# TruCluster Server

## Cluster LAN Interconnect

Part Number: AA-RP7CA-TE

**June 2001**

**Product Version:** TruCluster Server Version 5.1A

**Operating System and Version:** Tru64 UNIX Version 5.1A

This manual describes how to configure and administer a local area network (LAN) as a cluster interconnect.

**Compaq Computer Corporation**
**Houston, Texas**

# Contents

## Figures



## Tables

# About This Manual

This manual describes how to configure and manage local area network (LAN) hardware for use as a cluster interconnect in TruCluster™ Server Version 5.1A.

## Audience

This manual is intended for TruCluster Server administrators.

## Organization

This manual is organized as follows:

| | |
|---|---|
| *Chapter 1* | Describes the purpose of a cluster interconnect, its uses, and methods of controlling traffic over it. |
| *Chapter 2* | Provides basic information on how to configure LAN hardware as a cluster interconnect. |
| *Chapter 3* | Discusses the interconnect-specific issues that are involved in creating a cluster and adding a member to an existing cluster. |
| *Chapter 4* | Discusses the day-to-day administration of a LAN interconnect and miscellaneous configuration and management issues. |
| *Appendix A* | Discusses features of Ethernet switches that are required for a highly available LAN interconnect. |
| *Appendix B* | Contains sample installation logs for the `clu_create` and `clu_add_member` commands. |
| *Appendix C* | Lists the `/etc/sysconfigtab` attributes written by the cluster installation procedure to define the cluster interconnect. |

## Related Documents

See the following manuals for assistance in cluster hardware configuration, installation, administration, and programming tasks:

- TruCluster Server *Software Product Description* (SPD) — Provides a comprehensive description of the TruCluster Server Version 5.1A product. You can find the latest version of the SPD at the following URL: **`http://www.tru64unix.compaq.com/docs/pub_page/spds.html`**.

  You can find the latest version of the SPD at the following URL:

  `http://www.tru64unix.compaq.com/docs/pub_page/spds.html`

- *Cluster Release Notes* — Provides a brief introduction to new features in TruCluster Server and describes known problems and workarounds.
- *Cluster Technical Overview* — Provides an overview of the TruCluster Server technology.
- *Cluster Hardware Configuration* — Describes how to set up the processors that are to become cluster members, and how to configure cluster shared storage.
- *Cluster Installation* — Describes how to install the TruCluster Server software.
- *Cluster Highly Available Applications* — Describes how to run existing applications on a TruCluster Server cluster, and how to write cluster-aware applications.
- *Cluster Administration* — Describes cluster-specific administration tasks.

You can find the latest versions of the TruCluster Server documentation at the following URL: **http://www.tru64unix.compaq.com/docs/pub_page/cluster_list.html**.

We recommend that you read the Compaq Tru64™ UNIX operating system software *Release Notes*, the Tru64 UNIX *Network Administration: Connections* manual, and the Tru64 UNIX *System Administration* manual to become familiar with restrictions and new features in the base operating system before installing, configuring, and using your TruCluster Server cluster.

### Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from Compaq.) The following list describes this convention:

| | |
|---|---|
| G | Manuals for general users |
| S | Manuals for system and network administrators |
| P | Manuals for programmers |
| R | Manuals for reference page users |

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

## Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

## Conventions

The following typographical conventions are used in this manual:

| | |
|---|---|
| `%`<br>`$` | A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells. |
| `#` | A number sign represents the superuser prompt. |
| `% cat` | Boldface type in interactive examples indicates typed user input. |

| | |
|---|---|
| *file* | Italic (slanted) type indicates variable values, placeholders, and function argument names. |
| [ \| ]<br>{ \| } | In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed. |
| . . . | In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times. |
| cat(1) | A cross-reference to a reference page includes the appropriate section number in parentheses. For example, cat(1) indicates that you can find information on the cat command in Section 1 of the reference pages. |
| Return | In an example, a key name enclosed in a box indicates that you press that key. |
| Ctrl/*x* | This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C ). |

# 1

# Technical Overview

This chapter describes the purpose of a cluster interconnect, its uses, methods of controlling traffic over it, and how to decide what kind of interconnect to use. The chapter discusses the following topics:

- Understanding the role of the cluster interconnect (Section 1.1)
- Controlling storage traffic across the cluster interconnect (Section 1.2)
- Controlling application traffic across the cluster interconnect (Section 1.3)
- Controlling cluster alias traffic across the cluster interconnect (Section 1.4)
- Understanding the effect of cluster size and cluster interconnect traffic (Section 1.5)
- Selecting a cluster interconnect (Section 1.6)

See the *Cluster Technical Overview* for a general discussion of the features of the TruCluster Server and its operational components.

## 1.1 The Role of a Cluster Interconnect

A cluster must have a dedicated cluster interconnect to which all cluster members are connected. This interconnect serves as a private communication channel between cluster members. For hardware, the cluster interconnect can use either Memory Channel or a private local area network (LAN), but not both.

In general, the cluster interconnect is used for the following high-level functions:

- Health, status, and synchronization messages

  The connection manager uses these messages to monitor the state of the cluster and its members and to coordinate membership and application placement within the cluster. This type of message traffic increases during membership transitions (for example, when a member joins or leaves the cluster), but is minimal in a steady-state cluster. (See Section 1.5 for additional information.)

- Distributed lock manager (DLM) messages

  TruCluster Server uses the DLM to coordinate access to shared resources. User-level applications can also use this coordination function through the DLM application programming interface (API) library. The message traffic required to coordinate these locking functions is transmitted over the cluster's interconnect media. Although an application can make heavy use of this capability, the DLM traffic created by the cluster software itself is minimal.

- Accessing remote file systems

  TruCluster Server software presents a unified picture of the availability of storage devices across all cluster members. Storage located on one member's private storage bus is visible to all cluster members. Reads and writes from other cluster members to file systems on this storage are transmitted by means of the cluster interconnect. Whenever possible, I/O requests (reads, in particular) to files on shared storage are sent directly to the storage and bypass the cluster interconnect. How file systems and storage are configured within the cluster can significantly impact the throughput requirements placed on the cluster interconnect. (See Section 1.2 for additional information.)

- Application-specific traffic

  The cluster interconnect has a TCP/IP address associated with a virtual network interface (`ics0`) on each member. User applications can use this address to communicate over the interconnect. The load that this traffic places on the interconnect varies with the application mix. (See Section 1.3 for additional information.)

- Cluster alias routing

  While a cluster alias presents a single TCP/IP address that clients can use to reference the entire cluster or a subset of its members, the cluster alias establishes individual TCP/IP connections to processes on a given member. For example, while multiple simultaneous Network File System (NFS) operations to a cluster alias are balanced across cluster members, each individual NFS operation directed at the cluster alias is served by an NFS daemon on one member. The cluster interconnect is used when it is necessary to route the TCP/IP packets addressed to the cluster alias to the specific member that is hosting the connection. The bandwidth requirements that the cluster alias can place on the interconnect depend upon the degree to which the cluster alias is being used. (See Section 1.4 for additional information.)

Considering these high-level uses, the communications load of the cluster interconnect can be seen as being heavily influenced both by the cluster's storage configuration and by the set of applications the cluster runs.

Table 1–1 compares a LAN interconnect and a Memory Channel interconnect with respect to cost, performance, size, distance between members, support of the Memory Channel application programming interface (API) library, and redundancy. Subsequent sections discuss how to manage cluster interconnect bandwidth and make an appropriate choice of interconnect based on several factors.

**Table 1–1: Comparison of Memory Channel and LAN Interconnect Characteristics**

| Memory Channel | LAN |
|---|---|
| Higher cost | Generally lower cost |
| High bandwidth, low latency | Medium to high bandwidth and latency |
| Up to eight members, limited by the capacity of the Memory Channel hub | Up to eight members initially; will support more in the future |
| Up to 20 meters (65.6 feet) between members with copper cable; up to 2000 meters (1.2 miles) with fiber-optic cable in virtual hub mode; up to 6000 meters (3.7 miles) with fiber-optic cable using a physical hub. | Up to 200 meters (656.2 feet) between members with copper (100BASE-TX) cable (either directly connected or using a single Class I or Class II repeater (switch or hub)); up to 412 meters (1,351.7 feet) with direct-connect fiber-optic (100BASE-FX) cable. If a single Class II repeater is used to link fiber segments, the maximum distance between members is 320 meters (1,049.9 feet). If a single Class I repeater is used to link fiber segments, the maximum distance between members is 272 meters (892.4 feet). Use of an additional Ethernet switch or hub between members lessens the overall distance. |
| Supports the use of the Memory Channel application programming interface (API) library | Does not support the Memory Channel API library. Some applications may find the general mechanism, introduced in TruCluster Server Version 5.1A, for sending signals from one cluster member to another (clusterwide `kill`) sufficient for communicating between members. |
| Multirail (failover pair) redundant Memory Channel configuration | Redundancy by configuring multiple network adapters as a redundant array of independent network adapters (NetRAIN) virtual interface on each member, distributing their connections across multiple switches |

## 1.2 Controlling Storage Traffic Across the Cluster Interconnect

The Cluster File System (CFS) coordinates accesses to file systems across the cluster. It does so by designating a cluster member as the CFS server for

a given file system. The CFS server performs all accesses, reads or writes, to that file system on behalf of all cluster members.

Starting in TruCluster Version 5.1A, read accesses to a given file system can bypass the CFS server and go directly to the disk, thus not having to pass over the cluster interconnect. If all storage in the cluster is equally accessible from all cluster members, this feature minimizes the bandwidth read operations require of the cluster interconnect. Although some read accesses can bypass the interconnect, all non-direct-I/O write accesses to a file system served by another member must pass through the interconnect. To mitigate this traffic, we recommend that, where possible, applications that write large quantities of data to a file system be located on the same member that is the CFS server for that file system. Given these recommendations, the file system I/O that must traverse the interconnect is limited to remote writes. Understanding the application mix, the CFS server placement, and the volume of data that will be remotely written, can help you determine the most appropriate interconnect for the cluster.

An application, such as Oracle Parallel Server (OPS), can avoid traversing the cluster interconnect to the CFS server by having its disk writes sent directly to disk. This direct-I/O method (enabled by the application's specifying the `O_DIRECTIO` flag on a file open) asserts to CFS that the application is coordinating its own writes to this file across the entire cluster. Applications that use this feature can both increase their clusterwide write throughput to the specified files and eliminate their remote write traffic from the cluster interconnect.

This method is useful only to those applications, such as OPS, that would not otherwise obtain the performance benefit of data caching, read-aheads, or asynchronous writes. Application developers considering using this flag must be very careful, however. Setting this flag means that the operating system will not apply its normal write synchronization functions to this file for the duration of it being opened (or written) by the application. If the application does not perform its own cache management, locking, and asynchronous I/Os, severe performance degradation and data corruption can ensue.

See the *Cluster Technical Overview* and the *Cluster Administration* manuals for additional information on the use of the cluster interconnect by CFS and the device request dispatcher and on the optimizations provided by the direct-I/O feature.

## 1.3 Controlling Application Traffic Across the Cluster Interconnect

Applications use a cluster's compute resources in different ways. In some clusters, members can be considered as separate islands of computing that

share a common storage and management environment (for example, a timesharing cluster in which users are running their own programs on one system). Other applications, such as OPS, use distributed processing to focus the compute power of all cluster members onto a single clusterwide application. In this case, it is important to understand how the distributed application's components communicate:

- Do they communicate information by means of shared disk files?
- Do they communicate through direct process-to-process communications over the interconnect?
- How often do these pieces communicate and how much data is transferred per unit of time?
- What does the application require in terms of transmission latency?

With the answers to these questions, it becomes straightforward to map the application's requirements to the characteristics of the interconnect options. For example, an application that requires only 10,000 bytes per second of coordination messaging can fully utilize the compute resources of even a large cluster without stressing a LAN interconnect. On the other hand, distributed applications with high data rate and low latency requirements, such as OPS, benefit from having a Memory Channel as the interconnect, even in smaller clusters.

## 1.4 Controlling Cluster Alias Traffic Across the Cluster Interconnect

The mix of applications that will use a cluster alias, the amount of data being sent to the cluster via the cluster aliases, and the cluster network topology (for example, are members symmetrically or asymmetrically connected to the external networks?) are important factors to consider when deciding which type of cluster interconnect is appropriate.

Some common uses for the cluster alias (such as `telnet`, `ftp`, and Web hosting) typically add only small communication requirements to the interconnect. These applications are examples where the amount of data sent to the cluster's alias is generally far outweighed by the amount of data returned to clients from the cluster. Only the incoming data packets might need to traverse the interconnect to reach the process serving the request. All outgoing packets go directly to the external network and thus do not have to be conveyed over the interconnect. (This presumes that all members have connectivity to the external network.) Applications such as these, in most cases, place low bandwidth requirements on the interconnect.

The Network File System (NFS), on the other hand, is a commonly used application that can place a significant bandwidth requirement on the cluster interconnect. While reads from the served disks do not cause much

interconnect traffic (only the read request itself potentially traverses the interconnect), disk writes through NFS can create interconnect traffic. In this case, the incoming data that might need to be delivered over the interconnect is comprised of disk blocks. If the cluster is going to serve NFS volumes, compare the average rate that disk writes are likely to occur with the bandwidth offered by the various interconnect options.

TruCluster Server Version 5.1A introduces a feature that can lessen the impact of NFS writes. For the purposes of NFS serving, you can assign alternate cluster aliases to subsets of cluster members. This allows a selected set of cluster members to be identified as the NFS servers, thus lowering the average number of inbound packets that must be sent over the interconnect to reach that connection's serving process. (In a randomly distributed four-member cluster, an average of 75 percent of the disk writes will traverse the interconnect. If two of those members are assigned an alternate cluster alias for their NFS serving, the average number of writes traversing the interconnect drops to 50 percent.)

See the *Cluster Technical Overview* and the *Cluster Administration* manuals for information on how to use and tune a cluster alias.

