s /var/spool/sw -x autoreboot=true J4240AA
```

where J4240AA is the part number for the Auto-Port Aggregation software.

- b. OR, perform the upgrade of this node by using INTERACTIVE "swinstall".

```
# swinstall
```

- (4) At the "Specify Source" window, accept the default hostname and default depot name (/var/spool/sw).

- (5) Then, at the "Software Selection" window, select the "Auto-Port Aggregation J4240AA, version B.11.23.05 (or later) product.

- (6) From this point, proceed as you would normally for any "swinstall". Note that, after the analysis phase, and as the installation has begun, you will see a message that says "The system will be rebooted as soon as installation is complete." When you see this, proceed with the installation, including the reboot.

12. After the system has rebooted, and after logging in, check to be sure that this node has NOT rejoined the cluster.

Answer:

```
# cmviewcl -v
```

13. Now that <Node1> has been successfully upgraded, add it back into the cluster. Verify that it is now part of the cluster.

Answer:

```
# cmrunnode -v node_being_upgraded
```

14. Re-enable AUTOSTART_CMCLD on this node.

```
AUTOSTART_CMCLD=1
```

15. Return those packages currently running on the <node2> back to this node, and reset the AUTO_RUN and local package switching parameters (if necessary). Use cmhaltpkg, cmrunpkg and cmmodpkg to do this.

16. Congratulations on successfully upgrading the first node in your cluster!

17. To see if the Auto-Port Aggregation software has indeed been installed, execute the following:

Answer:

```
# swlist -l bundle | grep -i auto
```

Upgrade <Node2>

18. Repeat steps 2 through 17 on <node2>.

19. Congratulations on successfully upgrading your entire cluster using the "rolling upgrade" technique!!

Module 17 — LVM Maintenance for Packages

Objectives

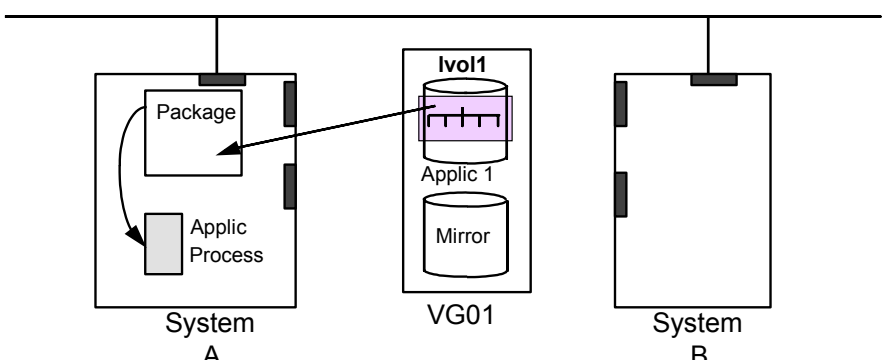
Upon completion of this module, you will be able to do the following:

- Add / remove a disk to/from a volume group owned by a package.
- Add / remove a logical volume to/from a volume group owned by a package.
- Add / remove a volume group to/from a package.
- Make a logical volume and file system (in a volume group owned by a package) larger or smaller.

17-1. SLIDE: Make a Logical Volume / File System Larger or Smaller

Make a Logical Volume / File System Larger or Smaller





The diagram illustrates a shared Logical Volume (VG01) between two systems, System A and System B. System A contains a Package and an Applic Process. VG01 contains lvol1 (Applic 1) and Mirror. System B is empty. A text box below the diagram asks: 'The /dev/vg01/lvol1 file system is full. How do we grow the logical volume so both systems recognize the new size?'

Student Notes

Using OnLine JFS, we can resize the file system using the standard tools.

If **Applic1** is running on SystemA to make the file system bigger we issue:

```
# lvextend -L newsize_in_MB /dev/vg01/sglvol1
# fsadm -F vxfs -b xxxM /sgdata1 (where xxx is in MB)
```

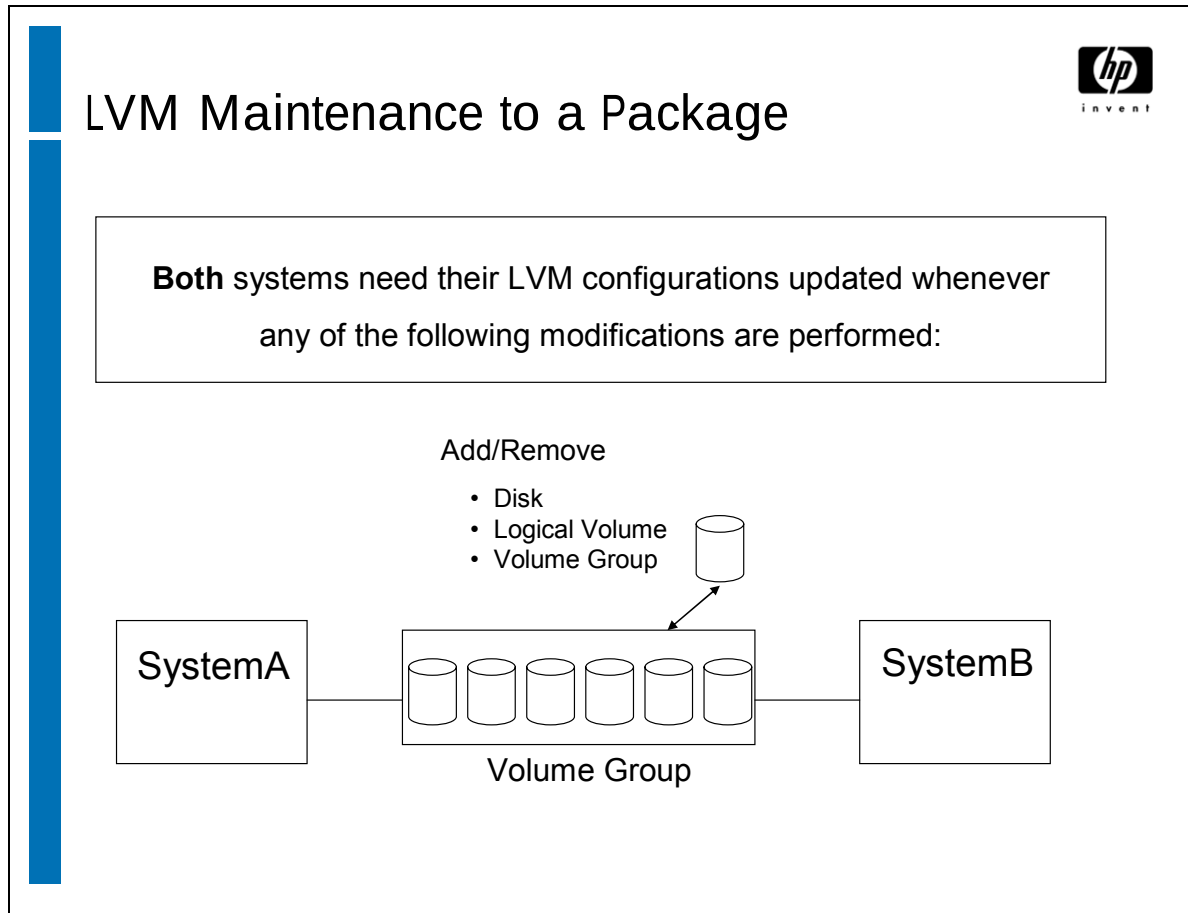
or to make it smaller we issue:

```
# fsadm -F vxfs -b xxxM /sgdata1 (where xxx is in MB)
# lvreduce -L newsize_in_MB /dev/vg01/sglvol1
```

NOTE: There is no need to issue any commands on the adoptive system.

When SystemB adopts this application the new size will be recorded in the reserved areas on the disk and will automatically be known to SystemB.

17-2. SLIDE: LVM Maintenance to a Package



Student Notes

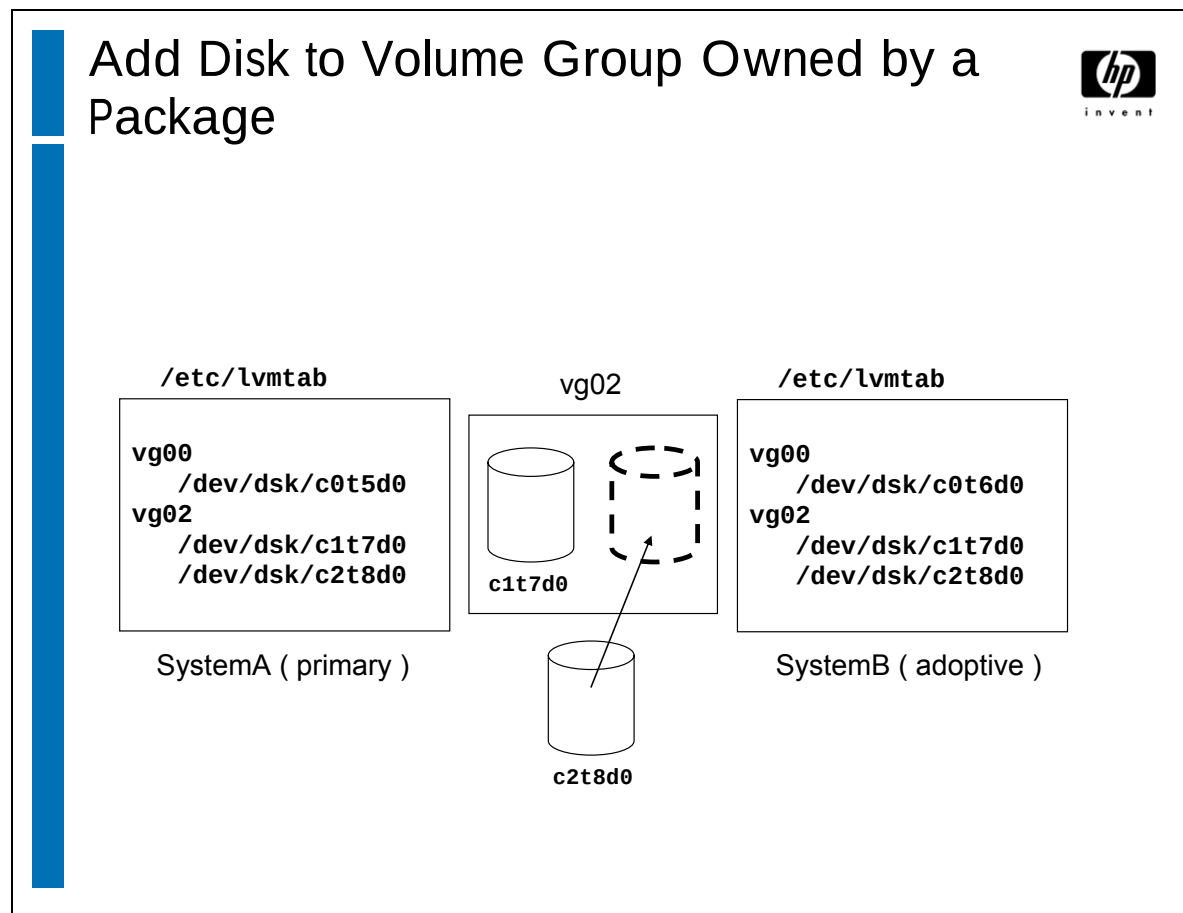
When multiple nodes are placed into a Serviceguard cluster, the procedure for performing a variety of administrative tasks must be modified to address the clustered environment.

In all three cases,

- adding or removing a disk from a volume group owned by a package,
- adding or removing a logical volume from a volume group owned by a package, and
- adding or removing a volume group from a package,

changes must be made to both the primary and adoptive systems. If this is not done properly, package activation may fail on the adoptive system.

17-3. SLIDE: Add Disk to Volume Group Owned by a Package



Student Notes

When a disk is added to a volume group with the **vgextend** command, `/etc/lvmtab` is updated with the additional disk. If we execute the **vgextend** on the primary, `/etc/lvmtab` on the primary gets updated correctly. The `/etc/lvmtab` on the adoptive system does not automatically get updated. To update `/etc/lvmtab` on the adoptive system we must `vgimport` this volume group after it is extended. This will require issuing a `vgexport` for it, and then recreating the volume group subdirectory in `/dev` and recreating the group file. If the map file no longer exists, we will need to recreate that also.

Special Note

In the following example, we are discussing volume group `vg02`. The Serviceguard administrator should bring down the package that uses `vg02`. If the administrator does not do this, and if the package should failover during the following process, then `vg02` could be left in an inconsistent state.

For example, if we are adding disk `/dev/dsk/c2t8d0` to volume group `vg02`:

- On the primary node:

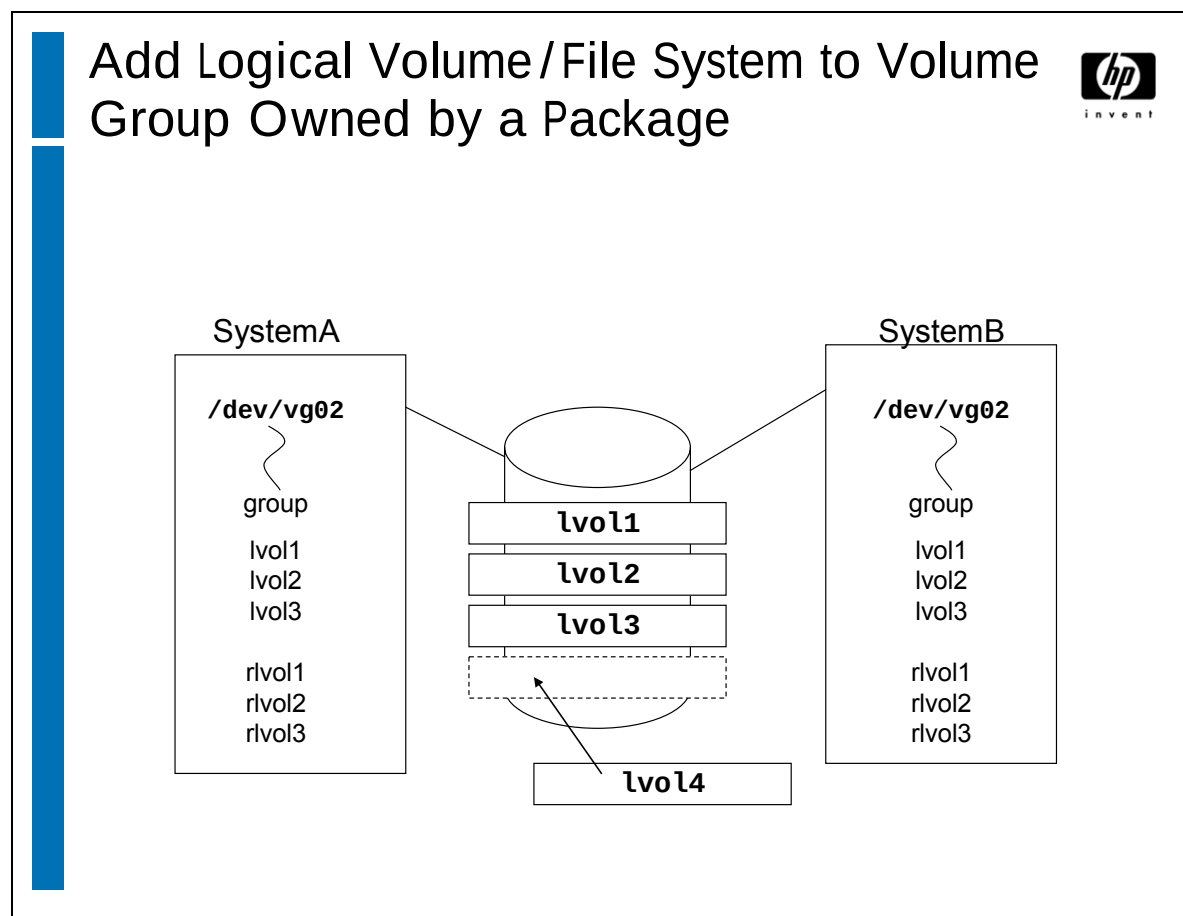
```
# cmhaltpkg -v package_name      ← The package using volume group vg02
# vgchange -a e vg02
# pvcreate [ -f ] /dev/rdisk/c2t8d0
# vgextend /dev/vg02 /dev/dsk/c2t8d0
# vgexport -v -p -s -m /etc/lvmconf/vg02.map vg02
# vgchange -a n vg02
```

- On the adoptive node:

```
# cd /etc/lvmconf
# rcp <primary_node>:/etc/lvmconf/vg02.map .
# vgexport -v vg02
# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000
# vgimport -v -s -m /etc/lvmconf/vg02.map vg02
# vgchange -a r vg02
# vgcfgbackup vg02
# vgchange -a n vg02
```

The same issue exists when removing a disk from a volume group. The **vgreduce** command will update `/etc/lvmtab` on the primary system but not on the adoptive system. So again, we must issue a `vgimport` command for the volume group on the adoptive system after issuing a `vgreduce` command for the primary.

17-4. SLIDE: Add Logical Volume / File System to Volume Group Owned by a Package



Student Notes

We can add a logical volume to a volume group with the `lvcreate` command. This command automatically creates a block and raw device file for the newly created logical volume on the primary system, but not on the adoptive system. So again, it is necessary to `re-vgimport` the volume group on the adoptive system after creating the new logical volume on the primary so that the new device files get created on the adoptive system.

Also, the package run/halt script needs to be modified if this logical volume will hold a file system, so that when the package starts this new file system is mounted and when the package halts this new file system is unmounted.

Special Note

Once again, in the following example, we are discussing volume group `vg02`. The Serviceguard administrator should bring down the package that uses `vg02`. If the administrator does not do this, and if the package should failover during the following process, then `vg02` could be left in an inconsistent state.

- On the primary:

```
# cmhaltpkg -v package_name
# vgchange -a e vg02
# lvcreate -L size -n lvol_name /dev/vg02
# newfs -F vxfs /dev/vg02/rlvol_name
# mkdir /new_mount_point
# vgexport -v -s -p -m /etc/lvmconf/vg02.map vg02
    (this will create a new map file that now includes new logical volume)
# vgchange -a n vg02
```

- On the adoptive:

```
# cd /etc/lvmconf
# rcp <primary_node>:/etc/lvmconf/vg02.map .
# mkdir /new_mount_point (needed on both systems)
# vgexport -v /dev/vg02
# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000
# vgimport -v -s -m /etc/lvmconf/vg02.map vg02
# vgchange -a r vg02
# vgcfgbackup vg02
# vgchange -a n vg02
```

Do not forget that we must halt the package with `cmhaltpkg`. (If the package control script needs to change it is recommended that the package be halted to avoid timing problems.)

Modify the control script on both systems.

After the scripts are changed on both systems, the package can be restarted with `cmrunpkg`.

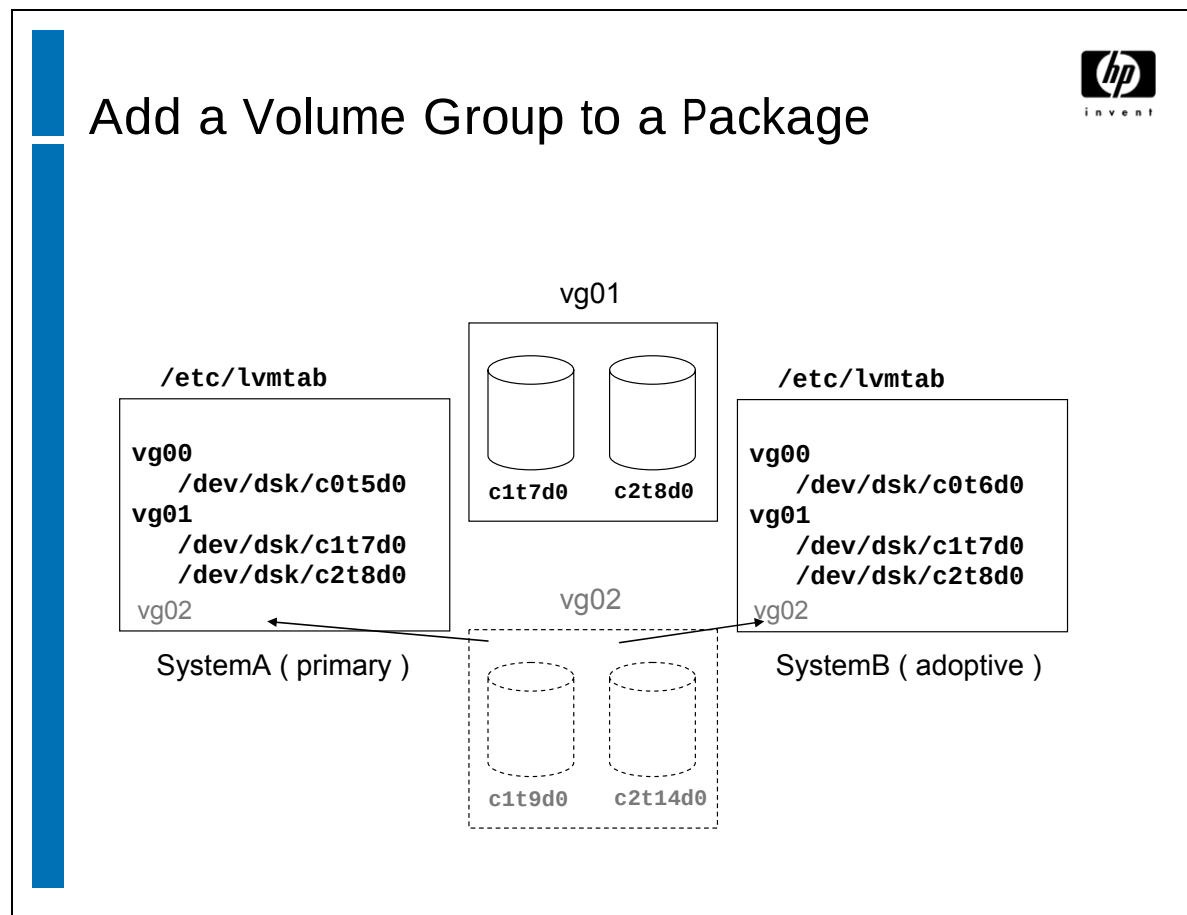
Re-enable package switching with `cmmodpkg -e`. (`cmhaltpkg` disables package switching)

Verify new file system has been mounted properly and is accessible on the primary.

You may want to test the adoptive system by failing the package over to the adoptive node and verifying the new file system is mounted and accessible on the adoptive node.

NOTE:	Removing a logical volume requires using the <code>lvremove</code> command. Again, the device files are removed on the primary but not on the adoptive node so we must <code>re-vgimport</code> . Also, the scripts must be changed so we do not try to mount / unmount the file system that has been removed. Again, the package should be halted before the control scripts are changed and then the package can be restarted after the scripts are changed on both systems.
--------------	--

17-5. SLIDE: Add a Volume Group to a Package



Student Notes

When the **vgcreate** command is issued on the primary node to create a new volume group, `/etc/lvmtab` on the primary gets updated but `/etc/lvmtab` on the adoptive node does not get updated. So again it is necessary to **vgimport** on the adoptive node.

Special Note

Once again, in the following example, we are discussing volume group `vg02`. The Serviceguard administrator should bring down the package that will use `vg02`. If the administrator does not do this, and if the package should failover during the following process, `vg02` could be left in an inconsistent state.

- First, on the primary node:

```
# pvcreate [ -f ] new_disk1
# pvcreate [ -f ] new_disk2
# mkdir /dev/vg_newone
# mknod /dev/vg_newone/group c 64 0x140000 ← Note minor number
# vgcreate vg_newone /dev/dsk/new_disk1 /dev/dsk/new_disk2
```

At this point, you would create any needed logical volumes and file systems for the volume group.

```
# vgexport -v -p -s -m /etc/lvmconf/vg_newone.map vg_newone
```

- Next, on the adoptive node:

```
# cd /etc/lvmconf
# rcp <primary_node>:/etc/lvmconf/vg_newone.map .
# mkdir /dev/vg_newone
# mknod /dev/vg_newone/group c 64 0x140000
# vgimport -v -s -m /etc/lvmconf/vg_newone.map vg_newone
```

We would now create any needed mount points for file systems.

- Now, back on the primary node:

```
# vgchange -c y vg_newone
```

We then mark the new volume group as being owned by the cluster. Unmount all file systems in the new volume group, deactivate the volume group, and halt the package that will own the new volume group.

```
# vgchange -a n vg_newone
# cmhaltpkg
```

- Modify the control scripts on both systems to insure this new vg is activated when the package starts and to insure all new file systems in this new volume group are mounted.

```
# cmrunpkg -v package_name
# cmmodpkg -v -e package_name
```

The new volume group should be activated and all new file systems in this vg should be mounted and accessible where the package is running.

Again, failing over the package onto the adoptive node should verify the new vg and file systems are available on the adoptive node.

If a volume group needs to be removed from a package we can stop the package, modify the control scripts, and restart the package. This would leave the vg on the systems, but remove it from the package.

If the volume group also needs to be removed from the systems, we can issue `vgexport` on both systems.

17-6. LAB: Increasing the Size of a Logical Volume / File System

Make the /sgdata1 file system larger. Recall that /sgdata1 is used by the test package.

17-7. LAB: Add and Remove an Unused Disk to Volume Group "vg01", Owned by the "test" Package

NOTE: This lab exercise assumes that the "test" package is running on its primary node and all switches are enabled. If necessary, make the appropriate changes now to ensure the above.

1. Make a note of an unused disk.
2. Make a note of the volume group owned by the **test** package.
3. Make a note of the disks currently included in this volume group.
4. Add the unused disk to the volume group.
5. Create an additional mirror copy of `/dev/vg01/sglvol1` on this newly added disk.
6. If the package failed over right now, would the new disk and its new mirror copy be accessible on the adoptive system? Has it been added to `/etc/lvmtab` on the adoptive system? Let's find out.

Failover the package onto the adoptive system. Issue an `lvdisplay` on the logical volume with the extra mirror copy. Is the extra mirror copy current and accessible? Why not?

7. Move package back to the primary and re-enable all package switching.
8. Re-vgimport the vg01 volume group on adoptive node, recreating map file if necessary. (Remember to vgexport on the adoptive node first).
9. Is new disk reflected in /etc/lvmtab on the adoptive node?
10. Failover the package again. Issue lvdisplay on the new logical volume. Is the extra mirror copy current and accessible?
11. Take all steps necessary to move the “test” package back to its primary node and then lvreduce the extra mirror copy from extra disk, and remove the extra disk from the volume group on the primary.
12. Does /etc/lvmtab on the adoptive node still have the extra disk even though it has been removed from the volume group on the primary? What error messages would we expect to receive on the adoptive node if the package failed over? Try it. How do we eliminate these error messages? Make sure that the “test” package is back on its primary node before correcting the adoptive node’s /etc/lvmtab file!!

17-8. LAB: Add and Remove a Logical Volume/File System to the Volume Group Owned by the "test" Package

On the Primary Node:

1. Add a 32M logical volume named `sglvol3` to the **test** package's volume group.
2. Build a file system in this new logical volume.
3. Create a mount point, `/sgdata3`, on both systems.
4. Remember, if the package control script needs to be modified, the package should first be halted. Halt the package.
5. Modify the control script on both systems to include this new logical volume/file system.
6. Restart the package. Verify the new file system is mounted.

7. What would happen if the package failed over right now? Does the adoptive system have device files for this new logical volume? Would the package have problems starting?

On the Adoptive Node:

8. Take the steps necessary on the adoptive node so that the package will failover properly. Remember, you will need a new map file that includes the newly-created logical volume.
9. Failover the package and verify the new file system is mounted and exported on the adoptive node.
10. Halt the test package.

On the Primary Node:

11. Remove the newly created logical volume on the primary and modify the control script to remove the references to this file system.
12. Restart the package on the primary and verify the package starts properly. The adoptive node now has extra device files. What do we need to do to ensure the adoptive node understands what this volume group looks like?

17-9. LAB: Add a New Volume Group to the "test" Package

1. On the test package's primary node, add a new volume group, named `vg05`, using 1 or (preferably) 2 unused disks.
2. In your new volume group, create a logical volume named `sglv015`, create a `vxfs` file system in the logical volume, and build a mount point at `/sgdata5`.
3. If you used two disks in step 1 above, mirror the `sglv015` logical volume.
4. Halt the test package, modify the control script `test.cnt1` to activate this new volume group when the package starts, and mount the newly-created file system.
5. What else needs to be done to the newly-created volume group so it can be successfully activated when the package starts up? (**Hint:** The script issues a `vgchange -a e`)
(If this is not done, the new volume group will not be activated)

LVM Maintenance for Packages

6. Start the package on the primary node. Verify that the new volume group is activated and the new file system is mounted.
7. What would happen if the package attempted to start on the adoptive node at this point?
8. Take the steps necessary to define the new volume group on the adoptive node. Don't forget to create the /sgdata5 mount point directory!
9. Verify that the package can failover onto the adoptive node properly.
10. Add the new vg05 volume group to the cluster ascii file and apply the change.

17-10. LAB Solution: Increasing the Size of a Logical Volume / File System

Make the /sgdata1 file system larger. Recall that /sgdata1 is used by the test package.

Answer:

```
# lvextend -L xxx /dev/vg01/sglvol1      (where xxx is in MB)
# fsadm -F vxfs -b xxxM /sgdata1        (where xxx is in MB)
```

17-11. LAB Solution: Add and Remove an Unused Disk to Volume Group "vg01", Owned by the "test" Package

NOTE: This lab exercise assumes that the "test" package is running on its primary node and all switches are enabled. If necessary, make the appropriate changes now to ensure the above.

1. Make a note of an unused disk.

Answer:

```
# iotop -f disk
# strings /etc/lvmtab
```

2. Make a note of the volume group owned by the **test** package.

Answer:

Review the `test.cnt l run/halt` script for the volume group.
It should be `vg01`.

3. Make a note of the disks currently included in this volume group.

Answer:

```
# vgs -v vg01
```

4. Add the unused disk to the volume group.

Answer:

```
# pvcreate [ -f ] /dev/rdisk/cXtYdZ
# vgextend vg01 /dev/dsk/cXtYdZ
```

5. Create an additional mirror copy of `/dev/vg01/sglvol1` on this newly added disk.

Answer:

```
# lvextend -m 2 /dev/vg01/sglvol1
```


6. If the package failed over right now, would the new disk and its new mirror copy be accessible on the adoptive system? Has it been added to `/etc/lvmtab` on the adoptive system? Let's find out.

Failover the package onto the adoptive system. Issue an `lvdisplay` on the logical volume with the extra mirror copy. Is the extra mirror copy current and accessible? Why not?

Answer:

The new disk has not been added to the `/etc/lvmtab` file on the adoptive system. The package should fail over, but the extra mirror copy will not be accessible. The `lvdisplay` command reflects this, as the extra mirror's extents should all show "stale". The package will failover, but the extra mirror copy will not be accessible.

7. Move package back to the primary and re-enable all package switching.

Answer:

```
# cmhaltpkg -v test
# cmmodpkg -v -e -n <primary> test
# cmrunpkg -v -n <primary> test
# cmmodpkg -v -e test
```

8. Re-vgimport the vg01 volume group on adoptive node, recreating map file if necessary. (Remember to `vgexport` on the adoptive node first).

Answer:

On the primary node:

```
# vgexport -p -s -m /tmp/vg01.map vg01 ← the map file will get
                                         created even if vg01
                                         is activated.
```

On the adoptive node:

```
# vgexport vg01
# rcp <primary node>:/tmp/vg01.map /tmp/vg01.map
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgimport -v -s -m /tmp/vg01.map /dev/vg01
```

9. Is new disk reflected in `/etc/lvmtab` on the adoptive node?

Answer:

Yes.

10. Failover the package again. Issue `lvdisplay` on the new logical volume. Is the extra mirror copy current and accessible?

Answer:

Yes.

11. Take all steps necessary to move the “test” package back to its primary node and then `lvreduce` the extra mirror copy from extra disk, and remove the extra disk from the volume group on the primary.

Answer:

On the node that the test package is currently running:

```
# cmhaltpkg test
# cmmodpkg -e -n <primary node> test
# cmmodpkg -e test
# lvreduce -m 1 /dev/vg01/sglvol1 /dev/dsk/cXtYdZ
# vgreduce /dev/vg01 /dev/dsk/cXtYdZ
```

12. Does `/etc/lvmtab` on the adoptive node still have the extra disk even though it has been removed from the volume group on the primary? What error messages would we expect to receive on the adoptive node if the package failed over? Try it. How do we eliminate these error messages? Make sure that “test” is back running on its primary node before correcting the adoptive node’s `/etc/lvmtab` file!

Answer:

Does `/etc/lvmtab` on the adoptive node still have the extra disk even though it has been removed from the volume group on the primary? **Yes**

What error messages would we expect to receive on the adoptive node if the package failed over? **Can’t find disks in the VG** Try it. How do we eliminate these error messages? **You will need to clean up the `/etc/lvmtab` file on the 2nd node, as well, as follows:**

First, bring the “test” package back to its primary node.

```
# cmhaltpkg test
# cmmodpkg -e -n <primary node> test
# cmmodpkg -e test
```

On the node on which the package is now running, execute:

```
# vgexport -p -s -m /tmp/vg01.map vg01
# rcp /tmp/vg01.map <other_node>:/tmp/vg01.map
```

On the other node:

```
# vgexport vg01
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgimport -s -m /tmp/vg01.map vg01
```

17-12. LAB Solution: Add and Remove a Logical Volume/File System to the Volume Group Owned by the "test" Package

On the Primary Node:

1. Add a 32M logical volume named `sglvol3` to the **test** package's volume group.

Answer:

```
# lvcreate -L 32 -n sglvol3 vg01
```

2. Build a file system in this new logical volume.

Answer:

```
# newfs -F vxfs /dev/vg01/rsglvol3
```

3. Create a mount point, `/sgdata3`, on both systems.

Answer:

```
# mkdir /sgdata3
```

 ← On both nodes

4. Remember, if the package control script needs to be modified, the package should first be halted. Halt the package.

Answer:

```
# cmhaltpkg -v test
```

5. Modify the control script on both systems to include this new logical volume/file system.

Answer:

Edit the `test.cntl` run/halt script and add:

```
LV[1]="/dev/vg01/sglvol3";FS[1]="/sgdata3"
```

6. Restart the package. Verify the new file system is mounted.

Answer:

```
# cmmodpkg -v -e test  
# tail -f /etc/cmcluster/test/test.cntl.log
```

7. What would happen if the package failed over right now? Does the adoptive system have device files for this new logical volume? Would the package have problems starting?

Answer:

The adoptive system does not have the device files `/dev/vg01/sgdata3` and `/dev/vg01/rsgdata3`. The run script will not be able to `fsck` and mount this file system. The package will have problems starting.

On the adoptive node:

8. Take the steps necessary on the adoptive node so that the package will failover properly. Remember, you will need a new map file that includes the newly-created logical volume.

Answer:

We need a new map file created on the primary node and copied over:

On the primary node:

```
# vgexport -v -p -s -m /tmp/vg01.map /dev/vg01
```

(This will recreate the map file on the primary node.)

On the adoptive node:

```
# rcp <primary_node>:/tmp/vg01.map /tmp/vg01.map
# vgexport -v vg01
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgimport -v -s -m /tmp/vg01.map vg01
# mkdir /sgdata3
```

9. Failover the package and verify the new file system is mounted and exported on the adoptive node.

Answer:

```
# ps -ef | grep test_program
# kill PID
```

10. Halt the test package.

Answer:

```
# cmhaltpkg test
```

On the primary node:

11. Remove the newly created logical volume on the primary and modify the control script to remove the references to this file system.

Answer:

With the package halted, the volume group should be deactivated.

```
# vgchange -a e vg01  
# lvremove -f /dev/vg01/sglvol3
```

Modify run/halt script to remove references to the sglvol3 logical volume and to the sgdata3 file system.

12. Restart the package on the primary and verify the package starts properly. The adoptive node now has extra device files. What do we need to do to ensure the adoptive node understands what this volume group looks like?

Answer:

Repeat the `vgimport vg01` process on the adoptive node.

17-13. LAB Solution: Add a New Volume Group to the "test" Package

1. On the test package's primary node, add a new volume group, named vg05, using 1 or (preferably) 2 unused disks.

Answer:

```
# mkdir /dev/vg05
# mknod /dev/vg05/group c 64 0x050000
# pvcreate [-f ] /dev/rdisk/c_t_d_
# pvcreate [-f ] /dev/rdisk/c_t_d_ ← If you have an extra disk
# vgcreate vg05 /dev/dsk/c_t_d_ /dev/dsk/c_t_d_
```

2. In your new volume group, create a logical volume named sglvol5, create a vxfs file system in the logical volume, and build a mount point at /sgdata5.

Answer:

```
# lvcreate -L 20 -n sglvol5 vg05
# newfs -F vxfs /dev/vg05/rsglvol5
# mkdir /sgdata5
```

3. If you used two disks in step 1 above, mirror the sglvol5 logical volume.

Answer:

```
# lvextend -m 1 /dev/vg05/sglvol5
```

4. Halt the test package, modify the control script test.cnt1 to activate this new volume group when the package starts, and mount the newly-created file system.

Answer:

```
# cmhaltpkg -v test
```

And then edit the test.cnt1 run/halt script and add:

```
VG[1]="vg05"
LV[1]="/dev/vg05/sglvol5";FS[1]="/sgdata5"
```

5. What else needs to be done to the newly-created volume group so it can be successfully activated when the package starts up? (**Hint:** The script issues a vgchange -a e) (If this is not done, the new volume group will not be activated.)

Answer:

```
# vgchange -a n vg05
# vgchange -c y vg05
```

6. Start the package on the primary node. Verify that the new volume group is activated and the new file system is mounted.

Answer:

```
# cmmodpkg -v -e test
```

7. What would happen if the package attempted to start on the adoptive node at this point?

Answer:

The new volume group would fail to activate and the new file system would not mount.

8. Take the steps necessary to define the new volume group on the adoptive node. Don't forget to create the /sgdata5 mount point directory!

Answer:

```
# mkdir /sgdata5  
Just vgimport -v vg05 on the adoptive node.
```

9. Verify that the package can failover onto the adoptive node properly.

Answer:

```
# cmviewcl -v -p test
```

10. Add the new vg05 volume group to the cluster ascii file and apply the change.

Answer:

```
# vi /etc/cmcluster/cmclconfig.ascii and add vg05 at the end.  
# cmapplyconf -C /etc/cmcluster/cmclconfig.ascii
```

Module 18 — Serviceguard Manager

Objectives

Upon completion of this module, you will be able to do the following:

- Discuss the components of the Serviceguard Manager.
- Navigate the Serviceguard Manager GUI.

18-1. SLIDE: HP Cluster Monitoring Tools



HP Cluster Monitoring Tools

Tools available for monitoring / administering a Serviceguard Cluster:

- Serviceguard Manager
- EMS
- OpenView Suite
- Command line

Student Notes

- Serviceguard Manager, an intuitive and easy to use, JAVA TM-based graphical user interface.
- EMS, Event Monitoring Service provides local monitoring services for applications. We will discuss EMS in much greater detail in a later module.
- Openview is a family of network and system management products like OpenView Operations (OVO). For more information on the Openview family of products, see the “HP Openview Network Node Manager on Unix” course (HP course number B4743S). More information on this class and other Openview classes can be seen at: <http://www.hp.com/education/sections/network.html>.
- Command line interface. This is the traditional tool for managing a Serviceguard cluster, and is the interface we have been using in this training class all week.

18-2. SLIDE: Serviceguard Manager



Student Notes

What Does Serviceguard Manager Provide?

Serviceguard Manager provides a visual tool to display and administer HP Serviceguard, Serviceguard SGeRAC, Metrocluster, and Continental clusters, maintaining high availability. Using Serviceguard Manager, operators see color-coded, graphically-intuitive icons to get the big-picture view of multiple clusters so that they can proactively manage the clusters, systems (nodes), and applications.

Serviceguard Manager integration with Openview (VPO 6.X, OVO 7.X and 8.0, NNM 6.X, 7.0 and 7.1) is supported on HP-UX. On XP Professional Edition, Windows 2000 and Windows 2003 Server Edition, Serviceguard Manager can be integrated with NNM 7.0, 6.31, 6.3 and 6.2.

Serviceguard Manager can be launched by HP Systems Insight Manager (versions 4.2 and 5.0) if Serviceguard Manager is installed on an HP Systems Insight Manager central management server.

In order to manage clusters, Serviceguard Manager must communicate with a server where Serviceguard has been installed. The server is **required** to act as a proxy to gather information from multiple clusters for Serviceguard Manager.

In order to administer clusters, a user must connect to an HP-UX server with Serviceguard version A.11.14 (or later) or a Linux server with Serviceguard version 11.14.02 (or later). If the server has Serviceguard A.11.15 (or earlier) installed, a user must connect as **root** to perform administration tasks. If the server has Serviceguard A.11.16 (or later) installed, a user can take advantage of the newly introduced role-based access capabilities, which allows non-root users to perform administration tasks. For detailed information on how to take advantage of this feature, please refer to the “**Administering Serviceguard Clusters**” section in Serviceguard Manager Online Help after Serviceguard Manager is installed.

To be discovered by Serviceguard Manager, a Serviceguard cluster needs to be on the same subnet as the server node (but not necessarily on the same subnet as the Serviceguard Manager system). Each cluster node must give access permission to the server node. Prior to Serviceguard A.11.16, such permission would be granted via the file `/etc/cmcluster/cmclnodelist`. With the introduction of role-based access in A.11.16, this can be achieved by configuring access control policies. For more information, see “**Before Installing Serviceguard Manager**” section in Serviceguard Manager Version A.05.00 Release Notes located at <http://www.docs.hp.com/hpux/ha>.

What's in Serviceguard Manager, Version A.05.00?

- Serviceguard Manager version A.05.00 supports Serviceguard A.11.16.
- In Serviceguard Manager A.04.00, you can create or modify configuration of clusters and packages on nodes with Serviceguard version A.11.16 and newer.

Many of your configuration options are discovered by Serviceguard and displayed in lists for you to select. The configuration file and control script can be automatically created, then distributed to the nodes by clicking a button.

- Serviceguard Manager uses Access Control Policies to allow non-root access to administer clusters and packages.
- The Properties have several new tabs, including the ability to display information about physical volumes, volume groups, EMS resources, and access control policies (roles).
- The new Operations Log window shows messages for administration and configuration actions. Messages from all operations are integrated into the one screen.
- If you connect to a session server with Serviceguard A.11.16 or later, it can discover these Serviceguard Extension for RAC clusters:
 - SGeRAC 11.14 clusters with Oracle 8i or Oracle 9i
 - SGeRAC 11.15 clusters with Oracle 8i, Oracle 9i, or Oracle 10g
 - SGeRAC 11.16 with Oracle 9i or Oracle 10g

Supported Environments

Nodes with the following Serviceguard versions can be servers and do the discovering:

- Serviceguard Version A.11.12 or later
- Serviceguard OPS Edition, Version A.11.12 or later
- Serviceguard SGeRAC, any version.

Managed Nodes

Cluster objects with the following Serviceguard versions can be discovered.

- Serviceguard version A.10.10 or later
- Serviceguard OPS Edition A.11.01 or later
- Serviceguard SGeRAC (all versions)
- Metrocluster (all versions)
- Continentalclusters (all versions)
- At least one SG Cluster Object Manager (installed with Serviceguard version A.11.12, Serviceguard OPS version A.11.12, or Serviceguard SGeRAC)

Serviceguard Management Consoles

You can install Serviceguard Manager on either a Windows PC or an HP-UX system.

The sections below outline the recommended hardware and software for running Serviceguard Manager on a Windows PC or an HP-UX system. These recommendations should be treated as *minimum* requirements for reliable operations at acceptable performance levels. It is possible to run with fewer resources at the risk of reduced performance and possibly incorrect operations.

HP-UX management station requirements:

- HP-UX 11.x operating system
- 256 MB of available memory
- 190 MB of available hard disk space under /opt
- 1 MB of available hard disk space under /usr
- 1 MB of available hard disk space under /etc/opt/OV if Openview has been installed
- 3-15 MB of available hard disk space under /var for log files

PC management station requirements:

- Windows XP Professional or Windows 2000 Professional (with SP1 or later) or Windows 2003 Server Edition
- Pentium II 200 MHz or higher
- 256 MB of memory
- 65 MB of available hard disk space
- 1 MB of additional hard disk space should be available for log files)
- SVGA or higher-resolution video adapter Installing Serviceguard Manager

If a previous version of Serviceguard Manager is already installed, you must uninstall it as directed in the next section before installing the newer version.

Installation Instructions

If a previous version of Serviceguard Manager is already installed, you must uninstall it before installing the new version. You can install Serviceguard Manager on a Windows, Linux, or HP-UX system. For additional information regarding Serviceguard Manager release A.05.00, go to <http://www.docs.hp.com/hpux/ha> and look for Serviceguard Manager Version A.05.00 Release Notes.

Installing Serviceguard Manager on a Windows PC

On a Windows PC, you can install from the download file as outlined below:

1. If a version of Serviceguard Manager earlier than A.03.00.01 has been installed on your system, you must uninstall it prior to installing this version (A.05.00). You can do this through the Windows Start menu by selecting the Start > Programs > Serviceguard Manager > Uninstall option. After you have uninstalled Serviceguard Manager, you may continue with the next step to install the latest version.
2. Click on the link under download software (sgmanager.exe) to access the self-extracting zip file used to install Serviceguard Manager.
3. You may choose to store this file on your system or run it directly from the website. If you download to your system, this file can be removed after installation.
4. Run sgmanager.exe.
5. The installation process will ask you to select a language for installation.
6. The installation process will ask for an installation directory. If you accept the default, Serviceguard Manager will be installed in:

C:\Program Files\Hewlett-Packard\Serviceguard Manager

You are now ready to run Serviceguard Manager.

Installing Serviceguard Manager on a HP-UX System

On a HP-UX system, you can install from the download file as outlined below:

1. Select your operating system from the “Software Specification” pull-down list. (To see the version of an HP-UX node, use the `uname` command from the node.)
2. Download and store this file on your system in a directory of your choice (for example, `/tmp`). Do not attempt to run this file from the website. This file can be removed after installation is completed.
3. If a version of Serviceguard Manager earlier than A.03.00.01 has been installed, run the command `swremove B8325BA` to remove it. Otherwise, go to the next step.
4. Run the command `swinstall -s /tmp/B8325BA_11.X.depot` to start the installation process.
5. Follow the directions presented by `swinstall` to mark, analyze, and install Serviceguard Manager. Serviceguard Manager will be installed under the `/opt/sgmgr` directory. Also, the `/var/opt/sgmgr` directory will be created for the storage of log files.

When Serviceguard Manager is installed on a HP-UX system, the installation process examines the system to see if Openview is present. If so, the installation process adds the HP Serviceguard Manager Launcher command to the Openview menu.

Serviceguard Manager installation process then checks to see if the OVO toolkit and certain tools are installed. If these tools are not present, the installation process automatically installs them. The Serviceguard Manager installation process will not replace any tools that have already been installed.

You are now ready to run Serviceguard Manager.

On the managed HP-UX Serviceguard nodes:

You should install SNMP master agent patch PHSS_27858 for HP-UX 11.x to receive SNMP traps properly.

For Serviceguard 11.16 nodes that will be used as proxy servers for Serviceguard Manager, install PHSS_31077 for HP-UX 11.11 or PHSS_31078 for HP-UX 11.23.

Complete Installation of Serviceguard Manager on an HP-UX System

- 1) Select the product or products to install from the bundle. This graphical user interface product is:

B8325BA	Serviceguard Manager
---------	----------------------

You may choose to also install:

B3935BA	Serviceguard
B8324BA	HP Cluster Object Manager

Follow the directions presented by the `swinstall` GUI to mark, analyze, and install Serviceguard Manager.

When Serviceguard Manager is installed on an HP-UX system, the installation process examines the system to see if Openview is present. If so, the installation process adds the HP Serviceguard Manager Launcher command to the Network Node Manager (NNM) and/or the Information Technology Operations (ITO) menu.

The Serviceguard Manager installation process then checks to see if the ITO toolkit is installed, and if certain tools are installed. If these tools are not present, the installation process automatically installs them. The Serviceguard Manager installation process will not replace any tools that have already been installed.

More information can be found at: `/opt/sgmgr/OV/Readme.txt`

Uninstalling Serviceguard Manager

Uninstalling Serviceguard Manager on a Windows PC

On a Windows system, there are two ways to remove Serviceguard Manager from your computer:

- From the Start menu, choose:
Programs -> Serviceguard Manager -> Uninstall
- From the Start menu, choose:
Settings -> Control Panel -> Add/Remove Programs
- Select Serviceguard Manager when the dialog box opens.

Uninstalling Serviceguard Manager on a HP-UX System

On an HP-UX system, run:

