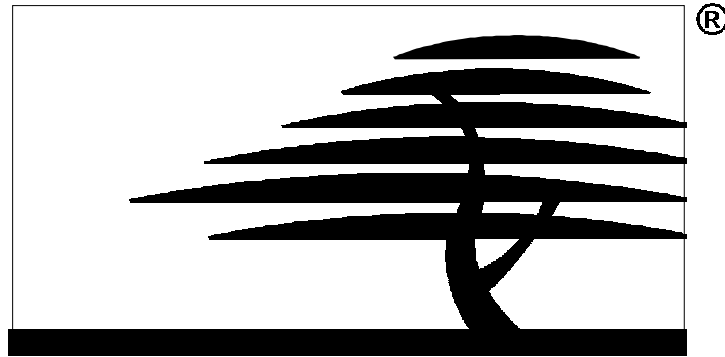




SCO

UnixWare[®] 7 NonStop[®] Clusters[®] Concepts

Student Guide

Course Release: 1.0

For Software: UnixWare[®] 7 NonStop[®] Clus-
ters Version 7.1.1

©2000 The Santa Cruz Operation, Inc. All rights reserved.

This publication is protected under copyright laws and international treaties.

Information in this document is subject to change without notice and does not represent a commitment on the part of The Santa Cruz Operation, Inc.

SCO, The Santa Cruz Operation, the SCO logos, VP/ix, SCO OpenServer, UnixWare, dbXtra, Panner, SCO Global Access, SCO MPX, SCO Doctor, SCO Doctor for Networks, SCO Doctor Lite, SCO Premier Motif, SCO Visual Tcl, SCO VisionFS, SCO ToolWare, SCO TermLite, SCO Vision 97, SCO CIFS Bridge, SCO WebTop, The Internet Way of Computing, and IWoC are trademarks or registered trademarks of The Santa Cruz Operation, Inc. in the USA and other countries. IXI, X.desktop, Deskworks, Wintif, IXI Panorama, Deskterm, and IXI Desktop are trademarks or registered trademarks of IXI Limited, a subsidiary of The Santa Cruz Operation, Inc. Visionware, Devkit.Vision, Esprit, Kodon, Super.Vision, Visionware Super.Vision, Term.Vision, Visionware Direction, the Visionware logo, Visionware SQL-Retriever, and XVision are registered trademarks of Visionware Limited, a subsidiary of The Santa Cruz Operation, Inc. SuperVision, TermVision, X-Visionware, SQL-Retriever, and Vision Builder are trademarks of Visionware Limited. UNIX is a registered trademark of The Open Group in the United States and other countries. Cheyenne and ARCServe are registered trademarks of Cheyenne Software, Inc. Netscape, Netscape Navigator, Netscape Communications Server, Netscape Commerce Server, Netscape Proxy Server, Netscape FastTrack Server, Netscape Enterprise Server, Netscape SuiteSpot, Netscape Catalog Server, Netscape News Server, Netscape Mail Server, and Netscape Navigator Gold are trademarks of Netscape Communications Corporation. NFS was developed by Legent Corporation (formerly Lachman Associates, Inc.) based on LACHMAN SYSTEM V NFS. LACHMAN is a trademark of Legent Corporation. NFS is a trademark of Sun Microsystems, Inc. TCP/IP was developed by Legent Corporation (formerly Lachman Associates, Inc.) based on LACHMAN SYSTEM V STREAMS TCP, a joint development of Lachman Associates and Convergent Technologies. MPX was developed by Corollary, Inc. VP/ix is a product developed and licensed by Phoenix Technologies, Ltd/INTERACTIVE Systems Corporation. XRemote is a registered trademark of Network Computing Devices, Inc. Oracle is a registered trademark of Oracle Corporation, Redwood City, California. Java is a trademark of Sun Microsystems, Inc. Compaq, Tandem, Integrity, NonStop, NetFlex, ProLiant, ROMPaq, ServerNet, SmartStart, registered in United States Patent and Trademark Office. Intel and Pentium are registered trademarks, and Pentium II Xeon and Pentium III Xeon are trademarks of Intel Corporation. SCO, Veritas, VxFS, and VxVA are registered trademarks of Veritas Software Corporation. Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies. All other brand and product names are or may be trademarks of, and are used to identify products or services of, their respective owners.

The SCO software that accompanies this publication is commercial computer software and, together with any related documentation, is subject to the restrictions on US Government use as set forth below. If this procurement is for a DOD agency, the following DFAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the US Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

If this procurement is for a civilian government agency, this FAR Restricted Rights Legend applies:

RESTRICTED RIGHTS LEGEND: This computer software is submitted with restricted rights under Government Contract No. _____ (and Subcontract No. _____, if appropriate). It may not be used, reproduced, or disclosed by the Government except as provided in paragraph (g)(3)(i) of FAR Clause 52.227-14 alt III or as otherwise expressly stated in the contract. Contractor/Manufacturer is The Santa Cruz Operation, Inc., 400 Encinal Street, Santa Cruz, CA 95060.

Table of Contents

Lesson 1: Course Introduction	1-1
Introduction.....	1-3
Course goal and purpose.....	1-3
Course Lessons	1-5
Target audience	1-5
Course roadmap	1-5
<i>NSC Curriculum Hardware Supplement</i>	1-6
Student Materials	1-7
The <i>Student Guide</i>	1-7
Typographical Conventions	1-8
Lesson 2: Driving Forces and Clusters	2-1
The Demand for More Computing Resources.....	2-7
Faster and Less Expensive Hardware	2-9
More Reliable Computing Environments	2-11
The Monolithic Solutions	2-15
The NSC Solution.....	2-17
Lesson Summary.....	2-19
Answers to Exercise and Review Questions.....	2-21
Lesson 3: Key Cluster Concepts	3-1
Defining Clusters	3-7
Clustering definition and single system image (SSI).....	3-7
Features of SSI Clustering	3-9
Scalability	3-11
Manageability	3-13
<i>Nodes</i>	3-15
Server area network (SAN) and fabrics.....	3-17
Protected resources and components	3-18
Failover, fault zones, N+1, N+0.....	3-24
Load balancing.....	3-25

NSC Specific Terms and Components.....	3-27
SSI in NSC.....	3-27
Scalability and Availability in NSC.....	3-29
Manageability in NSC	3-31
Nodes in NSC	3-33
SAN options for NSC and the Virtual Interface Architecture (VIA).....	3-37
Protected storage methods in NSC	3-41
Load balancing in NSC.....	3-41
Physical and Virtual IP Network Addresses	3-43
Virtual UPS	3-45
Clusterizing.....	3-46
Lesson Summary.....	3-47
 Lesson 4: Cluster Comparisons	 4-1
Distributed Computing Solutions	4-6
IP-Spraying Solutions	4-7
High Availability “Clustering” Solutions	4-8
True Clustering Solutions	4-9
Comparison Summary	4-11
Lesson Summary.....	4-12
 Lesson 5: Designing an NSC Cluster	 5-1
Design Concepts	5-6
Vulnerability and single point of failure	5-6
Node design	5-8
Node format—tower or rack mount.....	5-12
Disk configuration	5-14
Electrical power and UPS	5-16
Case Studies.....	5-17
Small business or replicated site	5-17
Large storage, low processing.....	5-18
Low storage, high processing	5-19
Heavy processing, heavy I/O	5-20
Lesson Summary.....	5-21
Answers to Exercise and Review Questions.....	5-22

1

Course Introduction



Lesson 1

Course Introduction

1.1 Introduction

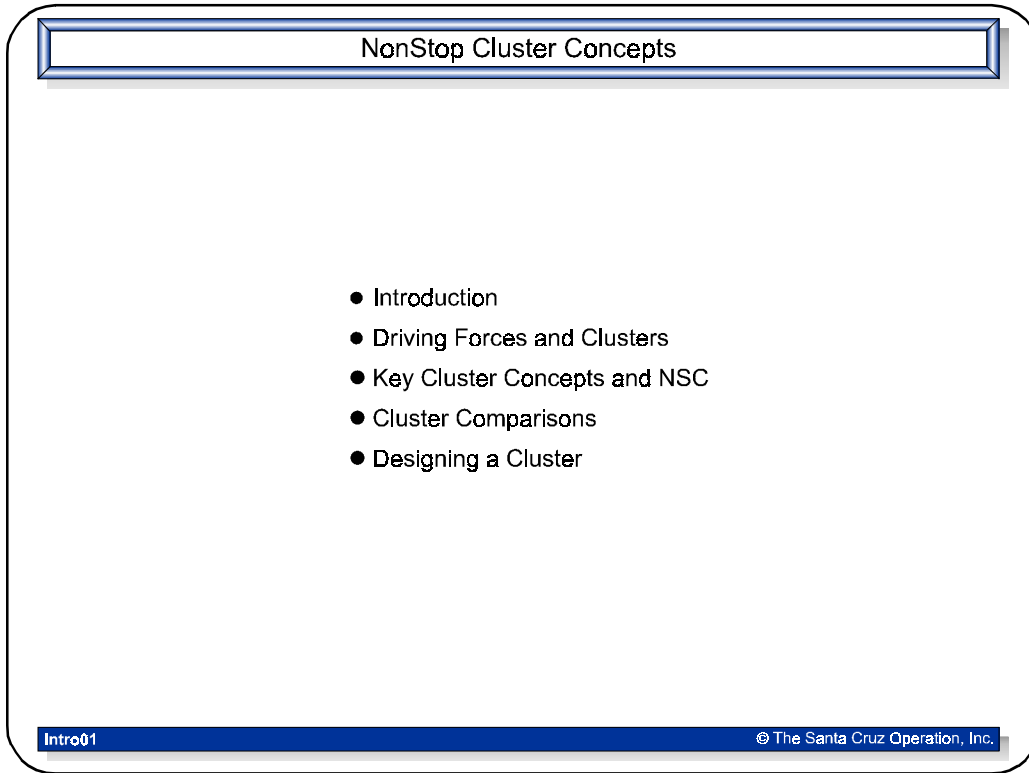
This ***UnixWare 7 NonStop Clustering Concepts*** course is designed for individuals who require a conceptual background on generic clustering and UnixWare 7 NonStop Clusters (NSC) features, components, and concepts. The target audience includes sales professionals who will sell NSC, technicians who will install and configure NSC, system administrators who will maintain NSC systems on a daily basis, and programmers who will develop cluster-aware applications.

1.1.1 Course goal and purpose

At the conclusion of this course, students should be able to:

- Explain the driving forces behind clusters
- Explain key clustering concepts and terms including *single point of failure*, *failover*, *nodes*, *node pairs*, *root nodes*, *dependent nodes*, *load balancing*, *protected storage*, *server area network*, and *single system image*
- Explain NSC features and capabilities
- List, compare, and contrast the various methods of protecting storage in NSC
- Differentiate NSC from other non-clustering solutions
- Differentiate NSC from other clustering solutions
- Design a cluster for small, medium, and large clients with various needs for failover, load balancing, and protected storage

AV 1-1

A presentation slide titled "NonStop Cluster Concepts" with a blue header bar. The slide contains a bulleted list of five topics: Introduction, Driving Forces and Clusters, Key Cluster Concepts and NSC, Cluster Comparisons, and Designing a Cluster. At the bottom, there is a blue footer bar with the text "Intro01" on the left and "© The Santa Cruz Operation, Inc." on the right.

NonStop Cluster Concepts

- Introduction
- Driving Forces and Clusters
- Key Cluster Concepts and NSC
- Cluster Comparisons
- Designing a Cluster

Intro01 © The Santa Cruz Operation, Inc.

1.1.2 Course Lessons

To achieve the course goals, this course contains the following lessons:

- Introduction (this lesson) — course goals, topics, lengths, and typographical conventions
- Driving Forces and Clustering — how market forces are driving demand for greater computing resources and how clustering answers those demands
- Key Cluster Concepts, NSC, and Cluster Components — cluster-specific terms, concepts, and components
- Cluster Comparisons — contrasting the SCO NonStop Cluster (NSC) solution with other “cluster” solutions
- Designing a Cluster — designing a cluster to meet specific criteria, including case studies

1.1.3 Target audience

This class is aimed at sales people, installation technicians, system administrators, support technicians, and developers who need to become familiar with clusters in general and specifically UnixWare 7 NonStop Clusters.

A general, high-level background in computers is a necessary prerequisite for this course. However, prior knowledge of a UNIX or any SCO operating system is not required.

1.1.4 Course roadmap

UnixWare 7 NonStop Clustering Concepts is the first course in a series on UnixWare 7 NonStop Clusters. The other courses are:

- ***UnixWare® 7 NonStop Clusters® Installation***
- ***UnixWare® 7 NonStop Clusters® Administration***

1.1.5 NSC Curriculum Hardware Supplement

The NSC product is constantly evolving. New hardware system vendors are announcing support for NSC, new server area network, and protected storage options are becoming available regularly.

The **NSC Curriculum Hardware Supplement** contains information. The supplement has one chapter per certified vendor. Each chapter contains the same relative information. For example, each chapter contains the current list of computers the vendor supports for use as nodes in a cluster.

1.2 Student Materials

This student guide is yours to keep and write notes in. Your instructor might have other reference materials in the classroom for your use while attending this class. Please do not take or write in these materials as they will be needed again for later classes.

1.2.1 The *Student Guide*

The *Student Guide* contains:

- Outline of presentation
- Note-taking space
- Copies of all transparencies

1.3 Typographical Conventions

This publication presents commands, filenames, keystrokes, and other special elements as shown here:

lp or lp(1)	Commands, device drivers, programs, and utilities (names, icons, or windows); the number in parentheses indicates the reference manual section in which the command, driver, program, or utility is documented
<i>/new/client.list</i>	Files, directories, desktops (names, icons, or windows), system, network, and printer names
root	User names
<i>arg</i> or <i>arg</i>	Placeholders (replace with appropriate name or value)
[<i>arg</i>]	Surrounds an <i>option</i> or <i>cmdarg</i> that is not mandatory
<i>arg ...</i>	Indicates multiple occurrences of the <i>option</i> or <i>cmdarg</i>
\$	Command-line prompt for commands executable from the Bourne shell by any user
#	Command-line prompt for commands only executable by the root user from the root account
	Keyboard keys
 	Press  and  keys simultaneously
  	Press all three keys,  ,  , and  , simultaneously
 d	Press  key. With  still depressed, press  .
 0240	Press  key. With  still depressed, press and release in sequence  ,  ,  , and  .
 Q + i	Press  key. With  still depressed, press  . Release both keys and press  .
 Q + I	Press  key. With  still depressed, press  . Release both keys and press   .
Exit program?	System output (prompts, messages)

tar tv	User input, such as on a command line
“Description”	Field names or column headings (on screen or in database)
Cancel	Button names
Edit	Menu names
Copy	Menu items
File ⇨ Find ⇨ Text	Sequences of menus and menu items
open or open(2)	Library routines, system calls, kernel functions, C keywords; the number in parentheses indicates the reference manual section in which the file is documented
\$HOME	Environment or shell variables
SIGHUP	Named constants, signals, and kernel parameters
“adm3a”	Data values
<i>employees</i>	Database names and tables
buf	C program structures
b_b.errno	Structure members



2

Driving Forces and Clusters



Lesson 2

Driving Forces and Clusters

Introduction

Current market forces are calling for faster, less-expensive, and more reliable solutions to increasing computing needs. This is true in almost every market segment, from small to medium businesses (SMBs), to the Fortune 100 (enterprise) companies.

This lesson examines these market forces, the solutions being put forth to address them, and how NonStop Clusters fits as a solution.

Objectives

By the end of this lesson, you will be able to:

- Describe the market forces that lead to the development and adoption of clusters
- Define clustering terms, such as *cluster*, *monolithic solution*, *SMP*, *NUMA*, and *ccNUMA*

AV 2-1

The slide is titled "Driving Forces and Clusters" and contains a bulleted list of five points. The footer includes the text "Driving01" and "© The Santa Cruz Operation, Inc.".

Driving Forces and Clusters

- The Need for More Computing Resources
- Faster and Less Expensive Hardware
- More Reliable Computing Environments
- The Monolithic Solutions
- The Cluster Solution

Driving01 © The Santa Cruz Operation, Inc.

