Guidelines for OpenVMS Cluster Configurations

Order Number: AA–Q28LC–TK

January 1999

OpenVMS Cluster availability, scalability, and system management benefits are highly dependent on configurations, applications, and operating environments. This guide provides suggestions and guidelines to help you maximize these benefits.

Revision/Update Information: This manual supersedes *Guidelines for OpenVMS Cluster Configurations* Version 7.1.

Software Version: OpenVMS Alpha Version 7.2
OpenVMS VAX Version 7.2

Compaq Computer Corporation
Houston, Texas

January 1999

ZK6318

The OpenVMS documentation set is available on CD–ROM.

Contents

4 Choosing OpenVMS Cluster Interconnects

5 Choosing OpenVMS Cluster Storage Subsystems

6 Configuring Multiple Paths to SCSI and Fibre Channel Storage

9 Configuring CI OpenVMS Clusters for Availability and Performance

10 Configuring OpenVMS Clusters for Scalability

11 OpenVMS Cluster System Management Strategies

A SCSI as an OpenVMS Cluster Interconnect

B MEMORY CHANNEL Technical Summary

C CI-to-PCI Adapter (CIPCA) Support

D Multiple-Site OpenVMS Clusters

Index

Examples

Figures

Tables

Preface

This document can help you design an OpenVMS Cluster configuration to suit your business, application, and computing needs.

It provides information to help you choose systems, interconnects, storage devices, and software. It can also help you combine these components to achieve high availability, scalability, performance, and ease of system management.

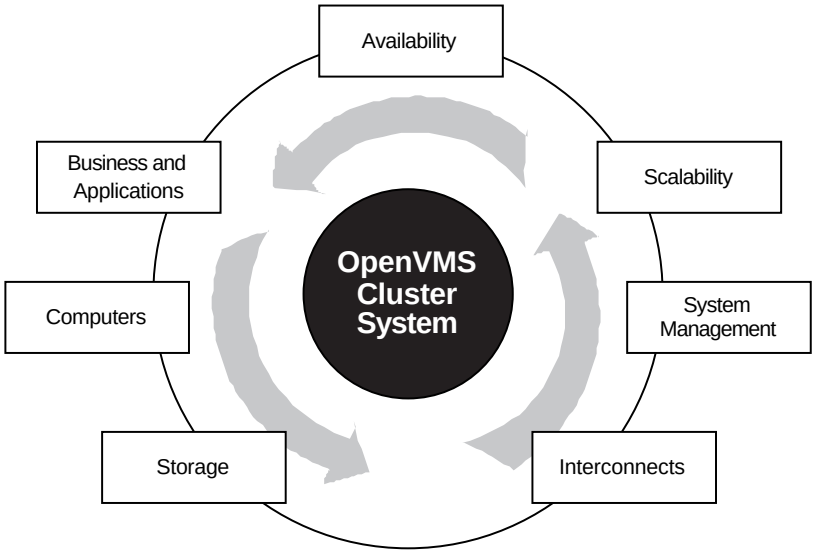
Intended Audience

This document is for people who purchase or recommend the purchase of OpenVMS Cluster products and for people who configure OpenVMS Cluster systems. It assumes a basic understanding of computers and OpenVMS Cluster concepts.

About This Guide

OpenVMS Cluster systems are designed to act as a single virtual system, even though they are made up of many components and features, as shown in Figure 1.

Figure 1 OpenVMS Cluster System Components and Features



Understanding the components and features of an OpenVMS Cluster configuration can help you to get the most out of your cluster. Table 1 shows how this guide is organized to explain these cluster concepts.

Table 1 Document Organization

Read...	Chapter Title	So that you can...
Chapter 1	Overview of OpenVMS Cluster System Configurations	Understand OpenVMS Cluster hardware, software, and general concepts
Chapter 2	Determining Business and Application Requirements	Learn to analyze your business and application needs and how they apply to your cluster
Chapter 3	Choosing OpenVMS Cluster Systems	Understand your computer requirements and make appropriate choices
Chapter 4	Choosing OpenVMS Cluster Interconnects	Learn about cluster interconnects and make appropriate choices
Chapter 5	Choosing OpenVMS Cluster Storage Subsystems	Learn to analyze your storage requirements and make appropriate choices
Chapter 6	Configuring Multiple Paths to SCSI and Fibre Channel Storage	Learn how to configure multiple paths to storage using Parallel SCSI or Fibre Channel interconnects, thereby increasing availability
Chapter 7	Configuring Fibre Channel as an OpenVMS Cluster Storage Interconnect	Learn how to configure an OpenVMS Cluster with Fibre Channel as a storage interconnect
Chapter 8	Configuring OpenVMS Clusters for Availability	Understand how to increase the availability of a cluster system
Chapter 9	Configuring CI OpenVMS Clusters for Availability and Performance	Learn how to use multiple components and advanced techniques to obtain high levels of availability and performance
Chapter 10	Configuring OpenVMS Clusters for Scalability	Learn how to expand an OpenVMS Cluster system in all of its dimensions, while understanding the tradeoffs
Chapter 11	OpenVMS Cluster System Management Strategies	Understand and deal effectively with some of the issues involved in managing an OpenVMS Cluster system
Appendix A	SCSI As an OpenVMS Cluster Interconnect	Configure multiple hosts and storage on a single SCSI bus so that multiple VAX and Alpha hosts can share access to SCSI devices directly
Appendix B	MEMORY CHANNEL Technical Summary	Learn why, when, and how to use the MEMORY CHANNEL interconnect
Appendix C	CI-to-PCI Adapter (CIPCA) Technical Summary	Learn why, when, and how to use the CIPCA adapter
Appendix D	Multiple-Site OpenVMS Clusters	Understand the benefits, the configuration options and requirements, and the management of multiple-site OpenVMS Cluster systems

Related Documents

For additional information on the topics covered in this manual, refer to the following documents:

- *DIGITAL Systems and Options Catalog* and its periodic supplements
- *Volume Shadowing for OpenVMS*
- *DECamds User's Guide*
- *OpenVMS Performance Management*
- *DECnet for OpenVMS Networking Manual*
- *DECnet/OSI documentation set*

- *Guide to OpenVMS File Applications*
- *OpenVMS Guide to System Security*
- *OpenVMS System Manager's Manual*

For additional information on the Open Systems Software Group (OSSG) products and services, access the following OpenVMS World Wide Web address:

`http://www.openvms.digital.com`

For ordering and configuring information for Alpha and VAX workstations and servers, access the *DIGITAL Systems and Options Catalog* at the following World Wide Web address:

`http://www.digital.com:80/info/soc/`

The *DIGITAL Systems and Options Catalog* web site also provides links to the AlphaServer web site and the *Network Products Guide*.

Reader's Comments

Compaq welcomes your comments on this manual.

Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT and send us your comments by:

Internet	**openvmsdoc@zko.mts.dec.com**
Fax	603 884-0120, Attention: OSSG Documentation, ZKO3-4/U08
Mail	Compaq Computer Corporation OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following World Wide Web address to order additional documentation:

`http://www.openvms.digital.com:81/`

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).

Conventions

In this manual, any reference to OpenVMS is synonymous with DIGITAL OpenVMS.

VMScluster systems are now referred to as OpenVMS Cluster systems. Unless otherwise specified, references to OpenVMS Clusters or clusters in this document are synonymous with VMSclusters.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

PF1 *x* or GOLD *x*	A sequence such as PF1 *x* or GOLD *x* indicates that you must first press and release the key labeled PF1 or GOLD and then press and release another key or a pointing device button. GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.) In the HTML version of this document, this convention appears as brackets, rather than a box.
. . .	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.
[ ]	In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)
{ }	In command format descriptions, braces indicate a required choice of options; you must choose one of the options listed.
bold text	This text style represents the introduction of a new term or the name of an argument, an attribute, or a reason.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Monospace text`	Monospace type indicates code examples and interactive screen displays.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1

Overview of OpenVMS Cluster System Configuration

This chapter contains information about OpenVMS Cluster hardware and software components, as well as general configuration rules.

1.1 Mixed Alpha and VAX Clusters

An OpenVMS Cluster is a group of OpenVMS Alpha and OpenVMS VAX systems, storage subsystems, interconnects, and software that work together as one virtual system.

In an OpenVMS Cluster system, each Alpha and VAX node:

- Shares processing resources, queues, and data storage
- Can boot or fail independently
- Runs the OpenVMS operating system

In addition, an OpenVMS Cluster system is managed as a single entity.

Table 1–1 shows the benefits that an OpenVMS Cluster system offers.

Table 1–1 OpenVMS Cluster System Benefits

Benefit	Description
Resource sharing	Multiple systems can access the same storage devices, so that users can share files clusterwide. You can also distribute applications, batch, and print-job processing across multiple systems. Jobs that access shared resources can execute on any system.
Availability	Data and applications remain available during scheduled or unscheduled downtime of individual systems. A variety of configurations provide many levels of availability up to and including disaster-tolerant operation.
Flexibility	OpenVMS Cluster computing environments offer compatible hardware and software across a wide price and performance range.
Scalability	You can add processing and storage resources without disturbing the rest of the system. The full range of systems, from high-end symmetric multiprocessing (SMP) systems to smaller workstations, can be interconnected and easily reconfigured to meet growing needs. You control the level of performance and availability as you expand.
Ease of management	OpenVMS Cluster management is efficient and secure. Because you manage an OpenVMS Cluster as a single system, many tasks need to be performed only once. OpenVMS Clusters automatically balance user, batch, and print work loads.

(continued on next page)

Table 1–1 (Cont.) OpenVMS Cluster System Benefits

Benefit	Description
Open systems	Adherence to IEEE® POSIX®, OSF/Motif®, OSF DCE®, ANSI SQL, and TCP/IP standards provides OpenVMS Cluster systems with application portability and interoperability.

1.2 Hardware Components

An OpenVMS Cluster system comprises many hardware components, such as systems, interconnects, adapters, storage subsystems, and peripheral devices. Table 1–2 describes these components and provides examples.

Table 1–2 Hardware Components in an OpenVMS Cluster System

Components	Description	Examples
System	A cabinet that contains one or more processors, memory, and input/output (I/O) adapters that act as a single processing body. **Reference:** See Chapter 3 for more information about Digital's OpenVMS systems.	OpenVMS Cluster systems can contain any supported Alpha, VAX, or MicroVAX system, including SMP systems.
Interconnect	The hardware connection between OpenVMS Cluster nodes over which the nodes communicate. **Reference:** See Chapter 4 for more information about OpenVMS Cluster interconnects.	An OpenVMS Cluster system can have one or more of the following interconnects: • CI • Digital Storage Systems Interconnect (DSSI) • Fiber Distributed Data Interface (FDDI) • MEMORY CHANNEL™ • Small Computer Systems Interface (SCSI) • Fibre Channel • Ethernet
Storage subsystems	Devices on which data is stored and the optional controllers that manage the devices. **Reference:** See Chapter 5 for more information about OpenVMS storage subsystems.	Storage subsystems can include: • SCSI disks and tapes • SDI, STI disks and tapes • Storage array cabinets • SDI/STI and SCSI storage controllers • InfoServer systems

(continued on next page)

Table 1–2 (Cont.) Hardware Components in an OpenVMS Cluster System

Components	Description	Examples
Adapter	Devices that connect nodes in an OpenVMS Cluster to interconnects and storage. **Reference:** See Chapter 4 for more information about adapters.	The following adapters are used on Peripheral Component Interconnect (PCI) systems: • CIPCA (CI) • KFPSA (DSSI) • KZPSA (SCSI) • DEFPA (FDDI) • DE435 (Ethernet) • KGPSA (Fibre Channel)
Peripheral devices	Devices that provide input to and produce output from a system. **Reference:** See the *Digital Systems and Options Catalog* for more information about peripheral devices.	Peripheral devices include: • Terminals, terminal servers, and modems • Printers, plotters

1.3 Software Components

OpenVMS Cluster system software can be divided into the following types:

- OpenVMS operating system components
- Networking components
- Storage enhancement software
- System management software
- Business applications

1.3.1 OpenVMS Operating System Components

The operating system manages proper operation of hardware and software components and resources.

Table 1–3 describes the operating system components necessary for OpenVMS Cluster operations. All of these components are enabled by an OpenVMS operating system license or an OpenVMS Cluster license.

Table 1–3 Operating System Components

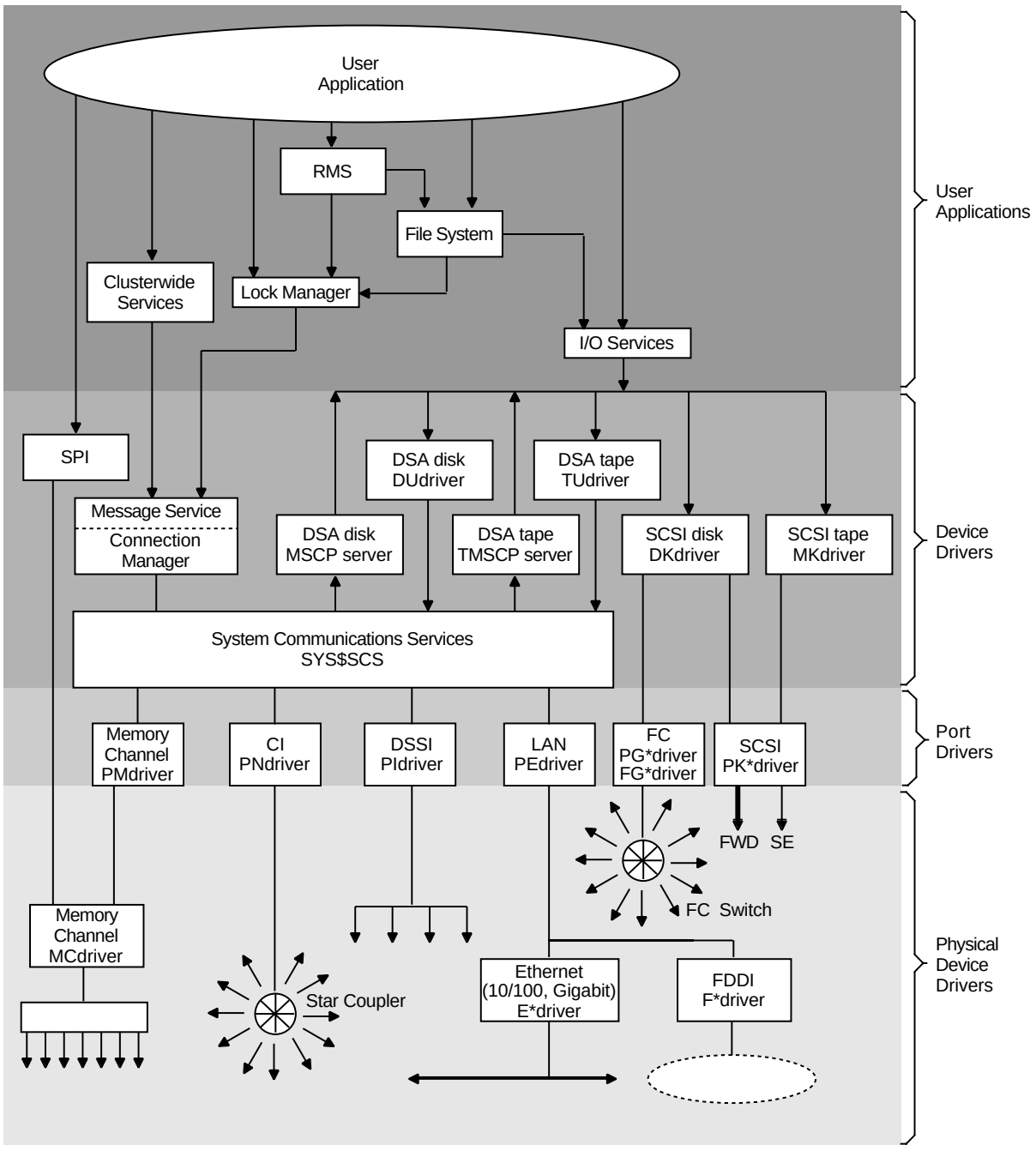
Component	Function
Record Management Services (RMS) and OpenVMS file system	Provide shared read and write access to files on disks and tapes in an OpenVMS Cluster environment.
Clusterwide process services	Enables clusterwide operation of OpenVMS commands, such as SHOW SYSTEM and SHOW USERS, as well as the ability to create and delete processes clusterwide.

(continued on next page)

Table 1–3 (Cont.) Operating System Components

Component	Function
Distributed Lock Manager	Synchronizes access by many users to shared resources.
Distributed Job Controller	Enables clusterwide sharing of batch and print queues, which optimizes the use of these resources.
Connection Manager	Controls the membership and quorum of the OpenVMS Cluster members.
SCS (System Communications Services)	Implements OpenVMS Cluster communications between nodes using Digital's System Communications Architecture (SCA).
MSCP server	Makes locally connected disks to which it has direct access available to other systems in the OpenVMS Cluster.
TMSCP server	Makes locally connected tapes to which it has direct access available to other systems in the OpenVMS Cluster.

Figure 1–1 shows how the hardware and operating system components fit together in a typical OpenVMS Cluster system.



1.3.2 Networking Components

Table 1–4 describes the optional networking software that enables OpenVMS Cluster system nodes to communicate and share resources with other OpenVMS Cluster nodes.

Table 1–4 OpenVMS Cluster Networking Components

Optional Software	Function
DECnet–Plus	A network transport such as DECnet–Plus or DIGITAL TCP/IP Services for OpenVMS software is necessary for internode communication.
DECnet–Plus System Services (DSS) products	Software to let you communicate and share resources among systems over extended distances. Products include VAX Distributed File Service (DFS), VAX Distributed Name Service (DNS), and the VAX Distributed Queuing Service (DQS).
LAT software	Used with terminal server hardware to support Ethernet-based character cell terminals. During a system failure, LAT software automatically makes a connection to one of the remaining systems.
PATHWORKS	Client and server networking software that links PCs and Macintosh® systems into OpenVMS Cluster systems.
DIGITAL TCP/IP Services for OpenVMS software	Provides Network File System (NFS) server capabilities for OpenVMS and supports Internet® networking protocols.
InfoServer software	Software used with InfoServer systems to serve physical device media to the local area network. Systems running the client software can connect to virtual devices and use them as if they were locally attached devices.

1.3.3 Storage Enhancement Software

Optional storage enhancement software improves the performance or availability of storage subsystems.

Examples include:

- Volume Shadowing for OpenVMS (redundant arrays of independent disks [RAID] level 1)
- DECram for OpenVMS (random access memory [RAM] disk)
- StorageWorks RAID Software for OpenVMS (supports RAID level 0 arrays (disk striping) and RAID level 5 arrays (disk striping with parity)
- DIGITAL Hierarchical Storage Manager (HSM)

For the latest information about StorageWorks products, refer to the StorageWorks web site at the following address:

`http://www.storage.digital.com`

1.3.4 System Management Software

System management software helps you manage your OpenVMS Cluster system.

Examples include:

- DIGITAL Availability Manager for Distributed Systems (DECamds)

  Note: DECamds is provided with the OpenVMS operating system.
- POLYCENTER Capacity Planner
- POLYCENTER Network Manager

1.3.5 Business Applications

Business applications are optional software packages that help you perform your business function.

Examples include:

- Database systems, such as Oracle® Rdb, Oracle CDD/Repository, and Sybase®
- Transaction processing systems, such as DIGITAL ACMSxp for OpenVMS Alpha
- Software development systems, such as COHESION
- Transaction routing systems, such as Reliable Transaction Router

1.4 Configuring an OpenVMS Cluster System

To take advantage of OpenVMS Cluster features and benefits, proper configuration is essential. An ideal OpenVMS Cluster configuration meets the following criteria:

- Provides the best combination of hardware and software components to meet your business requirements.
- Strategically distributes your budget dollars to return maximum value in the areas that are high priority for your business.
- Meets your current needs and retains your investment as your business needs grow and change.

Configuring your OpenVMS Cluster system requires careful planning because you need to consider many factors. You will probably modify plans as new factors arise. As your design evolves, you can weigh advantages against tradeoffs and make decisions that best meet your needs.

1.4.1 General Configuration Rules

The following general rules apply to OpenVMS Cluster systems:

- A node is an OpenVMS system. An OpenVMS Cluster system cannot contain more than 96 Alpha and VAX (combined total) nodes.
- An Alpha and a VAX system may not boot from the same system disk. System disks are architecture specific and can be shared only by systems of the same architecture.

  Cross-architecture *satellite* booting is supported. Alpha satellites (clients) can boot from a VAX boot server, and VAX satellites (clients) can boot from an Alpha boot server.
- Every OpenVMS node must be able to communicate directly with every other OpenVMS Cluster node.

  Configurations that use a shared (multihost) SCSI bus or a shared (multihost) Fibre Channel interconnect must also be configured with any of the other supported OpenVMS Cluster interconnects, because node-to-node communication does not occur across the SCSI bus.

  Reference: See Section 4.8 for more information about the SCSI interconnect.

Configurations that use a MEMORY CHANNEL interconnect must also be configured with any of the other supported OpenVMS Cluster interconnects for access to storage. Storage cannot be configured on MEMORY CHANNEL.

Reference: See Section 4.7 for more information about MEMORY CHANNEL.

- An OpenVMS Cluster node, storage controller, or storage device can participate in only one OpenVMS Cluster system at a time.
- DECnet–Plus software is not required in an OpenVMS Cluster configuration. However, DECnet–Plus is necessary if internode process-to-process communication using DECnet mailboxes is needed. Starting with OpenVMS Version 7.0, the Monitor utility uses a TCP/IP or DECnet transport, as appropriate, for intracluster communication.

In addition to these general rules, more detailed guidelines apply to different configurations. The rest of this manual discusses those guidelines in the context of specific configurations.

2

Determining Business and Application Requirements

This chapter contains information about how to determine your OpenVMS Cluster business and application requirements.

2.1 Determining Business Requirements

The kinds of business requirements that you have affect the way that you configure your OpenVMS Cluster. Typical business requirements for an OpenVMS Cluster system include:

- Budget
- Availability
- Scalability and future growth
- Physical location requirements
- Security

Some of these requirements may conflict with each other, such as scalability and physical location. For example, you may want to grow your OpenVMS Cluster, but you are limited by physical space or by the location of your systems. In situations like this, determine what your primary requirements are and where you are willing to make tradeoffs.

2.1.1 Budget

As with most business decisions, many of your choices will be determined by cost. Prioritizing your requirements can help you apply your budget resources to areas with the greatest business needs.

When determining your budget, plan for the initial system cost as well as the cost of ownership, which includes:

- Service and update
- Power consumption
- Cooling
- System management

2.1.2 Availability

Determine how available your computing system must be. Most organizations fall into one of the three broad (and sometimes overlapping) categories shown in Table 2–1.

Table 2–1 Availability Requirements

Availability Requirements	Description
Conventional	For business functions that can wait with little or no effect while a system or application is unavailable.
24 x 365	For business functions that require uninterrupted computing services, either during essential time periods or during most hours of the day throughout the year. Minimal down time is acceptable.
Disaster tolerant	For business functions with extremely stringent availability requirements. These businesses need to be immune to disasters like earthquakes, floods, and power failures.

Reference: For more information about availability, see Chapter 8 in this guide.

2.1.3 Scalability and Future Growth

Scalability is the ability to expand an OpenVMS Cluster in any system, storage, and interconnect dimension and at the same time fully use the initial configuration equipment. Scalability at the node level means being able to upgrade and add to your node's hardware and software. Scalability at the OpenVMS Cluster level means being able to increase the capacity of your entire OpenVMS Cluster system by adding processing power, interconnects, and storage across many nodes.

Among the low-end PCs and workstations, midrange departmental systems, and high-end data center systems offered by Digital, each level has different processing, storage, and interconnect characteristics. Investing in the appropriate level means choosing systems that meet and perhaps exceed your current business requirements with some extra capacity to spare. The extra capacity is for future growth, because designing too close to your current needs can limit or reduce the scalability of your OpenVMS Cluster.

If you design with future growth in mind, you can make the most of your initial investment, reuse original equipment, and avoid unnecessary upgrades later.

Reference: See Chapter 10 for more help with analyzing your scalability requirements.

2.1.4 Physical Location Requirements

Physical restrictions can play a key role in how you configure your OpenVMS Cluster. Designing a cluster for a small computer room or office area is quite different from designing one that will be spread throughout a building or across several miles. Power and air-conditioning requirements can also affect configuration design.

You may want to allow room for physical growth and increased power and cooling requirements when designing your cluster.

Reference: See Section 8.6 and Section 10.7.7 for information about multiple and extended local area network (LAN) configurations.

2.1.5 Security

A secure environment is one that limits physical and electronic access to systems by unauthorized users. Most businesses can achieve a secure environment with little or no performance overhead. However, if security is your highest priority, you may need to make tradeoffs in convenience, cost, and performance.

Reference: See the *OpenVMS Guide to System Security* for more information.

2.2 Determining Application Requirements

Applications require processing power, memory, storage, and I/O resources. Determining your application requirements allows you to design an OpenVMS Cluster system that will meet your application needs. To determine your application requirements, follow the steps described in Table 2–2.

Table 2–2 Determining Your Application Requirements

Step	Description
1	Make a list of the applications you currently run or expect to run.
2	For each application, write down your processor, memory, and I/O requirements (the application documentation provides this information.)
	Processor power must be proportional to the number of calculations your applications perform, with enough additional processor power to oversee data transfer between nodes and between nodes and storage.
	Memory capacity must be sufficient for your applications and for additional OpenVMS Cluster functions. Extra memory frequently improves system performance, so an initial investment in extra memory is probably a good one.
	I/O performance requirements differ among applications. As you choose components such as nodes, interconnects, and adapters, monitor the inherent speed of each component so that you can choose faster components and eliminate potential bottlenecks.
3	Add up the CPU, memory, and I/O requirements for all of your applications. Add to this sum any special requirements, such as user requirements and peripheral devices.
4	When you have determined your total application requirements, be sure that your CPU, memory, and I/O resources exceed these requirements by 20%.

2.2.1 Adding Memory

Systems require approximately 5% more memory to run in an OpenVMS Cluster than to run standalone. This additional memory is used to support the shared cluster resource base, which is larger than in a standalone configuration.

With added memory, a node in an OpenVMS Cluster generally can support the same number of users or applications that it supported as a standalone system. As a cluster configuration grows, the amount of memory used for system work by each node may increase. Because the per-node increase depends on both the level of data sharing in the cluster and the distribution of resource management, that increase does not follow fixed rules. If the node is a resource manager for a heavily used resource, additional memory may increase performance for cluster users of that resource.

Reference: For more information about using additional memory to improve performance, refer to the *OpenVMS Performance Management* manual.

2.2.2 Balancing Processor, Memory, and I/O Resources

Application performance depends on adequate processor, memory, and I/O resources. Depending on your applications, one of these resources may be more important than the others. Consider your application requirements, and find a balance among these three resources that meets your requirements. Table 2–3 provides some guidelines on the resource requirements of different application types.

Table 2–3 Resource Requirements of Application Types

Application Type	Example	Requirement
General timesharing	Program development, document preparation, office automation	Processor and I/O intensive
Searching and updating a database and displaying reports	Transaction processing, funds transfer, online order entry or reservation systems	I/O and memory intensive
Simulation, modeling, or calculation	Computer-aided design and manufacturing, image processing, graphics applications	Processor and memory intensive

2.2.3 Tools and Utilities

The OpenVMS operating system supports a number of utilities and tools that help you determine your business and application requirements in OpenVMS Cluster configurations. Table 2–4 describes many of these products and indicates whether each is supplied with the OpenVMS operating system or is an optional product.

Table 2–4 System Management Tools

Tool	Supplied or Optional	Function
Accounting utility	Supplied	Tracks how resources are being used.
AUTOGEN command procedure	Supplied	Optimizes system parameter settings based on usage.
DECamds (Digital Availability Manager for Distributed Systems)	Supplied	Collects and analyzes data from multiple nodes simultaneously, directing all output to a centralized DECwindows display. The analysis detects resource availability problems and suggests corrective actions.
Monitor utility	Supplied	Provides basic performance data.
POLYCENTER Capacity Planner	Optional	Provides a capacity-planning function to help analyze how changes in the configuration affect user performance.
Show Cluster utility	Supplied	Monitors activity and performance in a OpenVMS Cluster configuration.
DECevent	Supplied	Monitors system and device status and predicts failures.
OpenVMS Management Station	Supplied	Enables system managers to configure and manage user accounts, print queues, and storage across multiple OpenVMS Clusters and OpenVMS nodes. OpenVMS Management Station is a Microsoft® Windows® and Windows NT™ based management tool.

3

Choosing OpenVMS Cluster Systems

This chapter provides information to help you select systems for your OpenVMS Cluster to satisfy your business and application requirements.

3.1 Alpha and VAX Architectures

An OpenVMS Cluster can include systems running OpenVMS Alpha, OpenVMS VAX, or both. Compaq provides a full range of systems for both the Alpha and VAX architectures.

- OpenVMS Alpha operating system

  Based on a 64-bit RISC (reduced instruction set computing) architecture, OpenVMS Alpha provides industry-leading price/performance benefits with standard I/O subsystems for flexibility and expansion.

- OpenVMS VAX operating system

  Based on a 32-bit CISC (complex instruction set computing) architecture, OpenVMS VAX provides high CISC performance and a rich, powerful instruction set. OpenVMS VAX also supports a wide variety of standard I/O subsystems.

3.2 Types of Systems

Alpha and VAX systems span a range of computing environments, including:

- Personal computers (PCs)
- Workstations
- Departmental systems
- Enterprise systems
- Data center systems

3.3 Choosing Systems

Your choice of systems depends on your business, your application needs, and your budget. With a high-level understanding of systems and their characteristics, you can make better choices.

Table 3–1 is a comparison of recently shipped OpenVMS Cluster systems. While laptop and personal computers can be configured in an OpenVMS Cluster as client satellites, they are not discussed extensively in this manual. For more information about configuring PCs and laptops, see the *PATHWORKS Version 6.0 for DOS and Windows: Installation and Configuration Guide*

Table 3–1 System Types

System Type	Useful for	Examples
Workstations	Users who require their own systems with high processor performance. Examples include users running mechanical computer-aided design, scientific analysis, and data-reduction and display applications. Workstations offer the following features: • Lower cost than departmental and data center systems • Small footprint • Useful for modeling and imaging • 2D and 3D graphics capabilities	AlphaStation 200 AlphaStation 500 AlphaStation 600 VAXstation 4000 MicroVAX 3100
Departmental systems	Midrange office computing. Departmental systems offer the following capabilities: • High processor and I/O performance • Supports a moderate number of users, client PCs, and workstations	AlphaServer 400 AlphaServer 1000 AlphaServer 2000 AlphaServer 2100 AlphaServer 4100 VAX 4000
Data center systems	Large-capacity configurations and highly available technical and commercial applications. Data center systems have a high degree of expandability and flexibility and offer the following features: • Highest CPU and I/O performance • Ability to support thousands of terminal users, hundreds of PC clients, and up to 95 workstations	AlphaServer 8400 AlphaServer 8200 VAX 7800

3.4 Scalability Considerations

When you choose a system based on scalability, consider the following:

- Maximum processor capacity
- Maximum memory capacity
- Maximum storage capacity

The OpenVMS environment offers a wide range of alternative ways for growing and expanding processing capabilities of a data center, including the following:

- Many Alpha and VAX systems can be expanded to include additional memory, processors, or I/O subsystems.
- You can add systems to your OpenVMS Cluster at any time to support increased work load. The vast range of systems, from small workstation to high-end symmetric multiprocessing (SMP) systems, can interconnect and be reconfigured easily to meet growing needs.

- You can add storage to your OpenVMS Cluster system by increasing the quantity and speed of disks, CD–ROM devices, and tapes.

  Reference: For more information about storage devices, see Chapter 5.

Reference: For more information about scalability, see Chapter 10.

3.5 Availability Considerations

An OpenVMS Cluster system is a highly integrated environment in which multiple systems share access to resources. This resource sharing increases the availability of services and data. OpenVMS Cluster systems also offer failover mechanisms that are transparent and automatic, and require little intervention by the system manager or the user.

Reference: See Chapter 8 and Chapter 9 for more information about these failover mechanisms and about availability.

3.6 Performance Considerations

The following factors affect the performance of systems:

- Applications and their performance requirements
- The number of users that the system must support
- The type of storage subsystem that you require

With these requirements in mind, compare processor performance, I/O throughput, memory capacity, and disk capacity in the Alpha and VAX specifications that follow.

3.7 System Specifications

The *DIGITAL Systems and Options Catalog* provides ordering and configuring information for Intel, Alpha, and VAX workstations and servers. It also contains detailed information about storage devices, printers, and network application support.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4
Choosing OpenVMS Cluster Interconnects

An interconnect is a hardware connection between OpenVMS Cluster nodes over which the nodes can communicate. This chapter contains information about the following interconnects and how they are used in OpenVMS Clusters:

- Fibre Channel (storage only)
- MEMORY CHANNEL (node-to-node only)
- SCSI (Small Computer Systems Interface) (storage only)
- CI (computer interconnect)
- DSSI (Digital Storage Systems Interconnect)
- Ethernet
- FDDI (Fiber Distributed Data Interface)

The software that enables OpenVMS Cluster systems to communicate over an interconnect is the System Communications Services (SCS).

4.1 Characteristics

The six interconnects described in this chapter share some general characteristics. Table 4–1 describes these characteristics.

Table 4–1 Interconnect Characteristics

Characteristic	Description
Throughput	The quantity of data transferred across the interconnect.
	Some interconnects require more processor overhead than others. For example, Ethernet and FDDI interconnects require more processor overhead than do CI or DSSI.
	Larger packet sizes allow higher data-transfer rates (throughput) than do smaller packet sizes.
Cable length	Interconnects range in length from 3 m to 40 km.
Maximum number of nodes	The number of nodes that can connect to an interconnect varies among interconnect types. Be sure to consider this when configuring your OpenVMS Cluster system.
Supported systems and storage	Each OpenVMS Cluster node and storage subsystem requires an adapter to connect the internal system bus to the interconnect. First consider the storage and processor I/O performance, then the adapter performance, when choosing an interconnect type.

4.2 Comparison of Interconnect Types

Table 4–2 shows key statistics for a variety of interconnects.

Table 4–2 Comparison of Interconnect Types

Attribute	CI	DSSI	FDDI	SCSI	MEMORY CHANNEL	Ethernet	Fibre Channel
Maximum throughput (Mb/s)	140	32	100	160	800	10	1000
Hardware-assisted data link[1]	Yes	Yes	No	No	No	No	No
Connection to storage	Direct and MSCP served	Direct and MSCP served	MSCP served	Direct and MSCP served	MSCP served	MSCP served	Direct and MSCP served
Topology	Radial coaxial cable	Bus	Dual ring of trees	Bus or radial to a hub	Radial copper cable	Linear coaxial cable	Radial to a switch
Maximum nodes	32[2]	8[3]	96[4]	8–16[5]	4	96[4]	8[7]
Maximum length	45 m	6 m[6]	40 km	25 m	3 m	2800 m	400 m

[1]Hardware-assisted data link reduces the processor overhead required.

[2]Up to 16 OpenVMS Cluster computers; up to 31 HSC controllers.

[3]Up to 4 OpenVMS Cluster computers; up to 7 storage devices.

[4]OpenVMS Cluster computers.

[5]Up to 3 OpenVMS Cluster computers, up to 4 with the DWZZH-05 and fair arbitration; up to 15 storage devices.

[6]DSSI cabling lengths vary based on cabinet cables.

[7]Up to 4 OpenVMS Cluster computers; up to 4 storage ports (larger numbers are planned).

4.3 Multiple Interconnects

You can use multiple interconnects to achieve the following benefits:

- Failover

  If one interconnect or adapter fails, the node communications automatically move to another interconnect.

- MSCP server load balancing

  In a multiple MSCP server configuration, an OpenVMS Cluster performs load balancing to automatically choose the best path. This reduces the chances that a single adapter could cause an I/O bottleneck. Depending on your configuration, multiple paths from one node to another node may transfer more information than would a single path.

  Reference: See Section 10.7.3 for an example of dynamic MSCP load balancing.

4.4 Mixed Interconnects

A mixed interconnect is a combination of two or more different types of interconnects in an OpenVMS Cluster system. You can use mixed interconnects to combine the advantages of each type and to expand your OpenVMS Cluster system. For example, an Ethernet cluster that requires more storage can expand with the addition of CI, DSSI, or SCSI connections.

4.5 Interconnects Supported by Alpha and VAX Systems

Table 4–3 shows the OpenVMS Cluster interconnects supported by Alpha and VAX systems. You can also refer to the most recent OpenVMS Cluster SPD to see the latest information on supported interconnects.