```
swremove B8325BA
```

Running Serviceguard Manager

Serviceguard Manager in Operation

When Serviceguard Manager is run to monitor the operation of clusters, nodes and packages in real-time, Serviceguard Manager will open a live connection to a Serviceguard Object Manager server. It is the Object Manager program, running on a Serviceguard node, which in turn creates connections to the other Serviceguard nodes to gather the data presented by Serviceguard Manager.

The ability of Serviceguard Manager to monitor your systems depends on the ability of the Object Manager to connect to your systems and on the user's ability to log into the Object Manager.

Security Prerequisites

Before Serviceguard Manager is run, confirm that the security configuration of your nodes and clusters is correct. If you do not do this, Serviceguard Manager will run, but any incorrectly configured nodes and clusters will not be discovered.

The nodes in the clusters you wish to monitor must authorize the Object Manager host before it can access them for cluster information. This must be done on every node. Please refer to the *Managing Serviceguard* manual for more information. This manual can be found on the Web at:

<http://www.docs.hp.com/hpux/ha>

For more information, please see the `sgmgr(1M)` man page on any HP-UX system on which Serviceguard version A.11.12 or later has been installed.

18-3. SLIDE: Welcome to Serviceguard Manager



Student Notes

Serviceguard Manager Startup (Platform Independent)

After Serviceguard Manager is launched, a splash screen is displayed that presents the progress of Serviceguard Manager during its initialization.

After initialization, you will see the Welcome dialog box. You can connect to a server to see a snapshot of your clusters and their status. You can update this snapshot at any time by clicking Refresh on the toolbar. You can also look at a saved object (.sgm) file.

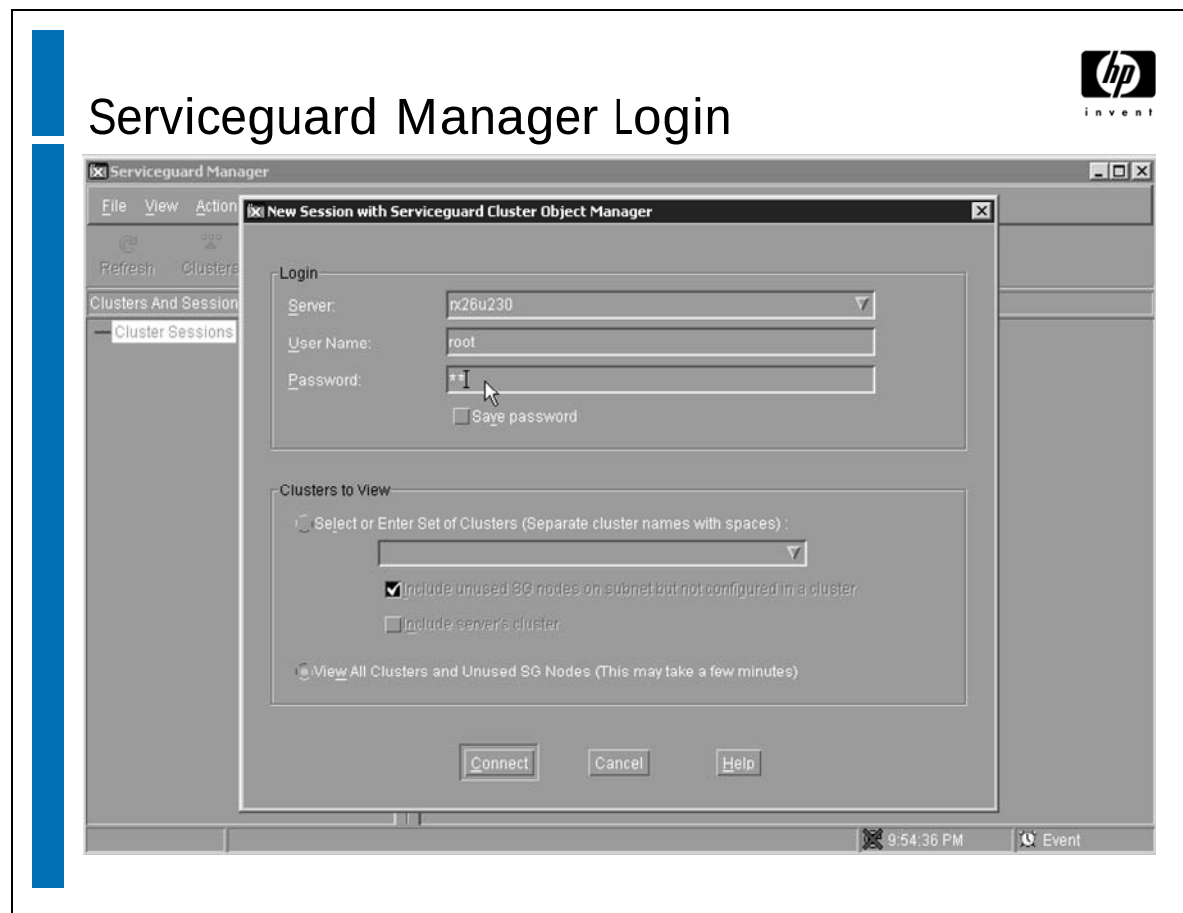
To connect to a server and display status, Serviceguard Manager must complete a discovery process that may take a significant amount of time. Once this one-time delay is incurred at startup, subsequent refresh operations are much faster.

- For a connection, choose the connect option and fill in the dialog box, specifying the host, user name and password. The server must be a node with Serviceguard version A.11.12 or later. It will discover the clusters on its subnet, and provide information to your management station. To access the connection feature later, select New Session from the File menu.

- To see a saved object (.sgm) file, choose “Open a previously saved file...” and select from the list of files presented in the dialog box. You can save maps as .sgm files from the Serviceguard Manager File menu.

To open a saved object file later, select File -> Open.

18-4. SLIDE: Serviceguard Manager Login



Student Notes

Running Serviceguard Manager on a Windows PC

On a Windows system, there are four ways to launch Serviceguard Manager:

- Double-click the **SGMgr** icon on your desktop.
- From the PC Start menu, choose:
Programs -> Serviceguard Manager -> SGMgr
- From the My Computer icon or from Windows Explorer, go to:
C:\Program Files\Hewlett-Packard\Serviceguard Manager\bin
Double-click on SGMgr (or SGMgr.exe, if common extensions are not hidden).
- From a DOS window, use the SGMgrDos command in the directory shown above.
Optional syntax at the command line is shown below.

Running from a DOS window allows the command line options to be easily entered at run-time. The command line options may also be specified from the SGMgr icon by changing the properties of the shortcut used to launch Serviceguard Manager.

Right-click on the icon and select **Properties**. Under the **Shortcut** tab, there is a parameter **Target**. The initial value of this parameter is probably:

```
C:\Program Files\Hewlett-Packard\SGManager\bin\SGMgr.exe
```

Note that the path name of SGMgr.exe is enclosed in quotes. This is good practice on a Windows PC, where the path name may include spaces, and in this case is required because of the embedded blank in the Program Files folder name. The quotes prevent the target of the shortcut from being interpreted incorrectly as a target program with one parameter. You may add any desired run-string parameters to the right of the closing double quote.

For example:

```
"C:\Program Files\Hewlett-Packard\SGManager\bin\SGMgr.exe" -c clus1 -c clus2
```

will limit the scope of a live connection from Serviceguard Manager (when it is launched by double-clicking on the icon) to clusters **clus1** and **clus2**.

Running Serviceguard Manager on a HP-UX System

On a HP-UX system, you may launch Serviceguard Manager by using the following sequence of commands:

```
export DISPLAY=<hostname>:0.0      (if needed)
/opt/sgmgr/bin/sgmgr <options>
```

where <hostname> is the domain name of the system where the graphical user interface is to be displayed.

See section **Command Line Syntax** for an explanation of the command line options.

Command Line Syntax

The Serviceguard Manager program may be executed from the command line in several ways:

- **sgmgr** — The HP-UX executable.
- **SGMgr** (or **SGMgr.exe**) — from the PC Desktop Icon
- **SGMgrDos** (or **SGMgrDos.exe**) — from a DOS window on the PC

For purposes of the following discussion, sgmgr will be used. There are no platform specific limitations or extensions of the command line parameters.

Serviceguard Manager provides the ability to examine live cluster data retrieved from a Serviceguard Object Manager. Command line options may be used to identify the Object Manager system, user, and password as shown below:

```
# sgmgr -s server -l user -p password
```

If these parameters are not supplied in the command line, the user will be asked to supply them in a connection dialog box.

By default, the live connection will display all clusters visible to the Object Manager. If a cluster does not appear, you will need to configure the security files on cluster nodes.

To increase performance and simplify activities, you can limit the scope of this discovery to a few chosen clusters, as shown below:

```
# sgmgr -c cluster1 -c cluster2 ...
```

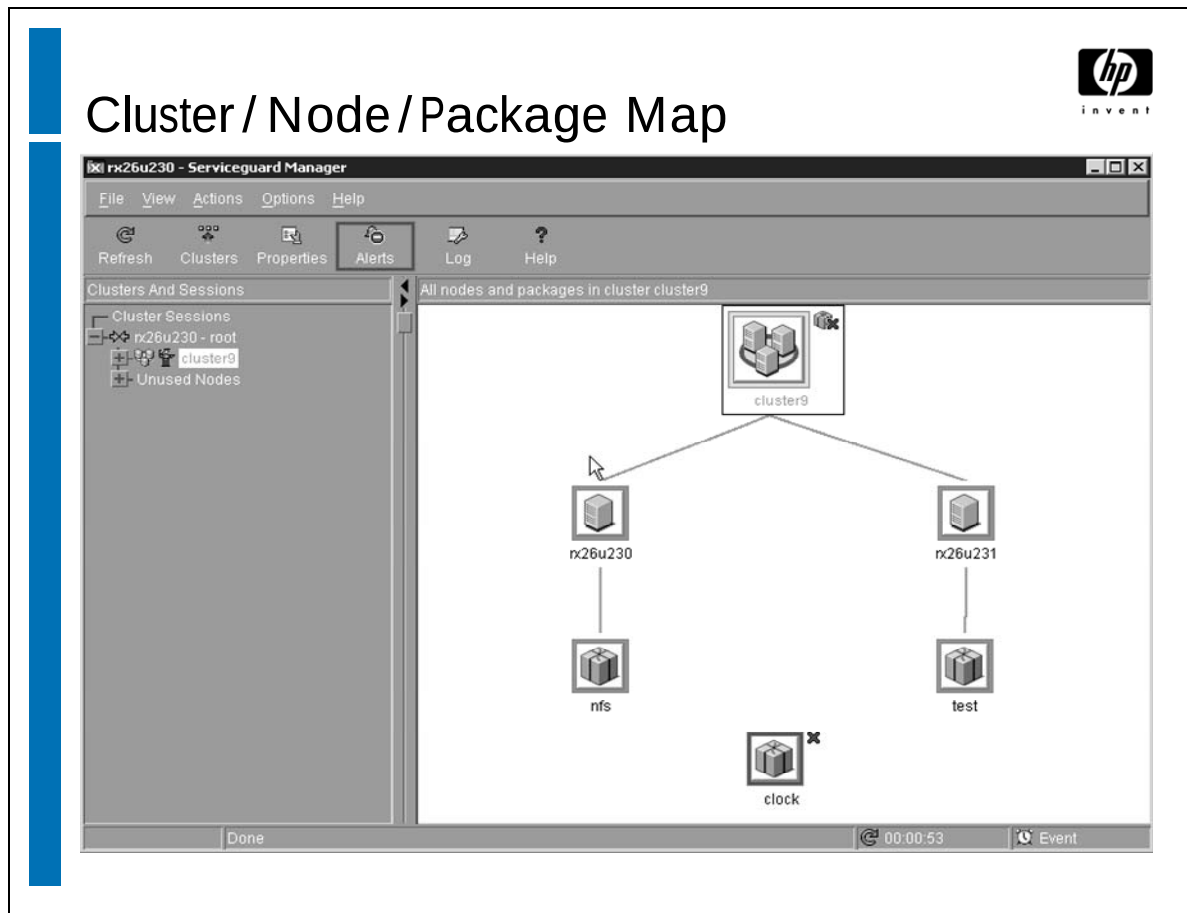
Since these parameters to limit the scope make sense only during a live connection, they can be used on the command line after the parameters for establishing the connection to the Object Manager, as shown below.

```
# sgmgr -s server -l user -p password -c cluster1 -c cluster2 ...
```

Serviceguard Manager can also view cluster data that has been saved to a file. The name of this file can be specified in the command line as shown below:

```
# sgmgr -f filename
```

18-5. SLIDE: Cluster / Node / Package Map

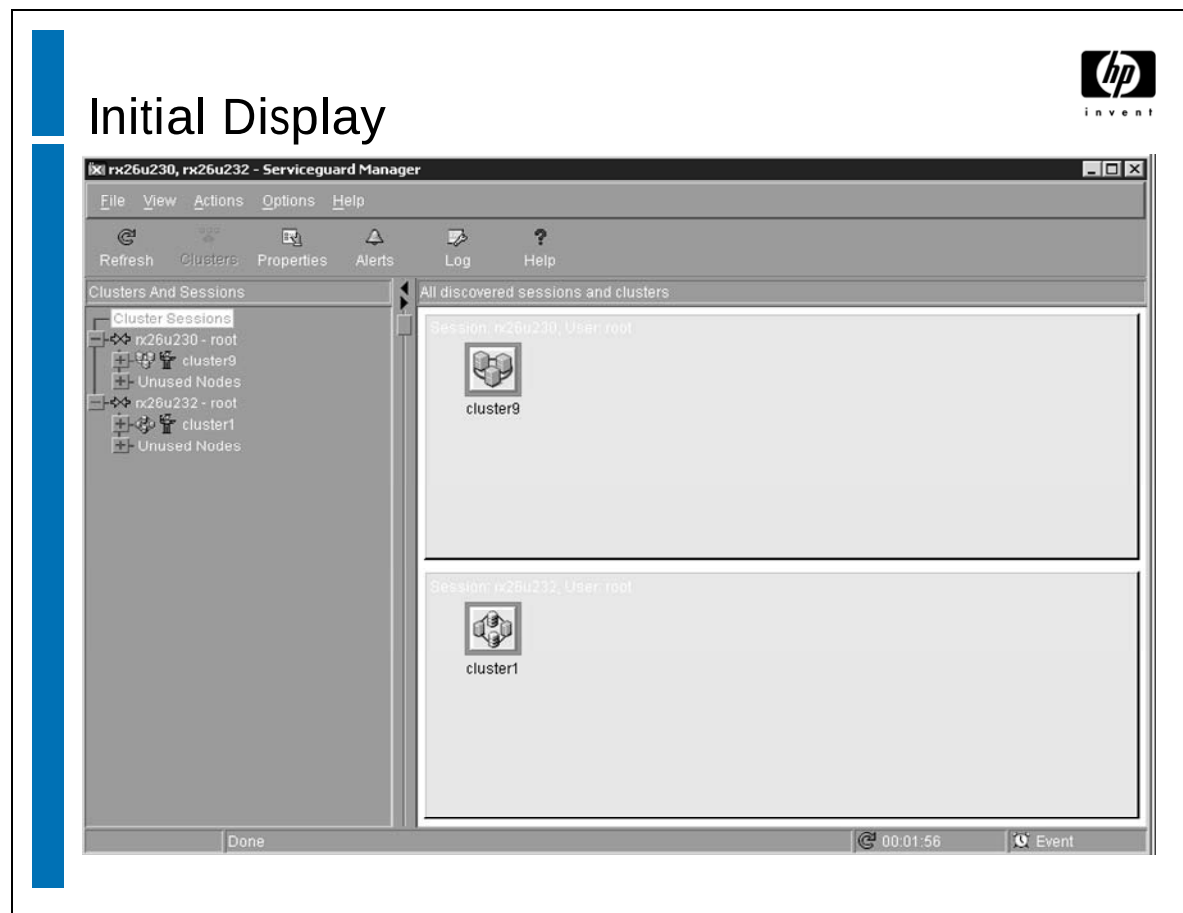


Student Notes

The clusters are shown in a two-paned window. The left pane shows a hierarchical tree view of the clusters and their components. The right pane shows them as a map. You can change the focus of the map by:

- Selecting a different element from the tree, using the mouse (single-clicking on the left button), or by using the arrow keys.
- Double-clicking a symbol on the map.

18–6. SLIDE: Initial Display



Student Notes

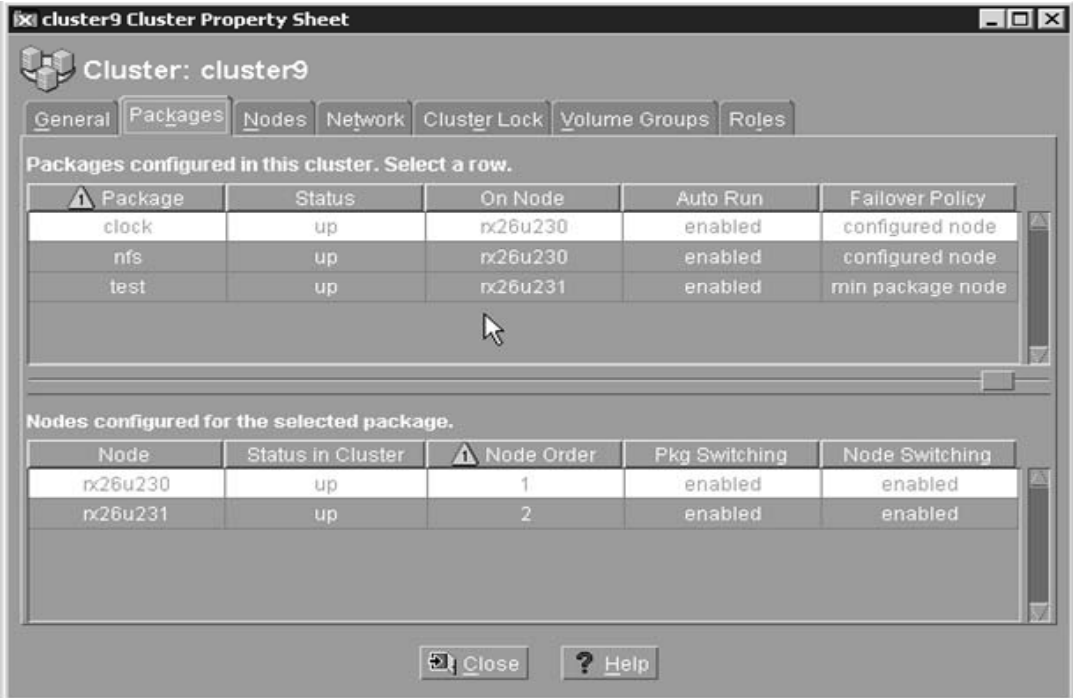
After starting Serviceguard Manager, and upon login and cluster specification, the initial display shows the cluster(s) icon(s) in the right side of the display.

More cluster-specific information can be determined by either left single-clicking a cluster icon and selecting properties from the toolbar –or– by right-clicking the cluster icon and selecting properties in the right-click menu. Note that it is also possible to halt the cluster and/or make cluster modifications via the right-click menu.

18-7. SLIDE: Cluster Properties



Cluster Properties



cluster9 Cluster Property Sheet

Cluster: cluster9

General Packages Nodes Network Cluster Lock Volume Groups Roles

Packages configured in this cluster. Select a row.

Package	Status	On Node	Auto Run	Failover Policy
clock	up	rx26u230	enabled	configured node
nfs	up	rx26u230	enabled	configured node
test	up	rx26u231	enabled	min package node

Nodes configured for the selected package.

Node	Status in Cluster	Node Order	Pkg Switching	Node Switching
rx26u230	up	1	enabled	enabled
rx26u231	up	2	enabled	enabled

Close Help

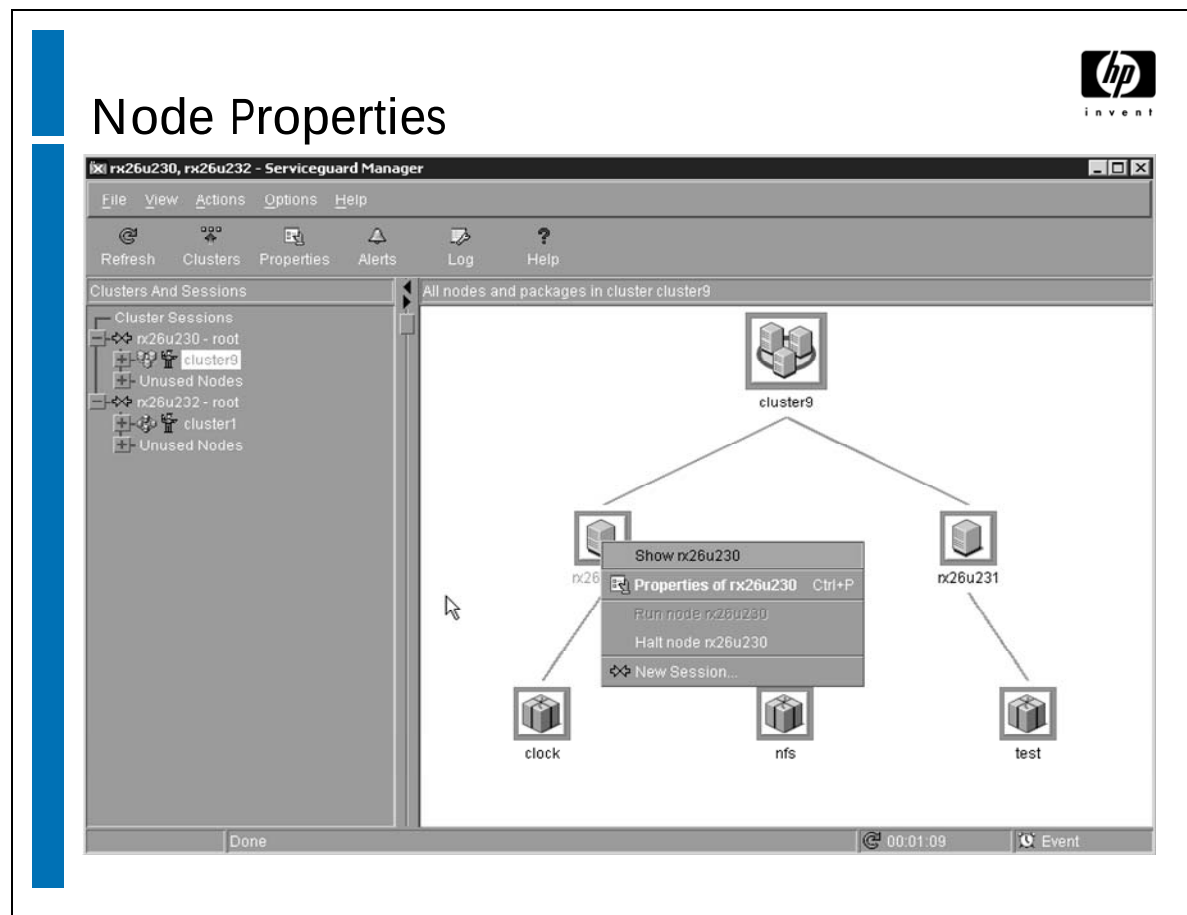
Student Notes

Right click on the cluster icon and select Properties to view the cluster Property Sheet.

The Cluster Property Sheet has seven tabs:

- General
- Packages
- Nodes
- Network
- Cluster Lock
- Volume Groups
- Roles

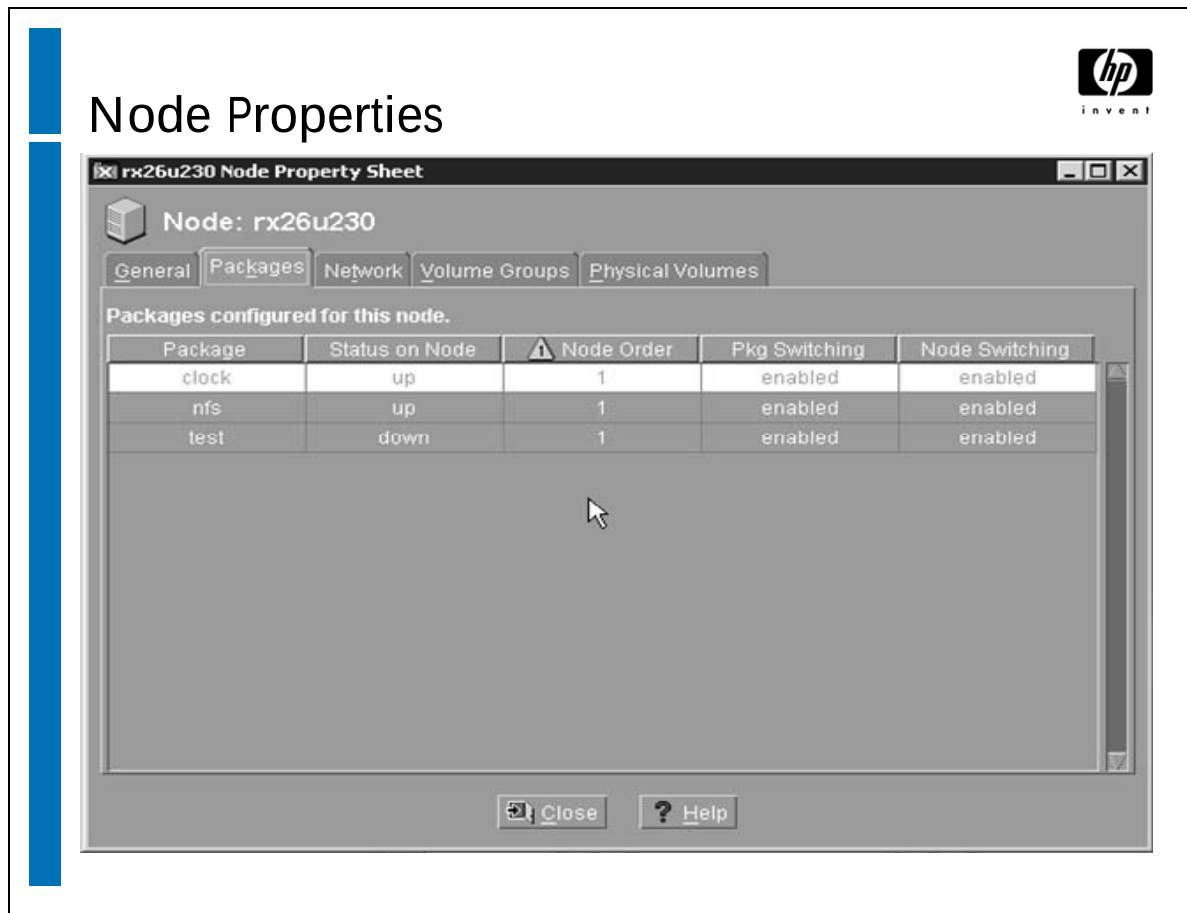
18-8. SLIDE: Node Properties



Student Notes

Double-clicking the cluster icon will expand the view of the cluster. Right-clicking on a node will allow the operator to either view the node properties or halt the node.

18-9. SLIDE: Node Properties



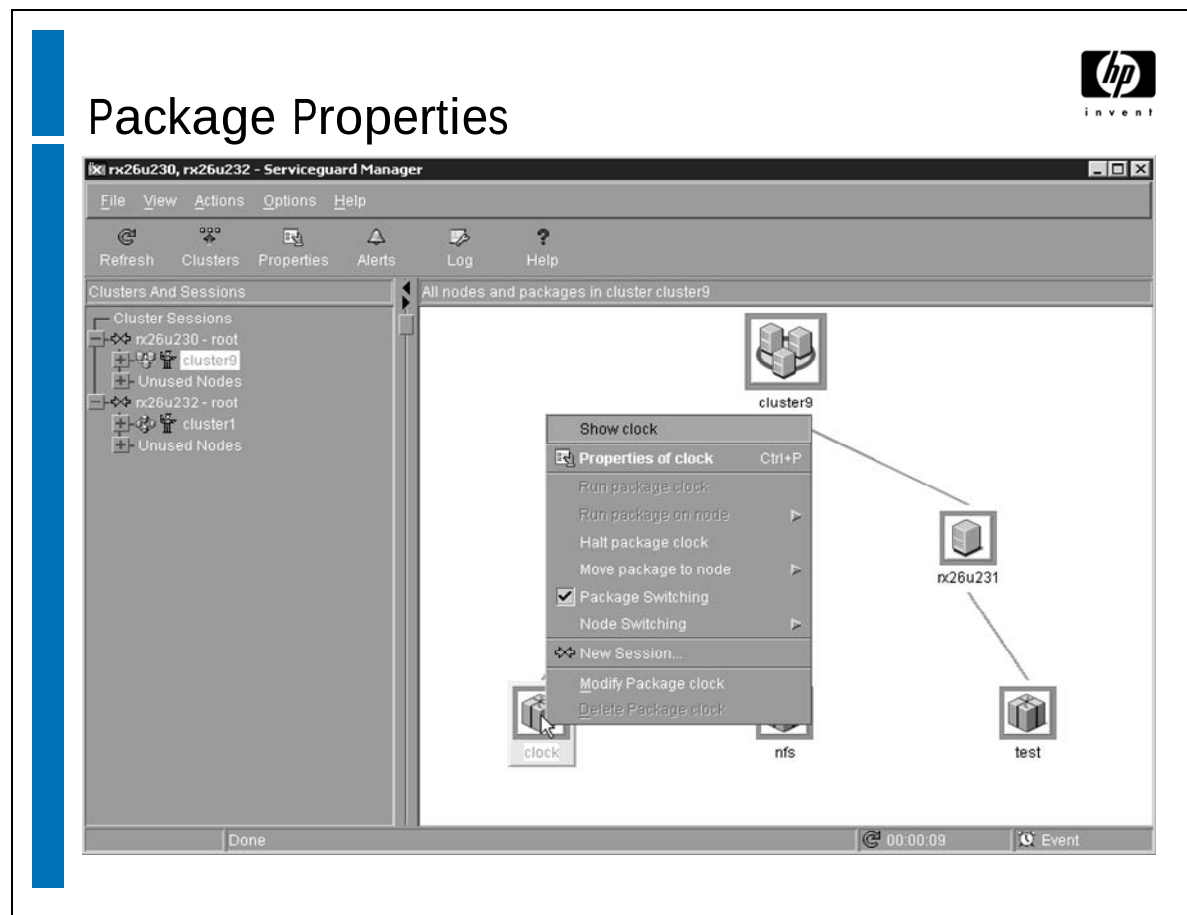
Student Notes

After expanding the cluster (by double-clicking the cluster icon), more information is available. For example, right-clicking on a node allows a view of the node properties.

The Node Property Sheet has six tabs:

- General
- Packages
- Network
- Volume Groups
- Physical Volumes

18-10. SLIDE: Package Properties



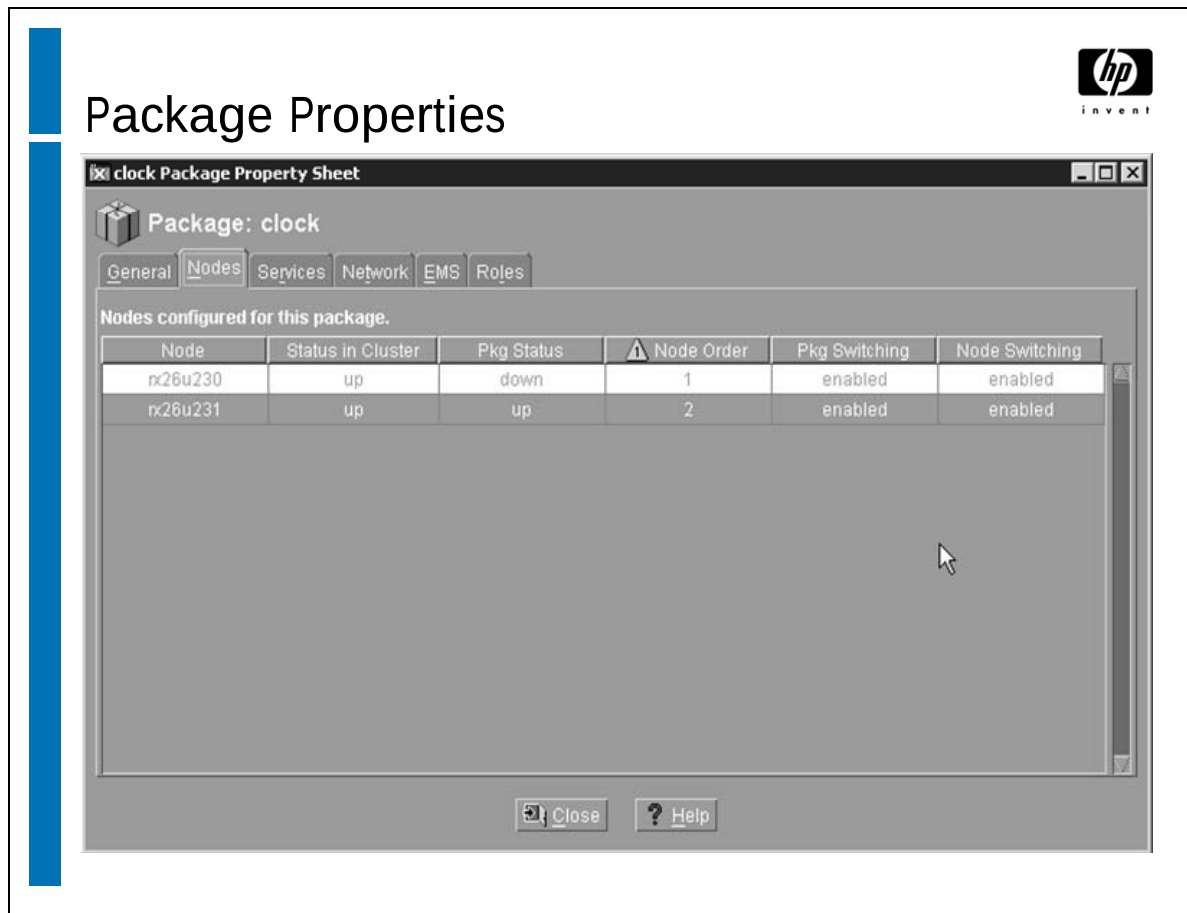
Student Notes

Right click on a package icon and select **Properties** to view the Package Property Sheet.

The Package Property Sheet has six tabs:

- General
- Nodes
- Services
- Network
- EMS
- Roles

18-11. SLIDE: Package Properties



Student Notes

Right-clicking on a package icon allows a view of the package properties and allows several actions. From the right-click menu, it is also possible to do the following:

- Halt the package
- Move the package (which halts the package on the current node and starts it up again on an alternate node)
- Enable/disable package switching
- Enable/disable node switching

NOTE: It is also possible to click and drag a package from one node to another.

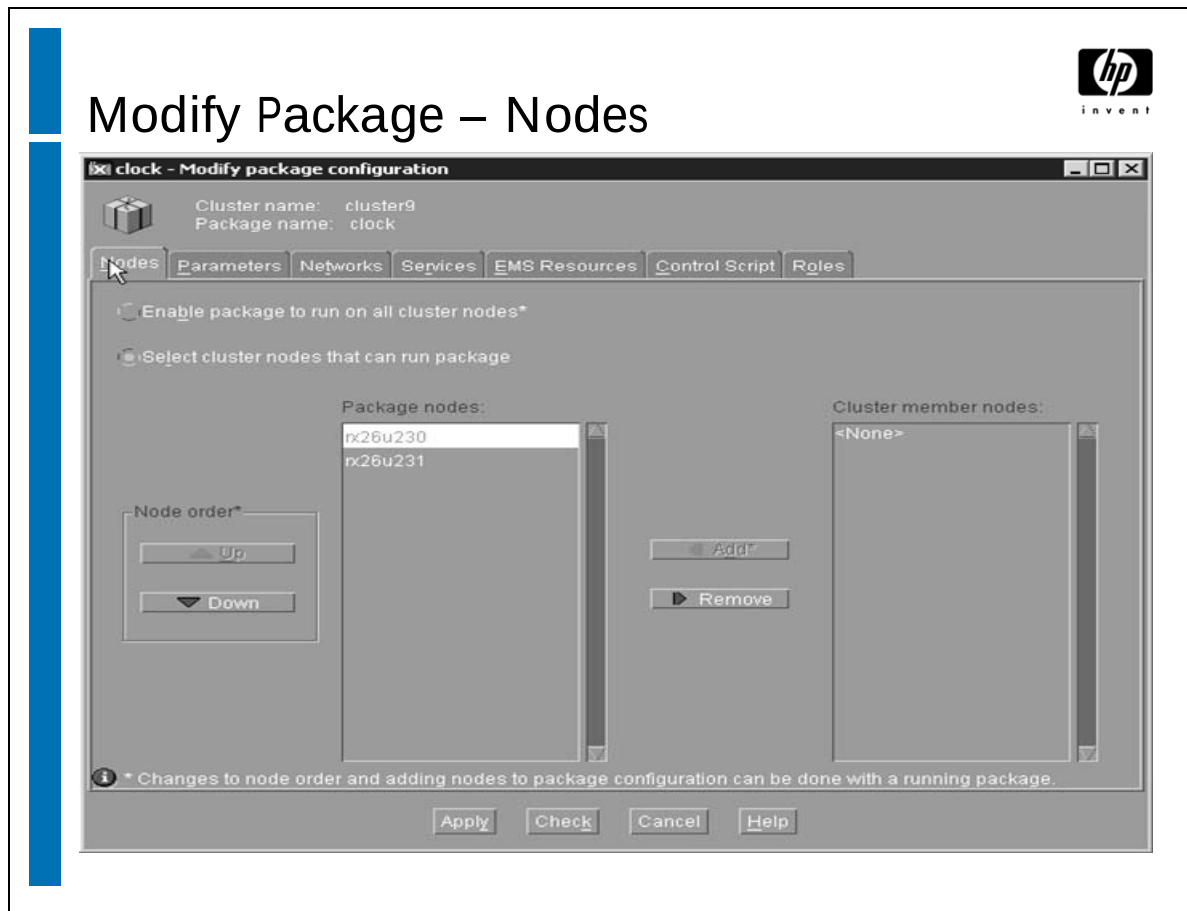
Additionally, the “Modify Package” selection allows the following:

- Changes to the order in which nodes will run the package
- Change initial setting for Auto Run
- Enable/Disable Local LAN Failover
- Changes to FAILOVER and FAILBACK Policies
- Changes to monitored subnets

Serviceguard Manager

- Addition/Deletion of package IP Addresses
- Addition/Modification/Deletion of services for the package
- Addition/Deletion/Deletion of EMS Resources
- Changes to the control script logfile
- Editing of the control script itself
- Access Control Policy changes

18-12. SLIDE: Modify Package – Nodes

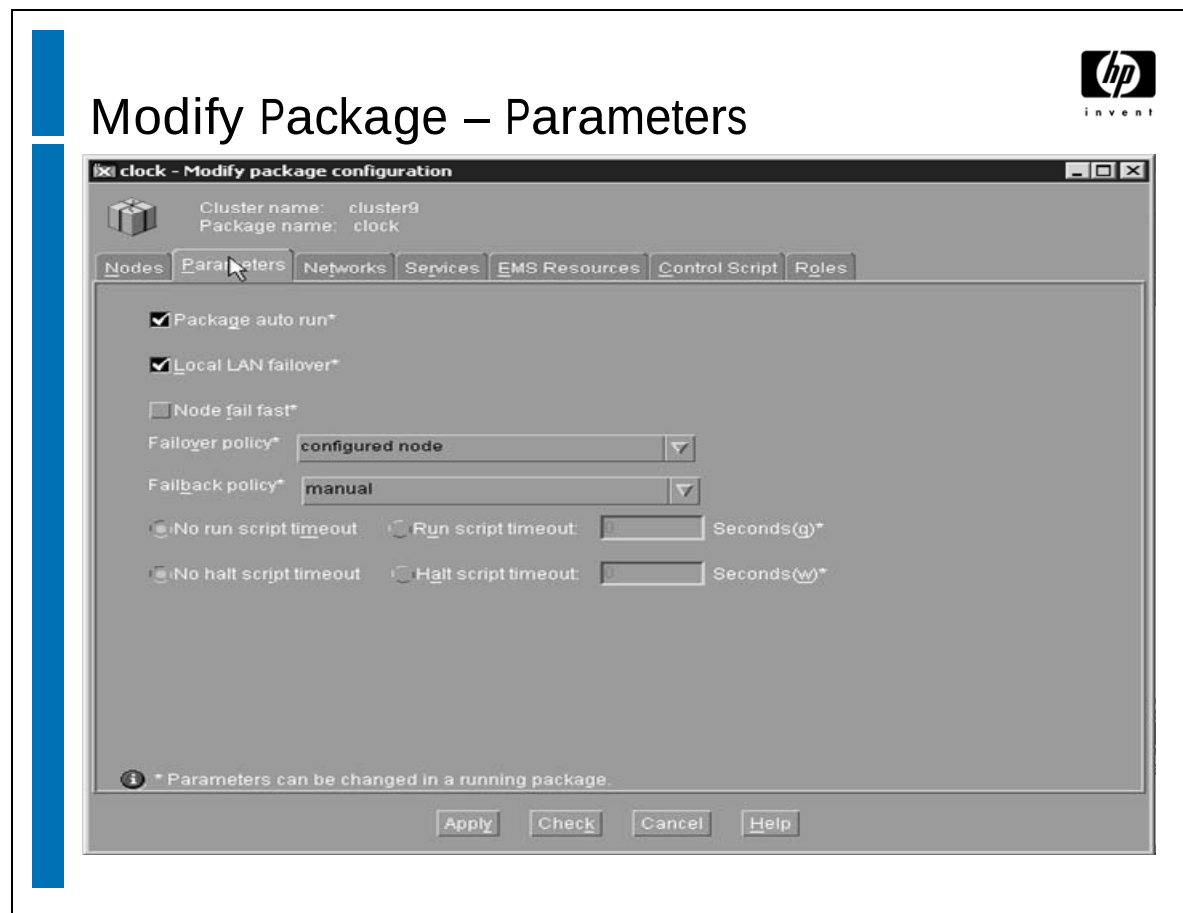


Student Notes

The “Nodes” tab allows the following changes:

- Changes to the order of the nodes which will run the package
- Addition/Deletion of nodes that can run the package

18-13. SLIDE: Modify Package – Parameters

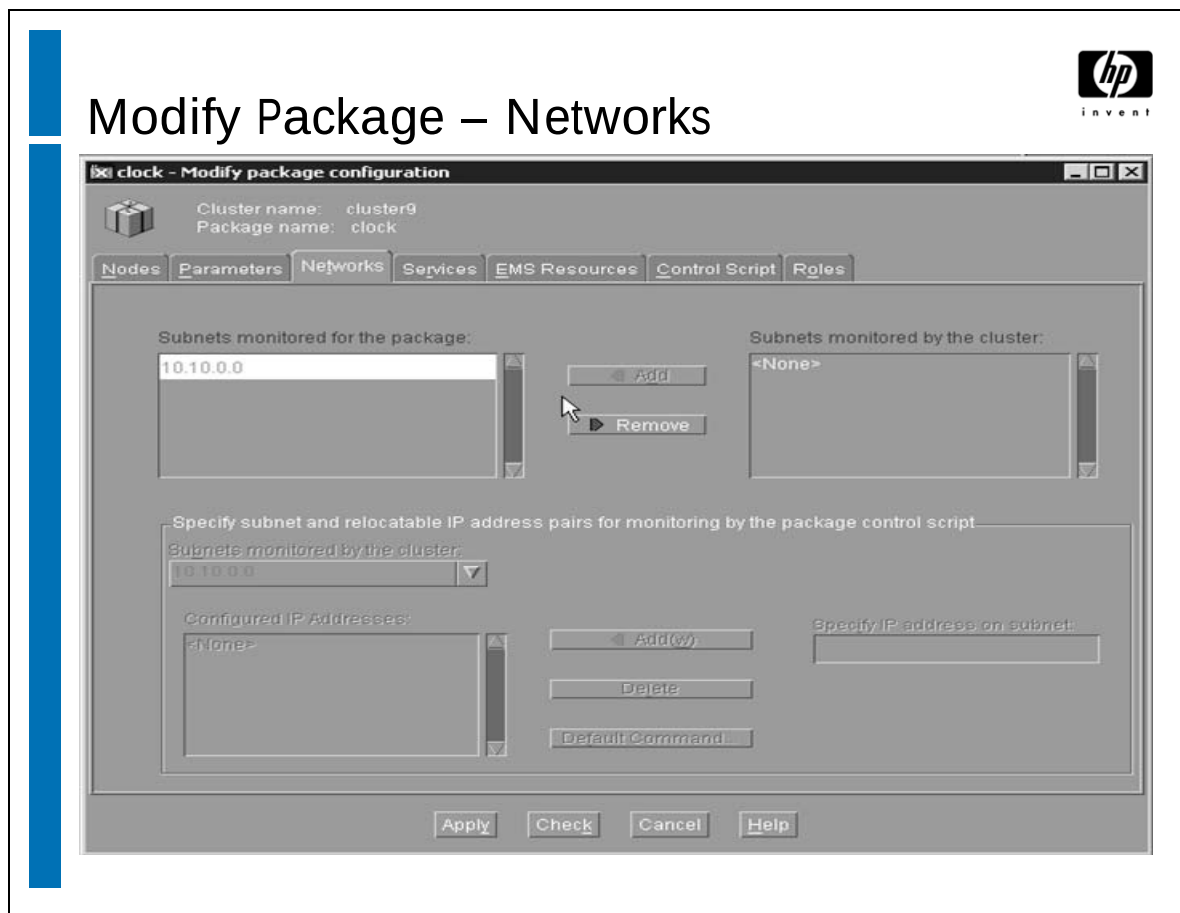


Student Notes

The “Parameters” tab allows the following changes:

- Change initial setting for Auto Run
- Enable/Disable Local LAN Failover
- Changes to FAILOVER and FAILBACK Policies

18-14. SLIDE: Modify Package – Networks

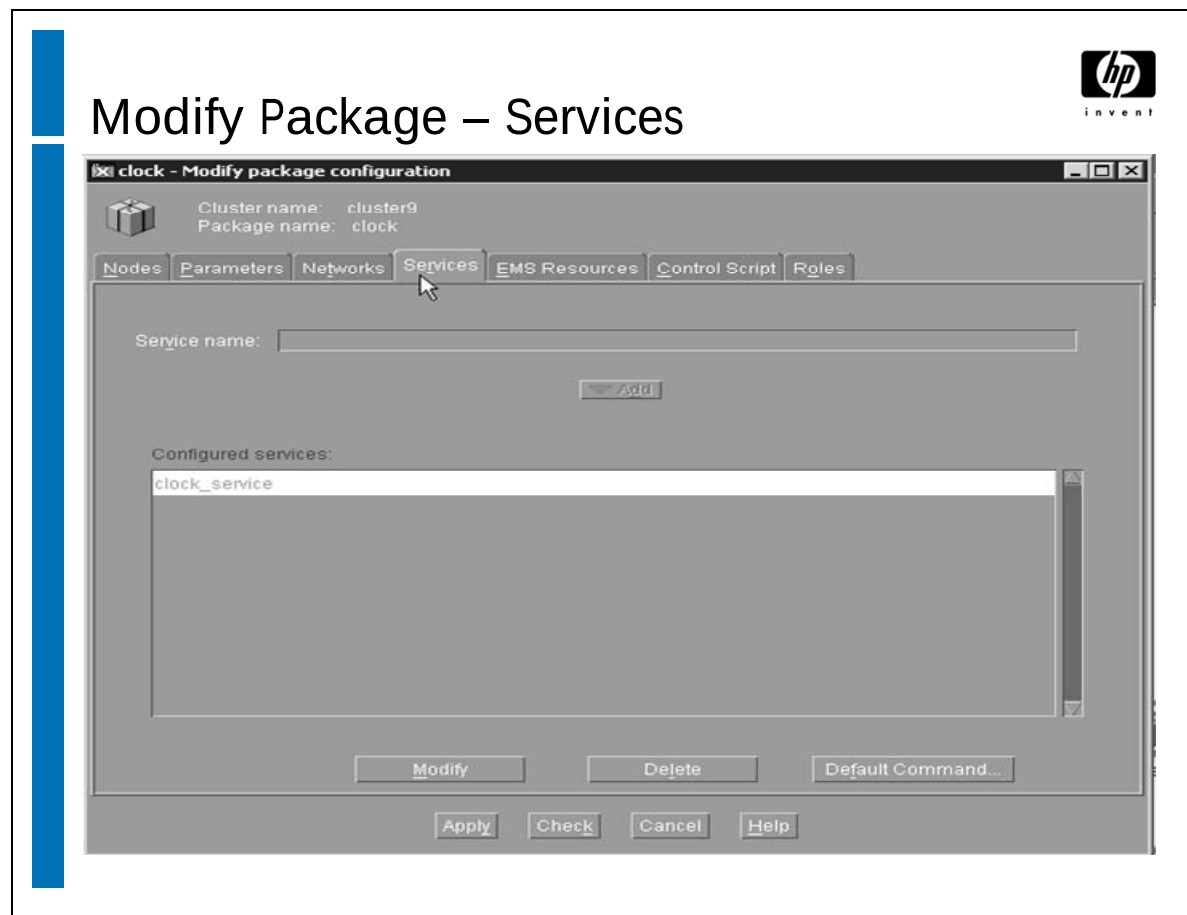


Student Notes

The Networks tab allows the following changes:

- Changes to monitored subnets
- Addition/deletion of Package IP addresses

18-15. SLIDE: Modify Package – Services

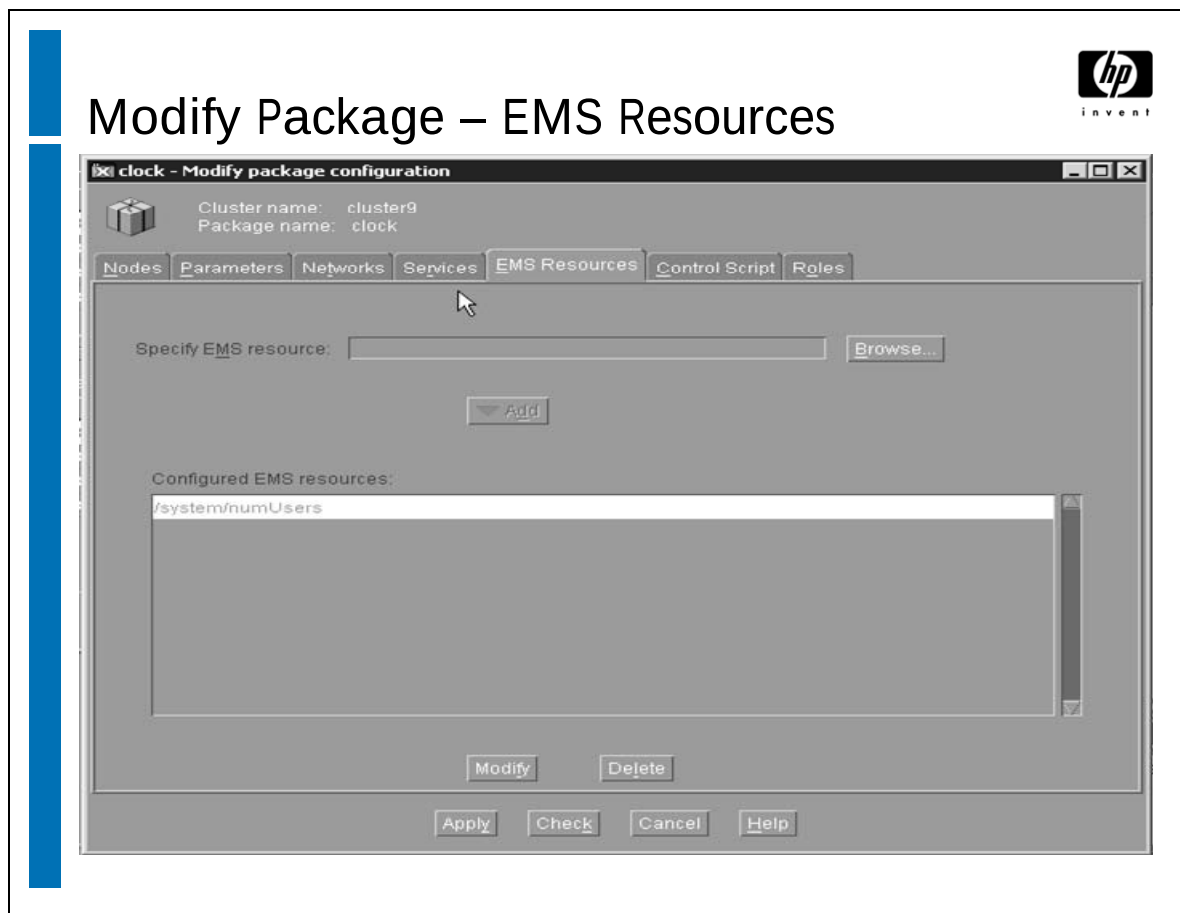


Student Notes

The “Services” tab allows the following changes:

- Addition/Modification/Deletion of services for the package

18-16. SLIDE: Modify Package – EMS Resources

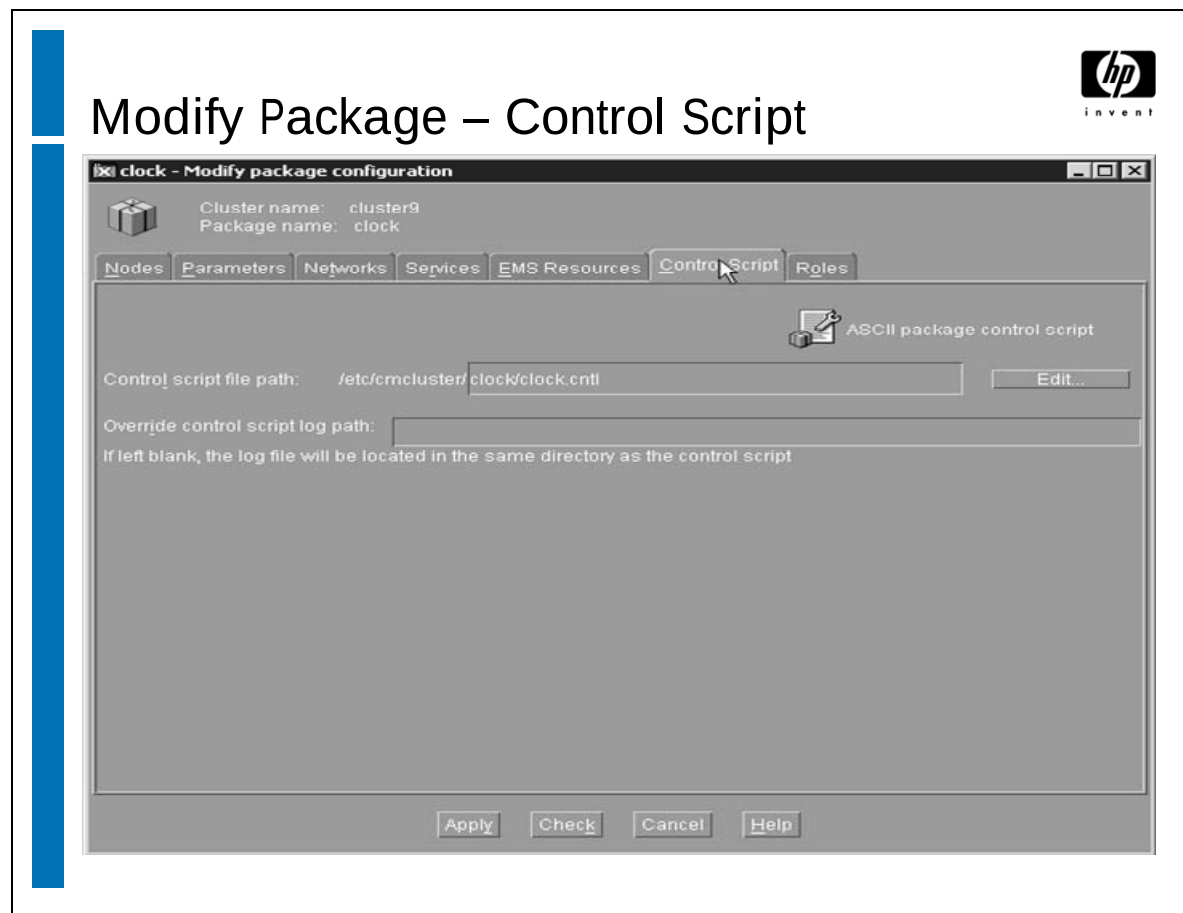


Student Notes

The “EMS Resources” tab allows the following changes:

- Addition/Deletion/Deletion of EMS Resources

18-17. SLIDE: Modify Package – Control Script



Student Notes

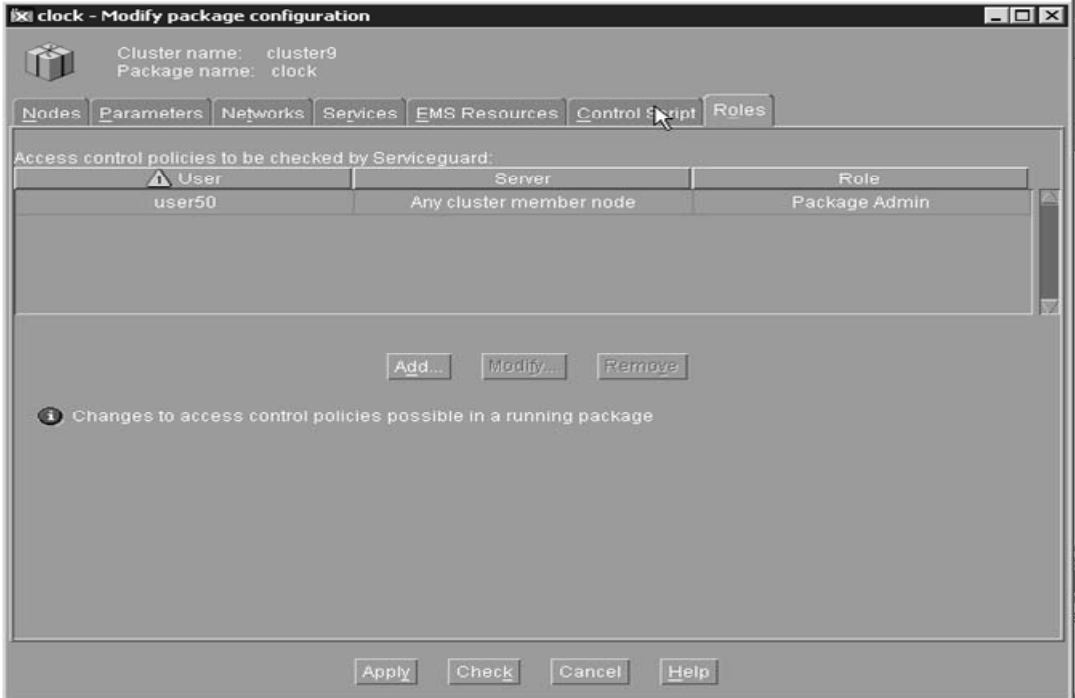
The “Control Script” tab allows the following changes:

- Changes to the control script logfile path
- Editing of the control script itself

18-18. SLIDE: Modify Package – Roles



Modify Package – Roles



Cluster name: cluster9
Package name: clock

Nodes Parameters Networks Services EMS Resources Control Script Roles

Access control policies to be checked by Serviceguard:

User	Server	Role
user50	Any cluster member node	Package Admin

Add... Modify... Remove

Changes to access control policies possible in a running package

Apply Check Cancel Help

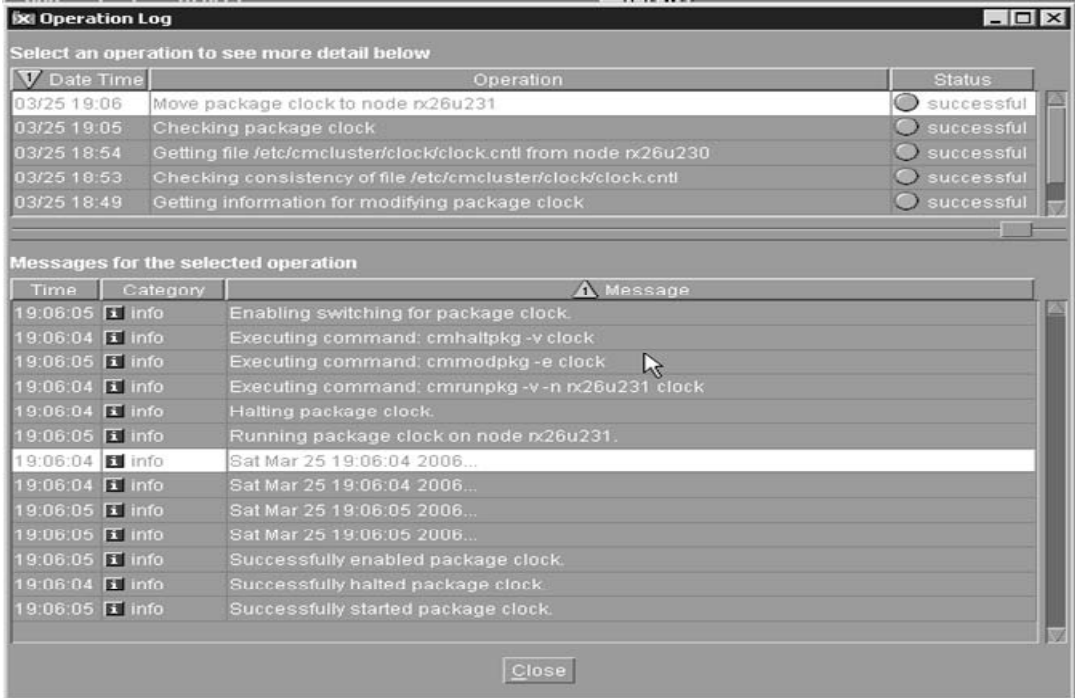
Student Notes

The “Roles” tab allows changes to the Access Control Policy.

18-19. SLIDE: Operation Log



Operation Log



Operation Log

Select an operation to see more detail below

Date Time	Operation	Status
03/25 19:06	Move package clock to node rx26u231	successful
03/25 19:05	Checking package clock	successful
03/25 18:54	Getting file /etc/cmcluster/clock/clock.cnf from node rx26u230	successful
03/25 18:53	Checking consistency of file /etc/cmcluster/clock/clock.cnf	successful
03/25 18:49	Getting information for modifying package clock	successful

Messages for the selected operation

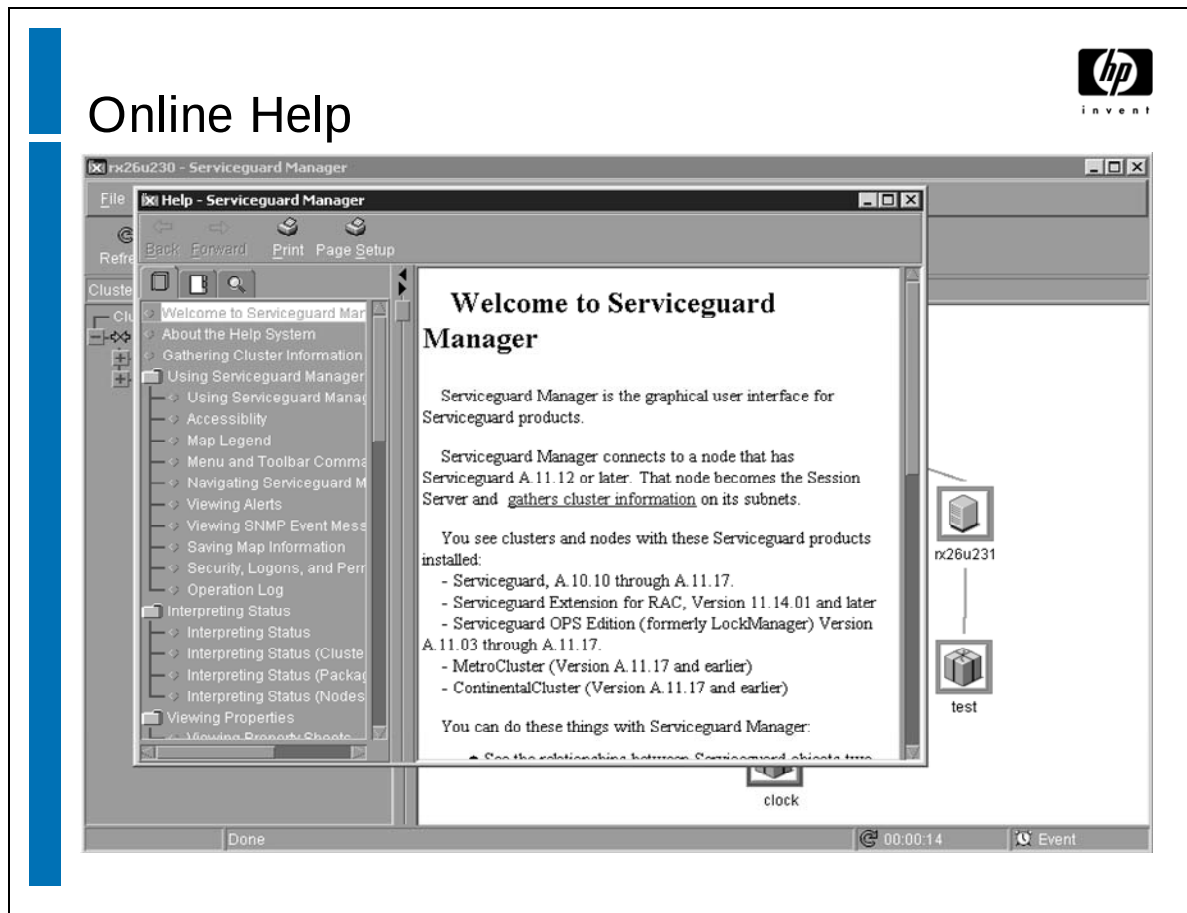
Time	Category	Message
19:06:05	Info	Enabling switching for package clock.
19:06:04	Info	Executing command: cmhaltpkg -v clock
19:06:05	Info	Executing command: cmmodpkg -e clock
19:06:04	Info	Executing command: cmrunpkg -v -n rx26u231 clock
19:06:04	Info	Halting package clock.
19:06:05	Info	Running package clock on node rx26u231.
19:06:04	Info	Sat Mar 25 19:06:04 2006...
19:06:04	Info	Sat Mar 25 19:06:04 2006...
19:06:05	Info	Sat Mar 25 19:06:05 2006...
19:06:05	Info	Sat Mar 25 19:06:05 2006...
19:06:05	Info	Successfully enabled package clock.
19:06:04	Info	Successfully halted package clock.
19:06:05	Info	Successfully started package clock.

Close

Student Notes

Nearly every operation performed by Serviceguard Manager will launch an Operation Log window after completion. This window logs the actual commands that were executed to complete the task, a benefit to those operators who would like to understand better what is happening “behind the scenes”.

18-20. SLIDE: Online Help



Student Notes

Getting Help in Serviceguard Manager

Serviceguard Manager contains extensive help. The above screen shot shows only the first screen after clicking on the Serviceguard Manager Help button.

While you are working in Serviceguard Manager, you can open help on such topics as:

- Properties
- Map Legend
- Configuring Clusters
- Configuring Packages
- Troubleshooting

18–21. LAB: Exploring Serviceguard Manager

Directions

Complete the following tasks.

1. Start up Serviceguard Manager.

Answer:

```
# cd /opt/sgmgr/bin  
# ./sgmgr -c <your_cluster_name>
```

2. Log in to the Serviceguard Manager GUI.

Answer:

```
Server Name: node1  
User Name:  root  
Password:   <as specified by your instructor>
```

3. Explore the cluster shown in the map.

Answer:

On the left side of the screen, single-click on the (+) symbol next to the cluster name.

List the node names in the cluster and the package names in the cluster:

Node Names in the Cluster	Packages Available in the Cluster

4. On the right side of the screen, double click the cluster icon and note the same nodes and packages in the cluster – this time in graphical form.

Note: If the cluster or package is/are up, the icons have a green border. If the cluster or package is/are down, the icons have a red border.

5. Explore the properties for the cluster and nodes.

Answer:

Single-click on the icon for the cluster.

Right-click.

Select Properties of <cluster_name>.

Select one or more of the property tabs shown.

After exploring several properties, close the properties window.

6. Using the Serviceguard Manager GUI, halt your cluster.

Answer:

Single-click on the cluster icon.

From the Actions pull-down menu, select Administration.

From the Administration pull-down, select Halt cluster <cluster_name>.

If prompted with a Confirm Operation popup window, confirm this action by clicking OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

7. Using the Serviceguard Manager GUI, start your cluster.

Answer:

Single-click on the cluster icon.

From the Actions pull-down menu, select Administration.

From the Administration pull-down menu, select Run cluster <cluster_name>.

At the Enable Network Probing popup window, click No.

If prompted with a Confirm Operation popup window, confirm this action by clicking OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

8. Using the Serviceguard Manager GUI, halt your test package, then observe the window in which the actual commands are logged.

Answer:

Instead of going to the Actions pull-down menu, let us try a slightly different procedure to accomplish the same task. Right click on the test package icon and select Halt package test.

If prompted with a Confirm Operation popup window, confirm this action by clicking OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

9. Now, start your test package on the other node.

Answer:

Right click on the test package icon and select Run package on node... Then select the other node.

If prompted with a Confirm Operation popup window, confirm this action by clicking OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

10. If there is a red "X" next to the test package icon, correct the problem.

Answer:

Right click on the test package icon.

Check the box next to "Package Switching".

If prompted with a Confirm Operation popup window, confirm this action by clicking OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

The red "X" next to the test package icon should now be gone. This indicates that all switches concerning this package's failover abilities have now been enabled.

11. Another feature of Serviceguard Manager is the ability to move a package. Currently, there is no command such as "cmmovepkg", but Serviceguard Manager can do it! To see this feature, move your test package back to its original node.

Answer:

Single-click on the test package icon.

From the Actions pull-down menu, select Administration.

From the Administration pull-down menu, click on Move Package to Node and select the node (to which the package will be moved) by clicking on the node name.

If prompted to confirm this move, click OK.

Watch the Operations Log popup window to see a detailed outline of events.

Click Close to close the Operations Log popup window.

12. Explore the online Help.

Answer:

Select Help at the top of the Serviceguard Manager screen (the "?" icon).

Select one of the help topics.

Select another Help topic.

Can you print a Help topic?

Close the Serviceguard Manager "Help" window.

13. Test connecting to other clusters in the classroom.

Answer:

Select File from the menu.

Select New Session.

Enter server name, user name, and password for another cluster. In the Clusters to view section, select: View all clusters and unused nodes.

Select connect.

14. As a courtesy to the owner of this other cluster, ask their permission before continuing. As time and interest allow, investigate this new cluster using several different features of Serviceguard Manager.

15. Exit Serviceguard Manager.

Answer:

Click on File, then click on Exit.

18-22. LAB: Creating a Package with Serviceguard Manager

Directions

Complete the following tasks.

1. Start up Serviceguard Manager, if it is not already running.

Answer:

```
# cd /opt/sgmgr/bin  
# ./sgmgr -c <your_cluster_name>
```

2. If needed, log in to the Serviceguard Manager GUI.

Answer:

```
Server Name:  node1  
User Name:    root  
Password:     <as specified by your instructor>
```

3. From the Actions pull down menu, select Configuration, and then click on Create Package.

Answer:

Click on Actions pull down menu.

Slide down to Configuration.

Click on Create Package.

4. If requested at the Authorization Dialog popup window, login as root with the appropriate password.

Answer:

Login: root

Password: < as specified by your instructor >

5. At the Operation Log popup window, you will see a message that says Getting Configuration for New Package ---- In progress. At this point, wait here for the next popup window.
6. In the Create Package Configuration popup window, ...
 - a. At this point, note that there are seven tabs across the top of the window. We will investigate each. Initially, we are first in the Nodes tab of the Create Package Configuration popup window.
 - b. With your mouse, highlight the generic name of the new package (New_Package1), and type in xload. xload will be our next package that we configure.
 - c. Still in the Nodes tab, note that, by checking the appropriate radio button, we could decide to use only a subset of all the available cluster nodes for this new xload package. However, leave this at its default: Enable package to run on all cluster nodes.
 - d. Click on the Parameters tab. In this tab, note the different fields and how their values could be modified. For this lab, leave all values at their default.
 - e. Click on the Networks tab. In this tab, towards to right center, in the “Subnets monitored by the cluster” field, click on the 10-net subnet, and then, in the middle, click Add, to add this subnet to our package.
 - f. Note also, towards the bottom of this tab, that there is an area where we could configure IP addresses. Our xload package does not require an IP address, so leave these fields at their defaults.
 - g. Click on the Services tab. In this tab, we assign the Service_Name of xload_service. Type this service name into the specified field, then click Add. Note that, when you click Add, a new popup window appears. This window allows you to specify several values. For us, we need only specify the number of restarts **and** the Service command. Use 3 restarts, and the following Service command:
`/usr/contrib/bin/X11/xload -display _____ -title _____`
For your display option, launch a ReflectionX session and find the value of your DISPLAY variable. Use that value here. For title, use your first name. Then click OK.
 - h. Click on the EMS Resources tab. In this tab, we could configure EMS resources. We will study EMS Resources in a later module. For now, leave this tab at its defaults.
 - i. Click on the Control Script tab. Note that we configure LVM Volume Groups, File Systems, Customer Defined Functions, and/or VxVM Disk Groups here. Since we are not using any of these for our xload package, leave all fields at their defaults.

- j. Click on the Roles tab. In this tab, we could configure Access Control Policies for different users and different servers. Again leave all values at their defaults.
 - k. Click the Apply button at the lower center of your window. A popup appears that asks Apply Configuration? Click OK to apply the configuration.
 - l. This returns us to the Operation Log window, where we see a message that says Checking & applying package xload --- in progress.
 - m. Once the Operations Log window reports Successful (instead of in progress), click the close button to close the Operations Log window.
 - n. This returns us to the Serviceguard Manager main window. We should now see the xload package. At this point, single-click on the xload package icon, click on the Actions pull-down menu, click on Administration, and click on Run package xload. If prompted with a Confirm Operation popup window, click OK to begin the xload package.
 - o. This will bring an Operations Log popup window, where you can watch the progress as the xload package begins running. When you see the Successfully started package xload message, click Close to return you to the Serviceguard Manager main screen.
 - p. Notice that the xload package has a red "X" near the package icon. Right-click on the xload package icon, and click on Properties of xload. In the xload Package Property Sheet popup window, we can see that Package Switching is disabled. Click Close to close the popup window.
 - q. To enable Package switching, single-click on the xload icon, then click on the Actions pull down menu, then click on Administration. Place a check mark on Package Switching.
 - r. Another Operation Log popup window appears where we can watch the progress in real time. Wait for the Successfully enabled package xload message, and then click OK to close this window.
7. The xload package should now be running correctly with all switches enabled.
8. Congratulations on configuring your first package using Serviceguard Manager!

Module 19 — Troubleshooting Scenarios and Final Review

Objectives

Upon completion of this module, you will be able to do the following:

- Perform troubleshooting activities to resolve nine different Serviceguard configuration problems.
- Successfully complete 15 final review questions.

Some Important Notes

There are two parts to this module:

- "Troubleshooting Scenarios", and
- "Final review"

To accomplish both parts of this module, it is important that you accomplish the following:

- a. Before beginning the "Troubleshooting Scenarios" section, some minor editing is required. Go to the `/labs/mod_19/trbl` directory. Once there, change line 11 in the `trbl.sh` script to point to the disk you want to use for this section. This therefore means that you will need disk space available. If necessary, free up some disk space. Since we are now done with all packages previously configured, it is certainly ok to `cmdeleteconf` some packages and `vgexport` some volume groups in order to accomplish this.
- b. Also, the `trbl.sh` script will setup volume group `vg04`. Therefore, again, we will need disk space to be available. Again, it is certainly ok to `cmdeleteconf` some packages and `vgexport` some volume groups in order to accomplish this.
- c. Be sure that volume group `vg04` with minor number `0x040000` is not currently in use.
- d. Do not modify the `vg04` setting in the `trbl.sh` script. It works fine as is as long as you have an available disk.
- e. Ensure that there are no NFS clients still mounted. The setup script will remove the NFS package. Any remaining clients will therefore hang.
- f. Ensure the "remsh" functionality works between the nodes of your cluster. This implies that we need to create a `~root/.rhosts` file on each node in our cluster.
- g. The second part of this final module asks you to run `/labs/mod_19/final/game_parisc` or `/labs/mod_19/final/integrity`), depending upon your platform.
- h. Good luck in this final module !!

Please be sure to complete all of the above before continuing.

19-1. SLIDE: Troubleshooting Scenarios



Troubleshooting Scenarios

```
# /labs/mod_19/trbl/trbl.sh
```

TROUBLESHOOTING Scenarios

```
*****
|
| 1. There exists a conflict within us
| 2. Missing the point ?
| 3. Busy, busy, busy
|
| 4. The import of this problem is crucial
| 5. What makes Johnny run ?
| 6. Fee, Fi, Foh, Fum
|
| 7. Care to comment ?
| 8. Thrown for a loop ?
| 9. What's in a name ?
|
*****
```

Enter 1-9, (q)uit:

Student Notes

This module focuses on nine different troubleshooting scenarios that can exist in a Serviceguard environment. In this section of the module, troubleshoot as many of the above scenarios as possible as presented when you make a selection using the interactive simulation program. The program instructions will appear when you execute the startup script as described below.

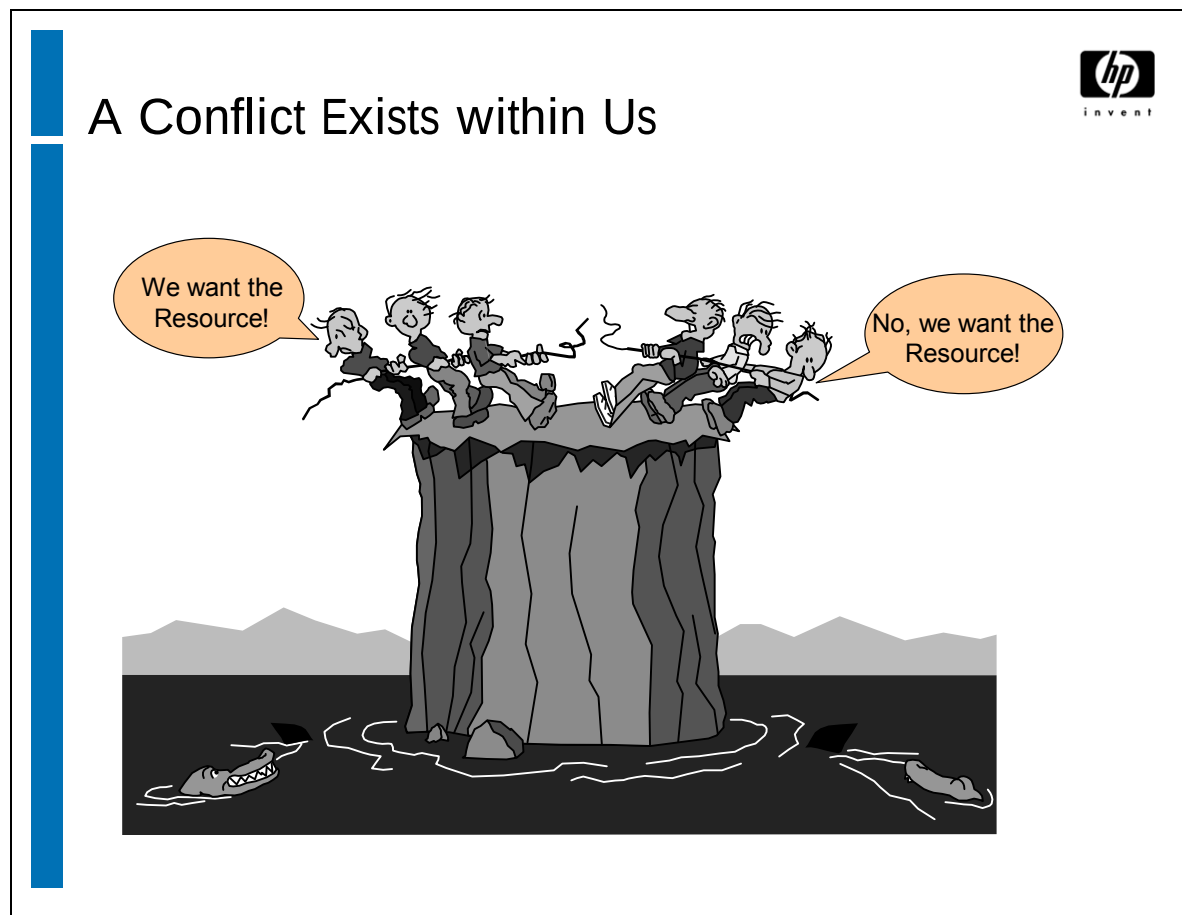
The last section of this module contains a set of "final review" questions to test your knowledge of Serviceguard. The script will create a new volume group (**vg04**), logical volume (`/dev/vg04/trbl`), and mount point (`trbl`). It will also create a new package (`/etc/cmcluster/pkg4`).

To start the simulation program execute:

```
# cd /labs/mod_19/trbl
# ./trbl.sh
```

To exit one simulation and go on to another, type exit at the `MCSG>` prompt. To quit the troubleshooting scenario, at the Main menu, enter quit at the `MCSG>` prompt.
Good luck!

19-2. SLIDE: A Conflict Exists within Us



Student Notes

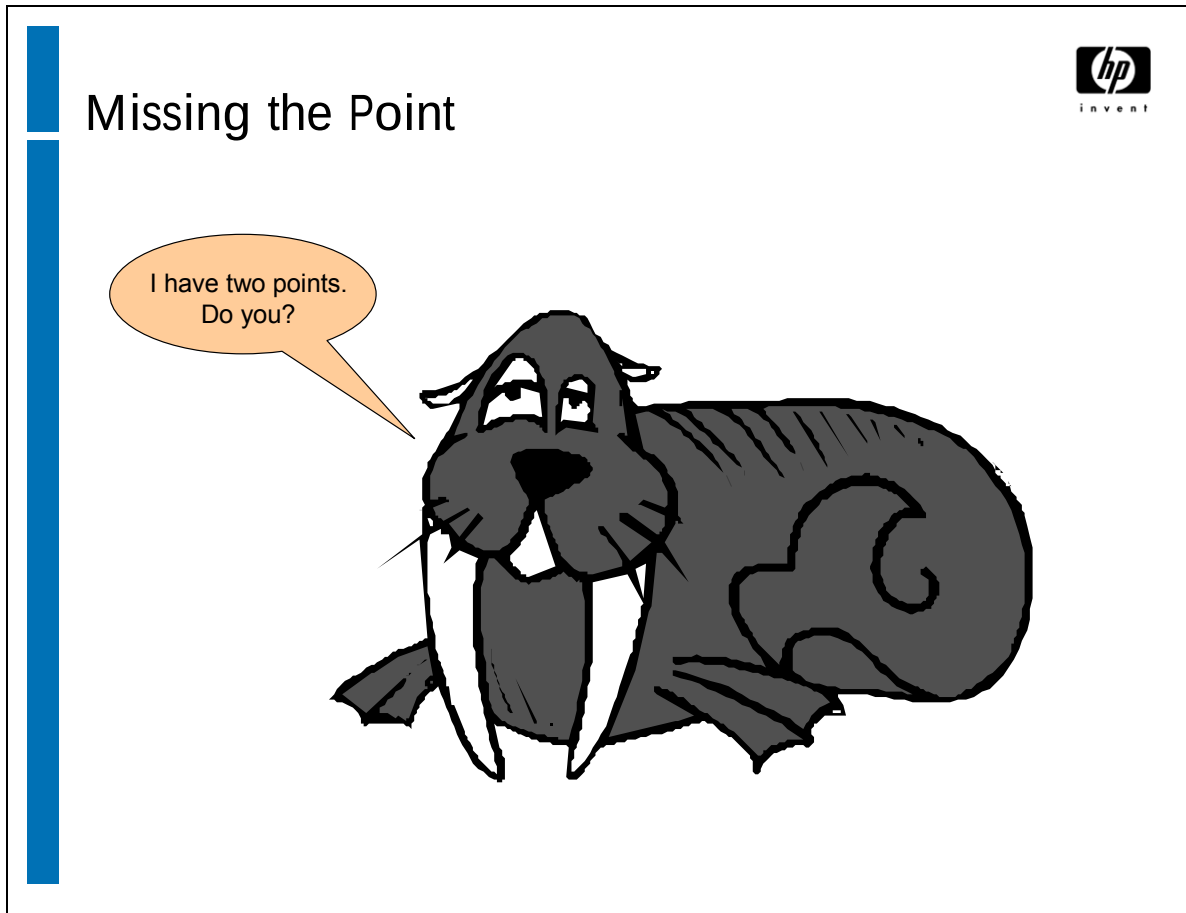
```
# cmruncl
cmruncl : Waiting for cluster to form.....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c17 and pkg log files for more detailed information.

# cd /etc/cmcluster/pkg4

# tail *.log
##### Node "r208c17": Starting package at Mon Mar 12 04:26:30 PST 2001
#####
Mar 12 04:26:30 - "r208c17": Activating volume group vg04 with exclusive option.
vgchange: Activation mode requested for the volume group "/dev/vg04" conflicts with
configured mode.
ERROR: Function activate_volume_group
ERROR: Failed to activate vg04
Mar 12 04:26:30 - Node "r208c17": Deactivating volume group vg04
vgchange: Volume group "vg04" has been successfully changed.
```

19-3. SLIDE: Missing the Point



Student Notes

```
# cmruncl
cmruncl : Waiting for cluster to form....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c17 and pkg log files for more detailed information.

# cd /etc/cmcluster/pkg4

# tail *.log
Mar 12 05:27:19 - Node "r208c17": Checking filesystems:
/dev/vg04/trbl
file system is clean - log replay is not required
Mar 12 05:27:20 - Node "r208c17": Mounting /dev/vg04/trbl at /trbl
mount: /trbl: No such file or directory
```

***** Continued on next page *****

Troubleshooting Scenarios and Final Review

```
ERROR: Function check_and_mount
ERROR: Failed to mount /dev/vg04/trbl to /trbl
Mar 12 05:27:20 - Node "r208c17": Deactivating volume group vg04
Deactivated volume group in Exclusive Mode.
Volume group "vg04" has been successfully changed.
```

19-4. SLIDE: Busy, Busy, Busy



Student Notes

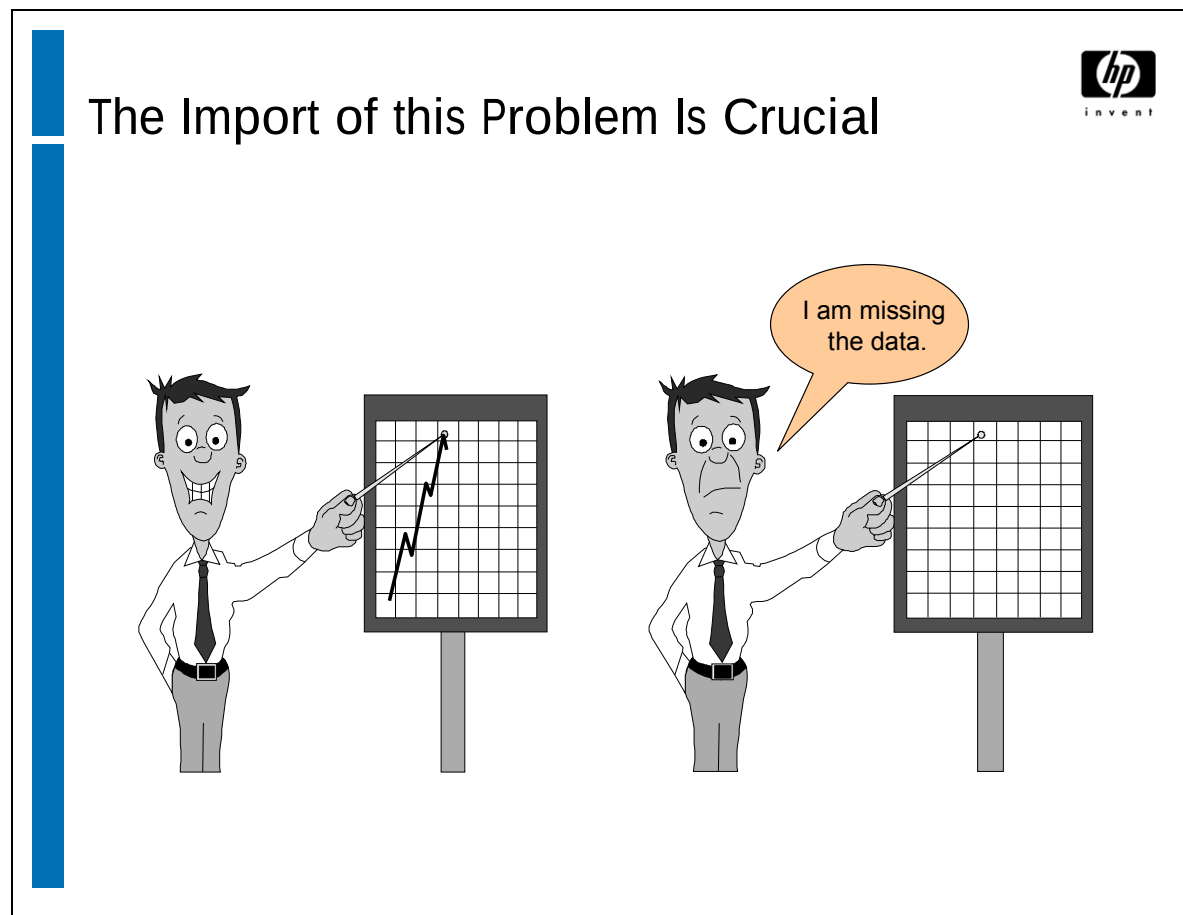
```
# cmruncl
cmruncl : Waiting for cluster to form....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c17 and pkg log files for more detailed information.

# cd /etc/cmcluster/pkg4

# tail *.log
Mar 12 05:35:27 - Node "r208c17": Mounting /dev/vg04/trbl at /trbl
vxfs mount: /dev/vg04/trbl is already mounted, /trbl is busy,
or allowable number of mount points exceeded
ERROR: Function check_and_mount
ERROR: Failed to mount /dev/vg04/trbl to /trbl
Mar 12 05:35:28 - Node "r208c17": Deactivating volume group vg04
Deactivated volume group in Exclusive Mode.
Volume group "vg04" has been successfully changed.
```

19-5. SLIDE: The Import of This Problem Is Crucial



Student Notes

```
# cmruncl
cmruncl : Waiting for cluster to form.....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.
```

```
# cmrunpkg pkg4
cmrunpkg : Warning: Package pkg4 is already running.
Check the syslog on node r208c17 and pkg log files for more detailed
information
```

```
# cmhaltnode -f
Disabling package switching to all nodes being halted.
Warning: Do not modify or enable packages until the halt operation is
completed
```

```
Halting cluster services on node r208c17
cmhaltnode : Successfully halted all nodes specified.
```

Halt operation completed.

```
# cmrunpkg -n node2 pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c18 and pkg log files for more detailed
information

# remsh node2 tail /etc/cmcluster/pkg4/pkg4.sh.log

#### Node "r208c18": Starting package at Mon Jul  7 20:45:49 PDT 2003####
Jul  7 20:45:49 - "r208c18": Activating volume group vg04 with exclusive
option.

vgchange: Volume group "/dev/vg04" does not exist in the "/etc/lvmtab"
file.
ERROR: Function activate_volume_group
ERROR: Failed to activate vg04
Jul  7 20:45:49 - Node "r208c18": Deactivating volume group vg04
vgchange: Volume group "/dev/vg04" does not exist in the "/etc/lvmtab"
file.
ERROR: Function deactivate_volume_group
ERROR: Failed to deactivate vg04
```

19-6. SLIDE: What Makes Johnny Run?



Student Notes

```
# cmruncl
cmruncl : Waiting for cluster to form.....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.
```

```
# cmrunpkg pkg4
cmrunpkg : Warning: Package pkg4 is already running.
Check the syslog on node r208c17 and pkg log files for more detailed
information
```

```
# cmhaltnode -f
Disabling package switching to all nodes being halted.
Warning: Do not modify or enable packages until the halt operation is
completed
```

Halting cluster services on node r208c17

```
..
cmhaltnode : Successfully halted all nodes specified.
Halt operation completed.
```

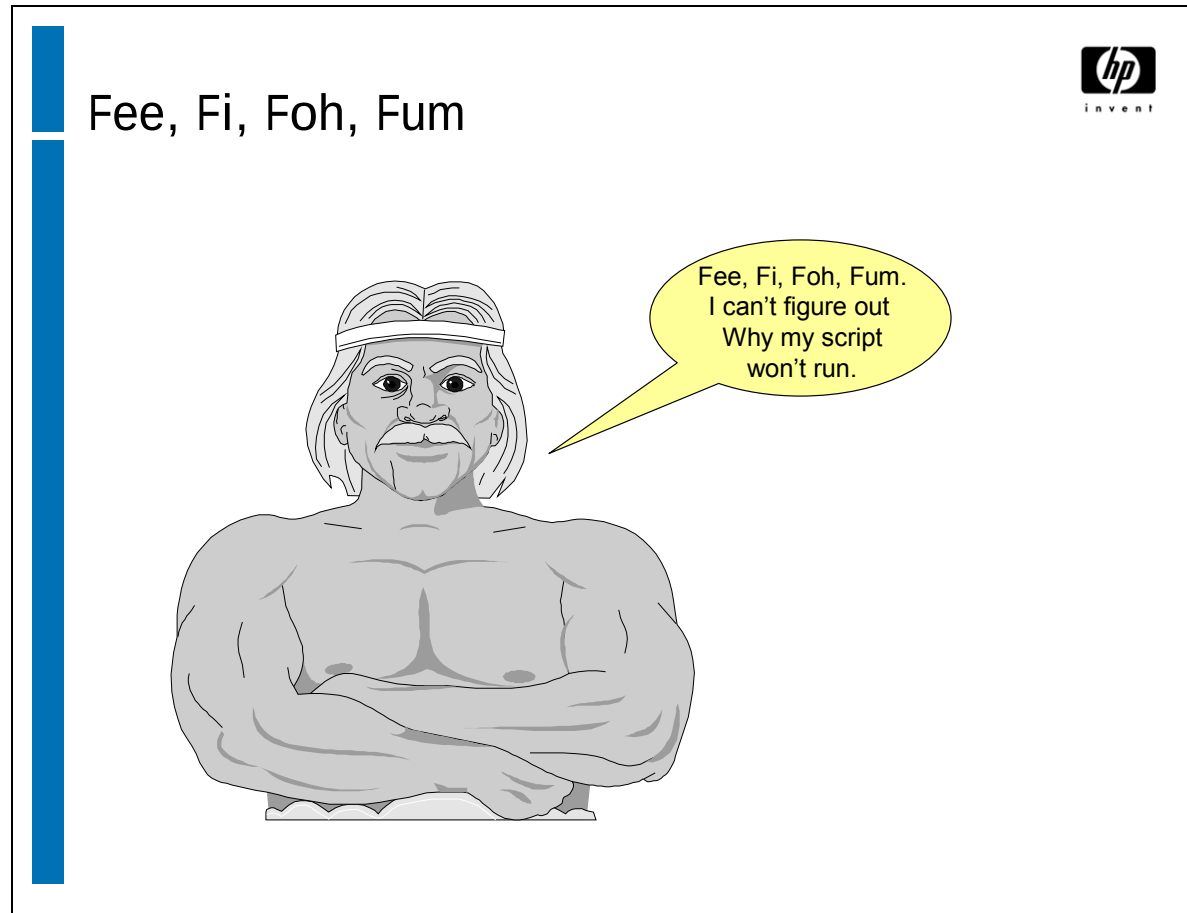


```
# cmrunpkg -n node2 pkg4
cmrunpkg : Unknown error returned from daemon: Error 0
Check the syslog on node r208c18 and pkg log files for more detailed
information
.

# remsh node2 tail /etc/cmcluster/pkg4/pkg4.sh.log

# remsh node2 tail /var/adm/syslog/syslog.log
Jul 7 20:49:23 r208c18 cmclld: 1 nodes have formed a new cluster, sequence
#2
Jul 7 20:49:23 r208c18 cmclld: The new active cluster membership is:
r208c18(id=2)
Jul 7 20:49:41 r208c18 cmclld: Request from node r208c18 to start package
pkg4 on node r208c18.
Jul 7 20:49:41 r208c18 cmclld: Executing '/etc/cmcluster/pkg4/pkg4.sh
start' for package pkg4, as service PKG*3073.
Jul 7 20:49:41 r208c18 cmsrvassisd[14428]: File
'/etc/cmcluster/pkg4/pkg4.sh' does not have execute permission.
Jul 7 20:49:41 r208c18 cmsrvassisd[14428]: Unable to start service
PKG*3073.
Jul 7 20:49:41 r208c18 cmclld: Package pkg4 run script failed.
Jul 7 20:49:41 r208c18 cmclld: Examine the file
/etc/cmcluster/pkg4/pkg4.sh.log for more details.
Jul 7 20:49:41 r208c18 cmclld: Disabled node r208c18 from running package
pkg4.
Jul 7 20:49:41 r208c18 cmclld: Unable to start package pkg4. Node r208c18
is not able to run it.
```

19-7. SLIDE: Fee, Fi, Foh, Fum



Student Notes

```
# cmruncl
cmruncl : Waiting for cluster to form....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Node is currently unable to run package pkg4.
Check the syslog on node r208c17 and pkg log files for more detailed information.

# cd /etc/cmcluster/pkg4

# tail *.log
/etc/cmcluster/pkg4/pkg4.sh[142]: Syntax error at line 150 : `}' is not expected.
```

19-8. SLIDE: Care to Comment?



Student Notes

```
# cmruncl
cmruncl : Waiting for cluster to form.....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Warning: Package pkg4 is already running.
Check the syslog on node r208c17 and pkg log files for more detailed
information

cmrunpkg : Completed successfully on all packages specified.