Topics

In order to meet these objectives, this lesson discusses:

- The Need for More Computing Resources — modern business dynamics that are creating the demand for greater resources, such as CPU, RAM, disk drive space, and disk drive access
- Faster and Less Expensive Hardware — how improvements in hardware technologies influence the market
- More Reliable Computing Environments — increasing demand for reliability in computing environments
- The Monolithic Solutions — extensions to the traditional computing solution
- The Cluster Solution — how clustering, and specifically NSC clustering, improves on monolithic solutions

AV 2-2

Demand for More Computing Resources

- Sophisticated user interfaces
- Improved business management features
- More Reliable Computing Environments
- Fast response times
- 24X7 Availability

Driving02 © The Santa Cruz Operation, Inc.

2.1 The Demand for More Computing Resources

One force driving the need for clusters is the increased demand for computing resources. This need comes from many places in many forms.

- Software requires more processing power to support sophisticated user-friendly interfaces. This is especially true of graphics software, which demands more resources than character applications. Such modern software requires more processor power, more RAM, and more disk space than its previous counterparts.
- Managers expect more business management features from their software. This also requires more resources. For example, managers expect applications to communicate with each other, reducing the requirement to re-enter the same data in multiple applications.
- People using computers as part of their every day job expect fast response from their computer hardware and software. Researchers have found a sharp decrease in the time that users expect to wait for a computer response.
 - Slow response times increase labor costs and inefficiency in all aspects of a business.
 - Fast responses require more resources.
- The explosion in popularity of the Internet means vastly increased demand on web servers. Simultaneously, users expectations are rising. If a response is not *fast enough*, they click to another web site. Therefore, web servers must have enough resources to handle increasing loads, for both browsing and e-business.

Further, web servers are becoming an increasingly critical component of many businesses. E-commerce demands service around the world, around the clock. Failure of a web server at any time is a critical problem that can cause major losses.

AV 2-3

Faster and Less Expensive Hardware

- Rapid Increases in CPU Speeds
- Lower Computer Equipment Costs

Driving03 © The Santa Cruz Operation, Inc.

2.2 Faster and Less Expensive Hardware

Due to many factors, computers become faster and less expensive every day. This phenomenon, almost unparalleled in the industrial revolution, provides inexpensive commodity computer hardware in large quantities. *Moore's Law* states that CPU speeds double every 18 months. This has many implications:

- Since some companies consider purchases under \$2000 to be supplies, most desktops, and even some servers are supplies. This means that computers are now considered disposable within many organizations.
- Organizations can purchase high-performance computers at little cost. This creates increased demand which feeds economies of scale. This further reduces the cost of computers. Alternatively, you may say that more performance is available for the same amount of money, especially when adjusting for inflation.
- In the first quarter of 2000, CPUs reached speeds of 1000 MHz, also known as one Giga Hertz (1GHz). Some time in 2001, processors will have speeds of 2GHz.
- Availability of relatively cheap and very fast computers provides a fertile development environment for groups, or *clusters*, of PCs to work together to provide improved computational ability and higher reliability than can be achieved by single computers.

NSC takes advantage of the wide availability of faster and cheaper hardware to build systems using *off-the-shelf* components that have scalability, availability and performance features formerly limited to high-end computing platforms.

AV 2-4

Downtime has a Price

Examples of Average Cost of 1 Hour Downtime

- Financial Market - Brokerage \$6.45M
- Credit Card Sales Authorization 2.60M
- Media Pay-Per-View Server 150K
- Retail Home Catalog Sales 90K

Internal SCO Research

DrivIng04 © The Santa Cruz Operation, Inc.

2.3 More Reliable Computing Environments

Many businesses are satisfied with the performance of their computer system. However, they would like one that is available non-stop—24 hours a day, 7 days a week. The need for this level of availability has several sources and implications:

- As business rely more heavily on data and voice communications, data mining, databases, and decision support systems, the computers that run these services must be up and running at all times.
- With society depending on computers for everything from traffic signals to military defense, computer down time becomes more than an inconvenience. Highly-available systems are required to turn computer failures from emergencies into maintenance problems.
- Businesses and individuals are increasingly reliant on their computer systems. As people become more dependent on computerized systems, they become mission critical to the organizations using them. Consider this example:

The failure of a computer can create havoc for a resort management company that has a weekly turnover of hundreds of families in hundreds of beach cottages. The critical time for these companies is the weekend when the check-outs and check-ins occur. Often, computer maintenance agreements are not 7x24x365, they are Monday through Friday from 8 to 5. Downtime on Saturday morning is a disaster for these types of businesses.

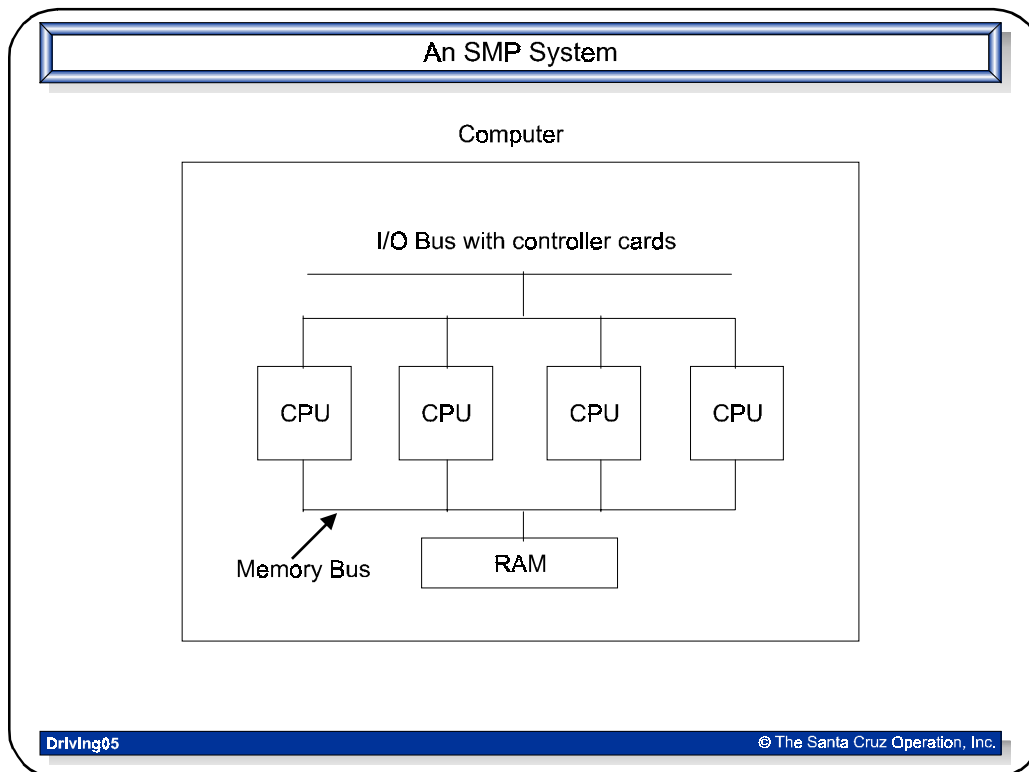
- The failure of a web server for an e-business means no business. For the customer seeking a web site, the failure of the web server leads to loss of service, dissatisfaction and possibly loss of business to a competitor who is only a click or two away.
- The failure of systems handling voice and data communications can cause great losses to the users of the systems as well as financial damage to the service provider



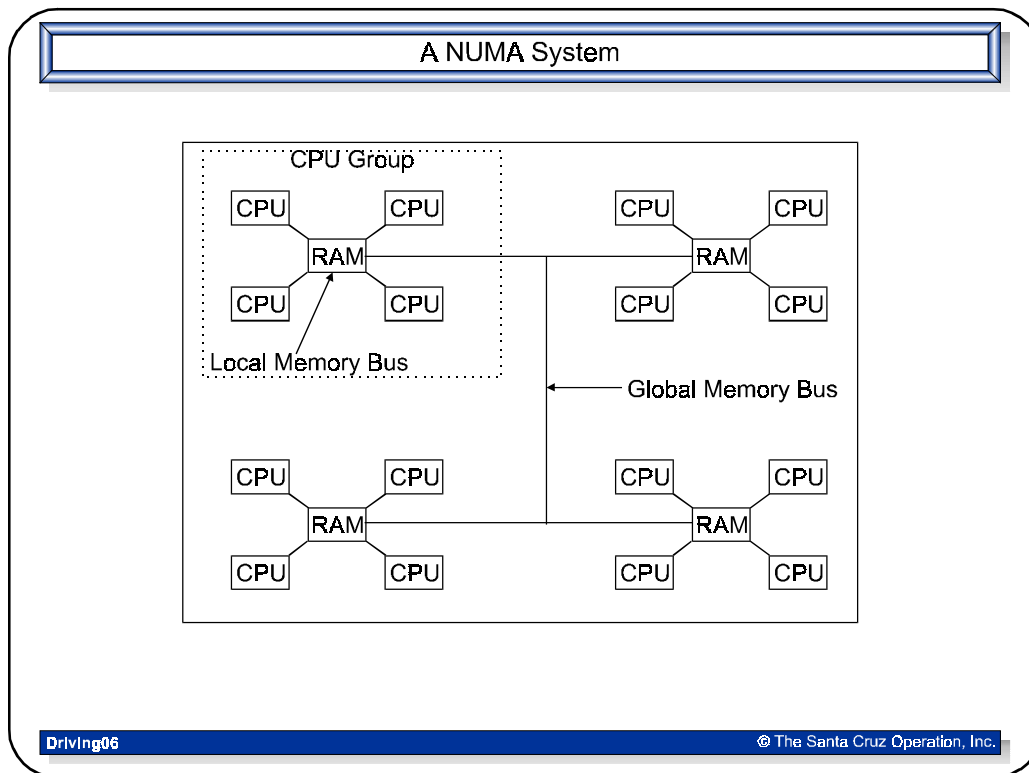
- In large businesses, some locations have no one with sufficient expertise to repair a inoperable server. For these locations, the reliability of the system is crucial. This is especially true where chain organizations, such as fast food stores or hotels, use computers in replicated sites. When was the last time you visited your favorite chain fast-food store and noticed a computer system administrator hanging around waiting for the system to crash?
- Downtime can have a substantial cost. AV 2-4 shows some sample loss figures for downtime.

NSC provides more reliable computing environments for all of these scenarios. It does so in ways that are discussed throughout the rest of this course.

AV 2-5



AV 2-6



2.4 The Monolithic Solutions

A *monolithic* solution for computing is one that employs a single computer which is typically large, complex, and proprietary. Such solutions:

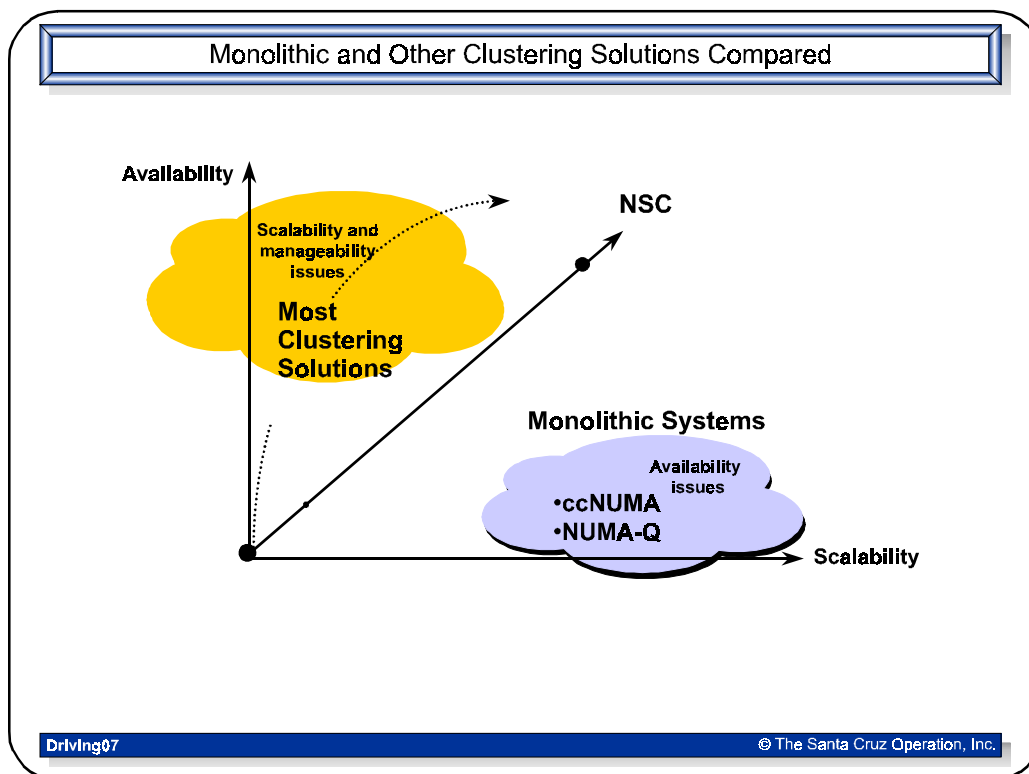
- Have been the traditional method of providing increased performance
- Always include adding some number of CPUs to the computer. The number of CPUs added can range from one on the low end, to thousands, in a *massively-parallel multiprocessor* solution.

While these solutions provide increased performance, the cost of the specialized hardware associated with this solution is high.

Some of the methods for building monolithic systems include SMP, NUMA, and ccNUMA:

- *Symmetric Multi Processing (SMP)* systems consist of multiple CPUs within a single computer. Multithreaded operating systems and applications spread the workload somewhat evenly over the CPUs to achieve higher throughput.
 - As the number of CPUs increases, the costs of computers that can accommodate the increase in CPUs grows at a non-linear rate. This is largely due to the expensive, high-speed interconnects needed to insure cache coherency between CPUs.
 - Due to bus bandwidth limitations and other hardware factors, there is a point at which scalability drops off as the number of CPUs increases.
 - Applications that are not multithreaded achieve no improvement in throughput when run on an SMP system.
- *Non Uniform Memory Access (NUMA)* systems use a number of small SMP modules, each of which has a few CPUs and some local RAM. The cost for any CPU within an SMP module to access their local RAM is the same. However, the cost for a CPU from one SMP module to access RAM in a different SMP module is greater than the local access. These costs are non-uniform, giving this solution its name.
- *ccNUMA* is a variation of NUMA. With ccNUMA, the CPU and other caches maintain coherency at all times.

AV 2-7



2.5 The NSC Solution

Clusters provide every aspect of high-performance, highly-available, reliable computing currently demanded by the market. With respect to:

Price	NSC runs on fast, low-cost, readily-available hardware. Incremental upgrades to an NSC cluster are possible with relatively low cost. NSC hardware components can be easily obtained and added to address computer emergencies quickly. The proprietary monolithic solutions are based on hardware that is low volume and therefore, not low cost or readily available.
Availability	Clusters use multiple computers, called <i>nodes</i>, to provide redundancy. If one node goes down, the cluster is still available to be used. Even between nodes, many resources are also provided in a redundant fashion, such as filesystem storage. Further, NSC provides the ability to quickly upgrade and expand a cluster. As computing resources run low, additional processors, additional nodes, and additional disk space can be added to a node <i>without shutting down the cluster</i>. This allows organizations the ability to easily expand the cluster without down time. Monolithic solutions may not offer all the online upgrade options of NSC.
Reliability	NSC uses the highly-tested and proven UnixWare 7 operating system as its foundation. This, combined with other well-tested clustering components, creates a system with a very degree of reliability. An estimated 99.99995% uptime rate can be achieved, translating to only 26 minutes of downtime per year, typically for upgrades and reboots. NSC provides a method to upgrade the operating system while the cluster is online. This means operating system upgrades become a matter of rebooting individual nodes while the cluster continues to function. Monolithic systems, even with redundant components, still have single points of failure, and lack a failover solution.

Scalability

Scalability refers to the ability of a system to expand and maintain a level of performance as it grows. Scalability is measured as the loss of potential performance as the system grows. Currently, NSC can scale to over 30 nodes with over 300 processors.

Future enhancements will exceed these limits. For extreme scalability, clusters are a better solution than monolithic systems.

Manageability

Manageability refers to the ease of administration offered by a system. The NSC design allows an administrator to manage the cluster as a single system and use familiar UNIX commands to manage the system. The skills needed to manage a single UNIX system are directly applicable to managing an NSC cluster.

Demand for NSC clusters is currently coming in two forms:

- Businesses with small systems that want greater reliability and availability. This includes many small business and large replicated sites.
- Those who need the increased scalability but are price sensitive.

2.6 Lesson Summary

This lesson described the forces driving cluster, specifically UnixWare 7 NonStop Clusters, to the forefront in computing environments. The points covered include:

- Faster and less expensive hardware
- The need for more computing resources
- More reliable computing environments
- Monolithic solutions
- The NSC solution

Review questions

1. List three market forces that are leading to the adoption of clustering.
2. What solution is available for high performance that does not provide clustering?
3. What is the difference between a NUMA and a SMP system?