## 1.5 Effect of Cluster Size on Cluster Interconnect Traffic

You cannot consider solely the number or size of the members in a cluster when determining the most appropriate interconnect, but must also look at how the cluster's use will affect the load placed on the interconnect. Although larger clusters tend to have higher data transfer requirements for a given application mix, how the cluster's storage is configured and the characteristics of its applications are better guides to determining the proper interconnect. However, one aspect of cluster size can impact the interconnect bandwidth requirements. Presuming a perfectly random (and unmanaged) distribution of work across the cluster and an equally random distribution of CFS servers, the percentage of disk writes that must traverse the cluster interconnect increases as the cluster size increases. In a two-member cluster, for example, 50 percent of the average writes might go over the interconnect. In a four-node cluster, this increases to 75 percent. In Section 1.2 we recommend the system that will be performing most writes to a file system be the CFS server for that file system. This recommendation minimizes the number of writes that must be sent over the interconnect and is appropriate regardless of which type of interconnect is used. To the degree that you can meet this recommendation, the less interconnect bandwidth the disk writes will require.

However, there is one situation in which the size of the cluster (measured both in terms of the number of members and the number of disks in use) has a direct impact on the interconnect traffic: cluster membership transitions. In particular, when a member leaves the cluster, the remaining members

must pass coordination messages to the other cluster members. Due to the lower latency characteristics of the Memory Channel interconnect, these transitions can be completed faster on a Memory Channel-based cluster. When deciding which interconnect to use, consider how often you expect membership transitions to occur (for example, whether cluster members will routinely be rebooted).

## 1.6 Selecting a Cluster Interconnect

In addition to the recommendations provided in the previous sections, the following rules and restrictions apply to the selection of a cluster interconnect:

- All cluster members must be configured either to use a LAN interconnect or to use Memory Channel. You cannot mix interconnect types within a cluster.
- Applications using the Memory Channel API library require Memory Channel. A cluster using a LAN interconnect can also be configured with a Memory Channel that is used by Memory Channel API applications only. Note that use of the Memory Channel API also generates some slight TCP/IP traffic over the cluster interconnect.
- A LAN interconnect is required when configuring one or more AlphaServer™ DS10L systems in a cluster. An AlphaServer DS10L system is shipped with two 10/100 Mb/s Ethernet ports, one 64-bit peripheral component interconnect (PCI) expansion slot, and a fixed internal integrated device electronic (IDE) disk. When you configure an AlphaServer DS10L in a cluster, we recommend that you use the single PCI expansion slot for the shared storage (where the cluster root, member boot disks, and optional quorum disk reside), one Ethernet port for the external network, and the other Ethernet port for the LAN interconnect. See Section 2.2.4 for a description of cluster configurations including AlphaServer DS10L systems.
- Replacing a Memory Channel interconnect with a LAN interconnect (or vice versa) requires some cluster downtime. Section 4.5 describes how to migrate from Memory Channel to a LAN interconnect.
- Although the Logical Storage Manager (LSM) provides for transparent mirroring and highly available access to storage, LSM is not a suitable data replication technology in an extended cluster. Although a disaster-tolerant configuration using a LAN-based or Memory Channel-based interconnect and LSM is not supported, there are supported configurations using the StorageWorks™ Data Replication Manager (DRM) solution.

# 2

# Hardware Configuration

This chapter provides basic information on how to configure local area network (LAN) hardware for use as a cluster interconnect. It discusses the following topics:

- Configuration guidelines (Section 2.1)
- Supported configurations and configuration examples (Section 2.2)

This chapter focuses on configuring LAN hardware as a cluster interconnect. For full cluster and storage configuration information, see the *Cluster Hardware Configuration* manual.

## 2.1 Configuration Guidelines

Any Ethernet adapter, switch, or hub that works in a standard LAN at 100 Mb/s should work within a LAN interconnect.

**Note**

Fiber Distributed Data Interface (FDDI), ATM LAN Emulation (LANE), 10 Mb/s Ethernet, and Gigabit Ethernet are not supported in a LAN interconnect.

The following features are required of Ethernet hardware participating in a cluster LAN interconnect:

- The LAN interconnect must be private to cluster members. A packet that is transmitted by one cluster member's LAN interconnect adapter can be received only by other members' LAN interconnect adapters.
- A LAN interconnect can be a single direct full-duplex connection between two cluster members or can employ either switches or hubs (but not both). One or more switches are required for a cluster of three or more members and for a cluster whose members use a redundant array of independent network adapters (NetRAIN) virtual interface for their cluster interconnect device.

**Note**

Although hubs and switches are interchangeable in most LAN interconnect configurations, switches are recommended for performance and scalability. Because most hubs run in half-duplex mode, their use in a LAN interconnect may limit cluster performance. Additionally, hubs do not provide the features (described in Appendix A) required for a dual redundant LAN interconnect configuration. Overall, using a switch, rather than a hub, in a LAN interconnect provides greater scalability for clusters with three or more members.

- Adapters and switch ports must be configured compatibly with respect for 100 Mb/s full-duplex operation.. If you are using a switch with any of the DE60*x* family of adapters (which have a console name of the form `ei`*x*`0`), use a switch that supports autonegotiation. If you are using a switch with network adapters in the DE50*x* family (which have a console name of the form `ew` *x*`0`) that do not autonegotiate properly, the switch must be capable of disabling autonegotiation. (See Section 4.7.1 for a discussion of the symptoms of misconfigured LAN hardware.)
- If you use two crossover cables to link two switches in a fully redundant LAN cluster interconnect (Figure 2–3 and Figure 2–4), you must configure the switches to avoid packet-forwarding problems caused by the routing loop created by the second link. Typical switches provide at least one of the following three mechanisms for support of parallel interswitch links. In order of decreasing desirability for cluster configurations, the mechanisms are:

| | |
|---|---|
| Link aggregation | Treats multiple physical links between a pair of switches as a single link and distributes packet traffic among them. |
| Link resiliency | Treats multiple physical links between a pair of switches as an active link and one or more standby links and fails over between them. |
| Spanning Tree Protocol | Employs a distributed routing algorithm that allows switches to cooperate to discover and remove routing loops. |

  See Appendix A for a detailed discussion of the switch requirements and configuration options appropriate to each mechanism.
- Although it may be used to eliminate routing loops on switch ports used for parallel links between switches, Spanning Tree Protocol (STP) must

be disabled on all Ethernet switch ports connected to cluster members, whether the members are using single adapters or multiple adapters included in NetRAIN devices. If this is not the case, cluster members will be flooded by broadcast messages which, in effect, create denial-of-service symptoms in the cluster. See Section 4.7.5 for additional information.

- All cluster members must have at least one point-to-point connection to all other members. If the Ethernet adapters that are used for the LAN interconnect fail on a given member, that member loses communication with all other members. A cluster interconnect configuration that requires a member to route interconnect traffic from another member to a different subnet is unsupported. That is, you cannot replace a switch with a member system.
- Up to two switches are allowed between two cluster members. You must not introduce unacceptable latencies by using, for example, a satellite uplink or a wide area network (WAN) as the path between two components of a LAN interconnect.
- Link aggregation of Ethernet adapters using Tru64 UNIX features (including the `lagconfig` command) is not supported for a LAN interconnect. Link aggregation of adapters is supported for the cluster's external networks.
- To simplify management, configure the LAN interconnect network adapters symmetrically on all cluster members. Installing the same type of adapter in each member in the same relative position with respect to other network adapters helps ensure that the adapters have similar names across cluster members. In a fully redundant LAN interconnect configuration using two or more interconnected switches, and NetRAIN virtual interfaces as member interconnect devices, you should uniformly connect the first network adapter listed in each member's NetRAIN set to the first switch and the second network adapter to the second switch. This simplifies the identification of the adapters for monitoring and maintenance. Additionally, it ensures that the active adapters of each member are connected to the same switch when the cluster is initially booted. As discussed in Section 2.2.3, one method for guarding against a network partition of the cluster in certain failure conditions is to ensure that all active adapters in the LAN interconnect are connected to the same switch.

## 2.2 Supported LAN Interconnect Configurations

TruCluster Server currently supports up to eight members in a cluster, regardless of whether its cluster interconnect is based on LAN or Memory Channel. Chapter 1 of the *Cluster Hardware Configuration* manual illustrates some cluster configurations using Memory Channel. The

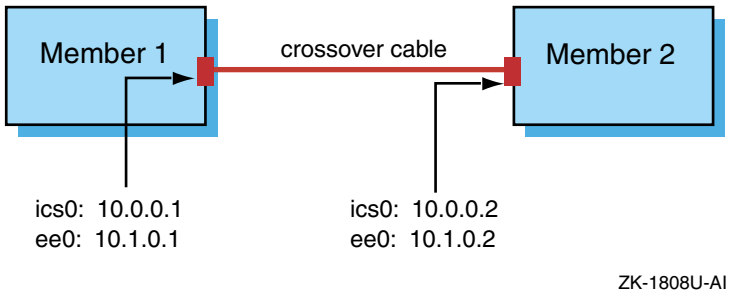
following sections supplement that chapter by discussing the supported LAN interconnect configurations:

- A single crossover cable directly connecting the Ethernet adapter of one member to the Ethernet adapter of a second member (two-member cluster only) (Section 2.2.1)
- A single switch connecting two to eight members (Section 2.2.2)
- Two switches (with one or more crossover cables between them), with two or more Ethernet adapters on each member, configured as a NetRAIN virtual interface but each connected to a different switch (Section 2.2.3)
- Clustered AlphaServer DS10L systems (Section 2.2.4)

### 2.2.1 Two Cluster Members Directly Connected by a Single Crossover Cable

You can configure a LAN interconnect in a two-member cluster by using a single crossover cable to connect the Ethernet adapter of one member to that of the other, as shown in Figure 2–1. (See Section 3.1.2 for an explanation of the IP addresses shown in the figure.)

**Figure 2–1: Two Cluster Members Directly Connected by a Single Crossover Cable**




**Note**

A crossover cable for point-to-point Ethernet connections is required to directly connect the network adapters of two members when no switch or hub is configured between them.

From a member's perspective, because this cluster does not employ redundant LAN interconnect components (each member has a single Ethernet adapter and a single cable connects the two members), a break in the LAN interconnect connection (for example, the servicing of a member's Ethernet adapter or a detached cable) will cause a member to leave the

cluster. However, if you configure a voting quorum disk in this cluster, the cluster itself will survive the failure of either member or of the quorum disk, or a break in the LAN interconnect connection. Similarly, if you configure one member with a vote and the other with no votes, the cluster will survive the failure of the nonvoting member or of its LAN interconnect connection.

You can expand this configuration by adding a switch between the two members. A switch is required in the following cases:

- When the cluster expands beyond two members (for example, the configuration discussed in Section 2.2.2).
- When you add a second Ethernet adapter to each member in order to configure the cluster interconnect device as a NetRAIN virtual interface. Merely adding the second adapters and a second crossover cable link does not provide the connectivity required for NetRAIN failover in all circumstances and is not supported.

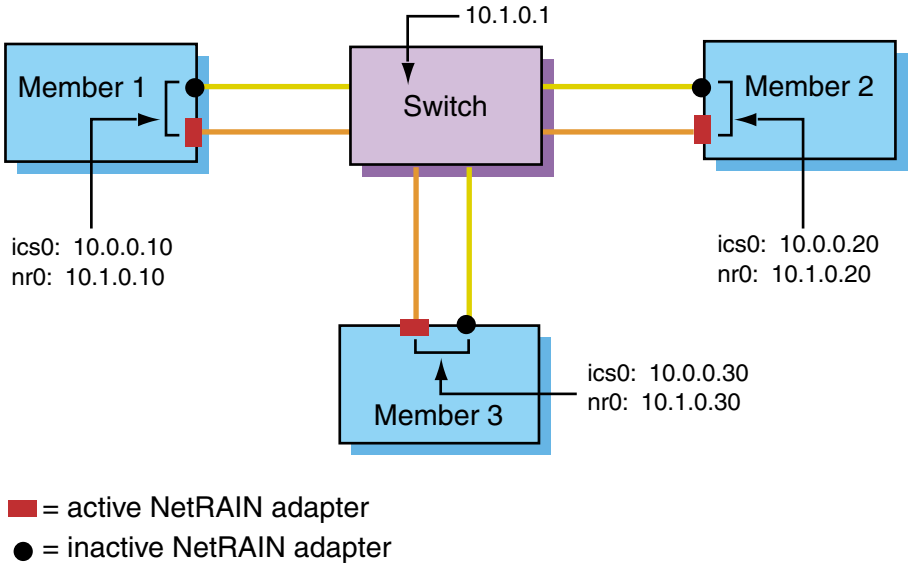
### 2.2.2 Cluster Using a Single Ethernet Switch

You can configure a cluster with a single Ethernet hub or switch connecting two through eight members. For optimal performance, we recommend a switch for clusters of three or more members.

Any member that has multiple Ethernet adapters can configure them as a NetRAIN set to be used as its LAN interconnect interface. Doing so allows those members to remain cluster members even if they lose one internal connection to the LAN interconnect.

The three-member cluster in Figure 2–2 uses a LAN interconnect incorporating a single Ethernet switch. Each member's cluster interconnect is a NetRAIN virtual interface consisting of two network adapters. (See Section 3.1.2 for an explanation of the IP addresses shown in the figure.)

**Figure 2–2: Three-Member Cluster Using a Single Ethernet Switch**




Assuming that each member has one vote, this cluster can survive the failure of a single member or a single break in a member's LAN interconnect connection (for example, the servicing of an Ethernet adapter or a detached cable). From a member's perspective, any member can survive a single break in its LAN interconnect connection. However, the servicing or failure of the switch will make the cluster nonoperational. The switch remains a single point of failure in a cluster of any size, except when it is used in one of the recommended two-member configurations using a quorum disk discussed in Section 2.2.1. For this reason, the cluster in Figure 2–2 is not a recommended configuration.

By adding a second switch to this cluster, and connecting a LAN interconnect adapter from each member to each switch (as discussed in Section 2.2.3), you can eliminate the switch as a single point of failure and increase cluster reliability.