Table 4–3 Cluster Interconnects Supported by Systems

Systems	CI	DSSI	SCSI	FDDI	Ethernet	MEMORY CHANNEL	Fibre Channel
AlphaServer 8400, 8200	X	X	X	X[1]	X	X	X
AlphaServer 4100, 2100, 2000	X	X	X	X[1]	X[1]	X	X[2]
AlphaServer 1000		X	X	X	X[1]	X	
AlphaServer 400		X	X	X	X[1]		
AlphaStation series			X	X	X[1]		
DEC 7000/10000	X	X		X[1]	X		
DEC 4000		X		X	X[1]		
DEC 3000			X	X[1]	X[1]		
DEC 2000				X	X[1]		
VAX 6000/7000/10000	X	X		X	X		
VAX 4000, MicroVAX 3100		X		X	X[1]		
VAXstation 4000				X	X[1]		

[1] Able to boot over the interconnect as a satellite node.
[2] Support on AlphaServer 4100; support on additional AlphaServer systems in the future.

As Table 4–3 shows, OpenVMS Clusters support a wide range of interconnects: CI, DSSI, SCSI, FDDI, Ethernet, and MEMORY CHANNEL. This power and flexibility means that almost anything will work well. The most important factor to consider is how much I/O you need, as explained in Chapter 2.

In most cases, the I/O requirements will be less than the capabilities of any one OpenVMS Cluster interconnect. Ensure that you have a reasonable surplus I/O capacity, then choose your interconnects based on other needed features.

4.6 Fibre Channel Interconnect

Note

The Fibre Channel functionality will be available shortly after the release of OpenVMS Version 7.2. This documentation is provided in advance to help you plan for the introduction of Fibre Channel in your computing environment.

Fibre Channel is a high-performance ANSI standard network and storage interconnect for PCI-based Alpha systems. It is a full-duplex serial interconnect and can simultaneously transmit and receive 100 megabytes per second. For the initial release, Fibre Channel will support simultaneous access of SCSI storage by multiple nodes connected to a Fibre Channel switch. A second type of interconnect is needed for node-to-node communications. Node-to-node communications over Fibre Channel is planned for a future release.

For multihost access to Fibre Channel storage, the following components are required:

- Fibre Channel host adapter (KGPSA)
- Multimode fiber-optic cable (BNGBX-*nn*), where *nn* represents distance in meters
- Fibre Channel switch (DSGGA)
- Storage devices that are supported in a multihost configuration (HSG80)

4.6.1 Advantages

The Fibre Channel interconnect offers the following advantages:

- High-speed transmission, 1.06 Gb/s
- Scalable configuration to support department to enterprise configurations. The first release of Fibre Channel supports up to four OpenVMS Cluster systems, up to four Fibre Channel switches, and up to four dual HSG storage controllers.
- Long-distance interconnects

  The first release of Fibre Channel supports multimode fiber at 400 meters per link. Longer distances are planned.
- High availability

  Multipath support is available. As many as four Fibre Channel interconnects can be connected to each OpenVMS Cluster system.

4.6.2 Throughput

The Fibre Channel interconnect transmits up to 1.06 Gb/s. It is a full-duplex serial interconnect that can simultaneously transmit and receive 100 MB/s.

4.6.3 Supported Adapter

The Fibre Channel adapter, the KGPSA, connects to the PCI bus.

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.7 MEMORY CHANNEL Interconnect

MEMORY CHANNEL is a high-performance cluster interconnect technology for PCI-based Alpha systems. With the benefits of very low latency, high bandwidth, and direct memory access, MEMORY CHANNEL complements and extends the unique ability of OpenVMS Clusters to work as a single, virtual system.

Three hardware components are required by a node to support a MEMORY CHANNEL connection:

- A PCI-to-MEMORY CHANNEL adapter
- A link cable (3 m or 10 feet long)

- A port in a MEMORY CHANNEL hub (except for a two-node configuration in which the cable connects just two PCI adapters)

A MEMORY CHANNEL hub is a PC size unit that provides a connection among systems. MEMORY CHANNEL can support up to four Alpha nodes per hub. You can configure systems with two MEMORY CHANNEL adapters in order to provide failover in case an adapter fails. Each adapter must be connected to a different hub.

A MEMORY CHANNEL hub is not required in clusters that comprise only two nodes. In a two-node configuration, one PCI adapter is configured, using module jumpers, as a virtual hub.

4.7.1 Advantages

MEMORY CHANNEL technology provides the following features:

- Offers excellent price/performance.

  With several times the CI bandwidth, MEMORY CHANNEL provides a 100 MB/s interconnect with minimal latency. MEMORY CHANNEL architecture is designed for the industry-standard PCI bus.

- Requires no change to existing applications.

  MEMORY CHANNEL works seamlessly with existing cluster software, so that no change is necessary for existing applications. The new MEMORY CHANNEL drivers, PMDRIVER and MCDRIVER, integrate with the System Communications Services layer of OpenVMS Clusters in the same way that existing port drivers do. Higher layers of cluster software are unaffected.

- Offloads CI, DSSI, and the LAN in SCSI clusters.

  You cannot connect storage directly to MEMORY CHANNEL, but you can use it to make maximum use of each interconnect's strength.

  While MEMORY CHANNEL is not a replacement for CI and DSSI, when used in combination with those interconnects, it offloads their node-to-node traffic. This enables them to be dedicated to storage traffic, optimizing communications in the entire cluster.

  When used in a cluster with SCSI and LAN interconnects, MEMORY CHANNEL offloads node-to-node traffic from the LAN, enabling it to handle more TCP/IP or DECnet traffic.

- Provides fail-separately behavior.

  When a system failure occurs, MEMORY CHANNEL nodes behave like any failed node in an OpenVMS Cluster. The rest of the cluster continues to perform until the failed node can rejoin the cluster.

4.7.2 Throughput

The MEMORY CHANNEL interconnect has a very high maximum throughput of 100 MB/s. If a single MEMORY CHANNEL is not sufficient, up to two interconnects (and two MEMORY CHANNEL hubs) can share throughput.

4.7.3 Supported Adapter

The MEMORY CHANNEL adapter connects to the PCI bus. The newest MEMORY CHANNEL adapter, CCMAA–BA, provides improved performance.

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.8 SCSI Interconnect

The SCSI interconnect is an industry standard interconnect that supports one or more computers, peripheral devices, and interconnecting components. SCSI is a single-path, daisy-chained, multidrop bus. It is a single 8-bit or 16-bit data path with byte parity for error detection. Both inexpensive single-ended and differential signaling for longer distances are available.

In an OpenVMS Cluster, multiple Alpha computers on a single SCSI interconnect can simultaneously access SCSI disks. This type of configuration is called multihost SCSI connectivity. A second type of interconnect is required for node-to-node communication. For multihost access to SCSI storage, the following components are required:

- SCSI host adapter that is supported in a multihost configuration (see Table 4–6)
- SCSI interconnect
- Terminators, one for each end of the SCSI interconnect
- Storage devices that are supported in a multihost configuration (RZ*nn*; refer to the OpenVMS Cluster SPD [29.78.*nn*])

For larger configurations, the following components are available:

- Storage controllers (HSZ*nn*)
- Bus isolators (DWZZA, DWZZB, or DWZZC) to convert single-ended to differential signaling and to effectively double the SCSI interconnect length

Reference: For a detailed description of how to connect SCSI configurations, see Appendix A.

4.8.1 Advantages

The SCSI interconnect offers the following advantages:

- Lowest cost, shared direct access to storage

  Because SCSI is an industry standard and is used extensively throughout the industry, it is available from many manufacturers at competitive prices.

- Scalable configuration to achieve high performance at a moderate price

  You can choose:

  - Width of SCSI interconnect

    Narrow (8 bits) or wide (16 bits).

  - Transmission mode

Single-ended signaling, the most common and least expensive, or differential signaling, which provides higher signal integrity and allows a longer SCSI interconnect.

- Signal speed (standard, fast, or ultra mode)
- Number of nodes sharing the SCSI bus (two or three)
- Number of shared SCSI buses to which a node can connect (maximum of six)
- Storage type and size (RZ*nn* or HSZ*nn*)
- Computer type and size (AlphaStation or AlphaServer)

4.8.2 Throughput

Table 4–4 show throughput for the SCSI interconnect.

Table 4–4 Maximum Data Transfer Rates in Megabytes per Second

Mode	Narrow (8-Bit)	Wide (16-Bit)
Standard	5	10
Fast	10	20
Ultra	20	40

4.8.3 SCSI Interconnect Distances

The maximum length of the SCSI interconnect is determined by the signaling method used in the configuration and, for single-ended signaling, by the data transfer rate.

There are two types of electrical signaling for SCSI interconnects: single ended and differential. Both types can operate in standard mode, fast mode, or ultra mode. For differential signaling, the maximum SCSI cable length possible is the same for standard mode and fast mode.

Table 4–5 summarizes how the type of signaling method affects SCSI interconnect distances.

Table 4–5 Maximum SCSI Interconnect Distances

Signaling Technique	Rate of Data Transfer	Maximum Cable Length
Single ended	Standard	6 m[1]
Single ended	Fast	3 m
Single ended	Ultra	20.5 m[2]
Differential	Standard or Fast	25 m
Differential	Ultra	25.5 m[3]

[1]The SCSI standard specifies a maximum length of 6 m for this interconnect. However, it is advisable, where possible, to limit the cable length to 4 m to ensure the highest level of data integrity.

[2]This length is attainable if devices are attached only at each end. If devices are spaced along the interconnect, they must be at least 1 m apart, and the interconnect cannot exceed 4 m.

[3]More than two devices can be supported.

4.8.4 Supported Adapters, Bus Types, and Computers

Table 4–6 shows SCSI adapters with the internal buses and computers they support.

Table 4–6 SCSI Adapters

Adapter	Internal Bus	Supported Computers
Embedded (NCR-810 based)/KZPAA[1]	PCI	AlphaServer 400
		AlphaServer 1000
		AlphaServer 2000
		AlphaServer 2100
		AlphaStation 200
		AlphaStation 250
		AlphaStation 400
		AlphaStation 600
KZPSA[2]	PCI	Supported on all Alpha computers that support KZPSA in single-host configurations.[3]
KZTSA[2]	TURBOchannel	DEC 3000
KZPBA-CB[4]	PCI	Supported on all Alpha computers that support KZPBA in single-host configurations.[3]

[1] Single-ended.

[2] Fast-wide differential (FWD).

[3] See the system-specific hardware manual.

[4] Ultra differential. The ultra single-ended adapter (KZPBA-CA) does not support multihost systems.

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.9 CI Interconnect

The CI interconnect is a radial bus through which OpenVMS Cluster systems communicate. It comprises the following components:

- CI host adapter.
- Star coupler—A passive device that serves as a common connection point for signals between OpenVMS nodes and HSC or HSJ controllers that are connected by the CI.
- Optional star coupler expander (CISCE)—Consists of two amplifiers, one for each of its two paths.
- CI cable.

4.9.1 Advantages

The CI interconnect offers the following advantages:

- High speed

  Suitable for larger processors and I/O-intensive applications.

- Efficient access to large amounts of storage

  HSC and HSJ controllers can connect large numbers of disk and tape drives to the OpenVMS Cluster system, with direct access from all OpenVMS nodes on the CI.

- Minimal CPU overhead for communication

  CI adapters are intelligent interfaces that perform much of the work required for communication among OpenVMS nodes and storage. The CI topology allows all nodes attached to a CI bus to communicate directly with the HSC and HSJ controllers on the same CI bus.

- High availability through redundant, independent data paths

  Each CI adapter is connected with two pairs of CI cables. If a single CI cable connection fails, failover automatically occurs.

- Multiple access paths to disks and tapes

  Dual HSC and HSJ controllers and dual-ported devices create alternative paths to storage devices.

4.9.2 Throughput

The CI interconnect has a high maximum throughput. CI adapters use high-performance microprocessors that perform many of the processing activities usually performed by the CPU. As a result, they consume minimal CPU processing power.

Because the effective throughput of the CI bus is high, a single CI interconnect is not likely to be a bottleneck in a large OpenVMS Cluster configuration. If a single CI is not sufficient, multiple CI interconnects can increase throughput.

4.9.3 Supported Adapters and Bus Types

The following are CI adapters and internal buses that each supports:

- CIPCA (PCI/EISA)
- CIXCD (XMI)
- CIBCA–B (VAXBI)

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.9.4 Multiple CI Adapters

You can configure multiple CI adapters on some OpenVMS nodes. Multiple star couplers can be used in the same OpenVMS Cluster.

With multiple CI adapters on a node, adapters can share the traffic load. This reduces I/O bottlenecks and increases the total system I/O throughput. Table 4–7 lists the limits for multiple CI adapters per system.

Table 4–7 Maximum CI Adapters per System

System	CIPCA	CIBCA–A	CIBCA–B	CIXCD	Comments
AlphaServer 8400	26	-	-	10	Can use a combination of CIPCA and CIXCD adapters, not to exceed 26. Prior to OpenVMS Version 7.1, the maximum is 10.
AlphaServer 8200	26	-	-	-	Prior to OpenVMS Version 7.1, the maximum is 10.
AlphaServer 4000, 4100	3	-	-	-	When using three CIPCAs, one must be a CIPCA-AA and two must be CIPCA-BA.
AlphaServer 4000 with I/O expansion module	6	-	-	-	When using six CIPCAs, only three can be CIPCA-AA.
AlphaServer 2100A	3	-	-	-	-
AlphaServer 2000, 2100	2	-	-	-	Only one can be a CIPCA-BA.
AlphaServer 1200	2	-	-	-	Only one can be a CIPCA-AA.
DEC 7000/10000	-	-	-	10	-
VAX 6000	-	1	4	4	CIPCA is not supported on a VAX.
VAX 7000, 10000	-	-	-	10	CIPCA is not supported on a VAX. CIBCA is not supported on these models.

Reference: For more extensive information about the CIPCA adapter, see Appendix C.

4.9.5 Configuration Guidelines for CI Clusters

Use the following guidelines when configuring systems in a CI cluster:

- The maximum number of nodes that you can connect to a star coupler is 32. Up to 16 of these nodes can be OpenVMS systems, and the remainder can be HSJ and HSC storage controllers.
- The number of star couplers is limited by the number of CI adapters configured on a system.
- Dual porting of devices between HSJ and HSC controllers is supported as long as they are connected to the same or separate star couplers. Dual porting of devices between HSJ and HSC controllers and local controllers is not supported.
- With the exception of the CIPCA and CIXCD, different types of CI adapters cannot be combined in the same system.
- You can use multiple CI adapters for redundancy and throughput. You can increase throughput by connecting additional CI adapters to separate star couplers; throughput does not increase substantially when you connect a second CI adapter to the same star coupler.

4.10 Digital Storage Systems Interconnect (DSSI)

DSSI is a single-path, daisy-chained, multidrop bus. It provides a single, 8-bit parallel data path with both byte parity and packet checksum for error detection.

4.10.1 Advantages

DSSI offers the following advantages:

- High reliability
- Shared, direct access to storage at a lower cost than CI
- Direct communication between systems and storage
- High-performance, intelligent storage controllers with embedded caches

4.10.2 Maintenance Consideration

DSSI storage often resides in the same cabinet as the CPUs. For these configurations, the whole system may need to be shut down for service, unlike configurations and interconnects with separately housed systems and storage devices.

4.10.3 Throughput

The maximum throughput is 32 Mb/s.

DSSI has highly intelligent adapters that require minimal CPU processing overhead.

4.10.4 DSSI Adapter Types

There are two types of DSSI adapters:

- Embedded adapter, which is part of the system.
- Optional adapter, which you can purchase separately and add to the system.

4.10.5 Supported Adapters and Bus Types

The following are DSSI adapters and internal bus that each supports:

- KFESA (EISA)
- KFESB (EISA)
- KFPSA (PCI)
- KFMSA (XMI)—VAX only
- KFMSB (XMI)—Alpha only
- KFQSA (Q–bus)
- N710 (embedded)
- SHAC (embedded)
- EDA640 (embedded)

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.10.6 DSSI-Connected Storage

DSSI configurations use HSD intelligent controllers to connect disk drives to an OpenVMS Cluster. HSD controllers serve the same purpose with DSSI as HSJ controllers serve with CI: they enable you to configure more storage.

Alternatively, DSSI configurations use integrated storage elements (ISEs) connected directly to the DSSI bus. Each ISE contains either a disk and disk controller or a tape and tape controller.

4.10.7 Multiple DSSI Adapters

Multiple DSSI adapters are supported for some systems, enabling higher throughput than with a single DSSI bus.

Table 4–8 lists the limitations for multiple DSSI adapters. You can also refer to the most recent OpenVMS Cluster SPD for the latest information about DSSI adapters.

Table 4–8 DSSI Adapters per System

System	Embedded	KFPSA[1]	KFQSA[2]	KFESA	KFESB	KFMSA[3]	KFMSB[3]
AlphaServer 8400	-	4	-	-	-	-	12
AlphaServer 8200, 4100	-	4	-	-	-	-	-
AlphaServer 2100	-	4	-	-	-	-	-
AlphaServer 2000, 1000	-	4	-	-	4	-	-
DEC 4000 (embedded N710)	2	-	-	-	-	-	-
DEC 7000/10000	-	-	-	-	-	-	12
MicroVAX II, 3500, 3600, 3800, 3900	-	-	2	-	-	-	-
MicroVAX 3300/3400 (embedded EDA640)	1	-	2	-	-	-	-
VAX 4000 Model 105A (embedded SHAC[4])	1 + 1[4]	-	2[5]	-	-	-	-
VAX 4000 Model 200 (embedded SHAC[4])	1	-	2	-	-	-	-
VAX 4000 Model 300, 400, 500, 600	2	-	2	-	-	-	-
VAX 4000 Model 505A /705A (embedded SHAC[3])	2 + 2[6]	-	2	-	-	-	-
VAX 6000	-	-	-	-	-	6	-
VAX 7000	-	-	-	-	-	12	-

[1] The KFPSA cannot be configured on the same DSSI as a KFMSB. Ensure that the specific AlphaServer system has sufficient PCI backplane slots to accept the number of KFPSAs required.

[2] The KFQSA cannot be used for node-to-node cluster communication. An additional interconnect must be configured between systems that use KFQSA for access to shared storage.

[3] Each KFMSA/B (XMI-to-DSSI) adapter contains two DSSI VAX system ports.

[4] Single Host Adapter Chip (SHAC).

[5] Requires a Q–bus expansion enclosure.

[6] Additional adapter embedded on daughter card.

4.10.8 Configuration Guidelines for DSSI Clusters

The following configuration guidelines apply to all DSSI clusters:

- Each DSSI interconnect can have up to eight nodes attached; four can be systems and the rest can be storage devices. Each of the following counts as a DSSI node:
  - DSSI adapter
  - Any member of the HSD*xx* family of DSSI and SCSI controllers
  - Any RF, TF, or EF integrated storage element (ISE)

  In some cases, physical cabling and termination limitations may restrict the number of systems to two or three that may be connected to a DSSI interconnect. For example:
  - Some variants of the DSSI adapter terminate the bus; for example, the N710. For this reason, only two DEC 4000 systems can be configured on a DSSI interconnect.
  - The size of a DEC or VAX 10000 system generally limits the number of systems that can be connected to a DSSI interconnect.
- Each DSSI adapter in a single system must be connected to a different DSSI bus.
- Configure VAX 6000, VAX 7000, and VAX 10000 systems with KFMSA adapters.
- Configure DEC 7000 and DEC 10000 systems with KFMSB adapters.
- Configure PCI-based AlphaServer systems with KFPSA adapters. EISA adapters (KFESA/KFESB) adapters can also be configured on most AlphaServer systems, but use of KFPSA is recommended whenever possible.
- Dual porting of devices between HSD controllers is supported as long as they are connected to the same or separate DSSI interconnects. Dual porting of devices between HSD controllers and local controllers is not supported.
- All systems connected to the same DSSI bus must have a common power or ground.

Reference: For more information about DSSI, see the *DSSI OpenVMS Cluster Installation and Troubleshooting Manual*.

4.11 Ethernet Interconnect

The Ethernet interconnect provides single path connections within an OpenVMS Cluster system and a local area network (LAN). Ethernet and FDDI are both LAN-based interconnects. See Section 4.12 for information about FDDI and for general LAN-based cluster guidelines.

4.11.1 Advantages

The Ethernet interconnect offers the following advantages:

- Lowest cost of all OpenVMS Cluster interconnects.
- Supports 96 nodes on a single interconnect.
- Allows for extended physical distribution of nodes.

4.11.2 Throughput

The maximum throughput of the Ethernet interconnect is 10 Mb/s. Because Ethernet adapters do not provide hardware assistance, processor overhead is higher than for CI or DSSI.

General network traffic on an Ethernet can reduce the throughput available for OpenVMS Cluster communication. The Ethernet can become an I/O bottleneck in an OpenVMS Cluster system. Therefore, consider the capacity of the total network design when you configure an OpenVMS Cluster system with many Ethernet-connected nodes or when the Ethernet also supports a large number of PCs or printers.

Reference: For information about reducing congestion on an Ethernet LAN, see Section 10.7.7.

4.11.3 Multiple Ethernet Load Balancing

If only Ethernet paths are available, the choice between which path the OpenVMS Cluster software uses is based on latency (computed network delay). If delays are equal, either path can be used. Otherwise, the OpenVMS Cluster software chooses the channel with the least latency. The network delay across each segment is calculated approximately every 3 seconds. Traffic is then balanced across all communication paths between local and remote adapters.

4.11.4 Supported Adapters and Buses

The following are Ethernet adapters and the internal bus that each supports:

- DETRA (TURBOchannel)
- DEFTA-FA (TURBOchannel)
- DEFTA-*xx* (TURBOchannel)
- DGLTA (TURBOchannel)
- DE2*xx* (ISA)
- DW110 (ISA)
- DE42*x* (EISA)
- DE435 (PCI)
- DW300 (EISA)
- TULIP (PCI)
- KZPSM (PCI)
- DE4*xx* (PCI)
- DE500-*xx* (PCI)
- TC4048 (PCI)
- DGLPB (PCI)
- 3COM (PCMCIA)
- DEMNA (XMI)
- TGEC (embedded)
- COREIO (TURBOchannel)
- PMAD (TURBOchannel)

- DE422 (EISA)
- DEBNI (VAXBI)
- DEBNA (VAXBI)
- SGEC (embedded)
- DESVA (embedded)
- DESQA (Q–bus)
- DELQA (Q–bus)

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.11.5 Ethernet-to-FDDI Bridges

You can use transparent Ethernet-to-FDDI translating bridges to provide an interconnect between a 10-Mb/s Ethernet segment and a 100-Mb/s FDDI ring. These Ethernet-to-FDDI bridges are also called "10/100" bridges. They perform high-speed translation of network data packets between the FDDI and Ethernet frame formats.

Reference: See Figure 10–21 for an example of these bridges.

4.12 Fiber Distributed Data Interface (FDDI)

FDDI is an ANSI standard LAN interconnect that uses fiber-optic cable. FDDI supports OpenVMS Cluster functionality over greater distances than other interconnects. FDDI also augments the Ethernet by providing a high-speed interconnect for multiple Ethernet segments in a single OpenVMS Cluster system.

4.12.1 Advantages

FDDI offers the following advantages:

- Supports 96 nodes on a single interconnect.
- Combines high throughput and longest distance between nodes.
- Supports a variety of topologies.

4.12.2 Types of FDDI Nodes

The FDDI standards define the following two types of nodes:

- Stations — The ANSI standard single-attachment station (SAS) and dual-attachment station (DAS) can be used as an interconnect to the FDDI ring. It is advisable to attach stations to wiring concentrators and to attach the wiring concentrators to the dual FDDI ring, making the ring more stable.
- Wiring concentrator — The wiring concentrator (CON) provides a connection for multiple SASs or CONs to the FDDI ring. A DECconcentrator 500 is an example of this device.

4.12.3 Distance

FDDI limits the total fiber path to 200 km (125 miles). The maximum distance between adjacent FDDI devices is 40 km with single-mode fiber and 2 km with multimode fiber. In order to control communication delay, however, it is advisable to limit the maximum distance between any two OpenVMS Cluster nodes on an FDDI ring to 40 km.

4.12.4 Throughput

The maximum throughput of the FDDI interconnect (100 Mb/s) is 10 times higher than that of Ethernet.

In addition, FDDI supports transfers using large packets (up to 4468 bytes). Only FDDI nodes connected exclusively by FDDI can make use of large packets.

Because FDDI adapters do not provide processing assistance for OpenVMS Cluster protocols, more processing power is required than for CI or DSSI.

4.12.5 Supported Adapters and Bus Types

Following is a list of FDDI adapters and the buses they support:

- DEFPA (PCI)
- DEFPZ (integral)
- DEMFA (XMI)
- DEFAA (Futurebus+)
- DEFTA (TURBOchannel)
- DEFZA (TURBOchannel)
- DEFEA (EISA)
- DEFQA (Q–bus)

Reference: For complete information about each adapter's features and order numbers, see the *DIGITAL Systems and Options Catalog*.

To access the most recent *DIGITAL Systems and Options Catalog* on the World Wide Web, use the following URL:

`http://www.digital.com:80/info/soc/`

4.12.6 Configuration Guidelines for FDDI-Based Clusters

FDDI-based configurations use FDDI for node-to-node communication. The following general guidelines apply to FDDI configurations:

- The HS1*xx* and HS2*xx* family of storage servers provide FDDI-based storage access to OpenVMS Cluster nodes.
- Alpha and VAX systems can be configured with any mix of FDDI adapters.
- All FDDI paths used for OpenVMS Cluster communication must operate with a minimum of 10 Mb/s throughput and low latency. You must use translating bridges when connecting nodes on an Ethernet to those on an FDDI. LAN segments can be bridged to form an extended LAN.
- Multiple, distinct OpenVMS Cluster systems can be configured onto a single, extended LAN. OpenVMS Cluster software performs cluster membership validation to ensure that systems join the correct LAN OpenVMS cluster.

Reference: For information about multiple LANs, see Section 8.6. For extended LAN (ELAN) configuration guidelines, see Table 10–3.

4.12.7 Multiple FDDI Adapters

Because FDDI is ideal for spanning great distances, you may want to supplement its high throughput with high availability by ensuring that critical nodes are connected to multiple FDDI rings. Physical separation of the two FDDI paths helps ensure that the configuration is disaster tolerant.

4.12.8 Multiple FDDI Load Balancing

If only FDDI paths are available, the OpenVMS Cluster software bases the choice between which path to use on latency (computed network delay). If delays are equal, either path can be used. Otherwise, OpenVMS Cluster software chooses the channel with the least latency. The network delay across each segment is calculated approximately every 3 seconds. Traffic is balanced across all communication paths between local and remote adapters.

5

Choosing OpenVMS Cluster Storage Subsystems

This chapter describes how to design a storage subsystem. The design process involves the following steps:

1. Understanding storage product choices
2. Estimating storage capacity requirements
3. Choosing disk performance optimizers
4. Determining disk availability requirements
5. Understanding advantages and tradeoffs for:
   - CI based storage
   - DSSI-based storage
   - SCSI-based storage
   - Host-based storage
   - LAN InfoServer

The rest of this chapter contains sections that explain these steps in detail.

5.1 Understanding Storage Product Choices

In an OpenVMS CLuster, storage choices include the StorageWorks family of products, a modular storage expansion system based on the Small Computer Systems Interface (SCSI–2) standard. StorageWorks helps you configure complex storage subsystems by choosing from the following modular elements:

- Storage devices such as disks, tapes, CD–ROMs, and solid-state disks
- Array controllers
- Power supplies
- Packaging
- Interconnects
- Software

5.1.1 Criteria for Choosing Devices

Consider the following criteria when choosing storage devices:

- Supported interconnects
- Capacity
- I/O rate
- Floor space
- Purchase, service, and maintenance cost

5.1.2 How Interconnects Affect Storage Choices

One of the benefits of OpenVMS Cluster systems is that you can connect storage devices directly to OpenVMS Cluster interconnects to give member systems access to storage.

In an OpenVMS Cluster system, the following storage devices and adapters can be connected to OpenVMS Cluster interconnects:

- HSJ and HSC controllers (on the CI)
- HSD controllers and ISEs (on the DSSI)
- HSZ and RZ series (on the SCSI)
- HSG controllers (on the Fibre Channel)
- Local system adapters

Table 5–1 lists the kinds of storage devices that you can attach to specific interconnects.

Table 5–1 Interconnects and Corresponding Storage Devices

Storage Interconnect	Storage Devices
CI	HSJ and HSC controllers and SCSI storage
DSSI	HSD controllers, ISEs, and SCSI storage
SCSI	HSZ controllers and SCSI storage
Fibre Channel	HSG controllers and SCSI storage
FDDI	HS*xxx* controllers and SCSI storage

5.1.3 How Floor Space Affects Storage Choices

If the cost of floor space is high and you want to minimize the floor space used for storage devices, consider these options:

- Choose disk storage arrays for high capacity with small footprint. Several storage devices come in stackable cabinets for labs with higher ceilings.
- Choose high-capacity disks over high-performance disks.
- Make it a practice to upgrade regularly to newer storage arrays or disks. As storage technology improves, storage devices are available at higher performance and capacity and reduced physical size. For example, replacing an HSC95 and SA800 with an HSJ40 and SW800 increases capacity and reduces floor-space consumption.
- Plan adequate floor space for power and cooling equipment.

5.2 Determining Storage Capacity Requirements

Storage capacity is the amount of space needed on storage devices to hold system, application, and user files. Knowing your storage capacity can help you to determine the amount of storage needed for your OpenVMS Cluster configuration.

5.2.1 Estimating Disk Capacity Requirements

To estimate your online storage capacity requirements, add together the storage requirements for your OpenVMS Cluster system's software, as explained in Table 5–2.

Table 5–2 Estimating Disk Capacity Requirements

Software Component	Description
OpenVMS operating system	Estimate the number of blocks[1] required by the OpenVMS operating system. **Reference:** Your OpenVMS installation documentation and Software Product Description (SPD) contain this information.
Page, swap, and dump files	Use AUTOGEN to determine the amount of disk space required for page, swap, and dump files. **Reference:** The *OpenVMS System Manager's Manual* provides information about calculating and modifying these file sizes.
Site-specific utilities and data	Estimate the disk storage requirements for site-specific utilities, command procedures, online documents, and associated files.
Application programs	Estimate the space required for each application to be installed on your OpenVMS Cluster system, using information from the application suppliers. **Reference:** Consult the appropriate Software Product Description (SPD) to estimate the space required for normal operation of any layered product you need to use.
User-written programs	Estimate the space required for user-written programs and their associated databases.
Databases	Estimate the size of each database. This information should be available in the documentation pertaining to the application-specific database.

[1] Storage capacity is measured in blocks. Each block contains 512 bytes.

(continued on next page)

Table 5–2 (Cont.) Estimating Disk Capacity Requirements

Software Component	Description
User data	Estimate user disk-space requirements according to these guidelines: • Allocate from 10,000 to 100,000 blocks for each occasional user. An occasional user reads, writes, and deletes electronic mail; has few, if any, programs; and has little need to keep files for long periods. • Allocate from 250,000 to 1,000,000 blocks for each moderate user. A moderate user uses the system extensively for electronic communications, keeps information on line, and has a few programs for private use. • Allocate 1,000,000 to 3,000,000 blocks for each extensive user. An extensive user can require a significant amount of storage space for programs under development and data files, in addition to normal system use for electronic mail. This user may require several hundred thousand blocks of storage, depending on the number of projects and programs being developed and maintained.
Total requirements	The sum of the preceding estimates is the approximate amount of disk storage presently needed for your OpenVMS Cluster system configuration.

5.2.2 Additional Disk Capacity Requirements

Before you finish determining your total disk capacity requirements, you may also want to consider future growth for online storage and for backup storage.

For example, at what rate are new files created in your OpenVMS Cluster system? By estimating this number and adding it to the total disk storage requirements that you calculated using Table 5–2, you can obtain a total that more accurately represents your current and future needs for online storage.

To determine backup storage requirements, consider how you deal with obsolete or archival data. In most storage subsystems, old files become unused while new files come into active use. Moving old files from online to backup storage on a regular basis frees online storage for new files and keeps online storage requirements under control.

Planning for adequate backup storage capacity can make archiving procedures more effective and reduce the capacity requirements for online storage.

5.3 Choosing Disk Performance Optimizers

Estimating your anticipated disk performance work load and analyzing the work load data can help you determine your disk performance requirements.

You can use the Monitor utility and DECamds to help you determine which performance optimizer best meets your application and business needs.

5.3.1 Performance Optimizers

Performance optimizers are software or hardware products that improve storage performance for applications and data. Table 5–3 explains how various performance optimizers work.

Table 5–3 Disk Performance Optimizers

Optimizer	Description
DECram for OpenVMS	A disk device driver that enables system managers to create logical disks in memory to improve I/O performance. Data on an in-memory DECram disk can be accessed at a faster rate than data on hardware disks. DECram disks are capable of being shadowed with Volume Shadowing for OpenVMS and of being served with the MSCP server.[1]
Solid-state disks	In many systems, approximately 80% of the I/O requests can demand information from approximately 20% of the data stored on line. Solid-state devices can yield the rapid access needed for this subset of the data.
Disk striping	Disk striping (RAID level 0) lets applications access an array of disk drives in parallel for higher throughput. Disk striping works by grouping several disks into a "stripe set" and by dividing the application data into "chunks," which are spread equally across the disks in the stripe set in a round-robin fashion. By reducing access time, disk striping may improve performance, especially if the application: • Performs large data transfers in parallel. • Requires load balancing across drives. Two independent types of disk striping are available: • Controller-based striping, in which HSJ and HSD controllers combine several disks into a single stripe set. This stripe set is presented to OpenVMS as a single volume. This type of disk striping is hardware based. • Host-based striping, which creates stripe sets on an OpenVMS host. The OpenVMS software breaks up an I/O request into several simultaneous requests that it sends to the disks of the stripe set. This type of disk striping is software based. **Note:** You can use Volume Shadowing for OpenVMS software in combination with disk striping to make stripe set members redundant. You can shadow controller-based stripe sets or you can host-based disk stripe shadow sets.
Virtual I/O cache (VIOC)	OpenVMS offers host-based caching in the form of VIOC, a clusterwide, file-oriented disk cache. VIOC reduces I/O bottlenecks within OpenVMS Cluster systems by reducing the number of I/Os from the system to the disk subsystem.
Controllers with disk cache	Some storage technologies use memory to form disk caches. Accesses that can be satisfied from the cache can be done almost immediately and without any seek time or rotational latency. For these accesses, the two largest components of the I/O response time are eliminated. The HSC, HSJ, HSD, and HSZ controllers contain caches. Every RF and RZ disk has a disk cache as part of its embedded controller.

[1]The MSCP server makes locally connected disks to which it has direct access available to other systems in the OpenVMS Cluster.

Reference: See Section 10.8 for more information about how these performance optimizers increase an OpenVMS Cluster's ability to scale I/Os.

5.4 Determining Disk Availability Requirements

For storage subsystems, availability is determined by the availability of the storage device as well as the availability of the path to the device.

5.4.1 Availability Requirements

Some costs are associated with optimizing your storage subsystems for higher availability. Part of analyzing availability costs is weighing the cost of protecting data against the cost of unavailable data during failures. Depending on the nature of your business, the impact of storage subsystem failures may be low, moderate, or high.

Device and data availability options reduce and sometimes negate the impact of storage subsystem failures.

5.4.2 Device and Data Availability Optimizers

Depending on your availability requirements, choose among the availability optimizers described in Table 5–4 for applications and data with the greatest need.

Table 5–4 Storage Availability Optimizers

Availability Optimizer	Description
Redundant access paths	Protect against hardware failures along the path to the device by configuring redundant access paths to the data.
Volume Shadowing for OpenVMS software	Replicates data written to a virtual disk by writing the data to one or more physically identical disks that form a shadow set. With replicated data, users can access data even when one disk becomes unavailable. If one shadow set member fails, the shadowing software removes the drive from the shadow set, and processing continues with the remaining drives. Shadowing is transparent to applications and allows data storage and delivery during media, disk, controller, and interconnect failure. A shadow set can contain up to three members, and shadow set members can be anywhere within the storage subsystem of an OpenVMS Cluster system. **Reference:** See *Volume Shadowing for OpenVMS* for more information about volume shadowing.
System disk redundancy	Place system files judiciously on disk drives with multiple access paths. OpenVMS Cluster availability increases when you form a shadow set that includes the system disk. You can also configure an OpenVMS Cluster system with multiple system disks. **Reference:** For more information, see Section 11.2.
Database redundancy	Keep redundant copies of certain files or partitions of databases that are, for example, updated overnight by batch jobs. Rather than using shadow sets, which maintain a complete copy of the entire disk, it might be sufficient to maintain a backup copy on another disk or even on a standby tape of selected files or databases.