2.7 Answers to Exercise and Review Questions

This section contains the answers to the review questions on page 2-19.

Review Question Answers

This section contains the answers to the review questions on page 2-19.

1. The market forces leading to the adoption of clustering were discussed throughout the course and include:
 - The need for higher availability
 - The need for more reliability
 - The need for more computing power including scalability beyond SMP systems.
2. Per Section 2.4, “The Monolithic Solutions,” on page 2-15, a solution that offers higher availability but does not provide the other features of clustering is SMP and NUMA systems.
3. The difference between a NUMA and an SMP system is that SMP systems use CPU-groups, which have faster access to local RAM and slower access to global RAM, whereas NUMA systems have all CPUs accessing the same RAM.



3

Key Cluster Concepts



Lesson 3

Key Cluster Concepts

Introduction

This lesson introduces you to the key concepts of both generic and UnixWare[®] 7 NonStop[®] Clusters, and the terms used to describe these concepts.

Objectives

By the end of this lesson, you will be able to:

- Explain key concepts of clusters, including *single point of failure*, *1+1* and *N+1* failover configurations, *N+0* and *N+1* load balancing configuration, *protected storage*, and *server area network (SAN)*
- Explain key concepts of NSC clusters including *single system image (SSI)*, *virtual UPSes*, *node pairs*, *active* and *potential root nodes*
- Define the components used in an NSC cluster

AV 3-1

The slide is titled "Key Cluster Concepts and NSC" in a blue header bar. It contains a bulleted list of two items: "Defining Clusters" and "NSC Specific Terms and Components". The footer is a blue bar containing the text "Key01" on the left and "© The Santa Cruz Operation, Inc." on the right.

Key Cluster Concepts and NSC

- Defining Clusters
- NSC Specific Terms and Components

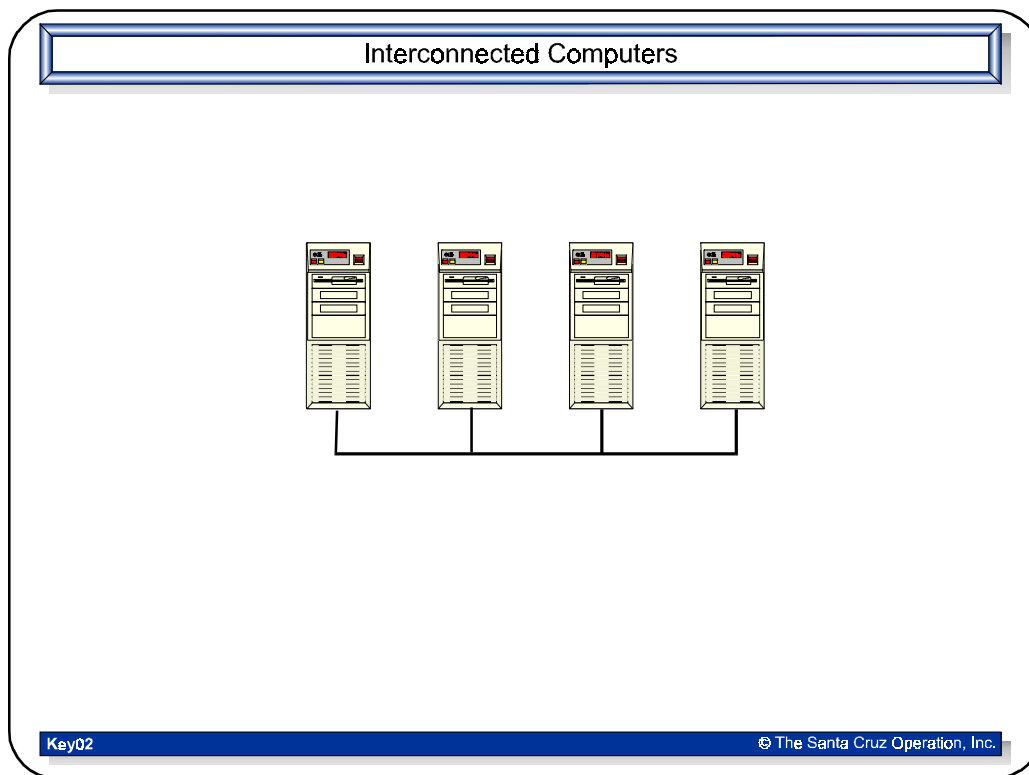
Key01 © The Santa Cruz Operation, Inc.

Topics

In order to achieve these objectives, this lesson discusses:

- Defining Clusters — defining terms and concepts used with clusters of all types
- NSC-specific Concepts and Components — defining terms and concepts specific to NSC. It also lists and describes the various components used in an NSC cluster.

AV 3-2



3.1 Defining Clusters

The term cluster, when associated with computers, has been used by various people to mean different things. However, all of the definitions have many things in common. This section describes the terms commonly used to describe clusters.

3.1.1 Clustering definition and single system image (SSI)

One of the best definitions of a cluster is found in the book *In Search of Clusters* by Gregory Pfister:

“A cluster is a type of parallel or distributed system that consists of a collection of two or more interconnected whole computers and is used as a single, unified computing resource.”

The key difference between a cluster and a collection of independent computers on a local area network (LAN), is that a cluster:

...is used as a single, unified computing resource.

AV 3-2 on page 3-6 illustrates a number of distinct, interconnected computers. They are either:

- A number of independent computers on a LAN
- A cluster—if they appear to users and administrators as a single computing resource.

In order for two or more computers to appear as a single computing resource, they must exhibit a property or behavior known as *single system image (SSI)*. For example, a UNIX cluster system that exhibits SSI:

- Shows a single process listing for all processes running on all computers, regardless of the computer on which the process listing command runs
- Shows a single set of devices, such as, printers, tapes, and disks, available cluster-wide
- Uses a single file tree, containing all filesystems on all disks attached to any computer in the cluster

A major advantage of SSI is that administrative effort of a cluster does not increase substantially as nodes are added. Management overhead for new nodes only adds incrementally to the administrative load.

Many clustering products offer some components of SSI. Others do not support SSI at all. NSC offers a full set of SSI capabilities. It is this full SSI capability that sets NSC apart from other clustering solutions.

3.1.2 Features of SSI Clustering

Clustering was conceived to address these issues:

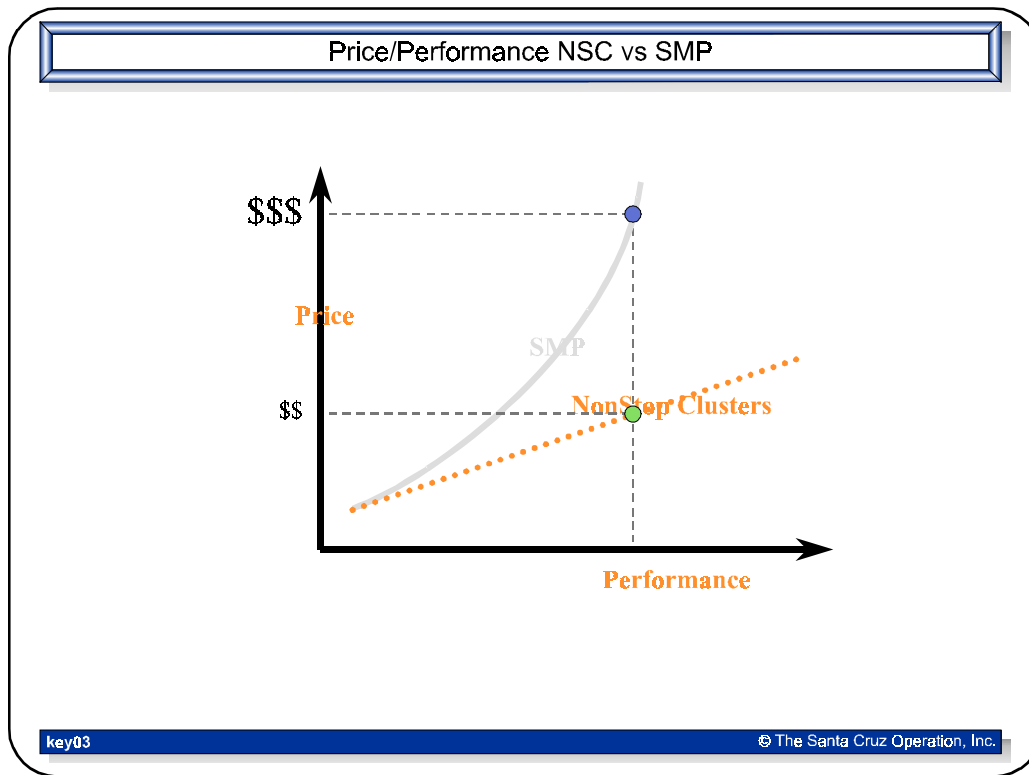
Reliability and Availability If one component, or even an entire node, fails, the failure does not effect the rest of the nodes in the cluster. Rather, they continue to operate and remain available to the users of the cluster. Specific information about how availability is increased in a cluster are presented throughout this lesson.

The key to ensuring availability is redundancy. If every component has a redundant counterpart that can be used in the event of failure, then there is no *single point of failure* within the cluster.

Scalability and Performance

By adding computers to the cluster, the total processing power of the cluster increases.

AV 3-3



3.1.3 Scalability

Scalability is a factor in many areas of a cluster, including:

- Number of nodes — How does the aggregate performance of all nodes in a cluster increase as nodes are added to a cluster?
- Number of CPUs in a node — How does performance increase in a node or throughout the cluster as CPUs are added to a node?
- Number of controllers — How does performance increase in a node or throughout a cluster as more devices (for example multiport serial controllers or SCSI HBAs) are added to a node?
- Number of devices attached to a controller — How does performance increase or decrease on a controller, node, or throughout a cluster, as more devices are attached to a controller? For example, how is performance affected as more disks are attached to a SCSI HBA?

Scalability is reported as a percentage of performance maintained at a certain system size compared to the performance yield at a minimal size.

The most common reason for performance degradation as systems grow is the increased overhead in keeping track of the additional items.

Even if a cluster is highly-scalable, it must provide services to applications and administrators that can take advantage of scalability in a manageable way. For example:

- It becomes difficult to manage multiple nodes in a cluster if each node has its own user database that must be synchronized with every other node. This concept is discussed further in Section 3.1.4, “Manageability,” on page 3-13.
- If applications must be re-written to use special APIs, then *off the shelf* applications will not scale. In NSC, the entire UNIX paradigm is preserved. NSC offers, but does not require, the use of any special APIs to enable applications to benefit from increased scaling performance from the cluster.

AV 3-4

Features of Manageability

- Cluster-wide System Administration
- Transparent Cluster-wide Processes List, Migration (For Load Balancing)
- Applications, utilities, programmers, users and administrators need not be cluster aware.
- Use of standard single node tools

key04 © The Santa Cruz Operation, Inc.

3.1.4 Manageability

Manageability of a cluster refers to the ease of administering the system. Without *clusterized* management tools, the cluster is just a collection of computers that must be separately managed.

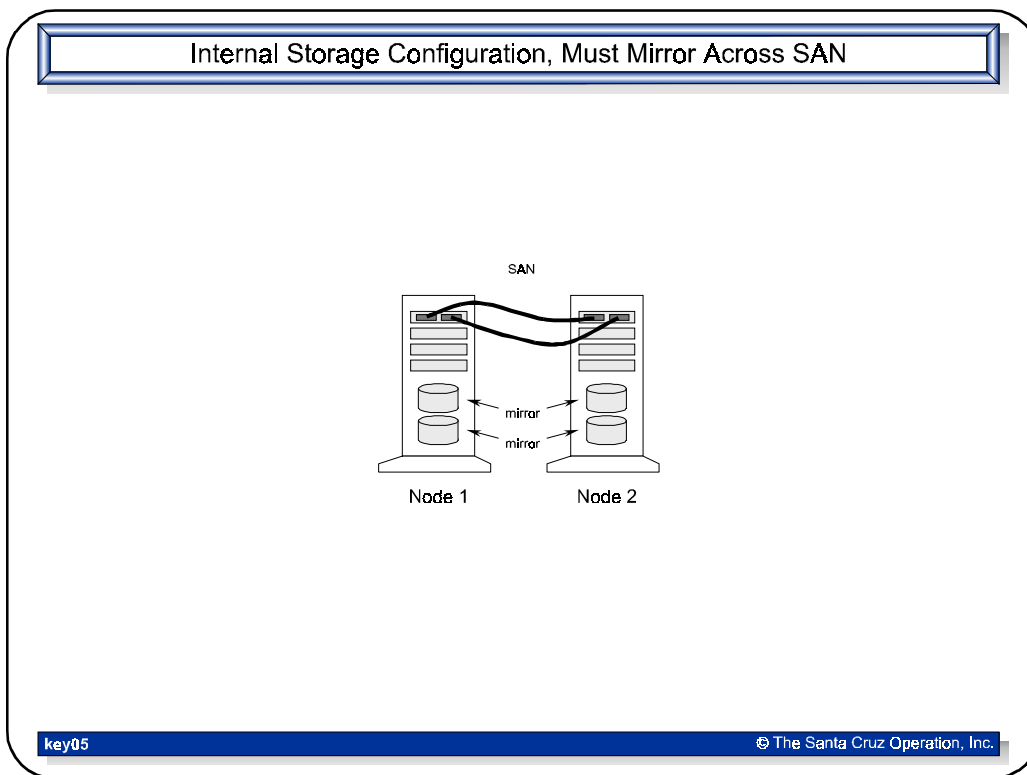
In a true cluster, you use the same tools to perform an operation on the cluster as you would on a single host-based system. For example:

- If you use **useradd**(1M) to add users to a single host, in a true cluster you should be able to use **useradd** to add users to the cluster.
- If you use **ifconfig**(1Mtcp) to configure network interfaces in a single host, in a true cluster you should be able to use **ifconfig** to configure an interface in the cluster.
- If you use **ps**(1) to look at the list of processes running on a single host, in a true cluster you should be able to use **ps** to look at a list of processes running on the entire cluster.
- If processes can migrate between processors within a single host, in a true cluster the processes should be able to migrate to any processor within the cluster.

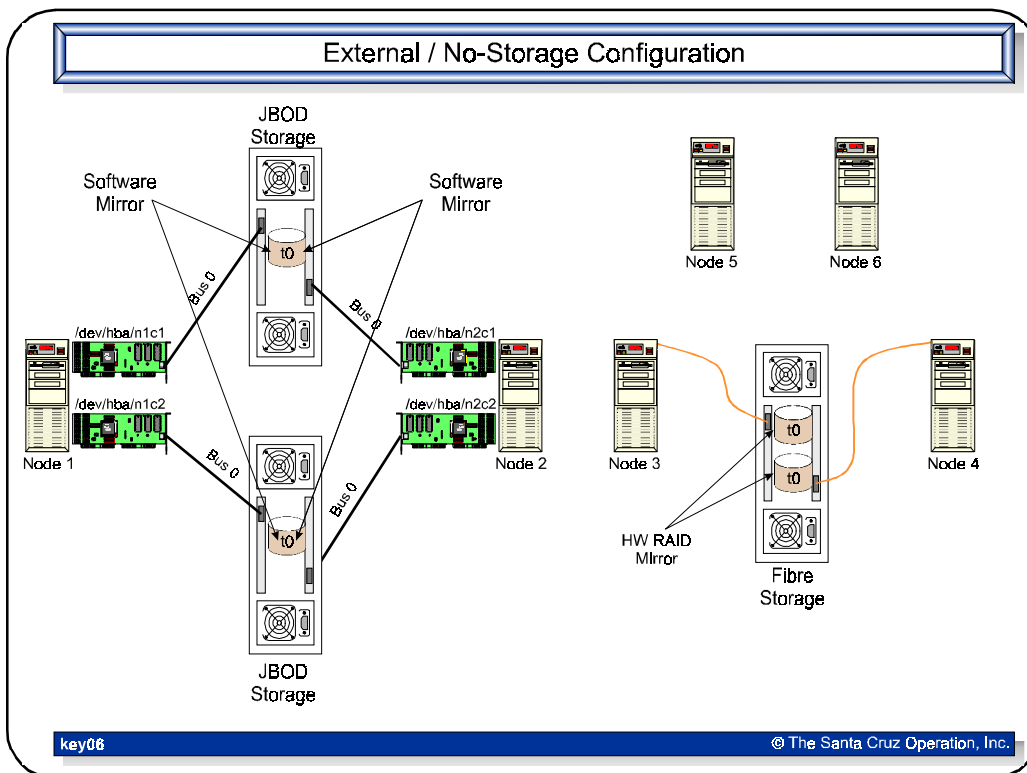
Further, any of the tasks listed above should be able to be performed from any node within the cluster. For example, you should not have to be on node 1 to configure a network interface on node 1.