### 2.2.3 Cluster Using Fully Redundant LAN Interconnect Hardware

You can achieve a fully redundant LAN interconnect configuration by using NetRAIN and redundant paths from each member through interconnected switches. In the four-member cluster in Figure 2–3 and Figure 2–4, two Ethernet adapters on each member are configured as a NetRAIN virtual interface, two switches are interconnected by two crossover cables, and the Ethernet connections from each member are split across the switches.

**Figure 2–3: Recommended Fully Redundant LAN Interconnect Configuration Using Link Aggregation or Link Resiliency**

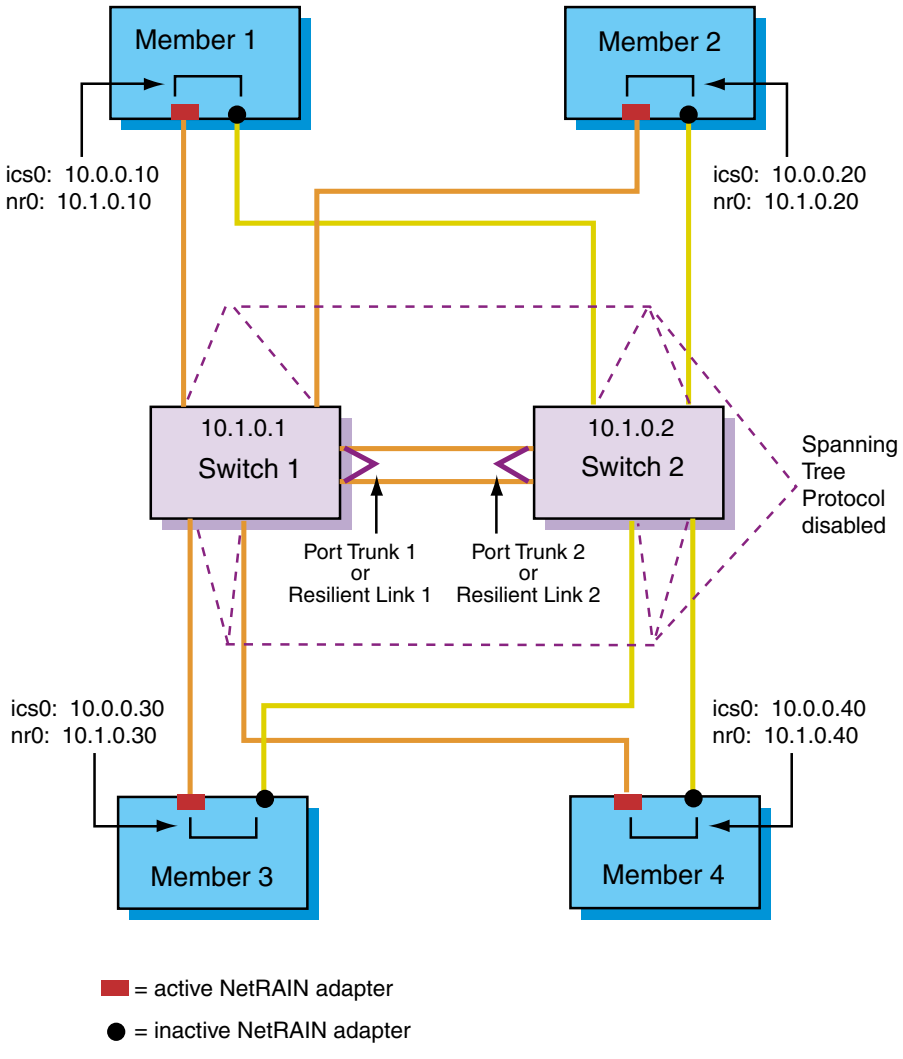
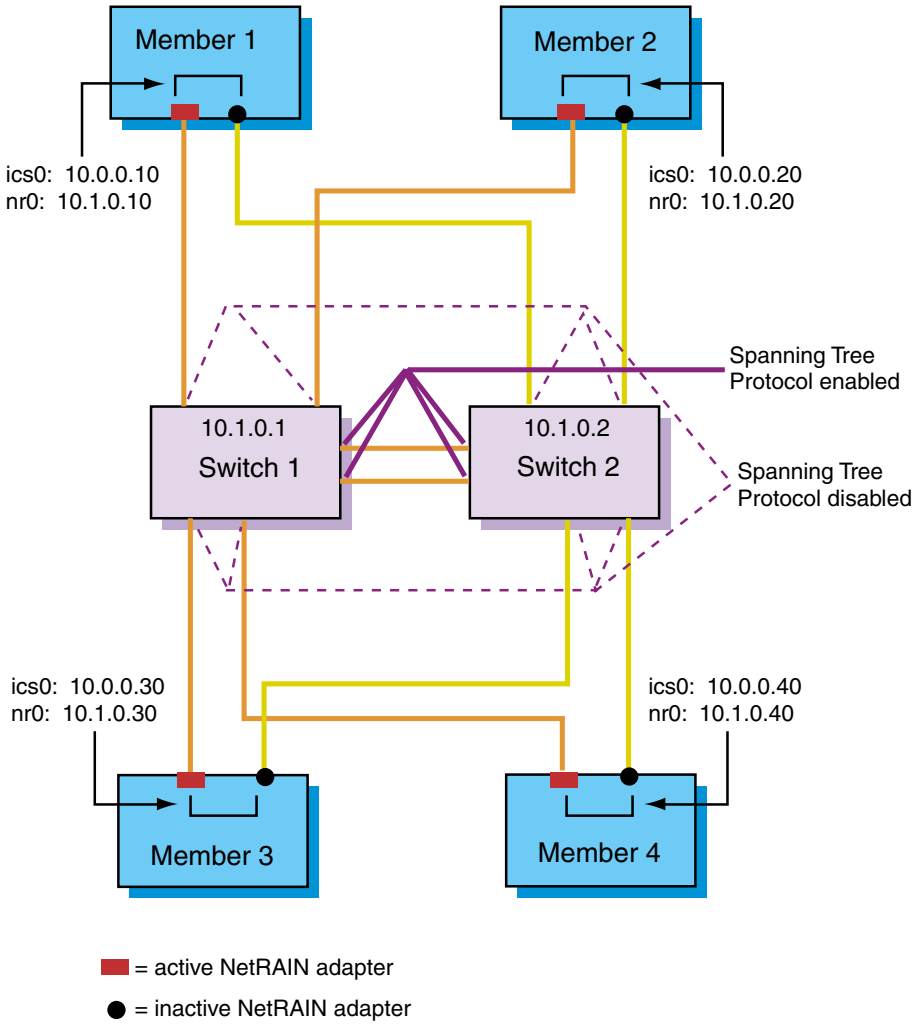


**Figure 2–4: Recommended Fully Redundant LAN Interconnect Configuration Using the Spanning Tree Protocol**




**Note**

If you are mixing switches from different manufacturers, consult with your switch manufacturers for compatibility between them.

Like the three-member cluster discussed in Section 2.2.2, this cluster can tolerate the failure of a single member or a single break in a member's LAN

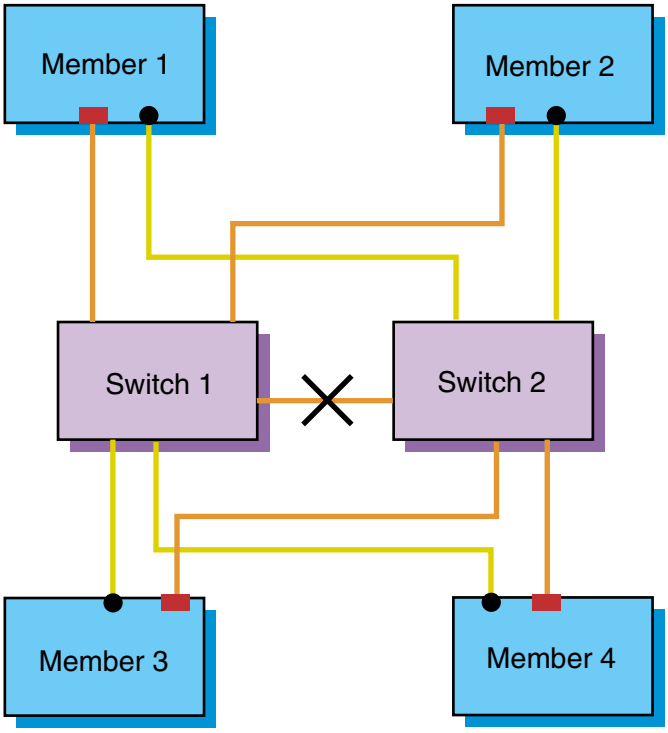
interconnect connection (for example, the servicing of an Ethernet adapter or a detached cable). (This assumes that each member has one vote and no quorum disk is configured.) However, this cluster can also survive a single switch failure and the loss of the crossover cables between the switches.

Because NetRAIN must probe the inactive LAN interconnect adapters across switches, the crossover cable connection between the switches is important. Two crossover cables are strongly recommended. When two crossover cables are used, as shown in Figure 2–3 and Figure 2–4, the loss of one of the cables is transparent to the cluster. As discussed in Appendix A, when using parallel interswitch links in this manner, it is important to employ one of the methods provided by the switches for detecting or avoiding routing loops between the switches. These figures indicate the appropriate port settings with respect to the most common methods provided by switches: link aggregation (also known as port trunking), link resiliency (both shown in Figure 2–3), and Spanning Tree Protocol (STP) (shown in Figure 2–4). (See Section 3.1.2 for an explanation of the IP addresses shown in the figure.)

In some circumstances (like the nonrecommended configuration, shown in Figure 2–5, that uses a single crossover cable), a broken crossover connection can result in a network partition. If the crossover connection is completely broken, its loss prevents NetRAIN from sending packets to the inactive adapters across the crossover connection. Although this situation will not cause the cluster to fail, it will disable failover between the adapters in the NetRAIN sets.

For example, in the configuration shown in Figure 2–5 the active LAN interconnect adapters of Members 1 and 2 are currently on Switch 1; those of Members 3 and 4 are on Switch 2. If the crossover connection is broken while the cluster is in this state, Members 1 and 2 can see each other but cannot see Members 3 and 4 (and thus will remove them from the cluster). Members 3 and 4 can see each other but cannot see Members 1 and 2 (and thus will remove them from the cluster). By design, neither cluster can achieve quorum; each has two votes out of a required three, and both will hang in quorum loss.

**Figure 2–5: Nonrecommended Redundant LAN Interconnect Configuration**




To decrease a cluster's vulnerability to network partitions in a dual-switched configuration, take any or all of the following steps:

- Configure the cluster with two crossover cables between the switches, as shown in Figure 2–3. This configuration reduces vulnerability to a network partition, but requires that the switches be additionally configured to avoid packet-forwarding problems caused by the routing loop created by the second link. See Appendix A for a detailed discussion of the switch requirements and configuration mechanisms.
- To avoid a cluster hang due to quorum loss that can occur when a cluster encounters a network partition, configure the cluster with an odd number of votes, either by providing an odd number of voting members or a voting quorum disk.
- After performing network maintenance (for example, when replacing cables or adapters) or at any other time you believe that NetRAIN

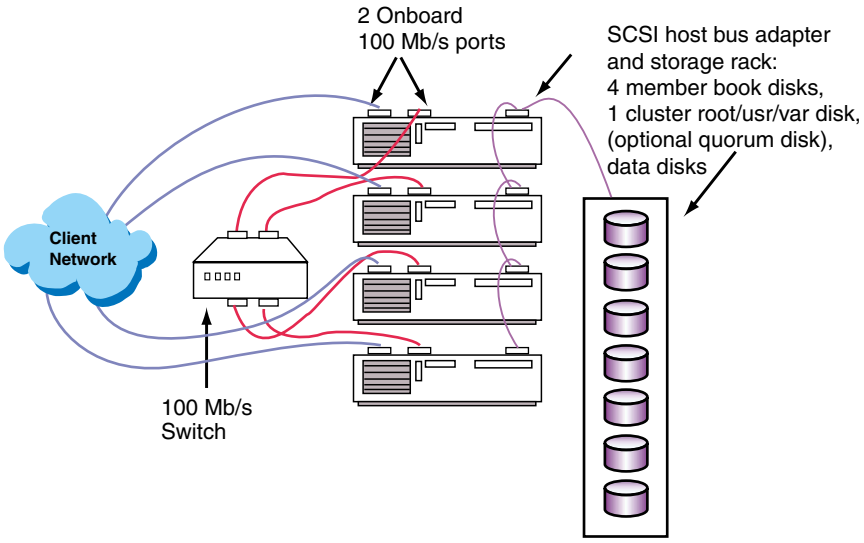
failover has occurred, examine the connectivity of the active network adapters on the NetRAIN devices on the cluster interconnect. On each member, issue an `ifconfig` command on the LAN interconnect's NetRAIN virtual interface to ensure that the active LAN interconnect adapter on each member is connected to the same switch. Uniformly connecting each member's first network adapter to the first switch and its second network adapter to the second switch facilitates identifying the member adapters that are connected to a given switch. If the active adapters are split across the switches, use the `ifconfig nr`*x* `switch` command, as appropriate, to consolidate them on a single switch.

### 2.2.4 Clustering AlphaServer DS10L Systems

Support for the LAN interconnect makes it possible to cluster more basic AlphaServer systems, such as the Compaq AlphaServer DS10L. The AlphaServer DS10L is an entry-level system that ships with two 10/100 Mb/s Ethernet ports, one 64-bit PCI expansion slot, and a fixed internal IDE disk. The 44.7 x 52.1 x 4.5-centimeter (17.6 x 20.5 x 1.75-inch (1U)) size of the AlphaServer DS10L, and the ability to rackmount large numbers of them in a single M-series cabinet, make clustering them an attractive option, especially for Web-based applications.

When you configure an AlphaServer DS10L in a cluster, we recommend that you use the single PCI expansion slot for the host bus adapter for shared storage (where the cluster root, member boot disks, and optional quorum disk reside), one Ethernet port for the external network, and the other Ethernet port for the LAN interconnect. Figure 2–6 shows a very basic low-end cluster of this type consisting of four AlphaServer DS10Ls.