(continued on next page)

Table 5–4 (Cont.) Storage Availability Optimizers

Availability Optimizer	Description
DECevent	DECevent, in conjunction with volume shadowing, can detect most imminent device failures with sufficient lead time to move the data to a spare device.
	Enhance device reliability with appropriate software tools. Use device-failure prediction tools, such as DECevent, where high availability is needed. When a shadow set member has an increasing fault rate that might indicate potential failure, DECevent works with Volume Shadowing for OpenVMS to make a shadow set copy of the suspect device to a spare device. After the copy is made, the suspect device can be taken off the system for examination and repair without loss of data availability.
Newer devices	Protect against failure by choosing newer devices. Typically, newer devices provide improved reliability and mean time between failures (MTBF). Newer controllers also improve reliability by employing updated chip technologies.
Implement thorough backup strategies	Frequent and regular backups are the most effective way to ensure the availability of your data.

5.5 CI Based Storage

The CI interconnect provides the highest OpenVMS Cluster availability with redundant, independent transmit-and-receive CI cable pairs. The CI offers multiple access paths to disks and tapes by means of dual-ported devices between HSC or HSJ controllers.

5.5.1 Supported Controllers and Devices

The following controllers and devices are supported by the CI interconnect:

- HSJ storage controllers
  - SCSI devices (RZ,TZ, EZ)
- HSC storage controllers
  - SDI and STI devices (RA, ESE, TA)
  - K.SCSI devices (RZ, TZ, EZ)

5.6 DSSI Storage

DSSI-based configurations provide shared direct access to storage for systems with moderate storage capacity. The DSSI interconnect provides the lowest-cost shared access to storage in an OpenVMS Cluster.

The storage tables in this section may contain incomplete lists of products.

5.6.1 Supported Devices

DSSI configurations support the following devices:

- EF-series solid-state disks
- RF-series disks
- TF-series tapes
- DECarray storage arrays

- HSD storage controller
  - SCSI devices (RZ,TZ, EZ)

Reference: RZ, TZ, and EZ SCSI storage devices are described in Section 5.7.

5.7 SCSI-Based Storage

The Small Computer Systems Interface (SCSI) bus is a storage interconnect based on an ANSI industry standard. You can connect up to a total of 8 or 16 nodes (3 of which can be CPUs) to the SCSI bus.

5.7.1 Supported Devices

The following devices can connect to a single host or multihost SCSI bus:

- RZ-series disks
- HSZ storage controllers

The following devices can connect only to a single host SCSI bus:

- EZ-series disks
- RRD-series CD–ROMs
- TZ-series tapes

5.8 Fibre Channel Based Storage

The Fibre Channel interconnect is a storage interconnect that is based on an ANSI industry standard.

5.8.1 Storage Devices

The HSG storage controllers can connect to a single host or to a multihost Fibre Channel interconnect.

5.9 Host-Based Storage

Host-based storage devices can be connected locally to OpenVMS Cluster member systems using local adapters. You can make this locally connected storage available to other OpenVMS Cluster members by configuring a node as an MSCP server.

You can use local adapters to connect each disk to two access paths (dual ports). Dual porting allows automatic failover of disks between nodes.

5.9.1 Internal Buses

Locally connected storage devices attach to a system's internal bus.

Alpha systems use the following internal buses:

- PCI
- EISA
- XMI
- SCSI
- TURBOchannel
- Futurebus+

VAX systems use the following internal buses:

- VAXBI
- XMI
- Q–bus
- SCSI

5.9.2 Local Adapters

Following is a list of local adapters and their bus types:

- KGPSA (PCI)
- KZPSM (PCI)
- KZPDA (PCI)
- KZPSC (PCI)
- KZPAC (PCI)
- KZESC (EISA)
- KZMSA (XMI)
- PB2HA (EISA)
- PMAZB (TURBOchannel)
- PMAZC (TURBOchannel)
- KDM70 (XMI)
- KDB50 (VAXBI)
- KDA50 (Q–bus)

6

Configuring Multiple Paths to SCSI and Fibre Channel Storage

This chapter describes multipath SCSI support for parallel SCSI and Fibre Channel configurations. Multipath support is available on OpenVMS Alpha Version 7.2.

Note

At the time that OpenVMS Version 7.2 is released, the following restrictions apply:

- Fibre Channel configurations are not supported.
- Devices with multiple SCSI paths can not be members of host-based shadow sets.
- Failover between a local path to a SCSI device and an MSCP-served path to that same device is not supported. The default setting of the MPDEV_REMOTE system parameter (MPDEV_REMOTE = 0) in OpenVMS V7.2 turns off this type of failover. (Although failover to an MSCP path is not supported, multipath devices **can be MSCP served** to other systems in an OpenVMS Cluster system.)

These restrictions will be removed by an update kit, shortly after the release of OpenVMS Version 7.2. The information in this chapter pertaining to these features is provided in advance as an aid for future planning.

The following topics are presented in this chapter:

- Overview of multipath SCSI support (Section 6.1)
- HS*x* failover modes (Section 6.2)
- Path selection by OpenVMS (Section 6.3)
- Configuration requirements and restrictions (Section 6.4)
- Parallel SCSI multipath configurations (Section 6.5)
- Device naming for parallel SCSI multipath configurations (Section 6.6)
- Fibre Channel multipath configurations (Section 6.7)
- Implementing multipath configurations (Section 6.8)

6.1 Overview of Multipath SCSI Support

A multipath SCSI configuration provides failover from one path to a device to another path to the same device. Multiple paths to the same device increase the availability of that device for I/O operations. Multiple paths also offer higher aggregate performance. Figure 6–1 shows a multipath SCSI configuration. Two paths are configured from a computer to the same virtual storage device.

Multipath SCSI configurations can use either parallel SCSI or Fibre Channel as the storage interconnect, as illustrated by Figure 6–1.

Two or more paths to a single device are called a **multipath set**. When the system configures a path to a device, it checks for an existing device with the same name but a different path. If such a device is found, and multipath support is enabled, the system either forms a multipath set or adds the new path to an existing set. If multipath support is not enabled, then no more than one path to a device is configured.

The system presents a multipath set as a single device. The system selects one path to the device as the "current" path, and performs all I/O over this path until there is a failure or the system manager requests that the system switch to another path.

Multipath SCSI support provides two types of failover:

- Direct SCSI to direct SCSI
- Direct SCSI to MSCP served

Direct SCSI to direct SCSI failover requires the use of multiported SCSI devices. Direct SCSI to MSCP served failover requires multiple hosts per SCSI bus, but does not require multiported SCSI devices. These two failover types can be combined. Each type and the combination of the two are described next.

6.1.1 Direct SCSI to Direct SCSI Failover

Direct SCSI to direct SCSI failover can be used on systems with multiported SCSI devices. The dual HSZ70, the HSZ80 and the HSG80 are examples of multiported SCSI devices. A multiported SCSI device can be configured with multiple ports on the same physical interconnect so that if one of the ports fails, the host can continue to access the device through another port. This is known as **transparent failover** mode and has been supported by OpenVMS since Version 6.2.

OpenVMS Version 7.2 introduces support for a new failover mode in which the multiported device can be configured with its ports on different physical interconnects. This is known as **multibus failover** mode.

The HS*x* failover modes are selected by HS*x* console commands. Transparent and multibus modes are described in more detail in Section 6.2.

A generic illustration of a multibus failover configuration is shown in Figure 6–1.

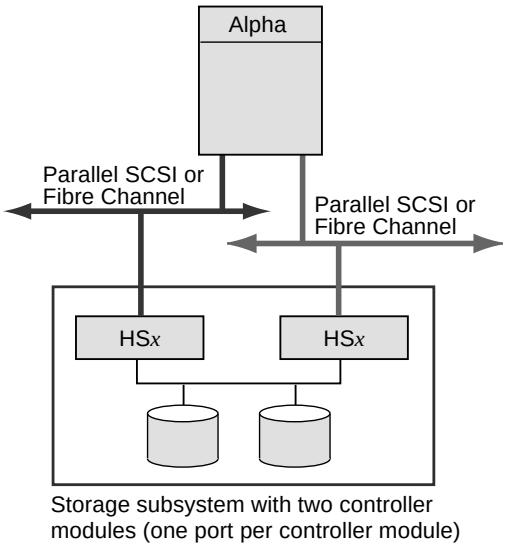
The two logical disk devices shown in Figure 6–1 represent virtual storage units that are presented to the host by the HS*x* controller modules. Each logical storage unit is "online" to one of the two HS*x* controller modules at a time. When there are multiple logical units, they can be online to different HS*x* controllers so that both HS*x* controllers can be active at the same time.

In transparent mode, a logical unit switches from one controller to the other when an HS*x* controller detects that the other controller is no longer functioning.

In multibus mode, a logical unit connection is switched from one controller to the other when one of the following events occurs:

- One HS*x* controller detects that the other controller is no longer functioning.
- The OpenVMS multipath software detects that the current path has failed.
- The OpenVMS system manager issues a command to cause a switch.

Figure 6–1 Multibus Failover Configuration



Note the following about this configuration:

- Host has two adapters.
- Interconnects can both be parallel SCSI (HSZ70 or HSZ80) or both be Fibre Channel (HSG80) but not mixed.
- Storage cabinet contains two HS*x* controllers configured for multibus failover mode.

The multibus configuration offers the following advantages over transparent failover:

- Higher aggregate performance with two host adapters and two HS*x* controller modules in operation.
- Higher availability because the storage is still accessible when a host adapter, the interconnect, or the HS*x* controller module on a path fails.

6.1.2 Direct SCSI to MSCP Served Failover

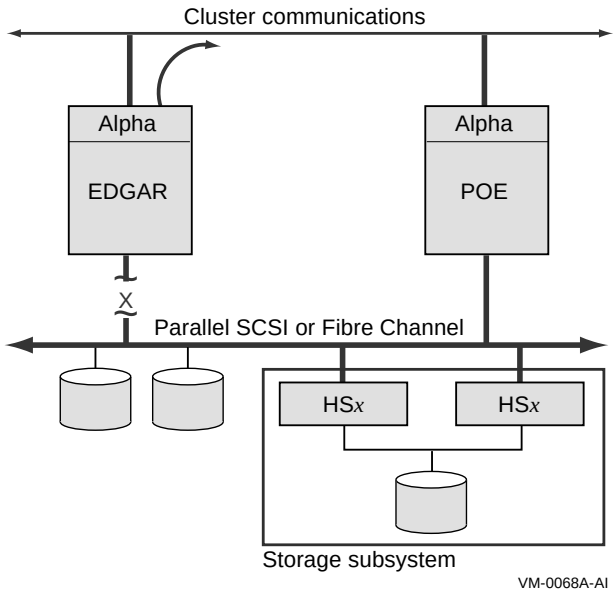
Note

The feature described in this section is not supported at the time OpenVMS Version 7.2 is released. This restriction will be removed by an update kit, shortly after the release of OpenVMS Version 7.2.

OpenVMS provides support for multiple hosts that share a SCSI bus. This is known as a multihost SCSI OpenVMS Cluster system. In this configuration, the SCSI bus is a shared storage interconnect. Cluster communication occurs over a second interconnect (LAN, DSSI, CI, or MEMORY CHANNEL).

Multipath support in a multihost SCSI OpenVMS Cluster system enables failover from directly attached SCSI storage to MSCP served SCSI storage, as shown in Figure 6–2.

Figure 6–2 Direct SCSI to MSCP Served Configuration With One Interconnect



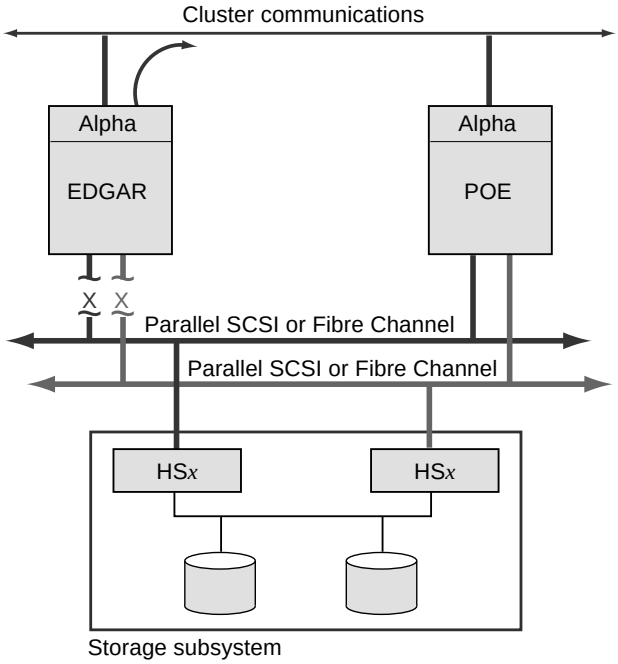
Note the following about this configuration:

- Two hosts are connected to a shared storage interconnect.
- Two hosts are connected by a second interconnect (LAN, CI, DSSI, or MEMORY CHANNEL) for cluster communications.
- The storage devices can have a single port or multiple ports.
- If node Edgar's SCSI connection to the storage fails, the SCSI storage is MSCP served by the remaining host over the cluster interconnect.

6.1.3 Configurations Combining Both Types of Multipath Failover

In a multihost SCSI OpenVMS cluster system, you can increase storage availability by configuring the cluster for both types of multipath failover (direct SCSI to direct SCSI and direct SCSI to MSCP served SCSI), as shown in Figure 6–3.

Figure 6–3 Direct SCSI to MSCP Served Configuration With Two Interconnects



Note the following about this configuration:

- Both nodes are directly connected to both storage interconnects.
- Both nodes are connected to a second interconnect for cluster communications.
- Each HS*x* storage controller is connected to only one interconnect.
- Both HS*x* storage controllers are in the same cabinet.

This configuration provides the advantages of both direct SCSI failover and direct to MSCP served failover.

6.2 HS*x* Failover Modes

The HSZ70, HSZ80, and the HSG80 implement two modes of failover operation when they are in a dual-redundant configuration, transparent failover mode and multibus failover mode. For the system to operate correctly, the HS*x* failover mode must be compatible with the configuration of the interconnect hardware and the host operating system software.

6.2.1 Transparent Failover Mode

In transparent failover mode, the HS*x* presents each logical unit on one port of the dual controller pair. Different logical units may be assigned to different ports, but an individual logical unit is accessible through one port at a time. When the HS*x* detects that a controller module has failed, it moves the logical unit to the corresponding port on the surviving controller.

The assumption in transparent mode is that the two ports are on the same host bus, so the logical unit can move from one port to the other without requiring any changes to the host's view of the device. The system manager must ensure that the bus configuration is correct for this mode of failover. Transparent failover has been supported by OpenVMS since V6.2.

To select transparent failover mode, enter the following command at the HS*x* console:

```
SET FAILOVER COPY=this_controller or other_controller
```

An example of the output of a console SHOW command on an HS*x* in transparent mode follows:

```
z70_A => sho this
Controller:
        HSZ70 ZG64100160 Firmware XB32-0, Hardware CX25
        Configured for dual-redundancy with ZG64100136
            In dual-redundant configuration
        Device Port SCSI address 7
        Time: 02-DEC-1998 09:22:09
Host port:
        SCSI target(s) (0, 2, 3, 4, 5, 6)
        Preferred target(s) (3, 5)
        TRANSFER_RATE_REQUESTED = 20MHZ
        Host Functionality Mode = A
        Allocation class           0
        Command Console LUN is target 0, lun 1
Cache:
        32 megabyte write cache, version 4
        Cache is GOOD
        Battery is GOOD
        No unflushed data in cache
        CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
        NOCACHE_UPS
```

6.2.2 Multibus Failover Mode

In multibus failover mode, the HS*x* makes each logical unit visible to the host on all ports of the dual controller pair. This allows the host to be aware of all the possible paths to the device. There are two advantages to having the host aware of all the paths to a device:

- The host can select an alternate path if it detects a failure on the current path. This is in addition to the failover that occurs when the HS*x* controller detects a failure, as is provided in transparent mode.

- The paths do no need to be on the same host bus. When the host is aware of the alternate paths, it can adjust its addressing methods appropriately to select a different path. This removes the SCSI bus as a single point of failure.

Note that although the logical unit is visible on all ports, it is online, and thus capable of doing I/O, on the ports of only one controller at a time. Different logical units may be online to different controllers, but an individual logical unit is online to one controller at a time.

You can determine which controller a logical unit is online to by entering the HS*x* console command, as follows:

```
z70_A => sho unit full
    LUN                                             Uses
--------------------------------------------------------------
  D200                                              DISK20300
        Switches:
          RUN                    NOWRITE_PROTECT        READ_CACHE
          MAXIMUM_CACHED_TRANSFER_SIZE = 32
          ACCESS_ID = ALL
        State:
          ONLINE to the other controller
          PREFERRED_PATH = OTHER_CONTROLLER
        Size: 2050860 blocks
```

The host executes I/O to a logical unit on one path at a time, until that path fails. If a controller has two ports, as the HSZ80 and the HSG80 controllers do, then different hosts can access the same logical unit over different ports of the controller to which the logical unit is online.

An HS*x* in multibus failover mode can only be used with the multipath functionality introduced in OpenVMS Version 7.2.

To select multibus failover mode, enter the following command at the HS*x*: console:

```
SET MULTIBUS_FAILOVER COPY=this_controller or other_controller
```

An example of the output of a console SHOW command on an HS*x* controller in multibus mode follows:

```
z70_B => sho this
Controller:
        HSZ70 ZG64100136 Firmware XB32-0, Hardware CX25
        Configured for MULTIBUS_FAILOVER with ZG64100160
            In dual-redundant configuration
        Device Port SCSI address 6
        Time: NOT SET
Host port:
        SCSI target(s) (0, 2, 3, 4, 5, 6)
        TRANSFER_RATE_REQUESTED = 20MHZ
        Host Functionality Mode = A
        Allocation class           0
        Command Console LUN is target 0, lun 1
Cache:
        32 megabyte write cache, version 4
        Cache is GOOD
        Battery is GOOD
        No unflushed data in cache
        CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
        NOCACHE_UPS
```

6.3 Path Selection by OpenVMS

In a multipath configuration, the first path to a device that OpenVMS configures is chosen as the initial current path. Note that this overrides the selection of a preferred path through HS*x* console commands such as the following:

```
SET UNIT PREFERRED_PATH=this_controller or other_controller
```

For this reason, it is not recommended that the OpenVMS system manager set preferred paths at the HS*x* console. Instead the manual path switching commands, described in Section 6.8.7 should be used.

In case of failover, OpenVMS chooses an alternate path according to the following order of precedence:

- The path specified by the system manager in a manual switch command
- A direct path where failover from one HS*x* controller module to another is not required
- A direct path, where HS*x* failover is required
- MSCP served path

OpenVMS avoids unnecessary failover from one HS*x* controller module to another because:

- Failover from one HS*x* controller module to another causes a delay of approximately 1 to 15 seconds, depending on the amount of cached data that needs to be synchronized.
- Other nodes that share access to the device must reestablish communication using an alternate path

6.3.1 How OpenVMS Performs Multipath Failover

When an I/O operation to a multipath disk fails and the failure suggests that a retry (on the current path or an alternate path) might succeed, then failover proceeds as follows:

- Mount verification is invoked.
- Mount verification attempts to communicate with the device on the current path and to validate that the volume is still correct.
- If mount verification fails on the current path, then a new path is sought according to the following order:
  - Direct paths that do not require an HS*x* failover are attempted a few times to maximize their chance of selection (the number of retries is controlled by the SYSGEN parameter MPDEV_LCRETRIES).
  - Any direct path is attempted.
  - Any path, including the served path is attempted.

  These steps execute in a loop, until a working path is located, or mount verification times out.
- A successful mount verification restarts all failed I/Os and allows new ones to proceed.

6.4 Configuration Requirements and Restrictions

The requirements for multipath SCSI configurations are presented in Table 6–1.

Table 6–1 Multipath SCSI Configuration Requirements

Component	Description
Alpha console firmware	For systems with HSZ70 and HSZ80, the minimum revision level is 5.3. For systems with HSG80, the minimum revision level is 5.4
Controller firmware	For HSZ70, the minimum revision level is 7.3; for HSZ80, it is 8.3; for HSG80, it is 8.4.
Controller module mode	Must be set to multibus mode. The selection is made at the HS*x* console.
Full connectivity	All hosts that are connected to an HS*x* in multibus mode must have a path to both HS*x* controller modules. This is because hosts that are connected exclusively to different controllers will switch the logical unit back and forth between controllers, preventing any I/O from executing. To prevent this from happening, always provide full connectivity from hosts to controller modules. If a host's connection to a controller fails, then take one of the following steps to avoid indefinite path switching: • Repair the connection promptly. • Prevent the other hosts from switching to the partially-connected controller. This is done by either disabling switching to the paths that lead to the partially-connected controller (see Section 6.8.8), or by shutting down the partially-connected controller. • Disconnect the partially-connected host from the both controllers.
Allocation classes	A valid HSZ allocation class is required (refer to Section 6.6.3). If a SCSI bus is configured with HSZ controllers only, and all the controllers have a valid HSZ allocation class, then it is not necessary to adhere to the older SCSI device naming rules. That is, the adapters do not require a matching port allocation class, a matching node allocation class, and matching OpenVMS device names. However, if there are non-HSZ devices on the bus, or HSZ controllers without an HSZ allocation class, then the standard rules for node and port allocation class assignments and controller device names for shared SCSI buses must be followed.

The restrictions for multipath SCSI configurations are presented in Table 6–2.

Table 6–2 Multipath SCSI Configuration Restrictions

Component	Description
Devices supported	DKDRIVER disk devices attached to HSZ70, HSZ80, and HSG80 controller modules are supported. Other device types, such as tapes, and class drivers, such as GKDRIVER, are not supported.
Mixed version and mixed architecture clusters	All hosts that are connected to an HSZ or HSG in multibus mode must be running OpenVMS Version 7.2 or higher. All Version 6.2 systems must have installed the cluster compatibility kit, as described in the *OpenVMS Version 7.2 Release Notes*.
Host based volume shadowing	In the initial release of OpenVMS Version 7.2, multipath devices can not be members of host-based shadow sets. This restriction will be removed by a Version 7.2 update kit.
SCSI to MSCP failover	In the initial release of OpenVMS Version 7.2, the MSCP path can not be included in the multipath set. This is accomplished by setting the SYSGEN parameter MPDEV_REMOTE = 0. This is the default setting in OpenVMS Version 7.2. This restriction will be removed by a Version 7.2 update kit.

6.5 Parallel SCSI Multipath Configurations

Prior to the introduction of multipath failover support, parallel SCSI configurations using HSZ70 storage controllers already provided transparent failover. Transparent failover is failover from one controller port to the corresponding port on the other controller module. Both ports must be on the same host bus.

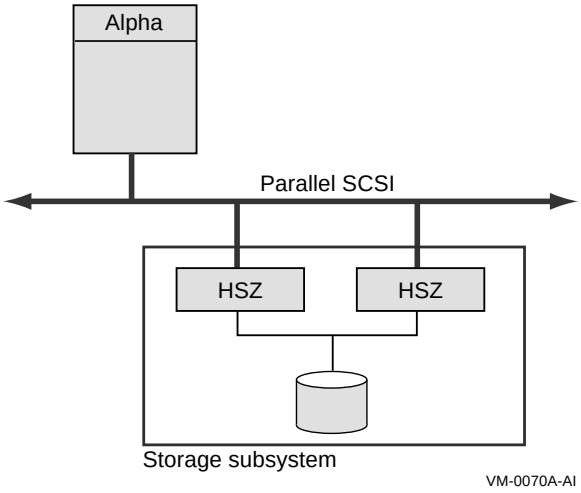
Multipath failover offers another level of failover, from one SCSI bus to another.

The figures in this section show systems configured for transparent failover, and for multipath failover. The special considerations for controller modules that have multiple ports, like the HSZ80 are also described.

6.5.1 Transparent Failover

Transparent failover in a parallel SCSI configuration, as shown in Figure 6–4, requires that both controller modules be on the same SCSI bus.

Figure 6–4 Parallel SCSI Configuration With Transparent Failover



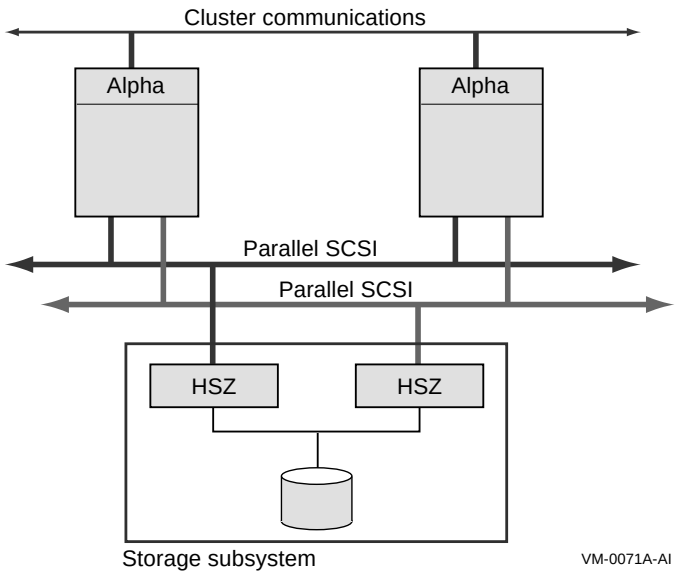
In this configuration:

- Each logical unit is visible to the host on only one controller module at a time. The other controller module does not answer at the same SCSI address, but it can be used for other SCSI addresses.
- One HSZ controller module detects the failure of the other controller and fails over the logical unit to itself. The surviving controller takes over the SCSI address or addresses of the failed controller.

6.5.2 Multibus Failover and Multiple Paths

A parallel SCSI configuration with multiple paths from the host to storage offers higher availability and performance than a configuration using transparent failover. Figure 6–5 shows this configuration.

Figure 6–5 Parallel SCSI Configuration With Multibus Failover and Multiple Paths



Note the following about this configuration:

- Each logical unit is visible to the host at the same ID on both controller modules so it can be configured. The logical unit responds to read/write I/O on only one controller at a time, the controller to which it is "online".
- The controller modules must be on different SCSI buses to prevent a bus ID conflict.
- The HSZ moves a logical unit to the other controller if either of the following events occurs:
  - HSZ detects a controller failure.
  - Host sends a SCSI START command for the logical unit to the other controller.

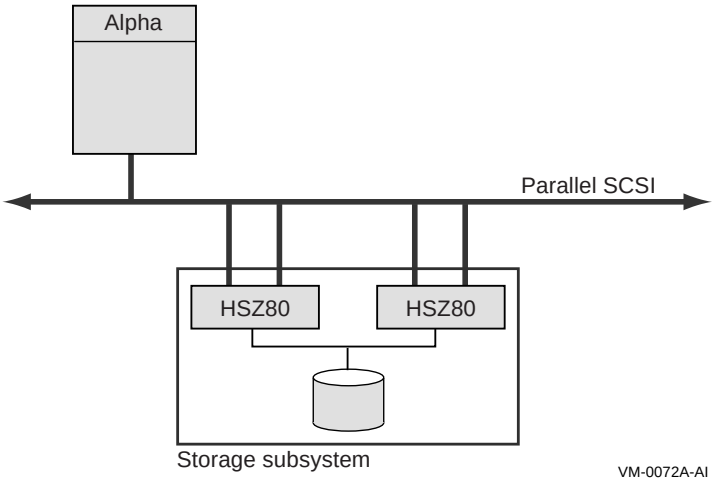
6.5.3 Configurations Using Multiported Storage Controllers

Higher levels of availability and performance can be achieved with the use of multiported storage controllers, such as the HSZ80. The HSZ80 storage controller is similar to the HSZ70 except that each HSZ80 controller has two ports.

This section shows three configurations that use multiported storage controllers. The configurations are presented in order of increasing availability.

Figure 6–6 shows a single host with a single interconnect, using an HSZ80 in transparent mode.

Figure 6–6 Multiported Parallel SCSI Configuration With Single Interconnect in Transparent Mode

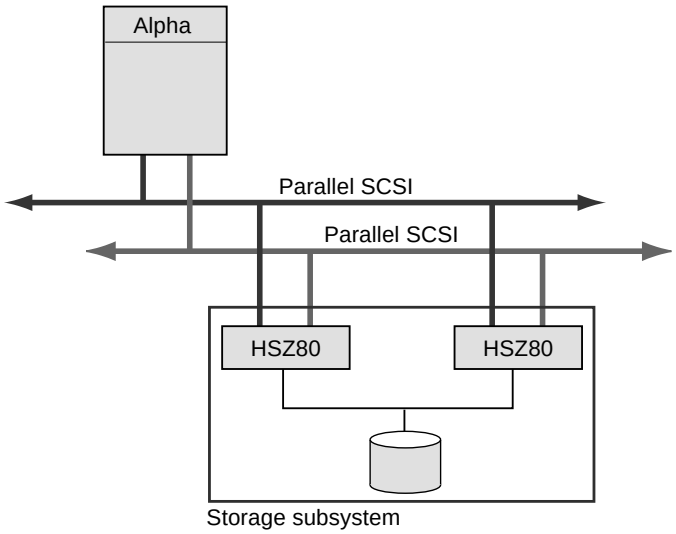


Note the following about this configuration:

- Each logical unit is visible on one port per storage controller.
- If a port fails, the HSZ80 fails over the traffic to the corresponding port of the other HSZ80.

Figure 6–7 shows a system configured in transparent mode using two paths from the host.

Figure 6–7 Multiported Parallel SCSI Configuration With Multiple Paths in Transparent Mode



In this configuration:

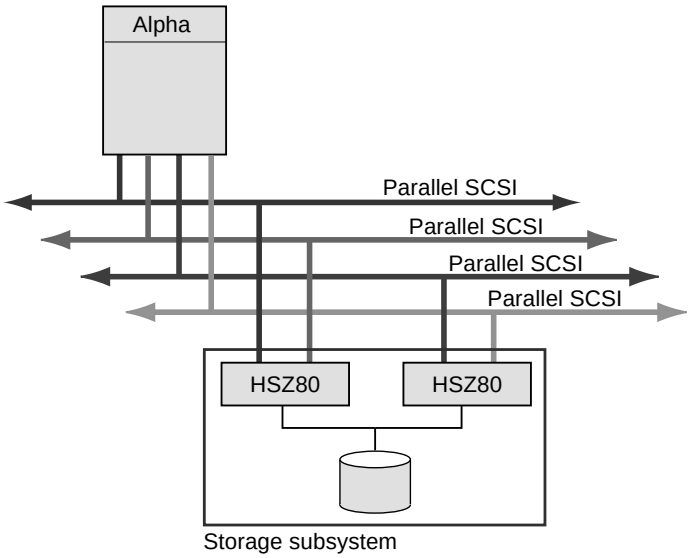
- Physically corresponding ports must be on the same SCSI bus.
- A maximum of two buses can be connected to each storage controller.

Note that in this configuration, although there are two buses, there is only one path from the host to a particular logical unit. When a controller fails, the logical unit moves to the corresponding port on the other controller. Both ports are on the same host bus.

This configuration has better performance than the one in Figure 6–6 because both SCSI buses can be simultaneously active. This ocnfiguration does not have higher availability, however, because there is still only one path from the host to the logical unit.

Figure 6–8 shows a system using the multiported HSZ80 storage controller configured in multibus mode.

Figure 6–8 Multiported Parallel SCSI Configuration With Multiple Paths in Multibus Mode



In this configuration:

- Each logical unit is visible to the host at the same ID on all ports (so they all will be configured by the host).
- All the ports must be on different SCSI buses.
- The host uses one path at a time.
- Each logical unit can execute I/O simultaneously over the two ports of the controller to which it is "on line." This means that if there are multiple hosts, then two paths to the storage device may be simultaneously active.

6.6 Device Naming for Parallel SCSI Multipath Configurations

SCSI device names have evolved as systems have become larger and more complex. At first, SCSI device names were entirely path dependent. The device name indicated the node, host adapter, SCSI bus ID, and logical unit number (LUN) used to access the device. Path-based names are not suitable for multiple host and multiple path environments because:

- The node name can not be used when there are multiple nodes with direct access to a device.
- The host adapter's controller letter can not be used when the controller letters on a shared bus do not match.
- The host adapter's controller letter can not be used when a node is connected to a device with multiple adapters.

The first two of these issues were addressed by the use of the node allocation class and the port allocation class. The third issue requires the introduction of an HSZ controller-based allocation class. These three allocation classes are reviewed in the following sections.

6.6.1 Review of Node Allocation Classes

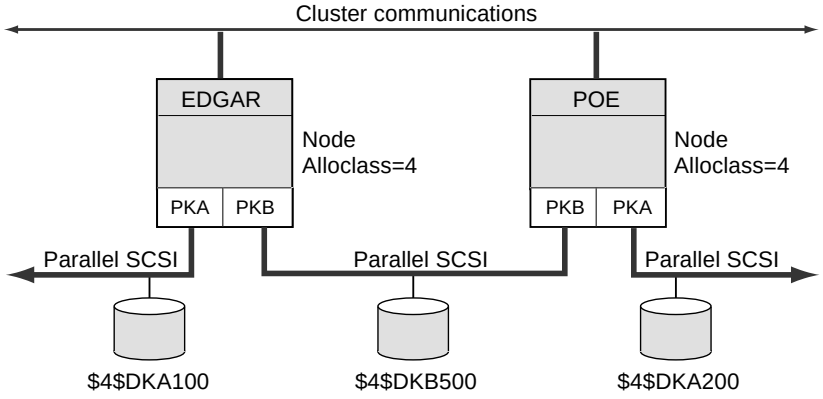
A node allocation class is used in a device name in place of a node name. A node allocation class is needed to produce a unique device name when multiple nodes have a direct connection to the same SCSI device.

A node allocation class can only be used in a device name when all nodes that share access to a SCSI storage device:

- Have only one direct path to the device.
- Use the same host controller name on the shared bus.
- Have sufficient SCSI IDs to produce unique names for nonshared devices.

Figure 6–9 shows a configuration whose devices are named using a node allocation class.

Figure 6–9 Devices Named Using a Node Allocation Class



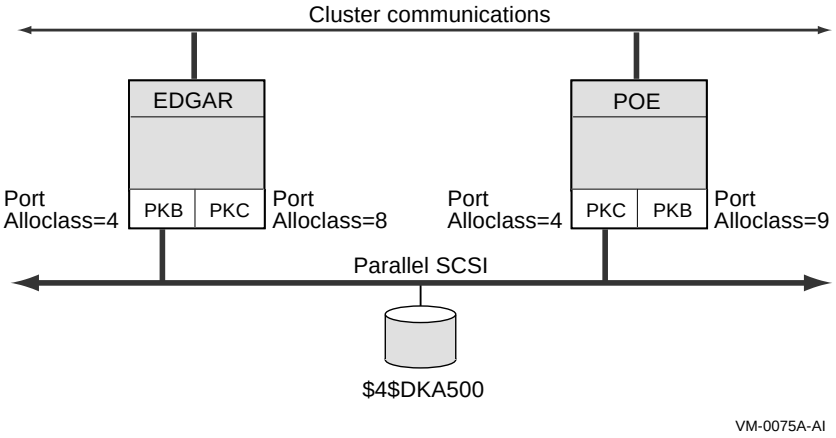
6.6.2 Review of Port Allocation Classes

A port allocation class in a device name designates the host adapter that is used to access the device. The port allocation class replaces the node allocation class in the device name, and the adapter controller letter is set to the constant A.

The port allocation class can be used when SCSI systems need more SCSI IDs to produce unique device names, or when the controller letter of the adapters on a shared bus do not match. A port allocation class can only be used in a device name when all nodes that share access to a SCSI storage device have only one direct path to the device.

Figure 6–10 shows a configuration whose devices are named using a port allocation class.