Section 3.2.3, “Manageability in NSC,” on page 3-31 provides additional information on how NSC improves manageability of a cluster.

AV 3-5



AV 3-6



3.1.5 Nodes

The definition of a cluster said that the cluster contains:

“... two or more interconnected whole computers ...”

In clusters, each whole computer is called a *node*. As a whole computer, each node contains:

- One or more CPUs

Some clustering implementations allow nodes to have multiple CPUs running a highly-scalable operating system. Nodes with multiple CPUs running such operating systems are called *symmetric multi-processor (SMP)* nodes.

- RAM to hold executing programs and data
- At least one video display, keyboard, and mouse interface

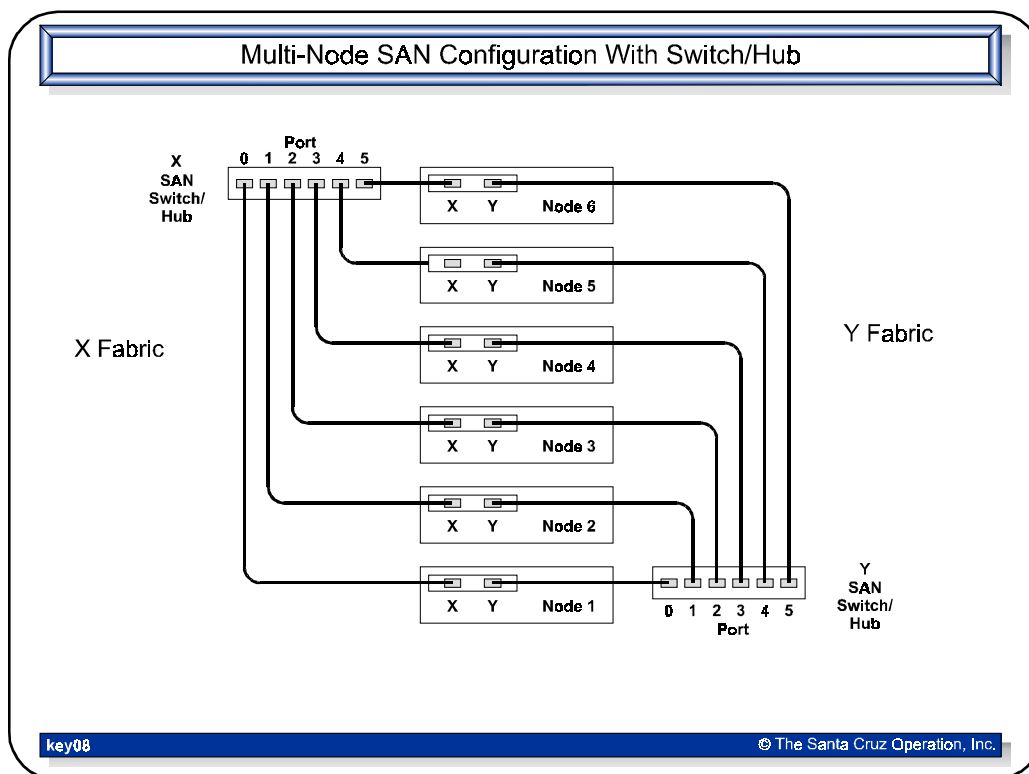
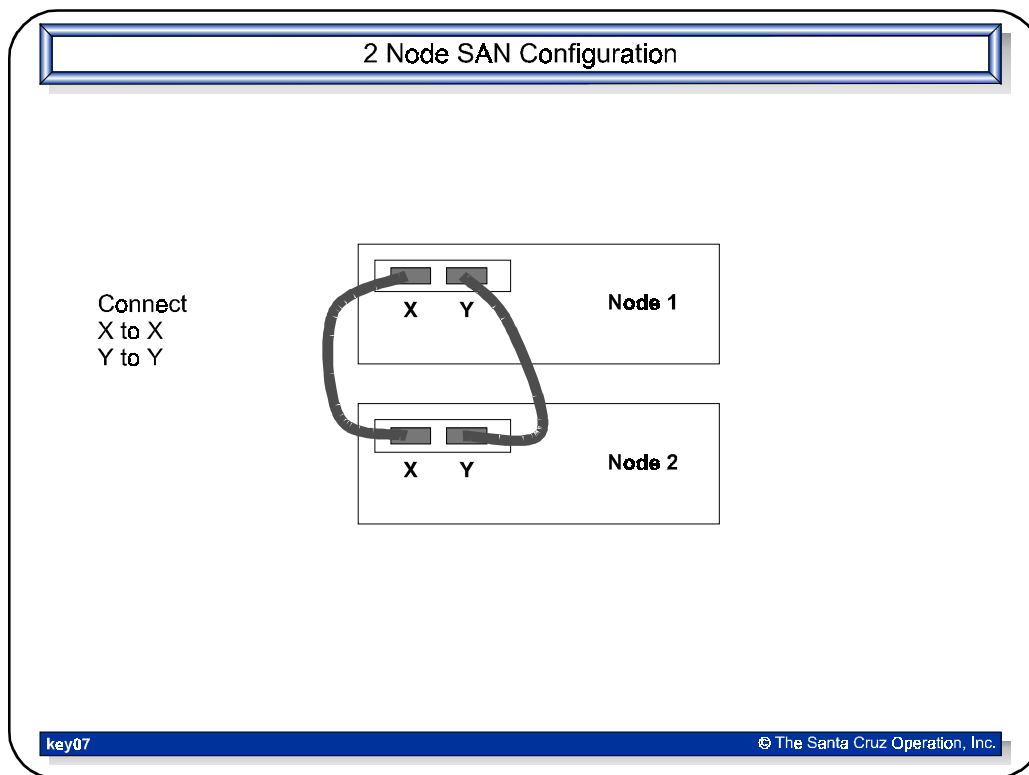
Nodes may also contain other components, such as:

- Networking Interface Cards (NICs), to communicate with the network
- Storage controllers and related devices, such as disks, tapes, CD-ROMs, and floppy drives. Some storage devices are:
 - Internal to a node, such as shown in AV 3-5 on page 3-14. For reasons discussed later, internal disks must be mirrored across the Server Area Network (SAN).
 - External to a node and may be attached to several nodes. An example of storage attached to nodes 1 and 2 is shown in AV 3-6 on page 3-14.
 - Non-existent on certain nodes, such as node three shown in AV 3-6 on page 3-14. However, if a cluster provides SSI, then such nodes can still access the storage devices attached to the other nodes in the cluster.

At least two nodes are required to insure no single point of failure. Hardware supported by NSC is provided on the SCO web site at URL:

<http://www.sco.com/products/clustering/uw7nschw.html>

AV 3-7



3.1.6 Server area network (SAN) and fabrics

The definition of a cluster stated that the nodes must be *interconnected*. In other words, the nodes of a cluster are interconnected through their own network. This network:

- Is a high-speed network
- Is called a *server area network (SAN)*
- Is often different from the networks which carry TCP/IP, SMB, or IPX/SPX traffic on an organization's LAN or WAN. This isolates node-to-node traffic from other client/server traffic.
- May employ proprietary hardware and protocols, which are optimized for node-to-node communications. The faster and more efficient the hardware and protocols, the more efficient the node-to-node communications.
- Can use various architectures, each of which must insure that no single point of failure causes the SAN to become unavailable. At a minimum, this means that the SAN must have:
 - Two sets of wires connecting each node. These wires may plug into the same or separate SAN cards within a node. AV 3-7 on page 3-16 illustrates a simple two-node cluster using a SAN topology that has only two cables.
 - Two switches or hubs, if such technology is required within a SAN. AV 3-7 on page 3-16 illustrates a SAN topology that uses switches or hubs.

Each separate set of SAN wires and hubs, if present, is called a *fabric*. Each of the slides have two labeled fabrics. For consistency with the NSC-specific discussion of fabrics later in this course, the fabrics are labeled X and Y.

The specific SAN topologies and hardware used in NSC are discussed in Section 3.2.5, "SAN options for NSC and the Virtual Interface Architecture (VIA)," on page 3-37.

3.1.7 Protected resources and components

All resources and components must be redundant in order to avoid a single point of failure. So far, redundancy has been discussed for nodes and SAN components. However, redundancy must be extended to:

Shared memory	If shared memory is not redundant, then failure of RAM that is shared will cause processes on other nodes to fail
Semaphores	If semaphores are not redundant, then failure of a node can result in processes losing contact with their source of information.
Filesystem connectivity	If filesystems connections are not redundant, then failure of a node results in the loss of the filesystem. Filesystems that are controlled and accessed by multiple nodes are considered <i>protected storage</i>. This is discussed in greater detail in Section 3.1.7.1, “Protected storage,” on page 3-18.

3.1.7.1 Protected storage

Whenever information stored on a disk is accessible from multiple nodes it is considered *protected*. Storage must be protected to avoid single point of failure. For example, when the node with control of the *root* filesystem fails, a *potential root* node will assume control of the *root* filesystem, thereby protecting the filesystem from loss of contact with the cluster.

Almost every method used to protect storage involves *redundant arrays of inexpensive disks (RAID)*.

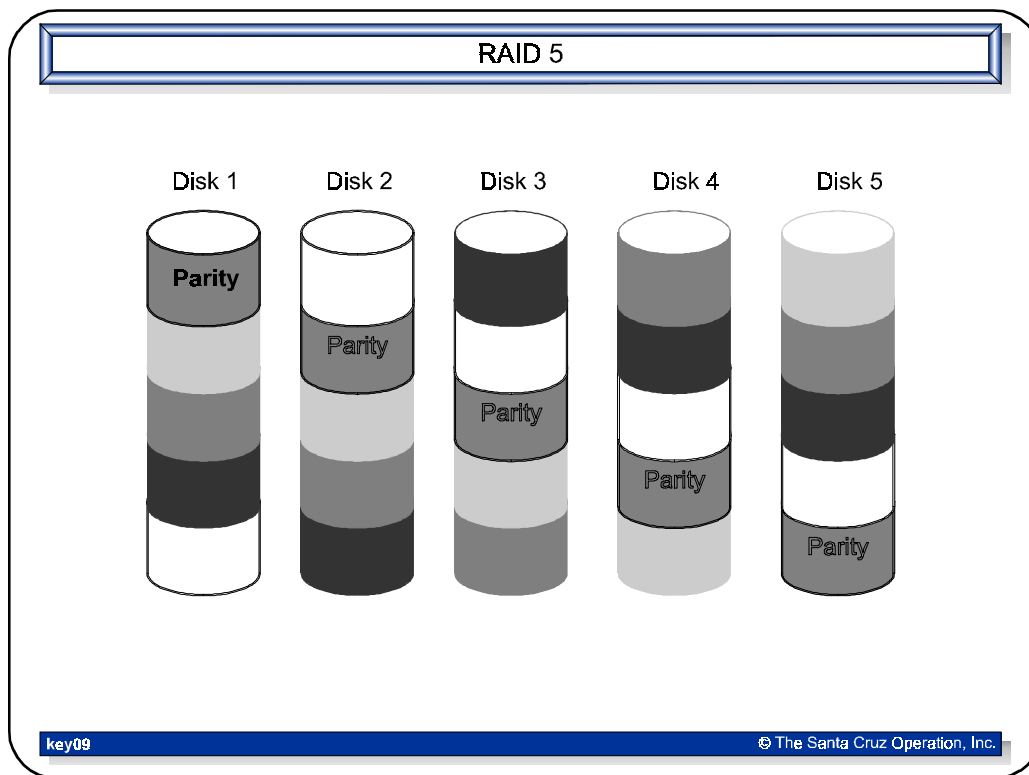
The methods used to protect storage in NSC are discussed in Section 3.2.6, “Protected storage methods in NSC,” on page 3-41.

3.1.7.2 Redundant arrays of inexpensive disks (RAID)

RAID (Redundant Arrays of Inexpensive Disks) provide redundancy of data for recovery in the event one disk in the array fails. Originally five RAID levels were defined. RAID levels 1 and 5 have become standards. Other variations have been developed.

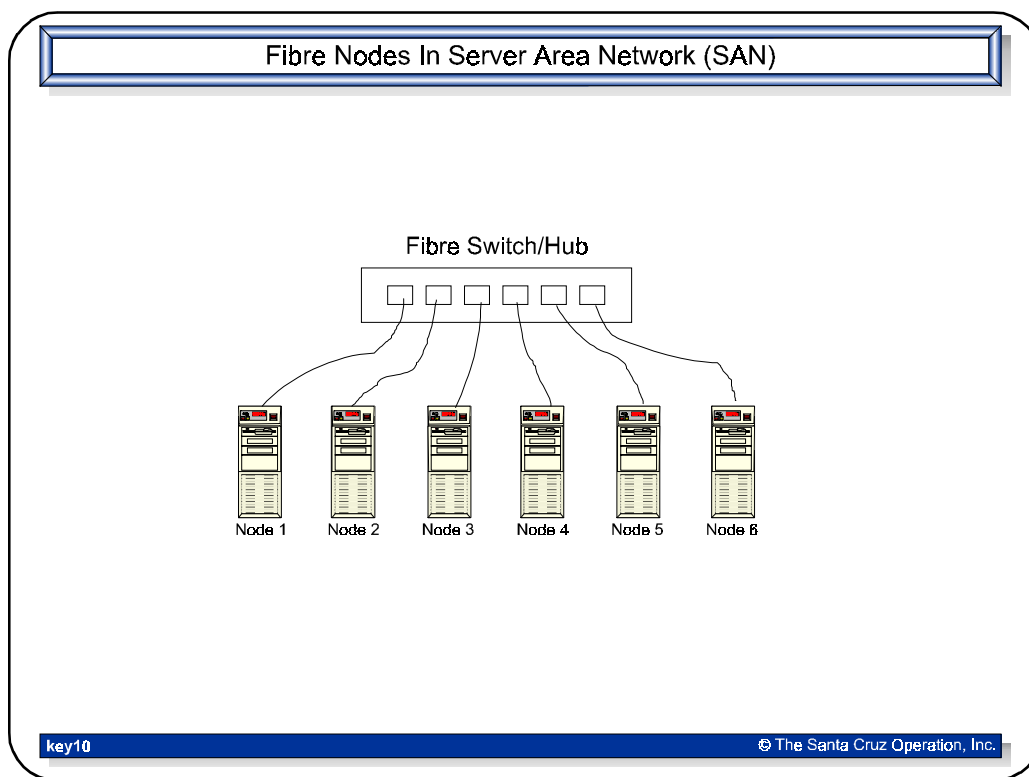
- RAID 0 stripes data across the disks in the array. It
 - Provides no data redundancy in the event of disk failure. If one disk fails, the entire array fails, and is therefore a single point of failure.
 - Provides improved read and write response times because the work is spread across several disks
 - Provides maximum use of available disk space since there is no redundancy as in RAID 1 and RAID 5 discussed below
- RAID 1 is also known as *mirroring*. It:
 - Uses disks in pairs
 - Writes all data to the primary disk and duplicates or mirrors it to the secondary disk
 - Allows the second drive of the mirrored pair to serve the data to ensure there is no single point of failure
 - Is the most expensive form of RAID because 50 percent of the disk space is consumed by mirroring