**Figure 2–6: Low-End AlphaServer DS10L Cluster**

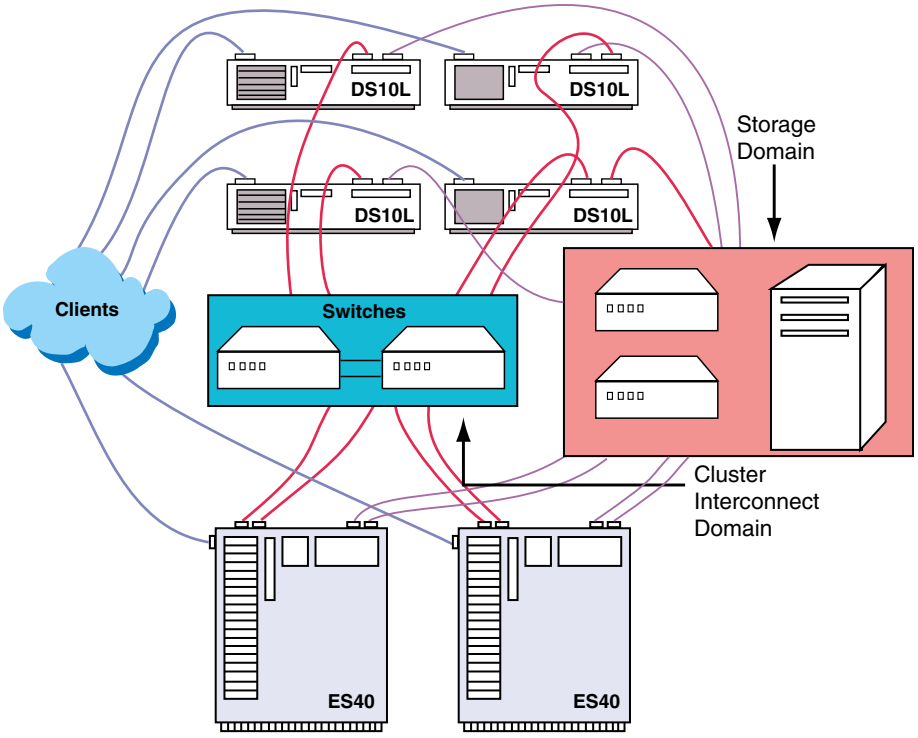



Although the configuration shown in Figure 2–6 represents an inexpensive and useful entry-level cluster, its LAN interconnect and shared SCSI storage bus present single points of failure. That is, if the shared storage bus or the LAN interconnect switch fails, the cluster becomes unusable.

To eliminate these single points of failure, the configuration in Figure 2–7 adds two AlphaServer ES40 members to the cluster, plus two parallel interswitch connections. Two AlphaServer DS10L members are connected via Ethernet ports to one switch on the LAN interconnect; two are connected to the other switch. A Fibre Channel fabric employing redundant Fibre Channel switches replaces the shared SCSI storage in the previous configuration.

Although not distinctly shown in the figure, the host bus adapters of two DS10Ls are connected to one Fibre Channel switch; those of the other two DS10Ls are connected to the other Fibre Channel switch.

**Figure 2–7: Cluster Including Both AlphaServer DS10L and AlphaServer ES40 Members**




The physical LAN interconnect device on each of the two AlphaServer ES40 members consists of two Ethernet adapters configured as a NetRAIN virtual interface. On each ES40, one adapter is cabled to the first Ethernet switch and the other is cabled to the second Ethernet switch. Similarly, each ES40 contains two host bus adapters connected to the Fibre Channel fabric. On each, one adapter is connected to the first Fibre Channel switch, the other is connected to the second Fibre Channel switch.

When delegating votes in this cluster, you have a number of possibilities:

- Assign one vote to each AlphaServer ES40 member and no votes to the AlphaServer DS10L members. Configure a quorum disk with a vote on the shared storage. This cluster can survive the loss of any one AlphaServer ES40 member, the quorum disk, or any or all AlphaServer DS10L members.
- Assign one vote to each member. Configure a quorum disk with a vote on the shared storage. This cluster can survive the loss of one or both of the AlphaServer ES40 members or the loss of three DS10L members.

(In other words, the AlphaServer ES40 members require the votes of at least one AlphaServer DS10L member, plus the quorum disk vote, to maintain quorum.)

# 3

# Software Installation

This chapter describes the interconnect-specific issues that are involved in creating a cluster or adding a member to an existing cluster. It discusses the following topics:

- Preparing to configure a LAN interconnect (Section 3.1)
- Creating a single-member cluster using the `clu_create` command (Section 3.2)
- Adding members to the cluster using the `clu_add_member` command (Section 3.3)

**Note**

This information complements information in chapters 1 through 5 of the *Cluster Installation* manual. See that manual for full procedural discussions of cluster installation.

In this release, there are some limitations on the cluster software's ability to facilitate the configuration of a LAN interconnect by performing hardware probes, enforcing configuration restrictions (such as 100 Mb/s speed), and detecting illegal and unwise configurations. For example, the cluster configuration scripts do not probe for all existing network adapters on a member. Rather, the `clu_create` command prompts you for the name of an adapter and verifies:

- That it exists
- That it is unconfigured

The `clu_add_member` command prompts you for an adapter name and verifies its syntax. Because `clu_add_member` runs on an existing cluster member before the new member has been booted, it cannot verify the existence of an adapter on the member that it is adding.

Finally, there are no configuration tests for the LAN interconnect in the `clu_check_config` utility. If you misconfigure the LAN interconnect in a cluster (for example, by specifying nonexistent adapters or NetRAIN virtual interfaces), the system may not be able to boot and form or join a cluster. (See Section 4.7 for information on how to detect and resolve such problems.)

## 3.1 Preparation

Before running the `clu_create` and `clu_add_member` commands to configure a cluster using a LAN interconnect, perform the following tasks. (If you are migrating from Memory Channel to a LAN interconnect, see Section 4.5.)

- Make sure that the `/vmunix` kernel contains support for the Ethernet hardware you have connected. If it does not, you must boot `/genvmunix` and rebuild the `/vmunix` kernel using the `doconfig -c` command.
- Configure the Ethernet hardware intended for use as the LAN interconnect so that it can be used as a standard network. Use various networking utilities like `ifconfig`, `ping`, `ftp`, and `telnet` to verify that the hardware is set up correctly.
- Obtain the device names of the physical Ethernet network adapters on each member system to be used for the LAN interconnect (Section 3.1.1).
- Obtain IP names and addresses for the cluster interconnect virtual interface (`ics0`) and the cluster interconnect physical interface (Section 3.1.2).

### 3.1.1 Obtain the Device Names of the Network Adapters

Obtain the names of eligible Ethernet network adapters on the member to be configured before issuing the `clu_create` or `clu_add_member` command. To be eligible, an adapter must:

- Be installed
- Not be configured
- Be set to run at 100 Mb/s

The cluster installation commands accept the names of either physical Ethernet network adapters or NetRAIN virtual interfaces.

**Caution**

The cluster installation commands automatically configure the NetRAIN virtual interfaces for the LAN interconnect. Do not manually create the NetRAIN devices prior to running the `clu_create` script. See Section 4.1 for a discussion of the consequences of doing so.

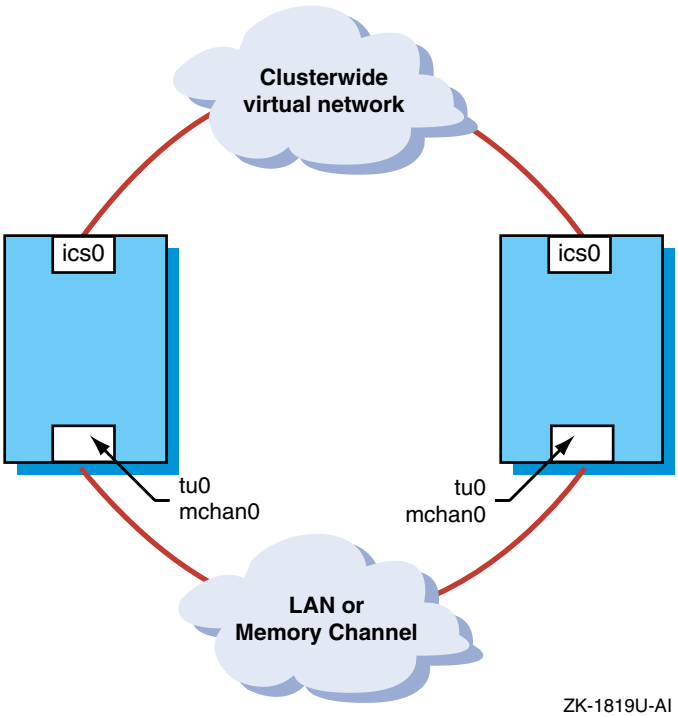
To learn the device names of eligible network adapters, run the `ifconfig -a` command on the system that will become the first member of the cluster. Use the `hwmgr -get attr -cat network` command to determine their speed and transmission mode.

To learn the device names for systems that you intend to add to the cluster, you must first boot the system from the Tru64 UNIX *Operating System Volume 1* CD-ROM. The UNIX device names of the Ethernet adapters scroll on the console during the boot process. If you enter a UNIX shell after the system boots, you can enter an `ifconfig -a` command to list the network adapter device names and the `hwmgr -get attr -cat network` command to list their properties.

### 3.1.2 Obtain IP Names and IP Addresses for Each Member's Cluster Interconnect

To allow cluster members to use TCP/IP mechanisms to communicate over the cluster interconnect, regardless of its underlying hardware, the cluster software creates a virtual network within the cluster (Figure 3–1).

**Figure 3–1: Cluster Virtual Network and Physical Communication Channel**




This virtual network exists side-by-side with the physical communications channel provided by the cluster interconnect.

For each member, the cluster software establishes a virtual network device for the cluster interconnect. This device is named `ics0` and its IP name and IP address are used when establishing the system's membership in the

cluster. This name and address represent a member's cluster interconnect in the IFCONFIG and NETDEV entries in its /etc/rc.config file.

**Note**

For TruCluster Server Version 5.1A, the name of a member's cluster interconnect virtual device has changed from mc0 to ics0. If you perform a full installation of Version 5.1A, or perform a rolling upgrade (as described in the *Cluster Installation* manual) from Version 5.1, the NETDEV_*x* configuration variable in each member's /etc/rc.config file that corresponds to this device will be defined as ics0.

Similarly the form of a member's default cluster interconnect IP name offered by the cluster installation scripts (clu_create and clu_add_member) has also changed. The default cluster interconnect IP name is visible in the value of the CLUSTER_NET configuration variable in the each member's /etc/rc.config file and in the value of the cluster_node_inter_name variable of the clubase kernel subsystem in the each member's /etc/sysconfigtab file. If you perform a full installation of Version 5.1A, the default for these attributes (formerly *member-name*-mc0) will be offered as *member-name*-ics0. If you perform a rolling upgrade to Version 5.1A, their file values remain *member-name*-mc0.

The number of IP names and IP addresses required for the cluster interconnect thus depends upon the type of cluster interconnect:

- For a cluster using Memory Channel, you need an IP name and IP address for the virtual cluster interconnect device on each member system.

  By default, the installation programs offer IP addresses on the 10.0.0 subnet for virtual cluster interconnect, with the host portion of the address set to the member ID and the IP name set to the short form of the member's host name followed by -ics0.

- For a cluster using a LAN interconnect, where communications between members traverse two TCP/IP layers, you need:

  – An IP name and IP address for the virtual cluster interconnect device on each member system.

    By default, the installation programs offer IP addresses on the 10.0.0 subnet for virtual cluster interconnect, with the host portion of the

address set to the member ID and the IP name set to the short form of the member's host name followed by `-ics0`.

– An IP name and IP address on a different subnet for the physical LAN interface.

  By default, the cluster installation programs offer IP addresses on the 10.1.0 subnet for the physical cluster interconnect, with the host portion of the address set to the member ID. The IP name is set to `member`*member-ID*`-icstcp0`.

**Notes**

Manufacturers typically associate a default address with an Ethernet switch to facilitate its management. This address may conflict with the default IP address the cluster installation scripts provide for the virtual cluster interconnect device or the physical LAN interface. In this case, you must ensure that the IP addresses selected for the cluster interconnect differ from that used by the switch. For example, in Figure 2–3, because the switch is addressable by the IP address 10.1.0.1, we have assigned the address 10.1.0.100 to member 1's physical LAN interface.

By default, the installation programs use Class C subnet masks for the IP addresses of both the virtual and physical LAN interconnect interfaces.

Cluster interconnect IP addresses cannot end with either `.0` or `.255`. Addresses of this type are considered broadcast addresses. A system with this type of address cannot join a cluster.

The following example shows the cluster interconnect IP names and IP addresses for two members of the `deli` cluster, `pepicelli` and `polishham`, which is running on a Memory Channel cluster interconnect:

```
10.0.0.1 pepicelli-ics0  # first member's  virtual interconnect IP name and address
10.0.0.2 polishham-ics0  # second member's virtual interconnect IP name and address
```

The following example shows the cluster interconnect IP names and IP addresses for two members of the same cluster running on a LAN interconnect:

```
# first member's cluster interconnect virtual interface IP name and address
10.0.0.1 pepicelli-ics0
# first member's cluster interconnect physical interface IP name and address
10.1.0.1 member1-icstcp0
# second member's cluster interconnect virtual interface IP name and address
10.0.0.2 polishham-ics0
```

```
# second member's cluster interconnect physical interface IP name and address
10.1.0.2 member2-icstcp0
```

The cluster installation scripts mark both the cluster interconnect virtual interface and physical interface with the cluster interface (CLUIF) flag. For example, the following output of the ifconfig -a command shows the cluster interconnect virtual interface (ics0) and the cluster interconnect physical interface (ee0):

```
# ifconfig -a | grep -p CLUIF
ee0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     inet 10.1.0.2 netmask ffffff00 broadcast 10.1.0.255 ipmtu 1500
ics0: flags=1100063<UP,BROADCAST,NOTRAILERS,RUNNING,NOCHECKSUM,CLUIF>
     inet 10.0.0.2 netmask ffffff00 broadcast 10.0.0.255 ipmtu 1500
```

The following example shows a cluster interconnect physical interface (nr0 that is a NetRAIN virtual interface:

```
# ifconfig -a | grep -p CLUIF
ee0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Virtual Interface: nr0
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )
ee1: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Virtual Interface: nr0
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )
ics0: flags=11000c63<BROADCAST,NOTRAILERS,NOCHECKSUM,CLUIF>
     inet 10.0.0.2 netmask ffffff00 broadcast 10.0.0.255 ipmtu 1500
nr0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )
     inet 10.1.0.2 netmask ffffff00 broadcast 10.1.0.255 ipmtu 1500
```

## 3.2 Create a Single-Member Cluster

When you create a cluster, the clu_create command prompts for the type of cluster interconnect type (LAN or Memory Channel), offering Memory Channel as a default if a Memory Channel adapter is installed:

- If you specify Memory Channel as the cluster interconnect type, clu_create offers a default physical cluster interconnect interface of mchan0.
- If you specify LAN, the clu_create command prompts you for a physical cluster interconnect device name. You have the following options:
  - Specify the device name of a single network interface, such as tu0 or ee0.
  - Iteratively specify the device names of multiple network interfaces. The clu_create command allows you to specify that these interfaces be configured in a NetRAIN virtual interface.