# cmhaltnode -f
Disabling package switching to all nodes being halted.
Warning: Do not modify or enable packages until the halt operation is
completed

Halting Package pkg4
Halting cluster services on node r208c17
```

Troubleshooting Scenarios and Final Review

cmhaltnode : Successfully halted all nodes specified.
Halt operation completed.

```
# cmrunpkg -n node2 pkg4
```

cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c18 and pkg log files for more detailed
information

```
# cd /etc/cmcluster/pkg4
```

```
# tail *.log
```

Volume group "vg04" has been successfully changed.

Node "r208c17": Package start completed at Mon Jul 7 20:52:02 PDT
2003 ####

Node "r208c17": Halting package at Mon Jul 7 20:52:25 PDT 2003####
Jul 7 20:52:25 - Node "r208c17": Deactivating volume group vg04
Deactivated volume group in Exclusive Mode.
Volume group "vg04" has been successfully changed.

Node "r208c17": Package halt completed at Mon Jul 7 20:52:25 PDT
2003 ####

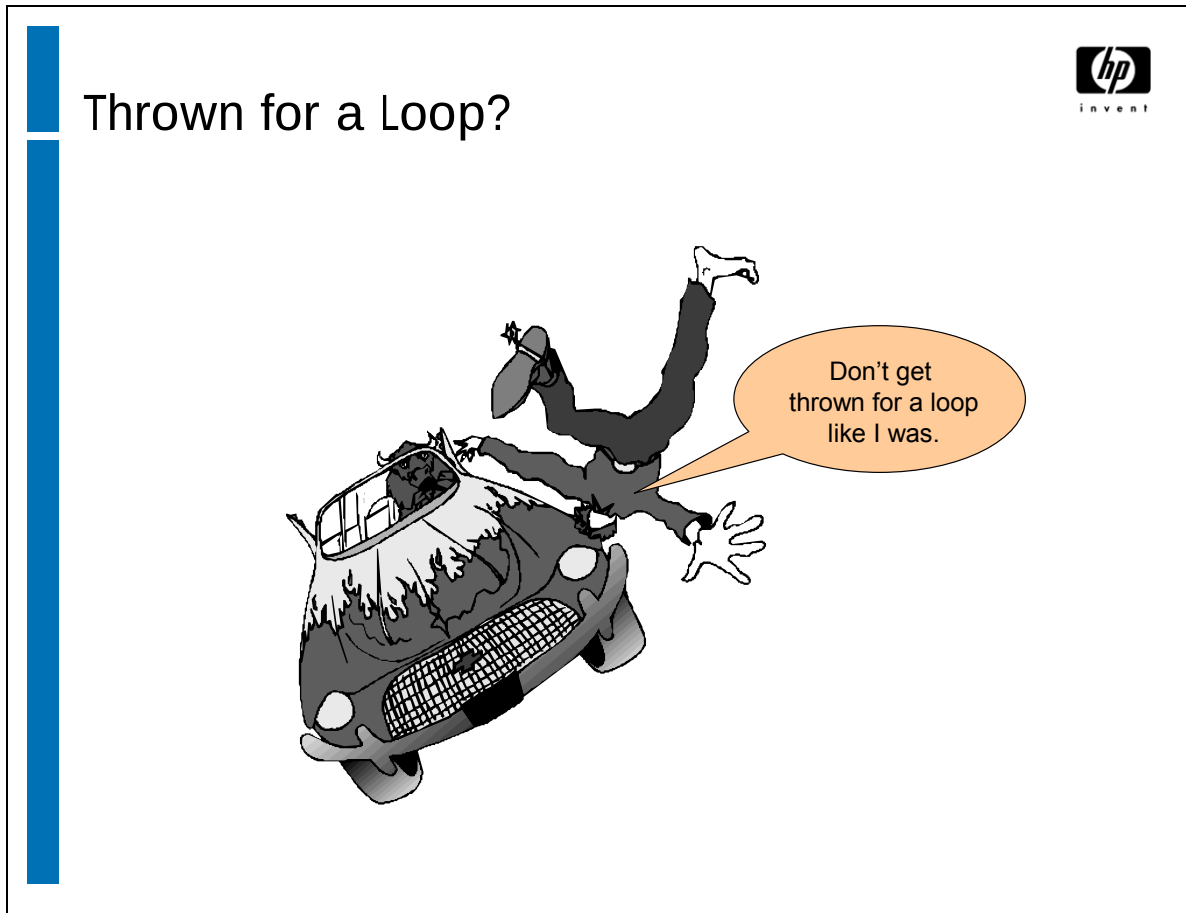
```
# tail /var/adm/syslog/syslog.log
```

Jul 7 20:52:25 r208c17 cmcld: Enabled switching for package pkg4.
Jul 7 20:52:25 r208c17 cmcld: Package pkg4 cannot run on this node because
switching has been disabled for this node
Jul 7 20:52:26 r208c17 cmtaped[20822]: cmtaped terminating. (ATS 1.14)
Jul 7 20:52:27 r208c17 cmcld: Disabled switching for package pkg4.
Jul 7 20:52:27 r208c17 cmcld: This node (r208c17) has ceased cluster
activities
Jul 7 20:52:27 r208c17 cmcld: Daemon exiting
Jul 7 20:52:27 r208c17 cmsrvassistd[20819]: The cluster daemon aborted our
connection.
Jul 7 20:52:27 r208c17 cmsrvassistd[20819]: Lost connection with
Serviceguard cluster daemon (cmcld): Software caused connection abort
Jul 7 20:52:27 r208c17 cmclconfd[20741]: The cluster daemon aborted our
connection.
Jul 7 20:52:44 r208c17 CM-CMD[20859]: cmrunpkg -n r208c18 pkg4

```
# remsh r208c18 tail /etc/cmcluster/pkg4/pkg4.sh.log
```

Node "r208c18": Starting package at Mon Jul 7 20:52:47 PDT 2003 ####
Jul 7 20:52:47 - "r208c18": Activating volume group vg04 with exclusive
option.
vgchange: Volume group "/dev/vg04" does not exist in the "/etc/lvmtab"
file.
ERROR: Function activate_volume_group
ERROR: Failed to activate vg04
Jul 7 20:52:47 - Node "r208c18": Deactivating volume group vg04
vgchange: Volume group "/dev/vg04" does not exist in the "/etc/lvmtab"
file.
ERROR: Function deactivate_volume_group
ERROR: Failed to deactivate vg04

19-9. SLIDE: Thrown for a Loop?



Student Notes

```
# cmruncl -v
cmruncl : Waiting for cluster to form.....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg -v pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c17 and pkg log files for more detailed
information

# tail /var/adm/syslog/syslog.log
Jul  3 19:37:40 r208c17 CM-CMD[12547]: cmrunpkg pkg4
Jul  3 19:37:40 r208c17 cmclld: Request from node r208c17 to start package
pkg4 on node r208c17.
Jul  3 19:37:40 r208c17 cmclld: Executing '/etc/cmcluster/pkg4/pkg4.sh
start' for package pkg4, as service PKG*50689.
Jul  3 19:37:42 r208c17 LVM[12558]: vgchange -a e vg04
Jul  3 19:37:43 r208c17 LVM[12596]: vgchange -a n vg04
Jul  3 19:37:43 r208c17 cmclld: Service PKG*50689 terminated due to an
exit(1).
```

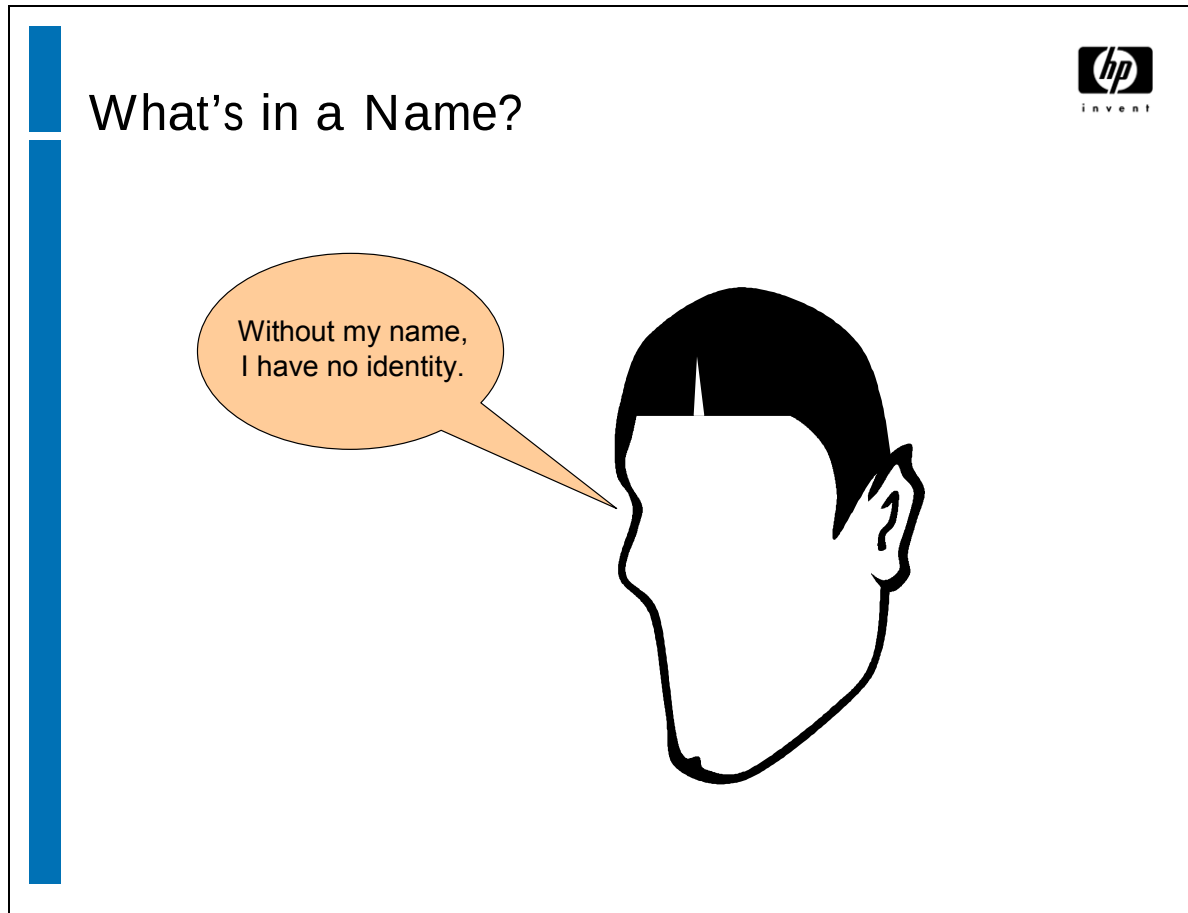
Troubleshooting Scenarios and Final Review

```
Jul  3 19:37:43 r208c17 cmcld: Package pkg4 run script exited with  
NO_RESTART.  
Jul  3 19:37:43 r208c17 cmcld: Examine the file  
/etc/cmcluster/pkg4/pkg4.sh.log for more details.  
Jul  3 19:37:43 r208c17 cmcld: Switching disabled on package pkg4.  
Jul  3 19:37:43 r208c17 cmcld: Unable to start package pkg4. Node r208c17  
is not able to run it.
```

```
# cd /etc/cmcluster/pkg4
```

```
# tail *.log  
/dev/vg04/trbl  
file system is clean - log replay is not required  
Jul  7 20:55:47 - Node "r208c17": Mounting /dev/vg04/trbl at /trbl  
vxfs mount: /dev/vg04/trbl is already mounted, /trbl is busy,  
or allowable number of mount points exceeded  
ERROR: Function check_and_mount  
ERROR: Failed to mount /dev/vg04/trbl to /trbl  
Jul  7 20:55:47 - Node "r208c17": Deactivating volume group vg04  
Deactivated volume group in Exclusive Mode.  
Volume group "vg04" has been successfully changed.
```

19-10. SLIDE: What's in a Name?



Student Notes

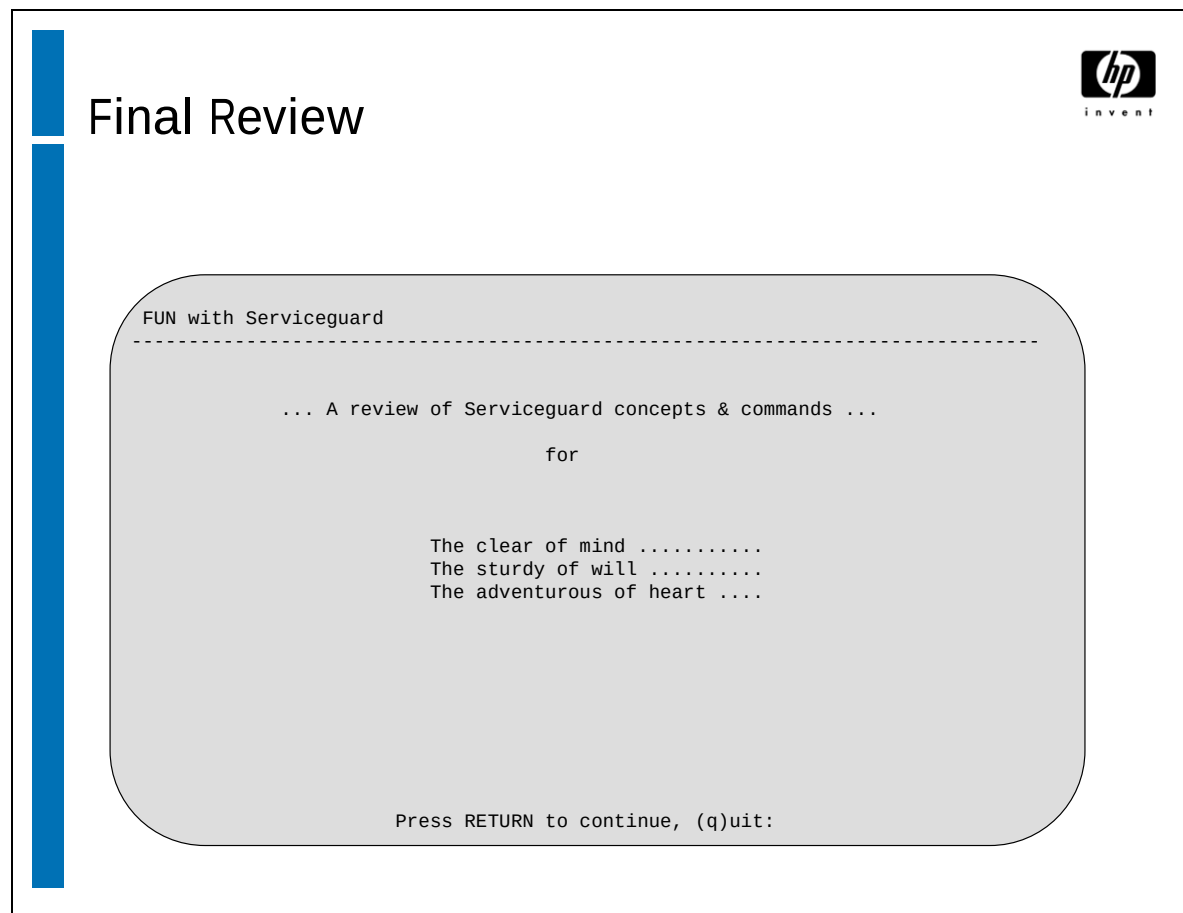
```
# cmruncl
cmruncl : Waiting for cluster to form....
cmruncl : Cluster successfully formed.
cmruncl : Check the syslog files on all nodes in the cluster
cmruncl : to verify that no warnings occurred during startup.

# cmrunpkg pkg4
cmrunpkg : Script failed with no restart: pkg4 should not be restarted.
Check the syslog on node r208c17 and pkg log files for more detailed information.

# cd /etc/cmcluster/pkg4

# tail *.log
/dev/vg04/trbl
file system is clean - log replay is not required
Jul  7 21:00:09 - Node "r208c17": Mounting /dev/vg04/trbl at /trbl
vxfs mount: /dev/vg04/trbl is already mounted, /trbl is busy, or allowable number
of mount points exceeded
ERROR: Function check_and_mount
ERROR: Failed to mount /dev/vg04/trbl to /trbl
Jul  7 21:00:09 - Node "r208c17": Deactivating volume group vg04
Deactivated volume group in Exclusive Mode.
```

19-11. SLIDE: Final Review



Student Notes

To start the review, enter the following at the command prompt:

```
# cd /labs/mod_19/final
```

Once in this directory, you will see two versions of the game program: the first is `game_parisc` (for use in those classes using PA-RISC equipment), while the second is `game_integrity` (for use in those classes using Integrity equipment). If you are not sure what kind of equipment you are using, ask your instructor.

```
# ./game_parisc
```

or

```
# ./game_integrity
```

This "question and answer" game serves as a review of cluster and package concepts and commands encountered during this class.

Multiple-choice questions are randomly chosen from a large "pool" of available questions; thus, if you replay the game, you may see questions not yet encountered. Your score is increased by one for each question answered correctly; however, it decreases by one for each question missed.

To "complete" the game, you must tally 15 points. If all possible questions from the pool have been presented, and you have not yet reached 15 points, the game will cease at that point.

You may quit the game at any time by answering 'q' to the current question. Partial games are not saved, and thus cannot be resumed.

19–12. LAB Solution: Troubleshooting Scenarios

Scenario #1 There exists a conflict within us

- Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
- Problem: vg04 fails to activate.
The exclusive activation mode bit for vg04 is not set.
- Solution: Start the cluster.
Reset the activation mode bit. (vgchange -c y vg04)
Run the package.

Scenario #2 Missing the point

- Hint: The package (pkg4) should be running on its primary node.
It should mount /dev/vg04/trbl on /trbl.
- Problem: The mount fails
The mount point (/trbl) is missing.
- Solution: Create the mount point (mkdir /trbl)
Run the package.

Scenario #3 Busy, busy, busy

- Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
- Problem: The mount fails.
The mount point (/trbl) is busy.
- Solution: Identify processes using the mount point (fuser /trbl).
Kill the processes.
Run the package.

Scenario #4 The import of this problem is crucial!

Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
Halt the primary node (cmhaltnode -f)
The package (pkg4) should switch to the other node.

Problem: The package does not switch.
vg04 fails to activate.
vg04 is not defined on the other node.

Solution: Import the vg04 definition on the other node.
mkdir /dev/vg04
mknod /dev/vg04/group c 64 0x040000
vgimport -m /tmp/vg04.map vg04 /dev/dsk/cXtYdZ

Run the package on the adaptive node.

Scenario #5 What makes Johnny run?

Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.

Halt the primary node (cmhaltnode -f)
The package (pkg4) should switch to the other node.
Look in sys log for helpful information.

Problem: The package does not switch.
The package control script (pkg4.sh) is not executable.

Solution: Make the package control script executable (chmod 700 pkg4.sh)
Run the package on the adoptive node.

Scenario #6 Fee, Fi, Foh, Fum

Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.

Problem: The customer_defined_run_cmds function contains a syntax error.
The if statement is missing a fi

Solution: Add the needed fi to the customer_defined_run_cmds function.
Run the package.

Scenario #7 Care to comment?

- Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
- Halt the primary node (`cmhaltnode -f`).
The package (pkg4) should switch to the other node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
- Problem: The package switches but does not mount the lvol.
The lvol definition is commented out in the control script.
- Solution: Halt the package.
Uncomment the lvol definition in the control script (pkg4.sh)
Run the package on the adoptive node.

Scenario #8 Thrown for a loop

- Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
- It should run a service named "servmon".
Check the status of the package after a 10-15 second wait.
- Look in `syslog.log` for helpful information.
- Problem: The package starts but halts shortly thereafter.
The service program exit causes the package to switch/halt.
- Solution: Add an infinite loop to the servmon code to prevent the exit.
Run the package.

Scenario #9 What's in a name?

- Hint: The package (pkg4) should be running on its primary node.
It should activate vg04 and mount /dev/vg04/trbl on /trbl.
It should run a customer-defined program.
- Problem: Error in the `customer_defined_run_cmds` function.
The program name should be: `/labs/mod_17/trbl/start_daemons`
- Solution: Fix the program name in the customer function.
Run the package.

Module 20 — Using HP Serviceguard with Integrity Virtual Machines

Objectives

Upon completion of this module, you will be able to do the following:

- Define Integrity Virtual Machine terminology
- List the Serviceguard/Integrity VM use models
- Configure Guest Virtual Machines as Serviceguard nodes
- Configure Guest Virtual Machines as Serviceguard packages

20-1. SLIDE: HP Integrity Virtual Machines Overview


invent

HP Integrity Virtual Machines Overview

HP Integrity Virtual Machine (HPVM)

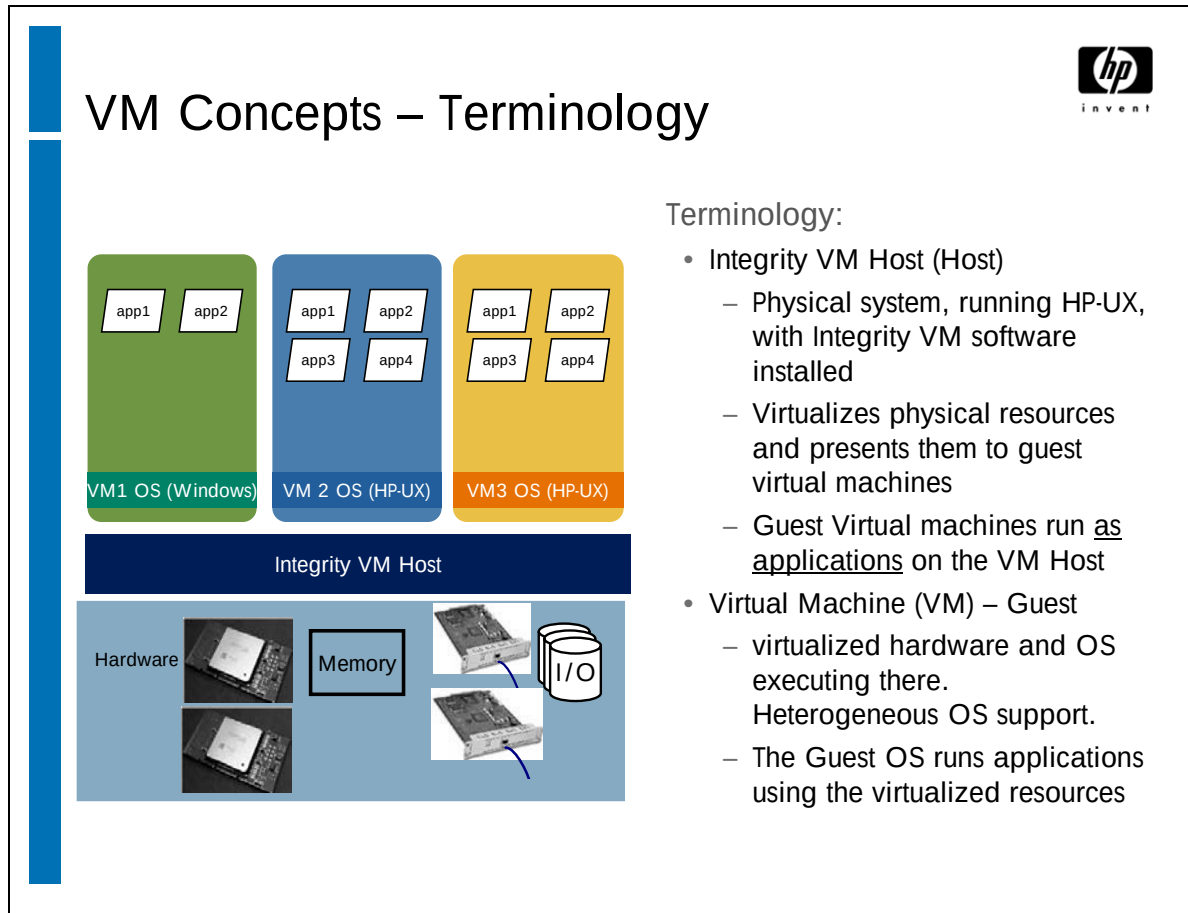
- Is a software partitioning and virtualization technology available on HP Integrity servers
- Allows multiple operating system instances to run on a single Integrity physical server or nPar

Student Notes

HP Integrity Virtual Machines (Integrity VM) is a product in HP's Virtual Server Environment (VSE) to help customers maximize server resource utilization and reduce total cost of ownership by allowing individual operating system environments to share CPU and I/O resources.

This module will show how the flexibility features of HP Integrity Virtual Machines and the high availability features of Serviceguard can be used together.

20-2. SLIDE: VM Concepts – Terminology



Student Notes

VM host

Physical system (or nPar), including the HP-UX OS, on which Integrity VM software is installed and enabled. It controls the physical hardware, handling resource requests from the VM guests. It is from the VM host that all virtual machines can be configured and managed. The VM host should be dedicated to running Virtual Machines. Each VM Guest appears to the VM host as an individual application.

Virtual machine (VM)

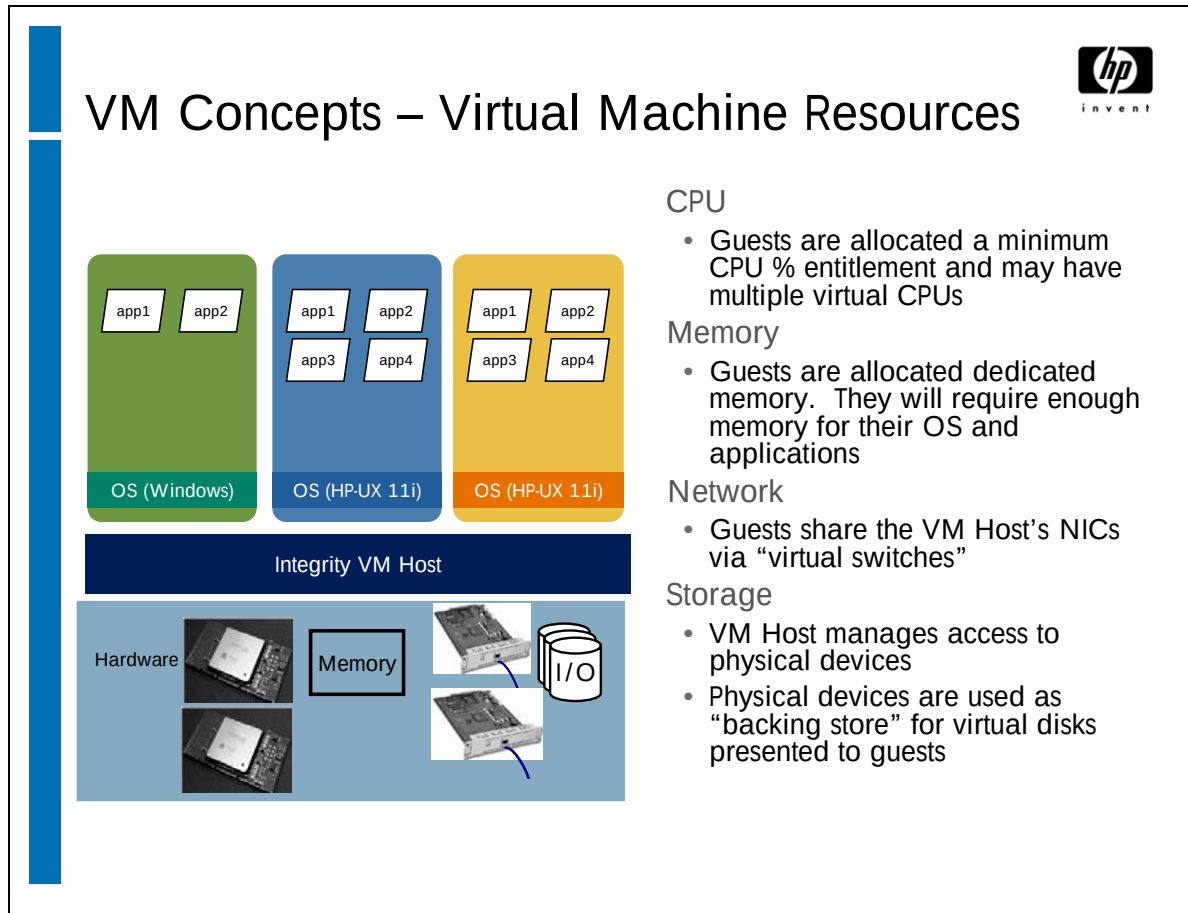
Each virtual machine is a complete system environment containing virtual implementations of CPU, memory, disk, I/O and other system resources capable of running an operating system called a guest OS. The guest OS can run any application supported by the OS and behave as it normally would on a single physical Integrity Server without the need for recompilation or other changes.

Module 20
Using HP Serviceguard with Integrity Virtual Machines

VM guest

This is the operating system booted and running in a virtual machine. Multiple operating system types are supported, including HP-UX and Windows. The VM Guest runs applications using the virtual resources allocated by the VM host.

20-3. SLIDE: VM Concepts – Virtual Machine Resources



Student Notes

CPU

When a guest is created, you can specify the number of virtual CPUs that guest can use, as well as the minimum amount of CPU processing power that is guaranteed to the guest for each virtual CPU. All VMs running on the same Host will share the physical CPUs available on the Host.

Memory

Memory is virtualized but not shared in the initial Integrity VM releases. The amount of memory assigned to each guest should take into consideration the OS and application requirements

Network

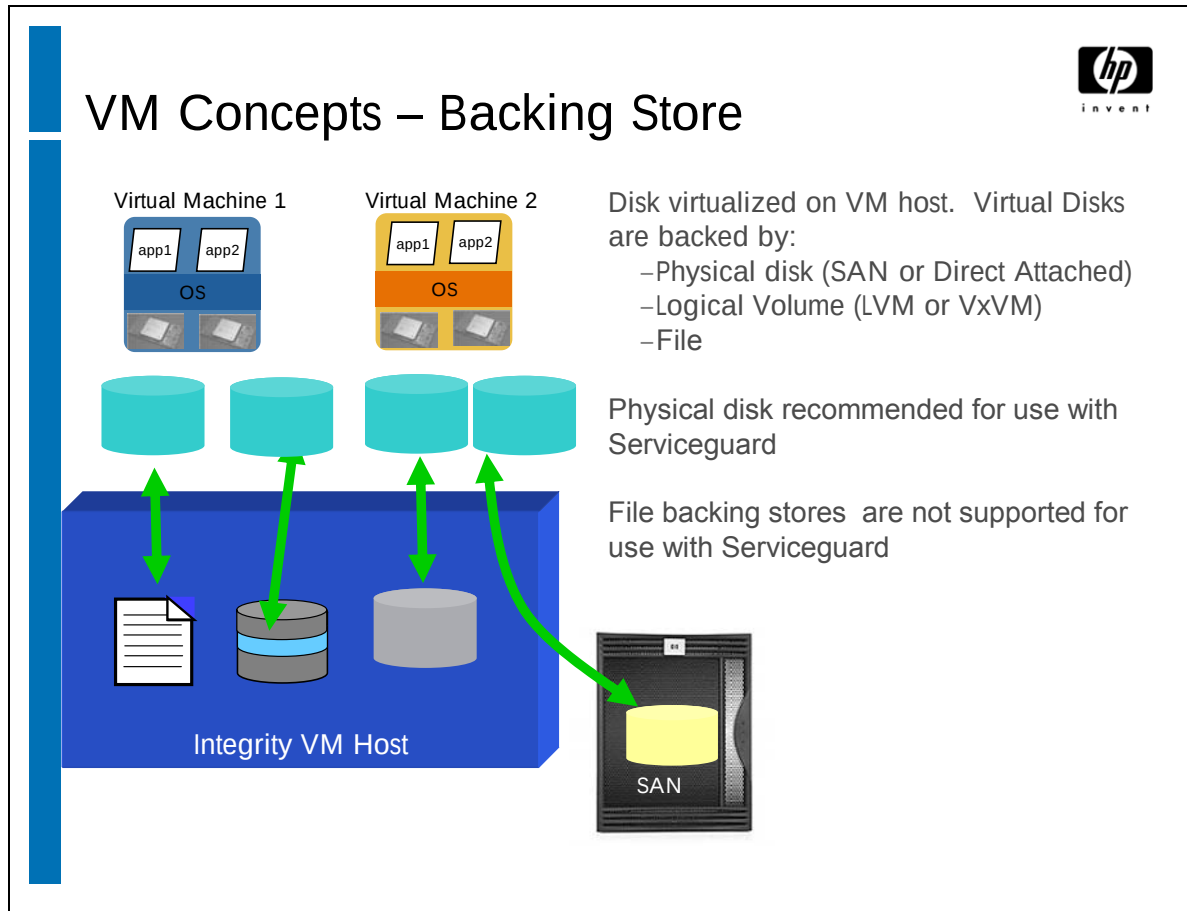
A virtual machine virtual network device is mapped to a vswitch. A vswitch can be mapped to a physical LAN card on the VM host, allowing the VM guest to communicate outside of the VM host.

Module 20
Using HP Serviceguard with Integrity Virtual Machines

Storage

Virtualized storage on a virtual machine maps to physical storage available on the VM Host, referred to as "backing store". The Host performs all physical I/O access for the VMs.

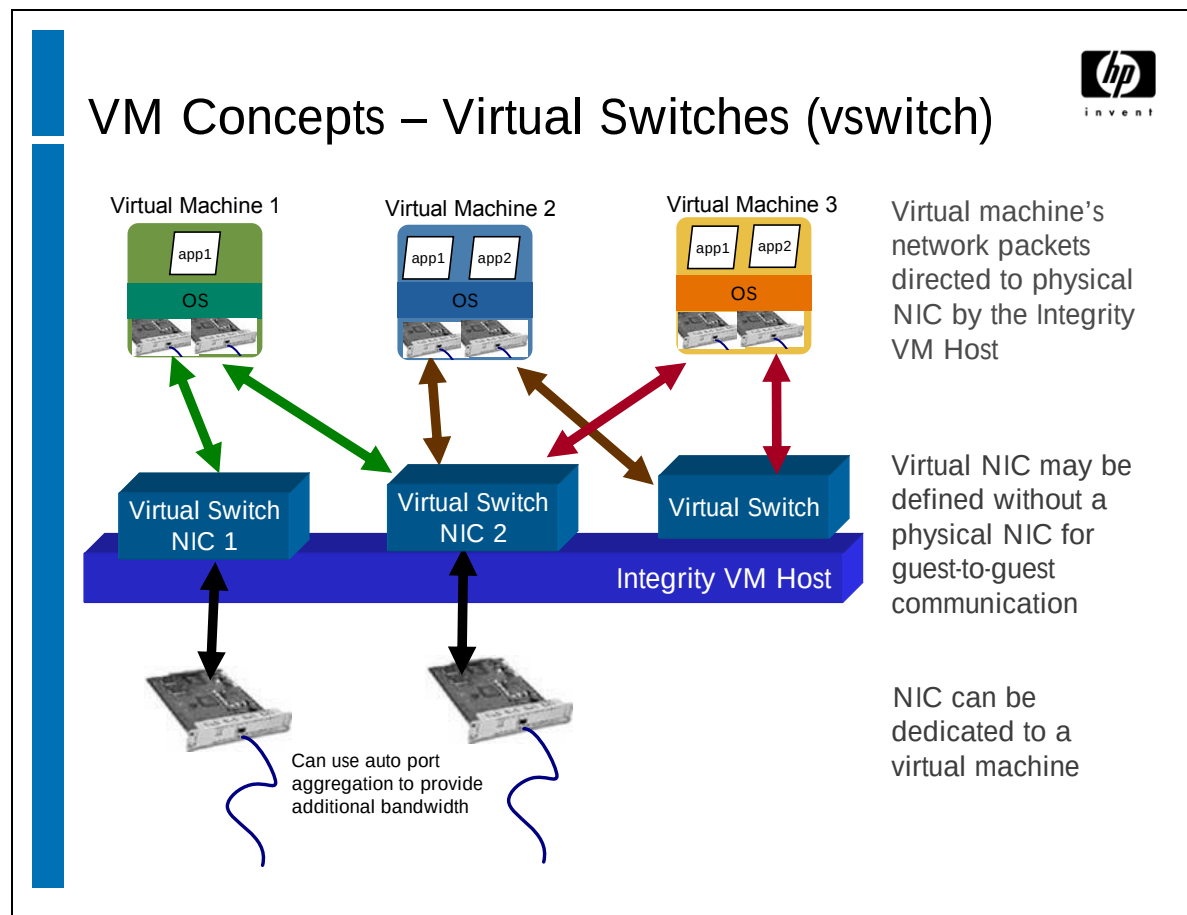
20-4. SLIDE: VM Concepts – Backing Store



Student Notes

Guest virtual storage devices are backed by physical devices on the VM Host. What appears to the Guest as a disk can be backed by either a file, a logical volume (VxVM or LVM) or a physical disk on the VM Host. HP recommends using physical disks as backing store for a VMs disk, both for best performance and ease of management. Files are not supported as backing store when used in conjunction with Serviceguard.

20-5. SLIDE: VM Concepts – Virtual Switches (vswitch)



Student Notes

For each physical network adapter on the VM Host, that you wish to make accessible to VM guest machines, you must create a separate vswitch. You may also create a vswitch with no associated physical network adaptor. If the virtual switch is associated with a physical network card on the VM Host, the guest virtual network devices will be able to communicate outside the VM Host. If no adapter is specified, the guest can only communicate within the VM Host environment.

A vswitch can only be associated with one interface on the VM Host. However, auto port aggregation can be used on the VM Host to provide additional bandwidth for a single interface (and therefore a single vswitch).

If you wish to build a highly available network configuration for a Virtual Machine, multiple virtual network devices, associated with separate virtual switches should be defined. This eliminates a NIC on the VM host being a single point of failure to the VM Guest.

20-6. SLIDE: Integrity VM Commands



Integrity VM Commands

hpvmnet	Creates and manages virtual switches
hpvmcreate	Creates a Virtual Machine (VM)
hpvmmodify	Add/remove resources to a VM
hpvmstart	Start a VM
hpvmstop	Shutdown a VM
hpvmstatus	View the status of a VM
hpvmdevmgmt	Manage the VM resource database

Student Notes

When configuring a highly available Integrity VM environment, you will use a combination of Serviceguard commands and HP Integrity VM commands. The slide introduces the HP Integrity VM commands.

20-7. SLIDE: Serviceguard and Integrity VM Use Models



Serviceguard and Integrity VM Use Models

VMs as Serviceguard Nodes

- The Guest Virtual Machine is a member of a Serviceguard cluster allowing failover of application packages between other physical or VM Guest nodes in the cluster
 - Supported with Integrity VM A.02.00 release and above

VMs as Serviceguard Packages

- Install Serviceguard on the VM Host machines
- The Guest Virtual Machine is encapsulated within a Serviceguard package allowing failover of the VM Guest between VM Host cluster nodes
 - Supported with Integrity VM A.01.20 release and above
 - Toolkit for VM A.01.20 downloadable from HP web
 - Toolkit for VM A.02.00 is integrated with Integrity VM software

Student Notes

An interesting duality exists with VMs in that they function both as “nodes” running an operating system while at the same being “application instances” running on a VM host. Since Serviceguard is designed to run on nodes and manage applications encapsulated within packages, integrating Serviceguard with Integrity VM can yield several different configuration possibilities.

After you set up Integrity VM, you can install HP Serviceguard A.11.16 or later either on the VM Host or on the HP-UX guest. Do not use Serviceguard on both the VM Host and the guest at the same time.

VMs as nodes

To protect guest applications, install Serviceguard on the HP-UX guest. Applications on a guest can fail over to any of the following:

- Another guest configured as a Serviceguard node that is running on the same VM Host system
- Another guest configured as a Serviceguard node running on a different VM Host system

- Another server or nPartition that is not running Integrity VM

Windows guests are not supported as HP Serviceguard nodes; therefore, Windows guest applications cannot be configured as Serviceguard packages.

VMs as packages

To protect guests, install HP Serviceguard on the VM Host system. Guests configured as Serviceguard packages (*distributed guests*) are subsequently managed using HP Serviceguard commands. If the VM Host system fails, the distributed guest automatically fails over to another node in the Integrity VM *multiserver environment*. Integrity VM guests which can be relocated between Integrity VM Hosts are configured into an Integrity VM multiserver environment that contains the same set of servers as is in the Serviceguard cluster. Guests of any operating system (HP-UX and Windows) can be configured as Serviceguard packages.

20–8. SLIDE: Availability Model for Integrity Virtual Machines as Serviceguard Nodes

Availability Model for Integrity Virtual Machines as Serviceguard Nodes



The application running in the VM is protected by Serviceguard

- A failed application can be:
 - Re-started in the same virtual machine
 - Failed over to another cluster member; whether physical host, vPar or VM
- An application can fail due to:
 - VM Host server hardware or software failure
 - VM Guest OS failure
 - Application-related problem

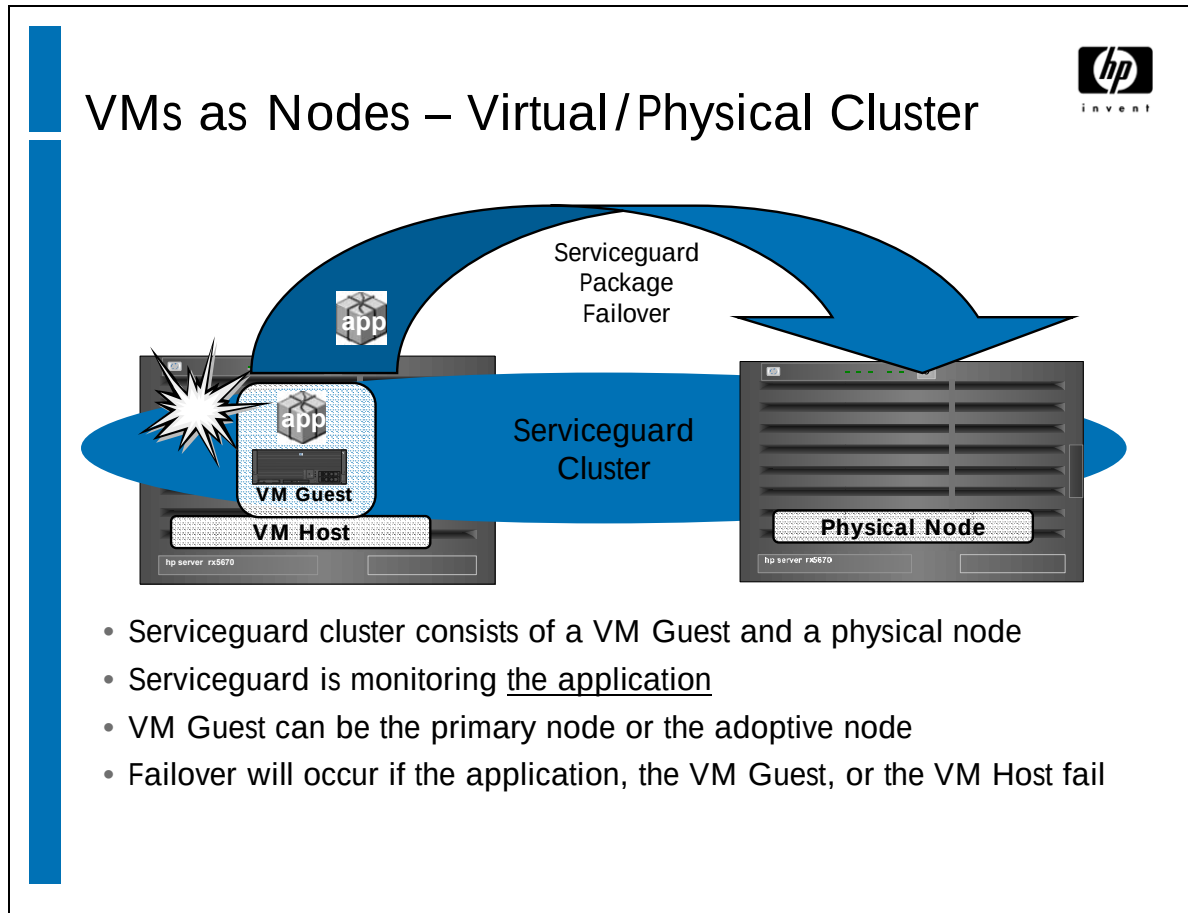
Student Notes

In the Virtual Machines as Serviceguard Nodes (or, VMs as Nodes) model that is supported in the Integrity VM A.02.00 release, VM guests are used as actual Serviceguard cluster nodes to provide the same HA failover capabilities found in traditional Serviceguard cluster configurations.

Important applications can be protected by Serviceguard and moved to another VM guest or physical node in the event of a failure to minimize application downtime. This functionality is identical to a traditional Serviceguard cluster using standalone physical systems or hard partitions.

Over the next several slides we'll examine possible configurations using VMs as Serviceguard nodes.

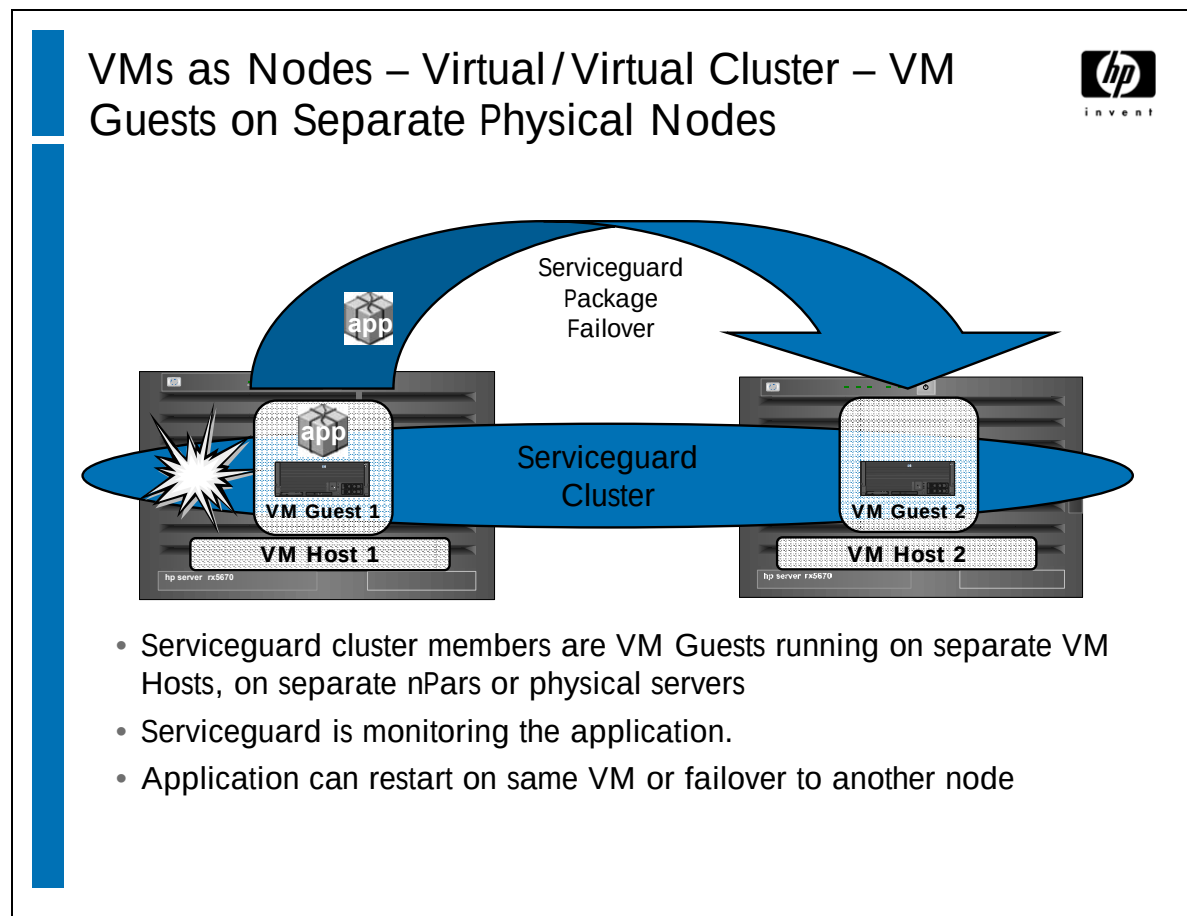
20-9. SLIDE: VMs as Nodes – Virtual/Physical Cluster



Student Notes

In this configuration, Serviceguard provides high availability in the event of a VM guest, VM host, or application failure. A failed application can either be restarted within the same VM guest or failed over to the physical node.

20-10. SLIDE: VMs as Nodes – Virtual/Virtual Cluster – VM Guests on Separate Physical Nodes



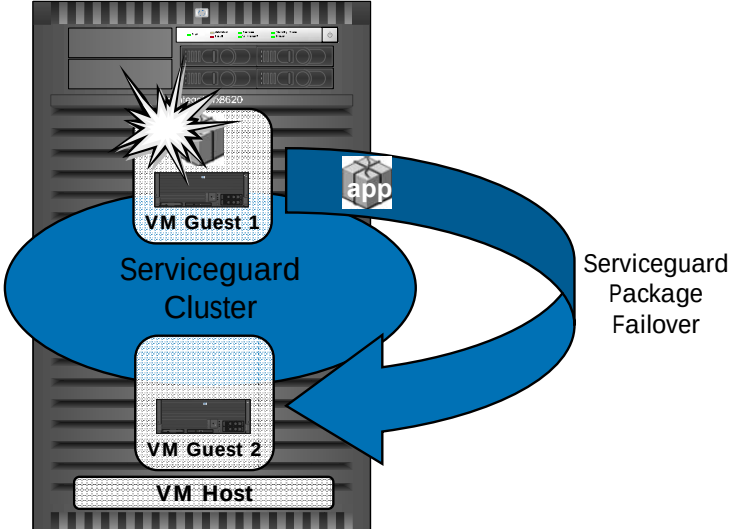
Student Notes

Serviceguard provides high availability in the event of a VM host, VM guest or application failure. However, in this configuration, a failed application can either be restarted within the same VM guest or failed over to another VM guest operating on a separate physical node.

20-11. SLIDE: VMs as Nodes – Virtual/Virtual "Cluster in a Box"



VMs as Nodes – Virtual/Virtual "Cluster in a Box"



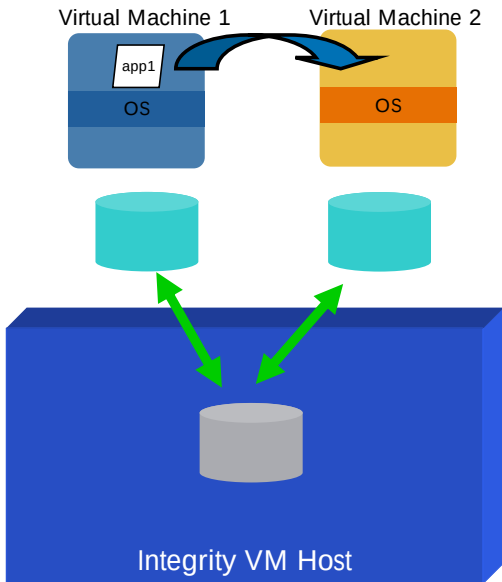
The diagram illustrates a Serviceguard Cluster within a VM Host. Two VM Guests, VM Guest 1 and VM Guest 2, are shown running on the VM Host. A blue oval labeled 'Serviceguard Cluster' encompasses both VM Guests. A blue arrow labeled 'app' points from VM Guest 1 to VM Guest 2. A large blue curved arrow labeled 'Serviceguard Package Failover' points from VM Guest 1 to VM Guest 2. A starburst icon is shown above VM Guest 1, indicating a failure or event.

- Serviceguard cluster members are VM Guests running in the same nPar or server
- Does not protect against a single point of failure

Student Notes

In this configuration, the primary node and the adoptive node are guests running on the same VM Host system. This cluster does not provide protection against Single Point of Failure (SPOF), because both the primary cluster member and the adoptive cluster member are guests on the same physical machine. However, this configuration is useful for Serviceguard cluster testing with minimal hardware.

20-12. SLIDE: Cluster in a Box – Sharing Backing Store



The diagram illustrates a 'Cluster in a Box' architecture. At the top, two virtual machines are shown: 'Virtual Machine 1' (blue) and 'Virtual Machine 2' (orange). VM 1 contains an 'app1' icon and an 'OS' layer. VM 2 contains an 'OS' layer. A curved blue arrow points from VM 1 to VM 2, indicating data sharing. Below each VM is a small cyan cylinder representing a disk. These two disks are connected by green double-headed arrows to a single gray cylinder disk located inside a large blue box labeled 'Integrity VM Host'. This shared disk serves as the backing store for both VMs.



Cluster in a Box – Sharing Backing Store

The VM Host must present the disk that will hold the application data to both virtual machines.

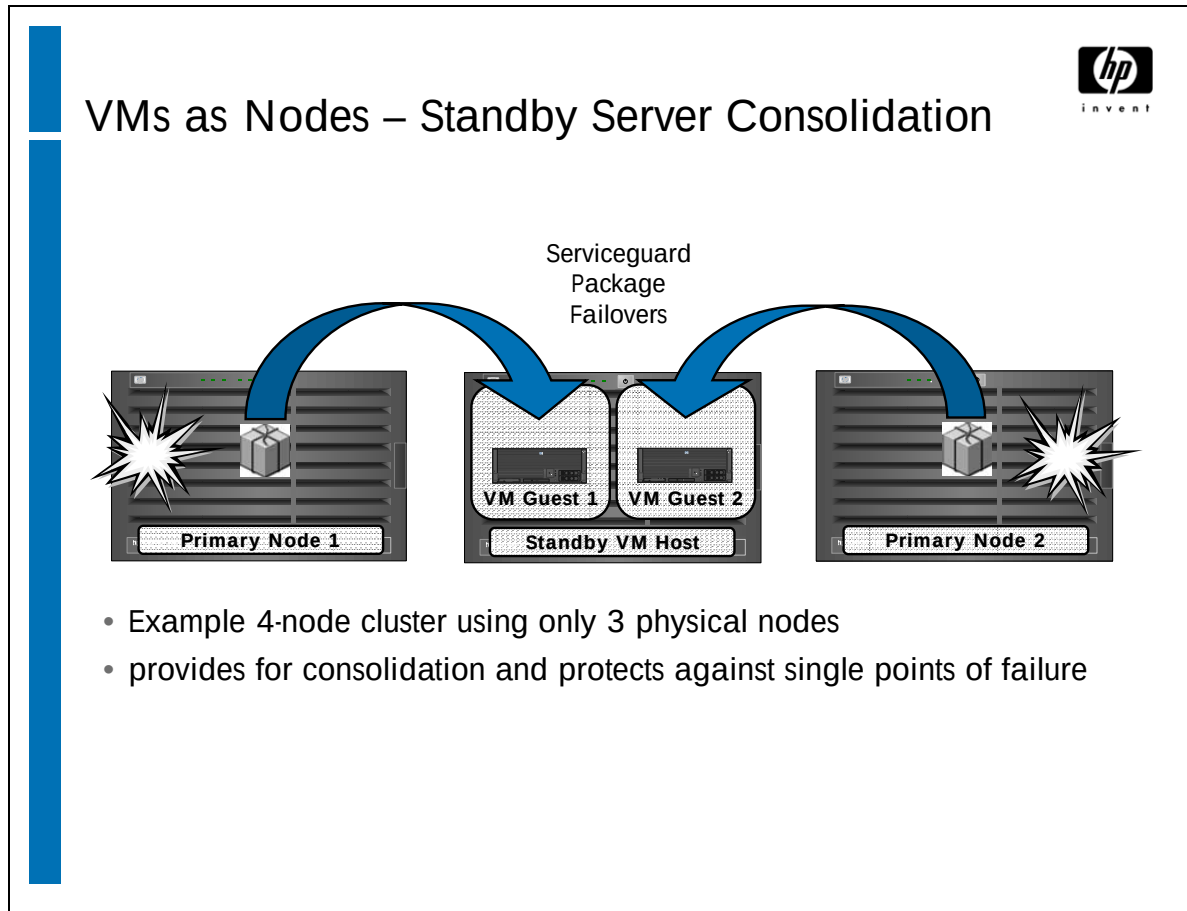
The disk used as the backing store must have the SHARED attribute set