AV 3-8

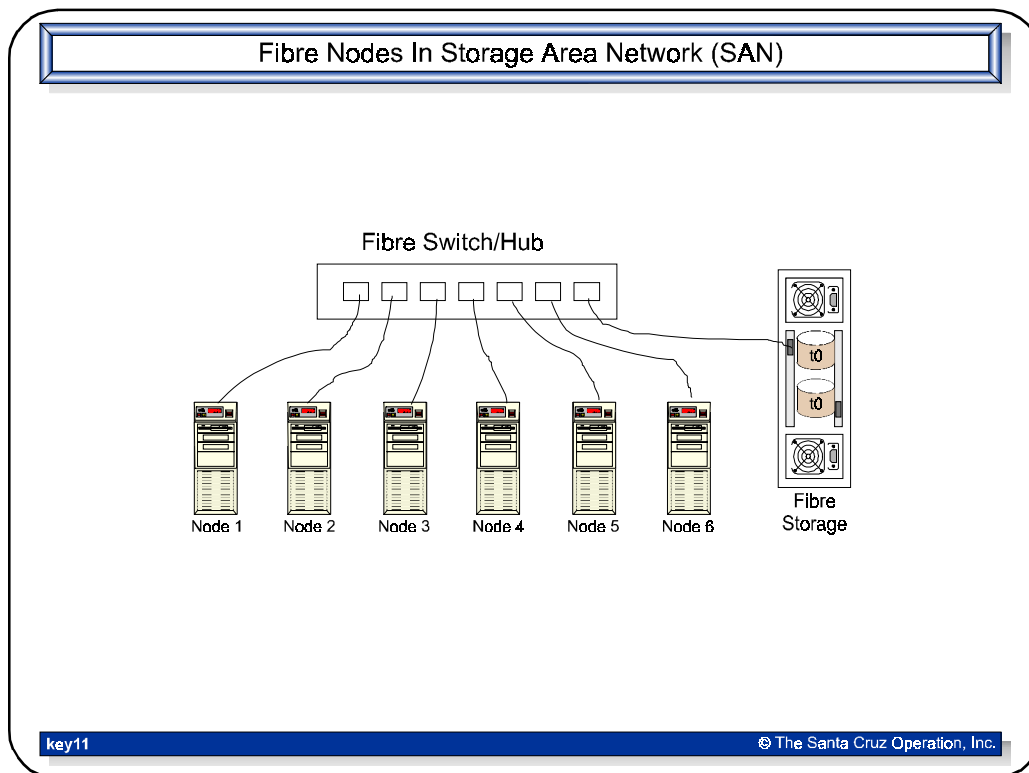


- RAID 5 *stripes* data across the disks with one segment of the stripe reserved for parity information. It:
 - Uses a minimum of 3 disks
 - Rotates parity segments across disks to provide even access to all disks. Thus, each parity stripe must be updated with each data write.
 - Does not have a single point of failure because loss of a single disk does not cause loss of data
 - Reconstructs lost data using parity information in conjunction with the remaining data
 - Provides good read performance because data can be read simultaneously from many disks simultaneously.
 - Provides only fair write performance because both the data segment and the parity segment must be updated on each data update. RAID 5 write performance is between 1/3 and 3/5 of the RAID 1 write performance according to DPT.
- Combinations of RAID
 - RAID 0+1
 - Multiple pairs can be striped together to appear as a single larger array
 - This configuration is sometimes referred to RAID 0+1 or RAID-10
 - RAID 0+5
 - Multiple RAID-5 groups can be striped together to appear as a single larger array
 - This configuration is sometimes referred to RAID 0+5 or RAID-50

AV 3-9



AV 3-10



3.1.7.3 Fibre Channel Arbitrated Loop (FCAL) and GBICS

Fibre Channel Arbitrated Loop (FCAL) is the term for fiber-optic based communications. In clusters, it can be used to connect:

- Nodes on a Server Area Network (SAN), illustrated in AV 3-9 on page 3-22. Note that some vendors use the acronym SAN to also refer to a storage area network.
- Nodes and storage devices on a storage array, illustrated in AV 3-10 on page 3-22.
- The devices that convert metal signals on controller cards inside the computer to fibre optic signals are called GigaBit Interface Converters (GBIC)s.

FCAL solutions support speeds of 100Mbps to 1Gbps. Currently there are two lengths supported:

Short wave	Supports lengths up to 500 meters. By fibre-optic industry standards, GBICs for short wave communications are colored black.
Long wave	Supports lengths up to 2 kilometers. By fibre-optic industry standards, GBICs for long wave communications are colored blue.

FCAL lengths are important for planning the location of nodes and storage. For example, with long wave FCAL, two nodes in a cluster could be 2 kilometers apart, allowing for catastrophe recovery.

3.1.8 Failover, fault zones, N+1, N+0

Whenever a protected resource fails, whether it is a node, SAN fabric, storage, or some other resource, the redundant component is pressed into service. This is called *failover*.

Failover has these important considerations:

- If the redundant component also fails while the original component is not operational, a catastrophic failure can occur, causing the entire cluster to fail and become unavailable.
- Each independent area where a single failure can occur without bringing down the entire cluster is called a *fault zone*.

For example, the X fabric SAN cable on node 1 could fail at the same time as a hard disk in the protected storage. As long as node 1's Y fabric SAN cable remains functional, and the mirror of the bad hard disk remains functional, the cluster remains available.

In this example, the SAN fabrics, and the disks serving as the protected storage are separate fault zones.

Again, to prevent a single point of failure, each fault zone must contain redundant components. Other areas of redundancy include the nodes, disk controllers for the external disk storage, UPSes, and network cards.

- Failover can be configured for software as well as hardware. If a node running a database fails, the software can be restarted on another node.
- Failover can be set up in two primary configurations, where N refers to the number of nodes:

N+0	All components are in use all the time. Loss of a single component will reduce overall capability, but not cause the cluster to become unavailable. An example of N+0 is using 2 nodes (2+0) to run data entry operations. If a single node fails, the other node will carry all of the work load.
N+1	A standby component, not in service, (the component represented by the 1) is available to take over the role of any one or more of the same components (the components represented by the N). An example is using 2 nodes (1+1) to run a database. If one node fails, the other node will start up the database applications.

3.1.9 Load balancing

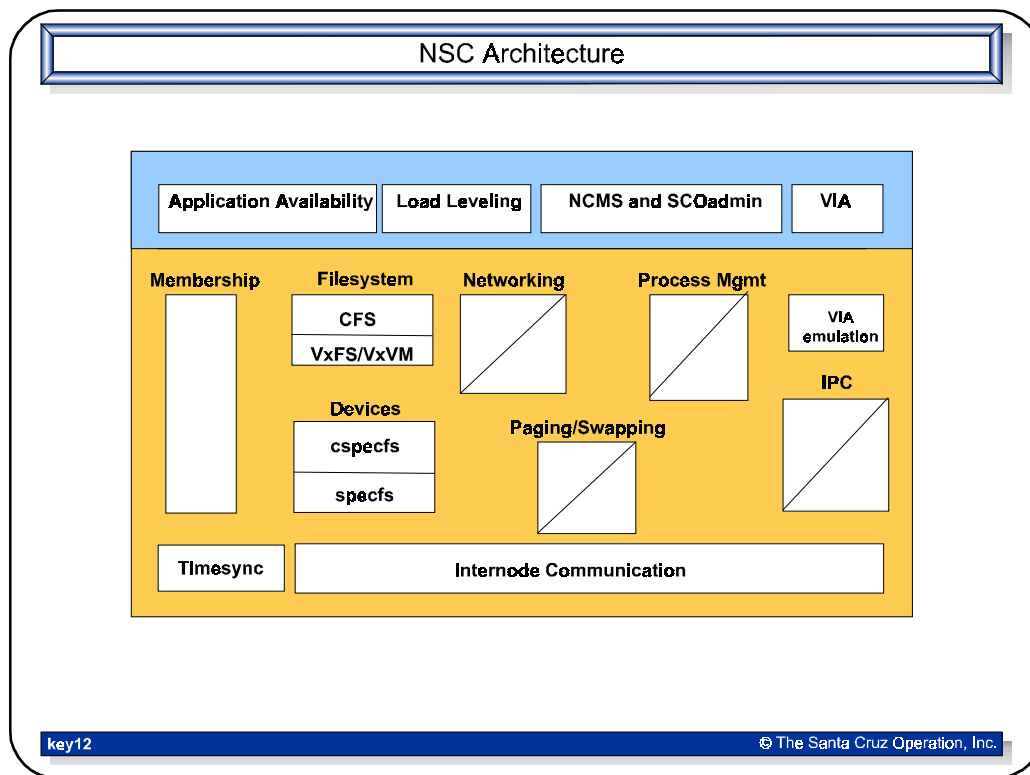
Load balancing is the movement of running processes between nodes so that the work load on each node is approximately equal. It is useful:

- When multiple nodes are running unrelated processes. This includes any heterogeneous mixture of jobs such as accounting, order entry, page layout, or file and print services.
- When the copies of the primary process can run on any one of several nodes. A good example is running the Apache web server on multiple nodes.
- To get the best performance out of the cluster as a whole

Many clustering solutions support only failover and do not provide load balancing. Another way to say this is that many clustering solutions provide only 1+1 failover. Examples of such solutions include Microsoft Wolfpack and Reliant HA.

The NSC load balancing is further described in Section 3.2.7, “Load balancing in NSC,” on page 3-41.

AV 3-11



3.2 NSC Specific Terms and Components

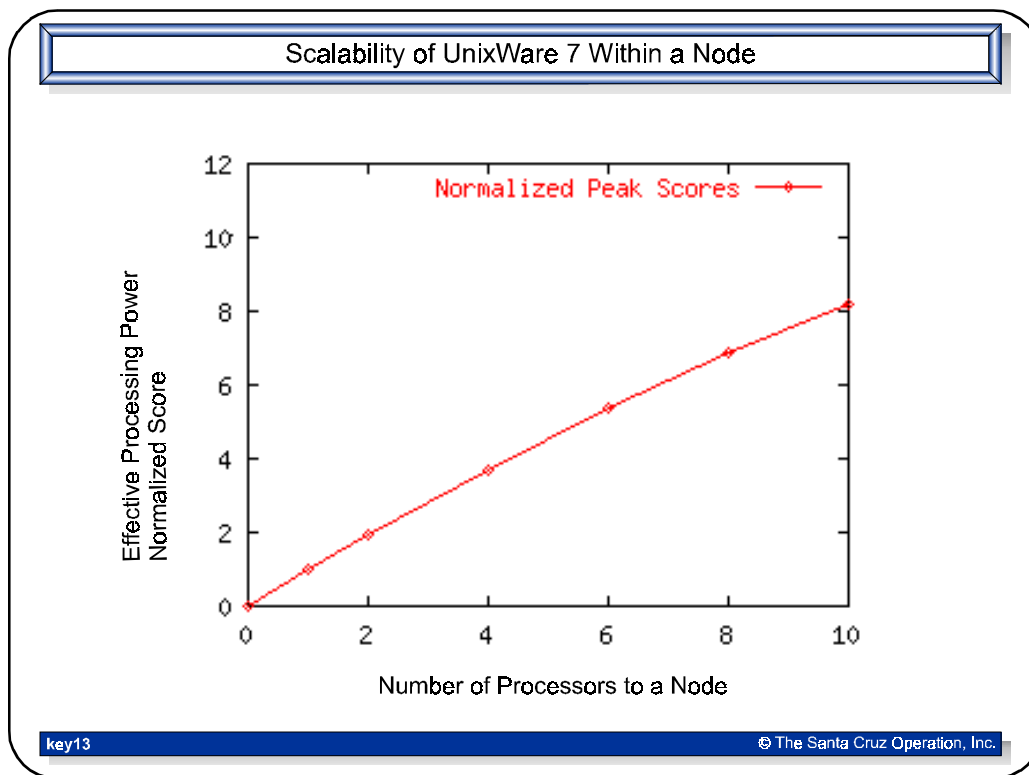
So far this course has discussed terms that apply to any clustering solution and has avoided mentioning any specific products that can be used to implement a cluster. This section introduces terms and components that are specific to NSC and discusses how these terms are used in the NSC environment.

3.2.1 SSI in NSC

NSC provides a full SSI implementation so that the entire cluster appears to the applications, users, and administrators as a single computing resource. At the present time, no other cluster implementation provides the level of SSI provided by NSC. NSC implements SSI in several ways:

- A clusterized kernel. As illustrated in AV 3-11 on page 3-26, the kernel has been modified from the standard UnixWare 7 kernel. This slide shows three types of modifications made when NSC is installed. Boxes with no lines are new capabilities, such as time-sync. Boxes with horizontal lines show added capability, such as *cspecfs* over *specfs*. Boxes with diagonal lines indicate merging of new capabilities, such as networking.
- A cluster-wide file tree and device nodes. The cluster filesystem (CFS) in the kernel implements a single file tree for the entire cluster. CFS intercepts filesystem calls and determines the actual location of the filesystem, routing the request to the node that controls the filesystem. All device nodes are visible to all processes without regard to location of device or process.
- A single process list. Only one process table exists cluster-wide. All process table entries are in the */proc* filesystem mounted at root. Administrators can manage processes without regard to the actual node on which the process is running.
- A single TCP space. Each node has a complete protocol stack and can use any NIC on the cluster.

AV 3-12



3.2.2 Scalability and Availability in NSC

There are two aspects to scalability within NSC:

- The scalability of adding another processor to a node. Since NSC uses the UnixWare 7 operating system, it has essentially the same scalability as UnixWare 7 for additional processors. This scalability is illustrated in AV 3-12 on page 3-28.

This slide shows the *normalized peak scores*, or the amount of work the node can do, with various numbers of processors present. At eight CPUs, the score is approximately 6.88. This indicates a scalability factor greater than 85 percent with eight CPUs.

- The scalability of adding another node to the cluster. NSC scales at 95% at 30 nodes. This means that if the performance of a single node yields throughput of 1 unit than the performance of 30 nodes yields throughput of $30 * .95 = 28.5$ units.

Several factors will increase the size and performance of NSC disk I/O in the future:

- Larger storage devices being certified for use with NSC
- Support for more than one HBA attached to protected storage per node. This is planned for a future release of NSC.
- Allowing more than two nodes to protect the same storage
- Allowing multiple nodes to perform physical I/Os on the same protected storage at the same time. This is planned for a future release of NSC.

AV 3-13

Features of Manageability in NSC

- Cluster-wide system administration
- Transparent Cluster-wide parallel TCP/IP networking
- Transparent Cluster-wide IPC
- Transparent Cluster-wide processes, migration, etc
- Automated application load balancing
- Applications, utilities, programmers, users and administrators often do not need to be cluster aware.
- Use of standard single node tools

key14

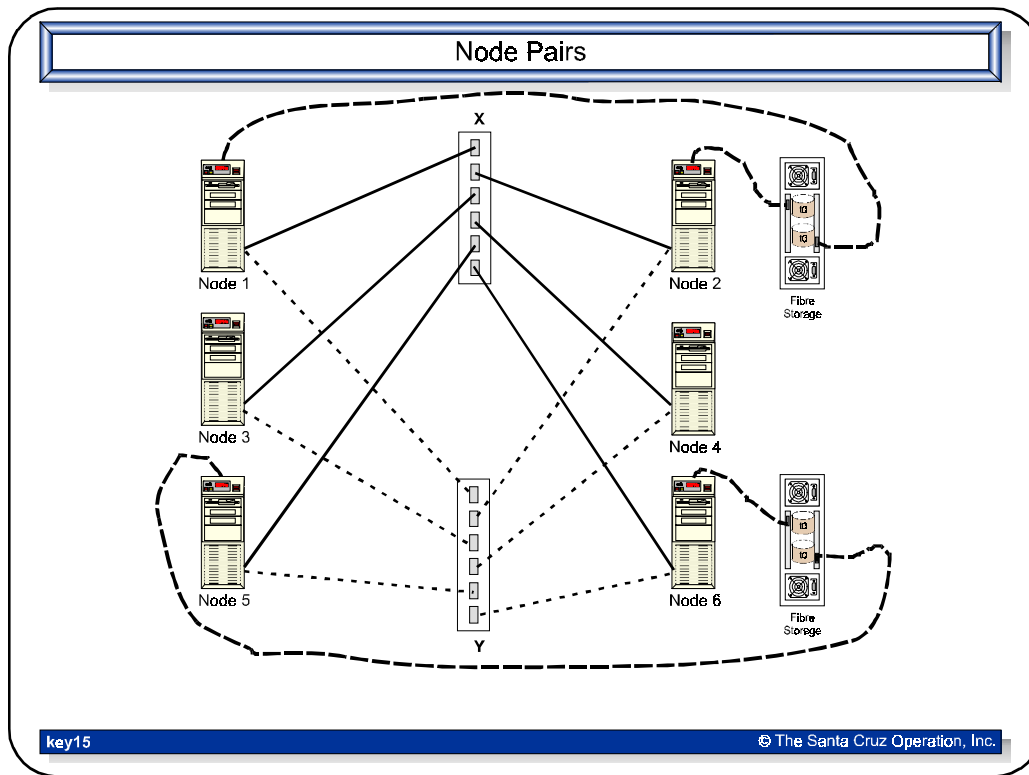
© The Santa Cruz Operation, Inc.