**Note**

If you specify the device name of an existing NetRAIN device (for example, one defined in the `/etc/rc.config` file), the `clu_create` command prompts you to confirm that you want to redefine this NetRAIN device as a physical cluster interconnect device. If you respond "yes," the `clu_create` command removes the definition of the NetRAIN device from the `/etc/rc.config` file and defines the device as the cluster interconnect device in the `ics_ll_tcp` stanza of the `/etc/sysconfigtab` file.

The `clu_create` command then creates an IP name for the physical cluster interconnect device of the form `member`*member-ID*`-icstcp0`, and by default offers an IP address of `10.1.0.`*member-ID* for this device.

See Appendix C for a list of `/etc/sysconfigtab` attributes written by the `clu_create` command to define the cluster interconnect.

## 3.3 Add Members

When you add a member to an existing cluster, the `clu_add_member` command prompts you for a physical cluster interconnect device name for the LAN interconnect (if the current cluster member was not configured to use Memory Channel). You have the following options:

- Specify the device name of a single network interface, such as `tu0` or `ee0`.
- Iteratively specify the device names of multiple network interfaces. The `clu_add_member` command allows you to specify that these interfaces be configured in a NetRAIN virtual interface.

**Note**

If you specify the device name of a NetRAIN device that is defined as the physical cluster interconnect device for the member on which you are running the `clu_add_member` command, the command prompts you to indicate whether you intend to use an identical NetRAIN device (same device name and same participating adapters) on the member you are adding. If you respond "yes," the `clu_add_member` command defines the device as the cluster interconnect device in the `ics_ll_tcp` stanza of the `/etc/sysconfigtab` file.

The `clu_add_member` command then creates an IP name for the physical cluster interconnect device of the form `member`*member-ID*`-icstcp0` and by default offers an IP address of `10.1.0.`*member-ID* for this device.

See Appendix C for a list of `/etc/sysconfigtab` attributes written by the `clu_add_member` command to define the cluster interconnect.

# 4

# Cluster Administration

This chapter discusses the following topics:

- Configuring a NetRAIN virtual interface for a cluster interconnect (Section 4.1)
- Tuning the LAN interconnect for optimal performance (Section 4.2)
- Obtaining network adapter configuration information (Section 4.3)
- Monitoring activity on the LAN interconnect (Section 4.4)
- Migrating from Memory Channel to a LAN interconnect (Section 4.5)
- Migrating from a LAN interconnect to Memory Channel (Section 4.6)
- Troubleshooting LAN interconnect problems (Section 4.7)

## 4.1 Configuring a NetRAIN Virtual Interface for a Cluster LAN Interconnect

If you do not configure the cluster interconnect from redundant array of independent network adapters (NetRAIN) virtual interfaces during cluster installation, you can do so afterwards. However, the requirements and rules for configuring a NetRAIN virtual interface for use in a cluster interconnect differ from those documented in the Tru64 UNIX *Network Administration: Connections* manual.

Unlike a typical NetRAIN virtual device, a NetRAIN device for the cluster interconnect is set up completely within the `ics_ll_tcp` kernel subsystem in `/etc/sysconfigtab` and not in `/etc/rc.config`. This allows the interconnect to be established very early in the boot path, when it is needed by cluster components to establish membership and transfer I/O.

**Caution**

Never change the attributes of a member's cluster interconnect NetRAIN device outside of its `/etc/sysconfigtab` file (that is, by using an `ifconfig` command or the SysMan Station, or by defining it in the `/etc/rc.config` file and restarting the network). Doing so will put the NetRAIN device outside of cluster

control and may cause the member system to be removed from the cluster. See Section 4.7.4 for more information.

---

To configure a NetRAIN interface for a cluster interconnect after cluster installation, perform the following steps on each member:

1. To eliminate the LAN interconnect as a single point of failure, one or more Ethernet switches are required for the cluster interconnect (two are required for a no-single-point-of-failure (NSPOF) LAN interconnect configuration), in addition to redundant Ethernet adapters on the member configured as a NetRAIN set. If you must install additional network hardware, halt and turn off the member system. Install the network cards on the member and cable each to different switches, as recommended in Section 2.1. Turn on the switches and reboot the member. If you do not need to install additional hardware, you can skip this step.
2. Use the `ifconfig -a` command to determine the names of the Ethernet adapters to be used in the NetRAIN set.
3. If you intend to configure an existing NetRAIN set for a cluster interconnect (for example, one previously configured for an external network), you must first undo its current configuration:

   a. Use the `rcmgr delete` command to delete the following variables from the member's `/etc/rc.config` file: `NRDEV_`*x*, `NRCONFIG_`*x*, `NETDEV_`*x*, `IFCONFIG_`*x*, variables associated with the device.

   b. Use the `rcmgr set` command to decrement the `NR_DEVICES` and `NUM_NETCONFIG` variables.

4. Edit the `/etc/sysconfigtab` file to add the new adapter. For example, change:

   ```
   ics_ll_tcp:

   ics_tcp_adapter0 = ee0
   ```

   to:

   ```
   ics_ll_tcp:

   ics_tcp_adapter0 = nr0
   ics_tcp_nr0[0] = ee0
   ics_tcp_nr0[1] = ee1
   ```

5. Reboot the member. The member is now using the NetRAIN virtual interface as its physical cluster interconnect.

6. Use the `ifconfig` command to show the NetRAIN device defined with the `CLUIF` flag. For example:

```
# ifconfig nr0
nr0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Attached Interfaces: ( ee0 ee1 ) Active Interface: ( ee0 )
    inet 10.1.0.2 netmask ffffff00 broadcast 10.1.0.255 ipmtu 1500
```

7. Repeat this procedure for each remaining member.

## 4.2 Tuning the LAN Interconnect

This section provides guidelines for tuning the LAN interconnect.

**Caution**

Do not tune a NetRAIN virtual interface being used for a cluster interconnect using those mechanisms used for other NetRAIN devices (including `ifconfig`, `niffconfig`, and `niffd` command options or `netrain` or `ics_ll_tcp` kernel subsystem attributes). Doing so is likely to disrupt cluster operation. The cluster software ensures that the NetRAIN device for the cluster interconnect is tuned for optimal cluster operation.

### 4.2.1 Improving Cluster Interconnect Performance by Setting Its ipmtu Value

Some applications may receive some performance benefit if you set the IP maximum transfer unit (`ipmtu`) for the cluster interconnect virtual interface (`ics0`) on each member to the same value used by its physical interface (`member`*n*`-tcp0`). The recommended value depends on the type of cluster interconnect in use.

- For 100 Mb/s Ethernet, the `ipmtu` value should be set to 1500.
- For Memory Channel, the `ipmtu` value should be set to 7000.

To view the current `ipmtu` settings for the virtual and physical cluster interconnect devices, use the following command:

```
# ifconfig -a
ee0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     inet 10.1.0.100 netmask ffffff00 broadcast 10.1.0.255 ipmtu 1500

ics0: flags=1100063<UP,BROADCAST,NOTRAILERS,RUNNING,NOCHECKSUM,CLUIF>
     inet 10.0.0.1 netmask ffffff00 broadcast 10.0.0.255 ipmtu 7000
```

Because this cluster member is using the `ee0` Ethernet device for its physical cluster interconnect device, change the `ipmtu` for its virtual cluster interconnect device (`ics0`) from 7000 to 1500.

To set the ipmtu value for the ics0 virtual device, perform the following procedure:

1. Add the following line to the /etc/inet.local file on each member, supplying an ipmtu value:

   ```
   ifconfig ics0 ipmtu value
   ```

2. Restart the network on each member using the rcinet restart command.

## 4.3 Obtaining Network Adapter Configuration Information

To display information from the datalink driver for a network adapter, such as its name, speed, and operating mode, use the SysMan Station or the hwmgr -get attr -cat network command. In the following example, tu2 is the client network adapter running at 10 Mb/s in half-duplex mode and ee0 and ee1 are a NetRAIN virtual interface configured as the LAN interconnect and running at 100 Mb/s in full-duplex mode:

```
# hwmgr -get attr -cat network | grep -E 'name|speed|duplex'
  name = tu2
  media_speed = 10
  full_duplex = 0
  user_name = (null) (settable)
  name = ee0
  media_speed = 100
  full_duplex = 1
  user_name = (null) (settable)
  name = ee1
  media_speed = 100
  full_duplex = 1
  user_name = (null) (settable)
```

## 4.4 Monitoring LAN Interconnect Activity

Use the netstat command to monitor the traffic across the LAN interconnect. For example:

```
# netstat -acdnots -I nr0
nr0 Ethernet counters at Mon Apr 30 14:15:15 2001

          65535 seconds since last zeroed
     3408205675 bytes received
     4050893586 bytes sent
        7013551 data blocks received
        6926304 data blocks sent
        7578066 multicast bytes received
         115546 multicast blocks received
        3182180 multicast bytes sent
          51014 multicast blocks sent
              0 blocks sent, initially deferred
              0 blocks sent, single collision
              0 blocks sent, multiple collisions
```

```
                0 send failures
                0 collision detect check failure
                0 receive failures
                0 unrecognized frame destination
                0 data overruns
                0 system buffer unavailable
                0 user buffer unavailable
nr0: access filter is disabled
```

Use the `ifconfig -a` and `niffconfig -v` commands to monitor the status of the active and inactive adapters in a NetRAIN virtual interface.

```
# ifconfig -a
ee0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Virtual Interface: nr0
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )

ee1: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Virtual Interface: nr0
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )

ics0: flags=1100063<UP,BROADCAST,NOTRAILERS,RUNNING,NOCHECKSUM,CLUIF>
     inet 10.0.0.200 netmask ffffff00 broadcast 10.0.0.255 ipmtu 15u00

lo0: flags=100c89<UP,LOOPBACK,NOARP,MULTICAST,SIMPLEX,NOCHECKSUM>
     inet 127.0.0.1 netmask ff000000 ipmtu 4096

nr0: flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,CLUIF>
     NetRAIN Attached Interfaces: ( ee1 ee0 ) Active Interface: ( ee1 )
     inet 10.1.0.2 netmask ffffff00 broadcast 10.1.0.255 ipmtu 1500

sl0: flags=10<POINTOPOINT>

tu0: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
     inet 16.140.112.176 netmask ffffff00 broadcast 16.140.112.255 ipmtu 1500

tun0: flags=80<NOARP>

# niffconfig -v
Interface:   ee1, description: NetRAIN internal, status:      UP, event:   ALERT, state: GREEN
         t1: 3, dt: 2, t2: 10, time to dead: 3, current_interval: 3, next time: 1
Interface:   nr0, description: NetRAIN internal, status:      UP, event:   ALERT, state: GREEN
         t1: 3, dt: 2, t2: 10, time to dead: 3, current_interval: 3, next time: 1
Interface:   ee0, description: NetRAIN internal, status:      UP, event:   ALERT, state: GREEN
         t1: 3, dt: 2, t2: 10, time to dead: 3, current_interval: 3, next time: 2
Interface:   tu0, description:                 , status:      UP, event:   ALERT, state: GREEN
         t1: 20, dt: 5, t2: 60, time to dead: 30, current_interval: 20, next time: 20
```

## 4.5 Migrating from Memory Channel to LAN

This section discusses how to migrate a cluster that uses Memory Channel as its cluster interconnect to a LAN interconnect.

Replacing a Memory Channel interconnect with a LAN interconnect requires some cluster downtime and interruption of service.

**Note**

If you are performing a rolling upgrade (as described in the *Cluster Installation* manual) from TruCluster Server Version 5.1 to TruCluster Server Version 5.1A and intend to replace the

Memory Channel with a LAN interconnect, plan on installing the LAN hardware on each member during the roll. Doing so allows you to avoid performing steps 1 through 4 in the following procedure.

---

To prepare to migrate an existing cluster using the Memory Channel interconnect to using a LAN interconnect, perform the following procedure for each cluster member:

1. Halt and turn off the cluster member.
2. Install the network adapters. Configure any required switches or hubs.
3. Turn on the cluster member.
4. Boot the member over Memory Channel into the existing cluster.

At this point, you can configure the newly installed Ethernet hardware as a private conventional subnet shared by all cluster members. You can verify that the hardware is configured properly and operates correctly before setting it up as a LAN interconnect. Do not use the `rcmgr` command or statically edit the `/etc/rc.config` file to permanently set up this network. Because this test network must not survive the reboot of the cluster over the LAN interconnect, use `ifconfig` commands on each member to set it up.