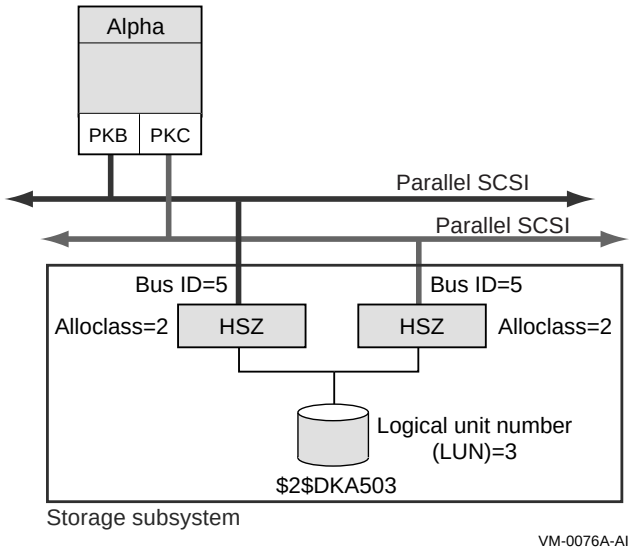
Figure 6–10 Devices Named Using a Port Allocation Class



6.6.3 Device Naming Using HSZ Allocation Classes

When any node has multiple buses connecting to the same storage device, the new HSZ allocation class shown in Figure 6–11 must be used.

Figure 6–11 Devices Named Using an HSZ Allocation Class



An HSZ allocation class is similar to the HSC, HSD, and HSJ allocation classes. The device name, using an HSZ allocation class number, takes the following form:

$*HSZ-allocation-class*$*ddcu*

where:

- *HSZ-allocation-class* is a decimal value from 1 to 999, assigned to a particular HSZ storage controller by the system manager
- *dd* represents the device class, which is DK for disk
- *c* represents the controller, which must be A when using an HSZ allocation class
- *u* represents the device unit number, which is determined by the SCSI bus ID and the logical unit number (LUN) of the device

The system manager sets an HSZ allocation class from the HSZ console, using the following command:

SET *this_controller or other_controller* ALLOCATION_CLASS = *n*

where *n* is a value from 1 to 999.

When the allocation class is set on one controller module in a dual redundant configuration, it is automatically set to the same value on the other controller.

In the following example, the allocation class is set to 199. The example shows that the value is set for both controllers.

```
z70_B => set this allo=199
z70_B => sho this
Controller:
        HSZ70 ZG64100136 Firmware XB32-0, Hardware CX25
        Configured for MULTIBUS_FAILOVER with ZG64100160
            In dual-redundant configuration
        Device Port SCSI address 6
        Time: NOT SET
Host port:
        SCSI target(s) (0, 2, 3, 4, 5, 6)

        TRANSFER_RATE_REQUESTED = 20MHZ
        Host Functionality Mode = A
        Allocation class         199
        Command Console LUN is target 0, lun 1
Cache:
        32 megabyte write cache, version 4
        Cache is GOOD
        Battery is GOOD
        No unflushed data in cache
        CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
        NOCACHE_UPS
z70_B => sho other
Controller:
        HSZ70 ZG64100160 Firmware XB32-0, Hardware CX25
        Configured for MULTIBUS_FAILOVER with ZG64100136
            In dual-redundant configuration
        Device Port SCSI address 7
        Time: NOT SET
Host port:
        SCSI target(s) (0, 2, 3, 4, 5, 6)

        TRANSFER_RATE_REQUESTED = 20MHZ
        Host Functionality Mode = A
        Allocation class         199
        Command Console LUN is target 0, lun 1
Cache:
        32 megabyte write cache, version 4
        Cache is GOOD
        Battery is GOOD
        No unflushed data in cache
        CACHE_FLUSH_TIMER = DEFAULT (10 seconds)
        NOCACHE_UPS
```

The following rules pertain to the use of an HSZ allocation class in SCSI device names:

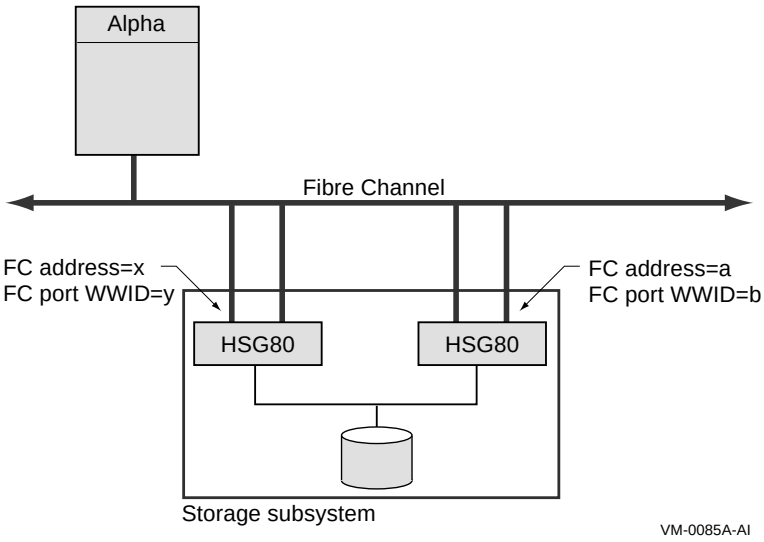
1. In multibus mode, an HSZ allocation class must be used in a device name (otherwise, the device is not configured).
2. In transparent mode, an HSZ allocation class can be used in a device name but it is not required.
3. The HSZ allocation class number must be the same for both controllers of an HSZ. This is handled automatically by the HSZ firmware.
4. The number must be unique among all types of allocation classes throughout the cluster. (Note that this requirement is specific to HSZ allocation classes; it does not apply to MSCP controller allocation classes.)
5. The HSZ allocation class must be specified when referring to devices that have an HSZ allocation class. For example, the names DKA500 and NODE10$DKA500 can not be used. In addition, the $GETDVI system service will only return the fully specified name, including the HSZ allocation class, for these devices.

6.7 Fibre Channel Multipath Configurations

The Fibre Channel storage controller (HSG80) provides the same configuration options in transparent mode as the parallel SCSI storage controller (HSZ80). That is, in transparent mode, failover can occur from one controller module to another provided they are on the same bus. Furthermore, dual ports on each storage controller increase the available failover options.

Figure 6–12 shows a single system with redundant paths to each HSG80 storage controller. The storage controllers are configured in transparent mode. Each HSG80 provides a standby port to which I/O can fail over.

Figure 6–12 Single Host With Two Dual-Ported Storage Controllers on a Single Bus



Note the following about this configuration:

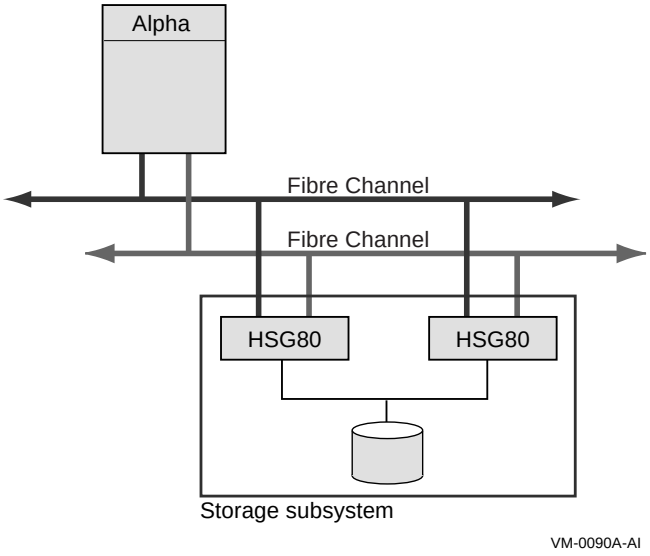
- Host has one adapter; thus it is connected to only one bus.
- Each HSG80 storage controller has two ports.
- Only one port per storage controller is active.
- Both storage controllers provide access to the same storage.

In this configuration, if one path fails, the standby port takes over the other port's Fibre Channel address and port worldwide ID (WWID).

Figure 6–13 shows a multipath configuration with the storage controllers configured in multibus mode. This means that each HSG80 port has its own Fibre Channel address and Fibre Channel port WWID.

This configuration is not possible using parallel SCSI as the interconnect because the SCSI address for a port is part of the device name. Because of this difference in naming, an additional configuration option is available with Fibre Channel—multiple ports on the same Fibre Channel.

Figure 6–13 Single Host With Two Dual-Ported Storage Controllers and Two Buses



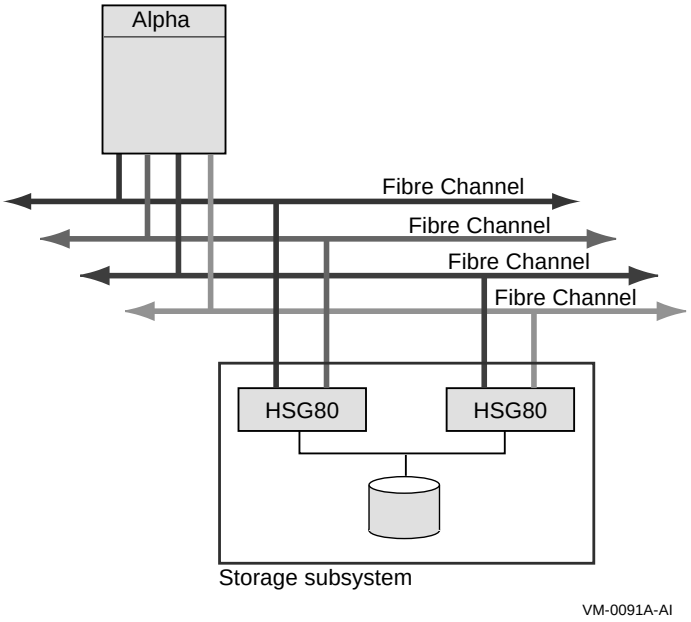
Note the following about this configuration:

- Host has two adapters, each attached to a different bus.
- Each port on each HSG80 storage controller is attached to a different interconnect.
- Both storage controllers can access the same disk.

Because of the second bus, this configuration provides a higher level of availability than the configuration in Figure 6–12.

Figure 6–14 shows another multipath configuration with its storage controllers configured in multibus mode.

Figure 6–14 Single Host With Two Dual-Ported Storage Controllers and Four Buses



Note the following about this configuration:

- Host has four adapters, each attached to a different interconnect.
- Each port on each HSG80 storage controller is attached to a different interconnect.
- Host has four paths to the same logical unit.

6.8 Implementing Multipath Configurations

Parallel SCSI and Fibre Channel interconnects support multipath configurations. Implementation of these configurations is similar, and the system parameters and the command for specifying paths are the same. The syntax for the path identifiers differs.

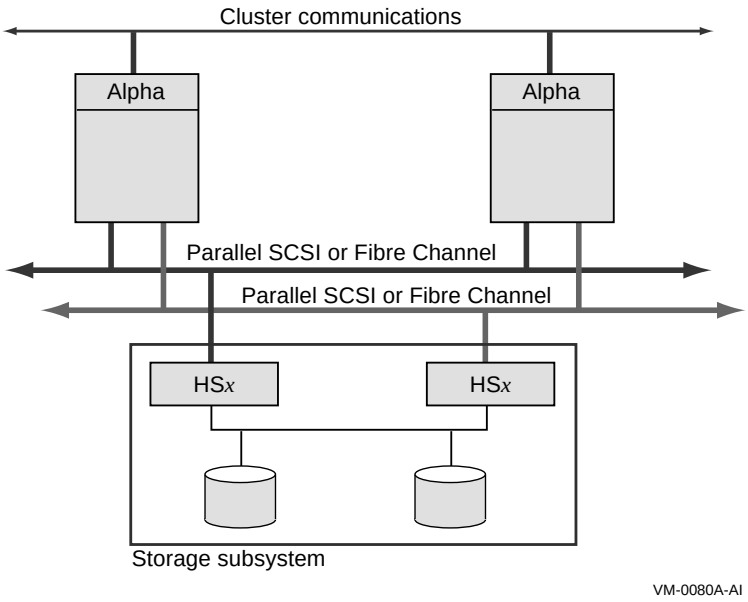
Implementing multiple paths to devices consists of the following steps:

1. Configuring a system or systems with multiple physical paths to those devices for which you want multipath support
2. Setting the HS*x* controller to multibus mode
3. Optionally qualifying multipath support by setting certain multipath system and console parameters, as appropriate for your configuration
4. Optionally tailoring the operation of multipath functionality, using the DCL command SET DEVICE/*qualifier*/PATH=*path-identifier*

6.8.1 Valid Multipath Configurations

Figure 6–15 shows a valid multipath, multihost configuration.

Figure 6–15 Two Hosts With Shared Buses and Shared Storage Controllers



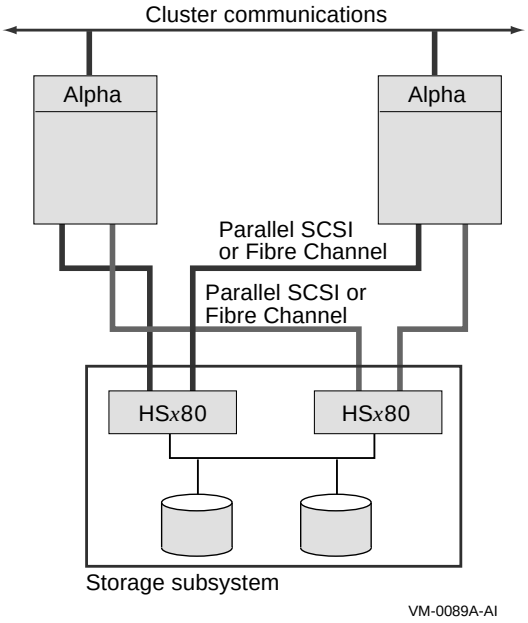
Note the following about this configuration:

- Each host has two adapters.
- Both hosts are connected to the same two buses.
- Both hosts share the storage.
- Each storage controller is connected to one bus only.
- The two storage controllers are connected to the same disks.

This configuration provides each host with two direct paths and one MSCP served path to each device.

Figure 6–16 shows a valid multipath configuration for systems that are not configured on the same bus.

Figure 6–16 Two Hosts With Shared, Multiported Storage Controllers



Note the following about this configuration:

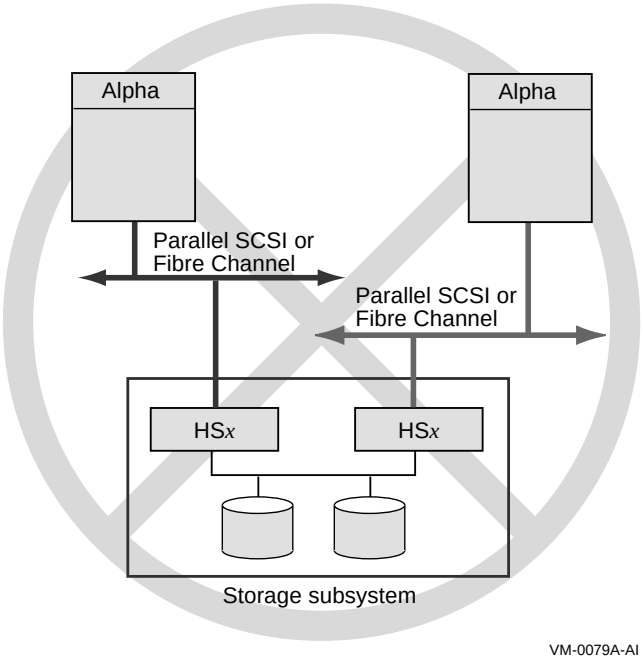
- Each host has two adapters.
- Each host is connected to two buses but the hosts do not share a bus.
- Both hosts share the storage.
- Each storage controller has two connections, one to each host.
- The two storage controllers are connected to the same disks.

This configuration provides each host with two direct paths, one to each storage controller, and one MSCP served path to each device.

6.8.2 Invalid Multipath Configuration

Figure 6–17 shows an invalid multipath configuration. The configuration is invalid because, if multiple hosts in a cluster are connected to an HSZ or HSG, they must all have connections to the same controller modules (see Table 6–1). In this configuration, each host is connected to a different controller module.

Figure 6–17 Invalid Multipath Configuration



6.8.3 Multipath System Parameters

Multipath support is enabled and qualified by the use of four system parameters, as described in Table 6–3. (A fifth multipath system parameter is reserved for the operating system.)

Table 6–3 Multipath System Parameters

Parameter	Description
MPDEV_ENABLE	Enables the formation of multipath sets when set to ON (1). If set to OFF (0), the formation of additional multipath sets is disabled. However, existing multipath sets remain in effect. The default is ON.
MPDEV_LCRETRIES	Controls the number of times the system retries direct paths to the controller that the logical unit is online to, before moving on to direct paths to the other controller, or to an MSCP served path to the device. The valid range for retries is 1 through 256. The default is 1.
MPDEV_POLLER	Enables polling of the paths to multipath set members when set to ON (1). Polling allows early detection of errors on inactive paths. If a path becomes unavailable or returns to service, the system manager is notified with an OPCOM message. If set to OFF (0), multipath polling is disabled. The default is ON.
MPDEV_REMOTE	Enables MSCP served disks to become members of a multipath set when set to ON (1). If set to OFF (0), only local paths to a SCSI device will be used in the formation of additional multipath sets. However, setting this parameter to OFF will not have any effect on existing multipath sets that have remote paths. The default is OFF.

Note

When OpenVMS Version 7.2 is released, the parameter MPDEV_REMOTE must remain set to the default value of OFF. This restriction will be removed by an update kit, shortly after the release of OpenVMS Version 7.2.

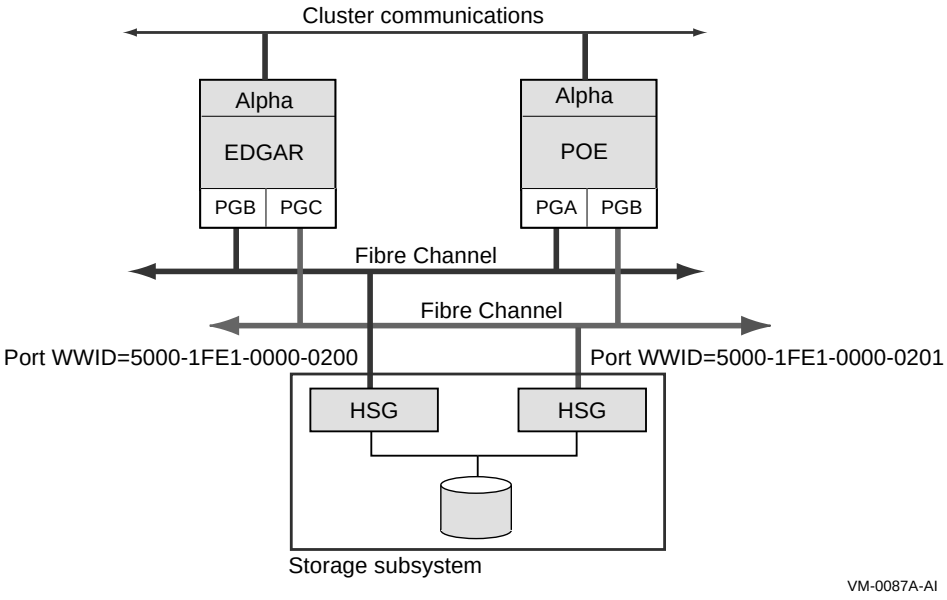
MPDEV_D1	Reserved for use by the operating system.

6.8.4 Path Identifiers

The system management commands described in the following sections allow you to monitor and control the operation of multipath failover. These commands provide a path identifier to uniquely specify each path in a multipath set.

Direct Fibre Channel paths are identified by the local host adapter name and the remote Fibre Channel port WWID — that is, the initiator and the target. For example, in Figure 6–18, the path identifier for the path from the host adapter on the left to the HSG storage controller on the left is PGB0.5000-1FE1-0000-0200. You can obtain the WWID for a storage controller from its console.

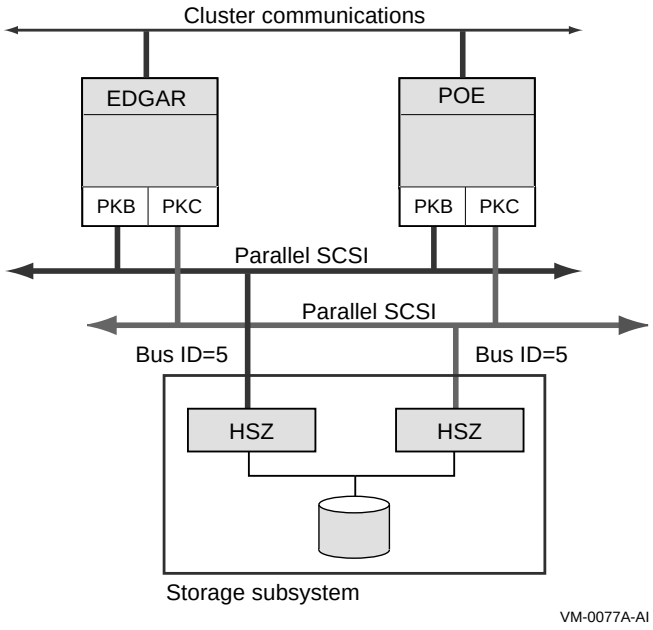
Figure 6–18 Fibre Channel Path Naming



Direct parallel SCSI paths are identified by the local host adapter name and the remote SCSI bus ID — that is, the initiator and the target. For example, in Figure 6–19, the path identifiers for node Edgar's two direct paths to the disk would be named PKB0.5 and PKC0.5.

The path identifier for MSCP served paths is MSCP.

Figure 6–19 Configuration With Multiple Direct Paths



6.8.5 Displaying Paths

When multipath support is enabled, you can display the multiple paths to a device using either of the following variants of the SHOW DEVICE DCL command:

```
SHOW DEVICE/FULL device-name
SHOW DEVICE/MULTIPATH_SET device-name
```

The SHOW DEVICE/FULL *device-name* command displays the traditional information about the device first and then lists all the paths to a device by their path identifiers (described in Section 6.8.4).

The SHOW DEVICE/MULTIPATH_SET *device-name* command lists only the multiple paths to a device.

6.8.5.1 Displaying Paths With SHOW DEVICE/FULL

The following example shows the output of a SHOW DEVICE/FULL *device-name* command. Note that the use of multiple paths is shown at the beginning of the display (`device has multiple I/O paths`), and the multiple path descriptions are shown toward the end of the display, beneath `I/O paths to device`. Note, too, that the values for `Error count` and `Operations completed` shown at the beginning of the display are the sums of these counts for each path.

```
$ SHOW DEVICE/FULL $70$DKA100:

Disk $70$DKA100: (FLAM10), device type DEC HSZ70, is online, mounted, file-
    oriented device, shareable, device has multiple I/O paths, served to cluster
    via MSCP Server, error logging is enabled.

    Error count                    2    Operations completed            123471
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                1    Default buffer size                512
    Total blocks             4109470    Sectors per track                   85
    Total cylinders             3022    Tracks per cylinder                 16
    Allocation class              70

    Volume label        "X6MV_SYS_1"    Relative volume number               0
    Cluster size                   9    Transaction count                    1
    Free blocks              2040228    Maximum files allowed           205524
    Extend quantity                5    Mount count                          4
    Mount status              System    Cache name        "_$600$DKA0:XQPCACHE"
    Extent cache size             64    Maximum blocks in extent cache   204022
    File ID cache size            64    Blocks currently in extent cache      0
    Quota cache size               0    Maximum buffers in FCP cache       2594
    Volume owner UIC           [1,1]    Vol Prot    S:RWCD,O:RWCD,G:RWCD,W:RWCD

  Volume Status:  ODS-2, subject to mount verification, file high-water marking,
      write-back caching enabled.
  Volume is also mounted on FIBRE3, COB2, FLAM10.

  I/O paths to device              2
  Path PKC0.1  (SISKO), primary path, current path.
    Error count                    1    Operations completed            121001
  Path PKB0.1  (SISKO).
    Error count                    1    Operations completed              2470
```

For each path of the multipath device, the path identifier, the host name associated with that path, the path status, the error count, and the operations count are displayed.

The terms that may appear in the multiple paths portion of the display are described in Table 6–4.

Table 6–4 SHOW DEVICE/FULL Multipath Terms

Term	Description
Primary path	This was the first path to the device found by the operating system.
Current path	This path is currently used for I/O.
User disabled	The DCL command SET DEVICE/NOENABLE has been executed for this path.
Polling disabled	The DCL command SET DEVICE/NOPOLL has been executed for this path.
Unavailable	The path is unavailable because the software driver has disconnected from the path.
MSCP host unavailable	The path is unavailable because the connection to the MSCP server on the remote host has been broken.

6.8.5.2 Displaying Paths With SHOW DEVICE/MULTIPATH_SET

You can obtain a brief listing of multiple paths for a specific device, for all the devices in an allocation class, or for all devices with the DCL command:

```
SHOW DEVICE/MULTIPATH_SET [device-name]
```

The device name is optional; when omitted, all devices that have formed multipath sets are shown. For each multipath device found, the device name, host name, device status, error count, number of accessible paths, total number of paths, and the current path's path identifier are displayed.

The host name displayed is the host name of the current path. For direct paths, this will be the local system's host name. For MSCP served paths, this will be the host name of the remote system which is serving access to the device.

The following example shows the output of a SHOW DEVICE /MULTIPATH command.

```
$ SHO DEV /MULTIPATH
Device                  Device           Error         Current
 Name                   Status           Count   Paths   path
$70$DKA0:     (FIBRE3)  Online               0   3/ 3   PKB0.0
$70$DKA100:   (FIBRE3)  MountVerify        11   2/ 3   PKB0.1
$70$DKA200:   (FIBRE3)  Mounted              0   3/ 3   PKC0.2
$70$DKA300:   (FIBRE3)  Mounted              2   1/ 3   PKC0.3
$70$DKA400:   (FIBRE3)  Mounted              4   3/ 3   PKC0.4
$70$DKA500:   (FIBRE3)  Mounted              7   3/ 3   PKC0.5
$70$DKA1400:  (FIBRE3)  Mounted              0   3/ 3   PKC0.14
$70$DKA1500:  (FIBRE3)  Mounted              4   3/ 3   PKB0.15
$100$DKA100:  (FIBRE3)  Online               0   2/ 2   PKA0.1
$100$DKA200:  (FIBRE3)  Mounted              3   2/ 2   PKA0.2
$100$DKA300:  (FIBRE3)  Online               0   2/ 2   PKA0.3
$100$DKA400:  (FIBRE3)  Online wrtlck        0   2/ 2   PKA0.4
```

If you choose to specify a partial device name, such as 70DKA, the display shows all devices with multiple paths whose names begin with 70DKA.

6.8.6 Path Polling

When SCSI multipath support is in effect, the system periodically polls all the I/O paths from each host adapter to each HSZ or HSG controller to determine the status of each I/O path. If the system detects any changes to a path, it outputs a message, similar to the following, to the console and to the operator's log:

```
All multipath devices on path PKB0.5 are either disabled or not reachable.
At least one multipath device on path PKB0.5 is enabled and reachable.
```

If all the devices on a path are removed, a path failure is reported. The path from the host to the HS*x* controller may still function, but this cannot be determined when there are no devices to poll.

You can turn polling on or off with the following command:

```
SET DEVICE device/[NO]POLL/PATH=path-identifier
```

Turning off polling for a path that will be out of service for a prolonged period is useful because it can reduce system overhead.

6.8.7 Switching Current Paths Manually

You can switch a device's current path manually using the SET DEVICE command with the /SWITCH qualifier. The most common reason for doing this is to balance the aggregate I/O load across multiple HS*x* controller modules and buses.

The command syntax for switching the current path is:

```
SET DEVICE device-name/SWITCH/PATH=path-identifier
```

The following command switches the path of device 2DKA502 to an MSCP served path.

```
$ SET DEVICE $2$DKA502/SWITCH/PATH=MSCP
```

6.8.8 Enabling or Disabling Paths as Path Switch Candidates

By default all paths are candidates for path switching. You can disable or re-enable a path as a switch candidate by using the SET DEVICE command with the /[NO]ENABLE qualifier. The reasons you might want to do this include the following:

- You know a specific path is broken, or that a failover to that path will cause some members of the cluster to lose access.
- To prevent automatic switching to a selected path while it is being serviced.

Note, the current path cannot be disabled.

The command syntax for enabling a disabled path is:

```
$ SET DEVICE device-name/[NO]ENABLE/PATH=path-identifier
```

The following command enables the MSCP served path of device 2DKA502.

```
$ SET DEVICE $2$DKA502/ENABLE/PATH=MSCP
```

The following command disables a local path of device 2DKA502.

```
$ SET DEVICE $2$DKA502/ENABLE/PATH=PKC0.5
```

Be careful when disabling paths. Avoid creating an invalid configuration, such as the one shown in Figure 6–17

6.8.9 Console Considerations

The console uses traditional, path-dependent, SCSI device names. For example, the device name format for disks is DK, followed by a letter indicating the host adapter, followed by the SCSI target ID, and the LUN.

This means that a multipath device will have multiple names, one for each host adapter it is accessible through. In the following sample output of a console show device command, the console device name is in the left column. The middle column and the right column provide additional information, specific to the device type.

Notice, for example, that the devices `dkb100` and `dkc100` are really two paths to the same device. The name `dkb100` is for the path through adapter PKB0, and the name `dkc100` is for the path through adapter PKC0. This can be determined by referring to the middle column, where the informational name includes the HSZ allocation class. The HSZ allocation class allows you to determine which console "devices" are really paths to the same HSZ device.

Note

The console may not recognize a change in the HSZ allocation class value until after you issue a console INIT command.

```
>>>sho dev
dkb0.0.0.12.0             $55$DKB0                      HSZ70CCL  XB26
dkb100.1.0.12.0           $55$DKB100                       HSZ70  XB26
dkb104.1.0.12.0           $55$DKB104                       HSZ70  XB26
dkb1300.13.0.12.0         $55$DKB1300                      HSZ70  XB26
dkb1307.13.0.12.0         $55$DKB1307                      HSZ70  XB26
dkb1400.14.0.12.0         $55$DKB1400                      HSZ70  XB26
dkb1500.15.0.12.0         $55$DKB1500                      HSZ70  XB26
dkb200.2.0.12.0           $55$DKB200                       HSZ70  XB26
dkb205.2.0.12.0           $55$DKB205                       HSZ70  XB26
dkb300.3.0.12.0           $55$DKB300                       HSZ70  XB26
dkb400.4.0.12.0           $55$DKB400                       HSZ70  XB26
dkc0.0.0.13.0             $55$DKC0                      HSZ70CCL  XB26
dkc100.1.0.13.0           $55$DKC100                       HSZ70  XB26
dkc104.1.0.13.0           $55$DKC104                       HSZ70  XB26
dkc1300.13.0.13.0         $55$DKC1300                      HSZ70  XB26
dkc1307.13.0.13.0         $55$DKC1307                      HSZ70  XB26
dkc1400.14.0.13.0         $55$DKC1400                      HSZ70  XB26
dkc1500.15.0.13.0         $55$DKC1500                      HSZ70  XB26
dkc200.2.0.13.0           $55$DKC200                       HSZ70  XB26
dkc205.2.0.13.0           $55$DKC205                       HSZ70  XB26
dkc300.3.0.13.0           $55$DKC300                       HSZ70  XB26
dkc400.4.0.13.0           $55$DKC400                       HSZ70  XB26
dva0.0.0.1000.0           DVA0
ewa0.0.0.11.0             EWA0             08-00-2B-E4-CF-0B
pka0.7.0.6.0              PKA0                 SCSI Bus ID 7
pkb0.7.0.12.0             PKB0                 SCSI Bus ID 7  5.54
pkc0.7.0.13.0             PKC0                 SCSI Bus ID 7  5.54
```

The console does not automatically attempt to use an alternate path to a device if I/O fails on the current path. For many console commands, however, it is possible to specify a list of devices that the console will attempt to access in order. In a multipath configuration, you can specify a list of console device names that correspond to the multiple paths of a device. For example, a boot command, such as the following, will cause the console to attempt to boot the multipath device

through the DKB100 path first, and if that fails, it will attempt to boot through the DKC100 path:

```
BOOT DKB100, DKC100
```

7

Configuring Fibre Channel as an OpenVMS Cluster Storage Interconnect

Note

To help you plan for the introduction of Fibre Channel in your computing environment, this documentation is provided in advance of the Fibre Channel functionality, which will be available shortly after the release of OpenVMS Version 7.2.

A major benefit of OpenVMS is its support of a wide range of interconnects and protocols for network configurations and for OpenVMS Cluster System configurations. This chapter describes OpenVMS Alpha support for Fibre Channel as a storage interconnect for single systems and as a shared storage interconnect for multihost OpenVMS Cluster systems.

The following topics are discussed:

- Overview of Fibre Channel (Section 7.1)
- Fibre Channel configuration requirements and restrictions (Section 7.2)
- Example configurations (Section 7.3)
- Fibre Channel addresses, WWIDs, and device names (Section 7.4)

For information about multipath support for Fibre Channel configurations, see Chapter 6.

Note that the Fibre Channel and parallel SCSI interconnects are shown generically in the figures in this chapter. Each is represented as a horizontal line to which the node and storage subsystems are connected. Physically, the Fibre Channel interconnect is always radially wired from a switch, as shown in Figure 7–1. Parallel SCSI can be radially wired to a hub or can be a daisy-chained bus.

The representation of multiple SCSI disks and SCSI buses in a storage subsystem is also simplified. The multiple disks and SCSI buses, which one or more HSZ*x* or HSG*x* controllers serve as a logical unit to a host, are shown in the figures as a single logical unit.

7.1 Overview of Fibre Channel

Fibre Channel is an ANSI standard network and storage interconnect that offers many advantages over other interconnects. Its most important features are described in Table 7–1.

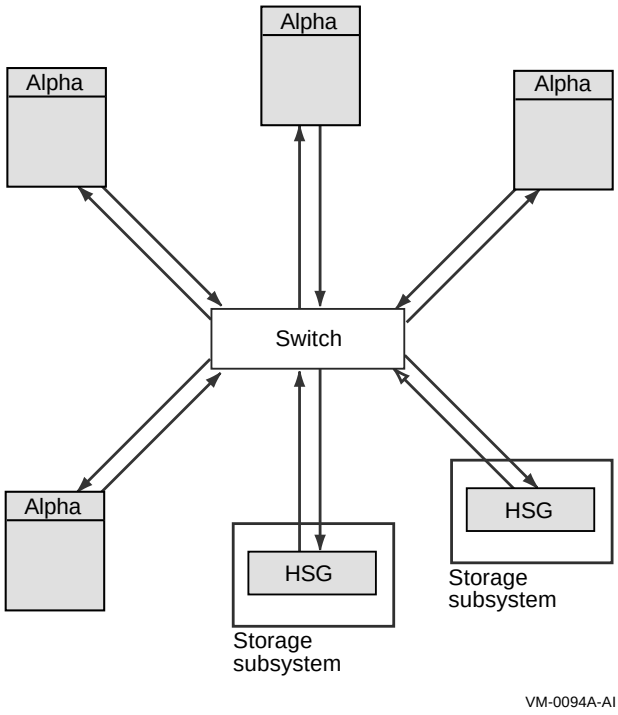
Table 7–1 Fibre Channel Features

Feature	Description
High-speed transmission	1.06 gigabits per second, full duplex, serial interconnect (can simultaneously transmit and receive 100 megabytes of data per second)
Choice of media	Initial OpenVMS support for fibre-optic media. Potential future support for copper media.
Long interconnect distances	Initial OpenVMS support for multimode fiber at 400 meters per link. Potential future support for 10 kilometer single-mode fiber links and 30 meter copper links.
Multiple protocols	Initial OpenVMS support for SCSI–3. Potential for IP, 802.3, HIPPI, ATM, IPI, and others in the future.
Numerous topologies	Initial OpenVMS support for switched FC (highly scalable, multiple concurrent communications). Potential future support for arbitrated loop (maximum number of nodes fixed at 126, shared bandwidth, hardware relatively inexpensive), fabric of switches, mixed arbitrated loop and switches, and point-to-point

The initial OpenVMS implementation supports a single-switch topology, with multimode fiber-optic media, at distances up to 400 meters per link.