```
hvvmdevmgt -m  
gdev:/dev/rdisk/c1t2d0  
attr:SHARE=YES
```

Student Notes

If you are running more than one guest on the VM Host system, and you need to share the same storage among the guests, you must change the SHARE attribute of the shared disk to YES using the `hvvmdevmgt` command,

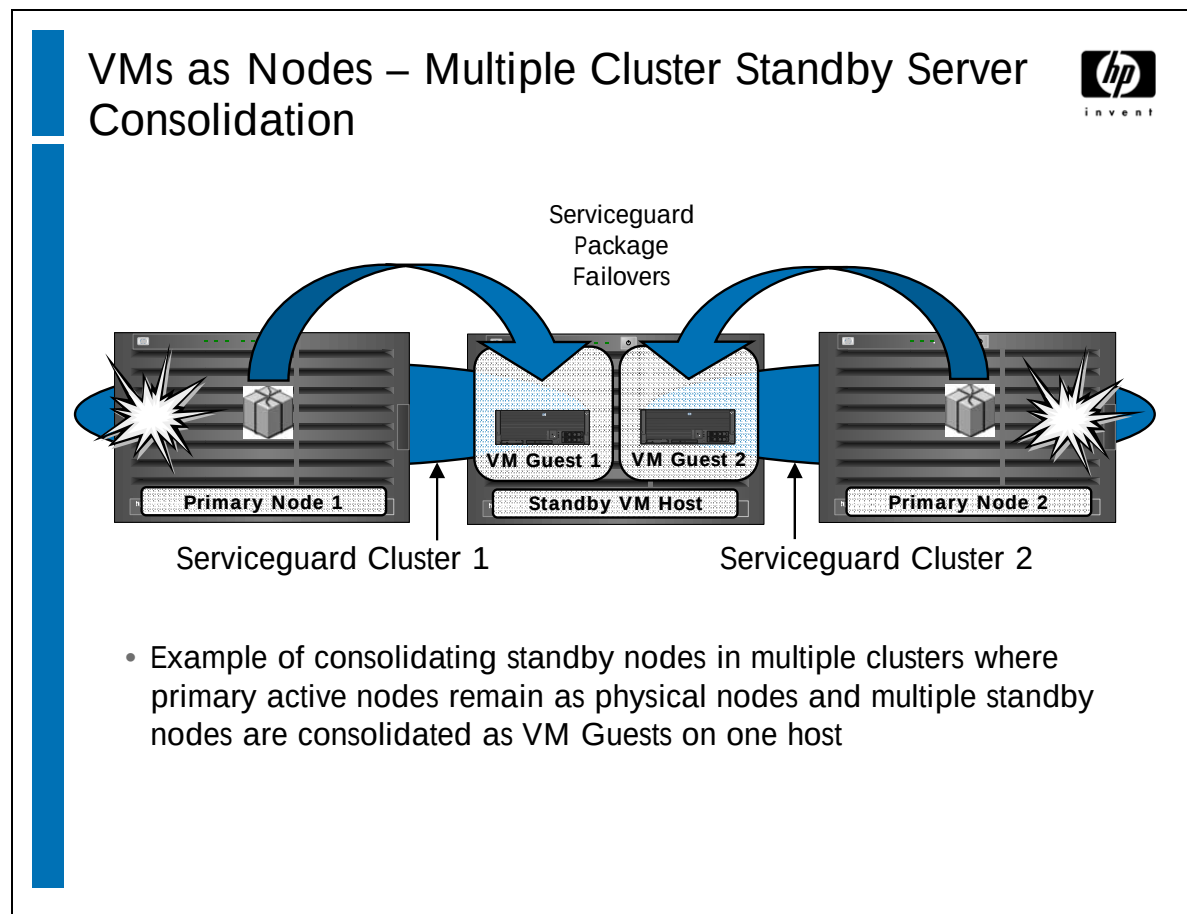
20-13. SLIDE: VMs as Nodes – Standby Server Consolidation



Student Notes

Integrity VM can be used to consolidate nodes within Serviceguard clusters. The packages that are normally running on Primary Nodes 1 and 2 can failover to their corresponding VM guest failover nodes running on a single standby host configured to support the execution of multiple VM guests. This is useful in environments where the applications may not run well together in a single stand-by OS environment.

20-14. SLIDE: VMs as Nodes – Multiple Cluster Standby Server Consolidation



Student Notes

This configuration has the benefit of being able to consolidate standby systems that are normally found in typical single Serviceguard cluster configurations, thus providing cost savings by reducing the number of required standby failover systems and making more efficient use of system hardware.

20-15. SLIDE: Configuring VM Guests as Serviceguard Nodes – Considerations

Configuring VM Guests as Serviceguard Nodes – Considerations



- Install and configure Serviceguard on VM Guest, as you would on any other node. Same minimum requirements.
- VM Considerations:
 - VM will need to have multiple virtual LAN devices, backed by multiple vswitches
 - Backing Store for the package/application data must be shared by other nodes in the cluster.
 - The VM Host must use physical disks as the backing store for the VM Guests application data.
 - The VM Guest can then use a volume manager to manage the virtual disk
 - The VM Guest OS disk need not be shared
 - VM Guest is sharing CPU with other VMs. `cmcl` can only run when the guest is running, therefore you may need to adjust `NODE_TIMEOUT`

Student Notes

Keep the following requirements in mind:

Network Requirements

To make sure network communication with guests is always available; provide identical network devices on both the primary and alternate nodes. Physical NICs (pNICs) and vswitches must be the same on both the original and adoptive nodes for virtual NICs (vNICs) to function after the failover. Just as multiple NICs are required on stand-alone Serviceguard nodes, multiple virtual network devices are required on VMs as Serviceguard nodes.

Storage Requirements:

To make sure the Serviceguard configuration is manageable, use identical backing stores on both the primary node and alternate nodes. To use Serviceguard in Guest configurations, the backing storage units must be whole disks. Integrity VM does not support using other types of backing stores on primary and alternate nodes for applications that are configured as Serviceguard packages. The VM Host system storage configurations must comply with both Integrity VM and Serviceguard product requirements.

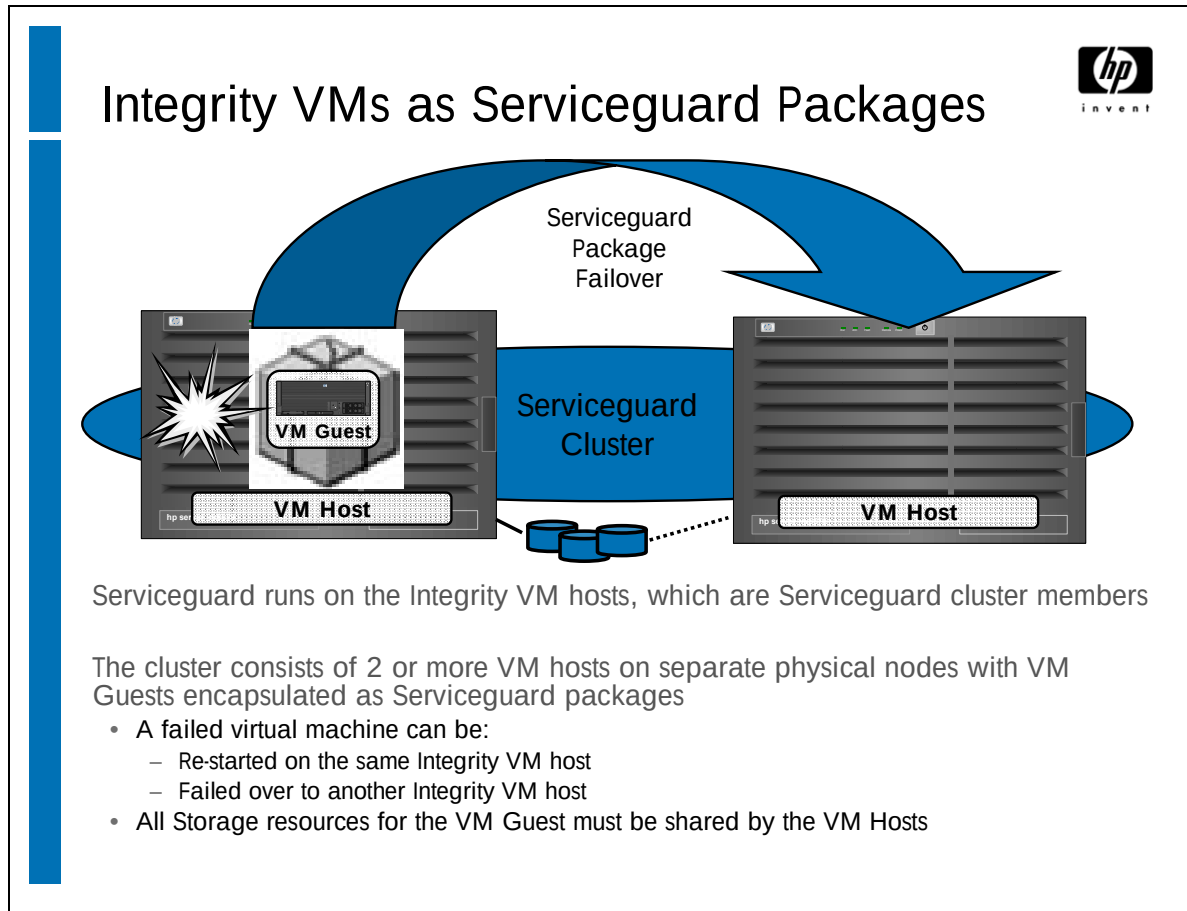
cmclld

In VMs as Serviceguard node configurations, there are some situations where VM guests defined with 2 or more vCPUs under heavy processing load conditions can potentially experience cmclld run time delays. If the run time delay is longer than the configured cluster NODE_TIMEOUT value (i.e., the time after which a node may decide that the other cluster node has become unavailable), cmclld will trigger a cluster reformation just as if a node had failed. However, since no node has actually failed, the cluster will reform with the same number of nodes it originally had before the cmclld run delay was reported.

Other factors that may contribute to this situation include vCPU processing entitlement percentages, the number of physical CPUs within the host and the VM host's physical processor speed.

Cmclld run delays can be identified by the following warning reported in the syslog file:
[date/time VM name] cmclld [PID]: Warning: cmclld process was unable to run for the last <x.yz> seconds

20-16. SLIDE: Integrity VMs as Serviceguard Packages



Student Notes

To protect guests, install HP Serviceguard on the VM Host system. Guests configured as Serviceguard packages (*distributed guests*) are subsequently managed using HP Serviceguard commands. In this configuration, if any of the resources used by a guest fail on the primary VM Host system, the guest fails over to an adoptive VM Host system in the Integrity VM *multiserver environment*. Depending on the type of failure, the Serviceguard package containing the VM guest may also be restarted on the same node.

When configuring guests as Serviceguard packages, the backing store for the VM Guests operating system disk as well as the Guest's application disks must be accessible to all the VM Hosts that can run the Guest Serviceguard package. The backing store can be either physical device or logical volume, but can not be files.

20-17. SLIDE: VMs as Serviceguard Packages Use Case Examples

VMs as Serviceguard Packages Use Case Examples



HA for Development and Test Environments:

- Traditional development and test environments typically are not highly available; using Integrity VM with Serviceguard provides HA for the VMs created to consolidate the applications used in these environments, which can be failed over to another node in the event of a VM Host failure

Maintenance and Load Balancing:

- VM Guests running applications used for development or production activities can easily be moved as Serviceguard packages to other nodes within a cluster when planned system maintenance is required or to balance system loads during peak operating times

HA for Non-Clustered Production Applications:

- Any production applications that have been consolidated into VMs and are not designed to run in a clustered environment will gain a level of improved availability without the need for any modification by Serviceguard restarting the VM on another VM Host in the event of a VM Host or VM Guest failure

Student Notes

20-18. SLIDE: VMs as Packages Usage Considerations



VMs as Packages Usage Considerations

- Integrity VM host nodes should only run VM guests and not other user applications
- VM Guests within Serviceguard packages can only failover to other Serviceguard cluster nodes that are running Integrity VM Host software
- Serviceguard can only monitor the status of the VM Guest running as a Serviceguard package not the applications running within the VM Guest
- VM Guest failover will take longer compared to failover of a clustered application

Student Notes

When implementing VMs as Serviceguard Packages, several key points should be considered:

- Integrity VM host nodes should only run VM guests and not other user applications. The Integrity VM Fair Share Scheduler provides control of entitlements (i.e., the percentage or number of CPU clock cycles that are guaranteed to a VM) for VMs running on the VM host. The scheduler cannot monitor or respond to VM processing requirements with other workloads running on the VM host; therefore it is recommended to only run Integrity VM host software and VM guests on the Serviceguard cluster nodes. This rule also applies to non-Serviceguard Integrity VM configurations.
- VM guests within Serviceguard packages can only failover to other Serviceguard cluster nodes that are running Integrity VM host software. VM guests can only run on pre-configured VM hosts. For a successful Serviceguard VM guest package failover, the failover node must be a VM host. Having prerequisite software installed on a failover node to support the execution of any application within a Serviceguard package is a basic requirement for Serviceguard cluster configurations.

Module 20

Using HP Serviceguard with Integrity Virtual Machines

- Serviceguard can only monitor the status of the VM guest running as a Serviceguard package not the applications running within the VM. With a VM guest running as a Serviceguard package, Serviceguard only monitors the status of the VM guest executing as an HP-UX process on the VM host and not the applications running within the guest. In this case, any VM guest package failovers would be based on a failure of either the VM guest or its VM host node, and not due to any application specific failures.
- Failover of VM guest applications is slower than clustered applications. Applications designed for clustered environments and running on a host node can be operational more quickly after a failover compared to an application running in a VM guest because they do not have the added time required to start (i.e., boot-up) the VM before the application can be restarted. In this case, the VM guest and its applications are being failed over together, so the total time required for an application to be available for use consists of the VM start time, guest OS boot-up time and the application start-up time. However, the VM start time and guest OS boot-up time for a VM guest is significantly faster than the boot-up time of a physical node.

20-19. SLIDE: Configuring VM Guests as Packages



Configuring VM Guests as Packages

1. Create the Serviceguard cluster
2. Configure an Integrity VM multi-server environment
3. Configure the VM Guest on all nodes in the cluster
 - VM Guests backing storage must be shared by all nodes
 - Virtual switch configurations must be the same on all nodes
4. Start the VM Guest on primary node using `hpvmstart`
5. Use the toolkit to package the VM Guest
6. Shutdown the VM Guest
7. Start the VM Guest by starting the Serviceguard package, test failover

Student Notes

To configure VM Guests as Serviceguard packages the VM hosts must be configured as nodes in a Serviceguard cluster. The cluster node configuration is done the same as you would any other Serviceguard cluster.

HP provides a HP Serviceguard for Integrity Virtual Machines Toolkit that includes package configuration files and utility scripts to help with the creation of VM guest Packages and monitoring of the VM guests within the packages. The toolkit is available for Integrity VM versions A.01.20 and A.02.xx using with Serviceguard versions A.11.16 and A.11.17. The A.01.20 version of the toolkit is available for download at <http://software.hp.com> under the "HP-UX 11i general" link, and the A.02.xx version is provided with the A.02.xx HPVM product.

Over the next several slides we'll view the steps in more detail. We will walk through an example of creating a two node Serviceguard cluster with an HP Integrity virtual machine package.

Module 20

Using HP Serviceguard with Integrity Virtual Machines

Example:

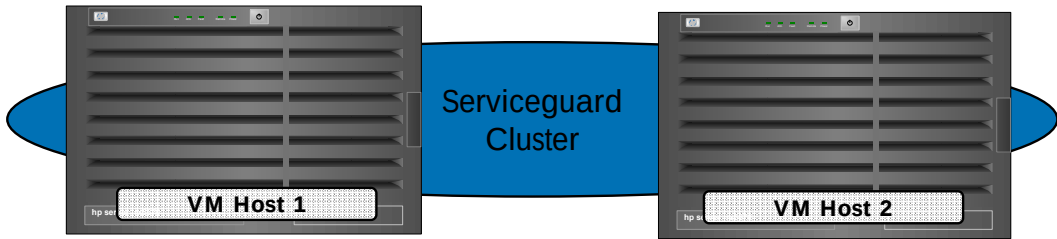
First, configure the Serviceguard cluster as you would any other cluster. In this example, Integrity Virtual Machine software and Serviceguard have been installed on two hosts; rx26u202 and rc26u203. A quorum server has been configured on rx26u205.

```
#cd /etc/cmcluster  
#cmquerycl -v -q rx26u205 -C cmclconfig.ascii -n rx26u202 -n rx26u203
```

Edit the cluster configuration file as necessary

```
#cmcheckconf -C cmclconfig.ascii  
#cmapplyconf -C cmclconfig.ascii  
#cmruncl
```

20-20. SLIDE: Configuring an Integrity Multi-Server Environment



The diagram shows two server racks representing VM Host 1 and VM Host 2. A blue box labeled 'Serviceguard Cluster' is positioned between them, with lines connecting it to both hosts. The HP logo is in the top right corner.

```
hpvmdevmgt -a server:host2
```

```
hpvmdevmgt -a server:host1
```

On both nodes:

```
hpvmdevmgt -m server:host1:attr:SERVERID=1
```

```
hpvmdevmgt -m server:host2:attr:SERVERID=2
```

```
hpvmdevmgt -m server:host1:attr:SERVERADDR=1.2.3.4
```

```
hpvmdevmgt -m server:host2:attr:SERVERADDR=1.2.3.5
```

This allows the VM Hosts to see the status of each others guests

Student Notes

The Integrity VM multiserver environment provides an integrated environment when guests are configured as Serviceguard packages. In a cluster that is configured as a multiserver environment, each VM Host is aware that the guests are Serviceguard packages and gives control of these distributed guests to ServiceGuard.

For example, two VM Hosts (host1 and host2) make up a Serviceguard cluster. A guest running on host1 is configured as a Serviceguard package that can fail over to host2. After you set up the multiserver environment, you can use the `hpvmstatus` command on each VM Host to display consistent guest package status information. When the guest package is running on host1, you can use the `hpvmstatus` command on host2 to display the guest's current status.

In order to run Integrity VMs as Serviceguard packages you must update the HP Integrity VM device management database to add records for each server in the multi-server environment. Each server must be assigned a unique server-id (1-255). You must also supply the IP address used by Serviceguard to monitor the VM host node. The HP Integrity VM device management database is maintained using the `hpvmdevmgt` command.

Module 20

Using HP Serviceguard with Integrity Virtual Machines

Example:

The following example shows the commands used to set up a multi-server environment on two hosts (rx26u202 and rx26u203):

On server rx26u202:

```
# hpvmdevmgt -a server:rx26u203
```

On server rx26u203:

```
# hpvmdevmgt -a server:rx26u202
```

On both systems:

```
# hpvmdevmgt -m server:rx26u202:attr:SERVERID=1
```

```
# hpvmdevmgt -m server:rx26u203:attr:SERVERID=2
```

```
# hpvmdevmgt -m server:rx26u202:attr:SERVERADDR=10.10.48.102
```

```
# hpvmdevmgt -m server:rx26u203:attr:SERVERADDR=10.10.48.103
```

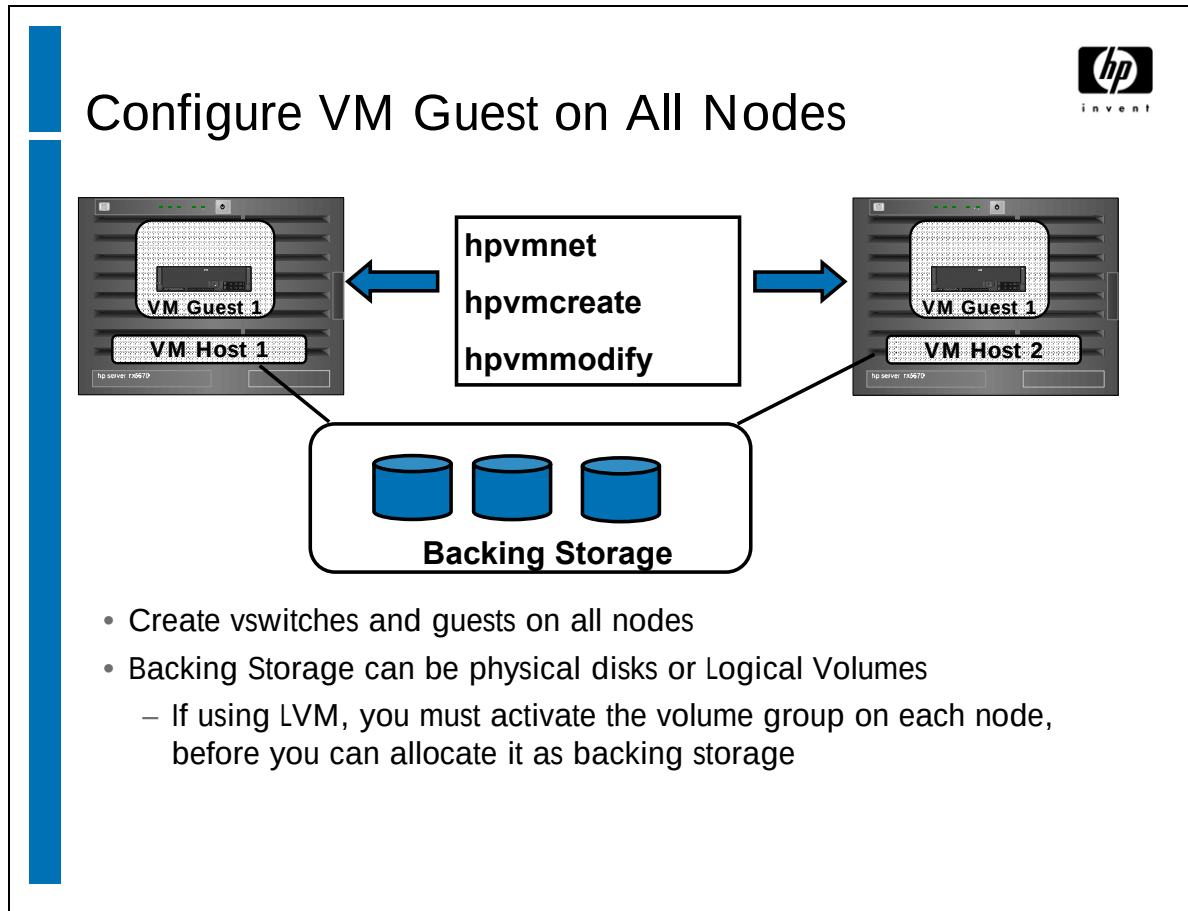
Verify the results:

```
# hpvmdevmgt -l server
```

```
rx26u202:CONFIG=SERVER,SERVERID=1,SERVERADDR=10.10.48.102::WWID_NULL
```

```
rx26u203:CONFIG=server,EXIST=NO,DEVTYPE=UNKNOWN,SHARE=NO,SERVERID=2,  
SERVERADDR=10.10.48.103::WWID_NULL
```


20-21. SLIDE: Configure VM Guest on All Nodes



Student Notes

Next, you must create the same VM guest definitions on each of the nodes. Make sure to configure the virtual switches (vswitches) identically. Also, to make it easier to manage the packages assign the same "virtual" device numbers for the backing stores on each VM host.

Example:

In this example we are configuring a VM guest with one virtual switch and one disk as backing store. Use `hpvmmodify` to add the guest's application disks as backing store.

Create and boot a virtual switch named `switch1`, backed by `lan1`. These commands would be run on both nodes:

```
#hpvmnet -c -S switch1 -n 1  
#hpvmnet -b -S switch1
```

Module 20

Using HP Serviceguard with Integrity Virtual Machines

Create a virtual machine named vm1 on both VM host Serviceguard nodes. In this example the VM is being assigned a 20% minimum CPU entitlement and will be running HP-UX as the guest operating system. The guest will have two virtual devices; a virtual disk (backed by a physical device) and a virtual lan card (backed by the vswitch). Additional virtual devices can be added using the `hpvmmodify` command:

```
#hpvmcreate -P vm1 -e 20 -O HPUX
#hpvmmodify -P vm1 -a network:lan:1,0:vswitch:switch1
#hpvmmodify -P vm1 -a disk:scsi:1,2,3:disk:/dev/rdisk/c10t1d1
```

If you prefer to use LVM as backing store, you would need to create the Volume Group on one node and import it on the second node before creating the virtual machines. The following example shows the `hpvmmodify` command using an LVM volume as backing store. Note that the volume group must be active (either in r/w or ro mode) in order for this command to succeed:

```
#hpvmmodify -P vm1 -a disk:scsi:1,2,3:lv:/dev/vg01/rlvol1
```

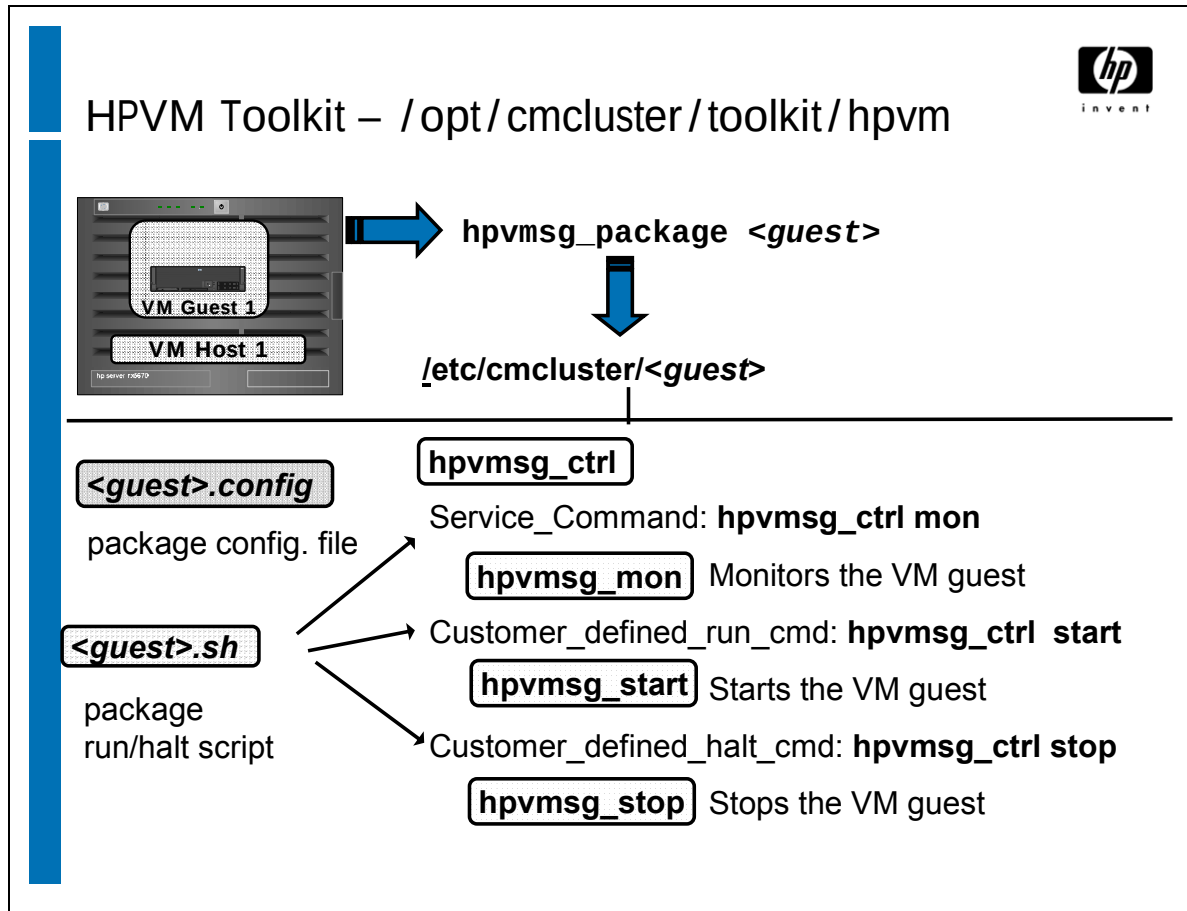
In order to keep hardware paths for the virtual disks consistent on the VM Host nodes, be sure to specify the optional address portion on the `hpvmcreate`/`hpcmmmodify` commands. For example:

```
#hpvmmodify -P vm1 -a disk:scsi:1,2,3:disk:/dev/rdisk/c10t1d1
```

Results in a virtual disk address of:

0/1/2/0.3.0

20-22. SLIDE: HPVM toolkit – /opt/cmcluster/toolkit/hpvm



Student Notes

The hpvmmsg_package utility is used to configure a guest as a Serviceguard package. The utility uses the guest name that you supply as an argument to create and populate the /etc/cmcluster/<guest-name>/ directory with a set of template files that contain basic Serviceguard parameter settings. HP recommends that you review and modify these template files as needed for your specific multiserver environment.

This example shows the use of the HPVM toolkit provided as part of the Integrity VM A.02.xx product. The toolkit for earlier versions is quite different. Refer to the readme files downloaded with the toolkit for more detailed instructions.

Before using the hpvmmsg_package utility to create the package files, you must first manually start the Virtual Machine on the primary node using the hpvmstart command. To build the files used by Serviceguard for the package, run the command:

```
/opt/cmcluster/toolkit/hpvm/hpvmmsg_package <guest>
```

Module 20

Using HP Serviceguard with Integrity Virtual Machines

The output of the toolkit will be the files used to create the Serviceguard package. The `hpvmmsg_package` command will offer to copy the appropriate files to all nodes in the cluster.

<guest>.config	This is the package configuration file
<guest>.sh	This is the package run/halt script
hpvmmsg_ctrl	This is script is used as the SERVICE_CMD, the start and the stop command for the package.
hpvmmsg_start	Called by hpvmmsg_ctrl to issue the hpvmstart command (starts the guest)
hpvmmsg_stop	Called by hpvmmsg_ctrl to issue the hpvmstop command (stops the guest)
hpvmmsg_mon	Called by hpvmmsg_ctrl to issue the hpvmstatus command (monitors the guest)

Example:

Start the VM guest:

```
# hpvmstart -P vm1
```

Run the packaging tool supplied with the toolkit:

```
# /opt/cmcluster/toolkit/hpvm/hpvmmsg_package vm1
```

This is the HP Virtual Machine Serviceguard Toolkit Package Template Creation script.

This script will assist the user develop and distribute a set of Serviceguard package configuration template files and associated start, stop and monitor scripts.

The templates generated by these scripts will handle many guest configurations, but it is only a template and may not be appropriate for your particular configuration needs. You are encouraged to review and modify these template files as needed for your particular environment.

Do you wish to continue? (y/n):y

[Virtual Machine Details]

Virtual Machine Name	VM #	OS	Type	State
----------------------	------	----	------	-------

vm1	1	HPUX	On	(OS)
-----	---	------	----	------

[Storage Interface Details]

Guest		Physical					
Device	Adaptor	Bus	Dev	Ftn	Tgt	Lun	Storage Device
disk	scsi	1	2	0	3	0	disk /dev/rdisk/c10t1d1

[Network Interface Details]

Interface	Adaptor	Name/Num	PortNum	Bus	Dev	Ftn	Mac Address
vswitch	lan	switch1	1		1	0	0 a2-10-1a-02-77-06

The package template files successfully copied to each cluster member

Would you like to configure this Virtual Machine as a Serviceguard distributed guest on each cluster member? (y/n):**y**

Would you like to create a failover package for the Virtual Machine Summarized above? (y/n):
y

The failover package template files for the Virtual Machine were successfully created.

Would you like to copy the package template files to each cluster member? (y/n):**y**

The Virtual Machine : **vm1**

Has been successfully configured as a Serviceguard distributed guest on each cluster member

Please see the HP Integrity Virtual Machines documentation for additional instruction on configuring Virtual Machines as Serviceguard packages.

In our example the following files were created by running `hpvmmsg_package vm1`:

```
# ls /etc/cmcluster/vm1
hpvmmsg_ctrl*  hpvmmsg_start*  vm1.config      vm1.sh.log
hpvmmsg_mon*  hpvmmsg_stop*   vm1.sh*
```

vm1.config This is the package configuration file. Here are some excerpts from this file:

```
PACKAGE_NAME vm1
PACKAGE_TYPE      FAILOVER
FAILOVER_POLICY   CONFIGURED_NODE
FAILBACK_POLICY   MANUAL
NODE_NAME rx26u202
NODE_NAME rx26u203
AUTO_RUN          YES
RUN_SCRIPT         /etc/cmcluster/vm1/vm1.sh
RUN_SCRIPT_TIMEOUT NO_TIMEOUT
HALT_SCRIPT        /etc/cmcluster/vm1/vm1.sh
HALT_SCRIPT_TIMEOUT NO_TIMEOUT
SERVICE_NAME      vm1
SERVICE_FAIL_FAST_ENABLED NO
SERVICE_HALT_TIMEOUT 300
SUBNET 10.10.0.0
```

Module 20

Using HP Serviceguard with Integrity Virtual Machines

vm1.sh This is the package run/halt script. Here are some excerpts from this file:

```
SUBNET[0]=10.10.0.0
SERVICE_NAME[0]="vm1"
SERVICE_CMD[0]="/etc/cmcluster/vm1/hpvmmsg_ctrl monitor vm1"
SERVICE_RESTART[0]=""
function customer_defined_run_cmds
{
    /etc/cmcluster/vm1/hpvmmsg_ctrl start vm1
    test_return 51
}
function customer_defined_halt_cmds
{
    /etc/cmcluster/vm1/hpvmmsg_ctrl stop vm1
    test_return 52
}
```

Note: if LVM is used as a backing store this file will also contain:

```
VGCHANGE="vgchange -a e"
VG[0]=/dev/vg01
```

hpvmmsg_ctrl This script is called by the run/halt script to start/stop/monitor the package. Here are some excerpts from this script:

```
case $1 in
    start)
        . /etc/cmcluster/vm1/hpvmmsg_start $2
        ;;

    stop)
        . /etc/cmcluster/vm1/hpvmmsg_stop $2
        ;;

    monitor)
        . /etc/cmcluster/vm1/hpvmmsg_mon $2
        ;;
esac
exit $?
```

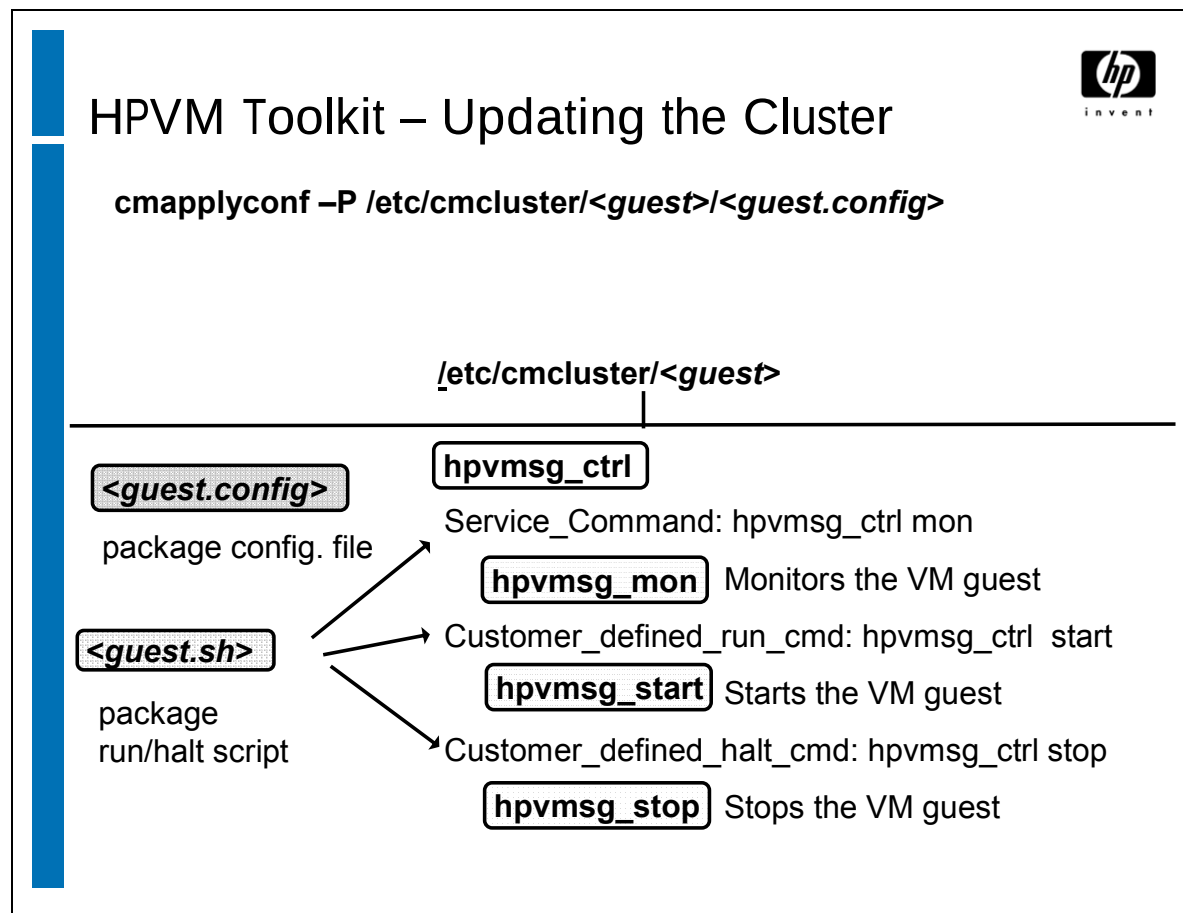
hpvmmsg_start This script is used to start the VM guest. Here are some excerpts from this script:

```
if /opt/hpvm/bin/hpvmstatus -P ${1} 2>/dev/null | grep -q "$1 .* On .*" ;
then
    echo "Virtual machine already started on this node"
else
    /opt/hpvm/bin/hpvmstart -P $1
    status=$?
    if (($status != 0)) ; then
        exit ${EXIT}
    fi
    exit $status
fi
```

hpvmmsg_stop This script is used to stop the VM guest. Here is an excerpt from this script:
/opt/hpvm/bin/hpvmstop -F -g -P \$1

hpvmmsg_mon This script is used to monitor the health of the VM guest. The monitor uses hpvmstatus to determine if the guest vm is running. There are some parameters in this script that you can modify in this script:
The number of times we will try to verify that the guest has rebooted or started before we failover reset on success
NUM_OF_RETRY=10
Number of seconds to wait between retries
MONITOR_INTERVAL=1
The total number of guest failures/reboot/panics before failing over.
NUM_OF_FAILURES=10

20-23. SLIDE: HPVM Toolkit – Updating the Cluster



Student Notes

Once the `hpvmmsg_package` command has been run and you have made any desired changes to the files created, you must update the cluster to add the new package. If you are using LVM as backing store make sure to add the volume group to the cluster configuration file.

```
#cd /etc/cmcluster
#cmcheckconf -v -C cmclconfig.ascii -P vm1/vm1.config
#cmapplyconf -v -C cmclconfig.ascii -P vm1/vm1.config
```


20-24. SLIDE: Starting a Serviceguard Packaged VM



Starting a Serviceguard Packaged VM

- Use Serviceguard commands

```
cmrunpkg <guest>
cmmodpkg -e <guest>
```
- Do not use hpvmstart

```
# hpvmstart -P vm2
hpvmstart: ERROR (vm2): Only Serviceguard can start
Guest vm2.
hpvmstart: Unable to start guest 'vm2'.
hpvmstart: Unable to continue.
```
- In a multi-server environment, hpvmstatus checks all servers for status of guest
- Use cmviewcl to see where guest is running

Student Notes

The Integrity VM multiserver environment provides an integrated environment when guests are configured as Serviceguard packages. In a cluster that is configured as a multiserver environment, each VM Host is aware that the guests are Serviceguard packages and gives control of these distributed guests to ServiceGuard.

After you configure a guest as a Serviceguard package, you cannot use Integrity VM commands to start and stop the guest. Only Serviceguard commands control the packaged guest. This provides consistent control of the guests and also protects shared whole disk backing stores. Use the Serviceguard commands; `cmrunpkg` and `cmmodpkg` to start VM Guest packages.

After you set up the multiserver environment, you can use the `hpvmstatus` command on each VM Host to display consistent guest package status information. When the guest package is running on host1, you can use the `hpvmstatus` command on host2 to display the guest's current status

20-25. SLIDE: Shutting down a Serviceguard Packaged VM

Shutting Down a Serviceguard Packaged VM



- Ordinary shutdown from Guest VM will look like a package failure to Serviceguard. If you plan to shutdown the guest for maintenance, disable AUTORUN first

```
cmmodpkg -d <guest>
```

```
#shutdown -h
```

- cmhaltpkg <guest> will issue → hpvmstop -P <guest>
- Do not manually issue hpvmstop. Can override with the -F option.