To configure the LAN interconnect, perform the following steps:

1. On each member, make backup copies of the member-specific `/etc/sysconfigtab` and `/etc/rc.config` files.
2. On each member, inspect the member-specific `/etc/rc.config` file, paying special attention to the `NETDEV_`*x* and `NRDEV_`*x* configuration variables. Because the network adapters used for the LAN interconnect must be configured very early in the boot process, they are defined in `/etc/sysconfigtab` (see next step) and must not be defined in `/etc/rc.config`. This applies to NetRAIN devices also. Decide whether you are configuring new devices or reconfiguring old devices for the LAN interconnect. If the latter, you must make appropriate edits to the `NRDEV_`*x*, `NRCONFIG_`*x*, `NETDEV_`*x*, `IFCONFIG_`*x*, `NR_DEVICES` and `NUM_NETCONFIG` variables so that the same network device names do not appear both in the `/etc/rc.config` file and the `ics_ll_tcp` stanza of the `/etc/sysconfigtab` file.
3. On each member, set the `clubase` kernel attribute `cluster_interconnect` to `tcp` and the following `ics_ll_tcp` kernel attributes as appropriate for the member's network configuration. For example:

```
clubase:
cluster_interconnect = tcp
#
```

```
ics_ll_tcp:
ics_tcp_adapter0 = nr0
ics_tcp_nr0[0] = ee0
ics_tcp_nr0[1] = ee1
ics_tcp_inetaddr0 = 10.1.0.1
ics_tcp_netmask0 = 255.255.255.0
```

For a cluster that was rolled to TruCluster Server Version 5.1A from TruCluster Server Version 5.1, also edit the `cluster_node_inter_name` attribute of the `clubase` kernel subsystem. For example:

```
clubase:
cluster_node_inter_name = pepicelli-ics0
```

4. Edit the clusterwide `/etc/hosts` file so that it contains the IP name and IP address of the cluster interconnect low-level TCP interfaces. For example:

```
127.0.0.1           localhost
16.140.112.238      pepicelli.zk3.dec.com       pepicelli
16.120.112.209      deli.zk3.dec.com            deli
10.0.0.1            pepicelli-ics0
10.1.0.1            member1-icstcp0
10.0.0.2            pepperoni-ics0
10.1.0.2            member2-icstcp0
16.140.112.176      pepperoni.zk3.dec.com       pepperoni
```

5. For a cluster that was rolled to TruCluster Server Version 5.1A from TruCluster Server Version 5.1, edit the clusterwide `/etc/hosts.equiv` file and the clusterwide `/.rhosts` file, changing the `mc0` entries to `ics0` entries. For example, change:

```
deli.zk3.dec.com
pepicelli-mc0
pepperoni-mc0
```

to:

```
deli.zk3.dec.com
pepicelli-ics0
member1-icstcp0
pepperoni-ics0
member2-icstcp0
```

6. For a cluster that was rolled to TruCluster Server Version 5.1A from TruCluster Server Version 5.1, use the `rcmgr set` command to change the `CLUSTER_NET` variable in the `/etc/rc.config` file on each member. For example:

```
# rcmgr get CLUSTER_NET
pepicelli-mc0
# rcmgr set CLUSTER_NET pepicelli-ics0
```

7. Halt all cluster members.
8. Boot all cluster members, one at a time.

## 4.6 Migrating from LAN to Memory Channel

This section discusses how to migrate a cluster that uses a LAN interconnect as its cluster interconnect to Memory Channel.

To configure the Memory Channel, perform the following steps:

1. On each member, make a backup copy of the member-specific `/etc/sysconfigtab` file.
2. On each member, set the `clubase` kernel attribute `cluster_interconnect` to `mct`.
3. Halt all cluster members.
4. If Memory Channel hardware is installed in the cluster, reboot all cluster members one at a time.

   If Memory Channel hardware is not yet installed in the cluster:

   a. Power off all members.

   b. Install and configure Memory Channel adapters, cables, and hubs as described in the *Cluster Hardware Configuration* manual.

   c. When the Memory Channel hardware has been properly set up, reboot all cluster members one at a time.

## 4.7 Troubleshooting

This section discusses the following problems that can occur due to a misconfigured LAN interconnect and how you can resolve them:

- A booting member joins the cluster but appears to hang later during the boot (Section 4.7.1).
- A booting member hangs while trying to join the cluster (Section 4.7.2).
- A booting member panics with an "ics_broadcast_setup" message (Section 4.7.3).
- A booting member displays an "ifconfig ioctl (SIOCIFADD): Function not implemented: nr0" message (Section 4.7.4).
- A booting member displays hundreds of broadcast errors and panics an existing member (Section 4.7.5).
- An `ifconfig` nr*x* switch command fails with a "No such device nr0" message (Section 4.7.6).

- An application running in the cluster cannot bind to a well-known port (Section 4.7.7).

### 4.7.1 Booting Member Joins Cluster But Appears to Hang Before Reaching Multi-User Mode

If a new member appears to hang at boot time sometime after joining the cluster, the speed or operational mode of the booting member's LAN interconnect adapter is probably inconsistent with that of the LAN interconnect. This problem can result from the adapter failing to autonegotiate properly, from improper hardware settings, or from faulty Ethernet hardware. To determine whether this problem exists, pay close attention to console messages of the following form on the booting member:

```
ee0: Parallel Detection, 10 Mbps half duplex
ee0: Autonegotiated, 100 Mbps full duplex
```

For a cluster interconnect running at 100 Mb/s in full-duplex mode, the first message may indicate a problem. The second message indicates that autonegotiation has completed successfully.

The autonegotiation behavior of the Ethernet adapters and switches that are configured in the interconnect may cause unexpected hangs at boot time if you do not take the following considerations into account:

- Autonegotiation settings must be the same on both ends of any given cable. That is, if an Ethernet adapter is configured for autonegotiation, the switch port to which it is connected must also be configured for autonegotiation. Similarly, if the adapter is cross-cabled to another member's adapter, the other member's adapter must be set to autonegotiate. If you violate this rule (for example, by setting one end to 100 Mb/s full-duplex, and the other to autonegotiate), the member set to autonegotiate may set itself to half-duplex mode while booting and cluster transactions will experience delays.
- Supported 100 Mb/s Ethernet network adapters in AlphaServer systems can use two different drivers: `ee` and `tu`.

  Network adapters in the DE50*x* family (which have a console name of the form `ew`*x*`0`) are based on the DECchip 21140, 21142, and 21143 chipsets and use the `tu` driver. If the network adapter uses the `tu` driver, it may or may not support autonegotiation.

**Note**

DE500-XA adapters do not support autonegotiation. Proper autonegotiation succeeds more often with DE500-BA and DE504 adapters than with DE500-AA adapters.

To use autonegotiation, set the ew*x*0_mode console variable to auto and set the port on the switch connected to the network adapter for autonegotiation.

With network adapters using the tu driver, it may be easier to force the adapter to use 100 Mb/s full-duplex mode explicitly. To force the adapter to use 100 Mb/s full-duplex mode, set the ew*x*0_mode variable to FastFD. In this case, you must use a switch that allows autonegotiation to be disabled and set the port on the switch connected the network adapter for 100 Mb/s full-duplex. See tu(7) and the switch's manual for more information.

Network adapters in the DE60*x* family (which have a console name of the form ei *x*0) use the ee driver. If the network adapter uses the ee driver, it by default uses IEEE 802.3u autonegotiation to determine which speed setting to use. Make sure that the port on the switch to which the network adapter is connected is set for autonegotiation. See ee(7) and the switch's manual for more information.

### 4.7.2 Booting Member Hangs While Trying to Join Cluster

If a new member hangs at boot time while trying to join the cluster, the new member might be disconnected from the cluster interconnect. The following may have caused the disconnect:

- A cable is unplugged.
- You specified an existing Ethernet adapter as the physical cluster interconnect interface to clu_add_member, but that adapter is not connected to other members (and perhaps is used for a purpose other than as a LAN interconnect, such as a client network).
- You specified an address for the cluster interconnect physical device that is not on the same subnet as those of other cluster members. For example, you may have specified an address on the cluster interconnect virtual subnet (ics0) for the member's cluster interconnect physical device.
- You specified a different interconnect type for this member (for example, the cluster_interconnect attribute in its clubase kernel subsystem is mct), whereas the rest of the cluster specifies tcp).

One of the following messages is typically displayed on the console:

```
CNX MGR: cannot form: quorum disk is in use.  Unable to establish contact
         with members using disk.
```

or

```
CNX MGR: Node pepperoni id 2 incarn 0xa3a71 attempting to form or join cluster deli
```

Perform the following steps to resolve this problem:

1. Halt the booting member.
2. Make sure the adapter is properly connected to the LAN interconnect.
3. Mount the new member's boot partition on another member. For example:

   ```
   # mount root2_domain#root /mnt
   ```

4. Examine the /mnt/etc/sysconfigtab file. The attributes listed in Table C–1 must be set correctly to reflect the member's LAN interconnect interface.
5. Edit /mnt/etc/sysconfigtab as appropriate.
6. Unmount the member's boot partition:

   ```
   # umount /mnt
   ```

7. Reboot the member.

### 4.7.3 Booting Member Panics with "ics_broadcast_setup" Message

If you boot a new member into the cluster and it panics with an "ics_broadcast_setup: sobind failed error=49" message, you may have specified a device that does not exist as the member's physical cluster interconnect interface to clu_add_member.

Perform the following steps to resolve this problem:

1. Halt the booting member.
2. Mount the new member's boot partition on another member. For example:

   ```
   # mount root2_domain#root /mnt
   ```

3. Examine the /mnt/etc/sysconfigtab file. The attributes listed in Table C–1 must be set to correctly reflect the member's LAN interconnect interface.
4. Edit /mnt/etc/sysconfigtab as appropriate.
5. Unmount the member's boot partition:

   ```
   # umount /mnt
   ```

6. Reboot the member.

### 4.7.4 Booting Member Displays "ifconfig ioctl (SIOCIFADD): Function not implemented: nr0" Message

If you boot a new member into the cluster and it displays the "ifconfig ioctl (SIOCIFADD): Function not implemented: nr0" message shortly after the installation tasks commence, a NetRAIN virtual interface used for the cluster interconnect has probably been misconfigured. Perhaps you have edited the `/etc/rc.config` file to apply traditional NetRAIN admin to the LAN interconnect. In this case, the NetRAIN configuration in the `/etc/rc.config` file is ignored and the NetRAIN interface defined in `/etc/sysconfigtab` is used as the cluster interconnect.

**Note**

If the address you specify in `/etc/rc.config` for the cluster interconnect NetRAIN device is on the same subnet as that used by the cluster interconnect virtual device (`ics0`), the boot will display hundreds of instances of the following message after the `ifconfig` message:

```
WARNING: ics_socket_event: reconfig: error 54 on channel xx,
                assume node 2 is down
```

Eventually, the cluster will remove either the booting member or an existing member (if only one member is up) with one of the following panics:

```
CNX QDISK: Yielding to foreign owner with quorum.

CNX MGR: this node removed from cluster.
```

As discussed in Section 4.1, you must never configure a NetRAIN set that is used for a cluster interconnect in the `/etc/rc.config` file. (The NetRAIN virtual interface for the cluster interconnect is configured in the `/etc/sysconfigtab` file.)

Perform the following steps to resolve this problem:

1. Use the `rcmgr delete` command to edit the newly booted member's `/cluster/members/{memb}/etc/rc.config` file to remove the `NRDEV_`*x*, `NRCONFIG_`*x*, `NETDEV_`*x*, and `IFCONFIG_`*x* variables associated with the device.
2. Use the `rcmgr set` command to decrement the `NR_DEVICES` and `NUM_NETCONFIG` variables that doubly define the cluster interconnect NetRAIN device.
3. Reboot the member.

### 4.7.5 Many Broadcast Errors on Booting or Booting New Member Panics Existing Member

The Spanning Tree Protocol (STP) must be disabled on all Ethernet switch ports connected to the adapters on cluster members, whether they are single adapters or included in the NetRAIN virtual interfaces. If this is not the case, cluster members may be flooded by broadcast messages that, in effect, create denial-of-service symptoms in the cluster. You may see hundreds of instances of the following message when booting the first and subsequent members:

```
arp: local IP address 10.1.0.100 in use by hardware address 00-00-00-00-00-00
```

These messages will be followed by:

```
CNX MGR: cnx_pinger: broadcast problem: err 35
```

Booting additional members into this cluster may result in a hang or panic of existing members, especially if a quorum disk is configured. During the boot, you may see the following message:

```
CNX MGR: cannot form: quorum disk is in use. Unable to establish
contact with members using disk.
```

However, after 30 seconds or so, the member may succeed in discovering the quorum disk and form its own cluster, while the existing members hang or panic.

### 4.7.6 Cannot Manually Fail Over Devices in a NetRAIN Virtual Interface

NetRAIN monitors the health of inactive interfaces by checking whether they are receiving packets and, if necessary, by sending probe packets from the active interface. If an inactive interface becomes disconnected, NetRAIN may mark it as `DEAD`. If you pull the cables on the active adapter, NetRAIN attempts to activate the `DEAD` standby adapter. Unless there is a real problem with this adapter, the failover works properly.

However, a manual NetRAIN switch operation (for example, `ifconfig nr0 switch`) behaves in a different way. In this case, NetRAIN does not attempt to fail over to a `DEAD` adapter when there are no healthy standby adapters. The `ifconfig nr0 switch` command returns a message such as the following:

```
ifconfig ioctl (SIOCIFSWITCH) No such device nr0
```

You may see this behavior in a dual-switch configuration if one switch is power cycled and you immediately try to manually fail over an active adapter from the other switch. After the switch that has been powered on has initialized itself (in a few minutes or so), manual NetRAIN failover should behave properly. If the failover does not work correctly, examine the cabling

of the switches and adapters and use the `ifconfig` and `niffconfig` commands to determine the state of the interfaces.

### 4.7.7 Applications Unable to Map to Port

By default, the communications subsystem in a cluster using a LAN interconnect uses port 900 as a rendezvous port for cluster broadcast traffic and reserves ports 901 through 910 and 912 through 917 for nonbroadcast channels. If an application uses a hardcoded reference to one of these ports, it will fail to bind to the port.