3.2.3 Manageability in NSC

The NSC SSI feature, combined with clusterized management tools, simplifies NSC administration. The degree of difficulty of NSC management is at the level of a single UnixWare host. Specifically:

- NSC has *clusterized* **SCOadmin** managers that perform their actions cluster wide. This is possible because of SSI, which stipulates that regardless of the size of the cluster, there is one password file, one file tree, one network configuration, one process list, and so on.
- Transparent cluster-wide parallel TCP/IP networking simplifies access to the cluster through a single IP address. This address, called the *Cluster Virtual IP* address (*CVIP*), acts like a network interface. The CVIP will failover from one node to another if the node with the CVIP fails. This provides uninterrupted service to clients.
- Using the **ps** command on any node lists all processes on the cluster. Any process can be signalled by using **kill** from any node without regard to the current location of the process. The administrator does not need to be aware of where the process is executing.
- Automated, configurable, load balancing relieves administrators of the task of constantly managing the work load.
- Applications, users, and administrators rarely need to be aware of the cluster. An application that runs on UnixWare 7 should run on NSC. A user might never be aware that their application is running on a cluster.
- Administrators and operators use standard UnixWare 7 commands to manage the cluster applications.

AV 3-14



3.2.4 Nodes in NSC

In addition to the concept of nodes discussed earlier, NSC has specific definitions of nodes. For a list of computers that have been certified for use as NSC nodes, see the SCO Clustering web site at URL:

<http://www.sco.com/products/clustering/uw7nschw.html>

3.2.4.1 Node Pairs

In NSC, a *node pair*:

- Consists of the two nodes that connect to any protected storage. In the current version of NSC, all protected storage must be protected by two nodes. Future releases may provide the ability for more nodes to participate in protecting storage.
- Contain nodes that are relatively similar in configuration to insure continued performance at the same level after a failover.
- Are designed using sequential odd-even numbered nodes. For example, the first node pair includes nodes 1 and 2, and the second (if present) includes nodes 3 and 4. This is illustrated by AV 3-14 on page 3-32.



3.2.4.2 Root node pair, potential root nodes, and current root node

The first two nodes are called the *root node pair* because:

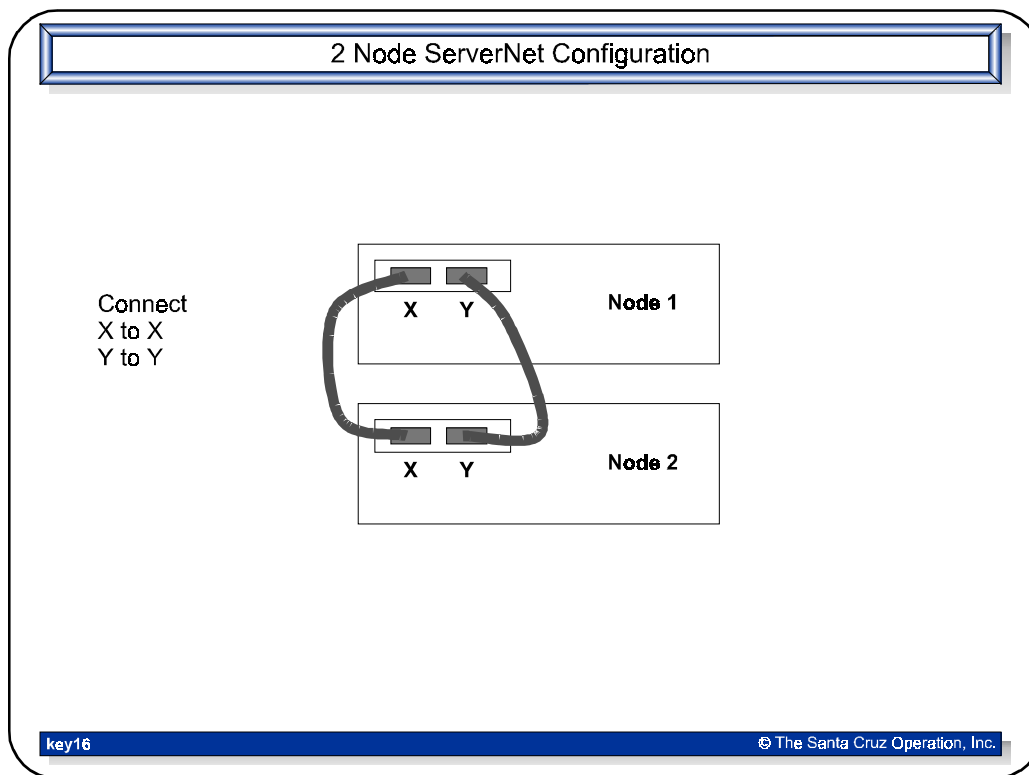
- When external storage is used for the *root* filesystem, one of these nodes physically manages the external storage of the *root* filesystem.
- When cross-node mirroring (CNM) is used, both nodes service read requests for the *root* filesystem. Writes are written to one disk and then mirrored to the other root node's disk.

Key properties of the root node pair include:

- One of these nodes must be in operation for the cluster to be available.
- The root node pair must have physical access to the disks with the root file system.
- They do not have to be the most powerful nodes in the cluster. Depending on the services provided by the cluster, the root node pair may be much less powerful than other nodes, with the powerful nodes running database applications and the root nodes providing less intensive services.
- Only one is acting as the root node at any time. This node is called the *current root node*.
- The other node, called the *potential root node*, while still capable of running applications, is standing by to assume the role of the current root node.
- Certain services only run on the current root node

In a future release of NSC, storage may be protected by more than two nodes. In that case, all nodes protecting the *root* filesystem will be *potential root nodes*. That is, any of those nodes could potentially become the root node.

AV 3-15



3.2.5 SAN options for NSC and the Virtual Interface Architecture (VIA)

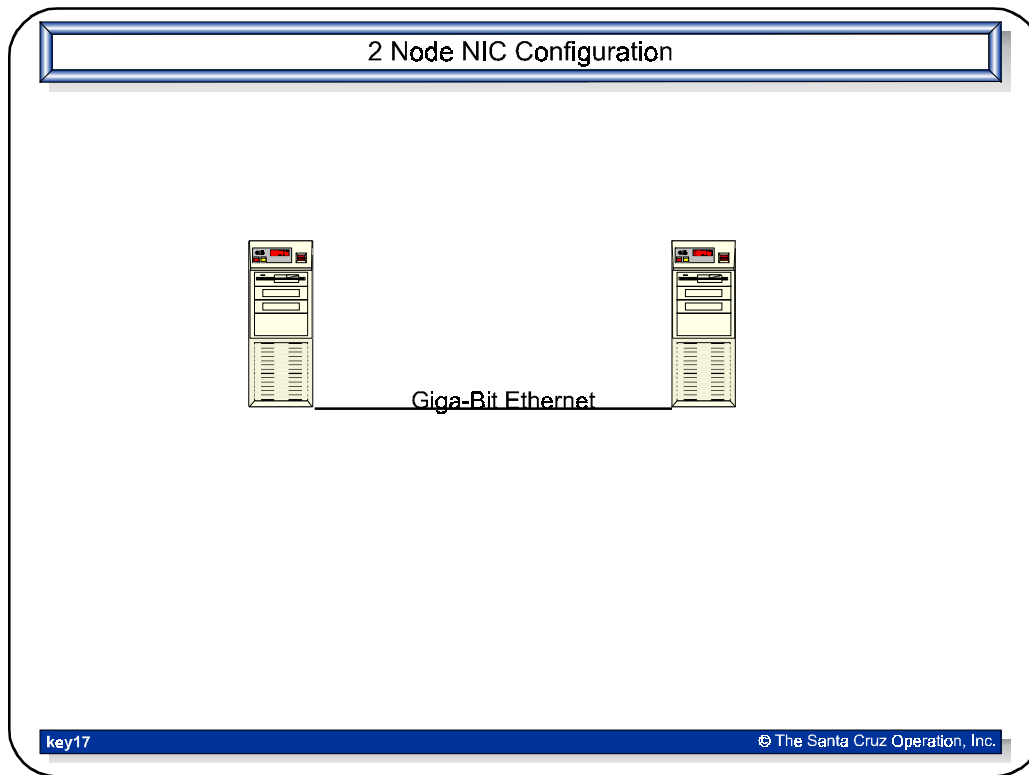
The options for implementing a SAN in NSC are being expanded rapidly. For a complete list of current options, see the SCO Clustering web site at URL:

<http://www.sco.com/products/clustering/uw7nschw.html>.

Projecting through the end of the year 2000, these SAN options are expected to be available:

- *ServerNet* is a proprietary solution from Compaq that:
 - Uses PCI cards that use few CPU resource because of the card's intelligence. They have two cable connections to allow transmitting and receiving messages on two interconnect fabrics, called X and Y.
 - Uses thick, many-pinned cables that are currently limited to 4 meters in length. This may be an issue for the physical location of nodes in a cluster.
 - Reaches higher maximum transfer rates because it does not use a standard networking protocol stack or physical layer
 - Has a transfer rate of 50MB/second (400Mb/second)
 - Has two possible configurations:
 - As illustrated in AV 3-15 on page 3-36, when only 2 nodes are present in a cluster, *ServerNet* uses only two cables, one from the X connector on one node's card to the X connector on the other node's card, and similarly for the Y.
 - As illustrated in AV 3-12 on page 3-28, when 3 or more nodes are present in a cluster, *ServerNet* requires a switch. It uses cables between the X and Y connectors on each node's card and the corresponding connector on the *ServerNet* switch.

AV 3-16



- Supported NICs. Releases 7.1.1b and later support a SAN architecture for only two nodes using Ethernet NICs. NICs up to gigabit capabilities will be included in the supported NICs. This implementation uses the network protocol stack to control traffic over the SAN and is illustrated in AV 3-16 on page 3-38.
- Virtual Interface Architecture (VIA). VIA was developed by Intel as a method for application software to access NICs directly, that is, without the need for kernel network protocol stacks or excessive user/kernel-mode transitions. VIA includes an API that must be implemented by library- and kernel-level routines.
- Servernet II. This is a follow-on to the current ServerNet technology. It will use VIA hardware as well.

3.2.5.1 Split Brain Avoidance

A *split brain* condition occurs when the potential root nodes all think that the other node is down and tries to become the root node for the cluster. In this scenario, one of three things occurs, all of which are cause problems and must be avoided:

- If you are using shared storage with an active/passive controller set, where each controller is attached to a potential root node, you will get *ping-pong panics*. In other words, one node will activate its controller, deactivating the other controller, causing the node that had the active controller to panic. When the panicked node reboots, it activates its controller, causing the other node to panic, and so on.
- If you are using JBOD, a type of dual-hosted SCSI external storage unit, data on the disks will likely be corrupted.
- If you are using CNM, both nodes update their own copy of the *root* filesystem. It will be almost impossible to merge the changes. Thus one set of changes will be lost, probably corrupting data. When a potential root node loses contact with the current root node, there are two possible reasons:
 - The root node has gone down. In this case, the potential root node must become the root node and reform the cluster.
 - The communication paths between the nodes have failed. In this case, the potential root node should take itself out of the cluster by panicking.

The same issues exist when the current root node loses contact with a potential root node. Either:

- The potential root node has gone down, in which case the root node must reform the cluster.
- The potential root node is up but can not be contacted. In this scenario, the current root node continues as the current root node assuming that the potential root node has made the same determination and panicked.

Certain SAN technologies can not distinguish between an inoperative connection and a node down. An example is a two node cluster using Compaq ServerNet without a ServerNet switch. When the potential root nodes lose contact and cannot distinguish between a node-down and an inoperable interconnect, a condition known as split-brain can occur.

To avoid a split brain condition when using such SAN technologies, you must configure *split-brain avoidance* by connecting a null-modem serial cable between the potential root nodes and setting up the NSC software to use this cable for split-brain avoidance.

3.2.6 Protected storage methods in NSC

NSC supports the following protected storage methods:

- *Cross-Node Mirroring (CNM)*. In a CNM cluster, the two root nodes use internal disks for the *root* filesystem. The two *root* filesystems are synchronized through Veritas Volume Manager. While this technique does not yield the performance of hardware RAID, the price is much lower. To improve performance, both nodes can read data from their internal disks, but only one node controls writes.
- *Shared storage*. This requires the use of external storage cabinets, populated with disks, connected to a node-pair. In a future release, NSC may support physically connecting protected storage to more than two nodes.

For a list of the protected storage components that have been certified for use with NSC, and their vendors, see the SCO Clustering web site at URL:

<http://www.sco.com/products/clustering/uw7nschw.html>..

3.2.7 Load balancing in NSC

In NSC, load balancing is turned off by default. When started, load balancing is available continuously to monitor CPU load and migrate processes among nodes. It is a highly-configurable subsystem, allowing for a great deal of flexibility in controlling process movement. For example, you can configure these metrics to control how load is balanced:

- What constitutes an *overloaded node*
- What constitutes an *underloaded node*
- How much CPU time a process running a candidate application must garner before it is eligible for migration
- The predicted effect on the load of the nodes affected by migration

Load balancing can greatly degrade performance of an application and the entire cluster in these circumstances:

- An application consists of multiple processes
- The processes perform a significant amount of interprocess communication

Databases, such as the non-parallel versions of Oracle, Informix, and Progress are all examples of applications that should not be subject to load-balancing.

The nodes of a cluster can be configured into *load balancing domains*. This allows you to configure which applications are balanced across which nodes. This allows you more fine-grained control over which applications are included in load-balancing among which nodes.

3.2.8 Physical and Virtual IP Network Addresses

Each NIC in a cluster has a *physical IP* address or *PIP*. All NICs can be on the same network or the cluster can have interfaces on multiple subnets. To simplify client access to the cluster, the cluster is assigned a *Cluster Virtual IP* address or *CVIP*. The CVIP is initialized on the first node to boot. The cluster sends out an ARP identifying the CVIP as having the MAC address of one of the NICs on the node to which the CVIP is currently assigned.

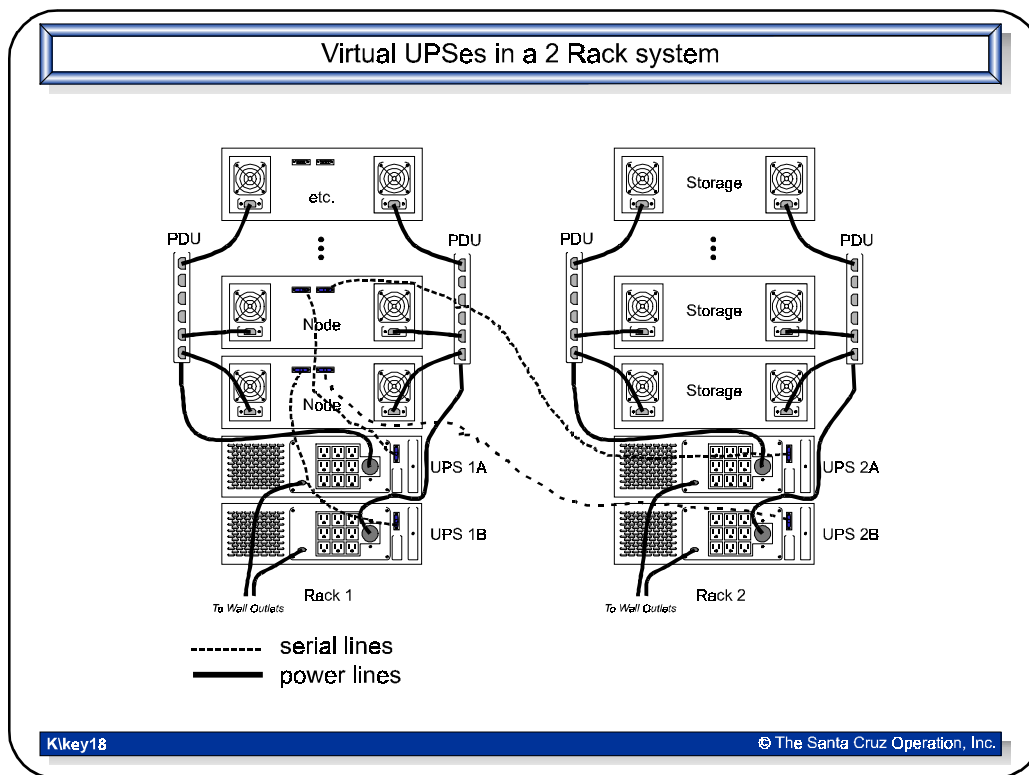
When clients need to access the cluster, they can access the CVIP or any of the PIPs in the cluster. The advantage of configuring clients to use the CVIP is that it will always be available if the cluster is running. A client that is configured to access a PIP will not connect if the node with the PIP is down.