```
# hpvmstop -P vm2
```

```
hpvmstop: Stop the virtual machine 'vm2'? [n]: y
```

```
hpvmstop: ERROR (vm2): Only Serviceguard can stop  
Guest vm2.
```

```
hpvmstop: Unable to continue.
```

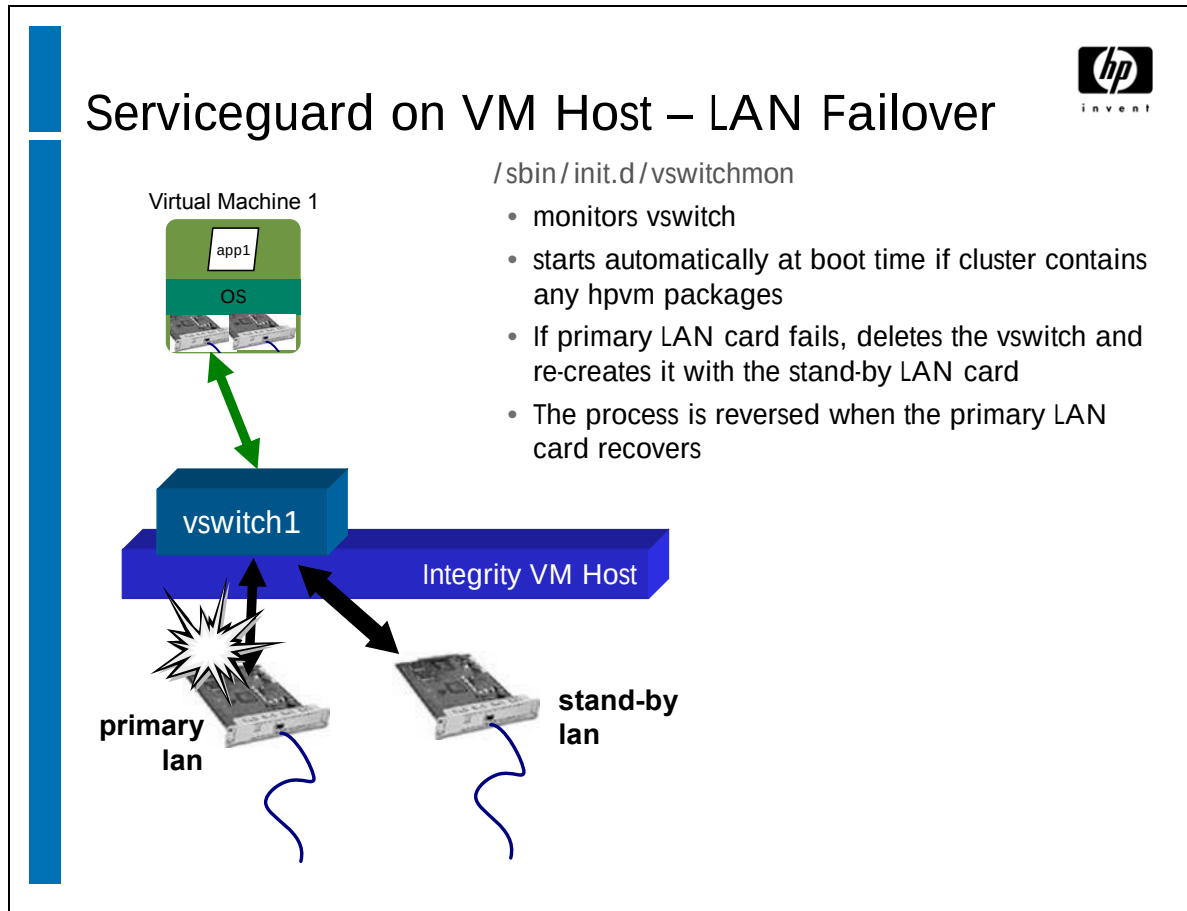
```
# hpvmstop -P vm2 -F
```

Student Notes

You can force a HP Virtual Machine to failover by issuing the shutdown command from the guest OS. When the monitor issues the hpvmstatus and detects that the guest is not running it will attempt a failover.

You can also shutdown a distributed guest by issuing the cmhaltpkg command. This results in an hpvmstop command being issued for the guest.

20-26. SLIDE: Serviceguard on VM Host - LAN failover



Student Notes

The vswitch monitor is responsible for monitoring the activities of the Serviceguard network monitor and for moving the vswitch configuration, when appropriate, between primary and standby network interfaces.

The vswitch monitor requires no user configuration and is installed as part of the Integrity VM product. If Serviceguard is running and any distributed guests are configured, the vswitch monitor is automatically started on the VM Host system when the VM Host system boots.

To start the vswitch monitor manually, use the following command:

```
# /sbin/init.d/vswitchmon start
```

To verify that the vswitch monitor is running, enter the following command:

```
# ps -ef | grep vswitchmon
```

Module 20

Using HP Serviceguard with Integrity Virtual Machines

The following example displays the output of `hpvmnet` before and after a Serviceguard LAN card failover:

Before:

```
# hpvmnet
```

Name	Number	State	Mode	PPA	MAC Address	IP Address
localnet	1	Up	Shared		N/A	N/A
switch1	2	Up	Shared	lan1	0x00306ef37c05	10.10.48.103

After:

```
# hpvmnet
```

Name	Number	State	Mode	PPA	MAC Address	IP Address
localnet	1	Up	Shared		N/A	N/A
switch1	3	Up	Shared	lan2	0x00306eea63d4	10.10.48.103

20-27. SLIDE: Summary



Summary

VMs as Serviceguard nodes:

- Serviceguard is installed on the Guest OS. Guest OS must be HP-UX.
- Serviceguard makes the applications running in VMs highly available
- VMs can be combined in a cluster with standard nodes (physical servers, nPar, vPar)

VMs as Serviceguard packages:

- Serviceguard is installed on the VM Host. Guest can be running any OS supported by Integrity Virtual Machines.
- Serviceguard makes the VM Guest highly available, does not monitor the applications running in the VM.
- Use the hpvm toolkit to configure VMs as packages

Student Notes

For further information consult the following docs.hp.com documentation:

- Managing Serviceguard
- HP Integrity Virtual Machines Installation, Configuration, and Administration HP Integrity Virtual Machines Version 2.0
- White Paper: Using Serviceguard to Manage Integrity Virtual Machines: Version 1.2

Module 20
Using HP Serviceguard with Integrity Virtual Machines

Appendix A — Serviceguard Command Summary

Serviceguard Commands

/usr/sbin/cmapplyconf	/usr/sbin/cmhaltserv	/usr/sbin/cmrunnode
/usr/sbin/cmcheckconf	/usr/sbin/cmmakepkg	/usr/sbin/cmrunpkg
/usr/sbin/cmdeleteconf	/usr/sbin/cmmigrate	/usr/sbin/cmrunserv
/usr/sbin/cmgetconf	/usr/sbin/cmmodnet	/usr/sbin/cmscancl
/usr/sbin/cmhaltcl	/usr/sbin/cmmodpkg	/usr/sbin/cmviewcl
/usr/sbin/cmhaltnode	/usr/sbin/cmquerycl	/usr/sbin/cmviewconf
/usr/sbin/cmhaltpkg	/usr/sbin/cmruncl	

Serviceguard Command Summary

cmapplyconf	Verify and apply Serviceguard cluster configuration and package configuration files
cmcheckconf	Check cluster configuration and/or package configuration files
cmdeleteconf	Delete either the cluster or the package configuration
cmgetconf	Get cluster or package configuration information
cmhaltcl	Halt/stop a high availability cluster
cmhaltnode	Halt/stop a node in a high availability cluster
cmhaltpkg	Halt/stop a high availability package
cmhaltserv*	Halt/stop a service from the high availability package halt script
cmmakepkg	Create a high availability package template file
cmmigrate	Migrate SwitchOver/UX to a Serviceguard cluster configuration file
cmmodnet*	Add or remove an IP address from a high availability cluster network
cmmodpkg	Enable or disable switching attributes for a high availability package
cmquerycl	Query cluster or node configuration information, create cluster configuration
cmruncl	Run/start a high availability cluster
cmrunnode	Run a node in an existing high availability cluster
cmrunpkg	Run/start a high availability package
cmrunserv*	Run/start a service from the high availability package run script
cmscancl	Gather system configuration information from nodes with Serviceguard installed.
cmviewcl	View information about a high availability cluster
cmviewconf	View Serviceguard cluster configuration information

** These commands are normally used within Serviceguard package scripts, not from the command line.*

Contributed Commands

/usr/contrib/bin/cmsetlog

- Adjust the log level(detail) of daemons

/usr/contrib/bin/cmsetsafety

- Disable/enable the kernel safety timer

Appendix A
Serviceguard Command Summary

```
***** W A R N I N G *****
*
* This is a special support tool supplied for debugging purposes
*
* only. It should only be used by trained and authorized HP personnel.
*
* Unauthorized or incorrect use of this tool can result in a system
*
* failure and possible data loss or corruption. Please contact your
*
* HP support representative.
*
*****
```

Appendix B — Managing Serviceguard with HP Openview

B-1. SLIDE: OpenView Products



Openview Products

- Generic SNMP Traps
- Network Node Manager
- Openview Operations for Unix and Windows
- Other Openview Products

Student Notes

In this module we will examine the ways that Serviceguard can be managed effectively with Openview products and other frameworks.

SNMP Management is a technique that can be used if you are only using network management tools or 3rd party frameworks that communicate primarily via SNMP. Serviceguard provides a SNMP subagent that can send alarms when cluster events happen and can be used to poll for cluster and package status.

Network Node Manager is the premier network management platform in use. It is a very mature platform for effective network and system discovery and management. It can receive alarms via snmp traps and poll cluster data via snmp.

Openview Operations is for system and application management since it employs agent technology to monitor and collect performance data on the Operating System and applications on a host.

We will then look at some additional Openview products that can be used to manage clusters and the applications on the systems. In addition, many of the Openview products which run on HP-UX can be configured to be made highly available as a cluster package.

B-2. SLIDE: SNMP Notification



SNMP Notification

- Serviceguard Notification
 - Log files (syslog.log)
 - SNMP Traps
- SNMP Traps
 - Sent by cmsnmpagt (Disabled by Default)
 - /etc/rc.config.d/cmsnmpagt
 - AUTOSTART_CMSNMPD=1
 - By all nodes every time cluster event happens
 - All trap destinations listed in
/etc/snmpd.conf -> /etc/SnmpAgent.d/snmpd.conf

Student Notes

By now you have seen the type of messages that Serviceguard puts in log files. This is the primary method by which you should be able to track cluster events.

The cluster events generate an SNMP trap which is a network packet that is sent out by each node in your cluster. An SNMP trap is a standard network management protocol that is used by all of the major network management products and devices.

To allow your system to send SNMP traps you need to enable this feature in the file /etc/rc.config.d/cmsnmpagt

```
# AUTOSTART_CMSNMPD:    If set to 1, the CM cluster SNMP subagent
#                        will be started automatically when the
#                        system boots. Set to 1 if you will be using
#                        ClusterView to monitor your Highly Available
#                        clusters through OpenView. Setting AUTOSTART_CMCLD
#                        to 1 in the /etc/rc.config.d/cmcluster file will
#                        cause the node to attempt to join its CM cluster
#                        automatically when the system boots.
#
#                        If set to 0, the CM cluster SNMP subagent
#                        will not be started.
#
```

Appendix B

Managing Serviceguard with HP OpenView

```
#
AUTOSTART_CMSNMPD=1
```

The agent that sends the traps will be started with the next reboot. If you wish to restart the agent without rebooting execute the following command.

```
# /sbin/init.d/cmsnmpagt start
```

The agent sends traps to all "trap-dest" listed in the file /etc/snmpd. If your network management station is not listed in this file automatically, you must manually add it. If you manually change this file you must restart the SNMP agent on the system as well as the Cluster subagent. A reboot will do this, or you can execute

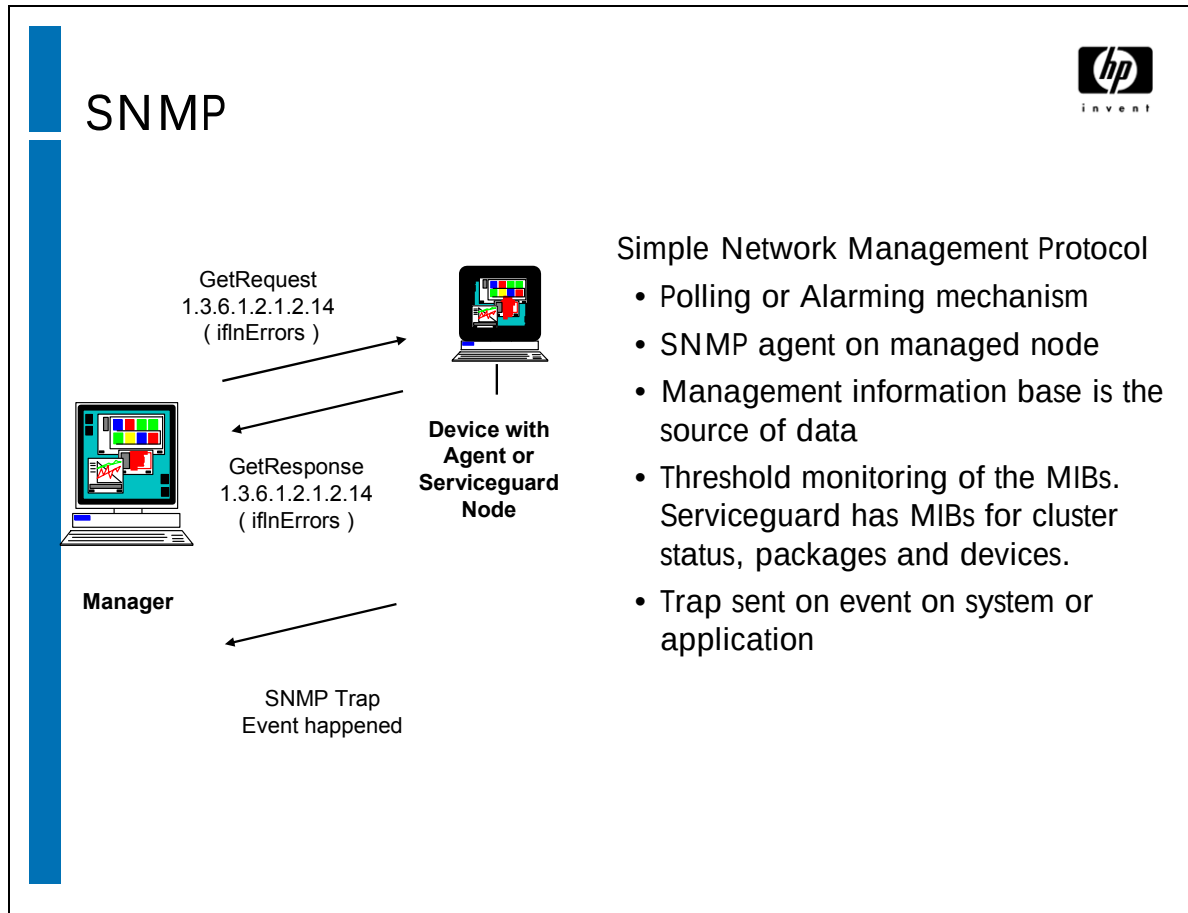
```
# ps -ef | grep snmpd
# kill <PID>
# cd /etc/snmpd
# /sbin/rc.init.d/cmsnmpagt stop
# /sbin/rc.init.d/cmsnmpagt start
```

The /etc/snmpd.conf file looks like:

```
#####
#
# snmpd.conf - HP / SNMP Research EMANATE SNMP Agent Configuration File
#
#####
#
# The EMANATE SNMP Agent included with HP OpenView Network Node Manager
# uses two configuration files to initialize operational settings on agent
# startup. They are:
#
#      snmpd.conf      Provides an HP OpenView SNMP Agent compatible
#      (this file)     configuration file format for the EMANATE SNMP
#                      Agent. The configuration values in this file
#                      override and/or supplement the configuration
#                      in snmpd.cnf.
#
#####
# get-community-name:   public
#set-community-name:   # enter community name
#contact:              # enter contact person for agent
#location:             # enter location of agent
#max-trap-dest:        # enter max no. of trap-dest entries to be maintained.

#trap-dest:            # enter trap destination
trap-dest: 156.153.206.68
trap-dest: 156.153.206.67
```

B-3. SLIDE: SNMP



Student Notes

- SNMP is a standards based protocol that NNM uses to get data from the device or program to the management station.
- There are three major components to this action: Manager, Agent and the MIB.
- In addition to the SNMP agent on the managed device being the initiator, you can configure Network Node manager to manage any MIB object on ANY/ALL devices on your network and when a metric on that device exceeds a threshold, that SNMP agent will send an alarm to NNM.
- SNMP is a polling mechanism or can be used to send alarms (traps) when something happens.

B-4. SLIDE: What Is Network Node Manager?

Network Node Manager



Device Management

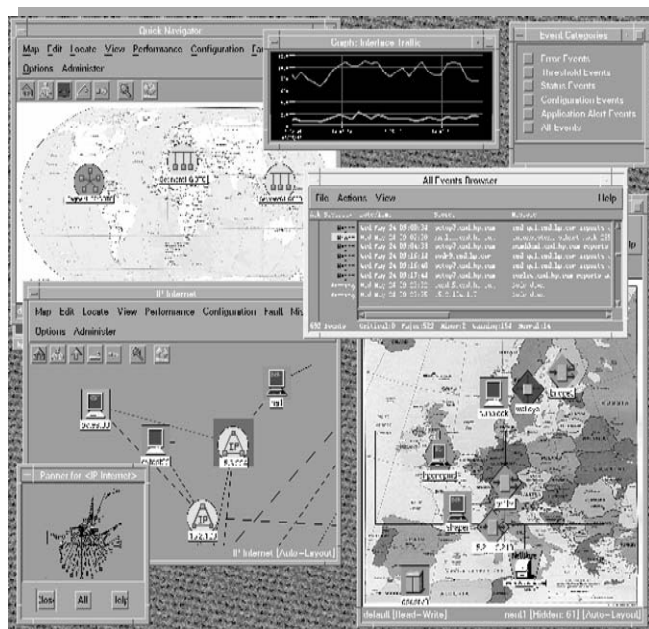
- There?
- Where?
- Up?
- Down?
- Talking?

Alarms

- External notification
- Auto actions
- Event forwarding

Data Source

- SNMP and MIBS



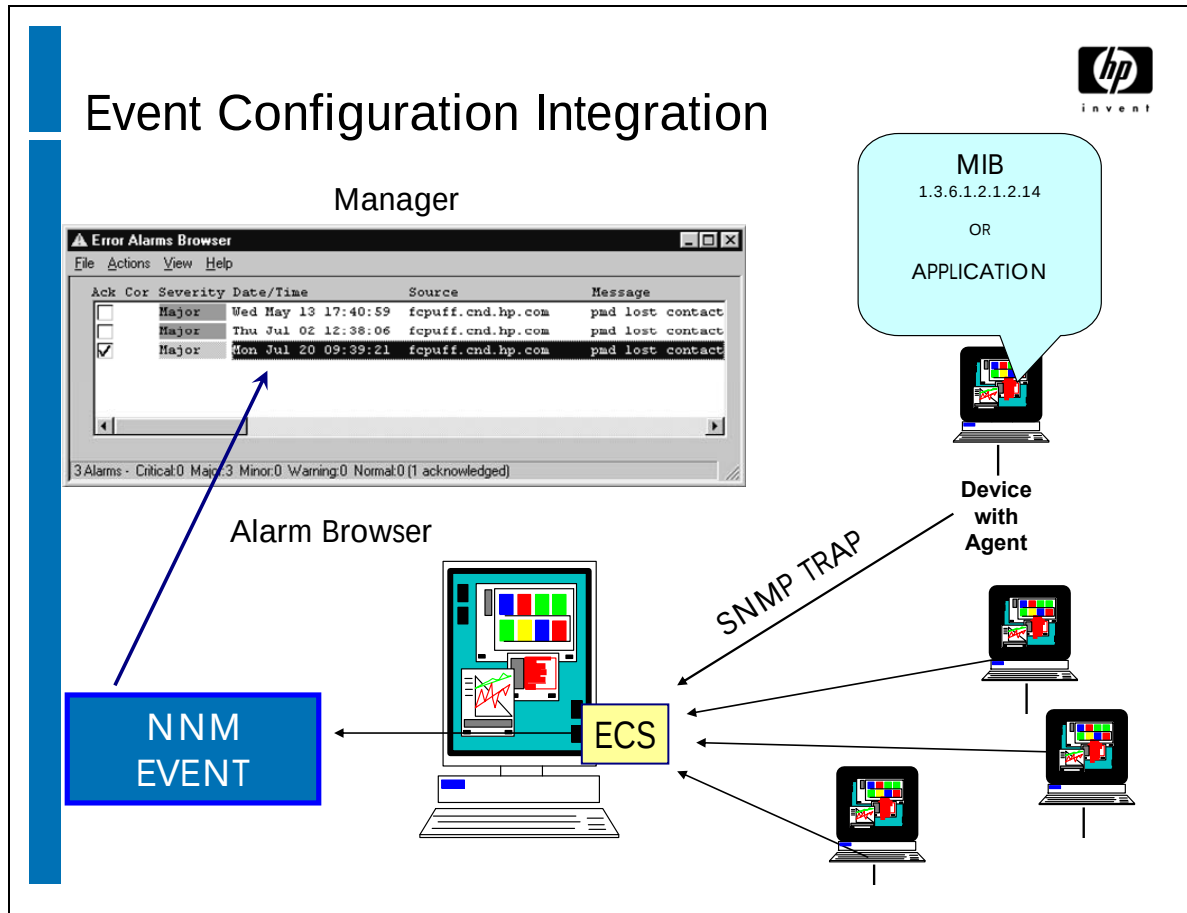
Student Notes

What Does NNM Provide?

HP OpenView Network Node Manager (NNM) provides you with a map of your network, shows you the status of objects on that network, and provides many tools for managing your network.

- Auto Discovery of network devices
- Displays "logical" network topology
- Uses standard SNMP technology
- Maintains database of network devices and status
- Allows filtering of network events
- Very flexible, many add on products, highly customizable

B-5. SLIDE: Event Configuration Integration



Student Notes

The event comes into the NNM application management server and then within NNM you can intercept and configure that event with an NNM built in GUI tool referred to as "Event Configuration".

- Event configuration is necessary if you want to:
Run a program when an event happens
- Generate a trouble ticket when a node goes down or specific network alarm occurs
- Process a specific SNMP trap received from an application or take any action ... when this event or alarm occurs.

This SNMP Trap can come from the SNMP Agent on a Server, on a Network device such as a bridge, hub, router, switch, or it may come from an application to your Central OV management server. Once in OV...the possibilities are endless.

You might have one single event on the net that generates many nuisance alarms. With event correlation Services ECS on the management server, NNM can correlate many hundreds or thousands of events into a single root cause event.

B-6. SLIDE: OpenView Operations Integration



OpenView Operations Integration

- Monitors systems, applications, service "operations"
- Management Server
 - HP-UX, Solaris, Windows (with OVOW)
- Agents
 - HPUX, Solaris, NT/2K, Sinix, TRU 64 Unix, SCO, RedHat Linux, Olivetti, Novell Netware, NCR, HP MPE/iX, Irix, Pyramid, IBM AS/400, Bull
- Utilizes SPIs (Smart Plug Ins)
- Service Management (Openview Navigator)
- Unix and Windows Management Server:
 - Functionally are the same.
 - The Agent technology is the same.
 - Products can work together for enterprise level scalability.

Student Notes

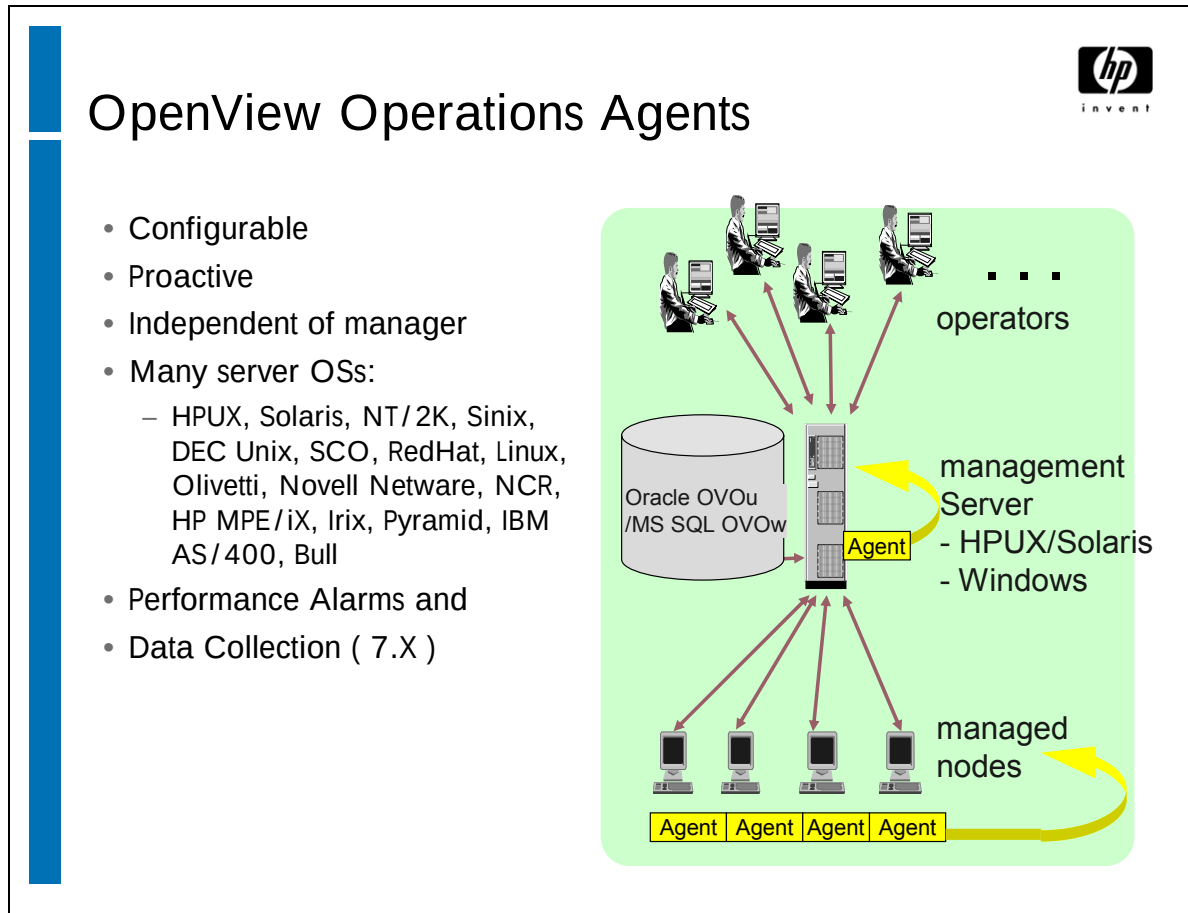
Openview Operations (OVO) is an OV product that is used to manage, systems, services, performance, and overall "operations".

Available on UNIX and Windows. The UNIX and Windows platform are functionally the same and utilize the same agent technology but there are some differences in the way the platform is setup and configured.

Both products can forward messages and alarms to each other to form a enterprise scaling framework. OVO can also receive SNMP traps directly or SNMP traps may be forwarded by NNM. Let us begin by taking a look at the two different platforms on which OVO is available ...

The operations staff can use a remote user interface that is not operating system specific for both platforms. There is a very robust "message browser" available in OVO with some added features and functionality. Some customers will choose to make OVO their single pane of glass and integrate all events, alarms, traps, and performance alarms into this single browser. How many of you are using OVO as a single pane of glass or common interface for your operations?

B-7. SLIDE: OpenView Operations Agents



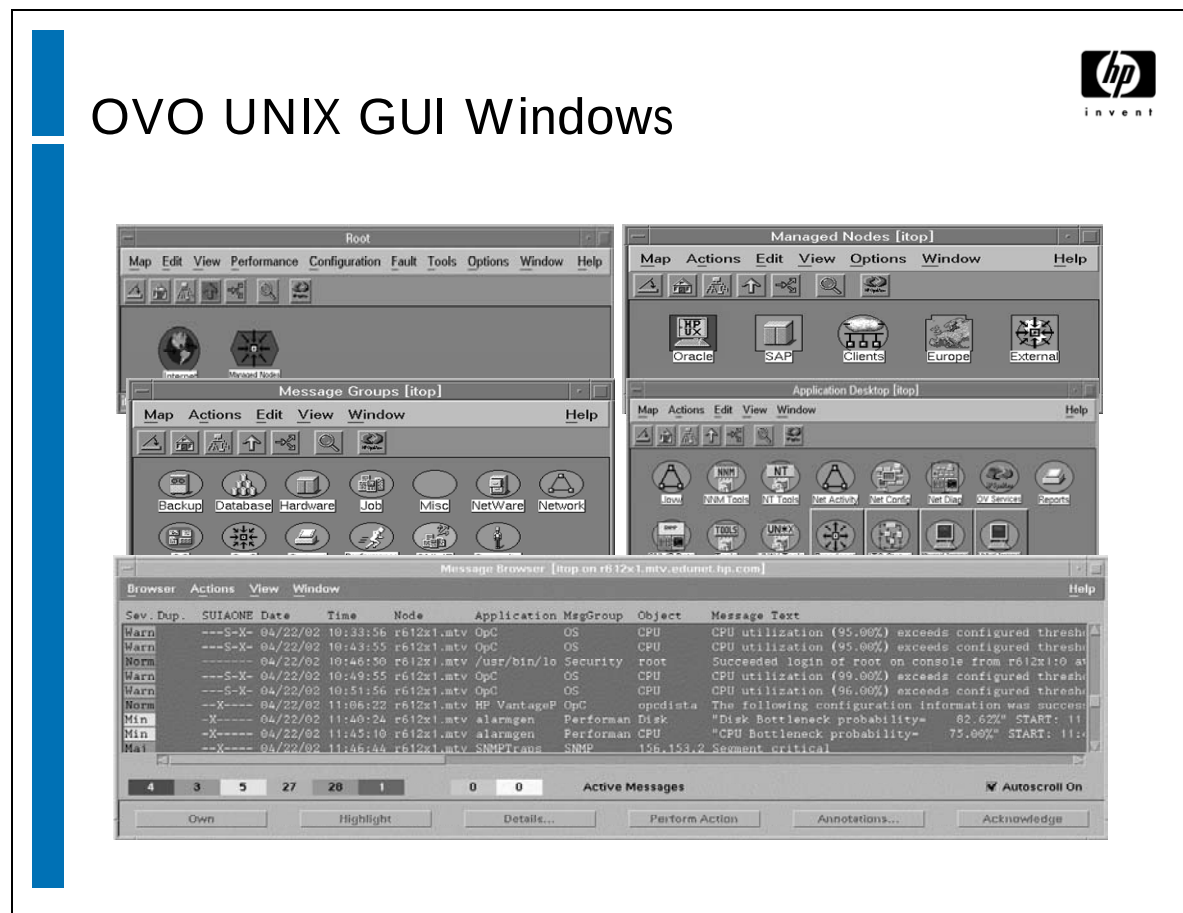
Student Notes

The source of data for OVO is provided by Agents. These OVO agents can be installed from the central management station and be pushed out to the managed node.

For example:

- Install the “OVO agent on a management server
- Create instructions for that Agent, telling what to manage
- You create instructions for an agent in UNIX via a template/policy.
- You create instructions for an agent in Windows via a policy.
- Once instructions (templates/policies) are created on the management server, the OVO administrator can Push them out to the agent residing on the managed node.
- If and when (and only when) there is an alarm (based on your instructions), the agent will send the event or alarm to the management server. The management system will not have to poll the node for this information because the agent is collecting and monitoring at the local level.
- OVO is also a recipient for SNMP traps and many other data sources as we will see.

B-8. SLIDE: OVO Unix GUI Windows



Student Notes

There is an OVO Admin interface as shown here for Openview Operations on UNIX. Alarms are generated by the Agents and displayed in the message browser. The affected nodes and message groups reflect the severity of the messages received.

A Java interface is available for operators only.

There are several indicators and each operator may have a different view of the world based on their job role or function. The customization for each operator is based on login ID, not on which interface you choose.

B-9. SLIDE: HP OpenView Unix Java GUI With Openview Navigator

HP OpenView UNIX Java GUI with Openview Navigator

Runs in a Web Browser or as a standalone JAVA application

More efficiency through easy identification of the root cause and the impacted services

Service-specific popup menu with appropriate actions

Student Notes

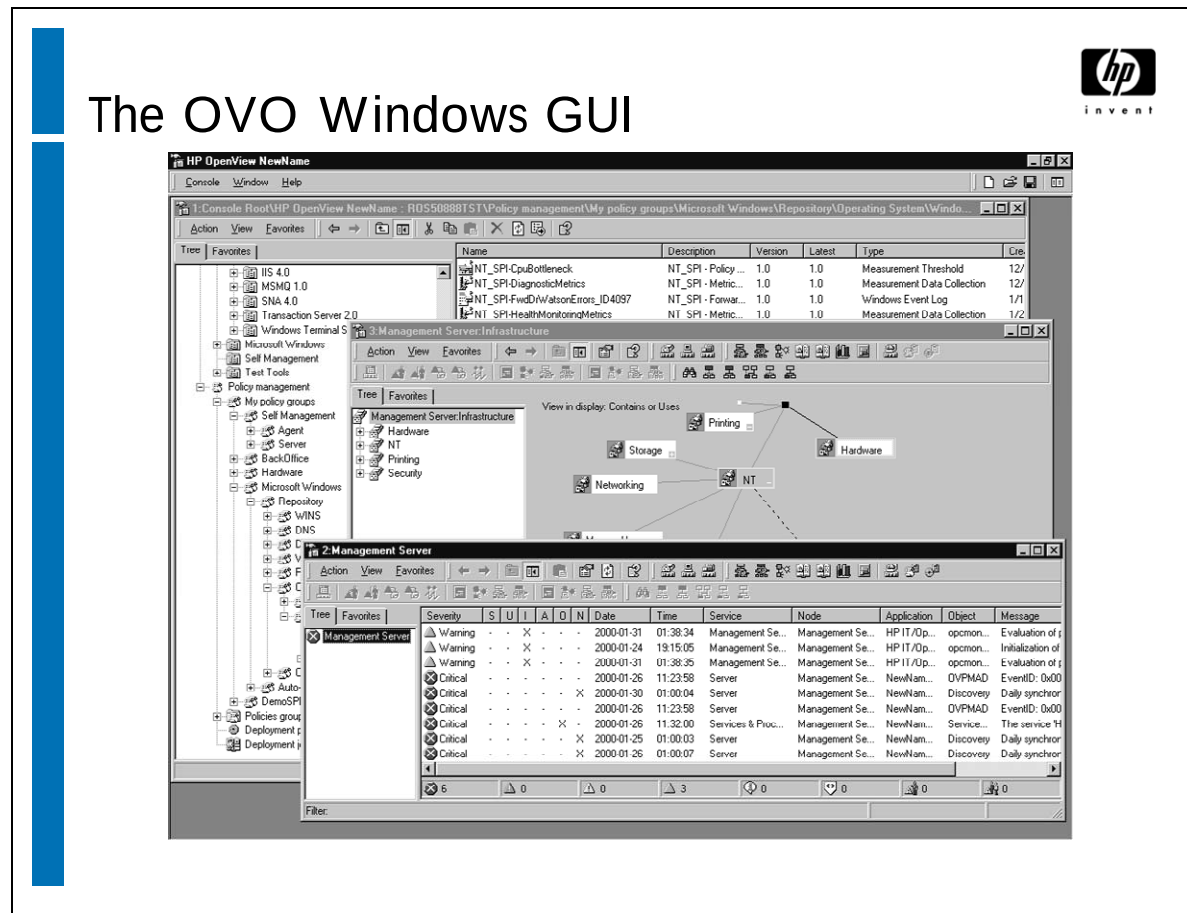
The operators have the option of using the very user friendly Java interface. The Java GUI is read only, and cannot be used for configuring or for customizing OVO.

OpenView Navigator will display only in the Java interface for OVO. Messages in the browser are related to a service. This provides a management view of the impact that an alarm has on a service.

The Service Navigator application now integrates the severity of impact for the event on one or more services in your infrastructure. This allows the operator to see the severity of an event on a service and then to drill down to the root cause of that service impact.

An SNMP trap from NNM or other source could come into OVO. OVO will propagate that to Service Navigator showing you all of the services that SNMP alarm has affected and the severity of that event on each service. This single source of data has now integrated or traversed through three different OV products.

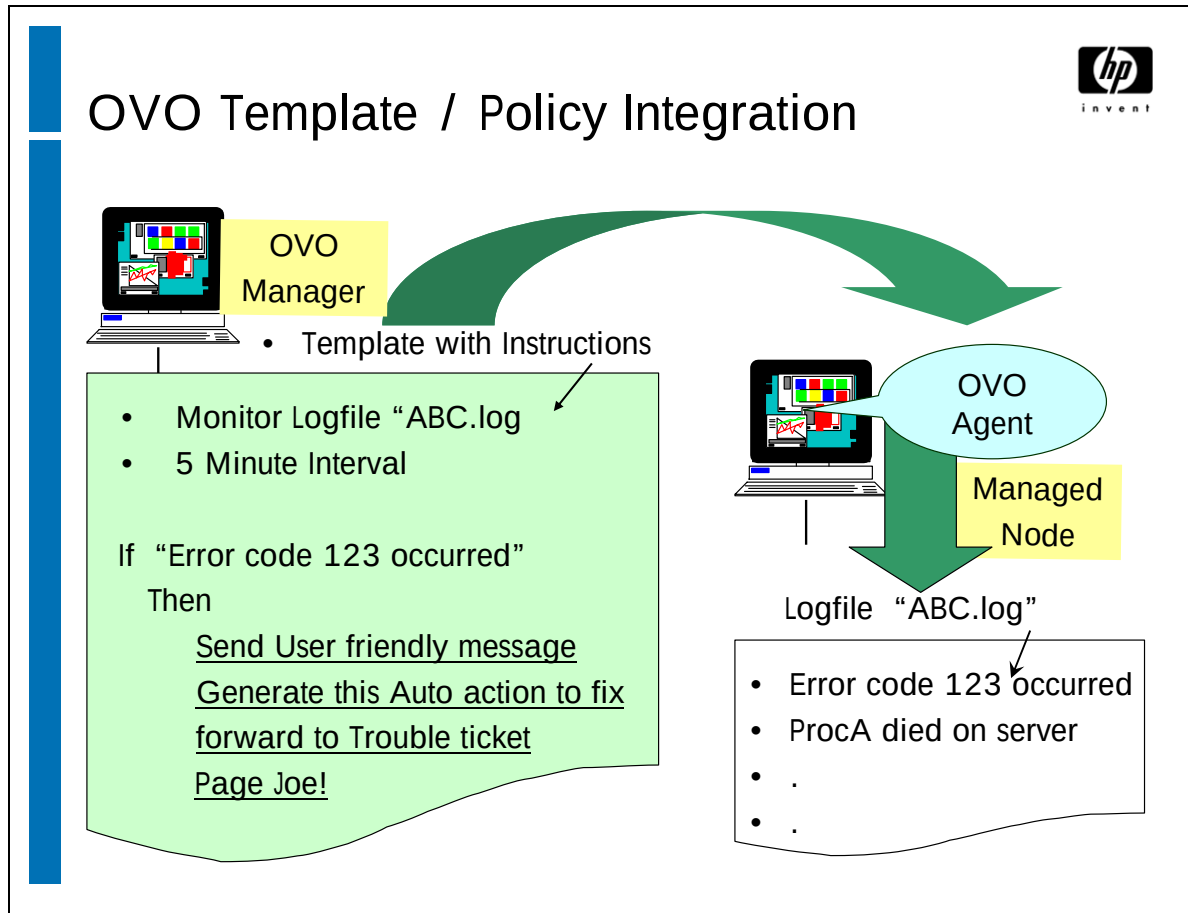
B-10. SLIDE: The OVO Windows GUI



Student Notes

This shows the Openview Operations for Windows user interface. Functionally, Openview Operations for Unix and Windows are the same product and can interoperate very effectively but are oriented towards different communities of IT departments.

B-11. SLIDE: OVO Template / Policy Integration



Student Notes

It is important to note that logfile encapsulation (monitoring log files) is only one of many sources of data that is available in OVO. All of the data sources are configured through the GUI for templates.

B-12. SLIDE: Sources of Data ... OVO



Sources Of Data ... OVO

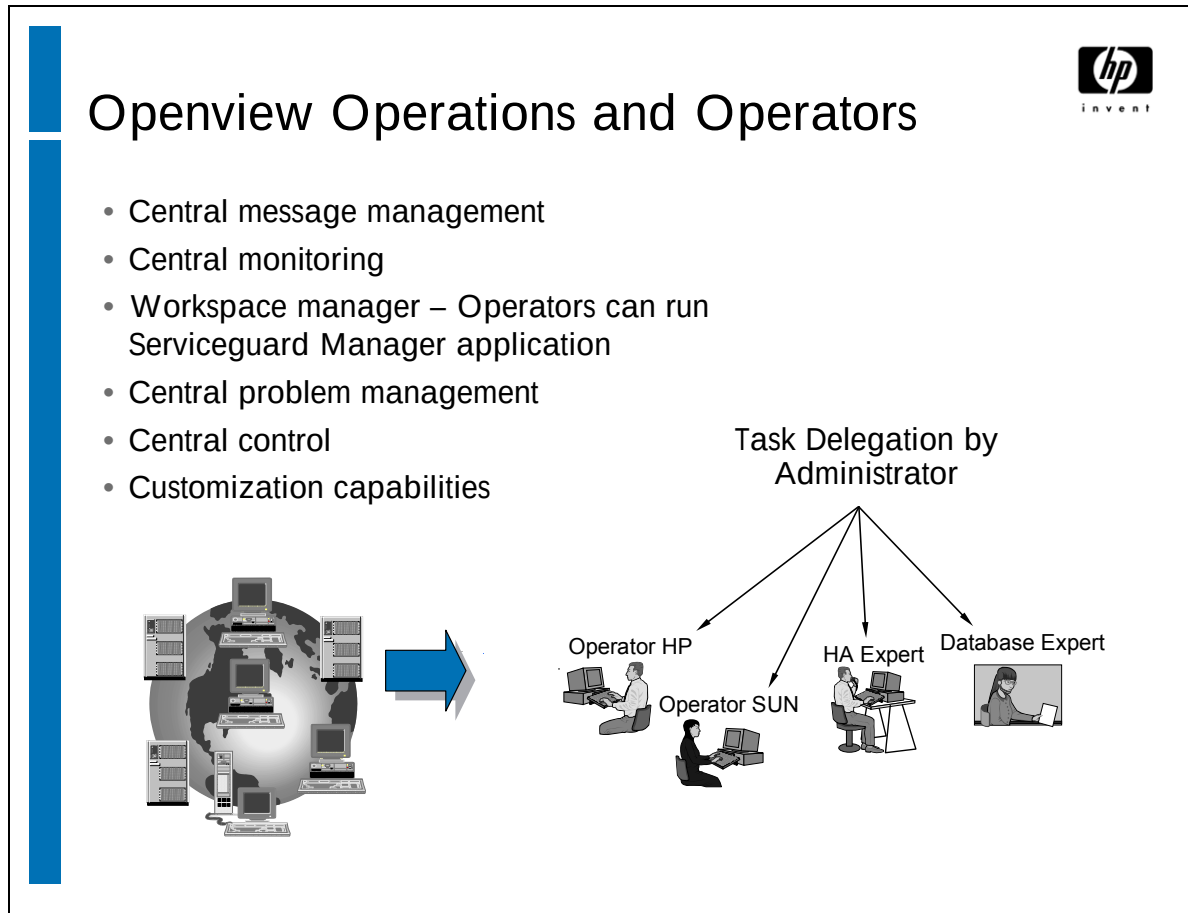
- Logfile
 - Check messages from OS or application logfiles
 - OVO comes with syslog.log policy that includes MC / ServiceGuard Messages
- Monitor
 - Run programs or MIB and check value
 - OVO Includes policies to monitor processes like cmcld,
- Schedule
 - Run Program at scheduled interval, generate message on success, failure or both
- SNMP trap – collect alarms from other programs like NNM
- Event Correlation – ECS circuit
- OPCMSG
 - API or command line
- WMI Policy (Windows only)

Student Notes

This slide lists the types of data sources for OVO and the templates that can be used to manage systems and applications. Many default templates are shipped with OVO for each operating system and groups of templates included in “Smart Plug-ins” are available for Operating systems and applications.

The term templates/policies is used in the UNIX platform and they are called policies on the Windows platform.

B-13. SLIDE: OpenView Operations and Operators



Student Notes

HP Openview Operations improves the productivity and management of multi-vendor distributed computing environments by providing:

- Central message management for consolidation, simplification and automation of message processing.
- Central monitoring to become more pro-active.
- Central problem management for notification, problem resolution, guidance, and tracking.
- Central control over distributed systems from a single console.
- Workspace manager to provide management information and capabilities according to the individual user's skills and responsibilities.
- Flexible customization capabilities for the administrator.

B-14. SLIDE: Other OpenView Products



Other Openview Products

- Openview Performance Agent
 - Collect OS Performance metrics every 5 minutes
 - Collect Process level metrics
 - Generate alarms with thresholds
- Openview Performance Manager
 - Graph OV Performance Agent metrics
- Openview Internet Services
 - Monitor and watch Application performance via simulation
 - Serviceguard applications can include DB and web servers
- Data Protector
 - Backup systems or application data.

Note: Many Openview products on HP-UX can be configured as a package in a Serviceguard configuration.

Student Notes

The Openview Performance Agent (formerly called Measureware) is a performance data collection tool that collects over 400 metrics every 1 or 5 minutes and is available for a series of platforms including HP-UX, Solaris, Linux, Windows, and AIX. This provides a consistent application and Operating system data collection technology across platforms.

The Openview Performance Manager is a graphing and data visualization tool for the Openview Performance Agent as well as the Agent for Openview Operations. It runs on HP-UX, Solaris and Windows.

Openview Internet Services is an application performance tool that collects via simulation data on web servers, web transactions, DNS servers, ODBC Servers, and other standard protocols. If your application packages that Serviceguard works with encompass standard protocols then applications can be monitored via OVIS.

Openview Data Protector backs up systems and application data on systems. The Virtual IP used by a Serviceguard package can also be used to define backups.

Many Openview products can also be setup as a Serviceguard package. These products include: Network Node Manager, Openview Operations for Unix, Data Protector, and Openview Performance Insight. This allows these applications to be highly available.

Appendix B
Managing Serviceguard with HP OpenView

Appendix C — Serviceguard Worksheets

C-1. SLIDE: Serviceguard Worksheets



Serviceguard Worksheets

For your use, the following worksheets are presented in this appendix:

- Hardware Configuration worksheet
- Power Supply Configuration worksheet
- Volume Group and Physical Volume worksheet
- Cluster Configuration worksheet
- Package Failover Behavior worksheet
- Package Configuration worksheet

Student Notes

For your use, we have prepared the following worksheets. They may be used as planning instruments as you prepare to configure one or more Serviceguard clusters.

Blank Hardware Worksheet

=====

SPU Information:

S800 Host Name _____ S800 Series No _____

Memory Capacity _____ Number of I/O Slots _____

=====

LAN Information:

Name of Subnet	Name of Interface	Node IP Addr	Traffic Type
_____	_____	_____	_____

Name of Subnet	Name of Interface	Node IP Addr	Traffic Type
_____	_____	_____	_____

Name of Subnet	Name of Interface	Node IP Addr	Traffic Type
_____	_____	_____	_____

=====

Serial (RS232) Heartbeat Interface Information:

Node Name _____ RS232 Device File _____

Node Name _____ RS232 Device File _____

=====

X.25 Information:

OTS subnet _____ OTS subnet _____

=====

Disk I/O Information for Shared Disks:

Bus Type _____	Slot Number _____	Address _____	Disk Device File _____
Bus Type _____	Slot Number _____	Address _____	Disk Device File _____
Bus Type _____	Slot Number _____	Address _____	Disk Device File _____
Bus Type _____	Slot Number _____	Address _____	Disk Device File _____

Attach a printout of the output from the `ioscan -fnC disk` command after installing disk hardware and rebooting the system. Mark this printout to indicate which physical volume group each disk belongs to.

Blank Power Supply Worksheet

=====

SPU Power:

S800 Host Name _____ Power Supply _____

S800 Host Name _____ Power Supply _____

=====

Disk Power:

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

Disk Unit _____ Power Supply _____

=====

Tape Backup Power:

Tape Unit _____ Power Supply _____

Tape Unit _____ Power Supply _____

=====

Other Power:

Unit Name _____ Power Supply _____

Unit Name _____ Power Supply _____

Blank Volume Group and Physical Volume Worksheet

=====

Volume Group Name: _____

Name of First Physical Volume Group: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Name of Second Physical Volume Group: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Physical Volume Name: _____

Cluster Configuration Worksheet

=====
Name and Nodes:
=====

Cluster Name: _____

Node Names: _____

Maximum Configured Packages: _____

Cluster Volume Groups: _____

=====
Subnets:
=====

Heartbeat IP Addresses: _____

Non-Heartbeat IP Addresses: _____

=====
Cluster Lock Volume Groups and Volumes:
=====

First Lock VG: First Lock Physical Volume:

_____ | Name (Node 1): _____ Disk Unit No: _____ Power Unit: _____

_____ | Name (Node 2): _____ Disk Unit No: _____ Power Unit: _____

Second Lock VG: Second Lock Physical Volume:

_____ | Name (Node 1): _____ Disk Unit No: _____ Power Unit: _____

_____ | Name (Node 2): _____ Disk Unit No: _____ Power Unit: _____

=====
Timing Parameters:
=====

Heartbeat Interval: _____
=====

Node Timeout: _____
=====

Network Polling Interval: _____
=====

Autostart Delay: _____

Package Configuration Worksheet

=====
Package Configuration File Data:
=====

Package Name: _____
Primary Node: _____
Failover Policy: _____ Failback Policy: _____
Primary Node: _____
First Failover Node: _____
Additional Failover Nodes: _____
Package Run Script: _____ Timeout: _____
Package Halt Script: _____ Timeout: _____
Package Switching Enabled? _____ Local Switching Enabled? _____
Node Failfast Enabled? _____
Additional Package Resource:
Resource Name: _____ Polling Interval _____ Resource UP Value _____

=====
Package Control Script Data:
=====

VG[0] _____ LV[0] _____ FS[0] _____ FS_MOUNT_OPT[0] _____
VG[1] _____ LV[1] _____ FS[1] _____ FS_MOUNT_OPT[1] _____
VG[2] _____ LV[2] _____ FS[2] _____ FS_MOUNT_OPT[2] _____

IP[0] _____ SUBNET[0] _____
IP[1] _____ SUBNET[1] _____

X.25 Resource Name _____

Service Name: _____ Run Command: _____ Retries: _____
Service Fail Fast Enabled? _____ Service Halt Timeout _____
Service Name: _____ Run Command: _____ Retries: _____
Service Fail Fast Enabled? _____ Service Halt Timeout _____

Appendix D — Integrating VxVM and Serviceguard

Objectives

Upon completion of this module, you will be able to do the following:

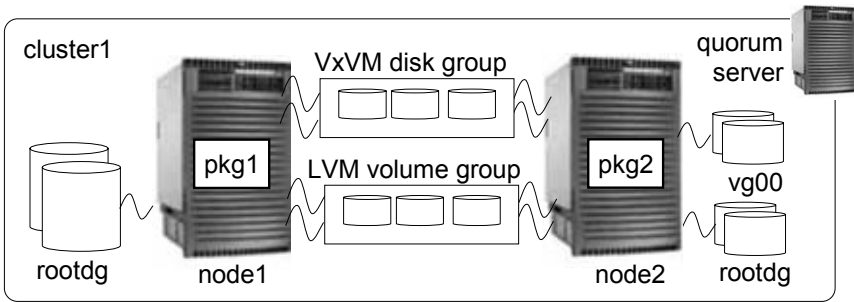
- Describe the role of VxVM disks, disk groups, and volumes in a Serviceguard cluster.
- Configure a basic VxVM rootdg, disk, disk group, and volume.
- Configure a basic VxVM-based Serviceguard package.

D-1. SLIDE: Serviceguard Disk Space Management Overview



Serviceguard Disk Space Management Overview

- Disk space in a Serviceguard cluster may be managed via ...
 - Logical Volume Manager (LVM)
 - Veritas Volume Manager (VxVM)
 - LVM *and* VxVM (in the same cluster/package, but not on the same disk)
- LVM and VxVM provide similar features and functionality



Student Notes

Disk space in a Serviceguard cluster may be managed using Logical Volume Manager (LVM), Veritas Volume Manager (VxVM), or some combination of the two.

The graphic on the slide depicts a two node cluster. One node has a VxVM rootdg boot disk. The second node has an LVM vg00 boot disk. The second node also has an empty VxVM rootdg disk group. A rootdg disk group is required on any system running VxVM.

The cluster currently has two packages: pkg1 and pkg2. One package appears to store its data in a VxVM disk group, and one package appears to store its data in an LVM volume group.

Though pkg1 is currently running on node1 and pkg2 is currently running on node2, the failure of either node would cause both packages and their associated disk/volume groups to be adopted by the other node.

Both LVM and VxVM provide online management, mirroring, striping, and other basic disk space management features. LVM has been available on HP-UX for over a decade, and continues to be the preferred disk space management solution for many cluster administrators. Some cluster administrators appreciate some of the more advanced cluster-friendly features provided by VxVM and have adopted this newer disk space management solution.

D-2. SLIDE: VxVM Terminology Overview



VxVM Terminology Overview

LVM	VxVM
Physical Volume	Disk Media
Volume Group	Disk Group
Logical Volume	Volume
Mirror	Data Plex
MWC / MCR	Log Plex
Extents	Subdisk
PVRA / VGRA	Private Region

Student Notes

Although these disk space management solutions differ in some respects, they share many similarities. The table on the slide introduces the VxVM equivalents to traditional LVM terms and concepts.

A disk that has been initialized for use in LVM is known as an LVM **Physical Volume**. Similarly, VxVM **Disk Media** objects represent disks that have been initialized for use in VxVM.

An LVM **Volume Group** is a collection of logical volumes and physical volumes that are managed together as a unit. Similarly, a VxVM **Disk Group** is a collection of volumes, disks, and other objects that are managed together as a unit.

An LVM **Logical Volume** is a virtual disk partition that may be configured as file system, swap, or raw application space. **Volumes** serve a similar purpose in VxVM.

Each **Mirror** associated with an LVM logical volume contains a complete image of the volume's data. Similarly, each **Data Plex** associated with a VxVM volume contains a complete image of the volume's data.

LVM uses the Mirror Write Cache / Mirror Consistency Record (**MWC/MCR**) structures to track pending write requests to mirrored volumes. After a system crash, LVM uses the information in the MCR to determine which portions of the system's logical volumes may require resynchronization. VxVM uses a similar structure called a **Log Plex** to track pending write requests to redundant volumes. After a system crash, VxVM uses these log plexes to determine which portions of the redundant volumes may require resynchronization.

LVM subdivides physical volumes into **Physical Extents**. Similarly, VxVM subdivides disks into contiguous swaths of disk space called **Subdisks**. Note, however, that the LVM extent size is set on a per volume group basis and doesn't change from disk to disk. VxVM subdisk sizes vary according to the subdisk's usage. One subdisk in a disk group may be 1K. Another subdisk on the same disk may be multiple gigabytes.

Every LVM physical volume has **PVRA/VGRA** headers that store configuration information for the disk's volume group. Similarly, every VxVM VM disk has a **Private Region** that contains configuration information for the disk's disk group.

D-3. SLIDE: VxVM Command Overview



VxVM Command Overview

LVM	VxVM
pvcreate	vxdisksetup
pvremove	vxdiskunsetup
vgcreate	vx dg init
vgextend	vx dg adddisk
vgreduce	vx dg rmdisk
vgremove	vx dg destroy
vgexport	vx dg deport
vgimport	vx dg import
lvcreate	vxassist make
lvremove	vxassist remove volume
lvextend	vxresize
lvreduce	vxresize
pvdisplay / vgdisplay / lvdisplay	vx disk / vxprint
pvmove	vxevac

Student Notes

The slide above provides a side-by-side comparison of the basic LVM/VxVM commands required to configure and manage basic VxVM disk groups and LVM volume groups. Although the command syntax varies, the concepts underlying the commands are relatively similar. The slides that follow explore these VxVM commands in greater detail.

D-4. SLIDE: Serviceguard/VxVM Software Overview



Serviceguard / VxVM Software Overview

- Base-VXVM, the core VxVM bundle, is included in every HP-UX 11i OE
- Serviceguard is included in the HP-UX 11i Mission Critical OE
- All of these products are also available on the HP-UX Applications DVD

Bundle	Codeword?	Comment
Base-VXVM	No	• Included in all HP-UX Operating Environments • Provides basic VxVM functionality, plus rootdg mirroring
B9116AA	Yes	• VxVM "Full" product • Required for mirroring/RAID5 in data disk groups
B9117AA	Yes	• VxVM "Cluster Volume Manager (CVM)" bundle • Required for disk group shared activation
B9118AA	Yes	• VxVM "Fast Resync Option" bundle • Enables fast mirror resync after splitting / merging mirrors
B3935DA	Yes	• Serviceguard product • Required on all nodes in a cluster
B8467BA	No	• Serviceguard "Quorum Server" product • Only required on the quorum server

Student Notes

The table on the slide describes the software products typically found in a Serviceguard or VxVM installation.

VxVM Software

The VxVM suite includes several components.

- Base VxVM provides the functionality necessary to create, import, and deport disk groups, and create, extend and reduce simple and striped volumes. Base-VXVM is included in all of the HP-UX 11i operating environments.
- The "Full" VxVM product bundle (B9116AA) adds the ability to create mirrored and RAID 5 volumes. B9116AA requires an additional license. Customers who plan to use array-based RAID functionality rather than VxVM software-based RAID to provide redundancy don't need this product.

Appendix D

Integrating VxVM and Serviceguard

- The Veritas “Cluster Volume Manager” product bundle (B9117AA) makes it possible to access a VxVM disk group read-only from one node in a cluster, while another node is concurrently making changes to the data in the disk group. B9117AA requires an additional license. Some customers use this functionality to do online, off-host backups. CVM is beyond the scope of this discussion.
- The VxVM “Fast Resync Option” bundle (B9118AA) dramatically improves performance when doing “snapshot” online backups with VxVM. B9118AA requires an additional license. Customers who plan to use array-based online backup functionality rather than VxVM’s snapshot volume online backup functionality don’t need this product.

Serviceguard Software

There are two Serviceguard bundles.

- B3935DA contains the Serviceguard product. This product must be installed on all nodes in the cluster.
- B8467BA contains the QuorumServer product. This product only needs to be installed on the quorum server.

The **swlist** command may be used to determine which of these products are installed on your system. If you installed Serviceguard as part of the Mission Critical Operating Environment, you may not have the B3935DA bundle wrapper – look for the Serviceguard *product* rather than the B3935DA *bundle*.