To remedy this situation, change the ports used by the LAN interconnect. Edit the `ics_tcp_rendezvous_port` and `ics_tcp_ports` attributes in the `ics_ll_tcp` subsystem, as described in `sys_attrs_ics_ll_tcp`(5), and reboot the entire cluster. The rendezvous port must be identical on all cluster members; the nonbroadcast ports may differ across members, although administration is simplied by defining the same ports on each member.

# A

# Configuring Switches for a Highly Available LAN Interconnect

The recommended highly available LAN interconnect configuration includes two network adapters per member configured as a two-member redundant array of independent network adapters (NetRAIN) virtual interface and connected to two independent switches. Proper operation of NetRAIN in this configuration requires an interswitch link to carry its maintenance and failover traffic. In this no-single-point-of-failure (NSPOF) LAN interconnect configuration, no single failure of the interconnect hardware will disable the whole cluster. However, the failure of this interswitch link can, under certain circumstances, result in a network partition that can cause the removal of up to half of the members from the cluster (see Section 2.2.3).

We recommend that you configure an additional interswitch link between the switches to avoid this behavior. However, the introduction of the additional link requires that the switches be additionally configured to avoid packet-forwarding problems caused by the routing loop created by the second link.

Typical switches provide at least one of the following three mechanisms to support parallel interswitch links. In order of decreasing desirability for cluster configurations, the mechanisms are:

| | |
|---|---|
| Link aggregation | Treats multiple physical links as a single link and distributes packet traffic among them. (Section A.1) |
| Link resiliency | Treats multiple physical links as an active link and one or more standby links and fails over between them. (Section A.2) |
| Spanning Tree Protocol | Employs a distributed routing protocol to permit switches to cooperate to remove routing loops. This is an IEEE standard mechanism (IEEE 802-1d). (Section A.3) |

The following sections discuss each of these in detail and describe the switch requirements and configuration options appropriate to each mechanism.

## A.1 Link Aggregation

If it is supported, link aggregation (also known as port trunking) is the best available solution to implement parallel interswitch links for a highly available LAN interconnect. Using link aggregation, you group the ports on each switch that are cross-cabled to the ports on the other switch. Each set of ports makes up a single virtual link. Traffic between the two switches is sent across the physical links that make up the virtual link.

This configuration provides several benefits:

- If any link or port in the virtual link fails, that physical link is disabled, but the other physical links that make up the virtual link continue to operate. The result is that there is no loss of connectivity between the two switches.
- Failover is normally immediate.
- Because each physical link can carry traffic between the two switches, the total available bandwidth between the switches may be greater than a single interswitch link can provide.

**Note**

Many switches, by default, use an algorithm based on the destination IP address or MAC address of a specific packet of data to decide which physical port will carry it. That is, traffic between two systems over an interswitch link always uses the same physical link. Depending on which adapters are active, this might not result in increased bandwidth. Some switches allow the choice of a round-robin algorithm that distributes traffic evenly, regardless of destination. If the switches used for the LAN interconnect support such an algorithm, using it may result in more efficient use of the interswitch links. The lack of support for such an algorithm does not impact the fault tolerance of the aggregated link; it only reduces the potential performance benefit.

## A.2 Link Resiliency

Some switches support link resiliency. If link aggregation is not supported, link resiliency is the next best option. Resilient links are specifically designed to support link failover. Typically, two links are involved: a main link and a standby link. Only the main link carries traffic between the two switches. When a failure is detected with the main link, the switches immediately start using the standby link. If the main link comes back on

line, the switches may either start using the main link again, or they may continue using the standby link.

Like link aggregation, link resiliency supports a quick failover in the event of link failure. However, unlink link aggregation, only one link is in use at a time, so there is no increase in available bandwidth.

## A.3 Spanning Tree Protocol (STP)

If neither of the previous two options are supported, you can use parallel links between the switches if both switches support the Spanning Tree Protocol standard (IEEE 802.1d). This industry-wide standard is designed to detect and remove packet loops in a network. When STP is enabled between the switches, only one interswitch link is used. If that link fails, the switches reconfigure themselves and use the other interswitch link, similar to resilient links.

When using STP in a LAN interconnect, the switch must adhere to the following requirements:

- The switch must allow STP to be disabled on a port-by-port basis. Some manufacturers who allow STP to be enabled or disabled only for the entire switch provide a mechanism (such as fast forwarding) to bypass the protocol on selected ports.
- STP route-learning time must be configurable to be shorter than the cluster NetRAIN link failover time (10 seconds).

When configuring a switch capable of STP in a LAN interconnect, comply with the following rules:

- Configure STP only on the ports that are used for the interswitch links. When some network cards are involved in a NetRAIN failover, they can trigger spanning tree reconfiguration if STP is enabled on their ports. The switches will drop packets during the spanning tree reconfiguration, which can result in a loss of connectivity for the node involved in the NetRAIN failover, even after the switches have finished the reconfiguration process. Consequently, spanning tree routing must be turned off on the ports of the switch that are connected to cluster members, and enabled only on those ports that are cross-cabled between the switches.

  Spanning tree routing has no use on ports connected to end nodes, and can cause problems. However, not all switches support selectively enabling and disabling spanning tree routing per port. In those cases, use link aggregation or link resiliency to implement parallel links (these are preferable to STP anyway), or do not use parallel interswitch links at all.
- Adjust the STP settings on the switches to minimize the amount of time they spend during the reconfiguration process, because the switches

will drop packets while they are in the reconfiguration process. Most switches allow three basic settings to be changed: hello time, forward delay, and maximum age. Set all three settings to their minimum values, which are normally 1 second for hello time, 4 seconds for forward delay, and 6 seconds for maximum age. These adjustments can help the switch recover more quickly in the event of the failure of an interswitch link.

# B

# Installation Examples

This chapter provides samples of the logs written by:

- `clu_create` (Section B.1)
- `clu_add_member` (Section B.2)

## B.1 clu_create Log

Each time you run `clu_create`, it writes log messages to `/cluster/admin/clu_create.log`. Example B–1 shows a sample `clu_create` log file.

**Example B–1: Sample clu_create Log File**

```
Do you want to continue creating the cluster? [yes]:yes

Each cluster has a unique cluster name, which is a hostname
used to identify the entire cluster.

Enter a fully-qualified cluster name []:deli.zk3.dec.com
Checking cluster name: deli.zk3.dec.com

You entered 'deli.zk3.dec.com' as your cluster name.
Is this correct? [yes]:yes

The cluster alias IP address is the IP address associated with the
default cluster alias.  (192.168.168.1 is an example of an IP address.)

Enter the cluster alias IP address []:16.140.112.209
Checking cluster alias IP address: 16.140.112.209

You entered '16.140.112.209' as the IP address for the default cluster alias.
Is this correct? [yes]:yes

The cluster root partition is the disk partition (for example, dsk4b)
that will hold the clusterwide root (/) file system.

    Note: The default 'a' partition on most disks is not large
    enough to hold the clusterwide root AdvFS domain.

Enter the device name of the cluster root partition []:dsk7b
Checking the cluster root partition: dsk7b

You entered 'dsk7b' as the device name of the cluster root partition.
Is this correct? [yes]:yes

The cluster usr partition is the disk partition (for example, dsk4g)
that will contain the clusterwide usr (/usr) file system.
```

## Example B–1: Sample clu_create Log File (cont.)

```
    Note: The default 'g' partition on most disks is usually
    large enough to hold the clusterwide usr AdvFS domain.

Enter the device name of the cluster usr partition [dsk7g]:dsk7g
Checking the cluster usr partition: dsk7g

You entered 'dsk7g' as the device name of the cluster usr partition.
Is this correct? [yes]:yes

To use this default value, press Return at the prompt.

The cluster var device is the disk partition (for example, dsk4h)
that will hold the clusterwide var (/var) file system.

    Note: The default 'h' partition on most disks is usually
    large enough to hold the clusterwide var AdvFS domain.

Enter the device name of the cluster var partition [dsk7h]:dsk7h
Checking the cluster var partition: dsk7h

You entered 'dsk7h' as the device name of the cluster var partition.
Is this correct? [yes]:yes

Do you want to define a quorum disk device at this time? [yes]:yes
The quorum disk device is the name of the disk (for example, 'dsk5')
that will be used as this cluster quorum disk.

Enter the device name of the quorum disk []:dsk6
Checking the quorum disk device: dsk6

You entered 'dsk6' as the device name of the quorum disk device.
Is this correct? [yes]:yes

By default the quorum disk is assigned '1' vote(s).
To use this default value, press Return at the prompt.

The number of votes for the quorum disk is an integer usually 0 or 1.
If you select 0 votes then the quorum disk will not contribute votes to the
cluster. If you select 1 vote then the quorum disk must be accessible to
boot and run a single member cluster.

Enter the number of votes for the quorum disk [1]:1
Checking number of votes for the quorum disk: 1

You entered '1' as the number votes for the quorum disk.
Is this correct? [yes]:yes

The default member ID for the first cluster member is '1'.
To use this default value, press Return at the prompt.

A member ID is used to identify each member in a cluster.
Each member must have a unique member ID, which is an integer in
the range 1-63, inclusive.

Enter a cluster member ID [1]:1
Checking cluster member ID: 1

You entered '1' as the member ID.
Is this correct? [yes]:yes

By default the 1st member of a cluster is assigned '1' vote(s).
```

## Example B–1: Sample clu_create Log File (cont.)

```
Checking number of votes for this member: 1

Each member has its own boot disk, which has an associated
device name; for example, 'dsk5'.

Enter the device name of the member boot disk []:dsk10
Checking the member boot disk: dsk10

The specified disk contains the required 'a', 'b', and 'h'
partitions.  The current partition sizes are acceptable for a member's
boot disk.  You can either keep the current disk partition layout or have
the installation program relabel the disk.  If the program relabels the disk,
the new label will contain the following partitions and sizes (in blocks):

    Current                 New
    -------                 ---
    a: 524288               a: 524288
    b: 7849648              b: 7853744
    h: 2048                  h: 2048

Do you want to use the current disk partitions? [yes]:yes

You entered 'dsk10' as the device name of this member's boot disk.
Is this correct? [yes]:yes

Device 'ics0' is the default virtual cluster interconnect device
Checking virtual cluster interconnect device: ics0

The virtual cluster interconnect IP name 'pepicelli-ics0' was formed by
appending '-ics0' to the system's hostname.
To use this default value, press Return at the prompt.

Each virtual cluster interconnect interface has a unique IP name (a
hostname) associated with it.

Enter the IP name for the virtual cluster interconnect [pepicelli-ics0]:pepicelli-ics0
Checking virtual cluster interconnect IP name: pepicelli-ics0

You entered 'pepicelli-ics0' as the IP name for the virtual cluster interconnect.
Is this name correct? [yes]:yes

The virtual cluster interconnect IP address '10.0.0.1' was created by
replacing the last byte of the default virtual cluster interconnect network
address '10.0.0.0' with the previously chosen member ID '1'.
To use this default value, press Return at the prompt.

The virtual cluster interconnect IP address is the IP address
associated with the virtual cluster interconnect IP name.  (192.168.168.1
is an example of an IP address.)

Enter the IP address for the virtual cluster interconnect [10.0.0.1]:10.0.0.1
Checking virtual cluster interconnect IP address: 10.0.0.1

You entered '10.0.0.1' as the IP address for the virtual cluster interconnect.
Is this address correct? [yes]:yes

What type of cluster interconnect will you be using?

    Selection   Type of Interconnect
---------------------------------------------------------------------
         1      Memory Channel
```

## Example B–1: Sample clu_create Log File (cont.)

```
         2      Local Area Network
         3      None of the above
         4      Help
         5      Display all options again
------------------------------------------------------------------------
Enter your choice [1]:2
You selected option '2' for the cluster interconnect
Is that correct? (y/n) [y]:y

The physical cluster interconnect interface device is the name of the
physical device(s) which will be used for low level cluster node
communications. Examples of the physical cluster interconnect interface
device name are: tu0, ee0, and nr0.

Enter the physical cluster interconnect device name(s) []:ee0
Would you like to place this Ethernet device into a NetRAIN set? [yes]:no
Checking physical cluster interconnect interface device name(s): ee0

You entered 'ee0' as your physical cluster interconnect interface
device name(s). Is this correct? [yes]:yes

The physical cluster interconnect IP name 'member1-icstcp0' was formed by
appending '-icstcp0' to the word 'member' and the member ID.
Checking physical cluster interconnect IP name: member1-icstcp0

The physical cluster interconnect IP address '10.1.0.1' was created by
replacing the last byte of the default cluster interconnect network address
'10.1.0.0' with the previously chosen member ID '1'.
To use this default value, press Return at the prompt.

The cluster physical interconnect IP address is the IP address
associated with the physical cluster interconnect IP name. (192.168.168.1
is an example of an IP address.)

Enter the IP address for the physical cluster interconnect [10.1.0.1]:10.1.0.100
Checking physical cluster interconnect IP address: 10.1.0.100

You entered '10.1.0.100' as the IP address for the physical cluster interconnect.
Is this address correct? [yes]:yes


You entered the following information:

    Cluster name:                                        deli.zk3.dec.com
    Cluster alias IP Address:                            16.140.112.209
    Clusterwide root partition:                          dsk7b
    Clusterwide usr  partition:                          dsk7g
    Clusterwide var  partition:                          dsk7h
    Clusterwide i18n partition:                          Directory-In-/usr
    Quorum disk device:                                  dsk6
    Number of votes assigned to the quorum disk:         1
    First member's member ID:                            1
    Number of votes assigned to this member:             1
    First member's boot disk:                            dsk10
    First member's virtual cluster interconnect device name: ics0
    First member's virtual cluster interconnect IP name:     pepicelli-ics0
    First member's virtual cluster interconnect IP address:  10.0.0.1
    First member's physical cluster interconnect devices     ee0
    First member's NetRAIN device name                       Not-Applicable
    First member's physical cluster interconnect IP address  10.1.0.100
```