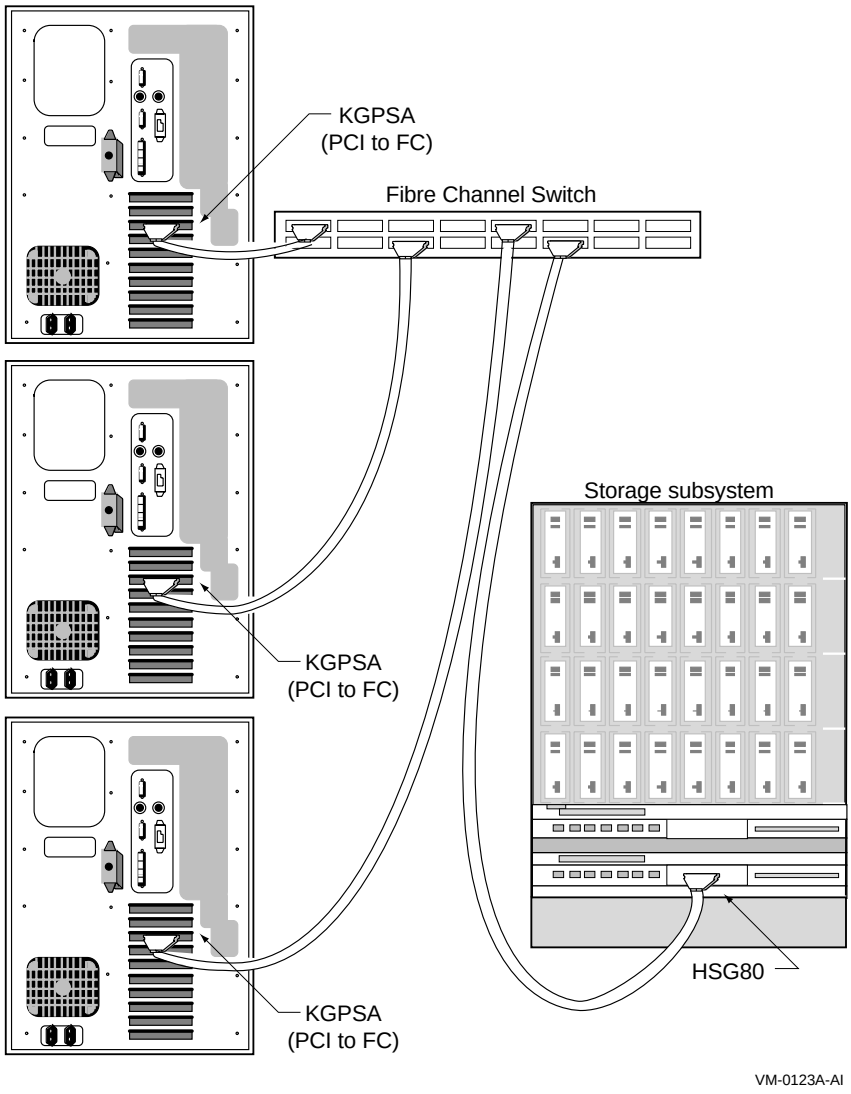
Figure 7–1 shows a logical view of a switched topology. The FC nodes are either Alpha hosts, or storage subsystems. Each link from a node to the switch is a dedicated FC connection. The switch provides store-and-forward packet delivery between pairs of nodes. Concurrent communication between disjoint pairs of nodes is supported by the switch.

Figure 7–1 Switched Topology, Logical View



A physical view of a Fibre Channel switched topology is shown in Figure 7–2.

Figure 7–2 Switched Topology, Physical View



7.2 Fibre Channel Configuration Requirements and Restrictions

OpenVMS Alpha supports the Fibre Channel devices presented in Table 7–2. Fibre Channel hardware names typically use the letter G to designate hardware that is specific to Fibre Channel.

Table 7–2 Fibre Channel Hardware Components

Component Name	Description	Minimum Revision
AlphaServer 8400, 8200, 4100, 1200[1]	Alpha host	OpenVMS Version 7.2, FW Rev. 5.4
HSG80	Fibre Channel controller module; 1 or 2 can be used in a Fibre Channel RAID storage cabinet	FW Rev. 8.4
KGPSA	OpenVMS Alpha PCI to Fibre Channel host adapter	FW Rev. TBD[2]
DSGGA	16-port, multimode, Fibre Channel switch	FW Rev. TBD[2]
BNGBX-*nn*	Multimode fiber-optic cable (*nn* denotes length in meters.)	n/a

[1]For a complete list, check the release notes in the update kit that enables Fibre Channel support and the OpenVMS Cluster SPD.

[2]For the FW revision level, check the release notes in the update kit that enables Fibre Channel support.

The following configurations are supported:

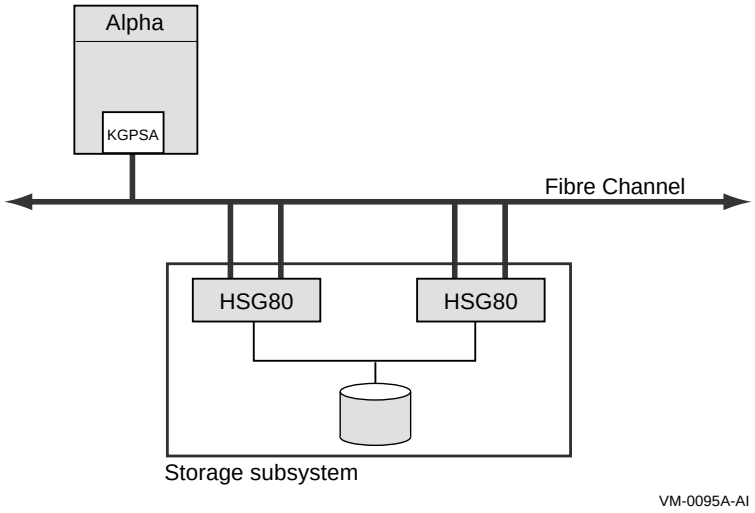
- Each link from the switch connects to at most one FC node. Connections from a switch to arbitrated loops, or to other switches, are not supported initially.
- Up to four Alpha hosts per switch.
- Up to two RAID storage cabinets per switch.
- Up to four adapters per host. Each adapter must be conected to a different switch.
- A storage cabinet may be connected to at most two switches.
- A storage cabinet may have multiple connections to the same switch.
- Multipath SCSI configurations are supported, as described in Chapter 6.
- Each host's ability to access FC storage may be restricted by the use of the zoning capability of the switch and by the use of host ID access lists in the HSG80. All hosts that are configured to have shared access to the same storage devices must be in the same OpenVMS Cluster. All nodes that are in the same cluster require a common cluster communication interconnect, such as a LAN, CI, or DSSI.
- Only disk devices may be connectd to the HSG80.
- FC disks support all of the OpenVMS disk features, such as system disks, dump disks, shadow set members, quorum disks, and MSCP served disks.
- Mixed version and mixed architecture clusters are allowed, provided that all systems with direct connection to the FC are running OpenVMS Alpha Version 7.2.
- The HSG80 must be in SCSI–3 mode, and each storage unit must be assigned a device identifier that is unique in the cluster, as described in Section 7.4.2.3.

Note that many of the configuration limits, such as four Alpha hosts with four adapters each and two RAID storage cabinets per switch, were determined by the testing that was possible before this first release, not by the hardware or the Fibre Channel standards. OpenVMS plans to increase these limits in future releases.

7.3 Example Configurations

Figure 7–3 shows a single system using Fibre Channel as a storage interconnect.

Figure 7–3 Single System With Dual-Ported Storage Controllers

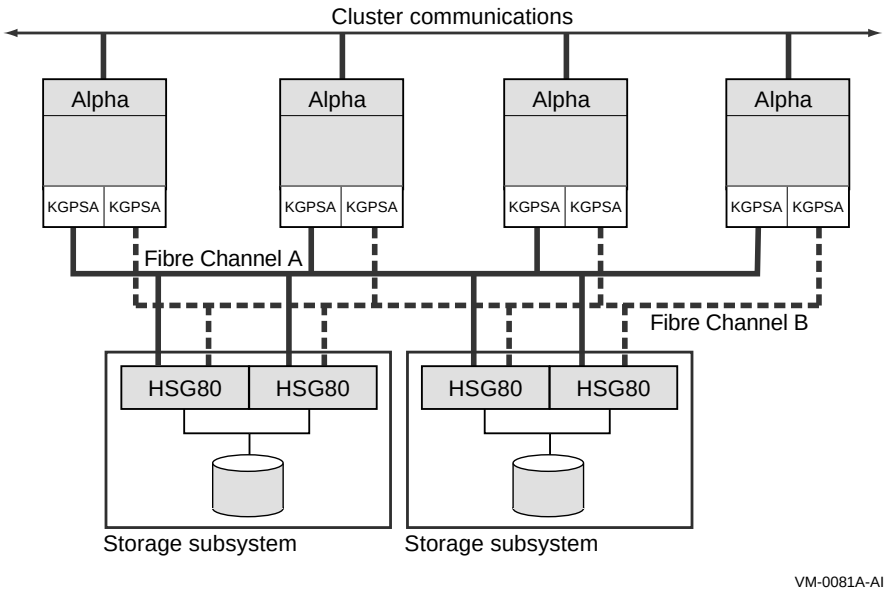


Note the following about this multipath configuration:

- The storage subsystem contains two storage controllers.
- Each storage controller has dual ports.
- Because of the two storage controllers, each with dual ports, there are four paths from the storage subsystem to the FC switch. These can be used to increase storage availability and performance.

Figure 7–4 shows a multihost configuration with two independent Fibre Channel interconnects connecting the hosts to the storage subsystems.

Figure 7–4 Multihost Fibre Channel Configuration



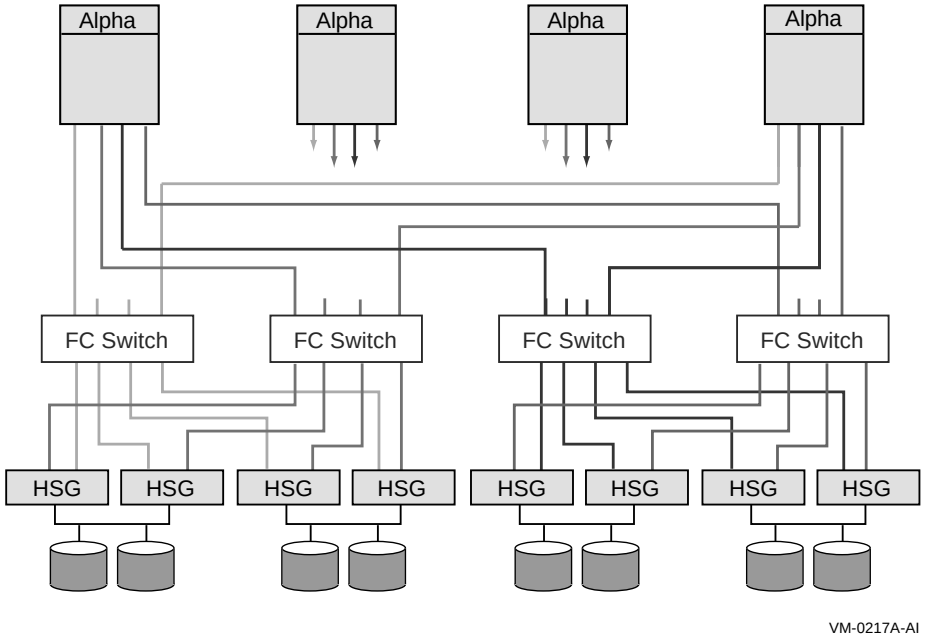
Note the following about this configuration:

- Two Fibre Channel interconnects, shown as Fibre Channel A and Fibre Channel B.
- Four Alpha hosts.
- Each host has two adapters.
- Each host is connected to both Fibre Channel interconnects.
- Each storage subsystem contains two HSG controller modules.
- Each controller module is connected to both Fibre Channel interconnects.
- Each host has two independent paths to each storage controller module.

The storage subsystems shown in Figure 7–4 are connected to two switches, which is the limit allowed in the initial Fibre Channel release. If additional host adapters and switches are desired, they must connect to additional RAID storage cabinets, as shown in Figure 7–5.

Figure 7–5 shows the largest configuration that is supported for the initial release of Fibre Channel.

Figure 7–5 Largest Initially Supported Configuration



Note the following about this configuration:

- Four hosts with four adapters each.
- Four Fibre Channel switches.
- Eight dual-ported controller modules, two per storage subsystem.
- Each host has two independent paths to each storage controller module.

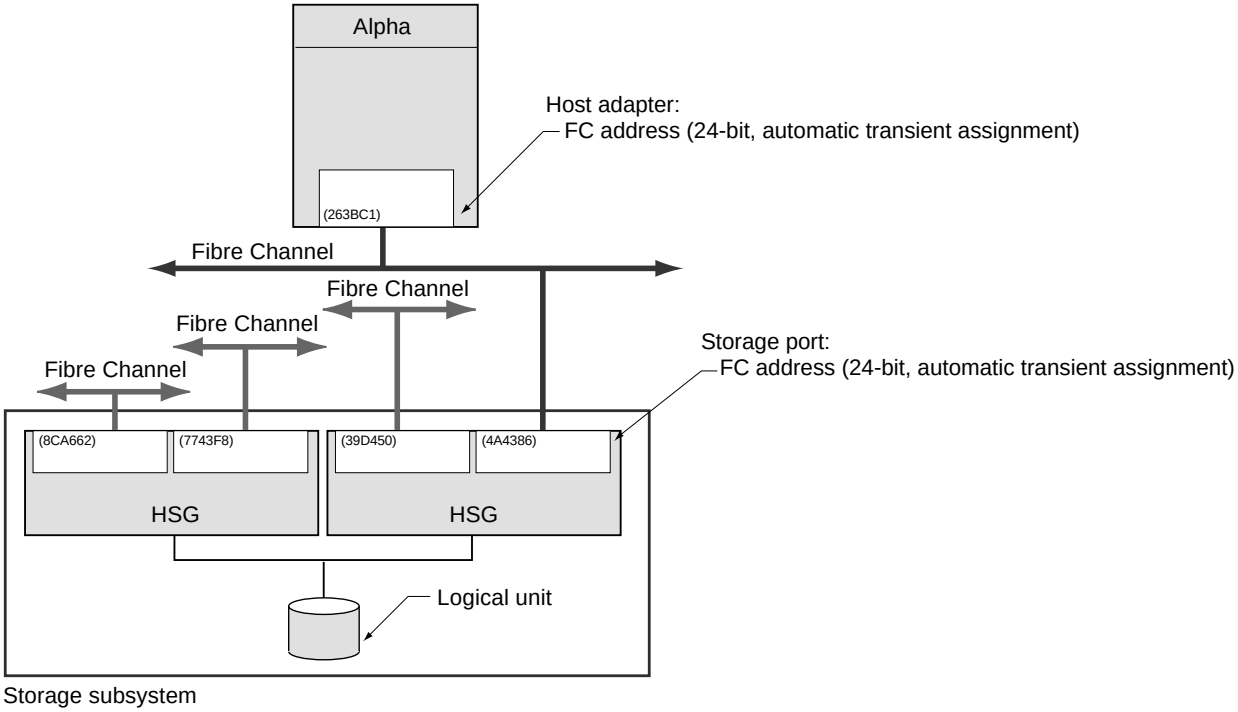
7.4 Fibre Channel Addresses, WWIDs, and Device Names

Fibre Channel devices come with factory-assigned worldwide IDs (WWIDs). These WWIDs are used by the system for automatic FC address assignment. The FC WWIDs and addresses also provide the means for the system manager to identify and locate devices in the FC configuration. The FC WWIDs and adresses are displayed, for example, by the Alpha console and by the HSG80 console. It is necessary, therefore, for the system manager to understand the meaning of these identifiers and how they relate to OpenVMS device names.

7.4.1 Fibre Channel Addresses and WWIDs

In most situations, Fibre Channel devices are configured to have temporary addresses. The device's address is assigned automatically each time the interconnect initializes. The device may receive a new address each time a Fibre Channel is reconfigured and reinitialized. This is done so that Fibre Channel devices do not require the use of address jumpers. There is one Fibre Channel address per port, as shown in Figure 7–6.

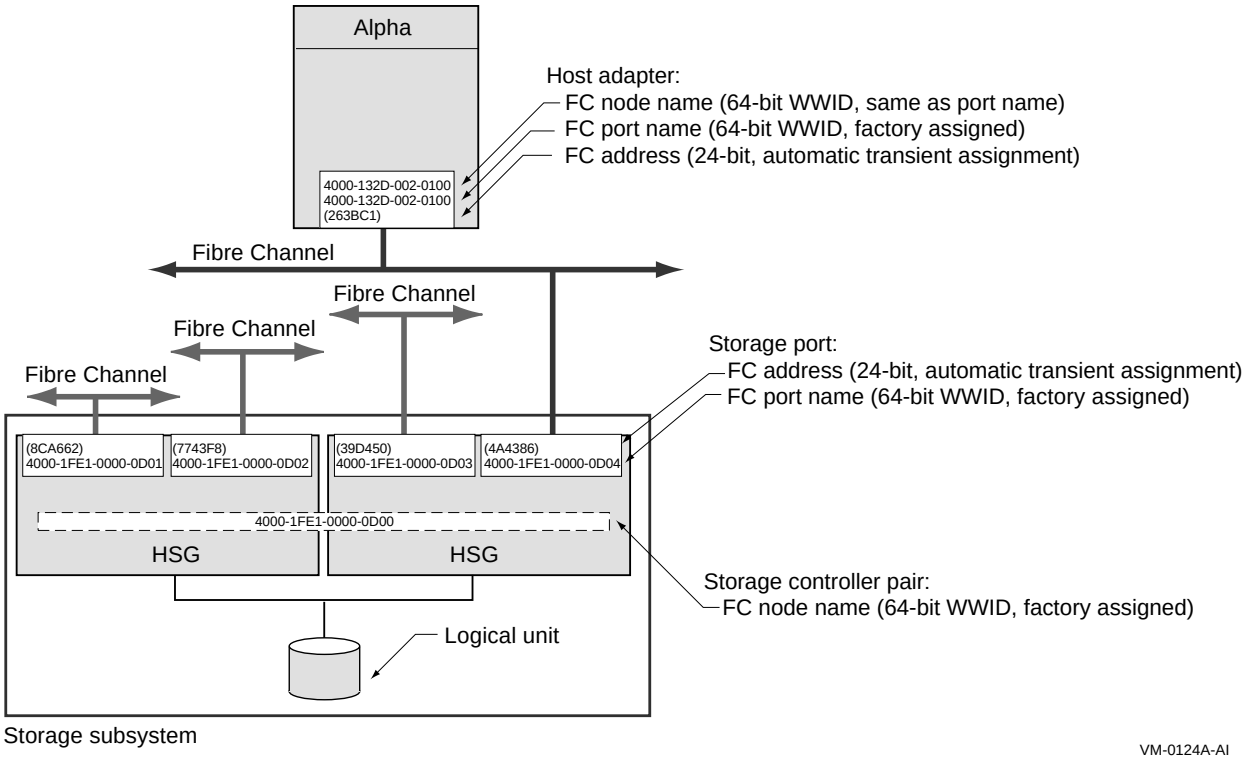
Figure 7–6 Fibre Channel Host and Port Addresses



In order to provide more permanent identification, each port on each device has a WWID, which is assigned at the factory. Every Fibre Channel WWID is unique. Fibre Channel also has node WWIDs to identify multiported devices. WWIDs are used by the system to detect and recover from automatic address changes. They are useful to system managers for identifying and locating physical devices.

Figure 7–7 shows Fibre Channel components with their factory-assigned WWIDs and their Fibre Channel addresses.

Figure 7–7 Fibre Channel Host and Port WWIDs and Addresses



Note the following about this figure:

- Node name and port name of the host adapter are identical.
- Host adapter's port and node name is a 64-bit, factory-assigned WWID.
- Host adapter's address is a 24-bit automatic, transient assignment.
- Each HSG storage port has a 64-bit, factory-assigned WWID, and a 24-bit transient address that is automatically assigned.
- HSG controller pair share a node name that is a 64-bit, factory-assigned WWID.

7.4.2 OpenVMS Names for Fibre Channel Devices

There is an OpenVMS name for each Fibre Channel storage adapter, for each path from the storage adapter to the storage subsystem, and for each storage device. These names are described in the following sections.

7.4.2.1 Fibre Channel Storage Adapter Names

Fibre Channel storage adapter names, which are automatically assigned by OpenVMS, take the form FG*x*0:

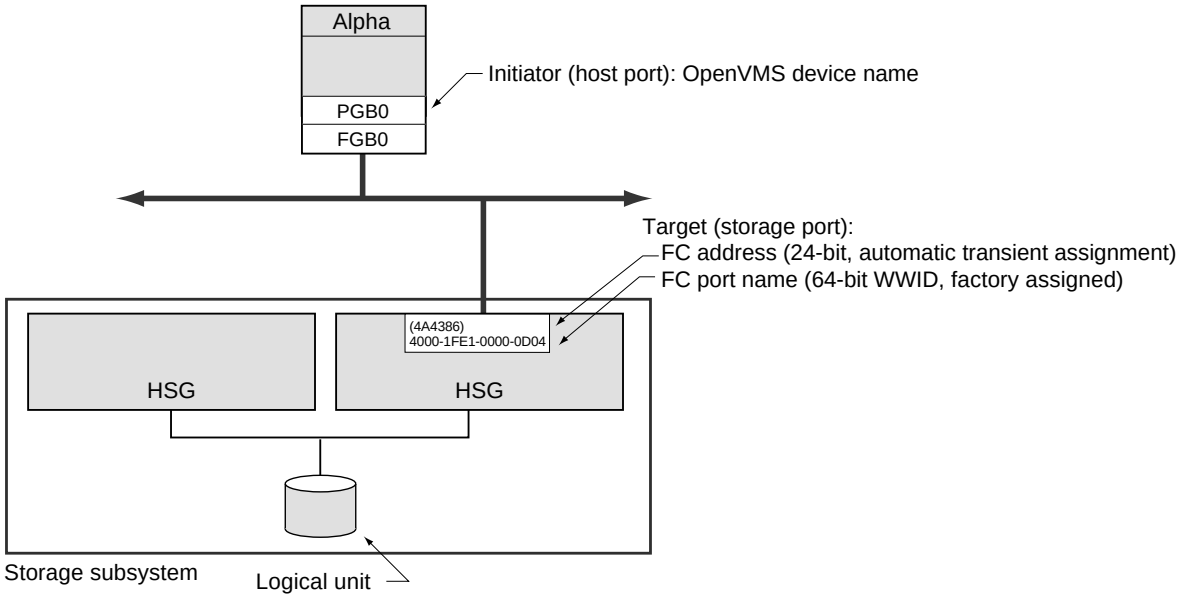
- FG represents Fibre Channel.
- *x* represents the unit letter, from A to Z.
- 0 is a constant.

The naming design places a limit of 26 adapters per system. (For the initial release, four adapters are supported per system.) This naming may be modified in future releases to support a larger number of adapters.

Fibre Channel adapters can run multiple protocols, such as SCSI and LAN. Each protocol is a pseudodevice associated with the adapter. For the initial implementation, just the SCSI protocol is supported. The SCSI pseudodevice name is PG*x*0, where *x* represents the same unit letter as the associated FG*x*0 adapter.

These names are illustrated in Figure 7–8.

Figure 7–8 Fibre Channel Initiator and Target Names



7.4.2.2 Fibre Channel Path Names

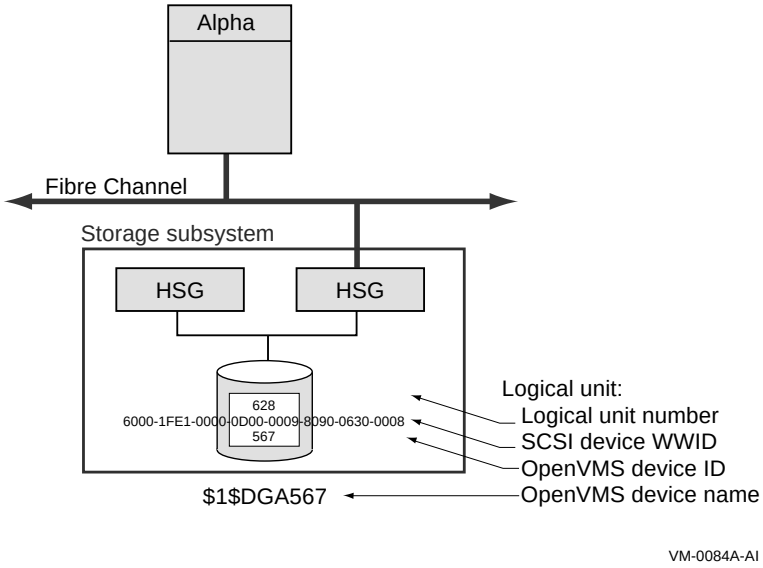
With the introduction of multipath SCSI support, as described in Chapter 6, it is necessary to identify specific paths from the host to the storage subsystem. This is done by concatenating the SCSI pseudodevice name, a decimal point (.), and the WWID of the storage subsystem port that is being accessed. For example, the Fibre Channel path shown in Figure 7–8 is named PGB0.4000-1FE1-0000-0D04.

Refer to Chapter 6 for more information on the display and use of the Fibre Channel path name.

7.4.2.3 Fibre Channel Storage Device Identification

The four identifiers associated with each FC storage device are shown in Figure 7–9.

Figure 7–9 Fibre Channel Storage Device Naming



The logical unit number (LUN) is used by the system as the address of a specific device within the storage subsystem. This number is set and displayed from the HSG80 console by the system manager. It can also be displayed by the OpenVMS SDA utility.

Each Fibre Channel storage device also has a WWID to provide permanent, unique identification of the device. The HSG80 device WWID is 128 bits. Half of this identifier is the WWID of the HSG80 that created the logical storage device, and the other half is specific to the logical device. The device WWID is displayed by the HSG80 console and the AlphaServer console.

Fibre Channel device WWIDs are uniquely assigned to the device, and they are path independent, so they could be used for OpenVMS device naming, if they were not so long. For example, the Fibre Channel storage unit WWID shown in Figure 7–9 is: `6000-1FE1-0000-0D00-0009-8090-0630-0008`.

To simplify the use of WWIDs in device names, OpenVMS enables the use of a shorter identifier, called an OpenVMS device ID. For the device with the WWID shown in Figure 7–9, a system manager might choose an OpenVMS device ID of 567.

An OpenVMS device ID for a Fibre Channel WWID has the following attributes:

- User assigned at the HSG console.
- User must ensure it is cluster unique.
- Moves with the device.
- Can be any decimal number from 0 to 32766.

A Fibre Channel storage device name is formed by the operating system from the constant `$1$DGA` and a device ID, *nnnnn*. The only variable part of the name is its device ID, which you assign at the HSG console. Figure 7–9 shows a storage device that is known to the host as `$1$DGA567`.

8

Configuring OpenVMS Clusters for Availability

Availability is the percentage of time that a computing system provides application service. By taking advantage of OpenVMS Cluster features, you can configure your OpenVMS Cluster system for various levels of availability, including disaster tolerance.

This chapter provides strategies and sample optimal configurations for building a highly available OpenVMS Cluster system. You can use these strategies and examples to help you make choices and tradeoffs that enable you to meet your availability requirements.

8.1 Availability Requirements

You can configure OpenVMS Cluster systems for different levels of availability, depending on your requirements. Most organizations fall into one of the broad (and sometimes overlapping) categories shown in Table 8–1.

Table 8–1 Availability Requirements

Availability Requirements	Description
Conventional	For business functions that can wait with little or no effect while a system or application is unavailable.
24 x 365	For business functions that require uninterrupted computing services, either during essential time periods or during most hours of the day throughout the year. Minimal down time is acceptable.
Disaster tolerant	For business functions with stringent availability requirements. These businesses need to be immune to disasters like earthquakes, floods, and power failures.

8.2 How OpenVMS Clusters Provide Availability

OpenVMS Cluster systems offer the following features that provide increased availability:

- A highly integrated environment that allows multiple systems to share access to resources
- Redundancy of major hardware components
- Software support for failover between hardware components
- Software products to support high availability

8.2.1 Shared Access to Storage

In an OpenVMS Cluster environment, users and applications on multiple systems can transparently share storage devices and files. When you shut down one system, users can continue to access shared files and devices. You can share storage devices in two ways:

- Direct access

  Connect disk and tape storage subsystems to CI and DSSI interconnects rather than to a node. This gives all nodes attached to the interconnect shared access to the storage system. The shutdown or failure of a system has no effect on the ability of other systems to access storage.

- Served access

  Storage devices attached to a node can be served to other nodes in the OpenVMS Cluster. MSCP and TMSCP server software enable you to make local devices available to all OpenVMS Cluster members. However, the shutdown or failure of the serving node affects the ability of other nodes to access storage.

8.2.2 Component Redundancy

OpenVMS Cluster systems allow for redundancy of many components, including:

- Systems
- Interconnects
- Adapters
- Storage devices and data

With redundant components, if one component fails, another is available to users and applications.

8.2.3 Failover Mechanisms

OpenVMS Cluster systems provide failover mechanisms that enable recovery from a failure in part of the OpenVMS Cluster. Table 8–2 lists these mechanisms and the levels of recovery that they provide.

Table 8–2 Failover Mechanisms

Mechanism	What Happens if a Failure Occurs	Type of Recovery
DECnet–Plus cluster alias	If a node fails, OpenVMS Cluster software automatically distributes new incoming connections among other participating nodes.	Manual. Users who were logged in to the failed node can reconnect to a remaining node.
		Automatic for appropriately coded applications. Such applications can reinstate a connection to the cluster alias node name, and the connection is directed to one of the remaining nodes.
I/O paths	With redundant paths to storage devices, if one path fails, OpenVMS Cluster software fails over to a working path, if one exists.	Transparent, provided another working path is available.

(continued on next page)

Table 8–2 (Cont.) Failover Mechanisms

Mechanism	What Happens if a Failure Occurs	Type of Recovery
Interconnect	With redundant or mixed interconnects, OpenVMS Cluster software uses the fastest working path to connect to other OpenVMS Cluster members. If an interconnect path fails, OpenVMS Cluster software fails over to a working path, if one exists.	Transparent.
Boot and disk servers	If you configure at least two nodes as boot and disk servers, satellites can continue to boot and use disks if one of the servers shuts down or fails. Failure of a boot server does not affect nodes that have already booted, providing they have an alternate path to access MSCP served disks.	Automatic.
Terminal servers and LAT software	Attach terminals and printers to terminal servers. If a node fails, the LAT software automatically connects to one of the remaining nodes. In addition, if a user process is disconnected from a LAT terminal session, when the user attempts to reconnect to a LAT session, LAT software can automatically reconnect the user to the disconnected session.	Manual. Terminal users who were logged in to the failed node must log in to a remaining node and restart the application.
Generic batch and print queues	You can set up generic queues to feed jobs to execution queues (where processing occurs) on more than one node. If one node fails, the generic queue can continue to submit jobs to execution queues on remaining nodes. In addition, batch jobs submitted using the /RESTART qualifier are automatically restarted on one of the remaining nodes.	Transparent for jobs waiting to be dispatched. Automatic or manual for jobs executing on the failed node.
Autostart batch and print queues	For maximum availability, you can set up execution queues as autostart queues with a failover list. When a node fails, an autostart execution queue and its jobs automatically fail over to the next logical node in the failover list and continue processing on another node. Autostart queues are especially useful for print queues directed to printers that are attached to terminal servers.	Transparent.

Reference: For more information about cluster alias, generic queues, and autostart queues, see *OpenVMS Cluster Systems*.

8.2.4 Related Software Products

Table 8–3 shows a variety of related OpenVMS Cluster software products that Compaq offers to increase availability.

Table 8–3 Products That Increase Availability

Product	Description
DECamds	Collects and analyzes data from multiple nodes simultaneously and directs all output to a centralized DECwindows display. The analysis detects availability problems and suggests corrective actions.
Volume Shadowing for OpenVMS	Makes any disk in an OpenVMS Cluster system a redundant twin of any other same-model disk in the OpenVMS Cluster.
DECevent	Simplifies disk monitoring. DECevent notifies you when it detects that a disk may fail. If the OpenVMS Cluster system is properly configured, DECevent can add a new disk and start a shadow copy operation.

(continued on next page)

Table 8–3 (Cont.) Products That Increase Availability

Product	Description
POLYCENTER Console Manager (PCM)	Helps monitor OpenVMS Cluster operations. PCM provides a central location for coordinating and managing up to 24 console lines connected to OpenVMS nodes or HSJ/HSC console ports.

8.3 Strategies for Configuring Highly Available OpenVMS Clusters

The hardware you choose and the way you configure it has a significant impact on the availability of your OpenVMS Cluster system. This section presents strategies for designing an OpenVMS Cluster configuration that promotes availability.

8.3.1 Availability Strategies

Table 8–4 lists strategies for configuring a highly available OpenVMS Cluster. These strategies are listed in order of importance, and many of them are illustrated in the sample optimal configurations shown in this chapter.

Table 8–4 Availability Strategies

Strategy	Description
Eliminate single points of failure	Make components redundant so that if one component fails, the other is available to take over.
Shadow system disks	The system disk is vital for node operation. Use Volume Shadowing for OpenVMS to make system disks redundant.
Shadow essential data disks	Use Volume Shadowing for OpenVMS to improve data availability by making data disks redundant.
Provide shared, direct access to storage	Where possible, give all nodes shared direct access to storage. This reduces dependency on MSCP server nodes for access to storage.
Minimize environmental risks	Take the following steps to minimize the risk of environmental problems: • Provide a generator or uninterruptible power system (UPS) to replace utility power for use during temporary outages. • Configure extra air-conditioning equipment so that failure of a single unit does not prevent use of the system equipment.
Configure at least three nodes	OpenVMS Cluster nodes require a quorum to continue operating. An optimal configuration uses a minimum of three nodes so that if one node becomes unavailable, the two remaining nodes maintain quorum and continue processing. **Reference:** For detailed information on quorum strategies, see Section 11.5 and *OpenVMS Cluster Systems*.
Configure extra capacity	For each component, configure at least one unit more than is necessary to handle capacity. Try to keep component use at 80% of capacity or less. For crucial components, keep resource use sufficiently *less* than 80% capacity so that if one component fails, the work load can be spread across remaining components without overloading them.

(continued on next page)

Table 8–4 (Cont.) Availability Strategies

Strategy	Description
Keep a spare component on standby	For each component, keep one or two spares available and ready to use if a component fails. Be sure to test spare components regularly to make sure they work. More than one or two spare components increases complexity as well as the chance that the spare will not operate correctly when needed.
Use homogeneous nodes	Configure nodes of similar size and performance to avoid capacity overloads in case of failover. If a large node fails, a smaller node may not be able to handle the transferred work load. The resulting bottleneck may decrease OpenVMS Cluster performance.
Use reliable hardware	Consider the probability of a hardware device failing. Check product descriptions for MTBF (mean time between failures). In general, newer technologies are more reliable.

8.4 Strategies for Maintaining Highly Available OpenVMS Clusters

Achieving high availability is an ongoing process. How you manage your OpenVMS Cluster system is just as important as how you configure it. This section presents strategies for maintaining availability in your OpenVMS Cluster configuration.

8.4.1 Strategies for Maintaining Availability

After you have set up your initial configuration, follow the strategies listed in Table 8–5 to maintain availability in OpenVMS Cluster system.

Table 8–5 Strategies for Maintaining Availability

Strategy	Description
Plan a failover strategy	OpenVMS Cluster systems provide software support for failover between hardware components. Be aware of what failover capabilities are available and which can be customized for your needs. Determine which components must recover from failure, and make sure that components are able to handle the additional work load that may result from a failover. **Reference:** Table 8–2 lists OpenVMS Cluster failover mechanisms and the levels of recovery that they provide.
Code distributed applications	Code applications to run simultaneously on multiple nodes in an OpenVMS Cluster system. If a node fails, the remaining members of the OpenVMS Cluster system are still available and continue to access the disks, tapes, printers, and other peripheral devices that they need.
Minimize change	Assess carefully the need for any hardware or software change before implementing it on a running node. If you must make a change, test it in a noncritical environment before applying it to your production environment.
Reduce size and complexity	After you have achieved redundancy, reduce the number of components and the complexity of the configuration. A simple configuration minimizes the potential for user and operator errors as well as hardware and software errors.