```
# swlist -l product ServiceGuard QuorumServer
# swlist -l bundle Base-VXVM B9116AA B9117AA B9118AA B3935AA
```

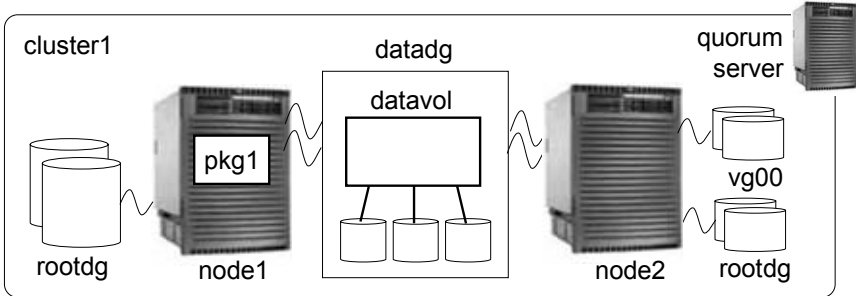
To install the products, mount the applications DVD and run `swinstall`.

D-5. SLIDE: Cookbook Overview



Cookbook Overview

- The remaining slides provide a cookbook for configuring the cluster shown below
- To learn more about VxVM, attend HP's H7085S course
- To learn more about Serviceguard, attend HP's H6487S course



Student Notes

The remaining slides provide a cookbook for configuring the simple cluster shown on the slide.

Note that this discussion only covers only the key features of VxVM and Serviceguard required to integrate the two products

- To learn more about VxVM mirroring, striping, recovery, performance, maintenance, and other issues, attend HP's H7085S VxVM course.
- To learn more about Serviceguard cluster and package configuration, attend HP's H6487S course.

D-6. SLIDE: Configure the rootdg (no vg00)



Configure the rootdg (no vg00)

- VxVM requires one special disk group on each node called rootdg
- If you are cold-installing HP-UX, Ignite-UX can configure rootdg as the boot disk
- For improved availability, mirror the rootdg boot disk

Cold install the rootdg boot disk using VxVM/VxFS

```

/opt/ignite/bin/itool ()
+-----+-----+-----+-----+-----+
| Basic | Software | System | File System | Advanced |
+-----+-----+-----+-----+-----+
| Configurations: [ HP-UX B.11.23 Default->]
| Environments:   [ Mission Critical OE-64bit->]
| Root Disk:     HP_36.4GMAS3367NC, 0/1/1/1.2.0, 34732 MBM
| File System:   VERITAS Volume Manager(VxVM) with VxFS
|                whole disk with VxFS
|                Logical Volume Manager(LVM) with VxFS
+-----+-----+-----+-----+-----+

```

Mirror the VxVM boot disk

```
node1# /etc/vx/bin/vxrootmir -v -b c0t0d0
```



Student Notes

Every system that uses VxVM must have a **rootdg** disk group with at least one disk. The **rootdg** is intended to store the operating system files and directories. If your prefer to configure your boot disk as an LVM volume group, **rootdg** will likely be empty, but must exist nonetheless in order to start the VxVM daemons.

If you are cold installing your system and wish to configure the boot disk via VxVM, select “VERITAS Volume Manager” from the Ignite File System pulldown menu. This creates a VxVM boot disk with the following VxVM volumes:

```
# bdf
```

Filesystem	kbytes	used	avail	%used	Mounted on
/dev/vx/dsk/rootdg/rootvol	409600	205816	202216	50%	/
/dev/vx/dsk/rootdg/standvol	307200	122072	183728	40%	/stand
/dev/vx/dsk/rootdg/varvol	2609152	1098472	1500408	42%	/var
/dev/vx/dsk/rootdg/usrvol	4465976	2577312	1873976	58%	/usr
/dev/vx/dsk/rootdg/tmpvol	204800	8496	194776	4%	/tmp
/dev/vx/dsk/rootdg/optvol	3660208	2154464	1494000	59%	/opt
/dev/vx/dsk/rootdg/homevol	20480	8432	11968	41%	/home

To ensure high availability, it's a good idea to mirror the VxVM boot disk using the automated vxrootmir shell script. Simply specify the *cxtxdx* name of the disk you wish to mirror to. The command works on both PARISC and Integrity servers.

```
# vxrootmir -v -b c0t0d0
vxrootmir: 10:57: Gathering information on the current VxVM root
configuration
vxrootmir: 10:57: Checking specified disk(s) for usability
vxrootmir: 10:57: Preparing disk c0t0d0 as a VxVM root disk
vxrootmir: 10:57: Disk c0t0d0 is now EFI partitioned disk c0t0d0s2
vxrootmir: 10:57: Adding disk c0t0d0s2 to rootdg as DM rootdisk02
vxrootmir: 10:57: Mirroring all volumes on root disk
vxrootmir: 10:57: Mirroring volume standvol
vxrootmir: 10:58: Mirroring volume swapvol
vxrootmir: 10:59: Mirroring volume rootvol
vxrootmir: 10:59: Mirroring volume tmpvol
vxrootmir: 10:59: Mirroring volume homevol
vxrootmir: 10:59: Mirroring volume optvol
vxrootmir: 11:01: Mirroring volume usrvol
vxrootmir: 11:03: Mirroring volume varvol
vxrootmir: 11:05: Current setboot values:
vxrootmir: 11:05: Primary:      0/1/1/1.2.0
vxrootmir: 11:05: Alternate:    <none>
vxrootmir: 11:05: Making mirror disk c0t0d0s2 (0/1/1/0.0.0) the alternate
boot disk
vxrootmir: 11:05: Disk c0t0d0s2 is now a mirrored root disk
```

D-7. SLIDE: Configure the rootdg (if vg00 already exists)



Configure the rootdg (if vg00 already exists)

- If the OS is installed in vg00, VxVM requires an empty rootdg
- Configure the rootdg via the interactive vxinstall script
- For improved availability, configure two disks in rootdg

Configure an empty rootdg with two disks

```
node2# vxinstall
...
Installation options for disk c1t1d1 Menu:
VolumeManager/Install/VolumeManager/Install/c0t0d0
1      Install as a new disk. (discards data on disk!)
2      Leave this disk alone.
?      Display help about menu
??     Display help about the menuing system
q      Exit from menus
Select an operation to perform: 1
...
```



Student Notes

If you already have an LVM boot disk, a **rootdg** disk group is still required if you plan to use VxVM volumes. Without a **rootdg** disk group, the **vxconfigd** daemon, which is required by VxVM, won't start. If **rootdg** is configured on a disk array, one RAID-protected LUN should be sufficient. If **rootdg** is configured on a JBOD or internal disk, however, configure two disks in **rootdg** to ensure that the **vxconfigd** daemon will still start if one of the disks in **rootdg** fails.

You can either convert your existing LVM **vg00** volume group into a VxVM **rootdg** disk group via the **vxcp_lvmroot** command, or use two unused disks to create an empty VxVM **rootdg**. Converting a **vg00** volume group to a **rootdg** disk group is beyond the scope of this discussion.

The interactive **vxinstall** shell script provides the easiest mechanism to configure a new **rootdg**.

1. First, remove old headers from the disk(s) you wish to configure in **rootdg**. VxVM won't overwrite existing LVM headers, even if the headers belong to a defunct volume group.

```
[node2] # dd if=/dev/zero of=/dev/rdisk/c0t0d0 bs=1024 count=1024
```

2. Run the `vxinstall` program to initialize VxVM. You only need to run this utility once, when you first install VxVM.

```
[node1] # vxinstall
```

3. When asked if you wish to enter a license key, answer `"n"`. We will only use the "base" VxVM functionality, which doesn't require a license.

```
Are you prepared to enter a license key [y,n,q,?] n
```

4. When asked if you wish to use enclosure based names for your disks, answer `"n"`. Enclosure-based names are most useful in disk array and SAN environments.

```
Do you want to use enclosure based names for all disks ? n
```

5. The program will interactively walk you through a series of menus. Whenever `vxinstall` asks if you wish to continue, press `[Return]`. If you make a mistake, you can press `^C` to break out and start over again at any point.
 - a) The program will search for and report your system's disk enclosures and/or controller cards. Press `[Return]`.
 - b) Next, `vxinstall` asks how you wish to configure the disks on your system. You can choose either a "Quick Installation" (in which VxVM initializes all currently unused disks with little interaction from the administrator) or "Custom Installation" (in which you get to choose which disks should and shouldn't be configured). Choose "Custom Installation".
 - c) If you chose "Custom Installation", `vxinstall` should list all of the currently unused disks on the system, and ask if you wish to initialize them. You can initialize selected disks, or all of the currently unused disks. For the sake of this exercise, choose "Install all disks as new disks".
 - d) Finally, VxVM gives you an opportunity to assign a meaningful name to each disk. These names will be used in place of device file names throughout the menus that follow. Accept the default disk names for the selected disks.

```
Use default disk names for these disks? [y,n,q,?] y
```
 - e) When asked to confirm that you wish to destroy the selected disks, answer `"y"`.

```
Are you sure you want to destroy the contents of this disk? y
```
 - f) Proceed through the remaining confirmation screens.

D-8. SLIDE: Configure VxVM Disks



Configure VxVM Disks

- Disks must be initialized with VxVM headers before they can be used in a disk group

Identify available disks

```
node1# vxdisk -o alldgs list
```

Clobber any existing headers on the disks

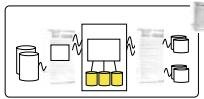
```
node1# dd if=/dev/zero of=/dev/rdisk/c1t1d1 bs=1024 count=1024
node1# dd if=/dev/zero of=/dev/rdisk/c2t2d2 bs=1024 count=1024
node1# dd if=/dev/zero of=/dev/rdisk/c3t3d3 bs=1024 count=1024
```

Initialize the disks for use in VxVM

```
node1# /etc/vx/bin/vxdisksetup -i c1t1d1
node1# /etc/vx/bin/vxdisksetup -i c2t2d2
node1# /etc/vx/bin/vxdisksetup -i c3t3d3
```

Verify that the disks initialized successfully

```
node1# vxdisk list
```



Student Notes

After configuring the **rootdg**, you will probably want to configure additional VxVM disks. A VxVM disk is a physical disk that has been initialized for use by VxVM. Almost any disk, from 500MB external disks to multi-terabyte HP XP disk array LUNs, can be initialized for use in VxVM.

Identifying Available Disks

The first step is to determine which disks are available. The `vxdisk` command is the best way to determine this. In a clustered environment, be sure to include the `-o alldgs` option. Otherwise, `vxdisk` will mistakenly indicate that disks being used by other systems on the SAN or in the cluster are unused.

Consider the sample `vxdisk -o alldgs list` output below:

```
[node1] # vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  -         -        online invalid
c2t2d2  simple  -         -        online invalidc3t3d3  simple  -
-       LVM
```

You should see several columns of output:

DEVICE	Lists the device file names of all accessible disks.
TYPE	Lists disk types. Other platforms use this column to distinguish memory-based devices from disk-based devices. VxVM on HP-UX only supports disk-based devices, so TYPE will always be simple .
DISK	Reports the VM disk names for disks that have been assigned to local disk groups.
GROUP	Reports which disk group each disk belongs to. If the disk group name appears in parentheses, then that indicates that the disk group isn't currently "imported" on the local system. Though these disks may be physically accessible, the disk groups that they belong to are owned by another node in your cluster, or another system on your SAN.
STATUS	Reports the current status of the disk as follows: Online Invalid indicates that the disk isn't being used by LVM or VxVM Online indicates that the disk has been initialized for use in VxVM LVM indicates that the disk has been initialized for use in LVM

Two categories of disks may be considered candidates for inclusion in a new disk group:

- Disks that report **Online Invalid** in the **STATUS** field are usually candidates. **Online Invalid** suggests that the disk doesn't have a recognizable LVM or VxVM header. Beware, however, that disks configured using the whole disk approach will also appear to be **Online Invalid**.
- Disks that report **Online** in the **STATUS** field, and don't list a disk group name in the **GROUP** field are also candidates. These disks have VxVM headers, but haven't yet been assigned to disk groups.

Clobbering Existing Headers

Disks that report **LVM** may or may not be available. If a disk belongs to a defunct volume group that wasn't properly removed, it will still have LVM headers, and will be reported as an LVM disk. If you are *certain* that the disk isn't currently in use, overwrite the LVM headers with the **dd** command.

```
node1# dd if=/dev/zero of=/dev/rdisk/c3t3d3 bs=1024 count=1024
```

After running **dd**, the disk should report status **online invalid**.

```
node1# vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  -         -        online invalid
c2t2d2  simple  -         -        online invalid
c3t3d3  simple  -         -        online invalid
```

Appendix D

Integrating VxVM and Serviceguard

Initialize the Disks for use in VxVM

Now initialize the disks

The command required to initialize a VxVM disk is `vxdisksetup`. The `-i` (initialize) option is required.

```
[node1] # vxdisksetup -i c1t1d1
[node1] # vxdisksetup -i c2t2d2
[node1] # vxdisksetup -i c3t3d3
```

Verify your work by running `vxdisk -o alldgs list`. The disk(s) should report online in the STATUS column.

```
[node1] # vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  -         -        online
c2t2d2  simple  -         -        online
c3t3d3  simple  -         -        online
```

D-9. SLIDE: Configure VxVM Disk Groups



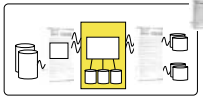
Configure VxVM Data Disk Groups

- A “disk group” is a group of disks that are managed together as a unit
- Disk group names are arbitrary, but typically end in “dg”
- Specify which disks you wish to include in the disk group
- Assign a “disk media name” to each disk
- Disk media names are formed by appending a unique 2-digit suffix to the dg name
- After disk group creation, disks are always referenced via their disk media names

Identify available disks
node1# vxdisk list

Create the disk group
node1# vxdg init datadg datadg01=c1t1d1 \
datadg02=c2t2d2 \
datadg03=c3t3d3

Verify that the disks initialized successfully
node1# vxdisk list



Student Notes

After initializing one or more VxVM disks, you can create a new disk group with the `vxdg` command.

Disk Group Overview

A VxVM disk group is a group of disks, volumes, and other VxVM objects that are managed together as a unit. When a disk group is “deported” from a node, the disks, volumes, and other objects within that disk group become inaccessible. When a disk group is “imported” on a node, the objects in that disk group become available. When a disk group is destroyed, the objects in the disk group are destroyed as well.

One special disk group, `rootdg`, must exist on every system running VxVM.

Other disk groups may be created as is necessary. You might, for instance, create a disk group called `oracledg` for your Oracle data. You might create another disk group called `financedg` for file systems and other partitions associated with your finance department. It is recommended that you (a) choose intuitive names for your disk groups and (b) append a “dg” suffix on the end of each disk group name.

Disk Media Overview

When a disk is assigned to a VxVM disk group, it is assigned a VxVM disk media name. Typically, disk media names are formed by appending an arbitrary two-digit number on the end of the disk's disk group name. Typically the number indicates the order in which the disks were added to the disk group; not the device's hardware address or physical location on the system.

The example on the slide shows a disk group called `datadg` that has three disks:

```
datadg01
datadg02
datadg03
```

Creating the Disk Group

Use the `vxvg` command to create a disk group.

```
[node1] # vxvg init datadg datadg01=c1t1d1 \
          datadg02=c2t2d2 \
          datadg03=c3t3d3
```

The options and arguments are explained below:

<code>init</code>	<code>vxvg</code> may be used for a variety of tasks from creating disk groups, to extending, reducing, and removing disk groups. The <code>init</code> argument indicates that you wish to initialize a new disk group.
<code>datadg</code>	Specify the name of the disk group you wish to create. By convention, disk groups typically include a dg suffix on the end of the disk group name.
<code>datadg01=c1t1d1</code>	Specify the disk name for each disk you want to include in the disk group, as well as the <code>cxtxdx</code> name of the associated disk device. Recall that VM disk names are formed by appending a two-digit number on the end of the disk group name. At least one disk must be included in the disk group initially, but you can include others, too, if you wish.

You can verify your work with `vxdisk`.

```
[node1] # vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  datadg01  datadg   online
c2t2d2  simple  datadg02  datadg   online
c3t3d3  simple  datadg03  datadg   online
```

D-10. SLIDE: Configure a VxVM Volume



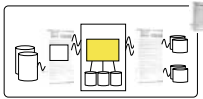
Configure a VxVM Volume

- A “volume” is a virtual partition of disk space in a disk group
- A volume is typically used to store a file system, swap, or raw data
- Volume names are arbitrary, but typically end in “vol”
- VxVM supports a variety of volume layouts

Create a non-mirrored volume
node1# vxassist -g datadg make datavol 16m

Create a mirrored volume
node1# vxassist -g datadg make datavol 16m layout=mirror nlog=1

Verify that the volume was created
node1# vxprint datavol



Student Notes

After you initialize your disk group, you can begin creating new volumes.

Volume Overview

Each disk group typically contains one or more volumes. A **volume** is a VxVM object representing a virtual partition that may be used to store a file system, raw data for an application, or swap space.

A disk group may contain one or many volumes, and each volume may be anywhere from 1K, to multiple terabytes in size.

Be sure to choose descriptive volume names so other administrators can easily determine how each volume is being used. Some administrators append a `vol` suffix on the end of each volume name, but this isn't necessary.

Appendix D

Integrating VxVM and Serviceguard

Volumes, like disks, are accessed via device files. Each volume has both a block and a character device file. The output below shows the device files for the `datavol` volume in the `datadg` disk group:

```
[node1] # ll /dev/vx/*/rootdg/datavol
brw----- 1 root root  1 0x000005 Sep 4 /dev/vx/dsk/datadg/datavol
crw----- 1 root root 54 0x000005 Sep 4 /dev/vx/rdsk/datadg/datavol
```

VxVM character device files are stored in disk group subdirectories under `/dev/vx/rdsk/`.
VxVM block device files are stored in disk group subdirectories under `/dev/vx/dsk/`.

Creating a Simple Volume

`vxassist` is the preferred command for creating and managing volumes. The sample command below creates a 16MB volume called `datavol` in the `datadg` disk group:

```
[node1] # vxassist -g datadg make datavol 16m
```

To ensure proper load balancing, some administrators prefer to specify which disk the volume is created on:

```
[node1] # vxassist -g datadg make datavol 16 datadg01
```

If the volume is too large to fit on one disk, you can list multiple disks on the end of the command line.

You can check your work with the `vxprint` command. Since the volume below is not mirrored, it has one data plex (`datavol-01`) which contains one subdisk (`datadg01-01`).

```
[node1] # vxprint -g datadg datavol
TY NAME          ASSOC          KSTATE  LENGTH  PLOFFS  STATE  TUTILO  PUTILO
v  datavol        fsgen           ENABLED 16384   -      ACTIVE -      -
pl datavol-01     datavol        ENABLED 16384   -      ACTIVE -      -
sd datadg01-01    datavol-01     ENABLED 16384   0      -      -      -
```

Creating a Mirrored Volume

Mirroring a volume ensures that the volume remains accessible, even if one data plex in the volume becomes inaccessible due to disk failure or other problems. In order to create a mirrored volume, you must have at least two disks in your disk group.

The sample command below creates 16MB mirrored volume called `datavol`. The `nlog=1` argument is optional, but highly recommended. It creates a log plex for the volume that ensures the plexes in the volume can be quickly resynchronized in case of a system crash.

```
[node1] # vxassist -g datadg make datavol 16m layout=mirror nlog=1
```

If you wish, you can specify which disks you want to use.

```
[node1] # vxassist -g datadg make datavol 16m layout=mirror nlog=1 \
          datadg01 datadg02 datadg03
```

The examples so far assume that you want to configure two plexes for the mirrored volume. VxVM actually supports up to 32 plexes per volume. Simply add the `nmirror` option to the `vxassist` command.

```
[node1] # vxassist -g datadg make datavol 16m layout=mirror \
          nmirror=2 nlog=1
```

You can check your work with the `vxprint` command. Note that mirrored volumes have multiple, equal-size plexes. This is the `vxprint` output for a volume with two data plexes and a log plex:

```
[node1] # vxprint -g datadg datavol
```

TY	NAME	ASSOC	KSTATE	LENGTH	PLOFFS	STATE	TUTILO	PUTILO
v	datavol	fsgen	ENABLED	16384	-	ACTIVE	-	-
pl	datavol-01	datavol	ENABLED	16384	-	ACTIVE	-	-
sd	datadg01-01	datavol-01	ENABLED	16384	0	-	-	-
pl	datavol-02	datavol	ENABLED	16384	-	ACTIVE	-	-
sd	datadg02-01	datavol-02	ENABLED	16384	0	-	-	-
pl	datavol-03	datavol	ENABLED	LOGONLY	-	ACTIVE	-	-
sd	datadg01-02	datavol-03	ENABLED	33	LOG	-	-	-

D-11. SLIDE: Import / Deport the Disk Group on the Other Node



Import/Deport the Disk Group on the Other Node

- Verify that the other node can import the disk group
- Unlike LVM, VxVM stores all of its disk group information in the disk headers
- Thus, there is no need to manually update /etc/lvmtab, device files, etc. on node2!

Unmount the file system and deport the disk group from the first node

```
node1# umount /data
node1# vxdg deport datadg
```

Import the disk group and mount the file system on the second node

```
node2# vxdg import datadg
node2# mkdir /data
node2# mount /dev/vx/dsk/datadg/datavol /data
```

Unmount the file system and deport the disk group from the second node

```
node2# umount /data
node2# vxdg deport datadg
```

Leave the disk group deported in preparation for package creation



Student Notes

Verify that the other node can import the disk group. Note that unlike LVM, VxVM stores all of its disk group information in the disk headers. Thus, there is no need to manually update /etc/lvmtab, device files, etc., on node2!

Unmount the file system and deport the disk group from the first node.

```
[node1] # umount /data
[node1] # vxdg deport datadg
```

Import the disk group and mount the file system on the second node.

```
[node2] # vxdg import datadg
[node2] # mkdir /data
[node2] # mount /dev/vx/dsk/datadg/datavol /data
```


Unmount the file system and deport the disk group from the second node.

```
[node2] # umount /data  
[node2] # vxvg deport datavg
```

Leave the disk group deported. It will be re-imported when the Serviceguard package is created and enabled later.

How Serviceguard Imports and Exports VxVM Diskgroups

Every Serviceguard package includes a package control script which executes automatically anytime a package is started or halted on a node. If a package includes VxVM disk groups, the package's control script uses the `vxvg deport` command to deport the package's disk groups when the package is halted, and `vxvg -tfc import` to import the package's disk groups when the package is restarted.

The `-t` option specifies that the disk group should be imported with the `noautoimport` flag, which means that the disk group will not be automatically re-imported at boot time. Since disk groups included in the package control script are only imported by Serviceguard packages, they should not be auto-imported by VxVM.

The `-f` option allows the disk group to be imported even if one or more disks (a mirror, for example) isn't currently available.

The `-C` option clears any existing host ID that might be written on the disk from a prior activation by another node in the cluster. If the disk had been in use on another node which has gone down with a TOC, then the other node's host ID may still be written on the disk. This needs to be cleared so the new node's ID can be written to the disk. Note that the disk groups are not imported clearing the host ID if the host ID is set and matches a node that is not in a failed state. This is to prevent accidental importation of a disk group on multiple nodes which could result in data corruption.

WARNING	Although Serviceguard uses the <code>-C</code> option within the package control script, this option should not normally be used from the command line.
----------------	---

D-12. SLIDE: Manage VxVM Disk Groups



Manage VxVM Disk Groups

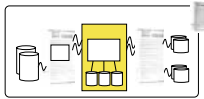
- Disk groups can be easily extended/reduced
- Unlike LVM, VxVM stores all of its disk group information in the disk headers
- Thus, there is no need to manually update /etc/lvmtab, device files, etc. on node2!

Add a disk to the disk group

```
node1# /etc/vx/bin/vxdisksetup -I c4t4d4
node1# vxdg -g datadg adddisk mydg04=c4t4d4
```

Remove a disk from a disk group

```
node1# vxevac -g datadg mydg04 mydg03
node1# vxdg -g datadg rmdisk mydg04
node1# /etc/vx/bin/vxdiskunsetup c4t4d4
```



Student Notes

Occasionally it may be necessary to add or remove disks in a disk group.

This task is much easier in VxVM versus LVM since all of the volume and disk group configuration information is stored in the disk headers – not in a configuration file like LVM's /etc/lvmtab file.

Adding a Disk to a Disk Group

Three commands are required to add a disk to a disk group. First, identify an available disk with `vxdisk -o alldgs list` and initialize it for use by VxVM.

```
[node1] # vxdisk -o alldgs list
[node1] # /etc/vx/bin/vxdisksetup -i c4t4d4
```

Next, add the disk to the disk group with the **vxdg** command.

```
[node1] # vxdg -g datadg adddisk datadg04=c4t4d4
```

You can verify your work with `vxdisk -o alldgs list`.

```
[node1] # vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  datadg01  datadg   online
c2t2d2  simple  datadg02  datadg   online
c3t3d3  simple  datadg03  datadg   online
```

Removing a Disk from a Disk Group

If a disk is underutilized, it might make sense to remove the disk from its current disk group and add it to a different disk group that can use the disk more effectively. Several steps are required. First, “evacuate” all the subdisks from the disk you wish to remove. You can specify a destination disk to move the subdisks to, or you can allow `vxevac` to find a new home for the subdisks automatically. The destination disk must be in the same disk group as the disk you intend to remove. If you wish to discard the data on the disk to be removed, use the procedure described on the previous slide to remove the appropriate volumes.

```
[node1] # vxevac -g datadg datadg04 datadg03
```

Next, remove the disk from the disk group via the `vx dg` command.

```
[node1] # vx dg -g datadg rmdisk datadg04
```

Finally, if you intend to use the disk in LVM, remove the VxVM headers from the disk.

```
[node1] # /etc/vx/bin/vxdiskunsetup c4t4d4
```

You may wish to verify your work with `vxdisk -o alldgs list`. Make sure the disk is `online invalid`.

```
[node1] # vxdisk -o alldgs list
DEVICE  TYPE    DISK      GROUP    STATUS
c0t0d0  simple  rootdg01  rootdg   online
c1t1d1  simple  datadg01  datadg   online
c2t2d2  simple  datadg02  datadg   online
c3t3d3  simple  datadg03  datadg   online
c4t4d4  simple  datadg03  datadg   online invalid
```

D-13. SLIDE: Manage VxVM Volumes



Manage VxVM Volumes

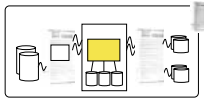
- Volumes can be easily extended/reduced
- The examples below assume that you have OnlineJFS

Extend a volume
node1# vxassist -g datadg growto datavol 32m

Extend a file system
node1# fsadm -F vxfs -b 32M /data

Reduce a file system
node1# fsadm -F vxfs -b 16m /data

Reduce a volume
node1# vxassist -g datadg shrinkto datavol 16m



Student Notes

Extending a Volume and File System

If a file system becomes full, it may be necessary to allocate additional space to a volume so the file system in the volume can be extended. The sample `vxassist` command below extends the `datavol` volume to 32 MB. Note that the `growto` argument specifies the final target size, *not* the amount of space that you wish to incrementally add to the volume.

```
[node1] # vxassist -g datadg growto datavol 32M
```

If you want to specify which disk the additional space should be taken from, specify a disk name on the end of the command line.

```
[node1] # vxassist -g datadg growto datavol 32M datadg01
```

In either of the commands above, you can use the **growby** argument in place of **growto** to specify the amount of space you want to add to the volume. The sample command below adds 16MB to **datavol**.

```
[node1] # vxassist -g datadg growby datavol 16M OR
[node1] # vxassist -g datadg growby datavol 16M datadg01
```

After extending the volume, extend the file system, too.

```
[node1] # fsadm -F vxfs -b 32m /data
```

vxprint to verify that the volume was properly extended. Note that the volume and plex sizes increased to 32 MB.

```
[node1] # vxprint -g datadg datavol
TY NAME          ASSOC          KSTATE  LENGTH PLOFFS  STATE  TUTILO  PUTILO
v  datavol        fsgen            ENABLED 32768  -      ACTIVE -      -
pl datavol-01     datavol        ENABLED 32768  -      ACTIVE -      -
sd datadg01-01    datavol-01     ENABLED 32768  0      -      -      -
```

Reducing a File System and Volume

The procedure for reducing a volume is similar to the process for extending a volume.

First, reduce the file system within the volume.

```
[node1] # fsadm -F vxfs -b 16m /data
```

Once the file system has been successfully reduced, it is safe to reduce the volume itself. The sample **vxassist** command below reduces the **datavol** volume to 16MB.

```
[node1] # vxassist -g datadg shrinkto datavol 16M
```

Alternatively, you can use the **shrinkby** argument to specify the amount of space you wish to remove from the volume. In either case, use **vxprint** to verify that the reduction succeeded.

```
[node1] # vxprint -g datadg
TY NAME          ASSOC          KSTATE  LENGTH PLOFFS  STATE  TUTILO  PUTILO
v  datavol        fsgen            ENABLED 16384  -      ACTIVE -      -
pl datavol-01     datavol        ENABLED 16384  -      ACTIVE -      -
sd datadg01-01    datavol-01     ENABLED 16384  0      -      -      -
pl datavol-02     datavol        ENABLED 16384  -      ACTIVE -      -
sd datadg02-01    datavol-02     ENABLED 16384  0      -      -      -
```

D-14. SLIDE: Configure a Package



Configure a Package

Create and customize a package startup script, and propagate it to all nodes.
Be sure to add your package's disk group names and filesystem mount parameters.

Create a template package control script

```
[node1] # cmmakepkg -s /etc/cmcluster/pkg1/pkg1.sh
```

Customize the package control script

```
[node1] # vi /etc/cmcluster/pkg1/pkg1.sh
# VxVM DISK GROUPS
VXVM_DG[0]="datadg"
...
# FILESYSTEMS
LV[0]=/dev/vx/dsk/datadg/datavol
FS[0]=/data;
FS_MOUNT_OPT[0]="-o rw"
FS_TYPE[0]="vxfsv"
```

Copy the package control script to the other node

```
[node1] # ssh node2 "mkdir /etc/cmcluster/pkg1"
[node1] # scp /etc/cmcluster/pkg1/pkg1.sh \
node2:/etc/cmcluster/pkg1/
```



Student Notes

Create a package control script for the package.

```
[node1] # cmmakepkg -s /etc/cmcluster/pkg1/pkg1.sh
```

Package control script is created.
This file must be edited before it can be used.

Edit the package control script. Look for the **VXVM_DG[]** array definition lines. Create an array entry for each VxVM disk group you wish to associate with the package. When Serviceguard runs a package on a node, it will use these array entries to determine which VxVM disk groups need to be imported on the node to support the package. Also add **LV[]**, **FS[]**, **FS_MOUNT_OPT[]**, and **FS_TYPE[]** array entries for each VxVM volume/file system that Serviceguard should mount to support the package. Other portions of the script may be customized to specify which network addresses and daemons/processes are included in the package. This discussion focuses only on the components required to support VxVM disk groups in Serviceguard packages.

```
[node1] # vi /etc/cmcluster/pkg1/pkg1.sh
# VXVM DISK GROUPS
VXVM_DG[0]="datadg"
...
# FILESYSTEMS
LV[0]=/dev/vx/dsk/datadg/datavol
FS[0]=/data
FS_MOUNT_OPT[0]="-o rw"
FS_TYPE[0]="vxfs"
```

Copy the control script to the other node.

```
[node1] # ssh node2 "mkdir /etc/cmcluster/pkg1"
[node1] # scp /etc/cmcluster/pkg1/pkg1.sh node2:/etc/cmcluster/pkg1/
pkg1.sh      100%   51KB   0.0KB/s   00:00
```

D-15. SLIDE: For Further Study



For Further Study

This appendix only discussed the most basic features of VxVM and Serviceguard. Administrators who manage VxVM/Serviceguard clusters should attend:

- H7085S (Veritas Volume Manager)
- H6487S (Hands-on with Serviceguard)
- U8601S (Disaster Tolerant Architectures with Serviceguard, Oracle 9i, and Oracle 10g RAC)

Student Notes

This discussion only covered the most basic aspects of VxVM and Serviceguard configuration. In order to successfully manage a VxVM/Serviceguard cluster, HP encourages customers to attend all three courses listed on the slide above.

Alternate Lab: New Package Configuration : xclock Package

Directions

Configure your cluster to contain a package, which displays an xclock on a screen.

Part 1: Create and Edit the Package Configuration File

On Node1

1. Make a directory for the xclock package files.

Answer:

```
# cd /etc/cmcluster
# mkdir xclock
```

2. Change directory to the xclock package directory that was just created. Once there, create the package configuration file template.

Answer:

```
# cd xclock
# cmmakepkg -v -p xclock.conf
```

3. Next, edit the package configuration file template.

```
# vi xclock.conf
```

and modify the following parameters:

PACKAGE_NAME	xclock
NODE_NAME	<node1>
NODE_NAME	<node2>
RUN_SCRIPT	/etc/cmcluster/xclock/xclock.cntl
RUN_SCRIPT_TIMEOUT	NO_TIMEOUT
HALT_SCRIPT	/etc/cmcluster/xclock/xclock.cntl
HALT_SCRIPT_TIMEOUT	NO_TIMEOUT
SERVICE_NAME	xclock_service
SUBNET	X.X.X.X ← Use netstat -in to obtain the 10-net subnet

4. Save the changes and exit the file.

Part 2: Create and Edit the Package Control Script, and Distribute this Script to All Nodes

On Node1

1. Create the package control script template.

Answer:

```
# cd /etc/cmcluster/xclock
# cmmakepkg -v -s xclock.cntl
```

2. Edit the package control script template. Remember that this file is a script and that all variable assignments cannot contain any spaces. See the “**Special Note**” box below before starting this step.

```
# vi xclock.cntl
```

and modify the following parameters:

```
SERVICE_NAME[0]="xclock_service"
SERVICE_CMD[0]="/usr/bin/X11/xclock -update 1 -bg blue \
               -title "Your_cluster_name -display _____"
SERVICE_RESTART[0]="-r 2"
```

Special Note

Ask your instructor for the value that you should use as an argument to the `-display` option.

3. Save changes and exit the file.
4. Ask your instructor if you should run `xhost +` on the machine where the `xclock` will display to make sure that host will accept a window from another machine. If your instructor says to run `xhost +`, please do so.

Answer:

```
# xhost +
```

 ← Only if needed.

5. Next, on Node2, create a subdirectory for your `xclock` package.

Answer:

```
# mkdir /etc/cmcluster/xclock
```

6. Now, back on Node1, make the `xclock` control script executable, and copy it to Node2 within your cluster.

Answer:

```
# chmod 755 xclock.cntl  
# rcp xclock.cntl <Node2>:/etc/cmcluster/xclock/.
```

7. Finally, on Node2, edit the `xclock.cntl` control script and change the background color of the `xclock` to **red**. Now, when a failover occurs, the color of the `xclock` will change. This change in color will indicate that the package has failed over since the control script on the second node starts the package (and starts `xclock` with a red background).

```
# cd /etc/cmcluster/xclock  
# vi xclock.cntl
```

Part 3: Create and Distribute the Binary File, and Start the Package

On Node1

1. Check the configuration file for errors and fix any that you find.

Answer:

```
# cd /etc/cmcluster/xclock
# cmcheckconf -v -P xclock.conf
```

2. Apply the new configuration, which will compile and distribute a new cluster binary file.

Answer:

```
# cd /etc/cmcluster/xclock
# cmapplyconf -v -P xclock.conf
```

3. If the cluster is not started, start it now. If the cluster was already running, start your xc lock package.

Answer:

```
# cmruncl -v
# cmmodpkg -e xclock
```

4. View the package status. We should see that the xclock package is now running on its primary node.

Answer:

```
# cmviewcl -v
```

5. We can test the `xclock` package behavior by doing the following:

```
# cmhaltpkg -v xclock
# cmviewcl -v

# cmrunpkg -v -n <adoptive_node> xclock
# cmmodpkg -v -e xclock

# cmviewcl -v
```

6. We can further test our package by killing the `xclock` PID. We will kill the `xclock` process PID 3 times; the first two times because we specified 2 retries, and the third time to force a failover.

Because of step 5 above, the package should now be running on `<adoptive_node>`. Execute the following steps on `<adoptive_node>` to force a failover back to `<primary_node>`.

```
# ps -ef | grep xclock      ← Kill the xclock process for the first time.
# kill <pid>                ← The process should restart on the same node.

# ps -ef | grep xclock      ← Kill the xclock process for the second time.
# kill <pid>                ← The process should restart on the same node.

# ps -ef | grep xclock      ← Kill the xclock process for the third time.
# kill <pid>                ← Process should fail over to <primary node>.

# cmviewcl -v              ← The xclock package restarts on <primary_node>.

# cmmodpkg -v -e -n <adoptive_node> xclock

# cmviewcl -v              ← xclock is now running on <primary node>. */
```

7. The package should now be running on `<primary_node>`. Execute the following steps on `<primary_node>` to force a failover back to `<adoptive_node>`.

```
# ps -ef | grep xclock      ← Kill the xclock process for the first time.
# kill <pid>                ← The process should restart on the same node.

# ps -ef | grep xclock      ← Kill the xclock process for the second time.
# kill <pid>                ← The process should restart on the same node.

# ps -ef | grep xclock      ← Kill the xclock process for the third time.
# kill <pid>                ← Process should fail over to <adoptive node>.

# cmviewcl -v              ← The xclock package restarts on <adoptive_node>.
# cmmodpkg -v -e -n <primary_node> xclock

# cmviewcl -v              ← xclock now running on <adoptive_node>.
```

Part 4: Modifying FAILBACK_POLICY

Directions

1. Modify your `xclock` package to fail back when the primary node comes back to the cluster, and then test it.

Answer:

In the "`xclock.conf`" file, change the line that now reads ...

FAILBACK_POLICY	MANUAL
-----------------	--------

to read ...

FAILBACK_POLICY	AUTOMATIC
-----------------	-----------

and then reconfigure the package in the normal way.

```
# cmhaltcl -f
# cd /etc/cmcluster/xclock
# cmapplyconf -P xclock.conf
# cmruncl
# cmviewcl -vp xclock
# cmhaltnode -f (on primary)
# cmrunnode (on primary)
```

2. Did the `xclock` package come back to the primary node?

Answer:

Yes.

3. While there might be compelling reasons to use the `AUTOMATIC` failback policy, we will be performing other operations with the `xclock` package that require the `MANUAL` failback policy. Therefore, modify your `xclock` package configuration file again to reflect a `MANUAL` failback.

Answer:

```
# cmhaltpkg xclock
# vi /etc/cmcluster/xclock.conf
```

Change `FAILBACK` from `AUTOMATIC` to `MANUAL`

```
# cmcheckconf -P /etc/cmcluster/xclock.conf
# cmapplyconf -P /etc/cmcluster/xclock.conf
```

Part 5: Adding a Package Monitoring Script to the xclock Package

Directions

1. Instead of using `/usr/bin/X11/xclock` as the service, create a separate service monitoring script called `/etc/cmcluster/xclock/xclock.mon` as follows:

```
# cd /etc/cmcluster/xclock
```

```
# vi xclock.mon
```

```
#!/usr/bin/sh

LOG="/etc/cmcluster/xclock/xclock.cntl.log"

echo "Now entering the xclock package monitor \c"      >> $LOG
echo "script on $(hostname) at $(date)."                >> $LOG
while true
do
    if ps -ef | grep -v grep | grep -q "/usr/bin/X11/xclock"
    then
        echo "Package xclock apparently ok \c"
        echo "on $(hostname) at $(date)."
        sleep 10
    else
        echo "Package xclock failed at $(date) \c"
        echo "from node $(hostname)."
        exit
    fi
done >> $LOG
```

The extra echo statements in the above script are sometimes referred to as "debug lines". Once the package and its scripts are working properly, these debug lines may be safely commented out. Good programming practice would suggest that these "debug lines" would be commented out (but not removed) from the script since they may be needed again in the future.

Ensure that the monitor is executable and copy it to the other system in your cluster.

Answer:

```
# chmod 755 /etc/cmcluster/xclock/xclock.mon
# rcp xclock.mon <node2>:/etc/cmcluster/xclock/.
```

2. In the `xclock` package control script `/etc/cmcluster/xclock/xclock.cntl`, change the following two lines.