## Example B–1: Sample clu_create Log File (cont.)

```
If you want to change any of the above information, answer 'n' to the
following prompt. You will then be given an opportunity to change your
selections.
Do you want to continue to create the cluster? [yes]:yes

Creating required disk labels.
  Creating disk label on member disk : dsk10
  Initializing cnx partition on member disk : dsk10h
  Creating disk label on quorum disk : dsk6
  Initializing cnx partition on quorum disk : dsk6h

Creating AdvFS domains:
  Creating AdvFS domain 'root1_domain#root' on partition '/dev/disk/dsk10a'.
  Creating AdvFS domain 'cluster_root#root' on partition '/dev/disk/dsk7b'.
  Creating AdvFS domain 'cluster_usr#usr' on partition '/dev/disk/dsk7g'.
  Creating AdvFS domain 'cluster_var#var' on partition '/dev/disk/dsk7h'.

Populating clusterwide root, usr, and var file systems:
  Copying root file system to 'cluster_root#root'.
....
  Copying usr file system to 'cluster_usr#usr'.
.....................................
  Copying var file system to 'cluster_var#var'.
..

Creating Content Dependent Symbolic Links (CDSLs) for file systems:
  Creating CDSLs in root file system.
  Creating CDSLs in usr  file system.
  Creating CDSLs in var  file system.
  Creating links between clusterwide file systems

Populating member's root file system.

Modifying configuration files required for cluster operation:
  Creating /etc/fstab file.
  Configuring cluster alias.
  Updating /etc/hosts - adding IP address '16.140.112.209' and hostname 'deli.zk3.dec.com'
  Updating member-specific /etc/inittab file with 'cms' entry.
  Updating /etc/hosts - adding IP address '10.0.0.1' and hostname 'pepicelli-ics0'
  Updating /etc/hosts - adding IP address '10.1.0.100' and hostname 'member1-icstcp0'
  Updating /etc/rc.config file.
  Updating /etc/sysconfigtab file.
  Retrieving cluster_root major and minor device numbers.
  Creating cluster device file CDSLs.
  Updating /.rhosts - adding hostname 'deli.zk3.dec.com'.
  Updating /etc/hosts.equiv - adding hostname 'deli.zk3.dec.com'
  Updating /.rhosts - adding hostname 'pepicelli-ics0'.
  Updating /etc/hosts.equiv - adding hostname 'pepicelli-ics0'
  Updating /.rhosts - adding hostname 'member1-icstcp0'.
  Updating /etc/hosts.equiv - adding hostname 'member1-icstcp0'
  Updating /etc/ifaccess.conf - adding deny entry for 'ee0'
  Updating /etc/ifaccess.conf - adding deny entry for 'ee1'
  Updating /etc/ifaccess.conf - adding deny entry for 'sl0'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu0'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu1'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu2'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu3'
  Updating /etc/ifaccess.conf - adding deny entry for 'tun0'
  Updating /etc/ifaccess.conf - adding deny entry for 'ee0'
  Updating /etc/ifaccess.conf - adding deny entry for 'ee1'
  Updating /etc/ifaccess.conf - adding deny entry for 'sl0'
```

**Example B–1: Sample clu_create Log File (cont.)**

```
  Updating /etc/ifaccess.conf - adding deny entry for 'tu0'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu1'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu2'
  Updating /etc/ifaccess.conf - adding deny entry for 'tu3'
  Updating /etc/ifaccess.conf - adding deny entry for 'tun0'
  Updating /etc/cfgmgr.auth - adding hostname 'ernest.zk3.dec.com'
  Finished updating member1-specific area.

Building a kernel for this member.
  Saving kernel build configuration.
  The kernel will now be configured using the doconfig program.


*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

Saving /sys/conf/ERNEST as /sys/conf/PEPICELLI.bck


*** PERFORMING KERNEL BUILD ***
 Working....Wed Apr 18 16:39:52 EDT 2001

The new kernel is /sys/PEPICELLI/vmunix
  Finished running the doconfig program.

  The kernel build was successful and the new kernel
   has been copied to this member's boot disk.
  Restoring kernel build configuration.

Updating console variables
  Setting console variable 'bootdef_dev' to dsk10
  Setting console variable 'boot_dev' to dsk10
  Setting console variable 'boot_reset' to ON
  Saving console variables to non-volatile storage

clu_create: Cluster created successfully.

To run this system as a single member cluster it must be rebooted.
If you answer yes to the following question clu_create will reboot the
system for you now. If you answer no, you must manually reboot the
system after clu_create exits.
Would you like clu_create to reboot this system now? [yes]:y
Shutdown at 16:53 (in 0 minutes) [pid 4211]
```

## B.2 clu_add_member Log

Each time you run `clu_add_member`, it writes log messages to `/cluster/admin/clu_add_member.log`. Example B–2 shows a sample `clu_add_member` log file.

## Example B–2: Sample clu_add_member Log File

```
Do you want to continue adding this member? [yes]:yes

Each cluster member has a hostname, which is assigned to the HOSTNAME
variable in /etc/rc.config.

Enter the new member's fully qualified hostname []:polishham.zk3.dec.com
Checking member's hostname: polishham.zk3.dec.com

You entered 'polishham.zk3.dec.com' as this member's hostname.
Is this name correct? [yes]:yes

The next available member ID for a cluster member is '2'.
To use this default value, press Return at the prompt.

A member ID is used to identify each member in a cluster.
Each member must have a unique member ID, which is an integer in
the range 1-63, inclusive.

Enter a cluster member ID [2]:2
Checking cluster member ID: 2

You entered '2' as the member ID.
Is this correct? [yes]:yes

By default, when the current cluster's expected votes are greater then 1,
each added member is assigned 1 vote(s). Otherwise, each added member is
assigned 0 (zero) votes.
To use this default value, press Return at the prompt.

The number of votes for a member is an integer usually 0 or 1
Enter the number of votes for this member [1]:1
Checking number of votes for this member: 1

You entered '1' as the number votes for this member.
Is this correct? [yes]:yes

Each member has its own boot disk, which has an associated
device name; for example, 'dsk5'.

Enter the device name of the member boot disk []:dsk11
Checking the member boot disk: dsk11

You entered 'dsk11' as the device name of this member's boot disk.
Is this correct? [yes]:yes

Device 'ics0' is the default virtual cluster interconnect device
Checking virtual cluster interconnect device: ics0

The virtual cluster interconnect IP name 'polishham-ics0' was formed by
appending '-ics0' to the system's hostname.
To use this default value, press Return at the prompt.

Each virtual cluster interconnect interface has a unique IP name (a
hostname) associated with it.

Enter the IP name for the virtual cluster interconnect [polishham-ics0]:polishham-ics0
Checking virtual cluster interconnect IP name: polishham-ics0

You entered 'polishham-ics0' as the IP name for the virtual cluster interconnect.
Is this name correct? [yes]:yes
```

## Example B–2: Sample clu_add_member Log File (cont.)

```
The virtual cluster interconnect IP address '10.0.0.2' was created by
replacing the last byte of the virtual cluster interconnect network address
'10.0.0.0' with the previously chosen member ID '2'.
To use this default value, press Return at the prompt.

The virtual cluster interconnect IP address is the IP address
associated with the virtual cluster interconnect IP name.  (192.168.168.1
is an example of an IP address.)

Enter the IP address for the virtual cluster interconnect [10.0.0.2]:10.0.0.2
Checking virtual cluster interconnect IP address: 10.0.0.2

You entered '10.0.0.2' as the IP address for the virtual cluster interconnect.
Is this address correct? [yes]:yes

The physical cluster interconnect interface device is the name of the
physical device(s) which will be used for low level cluster node
communications. Examples of the physical cluster interconnect interface
device name are: tu0, ee0, and nr0.

Enter the physical cluster interconnect device name(s) []:ee0, ee1
Would you like to enter another Ethernet device? [yes]:no
Checking physical cluster interconnect interface device name(s): ee0,ee1

You entered 'ee0,ee1' as your physical cluster interconnect interface
device name(s). Is this correct? [yes]:yes

Enter a NetRAIN interface device name []:nr0
Checking NetRAIN interface device: nr0

You entered 'nr0' as your NetRAIN interface device name.
Is this correct? [yes]:yes

The physical cluster interconnect IP name 'member2-icstcp0' was formed by
appending '-icstcp0' to the word 'member' and the member ID.
Checking physical cluster interconnect IP name: member2-icstcp0

The physical cluster interconnect IP address '10.1.0.2' was created by
replacing the last byte of the physical cluster interconnect network address
'10.1.0.0' with the previously chosen member ID '2'.
To use this default value, press Return at the prompt.

The cluster physical interconnect IP address is the IP address
associated with the physical cluster interconnect IP name. (192.168.168.1
is an example of an IP address.)

Enter the IP address for the physical cluster interconnect [10.1.0.2]:10.1.0.200
Checking physical cluster interconnect IP address: 10.1.0.200

You entered '10.1.0.200' as the IP address for the physical cluster interconnect.
Is this address correct? [yes]:yes

Each cluster member must have its own registered TruCluster Server
license. The data required to register a new member is typically located on
the License PAK certificate or it may have been previously placed on your
system as a partial or complete license data file. If you are prepared to
enter this license data at this time, clu_add_member can configure the new
member to use this license data. If you do not have the license data at this
time you can enter this data on the new member when it is up and running.
Do you wish to register the TruCluster Server license for this new member at
this time? [yes]:no
```

## Example B–2: Sample clu_add_member Log File (cont.)

```
You entered the following information:

    Member's hostname:                                polishham.zk3.dec.com
    Member's ID:                                      2
    Number of votes assigned to this member:          1
    Member's boot disk:                               dsk11
    Member's virtual cluster interconnect devices:    ics0
    Member's virtual cluster interconnect IP name:    polishham-ics0
    Member's virtual cluster interconnect IP address: 10.0.0.2
    Member's physical cluster interconnect devices:   ee0,ee1
    Member's NetRAIN device name:                     nr0
    Member's physical cluster interconnect IP address: 10.1.0.200
    Member's cluster license:                         Not Entered

If you want to change any of the above information answers 'n' to the
following prompt. You will then be given an opportunity to change your
selections.
Do you want to continue to add this member? [yes]:yes

Creating required disk labels.
  Creating disk label on member disk : dsk11
  Initializing cnx partition on member disk : dsk11h

Creating AdvFS domains:
  Creating AdvFS domain 'root2_domain#root' on partition '/dev/disk/dsk11a'.

Creating cluster member-specific files:
  Creating new member's root member-specific files
  Creating new member's usr  member-specific files
  Creating new member's var  member-specific files
  Creating new member's boot member-specific files

Modifying configuration files required for new member operation:
  Updating /etc/hosts - adding IP address '10.0.0.2' and hostname 'polishham-ics0'
  Updating /etc/hosts - adding IP address '10.1.0.200' and hostname 'member2-icstcp0'
  Updating /etc/rc.config
  Updating /etc/sysconfigtab
  Updating member-specific /etc/inittab file with 'cms' entry.
  Updating /etc/securettys - adding ptys entry
  Updating /.rhosts - adding hostname 'polishham-ics0'
  Updating /etc/hosts.equiv - adding hostname 'polishham-ics0'
  Updating /.rhosts - adding hostname 'member2-icstcp0'
  Updating /etc/hosts.equiv - adding hostname 'member2-icstcp0'
  Updating /etc/cfgmgr.auth - adding hostname 'polishham.zk3.dec.com'
  Configuring cluster alias.
  Configuring Network Time Protocol for new member
  Adding interface 'pepicelli-ics0' as an NTP peer to member 'polishham.zk3.dec.com'
  Adding interface 'polishham-ics0' as an NTP peer to member 'pepicelli.zk3.dec.com'

Configuring automatic subset configuration and kernel build.

clu_add_member: Initial member 2 configuration completed successfully.
From the newly added member's console, perform the following steps to
complete the newly added member's configuration:

    1. Set the console variable 'boot_osflags' to 'A'.
    2. Identify the console name of the newly added member's boots device.

       >>>show device
```

## Example B–2: Sample clu_add_member Log File (cont.)

```
      The newly added member's boot device has the following properties:

      Manufacturer: DEC
      Model: RZ1CF-CF (C) DEC
      Target: 12
      Lun: 0
      Serial Number: SCSI-WWID:04100024:"DEC     RZ1CF-CF (C) DEC    50066084"

      Note: The SCSI bus number may differ when viewed from different members.

   3. Boot the newly added member using genvmunix:

       >>>boot -file genvmunix <new-member-boot-device>

      During this initial boot the newly added member will:

      o  Configure each installed subset.

      o  Attempt to build and install a new kernel. If the system cannot
         build a kernel, it starts a shell where you can attempt to build
         a kernel manually. If the build succeeds, copy the new kernel to
         /vmunix. When you are finished exit the shell using ^D or 'exit'.

      o  The newly added member will attempt to set boot related console
         variables and continue to boot to multi-user mode.

      o  After the newly added member boots you should setup your system
         default network interface using the appropriate system management
         command.
```

# C

# Cluster Interconnect /etc/sysconfigtab Attributes Set by clu_create and clu_add_member

Table C–1 lists the /etc/sysconfigtab attributes written by the cluster installation procedure to define the cluster interconnect.

**Table C–1: Cluster Interconnect /etc/sysconfigtab Attributes Set by clu_create and clu_add_member**

| Subsystem | Attribute | Comment |
|---|---|---|
| clubase | cluster_interconnect | Interconnect type to be used for cluster internode communications. If a LAN interconnect is used, the clu_create and clu_add_member commands set this attribute to tcp. If Memory Channel is used, the scripts set it to mct. |
| ics_ll_tcp | ics_tcp_inetaddr0 | The clu_create and clu_add_member commands set this attribute to the IP address of the physical cluster interconnect device (for example, 10.1.0.1). |
| | ics_tcp_netmask0 | Subnet mask used for the cluster interconnect. The clu_create and clu_add_member commands write the value 255.255.255.0 to this attribute. |
| | ics_tcp_adapter0 | The clu_create and clu_add_member commands set this attribute to the name of the physical cluster interconnect device (for example, tu0 or nr0). |
| | ics_tcp_nr0 | If the ics_tcp_adapter0 attribute indicates a NetRAIN set, this attribute is an array indicating the device names of the network adapters that make up the set. |

# Index

## A



## B



## C

## D



## E



## F

## H



## I



## L

## S



## T



## V



## W