(continued on next page)

Table 8–5 (Cont.) Strategies for Maintaining Availability

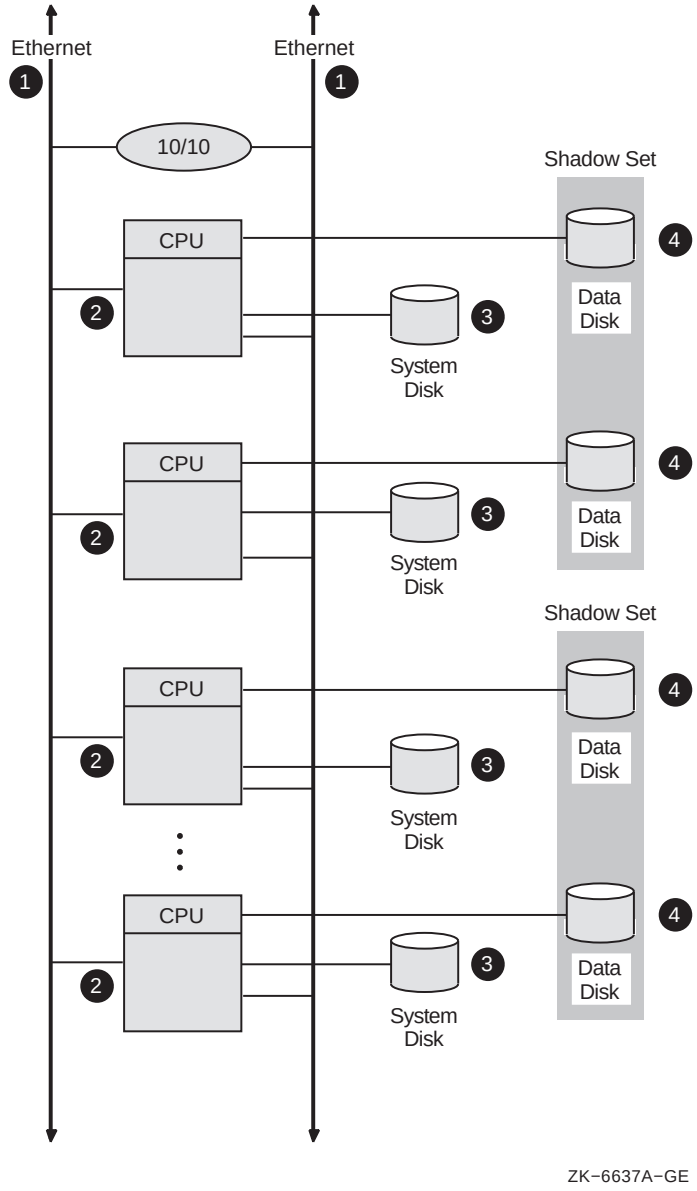
Strategy	Description
Set polling timers identically on all nodes	Certain system parameters control the polling timers used to maintain an OpenVMS Cluster system. Make sure these system parameter values are set identically on all OpenVMS Cluster member nodes. **Reference:** For information about these system parameters, see *OpenVMS Cluster Systems*.
Manage proactively	The more experience your system managers have, the better. Allow privileges for only those users or operators who need them. Design strict policies for managing and securing the OpenVMS Cluster system.
Use AUTOGEN proactively	With regular AUTOGEN feedback, you can analyze resource usage that may affect system parameter settings.
Reduce dependencies on a single server or disk	Distributing data across several systems and disks prevents one system or disk from being a single point of failure.
Implement a backup strategy	Performing frequent backup procedures on a regular basis guarantees the ability to recover data after failures. None of the strategies listed in this table can take the place of a solid backup strategy.

8.5 Availability in a LAN OpenVMS Cluster

Figure 8–1 shows an optimal configuration for a small-capacity, highly available LAN OpenVMS Cluster system. Figure 8–1 is followed by an analysis of the configuration that includes:

- Analysis of its components
- Advantages and disadvantages
- Key availability strategies implemented

Figure 8–1 LAN OpenVMS Cluster System



8.5.1 Components

The LAN OpenVMS Cluster configuration in Figure 8–1 has the following components:

Part	Description
1	Two Ethernet interconnects. For higher network capacity, use FDDI interconnects instead of Ethernet.
	Rationale: For redundancy, use at least two LAN interconnects and attach all nodes to all LAN interconnects.
	A single interconnect would introduce a single point of failure.
2	Three to eight Ethernet-capable OpenVMS nodes.
	Each node has its own system disk so that it is not dependent on another node.
	Rationale: Use at least three nodes to maintain quorum. Use fewer than eight nodes to avoid the complexity of managing eight system disks.
	Alternative 1: If you require satellite nodes, configure one or two nodes as boot servers. Note, however, that the availability of the satellite nodes is dependent on the availability of the server nodes.
	Alternative 2: For more than eight nodes, use a LAN OpenVMS Cluster configuration as described in Section 8.10.
3	System disks.
	System disks generally are not shadowed in LAN OpenVMS Clusters because of boot-order dependencies.
	Alternative 1: Shadow the system disk across two local controllers.
	Alternative 2: Shadow the system disk across two nodes. The second node mounts the disk as a nonsystem disk.
	Reference: See Section 11.2.4 for an explanation of boot-order and satellite dependencies.
4	Essential data disks.
	Use volume shadowing to create multiple copies of all essential data disks. Place shadow set members on at least two nodes to eliminate a single point of failure.

8.5.2 Advantages

This configuration offers the following advantages:

- Lowest cost of all the sample configurations shown in this chapter.
- Some potential for growth in size and performance.
- The LAN interconnect supports the widest choice of nodes.

8.5.3 Disadvantages

This configuration has the following disadvantages:

- No shared direct access to storage. The nodes are dependent on an MSCP server for access to shared storage.
- Shadowing disks across the LAN nodes causes shadow copies when the nodes boot.
- Shadowing the system disks is not practical because of boot-order dependencies.

8.5.4 Key Availability Strategies

The configuration in Figure 8–1 incorporates the following strategies, which are critical to its success:

- This configuration has no single point of failure.
- Volume shadowing provides multiple copies of essential data disks across separate nodes.
- At least three nodes are used for quorum, so the OpenVMS Cluster continues if any one node fails.
- Each node has its own system disk; there are no satellite dependencies.

8.6 Configuring Multiple LANs

Follow these guidelines to configure a highly available multiple LAN cluster:

- Bridge LAN segments together to form a single extended LAN.
- Provide redundant LAN segment bridges for failover support.
- Configure LAN bridges to pass the LAN and MOP multicast messages.

  Reference: Refer to the documentation for your LAN bridge and to the documentation for RBMS, DECelms, or POLYCENTER Framework for more information about configuring LAN bridges to pass these multicast messages.
- Use the Local Area OpenVMS Cluster Network Failure Analysis Program to monitor and maintain network availability. (See *OpenVMS Cluster Systems* for more information.)
- Use the troubleshooting suggestions in *OpenVMS Cluster Systems* to diagnose performance problems with the SCS layer and the NISCA transport protocol.
- Keep LAN average utilization below 50%.

Reference: See Section 10.7.7 for information about extended LANs (ELANs).

8.6.1 Selecting MOP Servers

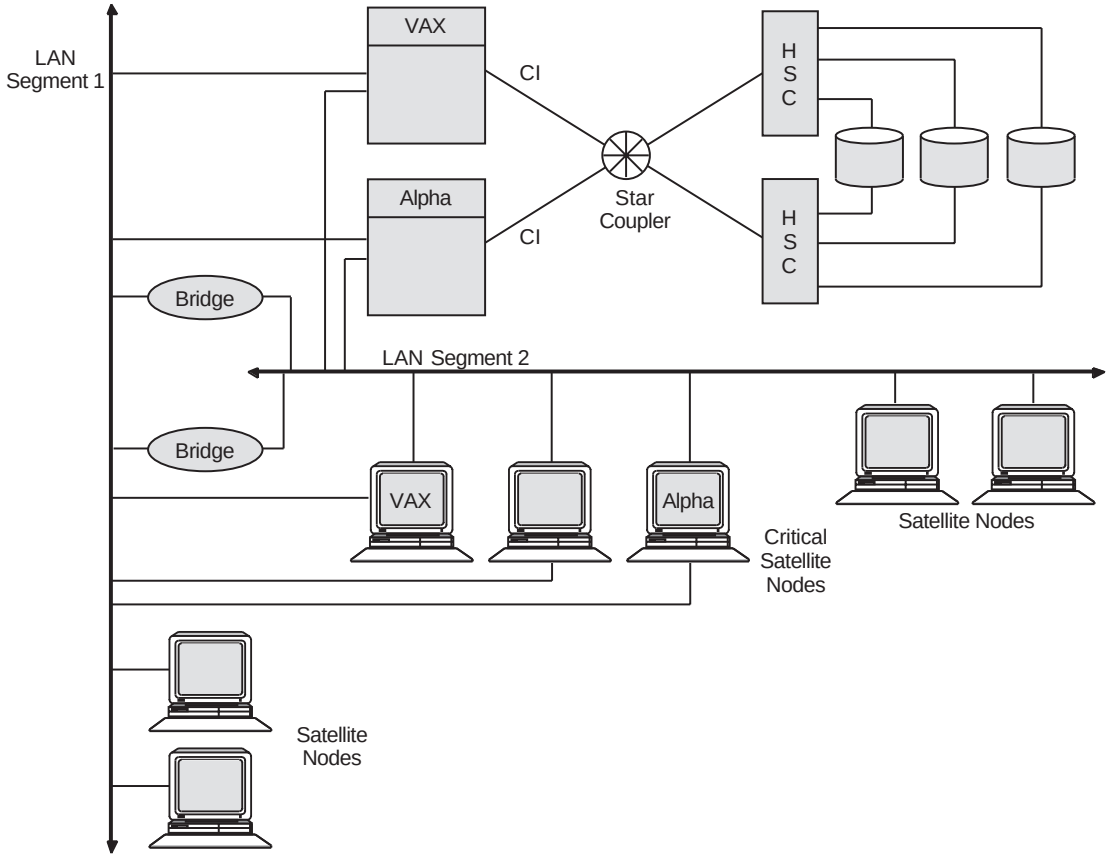
When using multiple LAN adapters with multiple LAN segments, distribute the connections to LAN segments that provide MOP service. The distribution allows MOP servers to downline load satellites even when network component failures occur.

It is important to ensure sufficient MOP servers for both VAX and Alpha nodes to provide downline load support for booting satellites. By careful selection of the LAN connection for each MOP server (Alpha or VAX, as appropriate) on the network, you can maintain MOP service in the face of network failures.

8.6.2 Configuring Two LAN Segments

Figure 8–2 shows a sample configuration for an OpenVMS Cluster system connected to two different LAN segments. The configuration includes Alpha and VAX nodes, satellites, and two bridges.

Figure 8–2 Two-LAN Segment OpenVMS Cluster Configuration



The figure illustrates the following points:

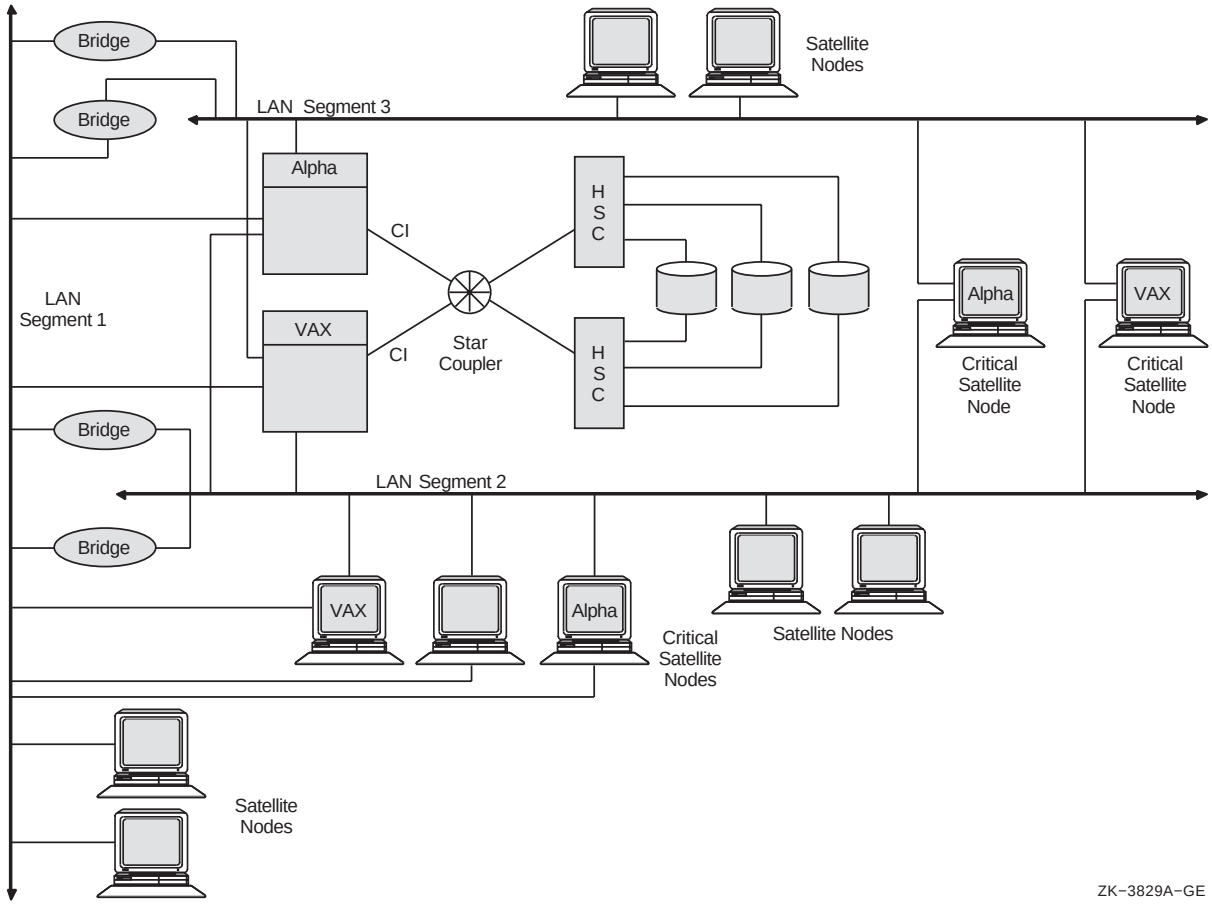
- Connecting critical nodes to multiple LAN segments provides increased availability in the event of segment or adapter failure. Disk and tape servers can use some of the network bandwidth provided by the additional network connection. Critical satellites can be booted using the other LAN adapter if one LAN adapter fails.
- Connecting noncritical satellites to only one LAN segment helps to balance the network load by distributing systems equally among the LAN segments. These systems communicate with satellites on the other LAN segment through one of the bridges.
- Only one LAN adapter per node can be used for DECnet and MOP service to prevent duplication of LAN addresses.
- LAN adapters providing MOP service (Alpha or VAX, as appropriate) should be distributed among the LAN segments to ensure that LAN failures do not prevent satellite booting.

- Using redundant LAN bridges prevents the bridge from being a single point of failure.

8.6.3 Configuring Three LAN Segments

Figure 8–3 shows a sample configuration for an OpenVMS Cluster system connected to three different LAN segments. The configuration also includes both Alpha and VAX nodes and satellites and multiple bridges.

Figure 8–3 Three-LAN Segment OpenVMS Cluster Configuration



The figure illustrates the following points:

- Connecting disk and tape servers to two or three LAN segments can help provide higher availability and better I/O throughput.
- Connecting critical satellites to two or more LAN segments can also increase availability. If any of the network components fails, these satellites can use the other LAN adapters to boot and still have access to the critical disk servers.
- Distributing noncritical satellites equally among the LAN segments can help balance the network load.

- A MOP server (Alpha or VAX, as appropriate) is provided for each LAN segment.

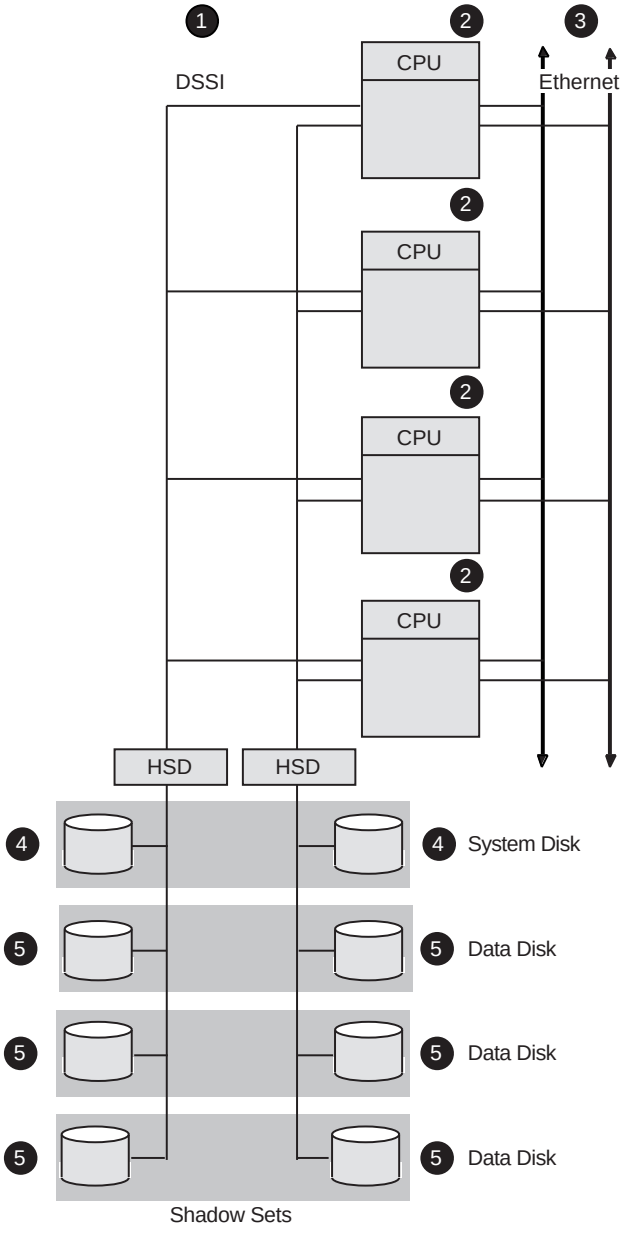
Reference: See Section 11.2.4 for more information about boot order and satellite dependencies in a LAN. See *OpenVMS Cluster Systems* for information about LAN bridge failover.

8.7 Availability in a DSSI OpenVMS Cluster

Figure 8–4 shows an optimal configuration for a medium-capacity, highly available DSSI OpenVMS Cluster system. Figure 8–4 is followed by an analysis of the configuration that includes:

- Analysis of its components
- Advantages and disadvantages
- Key availability strategies implemented

Figure 8–4 DSSI OpenVMS Cluster System



8.7.1 Components

The DSSI OpenVMS Cluster configuration in Figure 8–4 has the following components:

Part	Description
1	Two DSSI interconnects with two DSSI adapters per node.
	Rationale: For redundancy, use at least two interconnects and attach all nodes to all DSSI interconnects.
2	Two to four DSSI-capable OpenVMS nodes.
	Rationale: Three nodes are recommended to maintain quorum. A DSSI interconnect can support a maximum of four OpenVMS nodes.
	Alternative 1: Two-node configurations require a quorum disk to maintain quorum if a node fails.
	Alternative 2: For more than four nodes, configure two DSSI sets of nodes connected by two LAN interconnects.
3	Two Ethernet interconnects.
	Rationale: The LAN interconnect is required for DECnet–Plus communication. Use two interconnects for redundancy. For higher network capacity, use FDDI instead of Ethernet.
4	System disk.
	Shadow the system disk across DSSI interconnects.
	Rationale: Shadow the system disk across interconnects so that the disk and the interconnect do not become single points of failure.
5	Data disks.
	Shadow essential data disks across DSSI interconnects.
	Rationale: Shadow the data disk across interconnects so that the disk and the interconnect do not become single points of failure.

8.7.2 Advantages

The configuration in Figure 8–4 offers the following advantages:

- The DSSI interconnect gives all nodes shared, direct access to all storage.
- Moderate potential for growth in size and performance.
- There is only one system disk to manage.

8.7.3 Disadvantages

This configuration has the following disadvantages:

- Applications must be shut down in order to swap DSSI cables. This is referred to as "warm swap." The DSSI cable is warm swappable for the adapter, the cable, and the node.
- A node's location on the DSSI affects the recoverability of the node. If the adapter fails on a node located at the end of the DSSI interconnect, the OpenVMS Cluster may become unavailable.

8.7.4 Key Availability Strategies

The configuration in Figure 8–4 incorporates the following strategies, which are critical to its success:

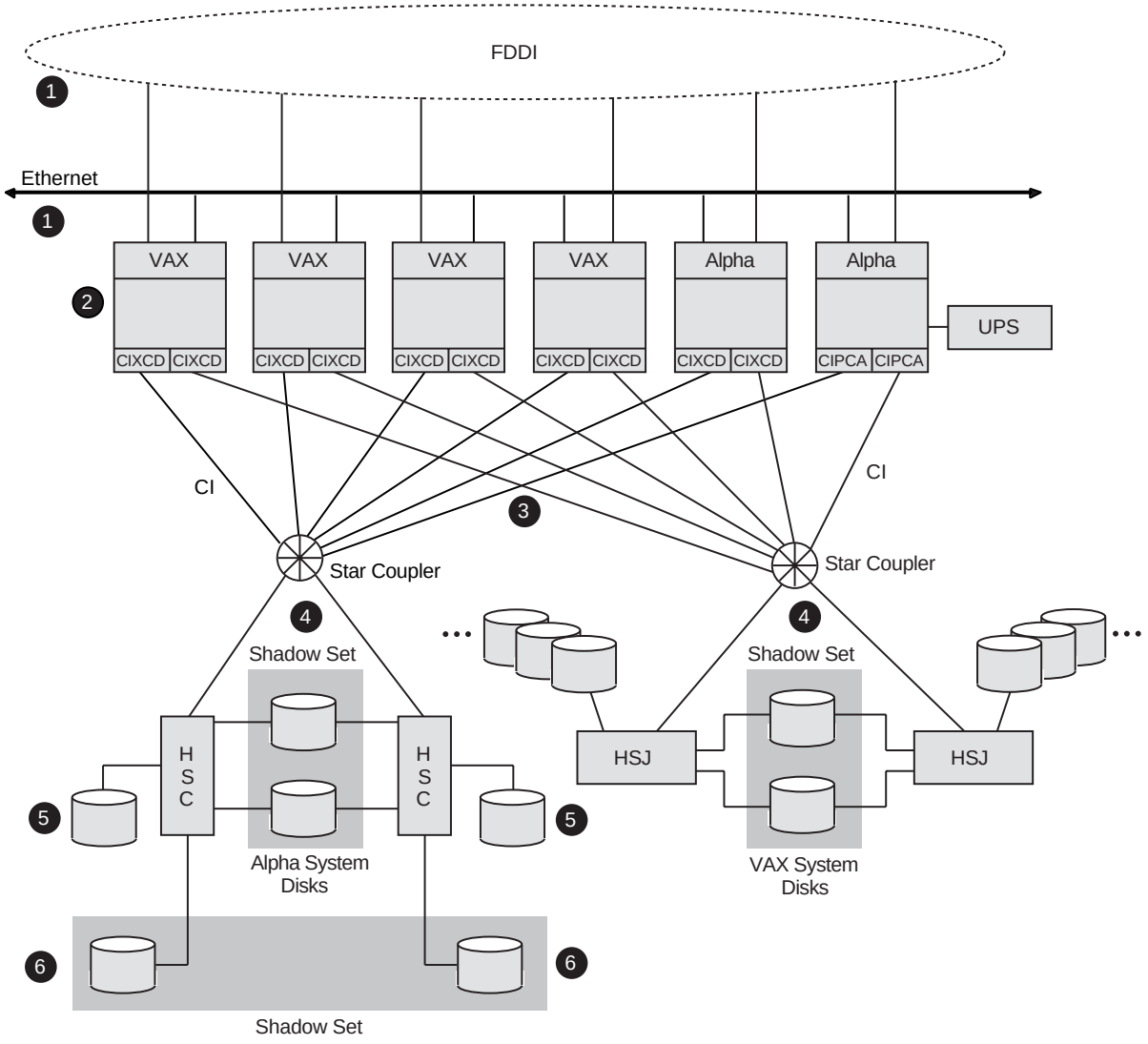
- This configuration has no single point of failure.
- Volume shadowing provides multiple copies of system and essential data disks across separate DSSI interconnects.
- All nodes have shared, direct access to all storage.
- At least three nodes are used for quorum, so the OpenVMS Cluster continues if any one node fails.
- There are no satellite dependencies.

8.8 Availability in a CI OpenVMS Cluster

Figure 8–5 shows an optimal configuration for a large-capacity, highly available CI OpenVMS Cluster system. Figure 8–5 is followed by an analysis of the configuration that includes:

- Analysis of its components
- Advantages and disadvantages
- Key availability strategies implemented

Figure 8–5 CI OpenVMS Cluster System



ZK–6973A–GE

8.8.1 Components

The CI OpenVMS Cluster configuration in Figure 8–5 has the following components:

Part	Description
1	Two LAN interconnects. **Rationale:** The additional use of LAN interconnects is required for DECnet–Plus communication. Having two LAN interconnects—Ethernet or FDDI—increases redundancy. For higher network capacity, use FDDI instead of Ethernet.

Part	Description
2	Two to 16 CI capable OpenVMS nodes. **Rationale:** Three nodes are recommended to maintain quorum. A CI interconnect can support a maximum of 16 OpenVMS nodes. **Reference:** For more extensive information about the CIPCA, see Appendix C. **Alternative:** Two-node configurations require a quorum disk to maintain quorum if a node fails.
3	Two CI interconnects with two star couplers. **Rationale:** Use two star couplers to allow for redundant connections to each node.
4	Critical disks are dual ported between CI storage controllers. **Rationale:** Connect each disk to two controllers for redundancy. Shadow and dual port system disks between CI storage controllers. Periodically alternate the primary path of dual-ported disks to test hardware.
5	Data disks. **Rationale:** Single port nonessential data disks, for which the redundancy provided by dual porting is unnecessary.
6	Essential data disks are shadowed across controllers. **Rationale:** Shadow essential disks and place shadow set members on different HSCs to eliminate a single point of failure.

8.8.2 Advantages

This configuration offers the following advantages:

- All nodes have direct access to all storage.
- This configuration has a high growth capacity for processing and storage.
- The CI is inherently dual pathed, unlike other interconnects.

8.8.3 Disadvantages

This configuration has the following disadvantage:

- Higher cost than the other configurations.

8.8.4 Key Availability Strategies

The configuration in Figure 8–5 incorporates the following strategies, which are critical to its success:

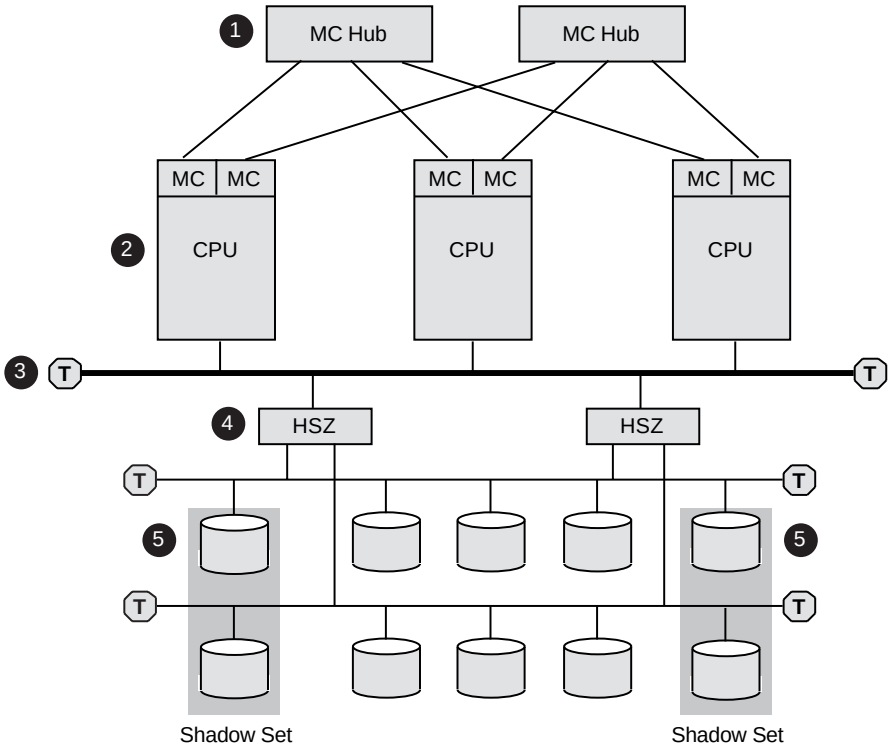
- This configuration has no single point of failure.
- Dual porting and volume shadowing provides multiple copies of essential disks across separate HSC or HSJ controllers.
- All nodes have shared, direct access to all storage.
- At least three nodes are used for quorum, so the OpenVMS Cluster continues if any one node fails.
- There are no satellite dependencies.
- The uninterruptible power supply (UPS) ensures availability in case of a power failure.

8.9 Availability in a MEMORY CHANNEL OpenVMS Cluster

Figure 8–6 shows a highly available MEMORY CHANNEL (MC) cluster configuration. Figure 8–6 is followed by an analysis of the configuration that includes:

- Analysis of its components
- Advantages and disadvantages
- Key availability strategies implemented

Figure 8–6 MEMORY CHANNEL Cluster



ZK–8709A–GE

8.9.1 Components

The MEMORY CHANNEL configuration shown in Figure 8–6 has the following components:

Part	Description
1	Two MEMORY CHANNEL hubs. **Rationale:** Having two hubs and multiple connections to the nodes prevents having a single point of failure.

Part	Description
2	Three to eight MEMORY CHANNEL nodes. **Rationale:** Three nodes are recommended to maintain quorum. A MEMORY CHANNEL interconnect can support a maximum of eight OpenVMS Alpha nodes. **Alternative:** Two-node configurations require a quorum disk to maintain quorum if a node fails.
3	Fast-wide differential (FWD) SCSI bus. **Rationale:** Use a FWD SCSI bus to enhance data transfer rates (20 million transfers per second) and because it supports up to two HSZ controllers.
4	Two HSZ controllers. **Rationale:** Two HSZ controllers ensure redundancy in case one of the controllers fails. With two controllers, you can connect two single-ended SCSI buses and more storage.
5	Essential system disks and data disks. **Rationale:** Shadow essential disks and place shadow set members on different SCSI buses to eliminate a single point of failure.

8.9.2 Advantages

This configuration offers the following advantages:

- All nodes have direct access to all storage.
- SCSI storage provides low-cost, commodity hardware with good performance.
- The MEMORY CHANNEL interconnect provides high-performance, node-to-node communication at a low price. The SCSI interconnect complements MEMORY CHANNEL by providing low-cost, commodity storage communication.

8.9.3 Disadvantages

This configuration has the following disadvantage:

- The fast-wide differential SCSI bus is a single point of failure. One solution is to add a second, fast-wide differential SCSI bus so that if one fails, the nodes can fail over to the other. To use this functionality, the systems must be running OpenVMS Version 7.2 or higher and have multipath support enabled.

8.9.4 Key Availability Strategies

The configuration in Figure 8–6 incorporates the following strategies, which are critical to its success:

- Redundant MEMORY CHANNEL hubs and HSZ controllers prevent a single point of hub or controller failure.
- Volume shadowing provides multiple copies of essential disks across separate HSZ controllers.
- All nodes have shared, direct access to all storage.
- At least three nodes are used for quorum, so the OpenVMS Cluster continues if any one node fails.

8.10 Availability in an OpenVMS Cluster with Satellites

Satellites are systems that do not have direct access to a system disk and other OpenVMS Cluster storage. Satellites are usually workstations, but they can be any OpenVMS Cluster node that is served storage by other nodes in the cluster.

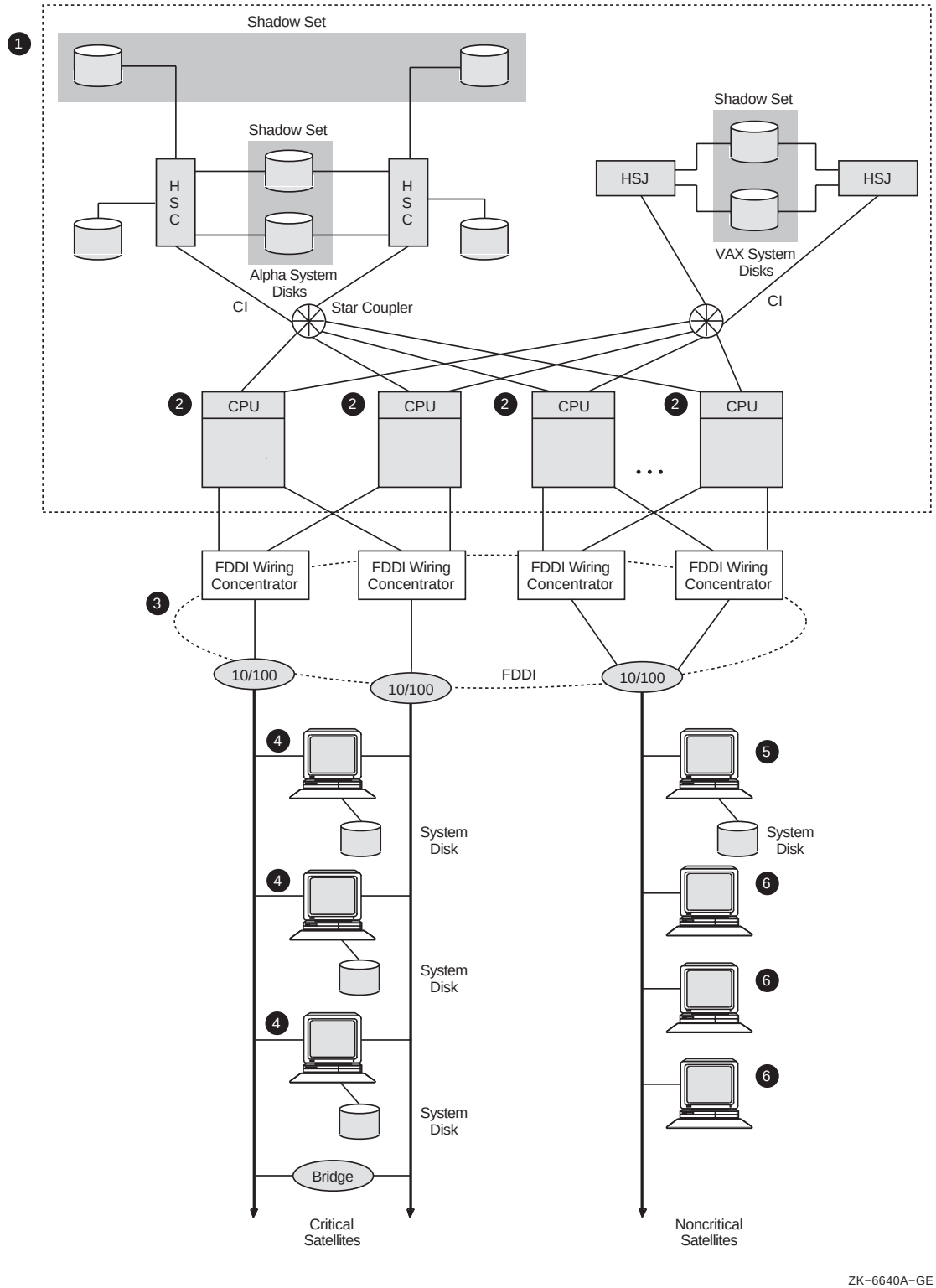
Because satellite nodes are highly dependent on server nodes for availability, the sample configurations presented earlier in this chapter do not include satellite nodes. However, because satellite/server configurations provide important advantages, you may decide to trade off some availability to include satellite nodes in your configuration.

Figure 8–7 shows an optimal configuration for a OpenVMS Cluster system with satellites. Figure 8–7 is followed by an analysis of the configuration that includes:

- Analysis of its components
- Advantages and disadvantages
- Key availability strategies implemented

The base configurations in Figure 8–4 and Figure 8–5 could replace the base configuration shown in Figure 8–7. In other words, the FDDI and satallite segments shown in Figure 8–7 could just as easily be attached to the configurations shown in Figure 8–4 and Figure 8–5.

Figure 8–7 OpenVMS Cluster with Satellites



8.10.1 Components

This satellite/server configuration in Figure 8–7 has the following components:

Part	Description
1	Base configuration.
	The base configuration performs server functions for satellites.
2	Three to 16 OpenVMS server nodes.
	Rationale: At least three nodes are recommended to maintain quorum. More than 16 nodes introduces excessive complexity.
3	FDDI ring between base server nodes and satellites.
	Rationale: The FDDI ring has increased network capacity over Ethernet, which is slower.
	Alternative: Use two Ethernet segments instead of the FDDI ring.
4	Two Ethernet segments from the FDDI ring to attach each critical satellite with two Ethernet adapters. Each of these critical satellites has its own system disk.
	Rationale: Having their own boot disks increases the availability of the critical satellites.
5	For noncritical satellites, place a boot server on the Ethernet segment.
	Rationale: Noncritical satellites do not need their own boot disks.
6	Limit the satellites to 15 per segment.
	Rationale: More than 15 satellites on a segment may cause I/O congestion.

8.10.2 Advantages

This configuration provides the following advantages:

- A large number of nodes can be served in one OpenVMS Cluster.
- You can spread a large number of nodes over a greater distance.

8.10.3 Disadvantages

This configuration has the following disadvantages:

- Satellites with single LAN adapters have a single point of failure that causes cluster transitions if the adapter fails.
- High cost of LAN connectivity for highly available satellites.