```
SERVICE_CMD[0]="/etc/cmcluster/xclock/xclock.mon"  
SERVICE_RESTART[0]="-r 0"
```

Serviceguard will now be watching the `xclock.mon` process, and will use zero restarts. In turn, the `xclock.mon` process will be watching the health of `xclock` itself.

3. In the `customer_defined_run_cmds` function of the `xclock` package control script, `/etc/cmcluster/xclock/xclock.cntl`, remove the line beginning with the colon (:). Then, in its place, enter the command to start our `xclock`. Do not remove the `"test_return 51"` line.

```
/usr/bin/X11/xclock -update 1 -bg blue \  
-title Your_cluster_name -display _____ &
```

Important: Don't forget to put your `xclock` process in the background!

4. In the `customer_defined_halt_cmds` function, remove the line beginning with the colon (:). Do not remove the `"test_return 52"` line. Insert the following lines just before the `"test_return 52"` line. This will stop your `xclock` if it is running:

```
function customer_defined_halt_cmds  
{  
  LOG="/etc/cmcluster/xclock/xclock.cntl.log"  
  
  echo "Now entering the customer_defined_halt_cmds \c"          >> $LOG  
  echo "on $(hostname) at $(date)."  
  if ps -ef | grep -v grep | grep -q "/usr/bin/X11/xclock"  
  then  
    echo "Found the xclock process at $(date)."  
    echo "Killing xclock process at $(date)."  
    kill -9 $(ps -ef | grep -v grep | grep "/usr/bin/X11/xclock" \  
      | cut -c10-14)  
  else  
    echo "Note: In customer_defined_halt_cmds \c"          >> $LOG  
    echo "on $(hostname), and could not find the \c"        >> $LOG  
    echo "xclock process at $(date)."  
  fi  
}
```

NOTE: If Serviceguard believes the halt script did not properly stop the application, the halt script will fail, and Serviceguard will not start the package on the adoptive node to avoid risking application data corruption.

5. Now, copy the revised package control script to node2.

Answer:

```
# rcp /etc/cmcluster/xclock/xclock.cntl <node2>:/etc/cmcluster/xclock
```

6. On node_2, again edit the control script to change the xc lock background color to **red**.

Answer:

```
# vi xclock.cntl
```

7. Now verify that all the pieces work. Use your "xclock pkg" desktop (or open a terminal window and issue the `tail -f /etc/cmcluster/xclock/xclock.cntl.log` command) to view the cluster activities.

Answer:

```
# tail -f /etc/cmcluster/xclock/xclock.cntl.log
```

8. Halt the cluster and then verify that the xc lock program has also exited.

Answer:

```
# cmhaltctl -v -f  
# ps -ef | grep xclock    ( if necessary, kill the PID for the xc lock process )
```

9. Now, restart the cluster, which automatically restarts the xc lock package.

Answer:

```
# cmrunctl -v
```

10. Kill the `xclock` process (one time) and verify that it restarts on the adoptive node.

Answer:

```
# ps -ef | grep xclock
# kill <pid>
# cmviewcl -v -p xclock
```

11. Verify that the echo message was written to the package log file on the failing node.

Answer:

```
# vi xclock.cntl.log
```

12. Return the `xclock` package to the primary node by halting, then immediately restarting the cluster.

Answer:

```
# cmhaltcl -v -f
# cmruncl -v
```

Part 6: xclock Package run / halt Script Modification

Directions

1. First, recognize that the "**cluster symmetry**" that was discussed in Module 3 has been seriously violated in our xclock package. Remember that we are currently running two different xclock.cntl files. The package control script on the primary node directs the xclock application to start with a blue background while the same script on the adoptive node directs the xclock application to start with a red background. This configuration violates cluster symmetry.

In cases like this, where different code is required for each server in the cluster, the strong recommendation is to avoid maintaining multiple versions of the run / halt script (one for each server in the cluster). Instead, use logic in the script to maintain a single version of the run/halt script that would be appropriate on all nodes in the cluster, thus maintaining a symmetric cluster. To fix this, modify the control script as follows:

```
if [ $(hostname) = "primary_node_name" ]
then
    COLOR="blue"
else
    COLOR="red"
fi
/usr/bin/X11/xclock -bg "$COLOR" -title "Your_cluster_name" \
-update 1 -display _____ &
```

2. Copy the control script to the adoptive node.

Answer:

```
# rcp xclock.cntl <node2>:/etc/cmcluster/xclock/.
```

3. To activate the above modifications, halt, and then immediately restart our xclock package.

The xclock package should be running on the primary node.

Answer:

```
# cmhaltpkg -v xclock
# cmrunpkg -v -n "primary_node" xclock
# cmmodpkg -v -e xclock
```

4. Test the revised xc lock package by killing the xc lock process. The xc lock package will start on the adoptive node.

After the package fails over to the adoptive node, re-enable the primary node to run the package.

Answer:

```
# ps -ef | grep xclock
# kill <pid>
# cmmodpkg -v -e -n <primary_node> xclock
```

5. To further test our revised xc lock package, halt and restart it.
This should return the xc lock package to its primary node.

Answer:

```
# cmhaltpkg -v xclock
# cmrunpkg -v xclock
# cmmodpkg -v -e xclock
```

6. Congratulations on returning your cluster to a symmetric state!

Appendix F — HP Serviceguard Storage Management Suite for HP-UX 11i

F-1. SLIDE: Oracle Is Pivotal to your Business



Oracle Is Pivotal to your Business



Your most critical data lives on Oracle



If you're responsible for the Oracle environment, you're responsible for the wellness of the business



Oracle data and applications touch a plethora of other applications in your organization, customers and suppliers



If Oracle environment stumbles, the ramifications are far reaching...

Oracle applications support critical functions

Student Notes

Chances are your customers are running Oracle DB in their environment. If so, they are probably using Serviceguard to protect that environment. Our new Serviceguard Storage Management Suite offers some great new capabilities that these Oracle customers can take advantage of.

F-2. SLIDE: Common Customer Pain Points

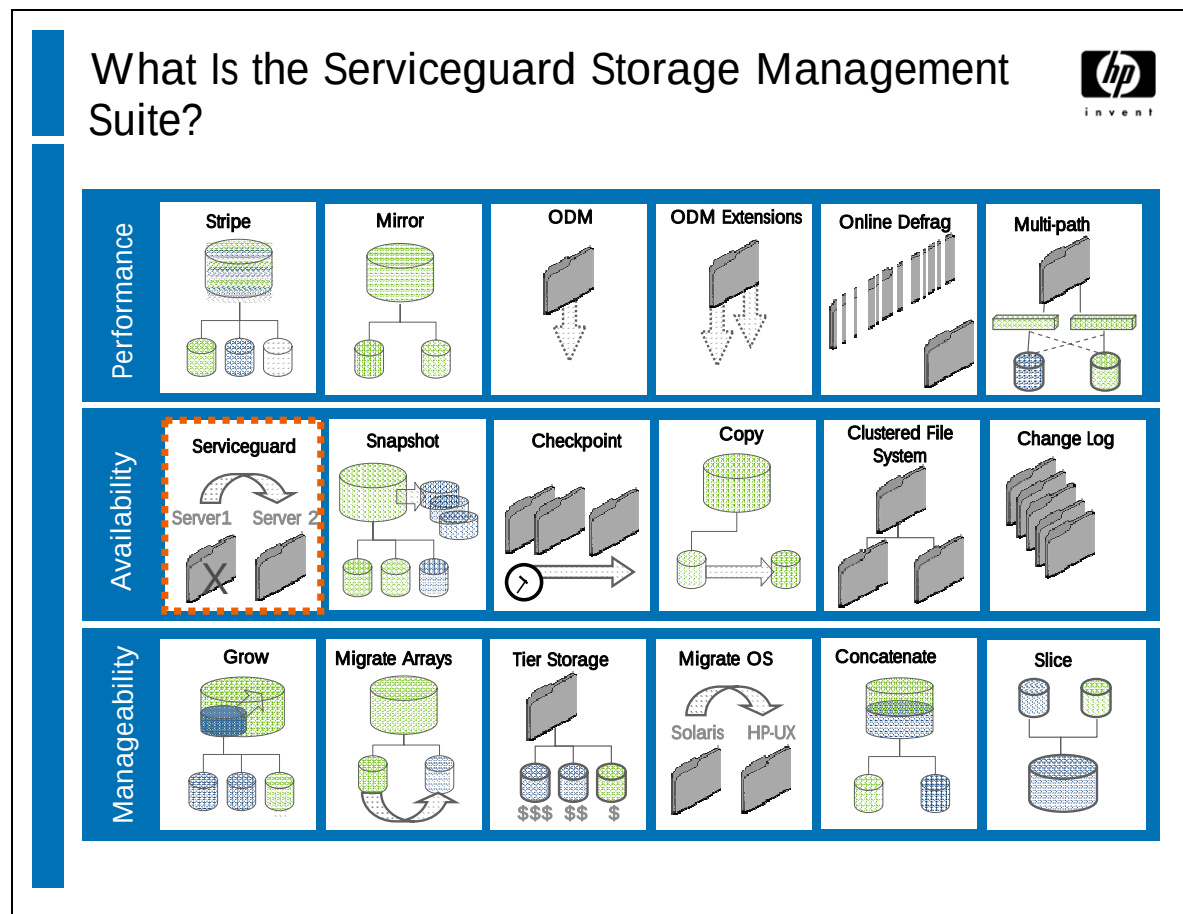


Student Notes

These are some of the typical customer pain points in managing and running a database.

- One main point to highlight is the manageability aspect. Managing Oracle in raw volumes is very complex. We are seeing a lot of customers actually using a file system to manage their data. However, they sacrifice some of the performance that can be achieved with raw. With Serviceguard Storage Mgmt Suite, we can help offer improved performance in a file system-based environment with a capability called Oracle Disk Manager (ODM)
- Database recovery is taking too long – Serviceguard Storage Mgmt Suite (SG SMS) offers capabilities like checkpoint to expedite the recovery time
- Protecting against I/O path failure is achieved with Dynamic Multi-pathing for redundant paths between server and storage.

F-3. SLIDE: What Is the Serviceguard Storage Management Suite?



Student Notes

This slide shows the various components available in different products within the suite spanning performance, availability and manageability. (Note: these are all the parts from the Veritas Storage Foundation offering by Symantec). The only difference by getting this with SG SMS is Serviceguard which replaces the HA clustering from Veritas/Symantec (Veritas Cluster Server) as highlighted above. The value of purchasing the SG SMS solution stack from HP is that you can get the additional benefits like the integration with HP Virtualization technologies for instance as well as the robust disaster tolerant solutions.

F-4. SLIDE: HP Serviceguard Storage Management Suite



HP Serviceguard Storage Management Suite

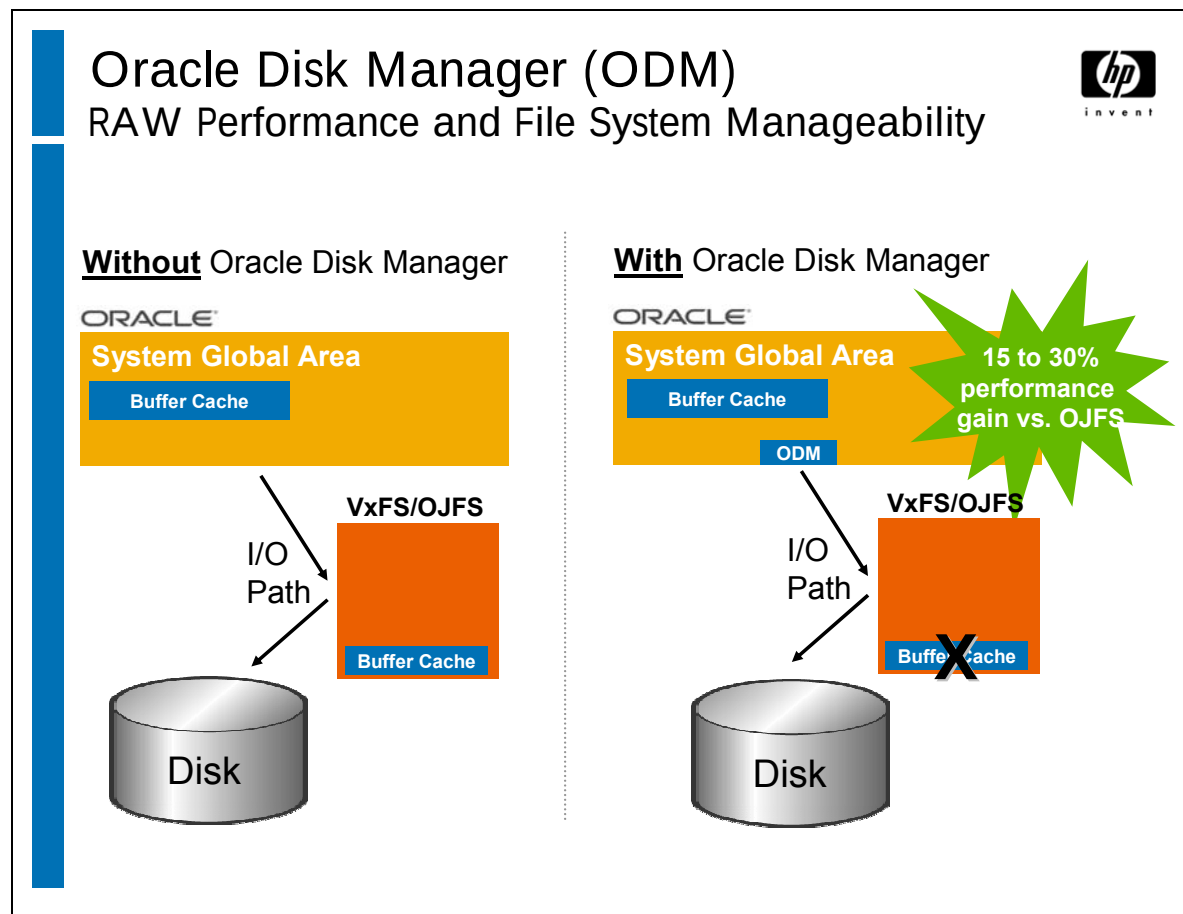
Features

Volume Management and File System	Online storage reconfiguration, online volume and file system creation and resizing, and database <i>tablespace</i> growth Hot relocation of failed redundant storage
Cluster File System	A CFS provides excellent I/O performance and simplifies the installation and ongoing management of a RAC database
Oracle Disk Manager (ODM)	Allows near raw performance within a file system for an Oracle database
Flashsnap	Let DBAs easily establish database clone, a duplicate database on a secondary host for off-host processing
Multi-Volume File System (MVFS)	Provides a single file system on multiple underlying volumes with different performance and availability characteristics
Dynamic Storage Tiers (formerly QoSS)	Allows single FS to span two tiers of disk (high speed and lower speed) "Online archiving" solution Use the most cost effective storage for needs
Portable Data Containers (PDCs)	Common on-disk data format across OSs Allows for easier OS migration (i.e. Solaris → HP-UX)
Dynamic Multi-Pathing (DMP)	Allows for redundant paths to storage to be utilized, protecting against link outages and improves performance with active-active load balancing
Storage Mapping	Quickly diagnose performance bottlenecks from table space down to disk

Student Notes

Feature benefits of some of the features from the previous slide.

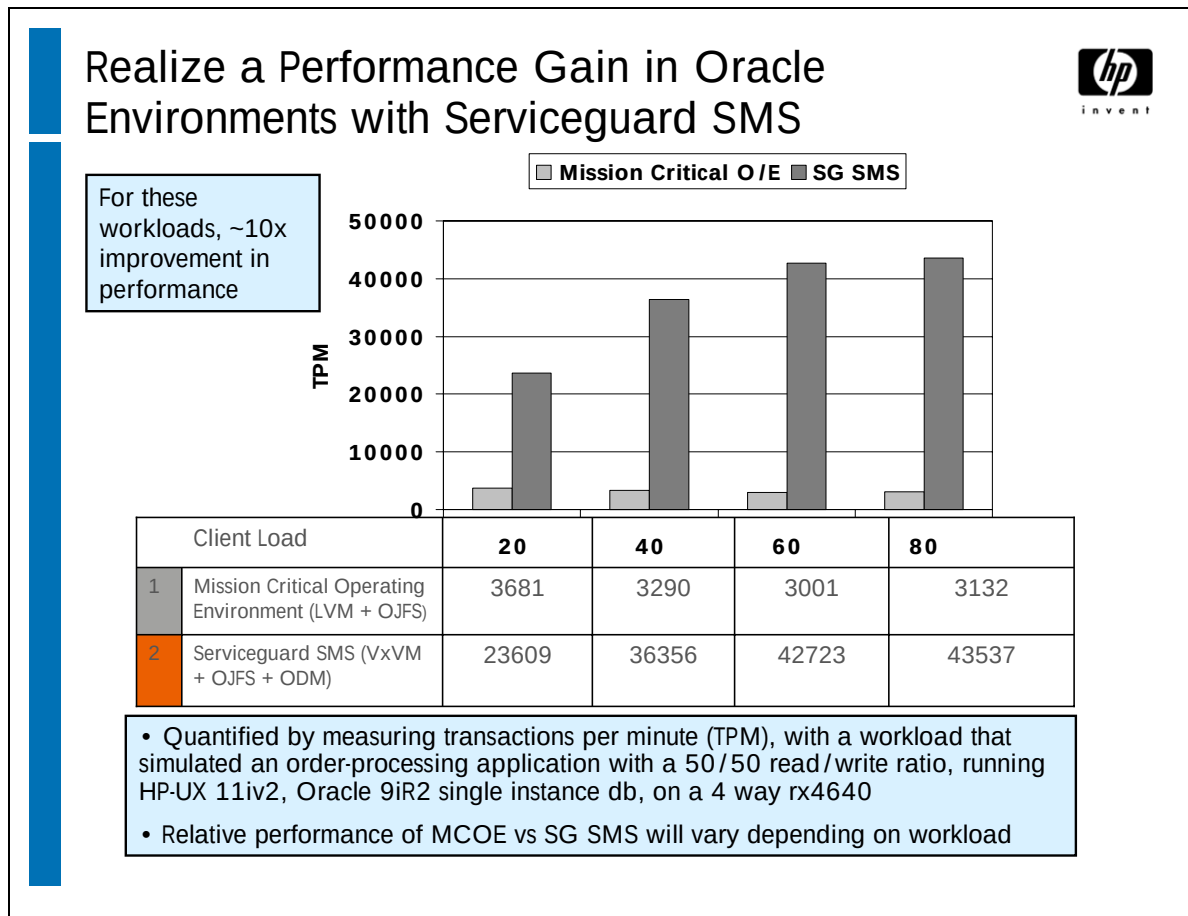
F-5. SLIDE: Oracle Disk Manager (ODM) RAW Performance and File System Manageability



Student Notes

Symantec has developed a feature specialized for Oracle environments called the Oracle Disk Manager. This slide shows on the left a VxFS file-based environment not using ODM and on the right an environment that is using ODM. Through this integration, you can bypass the cache of the file system and rely on the cache of Oracle. What this results in is the best of both worlds, you get the manageability of a file system while getting close to raw performance without getting the duplicate hit of the i/o path to the buffer cache. In benchmark testing by Symantec, they have seen performance gains of 15-30% with ODM versus just running a file system like OJFS that does not use ODM. It is important to note that running Oracle on raw volumes will still give you the best performance. We're seeing 8-15% better performance in raw vs. file system. But, as I've pointed out before, managing raw volumes is much more complex than managing a file system. So, this is a great feature to consider if you are want to lower your administrative and operating cost by using a file system without losing too much performance.

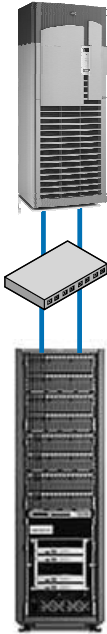
F-6. SLIDE: Realize a Performance Gain in Oracle Environments with Serviceguard SMS



Student Notes

HP SG SMS R&D team conducted its own performance test comparing the use of our own HP Logical Volume Manager (LVM) with OJFS (Veritas File System) versus the SG SMS capabilities that include the Veritas File System, Volume Manager and ODM. You can see up to a 10x performance improvement with the capabilities offered with ODM. Additional performance tests include comparison with raw with LVM and VxVM which still provides the best performance over file system (8-15% better). Again, with ODM you can get near raw performance.

F-7. SLIDE: Dynamic Multi-Pathing (DMP) Protect Against I/O Path Failure



The diagram illustrates a server connected to a storage array via a switch. Two blue lines represent redundant I/O paths between the server and the storage array, passing through the switch. The server is a tall rack unit, and the storage array is a shorter rack unit. The switch is a small square unit in the middle.

Dynamic Multi-Pathing (DMP)

Protect Against I/O Path Failure



Redundant I/O paths between HBA and storage is a best practice for availability

- Multi-pathing software automates I/O path failover
- Avoid downtime from switch/HBA/port upgrades

DMP advantage

- No addition cost – available in the Serviceguard Storage Management Suite
- Flexible storage array support (HP & EMC)
- Advanced load balancing algorithms for performance

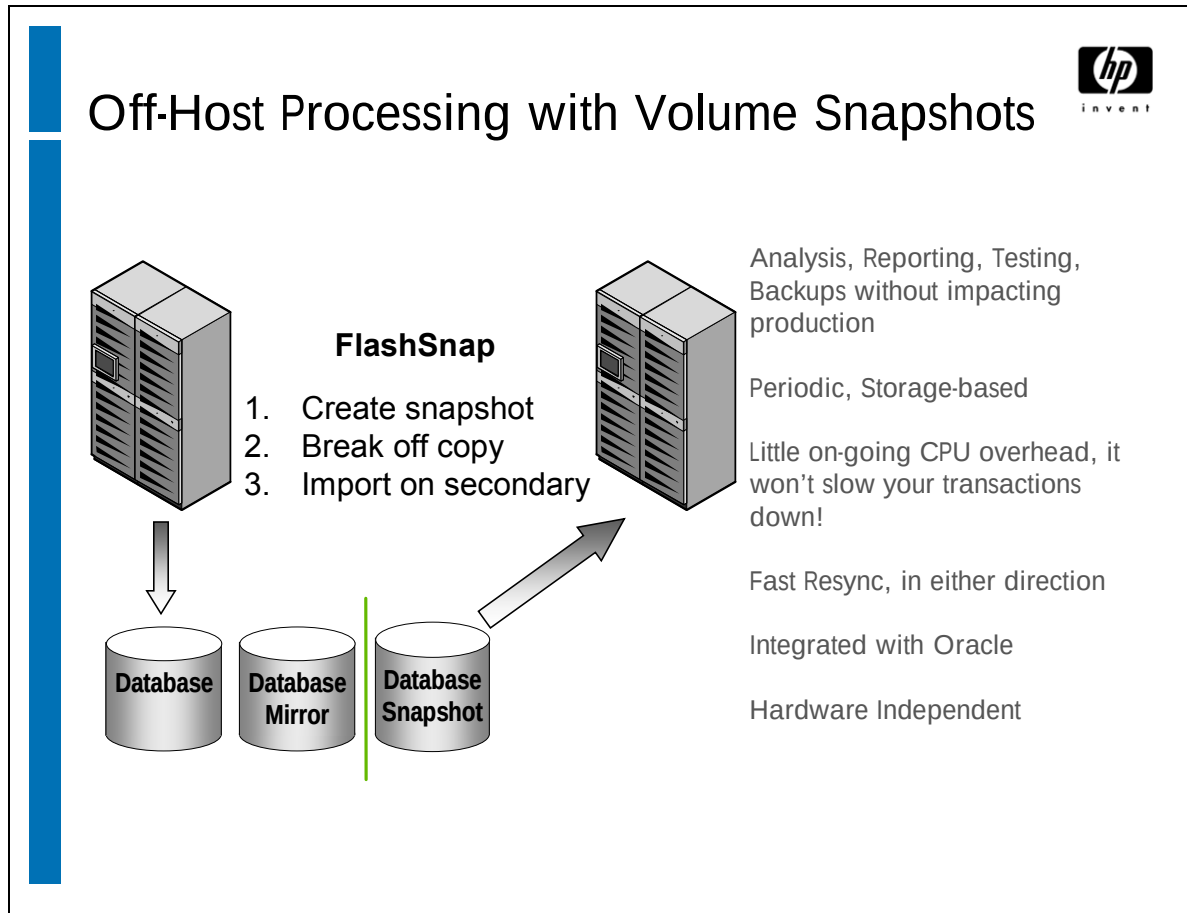
Student Notes

In addition to protecting the application and DB there are a number of other key considerations. Redundant i/o path between the HBA and server and underlying storage pool is now typically a best practice in most IT organization for availability. Multi-pathing software allows you to automate I/O path failover in the event of an outage. Multi-pathing software is a great way to reduce downtime if you're doing something like a switch/HBA or port upgrade, for instance.

The DMP capability is available in the SG SMS so there's no additional cost, flexible array support with HP and EMC.

In addition to having the capability of failover of i/o paths, it also has advanced load balancing capabilities to ensure you have optimized the performance of your i/o paths from your server down to the storage.

F-8. SLIDE: Off-Host Processing with Volume Snapshots



Student Notes

Another area that you can realize benefits of the SG SMS is with a capability called FlashSnap. This is where you can move your activities like analysis, reporting, testing, backups off of the primary DB host and onto a secondary host. (This FlashSnap feature is enabled through a capability in the volume manager). So, the main capability of this feature is to take off from the primary host some of the impact in performance related to running some of these activities. The way it works is that a mirror image of the database is taken and using the FlashSnap capability a copy (or snapshot of the database) is broken off and can be remounted and imported to a secondary host. This snapshot capability can support up to 32 concurrent copies of the data. You can mount these on different hosts, and changes on these copies can be synchronized back either from the primary copy to the secondary or the secondary copy back to the primary. (This Flashsnap capability is hardware independent meaning that you can run it on any supported HP storage including XP and EMC). This feature also allows you to move these point in time copies to multiple storage sub-systems so your i/o load is shared throughout your environment.

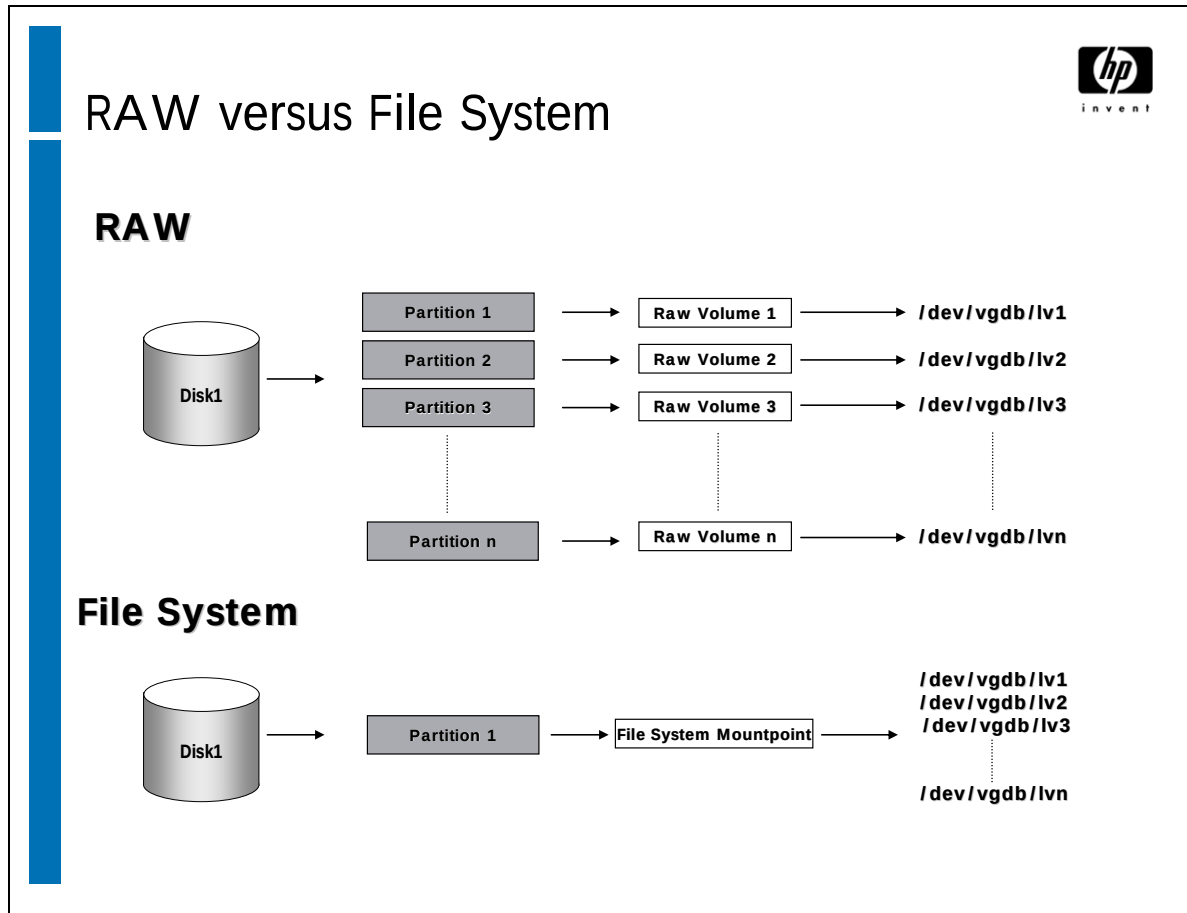
Appendix F

HP Serviceguard Storage Management Suite for HP-UX 11i

Other facts:

- Today volumes can only reside in the diskgroups they were created in. If you decide to reorganize the diskgroup, there is no easy method that is available. The Dynamic Group Split and Join feature of FlashSnap addresses this. This technology enables users to split a mirror from a server and join the disk group to another server. This will allow users to deport a split mirror to another host to offload operations such as backups or decision support from the production server. The disk group can then be imported back into the production machine.
- FlashSnap supports up to 32 (the original volume plus 31) concurrent copies of data. These may be mounted on different hosts and all changes to all copies of the data are tracked so that once you require a data copy to be brought up to date, it may be resynchronized to the primary or production copy of the data quickly with little overhead at the disk or the host.

F-9. SLIDE: RAW versus File System



Student Notes

It's important to remember that raw devices cannot be dynamically expanded. Therefore, the table spaces mapped to these devices cannot be expanded dynamically while the Oracle database remains online. Resizing raw partitions is difficult, and a backup version of the table must be written back to the larger partition once it is created offline.

Also, managing raw volumes is a lot more complex. Each file or database is required to have its own raw volume and managing it requires a lot more time relative to ease of managing a single partition using a file system.

F-10. SLIDE: Common Management Activities



Common Management Activities

Function	RAW	FS
Auto-extend	Hard	Easy
Performance	Best	Good
Storage Pooling	Hard	Easy
Block Level Incremental B/U	No	Yes
Isolate performance	Easy	Medium
Migrate to new storage	Medium	Easy
Grow	Medium	Easy
Overall manageability	Hard	Easy

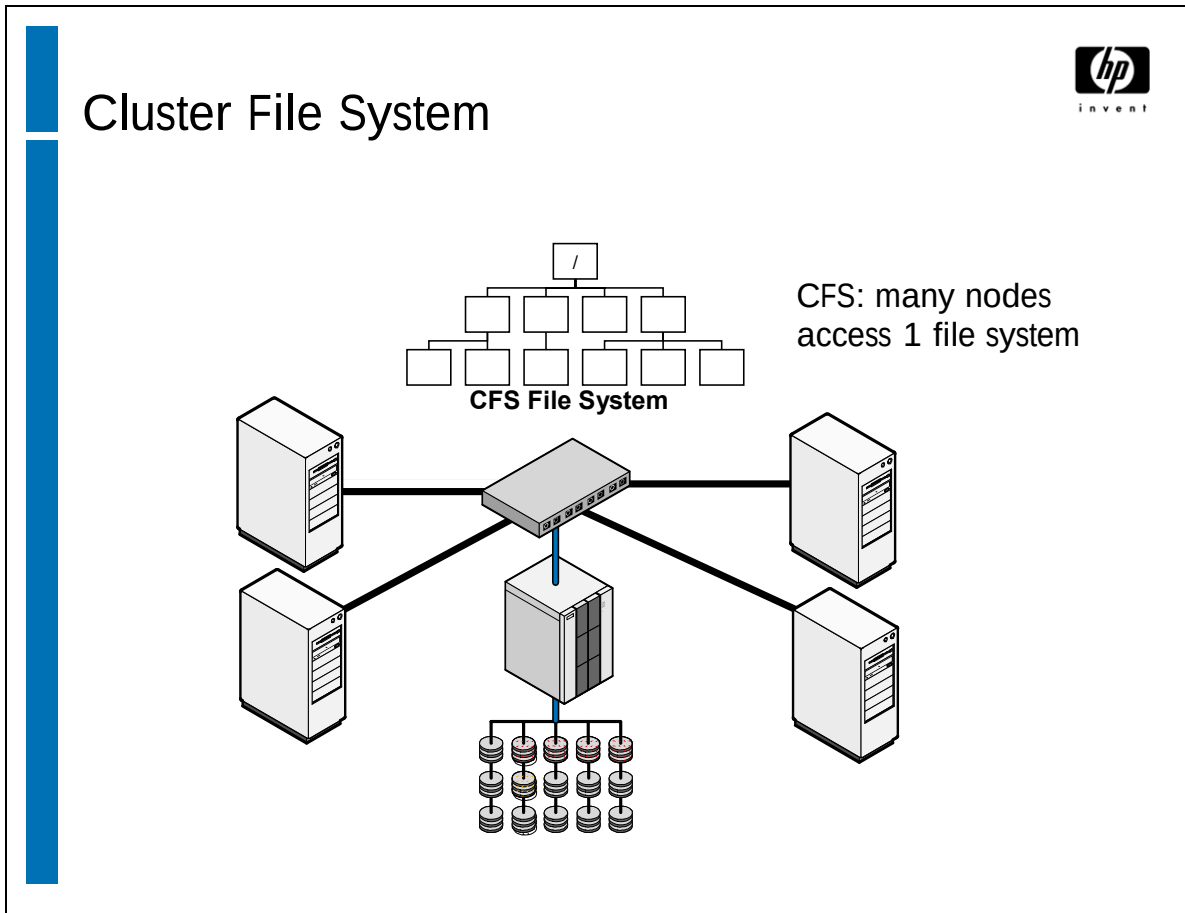
Student Notes

Many common day-to-day management activities are significantly more difficult with raw. For instance, auto-extend. What we mean here is that raw volumes can not be dynamically expanded, therefore, the table spaces that are mapped to these volumes can not be expanded dynamically while the Oracle DB is online. Resizing of raw partitions is more difficult than with a file system.

These other common activities you see here such as storage pooling, BLIB, migration to new storage and overall manageability are much more difficult in an environment where raw volumes are used.

So, as you can see file systems make life much easier. Again with the Serviceguard Storage Management Suite you get the best of both worlds. You can all the benefits of a file system with these activities shown on this slide, plus the performance advantage of running near raw.

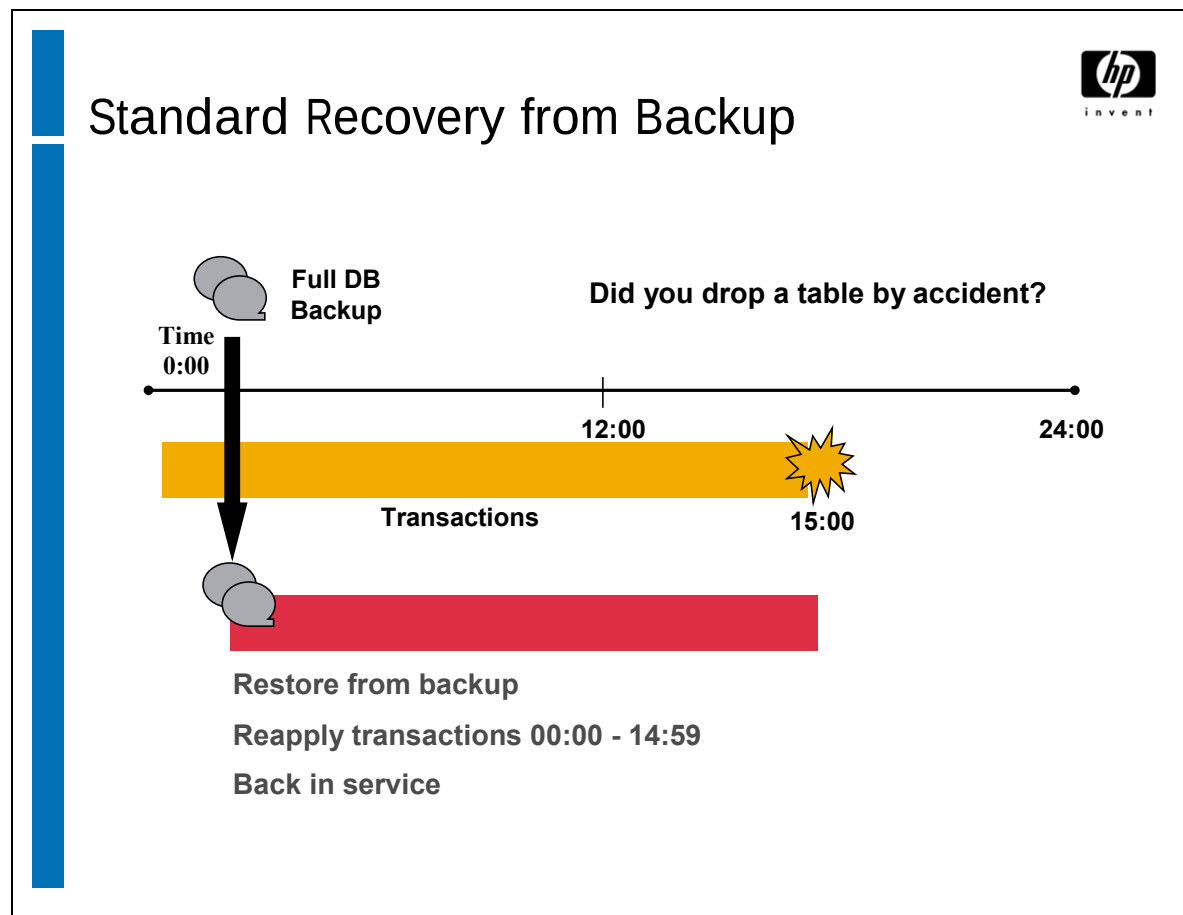
F-11. SLIDE: Cluster File System



Student Notes

With a Clustered File System, you now have many nodes being able to access a single file system offering improved manageability. This means that when an application writes to a file, any subsequent application that reads from the same area of the file will retrieve the new data. An simple example of this is with an Apache web server utilizing a CFS. If you make a change through one of your nodes to a web page, the other web servers connected to the cluster file system will read the same data and be able to make the change automatically without the user having to modify the web page changes on each web server.

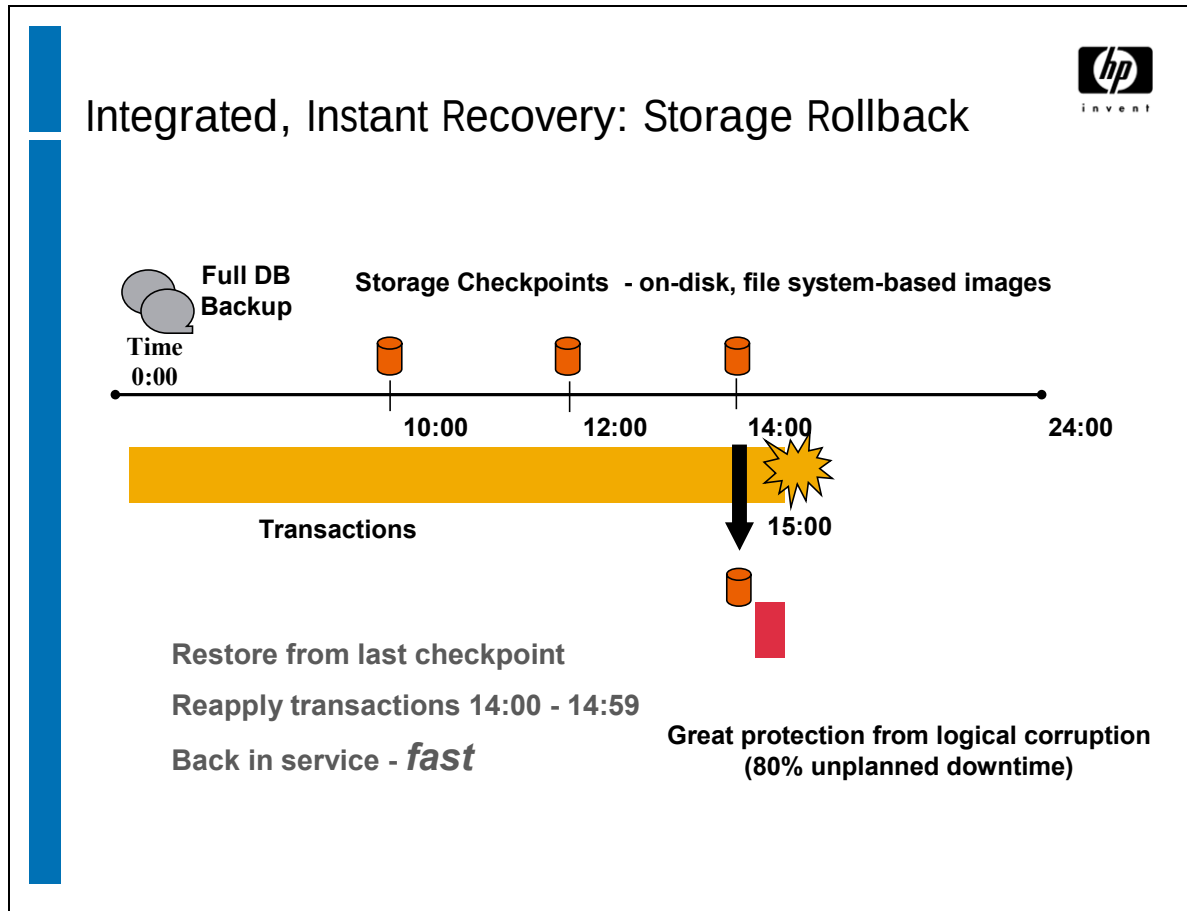
F-12. SLIDE: Standard Recovery from Backup



Student Notes

Let's take a look at a standard recovery from back-up scenario. In a database environment it is easy for accidents to happen. If you only have tape back-up for protection, then you're at risk for a long outage. This particular slide shows an outage that takes place at 3PM. So, for example an admin accidentally drops a table space. In this scenario you would have to go back to the nightly back-up to restore the entire transaction. This entire process both the tape back-up and restore, is very time-consuming typically. It's possible that this process can take up to a few hours to recover which is not an acceptable recovery window for many mission critical applications. This can be a common occurrence where an admin makes a mistake or is testing something out where it was not an appropriate action to take...

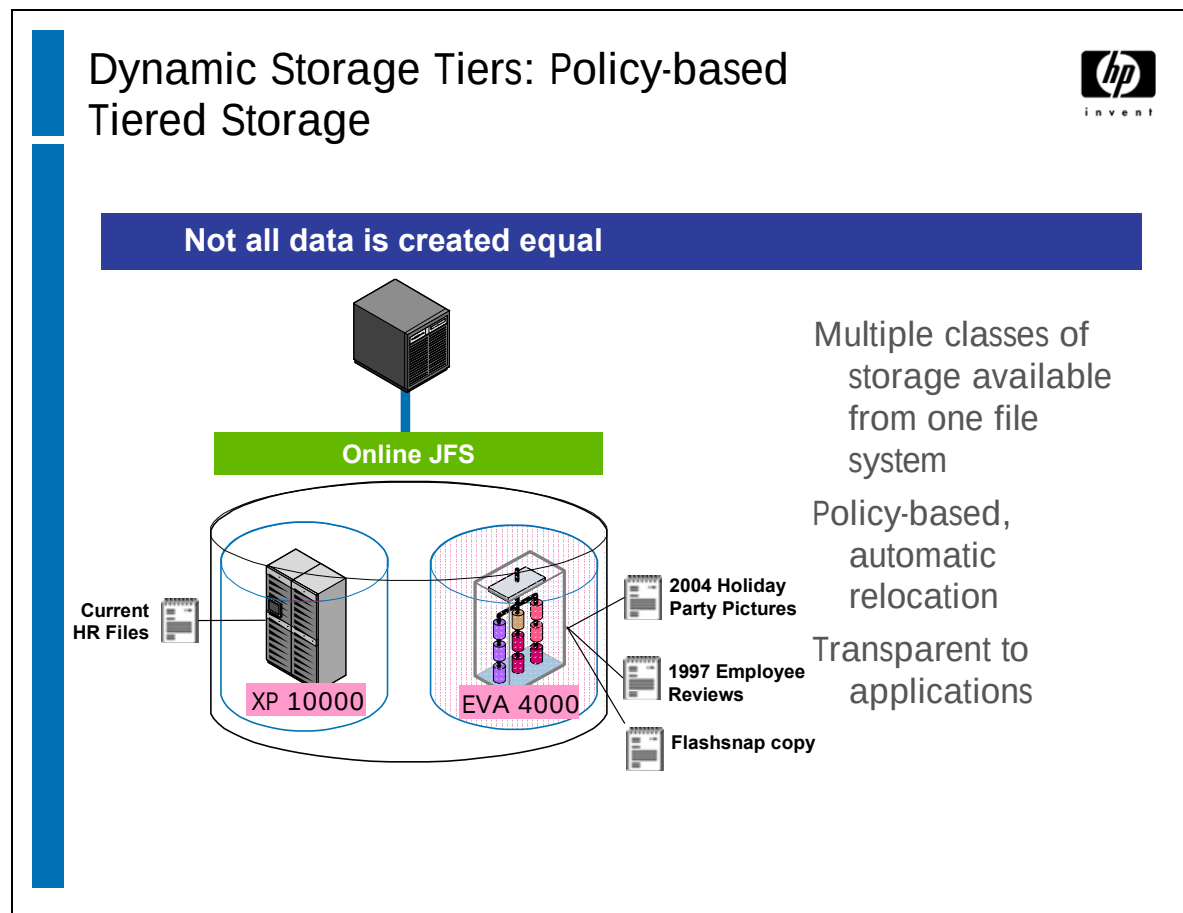
F-13. SLIDE: Integrated, Instant Recovery: Storage Rollback



Student Notes

To solve this issue, the Storage checkpoint and rollback feature in the SG SMS, greatly simplifies the recovery process. Storage checkpoints is a file-system based image of the data in a particular point in time. So, on this slide we illustrate the storage checkpoints at various time of day 10, 12, 2PM. There's a scheduler built in so you can set the checkpoints at whatever time interval you prefer. So, using the same scenario as in the previous slide, you can restore from the last checkpoint and reapply a fewer number of transactions and be back in service much quicker. Typically, what you'll experience in recovery time is in minutes using checkpoints rather than hours. Checkpoints is a great way to protect from logical corruption for both planned and unplanned downtime. Great capability for a DBA who may be testing out a command and is unsure of the impact it may have. They can take a checkpoint without having to coordinate with the storage or server admin. If it doesn't work, the DBA can simply roll-back with minimal impact. All this the DBA can control themselves.

F-14. SLIDE: Dynamic Storage Tiers: Policy-based Tiered Storage



Student Notes

With this feature, you can define certain classes of storage device, and set policies dictating what files can reside where. The policies automatically relocate files based on those policies, and the relocation is totally transparent to applications. Under the covers, to implement this feature, we have enabled what we call "Multi Volume File System" which allows for multiple volumes to house the mounted file system. Each volume can reside on different storage, allowing you to match data to storage type or RAID layout.

You can also set policies which state that all the less mission critical or historical data go to less expensive storage, while your mission critical data must reside on the most expensive, highly performing storage.

F-15. SLIDE: For More Information



For More Information

HP Serviceguard Storage Management Suite
www.hp.com/go/sms

HP Serviceguard High Availability and Disaster Tolerant
Solutions for HP-UX 11i and Linux
www.hp.com/go/serviceguard
www.hp.com/go/sglx

Student Notes