Since there may be many NICs on different subnets, up to sixteen CVIPs can be configured, one per subnet. The primary CVIP is associated with the cluster name and should always be on a subnet that connects to the root nodes.

When the root node fails, the cluster will regroup and a potential root node takes the role of root node. Part of the regrouping process is to set up the CVIP on the new root node and send out the ARP broadcast to identify the MAC address of the CVIP. This insures clients have access to the cluster and do not have to reconfigure their telnet or ODBC attachments every time the CVIP is moved.

Data traffic emanating from the cluster can depart on any NIC. For this reason, any system to which the cluster sends data that is authenticated by IP address must consider the cluster as a multihomed host. For example, a print server must list the CVIP and all PIPs for the cluster in its *host.equiv* or *hosts.lpd* file.

AV 3-17



3.2.9 Virtual UPS

With single computers, the system is often configured to shutdown if the UPS detects a power failure. A cluster should have multiple UPSes for redundancy.

NSC allows you to specify which UPSes can shut down without shutting down the cluster. UPSes configured in this manner are called a *Virtual UPS*.

As shown in AV 3-17 on page 3-44, two UPSes (labelled 1A and 1B) could be configured to run the dual power supplies on a cluster's nodes. If one of the UPSes, 1A or 1B, shuts down, there is ample power on the remaining UPS to run the cluster. Consider UPSes 1A and 1B to be Virtual UPS 1.

Likewise, the external storage arrays can be powered by two UPSes, labelled 2A and 2B. If one of those UPSes fails, the external array can continue to run. Consider UPSes 2A and 2B to be Virtual UPS 2.

NSC allows you to configure the virtual UPSes with a control file. When power failure signals are received, the NSC virtual UPS software can examine the status of each UPS and make an intelligent decision on shutdown.

In the example above:

- If any one fails, the cluster will continue to run.
- If either of the internal UPSes in a virtual UPS fail, the cluster will continue to run
- If a virtual UPS loses power (all internal physical UPSes lose power), then the cluster must shut down.

Virtual UPSes provide an additional layer to the concept of no single point of failure. In reality, the UPSes within a virtual UPS should be connected to different generators and power grids. For example, UPS 1A and 2A can be connected to one wiring grid backed up by a generator, and UPS 1B and 2B can be connected to an alternate grid and generator.

3.2.10 Clusterizing

While the existence of a cluster is transparent to an application, you can choose to make your applications cluster-aware. This process is referred to as *clusterizing*.

Clusterizing applications is beyond the scope of this course. The **UnixWare® 7 NonStop® Clusters Administration** course provides basic information about clusterizing shell scripts, configuring applications for failover, and troubleshooting applications running in an NSC environment.

Additional information about optimizing applications for NSC is available from the SCO Clustering web site at URL:

<http://www.sco.com/products/clustering/optc/download.html>

3.3 Lesson Summary

This lesson presented key concepts generic to all clustering environments and NSC specifically.

Review questions

1. What does no single point of failure mean?
2. How does protected storage differ from a RAID array connected to a server?
3. How does single system image reduce the work of managing an NSC cluster?
4. What is the difference between a clusterized and non-clusterized command or **SCOadmin** manager?

Review Question Answers

This section contains the answers to the review questions on page 3-47.

1. No single point of failure means that each fault zone has redundant components.
2. Information stored on a disk that is accessible from multiple nodes it is considered *protected*. Storage must be protected to avoid single point of failure. A RAID array is connected to a single server.
3. The SSI reduces the work of managing a cluster because it provides a view of the cluster as a single system. Regardless of the size of the cluster, there is one password file, one file tree, one network configuration.
4. A clusterized command or SCOadmin manager is made aware of the cluster. For example, the **SCOadmin Process Manager** displays the list of processes that are running cluster-wide rather than the list of processes on a single node. A non-clusterized command is not aware of the cluster. For example, the **SCOadmin Video Configuration Manager** is not cluster aware and will only show the video configuration of the node that it is running on.

4

Cluster Comparisons



Lesson 4

Cluster Comparisons

Introduction

There are several products on the marketing that provide one or more aspects of clustering as defined in the previous lesson. Even if they only implement one of the aspects of clustering, they are referred to as a “cluster” product.

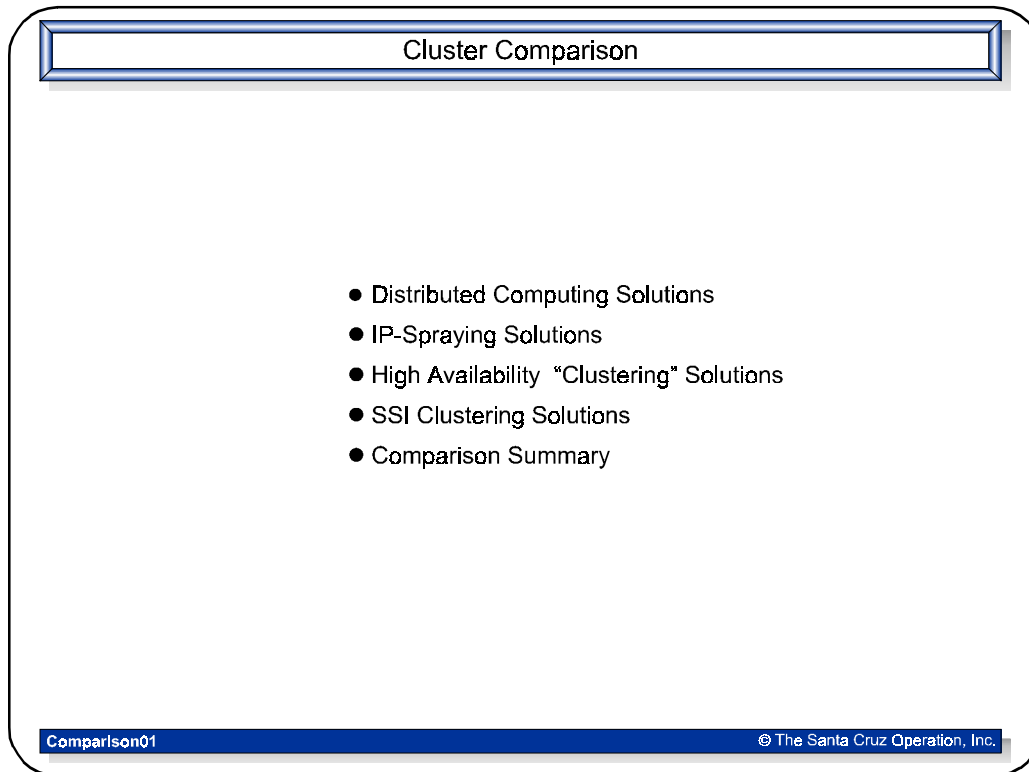
This lesson discusses some of the other “cluster” products and compares and contrasts them with UnixWare[®] 7 NonStop[®] Clusters (NSC).

Objectives

By the end of this lesson, you will be able to:

- List the names of some of the products with which NSC competes
- Indicate which products are truly cluster products
- Indicate which products are really not cluster products, and why
- Give at least three reasons why NSC is a better solution than its competitors

AV 4-1



Topics

In order to meet these objectives, this lesson discusses:

- Distributed Computing Solutions — products that claim to be clustering but are really just *distributed computing*
- IP-Spraying Solutions — products that claim to be clustering but are really a form of *IP-spraying*
- High Availability “Clustering” Solutions — products that exhibit many of the features of clustering, but do implement single system image (SSI)
- True Clustering Solutions — products that exhibit all of the features of clustering, including single system image (SSI).
- Comparison Summary — shows the differences between the various “clustering” products

4.1 Distributed Computing Solutions

The idea of using the CPUs of multiple computers to increase performance of applications has been around for many years. Many of these types of systems have been developed, both for commercial and experimental use.

However, these systems are not clustering solutions as defined by this course.

- They do not have any sort of single system image (SSI).
- They do not share a filesystem.
- Their only form of cooperation is sharing the answers to their computational problems.

Therefore, such systems are typically only useful for performing distributed computing or distributed number crunching.

Examples of these types of solutions include:

Beowulf This is a solution that uses many commodity, off-the-shelf computers, linked by an LAN, to solve specific computational problems. For more information on this solution, see <http://beowulf.gfsc.nasa.gov>.

SETI@home This is a project that is used to search through the data generated by the radio telescope run by the Search for Extra-Terrestrial Intelligence (SETI) project. The goal of the project is to find evidence of intelligent life in the radio telescope data. For more information on this solution, see <http://www.setiathome.ssl.berkeley.edu>.

Note The SETI@home project does not claim to be a clustering solution. However, it is an example of a distributed computing solution.

In each of the examples above, each computer is its own independent unit, with its own disks, user database, and so on. The only resemblance they have to clustering is that there are multiple computers involved and they are all cooperating on the same task.

4.2 IP-Spraying Solutions

The advent of the World Wide Web also generated a need for a single organization to be able to distribute the load of incoming HTTP requests across many computers. That is, when a request comes into an HTTP daemon on one machine, it can deflect the request to another server within a cooperating group. This group is referred to as a “cluster.”

This same approach can be used for other Internet-related services, but is primarily currently used only for HTTP servers, which serve most of the Web content.

An example of a solution that uses this approach is TurboLinux Turbo Clusters. However, these solutions are not clustering, as defined in this course, for several reasons:

- They do not have any sort of single system image (SSI).
- While they may NFS-mount the HTTP documents to be served, each machine has its own filesystem.
- There is no dynamic load balancing.

4.3 High Availability “Clustering” Solutions

There are a number of products that closely resemble a cluster, as defined in this course. The key feature they all lack is single system image (SSI) and dynamic load balancing. These two features are critical to cluster management and scalability.

Most of these high-availability solutions have shared filesystems and some form of failover. Examples include:

- Novell Cluster Server (NCS)
- Microsoft Wolfpack
- Windows 2000 Data Center Edition
- IBM HACMP
- SunCluster
- HP ServiceGuard
- DEC Truclusters
- SCO Reliant HA

4.4 True Clustering Solutions

There is currently only one clustering solution that meets the full definition of a cluster, as presented in this course; UnixWare® 7 NonStop® Clusters (NSC). It implements SSI, shared storage, failover of disk and applications, as well as dynamic load balancing of applications.

Because NSC implements SSI, standard UNIX administration skills can be directly applied to managing a UnixWare 7 NonStop Cluster.

AV 4-2

Comparison Summary								
Solution Type	Example Solution	SSI	Shared Disk	Disk Failover	Application Failover	Dynamic Load Balancing	Max Nodes	CPUs / Node
Distributed Computing	Beowulf	N	N	N	N	N	Unlimited	HW/OS Dependent
Distributed Computing	SETI@Home	N	N	N	N	N	Unlimited	HW/OS Dependent
IP Spraying	TurboLinux Turbo Clusters	N	HTTP Docs Only	N	Y	HTTP Requests Only	Unspecified	8
High Availability	Novell Cluster Server (NCS)	N	Y	Y	Y	N	32	8
High Availability	Microsoft Cluster Server (Wolfpack)	N	Y	Y	Y	N	2	4
High Availability	Microsoft Windows 2000 Data Center Edition	N	Y	Y	Y	N	4	8
High Availability	IBM HACMP	N	Y	Y	Y	N	32	24
High Availability	HP Serviceguard	N	Y	Y	Y	N	16	Unspecified
High Availability	DEC Trueclustes	N	Y	Y	Y	N	4	14
High Availability	SCO Reliant HA	N	Y	Y	Y	N	10	32
Real	UnixWare 7 NonStop Clusters	Y	Y	Y	Y	Y	30	32
Comparison02						© The Santa Cruz Operation, Inc.		

4.5 Comparison Summary

AV 4-2 on page 4-10 summarizes the different “clustering” solutions discussed in this lesson, grouped by type of solution. Note that only UnixWare 7 NonStop Clusters has a “Y” in each column indicating capabilities found in true clustering solutions.

4.6 Lesson Summary

This lesson discussed the various kinds of solutions on the market today that claim to be clustering solutions and explained why most of these solutions are not true clustering, as defined in this course.

Review questions

1. List three products on the market today that claim to be clustering solutions.
2. Provide two reasons why Microsoft Wolfpack is not true clustering, as defined in this course.
3. Provide two reasons why TurboLinux Turbo Clusters is not true clustering, as defined in this course.
4. Give three reasons why NSC is a better solution than its competitors.

Review Question Answers

1. The products on the market today that claim to be clustering solutions include:
 - Beowulf
 - TurboLinux Turbo Clusters
 - Novell Cluster Server (NCS)
 - Microsoft Cluster Server (Wolfpack)
 - Microsoft Windows 2000 Data Center Edition
 - IBM HACMP
 - HP ServiceGuard
 - DEC Truclusters
 - SCO Reliant HA
 - SCO UnixWare 7 NonStop Clusters (NSC)
2. The two reasons why Microsoft Wolfpack is not true clustering, as defined in this course, are that it lacks single system image (SSI) and dynamic load balancing of applications.
3. There are three reasons why TurboLinux Turbo Clusters is not a true clustering solution, as defined in this course. They include:
 - It does fully share disk storage between servers
 - It does not implement single system image (SSI)
 - It does not perform dynamic load balancing of applications
4. Three reasons why NSC is better than other solutions listed in this course include:
 - It has full single system image (SSI)
 - It is easier to administer
 - It can perform dynamic load balancing of applications, increasing scalability



5

Designing an NSC Cluster



Lesson 5

Designing an NSC Cluster

Introduction

A cluster is more than a collection of computers, hence it is more complex to design and build. Cluster design must include careful consideration of the overall system requirements, including performance goals, failover capability, and redundancy within fault zones.

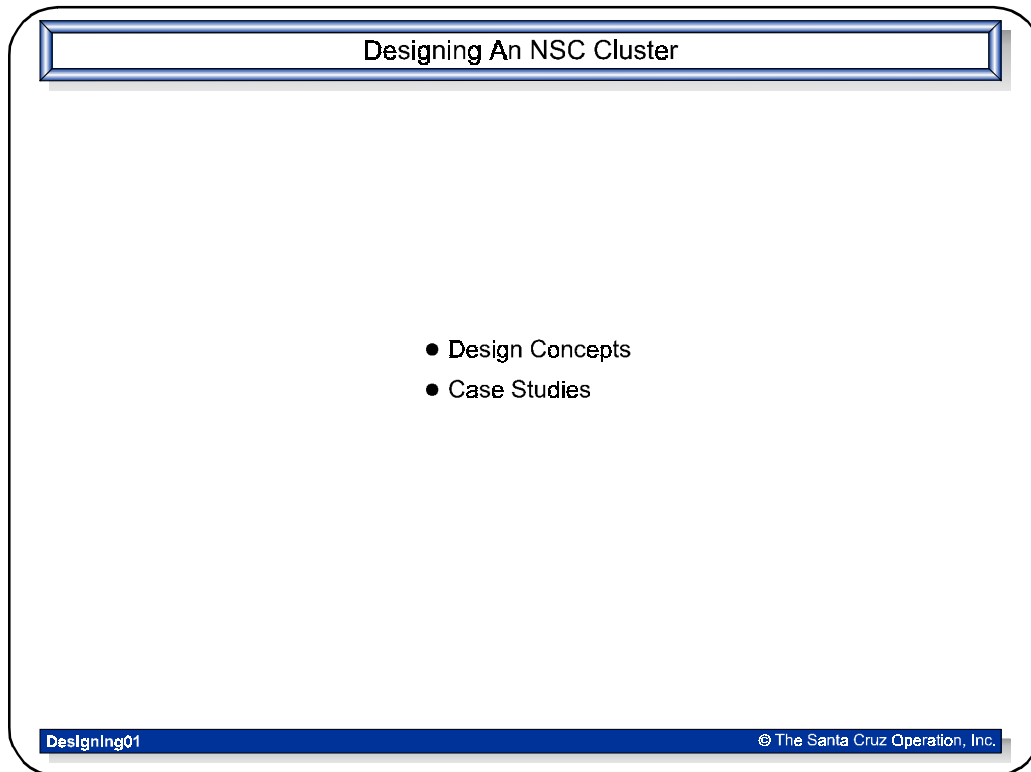
This lesson presents guidelines and rules for designing clusters.