8.10.4 Key Availability Strategies

The configuration in Figure 8–7 incorporates the following strategies, which are critical to its success:

- This configuration has no single point of failure.
- The FDDI interconnect has sufficient bandwidth to serve satellite nodes from the base server configuration.
- All shared storage is MSCP served from the base configuration, which is appropriately configured to serve a large number of nodes.

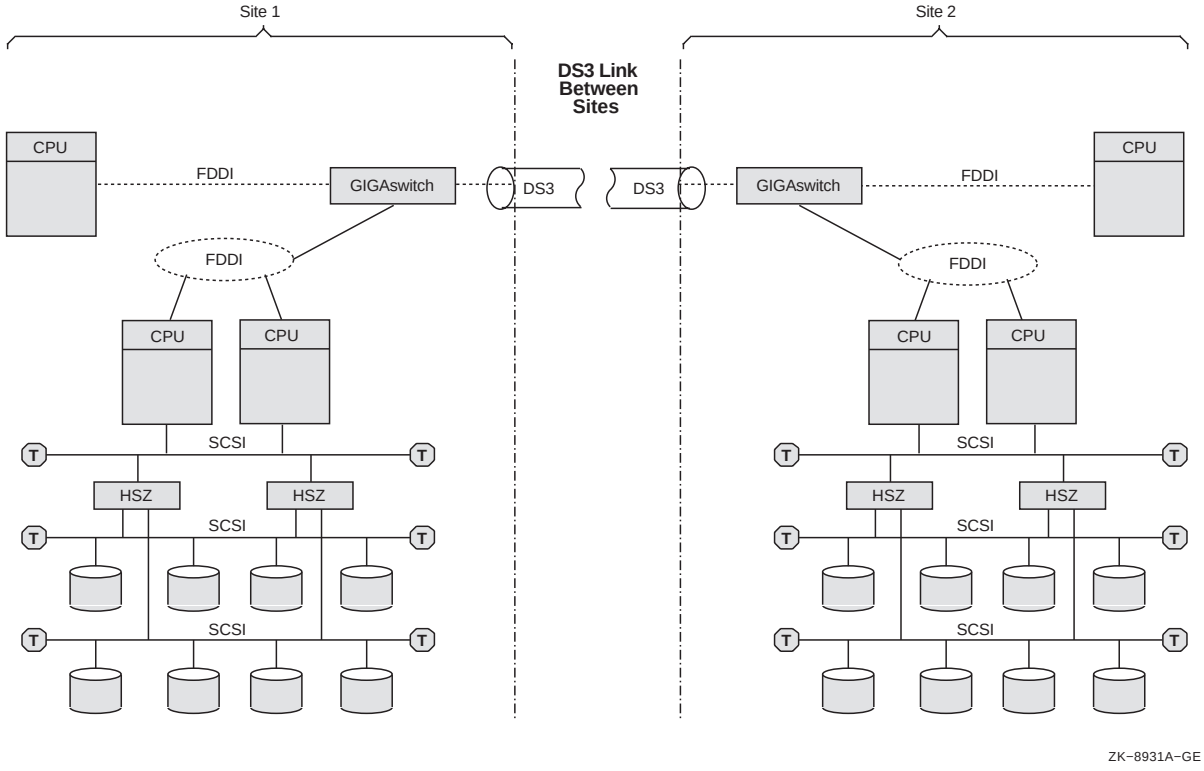
8.11 Multiple-Site OpenVMS Cluster System

Multiple-site OpenVMS Cluster configurations contain nodes that are located at geographically separated sites. Depending on the technology used, the distances between sites can be as great as 150 miles. FDDI, asynchronous transfer mode (ATM), and DS3 are used to connect these separated sites to form one large cluster. Available from most common telephone service carriers, DS3 and ATM services provide long-distance, point-to-point communications for multiple-site clusters.

Figure 8–8 shows a typical configuration for a multiple-site OpenVMS Cluster system. Figure 8–8 is followed by an analysis of the configuration that includes:

- Analysis of components
- Advantages

Figure 8–8 Multiple-Site OpenVMS Cluster Configuration Connected by DS3



8.11.1 Components

Although Figure 8–8 does not show all possible configuration combinations, a multiple-site OpenVMS Cluster can include:

- Two data centers with an intersite link (FDDI, ATM, or DS3) connected to a DECconcentrator or GIGAswitch crossbar switch.
- Intersite link performance that is compatible with the applications that are shared by the two sites.

- Up to 96 Alpha and VAX (combined total) nodes. In general, the rules that apply to OpenVMS LAN and extended LAN (ELAN) clusters also apply to multiple-site clusters.

  Reference: For LAN configuration guidelines, see Section 4.12.6. For ELAN configuration guidelines, see Section 10.7.7.

8.11.2 Advantages

The benefits of a multiple-site OpenVMS Cluster system include the following:

- A few systems can be remotely located at a secondary site and can benefit from centralized system management and other resources at the primary site. For example, a main office data center could be linked to a warehouse or a small manufacturing site that could have a few local nodes with directly attached, site-specific devices. Alternatively, some engineering workstations could be installed in an office park across the city from the primary business site.
- Multiple sites can readily share devices such as high-capacity computers, tape libraries, disk archives, or phototypesetters.
- Backups can be made to archival media at any site in the cluster. A common example would be to use disk or tape at a single site to back up the data for all sites in the multiple-site OpenVMS Cluster. Backups of data from remote sites can be made transparently (that is, without any intervention required at the remote site).
- In general, a multiple-site OpenVMS Cluster provides all of the availability advantages of a LAN OpenVMS Cluster. Additionally, by connecting multiple, geographically separate sites, multiple-site OpenVMS Cluster configurations can increase the availability of a system or elements of a system in a variety of ways:
  - Logical volume/data availability—Volume shadowing or redundant arrays of independent disks (RAID) can be used to create logical volumes with members at both sites. If one of the sites becomes unavailable, data can remain available at the other site.
  - Site failover—By adjusting the VOTES system parameter, you can select a preferred site to continue automatically if the other site fails or if communications with the other site are lost.

Reference: For additional information about multiple-site clusters, see *OpenVMS Cluster Systems*.

8.12 Disaster-Tolerant OpenVMS Cluster Configurations

Disaster-tolerant OpenVMS Cluster configurations make use of Volume Shadowing for OpenVMS, high-speed networks, and specialized management software.

Disaster-tolerant OpenVMS Cluster configurations enable systems at two different geographic sites to be combined into a single, manageable OpenVMS Cluster system. Like the multiple-site cluster discussed in the previous section, these physically separate data centers are connected by FDDI or by a combination of FDDI and ATM, T3, or E3.

The OpenVMS disaster tolerant product was formerly named the Business Recovery Server (BRS). BRS has been subsumed by a services offering named Disaster Tolerant Cluster Services, which is a system management and software service package. For more information about Disaster Tolerant Cluster Services, contact your Compaq Services representative.

9

Configuring CI OpenVMS Clusters for Availability and Performance

There are many ways to configure a CI (cluster interconnect) OpenVMS Cluster system. This chapter describes how to configure CI OpenVMS Clusters to maximize both availability and performance. This is done by presenting a series of configuration examples of increasing complexity, followed by a comparative analysis of each example. These configurations illustrate basic techniques that can be scaled upward to meet the availability, I/O performance, and storage connectivity needs of very large clusters.

9.1 CI Components

The CI is a radial bus through which OpenVMS Cluster systems communicate with each other and with storage. The CI consists of the following components:

- CI host adapter
- HSJ or HSC storage controller

  An HSJ or HSC storage controller is optional but generally present.
- CI cables

  For each of the CI's two independent paths (called path A and path B), there is a transmit and receive cable pair.
- Star coupler

  This is a passive device that serves as a common connection point for signals between OpenVMS nodes and HSC or HSJ controllers that are connected to the CI. A star coupler consists of two completely independent and electrically isolated "path hubs." Each CI path hub is extremely reliable because it contains only transformers carrying low-power signals.

Availability and performance can both be increased by adding components. Components added for availability need to be configured so that a redundant component is available to assume the work being performed by a failed component. Components added for performance need to be configured so that the additional components can work in parallel with other components.

Frequently, you need to maximize both availability and performance. The techniques presented here are intended to help achieve these dual goals.

9.2 Configuration Assumptions

The configurations shown here are based on the following assumptions:

1. MSCP serving is enabled.
2. Volume Shadowing for OpenVMS is installed.
3. When performance is being discussed:
   a. CI host adapters are CIPCA or CIXCD.

      Older CI adapter models are significantly slower.

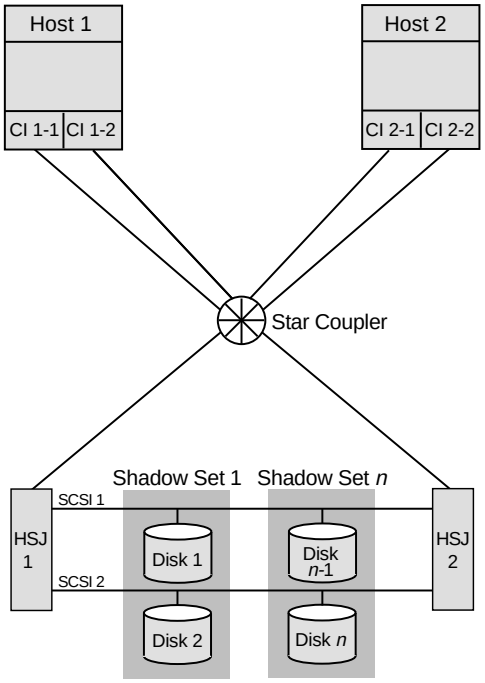
   b. CI storage controllers are HSJ50s.

      Compared with HSJ50s, HSJ40s are somewhat slower, and HSC models are significantly slower.

9.3 Configuration 1

Configuration 1, shown in Figure 9–1, provides no single point of failure. Its I/O performance is limited by the bandwidth of the star coupler.

Figure 9–1 Redundant HSJs and Host CI Adapters Connected to Same CI (Configuration 1)



9.3.1 Components

The CI configuration shown in Figure 9–1 has the following components:

Part	Description
Host 1, Host 2	Dual CI capable OpenVMS Alpha or VAX hosts. **Rationale:** Either host can fail and the system can continue. The full performance of both hosts is available for application use under normal conditions.
CI 1-1,CI 1-2, CI 2-1,CI 2-2	Dual CI adapters on each host. **Rationale:** Either of a host's CI adapters can fail and the host will retain CI connectivity to the other host and to the HSJ storage controllers.
Star Coupler	One star coupler cabinet containing two independent path hubs. The star coupler is redundantly connected to the CI host adapters and HSJ storage controllers by a transmit/receive cable pair per path. **Rationale:** Either of the path hubs or an attached cable could fail and the other CI path would continue to provide full CI connectivity. When both paths are available, their combined bandwidth is usable for host-to-host and host-to-storage controller data transfer.
HSJ 1, HSJ 2	Dual HSJ storage controllers in a single StorageWorks cabinet. **Rationale:** Either storage controller can fail and the other controller can assume control of all disks by means of the SCSI buses shared between the two HSJs. When both controllers are available, each can be assigned to serve a portion of the disks. Thus, both controllers can contribute their I/O-per-second and bandwidth capacity to the cluster.
SCSI 1, SCSI 2	Shared SCSI buses between HSJ pairs. **Rationale:** Provide access to each disk on a shared SCSI from either HSJ storage controller. This effectively dual ports the disks on that bus.
Disk 1, Disk 2, . . . Disk *n*-1, Disk *n*	Critical disks are dual ported between HSJ pairs by shared SCSI buses. **Rationale:** Either HSJ can fail, and the other HSJ will assume control of the disks that the failed HSJ was controlling.
Shadow Set 1 through Shadow Set *n*	Essential disks are shadowed by another disk that is connected on a different shared SCSI. **Rationale:** A disk or the SCSI bus to which it is connected, or both, can fail, and the other shadow set member will still be available. When both disks are available, their combined READ I/O capacity and READ data bandwidth capacity are available to the cluster.

9.3.2 Advantages

This configuration offers the following advantages:

- All nodes have direct access to storage.
- Highly expandable.
- CI is inherently dual pathed.
- No single component failure can disable the cluster.
- If a CI adapter fails, or both its paths are disabled, OpenVMS will automatically fail over all I/O and cluster traffic to the other CI adapter.
- Disks are dual ported between HSJ controllers; automatic disk failover to the other controller if an HSJ fails or if an HSJ loses both paths to a star coupler.
- Redundant storage controllers can be used to provide additional performance by dividing disks between the two storage controllers.

Disks can be assigned to HSJ storage controllers by the OpenVMS Prefer utility supplied in SYS$EXAMPLES, or by issuing a $QIO call with IO$_SETPRFPATH and IO$M_FORCEPATH modifiers, or by using the HSJ SET_PREFERRED command (less desirable; use only for this configuration).

- Critical disks are shadowed with shadow set members on different SCSI buses.
- Read I/Os are automatically load balanced across shadow set members for performance.
- Lowest cost.

9.3.3 Disadvantages

This configuration has the following disadvantages:

- Second CI adapter in each host is unlikely to enhance performance.
- Both HSJs have to share the bandwidth of a single CI.
- Failure of a CI path hub or path cable halves the bandwidth available to all CI components that use the failed component.
- Physical damage to a star coupler or associated cables is likely to disable the entire CI, rendering the cluster unusable.
- Physical damage to the StorageWorks cabinet could render the cluster unusable.

9.3.4 Key Availability and Performance Strategies

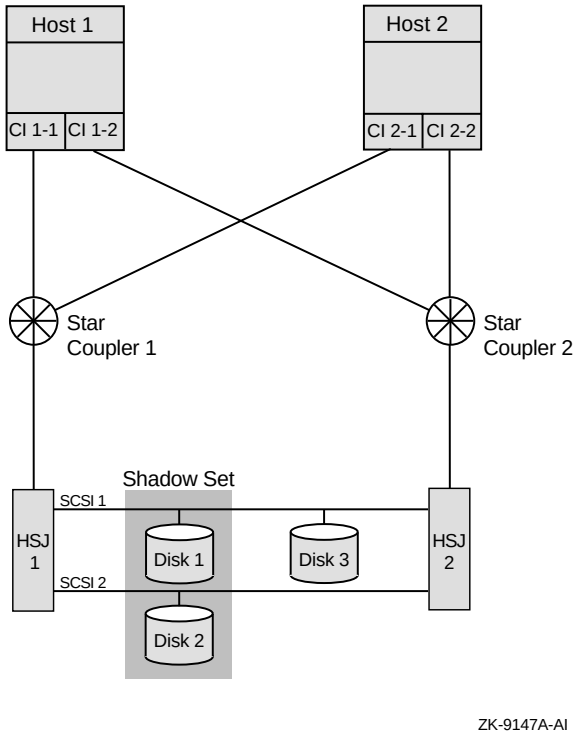
This configuration incorporates the following strategies:

- All components are duplicated.
- Redundant storage controllers are included.
- This configuration has no single point of failure.
- Dual porting and volume shadowing provide multiple copies of essential disks across separate HSJ controllers.
- All nodes have shared, direct access to all storage.
- A quorum disk allows other node to continue if one node fails. (Alternatively, at least three nodes could be used.)

9.4 Configuration 2

The configuration illustrated in Figure 9–2 with redundant HSJs, host CI adapters, and CIs provides no electrical single point of failure. Its two star couplers provide increased I/O performance and availability over configuration 1.

Figure 9–2 Redundant HSJs and Host CI Adapters Connected to Redundant CIs (Configuration 2)



9.4.1 Components

Configuration 2 has the following components:

Part	Description
Host 1, Host 2	Dual CI capable OpenVMS Alpha or VAX hosts. **Rationale:** Either host can fail and the system can continue to run. The full performance of both hosts is available for application use under normal conditions.
CI 1-1,CI 1-2, CI 2-1, CI 2-2	Dual CI adapters on each host. Adapter CI 1-*n* is Host 1's CI adapter connected to CI *n*, and so on. **Rationale:** Either of a host's CI adapters can fail, and the host will retain CI connectivity to the other host and to the HSJ storage controllers. Each CI adapter on a host is connected to a different star coupler. In the absence of failures, the full data bandwidth and I/O-per-second capacity of both CI adapters is available to the host.
Star Coupler 1, Star Coupler 2	Two star couplers, each consisting of two independent path hub sections. Each star coupler is redundantly connected to the CI host adapters and HSJ storage controllers by a transmit/receive cable pair per path. **Rationale:** Any of the path hubs or an attached cable could fail and the other CI path would continue to provide full connectivity for that CI. Loss of a path affects only the bandwidth available to the storage controller and host adapters connected to the failed path. When all paths are available, the *combined bandwidth of both CIs is usable.*

Part	Description
HSJ 1, HSJ 2	Dual HSJ storage controllers in a single StorageWorks cabinet. **Rationale:** Either storage controller can fail and the other controller can control any disks the failed controller was handling by means of the SCSI buses shared between the two HSJs. When both controllers are available, each can be assigned to serve a subset of the disks. Thus, both controllers can contribute their I/O-per-second and bandwidth capacity to the cluster.
SCSI 1, SCSI 2	Shared SCSI buses connected between HSJ pairs. **Rationale:** Either of the shared SCSI buses could fail and access would still be provided from the HSJ storage controllers to each disk by means of the remaining shared SCSI bus. This effectively dual ports the disks on that bus.
Disk 1, Disk 2, . . . Disk *n*-1, Disk *n*	Critical disks are dual ported between HSJ pairs by shared SCSI buses. **Rationale:** Either HSJ can fail and the other HSJ will assume control of the disks the failed HSJ was controlling.
Shadow Set 1 through Shadow Set *n*	Essential disks are shadowed by another disk that is connected on a different shared SCSI. **Rationale:** A disk or the SCSI bus to which it is connected, or both, can fail and the other shadow set member will still be available. When both disks are available, both can provide their READ I/O capacity and their READ data bandwidth capacity to the cluster.

9.4.2 Advantages

Configuration 2 offers all the advantages of Configuration 1 plus the following advantages:

- CI is likely to remain fully usable in the event of localized damage to a star coupler cabinet or to the cables connected to it.
- Failure of an HSJ, or an HSJ losing both paths to a star coupler host will result in the other HSJ assuming control of all dual-pathed disks.
- Host with connectivity will automatically MSCP serve unreachable disks to a host that has lost connectivity to an HSJ.
- Dual-pathed disks can be switched to an HSJ with full host connectivity if a host loses connectivity to an HSJ.

  The disks on the HSJ that cannot be reached by a host can be reassigned to the HSJ with full connectivity by operator command, or by a DCL or other program that monitors for connectivity loss. You can assign disks to another storage controller with the Prefer utility supplied in SYS$EXAMPLES, or by issuing a $QIO call with IO$_SETPRFPATH and IO$M_FORCEPATH modifiers, or by powering off the HSJ with reduced connectivity (less desirable).

  The HSJ SET_PREFERRED command is **not** recommended in this configuration because this command can not be overridden by a host PREFER or IO$_SETPRFPATH modifier. A new SET_PREFERRED command assigning a device to another HSJ will not take effect until the HSJ, to which the device was assigned by a previous SET_PREFERRED command, is power cycled.)

9.4.3 Disadvantages

Configuration 2 has the following disadvantages:

- Host CI adapter failure will *not* cause disks to automatically fail over to an HSJ that still has full host connectivity.

  If a host CI adapter fails and if MSCP serving is enabled, then the other OpenVMS system will begin serving the unreachable HSJ's disks to the host with the failed adapter.

- Physical damage to a star coupler or associated cables is likely to disable the entire CI, rendering the cluster unusable.
- Physical damage to the StorageWorks cabinet could render the cluster unusable.
- Higher cost than configuration 1.

9.4.4 Key Availability and Performance Strategies

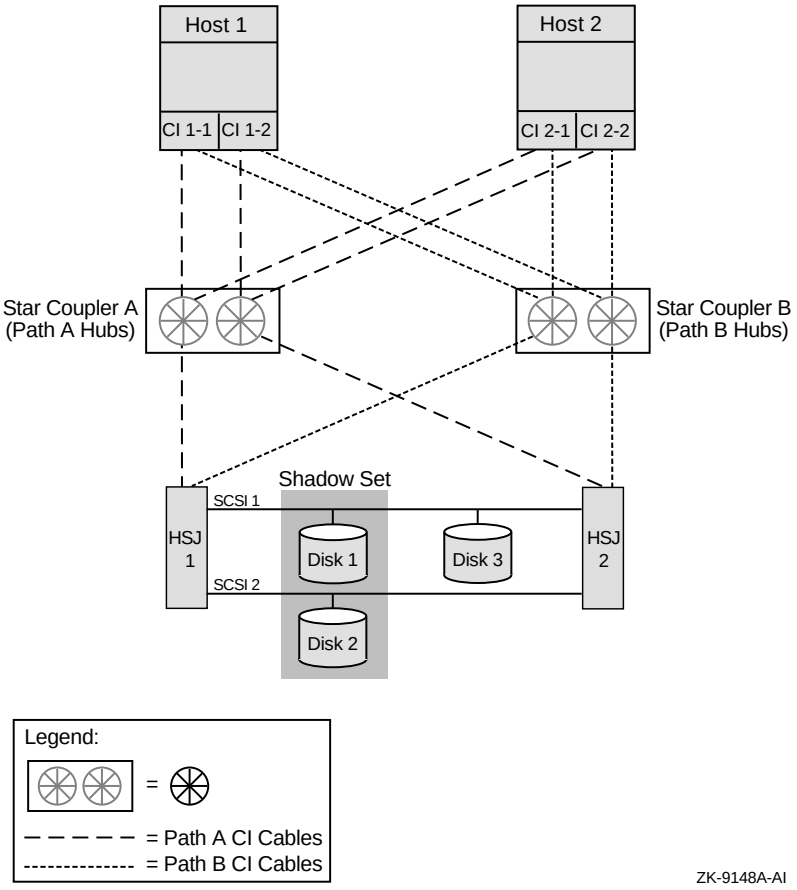
Configuration 2 provides configuration 1 strategies, plus:

- Second CI provides additional redundancy.
- Second CI cables and star coupler are physically separate.

9.5 Configuration 3

The availability of a CI configuration can be further improved by physically separating the path A and path B CI cables and their associated path hubs. This significantly reduces the probability of a mechanical accident or other localized damage destroying both paths of a CI. This configuration is shown in Figure 9–3.

Figure 9–3 Redundant Components and Path-Separated Star Couplers (Configuration 3)



Configuration 3 is electrically identical to configuration 2. However, the path A and path B cables are physically separate for both CIs. The path A cables for both CI 1 and CI 2 are routed together to star coupler cabinet A, but are connected to different CI path hubs in that cabinet.

Similarly, the path B cables for both CIs go to different CI path hubs in star coupler cabinet B. The path-specific star coupler cabinets and associated path cables should be separated as much as possible. For example, the star coupler cabinets could be installed on opposite sides of a computer room, and the CI cables could be routed so that path A and path B cables follow different paths.

Note

The path separation technique illustrated for configuration 3 (Figure 9–3) can also be applied to configuration 1 (Figure 9–1). In this case, each star coupler cabinet would have only one path hub. The CI's path A cables would go to the path hub in Star Coupler A. Similarly, the path B cables would go to Star Coupler B.

9.5.1 Components

The CI OpenVMS Cluster configuration shown in Figure 9–3 has the following components:

Part	Description
Host 1, Host 2	Dual CI capable OpenVMS Alpha or VAX hosts. **Rationale:** Either host can fail and the system can continue. The full performance of both hosts is available for application use under normal conditions.
CI 1-1,CI 1-2, CI 2-1, CI 2-2	Dual CI adapters on each host. Adapter CI 1-*n* is Host 1's CI adapter connected to CI *n*, and so on. **Rationale:** Either host's CI adapters can fail and the host will retain CI connectivity to the other host and to the HSJ storage controllers. Each CI adapter on a host is connected to a different star coupler. In the absence of failures, the full data bandwidth and I/O-per-second capacity of both CI adapters is available to the host.
Star Coupler A (Path A Hubs), Star Coupler B (Path B Hubs)	Two CI star couplers, each comprising two independent path hubs. Star Coupler A's path hubs are connected to path A cables for both CIs, and Star Coupler B's path hubs are connected to path B cables for both CIs. **Rationale:** Mechanical or other localized damage to a star coupler or an attached cable would probably not affect the other CI paths. The other paths and star coupler would continue to provide full connectivity for both CIs. Loss of a path affects only the bandwidth available to the storage controllers and host adapters connected to the failed path. When all paths are available, the *combined bandwidth of both CIs is usable*.
Path A CI Cables, Path B CI Cables	Each path's hub is connected to the CI host adapters and HSJ storage controllers by a transmit/receive cable pair per path. The path A cables of both CIs are routed together, but their routing differs from the routing of the path B cables.
HSJ 1, HSJ 2	Dual HSJ storage controllers in a single StorageWorks cabinet. **Rationale:** Either storage controller can fail and the other controller can control any disks the failed controller was handling by means of the SCSI buses shared between the two HSJs. When both controllers are available, each can be assigned to serve a subset of the disks. Thus, both controllers can contribute their I/O-per-second and bandwidth capacity to the cluster.
SCSI 1, SCSI 2	Shared SCSI buses connected between HSJ pairs. **Rationale:** Provide access to each disk on a shared SCSI bus from either HSJ storage controller. This effectively dual ports the disks on that bus.
Disk 1, Disk 2, . . . Disk *n*-1, Disk *n*	Critical disks are dual ported between HSJ pairs by shared SCSI buses. **Rationale:** Either HSJ can fail and the other HSJ will assume control of the disks that the failed HSJ was controlling.
Shadow Set 1 through Shadow Set *n*	Essential disks are shadowed by another disk that is connected on a different shared SCSI. **Rationale:** A disk, or the SCSI bus to which it is connected, or both, can fail and the other shadow set member will still be available. When both disks are available, their combined READ I/O-per-second capacity and READ data bandwidth capacity is available to the cluster.

9.5.2 Advantages

Configuration 3 offers the same individual component advantages as configuration 2, plus:

- Both CIs remain usable in the event of localized damage to a CI path.
- Same cost as configuration 2 with better availability.

9.5.3 Disadvantages

Configuration 3 has the following disadvantages:

- Host CI adapter failure will **not** cause disks to automatically fail over to an HSJ that still has full host connectivity.

  If a host CI adapter fails and if MSCP serving is enabled, the other OpenVMS system will begin serving the unreachable HSJ's disks to the host with the failed adapter.
- Damage to star coupler or associated cables likely to affect the same path of both CIs.
- Damage affecting path *n* of both CIs can halve the usable bandwidth of both CIs.
- Physical damage to the StorageWorks cabinet could render the cluster unusable.

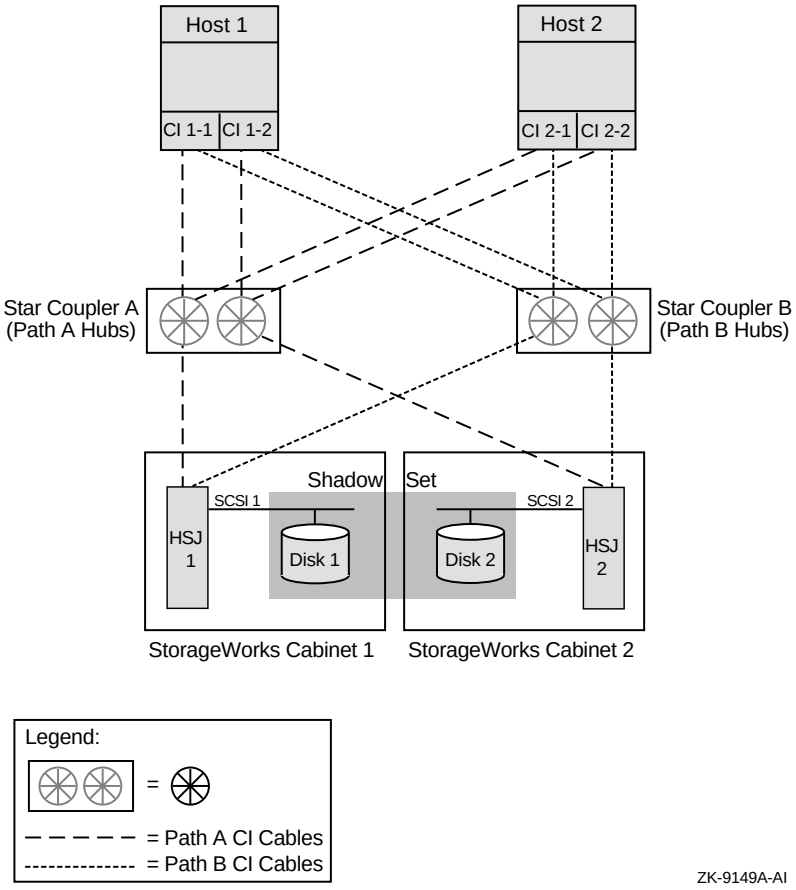
9.5.4 Key Availability and Performance Strategies

Configuration 3 provides all the strategies of configuration 2 except for physical separation of CIs. The major advantage over configuration 2 are the path-specific star coupler cabinets. They provide physical isolation of the path A cables and the path A hub from the path B cables and the path B hub.

9.6 Configuration 4

The availability of a CI configuration can be further improved by physically separating shadow set members and their HSJ controllers. This significantly reduces the probability of a mechanical accident or other localized damage that could destroy both members of a shadow set. This configuration is shown in Figure 9–4.

Figure 9–4 Redundant Components, Path-Separated Star Couplers, and Duplicate StorageWorks Cabinets (Configuration 4)



Configuration 4 is similar to configuration 3 except that the shadow set members and their HSJ controllers are mounted in separate StorageWorks cabinets that are located some distance apart.

The StorageWorks cabinets, path-specific star coupler cabinets, and associated path cables should be separated as much as possible. For example, the StorageWorks cabinets and the star coupler cabinets could be installed on opposite sides of a computer room. The CI cables should be routed so that path A and path B cables follow different paths.

Note

The separate StorageWorks cabinets technique illustrated in configuration 4 (Figure 9–4) can also be applied to configuration 1 (Figure 9–1) and configuration 2 (Figure 9–2).

9.6.1 Components

The CI OpenVMS Cluster configuration shown in Figure 9–4 has the following components:

Part	Description
Host 1, Host 2	Dual CI capable OpenVMS Alpha or VAX hosts. **Rationale:** Either host can fail and the system can continue to run. The full performance of both hosts is available for application use under normal conditions.
CI 1-1,CI 1-2, CI 2-1, CI 2-2	Dual CI adapters on each host. Adapter CI 1-*n* is Host 1's CI adapter connected to CI *n*, and so on. **Rationale:** Either of a host's CI adapters can fail and the host will retain CI connectivity to the other host and the HSJ storage controllers. Each CI adapter on a host is connected to a different star coupler. In the absence of failures, the full data bandwidth and I/O-per-second capacity of both CI adapters are available to the host.
Star Coupler A (Path A Hubs), Star Coupler B (Path B Hubs)	Two CI star couplers, each comprising two independent path hub sections. Star Coupler A's path hubs are connected to the path A cables for both CIs, and Star Coupler B's path hubs are connected to the path B cables for both CIs. **Rationale:** Mechanical or other localized damage to a star coupler or an attached cable would probably not affect the other CI paths. The other paths and star coupler would continue to provide full connectivity for both CIs. Loss of a path affects the bandwidth available to the storage controllers and host adapters that are connected to the failed path. When all paths are available, the combined bandwidth of both CIs is usable.
Path A CI cables, Path B CI cables	Each path's hub is connected to the CI host adapters and HSJ storage controllers by a transmit/receive cable pair per path. The path A cables of both CIs are routed together, but their routing differs from the routing of the path B cables.
HSJ 1, HSJ 2	Dual HSJ storage controllers, each in a separate StorageWorks cabinet. Data is replicated across StorageWorks cabinets using Volume Shadowing for DIGITAL OpenVMS. **Rationale:** A StorageWorks cabinet can be destroyed, or one storage controller can fail, and the remaining controller located in the other StorageWorks cabinet can control shadow copies of all disks. When both controllers are available, each can be assigned to serve a subset of the disks. Volume shadowing will distribute READ I/Os across the HSJs. Thus, both controllers can contribute their I/O-per-second and bandwidth capacity to the cluster.
SCSI 1, SCSI 2	Private SCSI buses connected to an HSJ. **Rationale:** Provide host access to each shadow set member.
Shadow Set	Essential disks are shadowed between HSJ pairs using volume shadowing. Each HSJ and its disks are in a StorageWorks cabinet that is physically separated from the other StorageWorks cabinet. **Rationale:** An entire StorageWorks cabinet can be destroyed, or a disk, the SCSI bus, or the HSJ to which it is connected can fail, and the other shadow set member will still be available. When both disks are available, they can each provide their READ I/O per second capacity and READ data bandwidth capacity to the cluster.

9.6.2 Advantages

Configuration 4 offers most of the individual component advantages of configuration 3, plus:

- Physical damage to one StorageWorks cabinet is not likely to disable the cluster.
- Disk failover to another HSJ is not a problem.

9.6.3 Disadvantages

Configuration 4 has the following disadvantages:

- Does not have redundant paths to each disk.
- Write I/Os to shadow sets use twice the CI bandwidth of other alternatives.
- Higher cost than configuration 2 and configuration 3.

9.6.4 Key Availability and Performance Strategies

Configuration 4 (Figure 9–4) provides all of the strategies of configuration 3. It also provides shadow set members that are in physically separate StorageWorks cabinets.

9.7 Summary

All four configurations illustrate how to obtain both availability and performance by:

- Adding redundant components
- Dual-porting critical disks
- Shadowing essential disks

An advanced technique, separating the CI path A and path B cables and associated hubs, is used in configuration 3 and configuration 4. This technique increases availability and maintains performance with no additional hardware. Configuration 4 provides even greater availability without compromising performance by physically separating shadow set members and their HSJ controllers.

Using these configurations as a guide, you can select the techniques that are appropriate for your computing needs and adapt your environment as conditions change. The techniques illustrated in these configurations can be scaled for larger CI configurations.

10

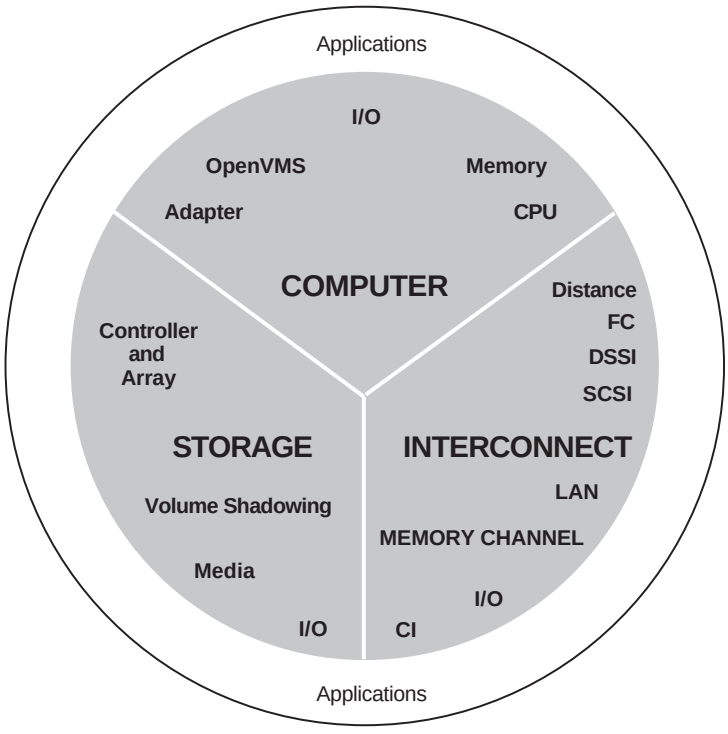
Configuring OpenVMS Clusters for Scalability

This chapter explains how to maximize scalability in many different kinds of OpenVMS Clusters.

10.1 What Is Scalability?

Scalability is the ability to expand an OpenVMS Cluster system in any system, storage, and interconnect dimension and at the same time fully use the initial configuration equipment. Your OpenVMS Cluster system can grow in many dimensions, as shown in Figure 10–1. Each dimension also enables your applications to expand.

Figure 10–1 OpenVMS Cluster Growth Dimensions



VM-0218A-AI

10.1.1 Scalable Dimensions

Table 10–1 describes the growth dimensions for systems, storage, and interconnects in OpenVMS Clusters.