Objectives

By the end of this lesson, you will be able to design a cluster based on a set of system criteria, including:

- I/O performance
- Vulnerability time
- Failover
- Load balancing
- Hardware capabilities, such as CPU cycles, amount of RAM, and disk space

AV 5-1



Topics

This lesson discusses:

- Design Concepts — using guidelines and rules for designing a cluster
- Case Studies — citing small, medium, and large systems with varying needs.

5.1 Design Concepts

When designing a cluster, carefully consider these criteria:

- Minimizing cluster vulnerability and avoiding single points of failure.
- Selecting and configuring nodes
- Deciding whether to use a tower or a rack
- Protecting your storage
- Protecting your power

Each of these issues is discussed more fully in this section.

5.1.1 Vulnerability and single point of failure

The two key design issues for clusters are *vulnerability* and *no single point of failure*.

No single point of failure is paramount to cluster design. You cannot over-emphasize redundancy in your cluster design. Customers that are extremely cost-conscious might want to trade cost savings for adequate design. *Resist these attempts*; you will certainly be blamed if the cluster fails, regardless of whether it was the customer who encouraged design changes that introduced points of failure.

Keep these points in mind:

- Plan for hardware failure and design the cluster around component failures
- Plan for electrical failures and design the power system accordingly. Use redundant generators supplying redundant power to the redundant UPSes.
- Plan for slow response time by component vendors and maintain a spare parts inventory. Geographical location, vendor response time, and other factors determine how many spares of each component to keep on hand. The more remote the location, the more components you should have on hand.

Include these components in your inventory:

- Keep enough disk drives on hand to withstand a failure of some percentage of disks. Consider the need for same or similar sized drives in RAID arrays.
- Consider keeping a complete spare node on hand. In a two node cluster, the vulnerability of the cluster is large as long as one node is down. A spare node can be used for daily non-essential tasks and added to the cluster when a node fails.
- Consider the need for spare cables. While cables do not fail frequently, pins can be bent when cabling repaired components.
- Check the condition of the batteries and keep replacements on hand. UPS batteries fail with age.

In addition to preventing a single point of failure, you must consider the vulnerability of the cluster when a single component fails.

If a component fails, the entire cluster is vulnerable to complete failure until the component is replaced. You should replace failed components as quickly as possible.

Consider CNM vulnerability during re-mirroring of the root hard drive. The cluster cannot withstand a failure of the current root node while the potential root node is being re-mirrored.

5.1.2 Node design

Node capacity is important to the overall performance of the cluster. Because nodes are usually used in pairs to protect disks and other resources, the two nodes in a node-pair are usually configured equally.

Determining node capacity requires an overall view of the cluster, its components, the workload, load balancing and the failover design. First consider the root node. It:

- Is required to access the root disk
- Might also control the disks used for non-root data storage
- Can perform tasks other than serving the *root* filesystem
- The root node and the failover nodes are approximately the same configuration. If the current root node fails, a potential root node assumes the role of current root node. The configuration for node 1 is used for other root node candidates.
- Might or might not have disks directly attached. Remember, all nodes have an internal disk for booting.
- Is sized according to the expected usage. The amount of RAM and the number and speed of processors are the main considerations.
- Should have ample network access. Plan to include one or more NICs per node. While each node has a complete TCP stack and can use the NICs on other nodes, the preferred path is through a local NIC. Use additional NICs to provide access to additional subnets.
- Should have enough PCI slots to handle the boards for disk controllers, NICs, SAN connection, serial and other connections
- Should have redundant tapes and CD-ROM drives.

With rack mounted systems that have external storage, do not choose nodes with large capacity (lots of slots) for internal disk drives. Because only one internal disk is normally used for booting, the unused disk slots in the server take up rack space. Further, the cost of nodes increases with the increased capacity for internal disks.

When sizing a particular node, consider the applications requirements:

- Is the application a single monolithic database?
- Will multiple applications (mixed workload) or multiple copies of a single program be run?
- Do all applications require maximized availability? Is downtime acceptable for certain applications?
- Is degraded performance acceptable when component failures occur?
- How do components of the application communicate?

Monolithic databases require heavy inter process communications and should not be spread across multiple nodes.

- What resources are required by the applications? Disk access? RAM? CPU?
- How do the applications scale?
 - For monolithic applications, add CPUs
 - For process replication, add nodes
- Do the applications use the disk primarily for reads or writes?

5.1.2.1 Cluster groups

Determine the resources required for each application or group of applications by:

1. Dividing the applications into groups
2. Analyzing the requirements of each group
3. Mapping the requirements to physical resources

Determine these requirements by calculating each application's resource needs:

- Disk space required for the applications and their data
- The disk I/O workload
- RAM amounts for each application running simultaneously
- Number of CPUs to handle the processing

After the physical requirements are established, consider the cluster requirements from an availability and scalability perspective. Is the configuration primarily for:

- High-availability—such as a small server that provides a critical business service
- Scalability—the cluster is expected to grow
- Both scalability and availability—you cannot afford downtime and need expansion capability

Another factor is the sensitivity to cost. A small business installing their only server has a different view of cost than an international company rolling out hundreds of replicated sites. Cost can be contained by using CNM instead of external disks, or by accepting some performance degradation if a node fails.

5.1.2.2 Mapping applications to nodes

After the computing requirements are determined, map the applications to nodes. Configure the nodes according to their work load. Keep in mind that external disks provide better performance than CNM.

If the total computing requirements easily fit into a single computer and the reason for clustering is high availability, a two node cluster is sufficient. With a two node cluster, other considerations are:

- Disk I/O requirements for the applications:
 - If the I/O requirements are low, then a CNM system provides a low cost hardware solution.
 - If disk I/O requirements are high, a high-performance external disk enclosure with RAID or JBOD is needed
- With a two node cluster, particularly with CNM, a pair of tower servers is less expensive than a rack mount system and cabling should not be a problem.
- If the cluster is expected to grow rapidly, a rack mount configuration should be considered.
- Allow for expansion to accommodate increases in computing requirements

If the applications fall into two or more distinct groups, decide if the groups can be run on different nodes, each with its own failover resources. For example, an e-commerce server might have a large, monolithic database that requires a node pair for the database and a web server running multiple copies on several other nodes.

Plan for the failover resources. In the above example, the database runs in an N+1 configuration: one server in use and one standing by to run the database in the event the first node fails. The multiple web server processes can run on servers in an N+0 configuration. If one node fails, the others share the work although there might be some performance degradation. To avoid degradation in an N+0 configuration, add an additional node to cover the loss of performance.

5.1.3 Node format—tower or rack mount

Clusters might be small enough to use two tower nodes connected by SAN cables. If the cluster grows beyond two nodes, a rack system must be considered. Racks provide an organized method of mounting the nodes and other equipment, as well as a mechanism for organizing cables.

Consider the cabling for a six node cluster using Compaq ServerNet, three external disk arrays, a keyboard, video, mouse (KVM) switch, and a pair of UPSes:

- Six nodes with two ServerNet cables and dual power supplies each requires 12 cables and 12 power cords
- The keyboard/video/mouse switch requires 21 cables—1 for each keyboard, video, and mouse, times 6 nodes is 18 cables, plus 1 for each keyboard, video, and mouse going to the actual keyboard, monitor, and mouse
- One power supply for the KVM switch requires one (1) power cord
- One power supply for the monitor requires one (1) power cord
- Three external drive arrays:
 - Connected to drive controllers requires 6 fiber cables
 - With dual power supplies requires 6 power cords
- Two UPSes connected to power outlets requires 2 power cords
- Two ServerNet switches require 2 power cords
- Additional cables needed for network switches or hubs, drive array hubs, external tape drives, and so on

For this 6 node cluster, a minimum of 39 cables and 24 power cords are required. These must be labelled and secured within the rack. Cables should be color coded or labelled by node number and additional coding for X or Y fabric. One method is to use color coded tape on each end with the X fabric cables having an additional white tape strip.

When planning for a rack mount system, the following must be taken into consideration:

- Where will the rack be located?
 - Will the floor hold the concentrated weight of the rack?
 - Is there adequate cooling?
 - Is there adequate power?
 - Is the area large enough to allow working around the rack?
- Will the rack arrive pre-configured?
 - How will the rack be moved from the loading dock to the computer room?
 - Is there sufficient clearance through doorways and halls to move the rack? Note: loaded racks should not be tipped over to go through doorways.
 - Are the floors level and smooth between loading dock and computer room?
- If the rack is configured on-site:
 - Is there adequate room for unboxing the components and assembling the rack?
 - Will there be enough help to assemble the rack? Some components are heavy and require two people for assembly.
 - Consider the layout of the rack ahead of time. Some cables might be limited in length, forcing the placement of components based on cable length. Power cables should not run between racks when multiple racks are being used. Place heavy components low in the rack.

5.1.4 Disk configuration

While all filesystems and disk devices are transparently accessible from all nodes, accessing data on disks attached to another node is slower because it requires moving the data across the SAN. While the SAN is a high speed connection, there is no advantage to moving heavy data traffic over it and adversely impacting the entire interconnect.

Therefore, where possible, disk drives are attached to the nodes that access the drive's data. For example, if a database is located on an external drive array and the database application is run on nodes three and four, consider attaching the external drive array to nodes three and four.

The two primary methods of configuring protected storage are external shared storage or CNM.

- Use external storage when:
 - Data storage is greater than 20 GB (consider re-mirror time)
 - Data access performance is critical
 - Storage expansion will be required
- Use CNM when:
 - The installation is cost sensitive
 - Limited storage requirements
 - Write performance is not an issue

5.1.4.1 CNM protected storage

CNM root disks are internal to the two root nodes and are mirrored using the **Veritas Volume Manager**. The advantage of CNM is low cost. CNM does not require an external disk cabinet nor does it require fibre array or other disk array controllers.

The two major disadvantages of CNM are:

- Write performance
- Vulnerability to failure during re-mirroring

When using NSC, all I/O must be performed by the node controlling the filesystem. In a CNM configuration, I/O requests originating on the non-controlling node must be sent over the SAN to the controlling node:

- In the event of a read request, the current root node returns the requested data to the requesting node over the SAN.
- In the event of a write request, the requesting node must send the data over the SAN with the write request, then the current root node must write the data to its local disk, *and* send the write request (with data) over to the potential root node.

This extra traffic for a write makes CNM less suitable for environments with a high volume of write requests.

The other major disadvantage of CNM is vulnerability to single point of failure. If either root node fails, upon rebooting, the *root* filesystem must be mirrored across the SAN before the restarted node is capable of performing as a root node. The re-mirror time is approximately 7 minutes per Gigabyte.

During the re-mirror time, the potential root node is available for work, but not to assume the role of current root node. Because of this window of vulnerability, you should limit the size of CNM disks to 20GB or less. The re-mirroring problem does not exist with the external shared solution.

5.1.4.2 External shared storage

External shared disks reside in disk cabinets. The cabinets themselves may be RAID enabled. These storage units:

- Can be attached to the nodes where primary access will occur
- Have no window of vulnerability for re-mirroring the *root* filesystem each time a root or potential-root node fails
- Are more expensive, more flexible in arrangement, and capable of more storage than CNM
- Should be used where high I/O and large data volumes are required

5.1.5 Electrical power and UPS

You must determine the cost-per-hour of potential downtime due to power loss and consider using virtual UPSes with generators as a backup power supply.

If generators are used, check them for proper configuration. Specifically check to ensure that the fuel pumps for the generators are not powered by commercial power, but by the generators themselves.

Test your UPS design before the cluster goes live.

5.2 Case Studies

The following case studies provide four different scenarios for clusters.

5.2.1 Small business or replicated site

This is the low end of the cluster spectrum. The requirements include:

- Less than 20 GB of storage is required.
- Applications easily run on a single server.
- Expansion of processing power is not a concern or can be resolved with CPU and RAM upgrades
- Low disk I/O load
- Extremely cost-sensitive customer
- Availability is very important
- Space is not a limitation

Exercise: Design the SMB replicated site cluster

Design a cluster that meets the specifications above. Include the number of nodes, protected storage option, and if it is a rack or tower configuration.

This is a good configuration for small businesses that cannot afford downtime. Businesses that are located in remote areas and require the computer to be available nights and weekends, can benefit from this approach.

5.2.2 Large storage, low processing

A step up from the previous system is the requirement for a large storage subsystem without an increase in processing power. The requirements include:

- Greater than 20 GB of storage required, perhaps into the hundreds of gigabytes. Ability to expand storage is a requirement.
- The application easily fits on a single server.
- Expansion of processing power is not a concern or can be resolved with CPU and RAM upgrades
- Low to heavy disk I/O load
- Cost sensitive installation and availability is very important

Exercise: Design the large-storage, low-processing cluster

Design a cluster that meets the specifications above. Include the number of nodes, the protected storage option and if is a rack or tower configuration.

This system can support a large database and provides failover in the event of node failure.

5.2.3 Low storage, high processing

In this scenario, the system has low storage and high processing power requirements. This is typical of a graphics-intensive or perhaps a compute-bound scientific application. The requirements include:

- Less than 20 GB of storage required
- The application(s) will not easily fit on a single server, very high processing power is required
- Expansion of storage is not an issue
- Expansion of processing power is a requirement
- Low disk I/O load
- Scalability and availability are important

Exercise: Design the low storage, high processing cluster

Design a cluster that meets the specifications above. Include the number of nodes, the protected storage option, and if it is a rack or tower configuration.

5.2.4 Heavy processing, heavy I/O

At the high of the spectrum are the large clusters, such as those used for heavy-volume e-commerce. These clusters have at least two application groups: web servers and databases. The requirement includes:

- Hundreds of gigabytes of storage
- The database application fits on a single server. Multiple copies of the web servers must be running
- Expansion of storage is an issue
- Heavy requirement for network access
- Heavy disk I/O load
- Performance-sensitive installation.
- Availability and scalability are very important.

Exercise: Design the high storage, high processing cluster

Design a cluster that meets the specifications above. Include the number of nodes, protected storage option and it is a rack or tower configuration.

5.3 Lesson Summary

This lesson described the considerations for designing a cluster. Several case studies were presented.

Review questions

1. List two considerations for designing a cluster disk configuration.
2. List two considerations for the number of nodes in a cluster
3. What is the maximum size of required storage that is recommended for CNM? Why?
4. When should a rack mount be used for a cluster?

5.4 Answers to Exercise and Review Questions

This section contains the answers to exercises, review questions, and labs presented in this lesson.

Exercise answers

“Design the SMB replicated site cluster,” on page 5-17

A two-node CNM installation is sufficient. There is no requirement for expansion, SAN switches, or other external equipment to complicate the cabling. A pair of tower systems is a good choice.

Choose servers with adequate RAM and CPUs for the application load.

“Design the large-storage, low-processing cluster,” on page 5-18

The solution is a rack mount system with external shared storage. The need for a large amount of storage, combined with high-availability requirements, rules out the use of CNM because of the limits on internal storage and the time to re-mirror.

“Design the low storage, high processing cluster,” on page 5-19

Because of the large processing power required, use the maximum number of supported nodes. NSC 7.1.1 currently supports six.

Use CNM due to the low storage capacity and I/O load.

With more than two nodes, depending upon your SAN technology, SAN switches might be required, as well as a KVM switch.

With all the cable requirements, a rack mount would be the solution. Plan expansion room for additional nodes and external storage.

“Design the high storage, high processing cluster,” on page 5-20

For this cluster, use external storage and multiple nodes in a rack mount configuration.

Use one 1+1 node pair for each database served. Only combine the databases on the same node pair where performance allows.

Use several nodes configured in a N+1 configuration for the web servers. Components will likely include a SAN switch, a KVM switch, external storage units attached to multiple node pairs, and possibly fibre channel hubs to connect multiple external storage units to one node pair.

Review question answers

This section contains the answers to the review questions on page 5-21.

1. Per Section 5.1, “Design Concepts,” on page 5-6, the issues you must consider when designing a cluster configuration include:
 - No single point of failure
 - Cluster format (tower or rack)
 - Power of machines including CPU speed, RAM, and disk size
2. Per Section 5.1.2, “Node design,” on page 5-8, the issues you must consider when deciding how many nodes to place in the cluster include:
 - Mapping of applications to nodes
 - Providing sufficient processing power to meet performance specifications
3. Per Section 5.1.4, “Disk configuration,” on page 5-14, the maximum amount of storage you should have on a CNM system is 20 MB, due to the time it takes to re-mirror a volume in the event of a node-loss.
4. Per Section 5.1.3, “Node format—tower or rack mount,” on page 5-12, you should consider using a rack-mounted system whenever the cabling or floor space required for a tower system becomes unmanageable. While the amount of cabling does not decrease, a rack provides better tools for managing the cables.