Table 10–1 Scalable Dimensions in OpenVMS Clusters

This Dimension	Grows by...
Systems	
CPU	Implementing SMP within a system. Adding systems to a cluster. Accommodating various processor sizes in a cluster. Adding a bigger system to a cluster. Migrating from VAX to Alpha systems.
Memory	Adding memory to a system.
I/O	Adding interconnects and adapters to a system. Adding MEMORY CHANNEL to a cluster to offload the I/O interconnect.
OpenVMS	Tuning system parameters. Moving to OpenVMS Alpha.
Adapter	Adding storage adapters to a system. Adding CI and DSSI adapters to a system. Adding LAN adapters to a system.
Storage	
Media	Adding disks to a cluster. Adding tapes and CD–ROMs to a cluster.
Volume shadowing	Increasing availability by shadowing disks. Shadowing disks across controllers. Shadowing disks across systems.
I/O	Adding solid-state or DECram disks to a cluster. Adding disks and controllers with caches to a cluster. Adding RAID disks to a cluster.
Controller and array	Moving disks and tapes from systems to controllers. Combining disks and tapes in arrays. Adding more controllers and arrays to a cluster.
Interconnect	
LAN	Adding Ethernet and FDDI segments. Upgrading from Ethernet to FDDI. Adding redundant segments and bridging segments.
CI, DSSI, Fibre Channel, SCSI, and MEMORY CHANNEL	Adding CI, DSSI, Fibre Channel, SCSI, and MEMORY CHANNEL interconnects to a cluster or adding redundant interconnects to a cluster.
I/O	Adding faster interconnects for capacity. Adding redundant interconnects for capacity and availability.
Distance	Expanding a cluster inside a room or a building. Expanding a cluster across a town or several buildings. Expanding a cluster between two sites (spanning 40 km).

The ability to add to the components listed in Table 10–1 in any way that you choose is an important feature that OpenVMS Clusters provide. You can add hardware and software in a wide variety of combinations by carefully following the suggestions and guidelines offered in this chapter and in the products' documentation and *DIGITAL Systems and Options Catalog*. When you choose to expand your OpenVMS Cluster in a specific dimension, be aware of the advantages and tradeoffs with regard to the other dimensions. Table 10–2 describes strategies that promote OpenVMS Cluster scalability. Understanding these scalability strategies can help you maintain a higher level of performance and availability as your OpenVMS Cluster grows.

10.2 Strategies for Configuring a Highly Scalable OpenVMS Cluster

The hardware that you choose and the way that you configure it has a significant impact on the scalability of your OpenVMS Cluster. This section presents strategies for designing an OpenVMS Cluster configuration that promotes scalability.

10.2.1 Scalability Strategies

Table 10–2 lists strategies in order of importance that ensure scalability. This chapter contains many figures that show how these strategies are implemented.

Table 10–2 Scalability Strategies

Strategy	Description
Capacity planning	Running a system above 80% capacity (near performance saturation) limits the amount of future growth possible. Understand whether your business and applications will grow. Try to anticipate future requirements for processor, memory, and I/O.
Shared, direct access to all storage	The ability to scale compute and I/O performance is heavily dependent on whether all of the systems have shared, direct access to all storage. The CI and DSSI OpenVMS Cluster illustrations that follow show many examples of shared, direct access to storage, with no MSCP overhead. **Reference:** For more information about MSCP overhead, see Section 10.8.1.
Limit node count to between 3 and 16	Smaller OpenVMS Clusters are simpler to manage and tune for performance and require less OpenVMS Cluster communication overhead than do large OpenVMS Clusters. You can limit node count by upgrading to a more powerful processor and by taking advantage of OpenVMS SMP capability. If your server is becoming a compute bottleneck because it is overloaded, consider whether your application can be split across nodes. If so, add a node; if not, add a processor (SMP).
Remove system bottlenecks	To maximize the capacity of any OpenVMS Cluster function, consider the hardware and software components required to complete the function. Any component that is a bottleneck may prevent other components from achieving their full potential. Identifying bottlenecks and reducing their effects increases the capacity of an OpenVMS Cluster.
Enable the MSCP server	The MSCP server enables you to add satellites to your OpenVMS Cluster so that all nodes can share access to all storage. In addition, the MSCP server provides failover for access to shared storage when an interconnect fails.
Reduce interdependencies and simplify configurations	An OpenVMS Cluster system with one system disk is completely dependent on that disk for the OpenVMS Cluster to continue. If the disk, the node serving the disk, or the interconnects between nodes fail, the entire OpenVMS Cluster system may fail.
Ensure sufficient serving resources	If a small disk server has to serve a large number disks to many satellites, the capacity of the entire OpenVMS Cluster is limited. Do not overload a server because it will become a bottleneck and will be unable to handle failover recovery effectively.
Configure resources and consumers close to each other	Place servers (resources) and satellites (consumers) close to each other. If you need to increase the number of nodes in your OpenVMS Cluster, consider dividing it. See Section 11.2.4 for more information.

(continued on next page)

Table 10–2 (Cont.) Scalability Strategies

Strategy	Description
Set adequate system parameters	If your OpenVMS Cluster is growing rapidly, important system parameters may be out of date. Run AUTOGEN, which automatically calculates significant system parameters and resizes page, swap, and dump files.

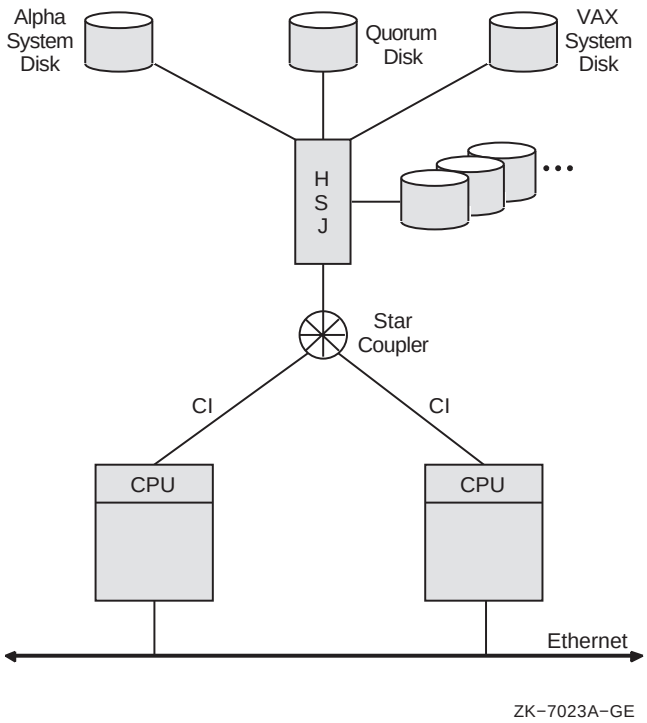
10.3 Scalability in CI OpenVMS Clusters

Each CI star coupler can have up to 32 nodes attached; 16 can be systems and the rest can be storage controllers and storage. Figure 10–2, Figure 10–3, and Figure 10–4 show a progression from a two-node CI OpenVMS Cluster to a seven-node CI OpenVMS Cluster.

10.3.1 Two-Node CI OpenVMS Cluster

In Figure 10–2, two nodes have shared, direct access to storage that includes a quorum disk. The VAX and Alpha systems each have their own system disks.

Figure 10–2 Two-Node CI OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–2 include:

Advantages

- All nodes have shared, direct access to all storage.
- As the nodes and storage in this configuration grow, all nodes can still have shared, direct access to storage.
- The MSCP server is enabled for failover to the LAN interconnect in case the CI fails. Enabling the MSCP server also allows you to add satellites.
- This configuration has the lowest cost of the CI configurations shown in this section.

Disadvantage

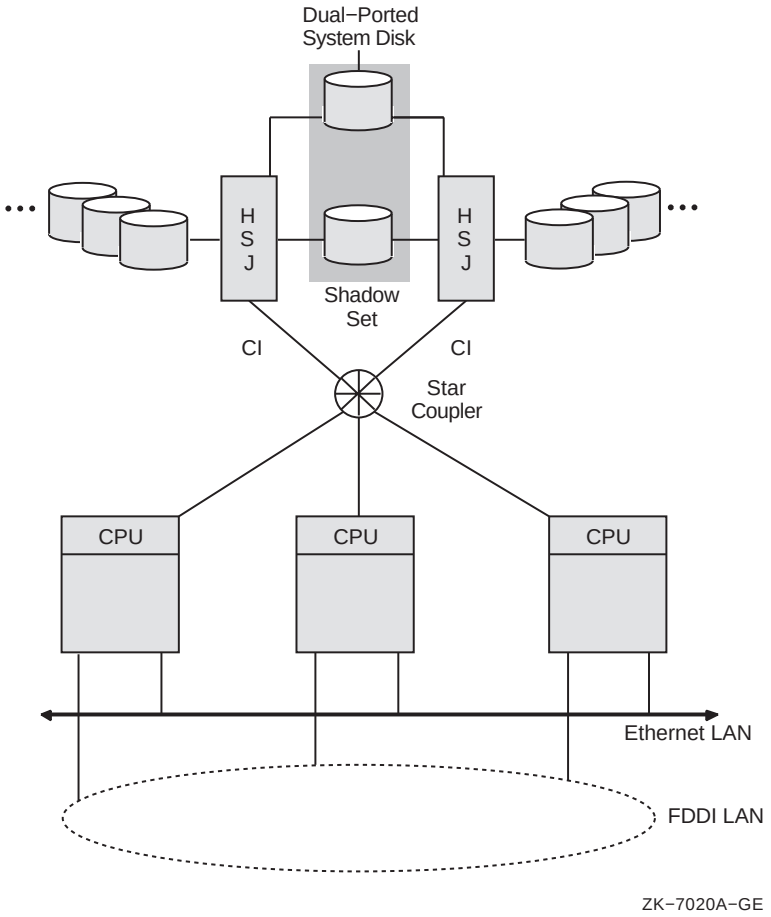
- The single HSJ/HSC is a potential bottleneck and single point of failure.

An increased need for more storage or processing resources could lead to an OpenVMS Cluster configuration like the one shown in Figure 10–3.

10.3.2 Three-Node CI OpenVMS Cluster

In Figure 10–3, three nodes are connected to two HSC controllers by the CI interconnects. The critical system disk is dual ported and shadowed.

Figure 10–3 Three-Node CI OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–3 include:

Advantages

- All nodes have shared, direct access to all storage.
- As the nodes and storage in this configuration grow, all nodes can still have shared, direct access to storage.
- The MSCP server is enabled for failover to the LAN interconnect, in case the CI fails. Enabling the MSCP server also allows you to add satellites.
- Volume shadowed, dual-ported disks increase data availability.

Disadvantage

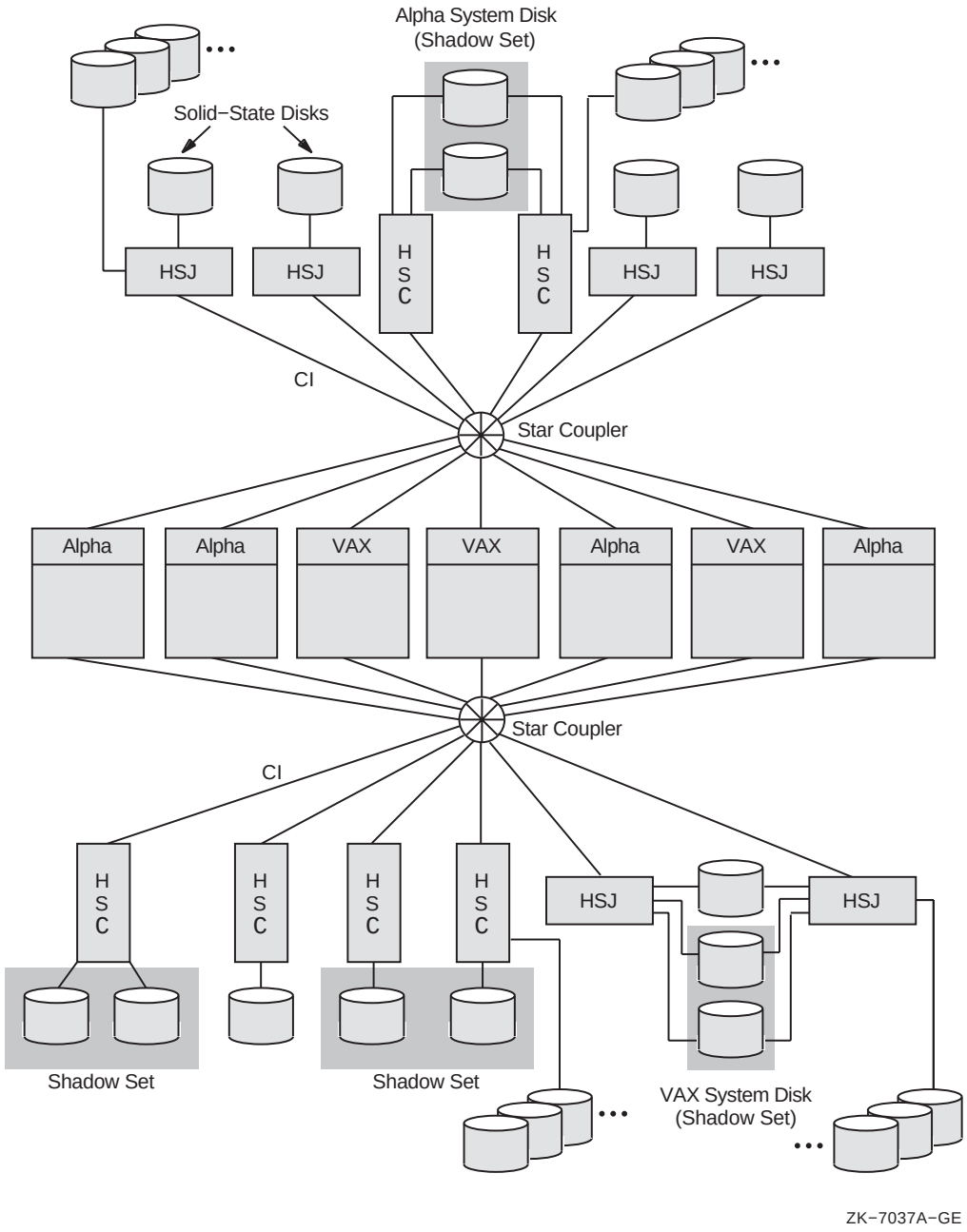
- The HSJs/HSCs are potential bottlenecks.

If the I/O activity exceeds the capacity of the CI interconnect, this could lead to an OpenVMS Cluster configuration like the one shown in Figure 10–4.

10.3.3 Seven-Node CI OpenVMS Cluster

In Figure 10–4, seven nodes each have a direct connection to two star couplers and to all storage.

Figure 10–4 Seven-Node CI OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–4 include:

Advantages

- All nodes have shared, direct access to all storage.
- This configuration has more than double the storage, processing, and CI interconnect capacity than a configuration like the one shown in Figure 10–3.
- Two CI interconnects between processors and storage provide twice the communication performance of one path.
- Volume shadowed, dual-ported disks increase data availability.

Disadvantage

- This configuration is complex and requires experienced personnel to configure, tune, and manage it properly.

10.3.4 Guidelines for CI OpenVMS Clusters

The following guidelines can help you configure your CI OpenVMS Cluster:

- Every system should have shared, direct access to all storage.
- In a CI OpenVMS Cluster larger than four nodes, use a second system disk for increased system performance.

  Reference: For more information on system disks, see Section 11.2.
- Enable your systems, interconnects, and storage to work to their full capacity by eliminating bottlenecks. If any of these components is not able to handle the I/O capacity, none of the other components will work at its best. Ensure that the sum of all the I/O on your nodes is less than or equal to your CI capacity and to your storage capacity. In calculating the sum of the I/O on your nodes, factor in 5–10% extra for lock manager internode communications.

  In general, use the following rules of thumb:

  – The sum of all the I/Os on your nodes plus internode communications should be less than or equal to the sum of your CI capacity.

  – Your CI capacity should be less than or equal to the sum of your storage capacity.

10.3.5 Guidelines for Volume Shadowing in CI OpenVMS Clusters

Volume shadowing is intended to enhance availability, not performance. However, the following volume shadowing strategies enable you to utilize availability features while also maximizing I/O capacity. These examples show CI configurations, but they apply to DSSI and SCSI configurations, as well.

Figure 10–5 Volume Shadowing on a Single Controller

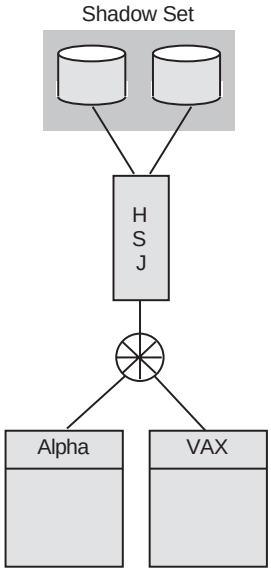
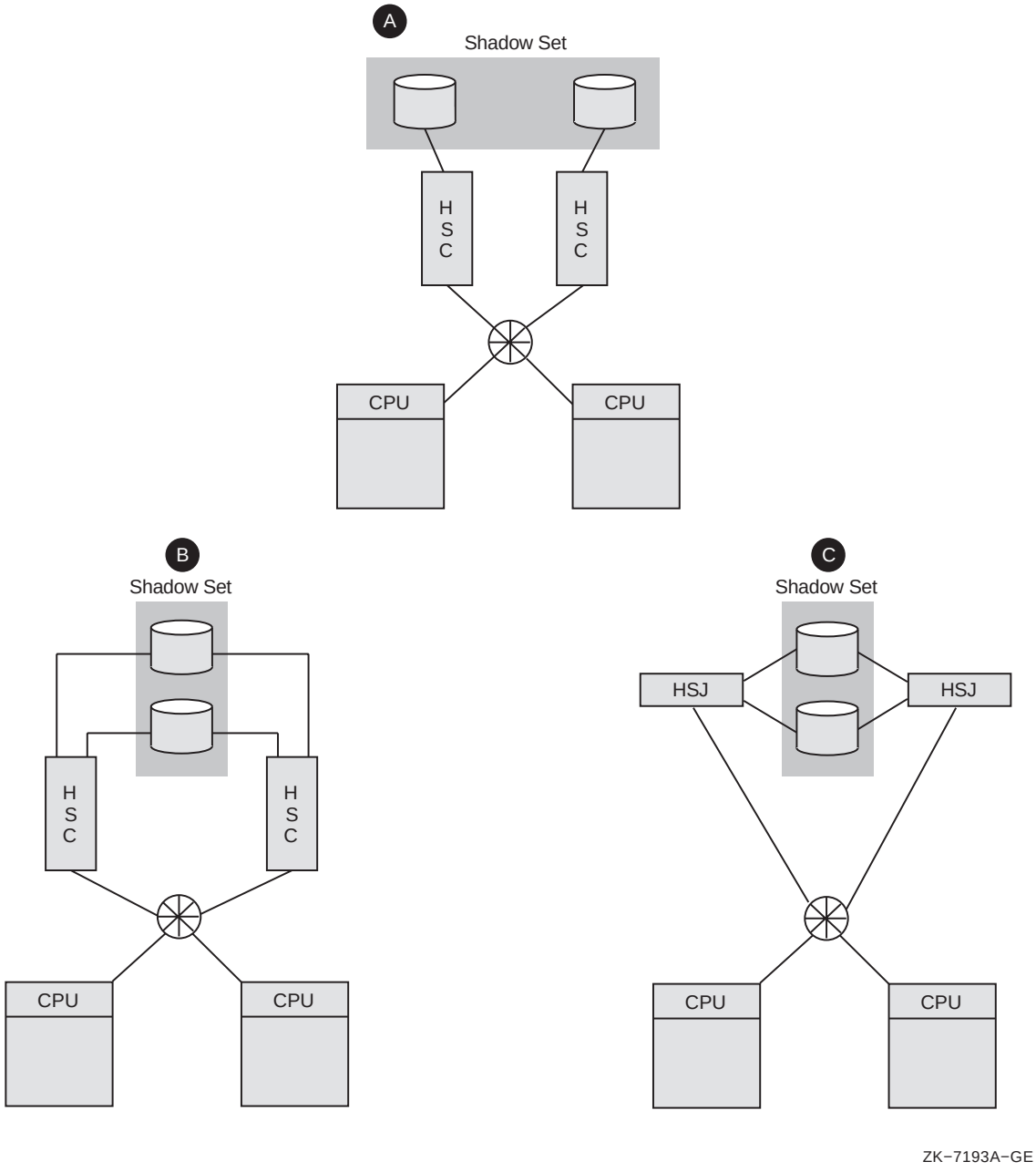


Figure 10–5 shows two nodes connected to an HSJ, with a two-member shadow set.

The disadvantage of this strategy is that the controller is a single point of failure. The configuration in Figure 10–6 shows examples of shadowing across controllers, which prevents one controller from being a single point of failure. Shadowing across HSJ and HSC controllers provides optimal scalability and availability within an OpenVMS Cluster system.

Figure 10–6 Volume Shadowing Across Controllers



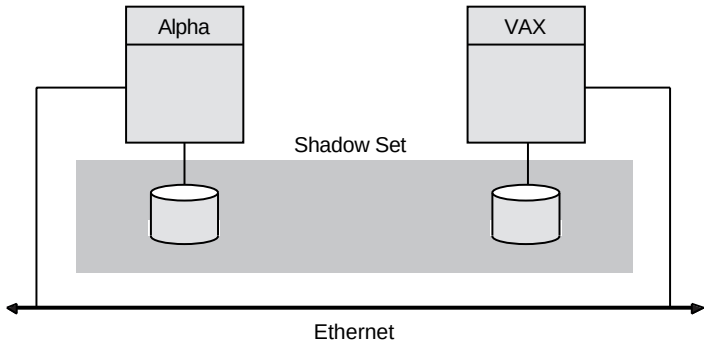
As Figure 10–6 shows, shadowing across controllers has three variations:

- Strategy A shows each volume in the shadow set attached to separate controllers. This configuration is not optimal because each volume is not attached to each controller.
- Strategy B shows dual-ported devices that have two paths to the volumes through separate controllers. This strategy is an optimal variation because two HSC controllers have direct access to a single storage device.

- Strategy C shows HSJ controllers shadowed across SCSI buses. It also is an optimal variation because two HSJ controllers have direct access to a single storage device.

Figure 10–7 shows an example of shadowing across nodes.

Figure 10–7 Volume Shadowing Across Nodes



As Figure 10–7 shows, shadowing across nodes provides the advantage of flexibility in distance. However, it requires MSCP server overhead for write I/Os. In addition, the failure of one of the nodes and its subsequent return to the OpenVMS Cluster will cause a copy operation.

If you have multiple volumes, shadowing inside a controller and shadowing across controllers are more effective than shadowing across nodes.

Reference: See *Volume Shadowing for OpenVMS* for more information.

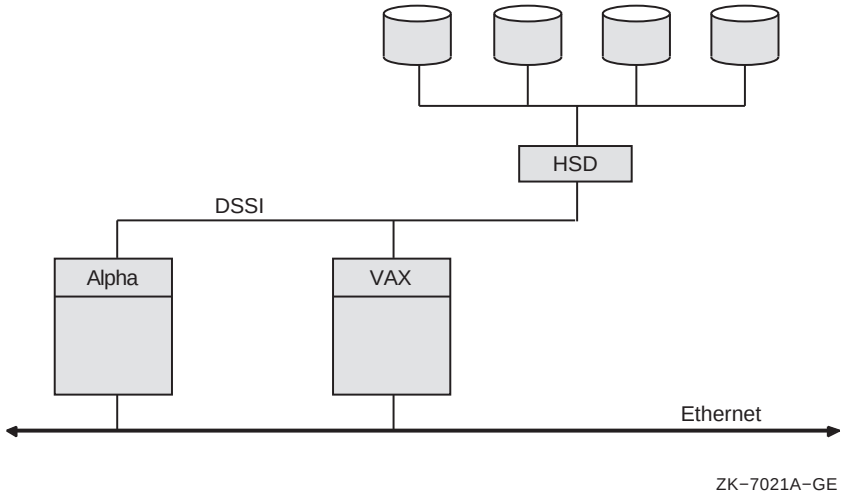
10.4 Scalability in DSSI OpenVMS Clusters

Each DSSI interconnect can have up to eight nodes attached; four can be systems and the rest can be storage devices. Figure 10–8, Figure 10–9, and Figure 10–10 show a progression from a two-node DSSI OpenVMS Cluster to a four-node DSSI OpenVMS Cluster.

10.4.1 Two-Node DSSI OpenVMS Cluster

In Figure 10–8, two nodes are connected to four disks by a common DSSI interconnect.

Figure 10–8 Two-Node DSSI OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–8 include:

Advantages

- Both nodes have shared, direct access to all storage.
- The Ethernet LAN ensures failover capability if the DSSI interconnect fails.

Disadvantages

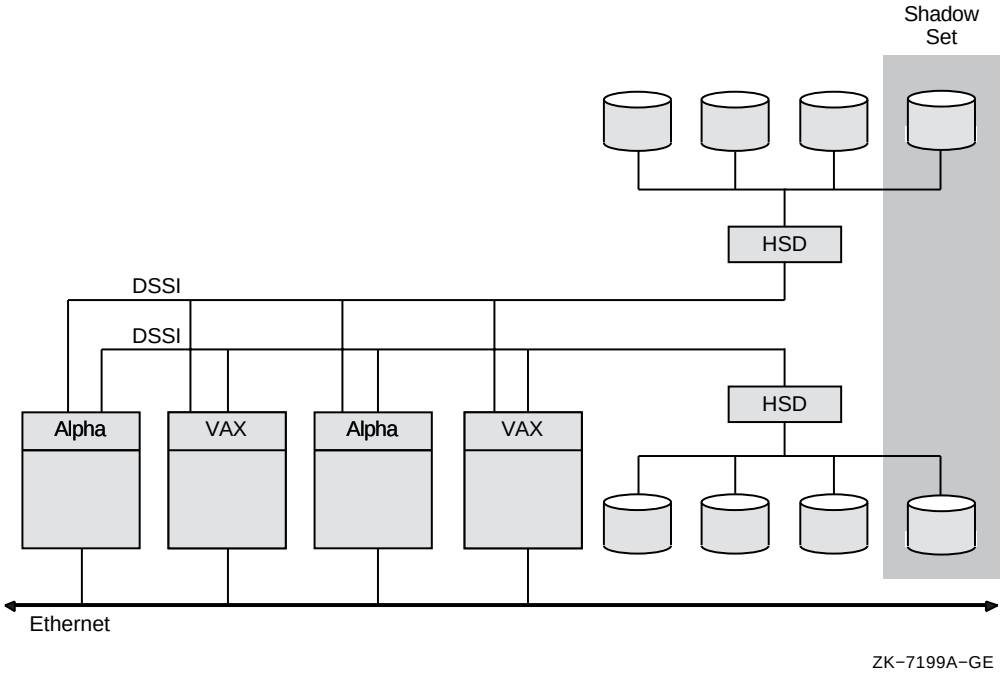
- The amount of storage that is directly accessible to all nodes is limited.
- A single DSSI interconnect can become a single point of failure.

If the OpenVMS Cluster in Figure 10–8 required more processing power, more storage, and better redundancy, this could lead to a configuration like the one shown in Figure 10–9.

10.4.2 Four-Node DSSI OpenVMS Cluster with Shared Access

In Figure 10–9, four nodes have shared, direct access to eight disks through two DSSI interconnects. Two of the disks are shadowed across DSSI interconnects.

Figure 10–9 Four-Node DSSI OpenVMS Cluster with Shared Access



The advantages and disadvantages of the configuration shown in Figure 10–9 include:

Advantages

- All nodes have shared, direct access to all storage.
- The Ethernet LAN ensures failover capability if the DSSI interconnect fails.
- Shadowing across DSSI interconnects provides increased performance and availability.

Disadvantage

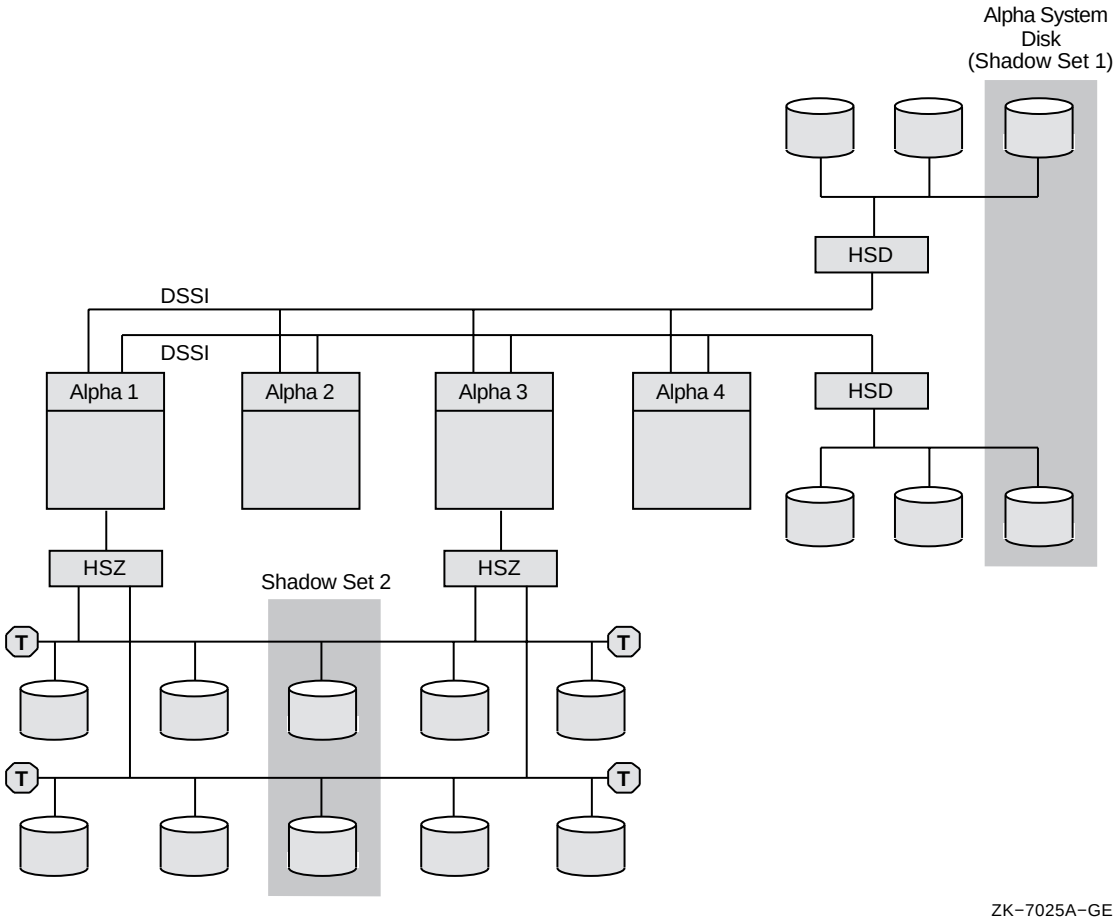
- The amount of storage that is directly accessible to all nodes is limited.

If the configuration in Figure 10–9 required more storage, this could lead to a configuration like the one shown in Figure 10–10.

10.4.3 Four-Node DSSI OpenVMS Cluster with Some Nonshared Access

Figure 10–10 shows an OpenVMS Cluster with 4 nodes and 10 disks. This model differs from Figure 10–8 and Figure 10–9 in that some of the nodes do not have shared, direct access to some of the disks, thus requiring these disks to MSCP served. For the best performance, place your highest-priority data on disks that are directly connected by common DSSI interconnects to your nodes. Volume shadowing across common DSSI interconnects provides the highest availability and may increase read performance.

Figure 10–10 DSSI OpenVMS Cluster with 10 Disks



The advantages and disadvantages of the configuration shown in Figure 10–10 include:

Advantages

- All nodes have shared, direct access to most of the storage.
- The MSCP server is enabled to allow failover to the alternate DSSI interconnect if one of the DSSI interconnects fails.
- Shadowing across DSSI interconnects provides increased performance and availability.
- The SCSI storage connected through the HSZ controllers provides good performance and scalability.

Disadvantages

- The amount of storage that is directly accessible to all nodes is limited.
- Shadow set 2 requires MSCP serving to coordinate shadowing activity.
- Some nodes do not have direct access to storage. For example, Alpha 2 and Alpha 4 do not have direct access to disks connected to Alpha 1 and Alpha 3.

10.5 Scalability in MEMORY CHANNEL OpenVMS Clusters

Each MEMORY CHANNEL (MC) interconnect can have up to four nodes attached to each MEMORY CHANNEL hub. For two-hub configurations, each node must have two PCI adapters, and each adapter must be attached to a different hub. In a two-node configuration, no hub is required because one of the PCI adapters serves as a virtual hub.

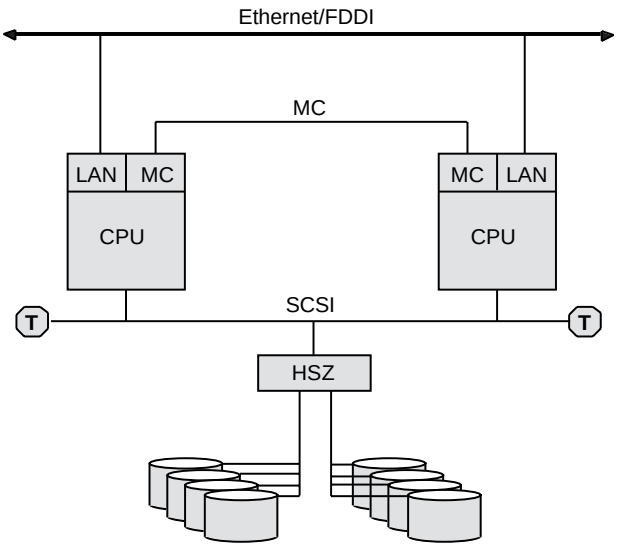
Figure 10–11, Figure 10–12, and Figure 10–13 show a progression from a two-node MEMORY CHANNEL cluster to a four-node MEMORY CHANNEL cluster.

Reference: For additional configuration information and a more detailed technical summary of how MEMORY CHANNEL works, see Appendix B.

10.5.1 Two-Node MEMORY CHANNEL Cluster

In Figure 10–11, two nodes are connected by a MEMORY CHANNEL interconnect, a LAN (Ethernet, FDDI, or ATM) interconnect, and a SCSI interconnect.

Figure 10–11 Two-Node MEMORY CHANNEL OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–11 include:

Advantages

- Both nodes have shared, direct access to all storage.
- The Ethernet/FDDI/ATM interconnect enables failover if the MEMORY CHANNEL interconnect fails.
- The limit of two MEMORY CHANNEL nodes means that no hub is required; one PCI adapter serves as a virtual hub.

Disadvantages

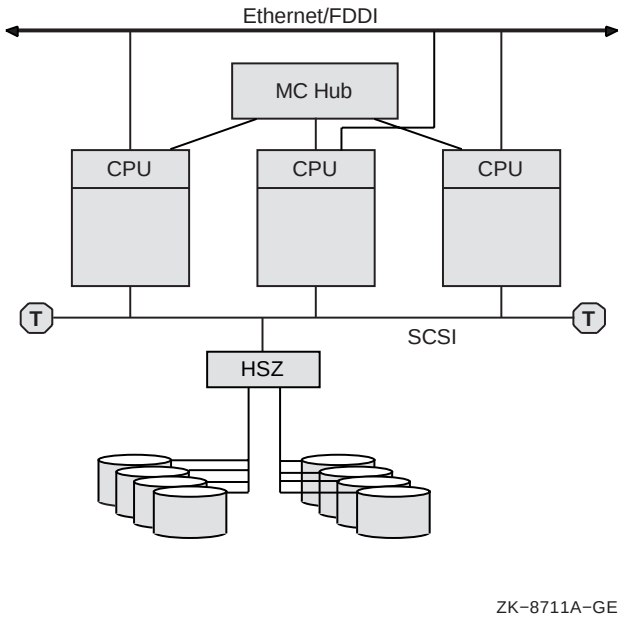
- The amount of storage that is directly accessible to all nodes is limited.
- A single SCSI interconnect or HSZ controller can become a single point of failure.

If the OpenVMS Cluster in Figure 10–11 required more processing power and better redundancy, this could lead to a configuration like the one shown in Figure 10–12.

10.5.2 Three-Node MEMORY CHANNEL Cluster

In Figure 10–12, three nodes are connected by a high-speed MEMORY CHANNEL interconnect, as well as by a LAN (Ethernet, FDDI, or ATM) interconnect. These nodes also have shared, direct access to storage through the SCSI interconnect.

Figure 10–12 Three-Node MEMORY CHANNEL OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–12 include:

Advantages

- All nodes have shared, direct access to storage.
- The Ethernet/FDDI/ATM interconnect enables failover if the MEMORY CHANNEL interconnect fails.
- The addition of a MEMORY CHANNEL hub increases the limit on the number of nodes to a total of four.

Disadvantage

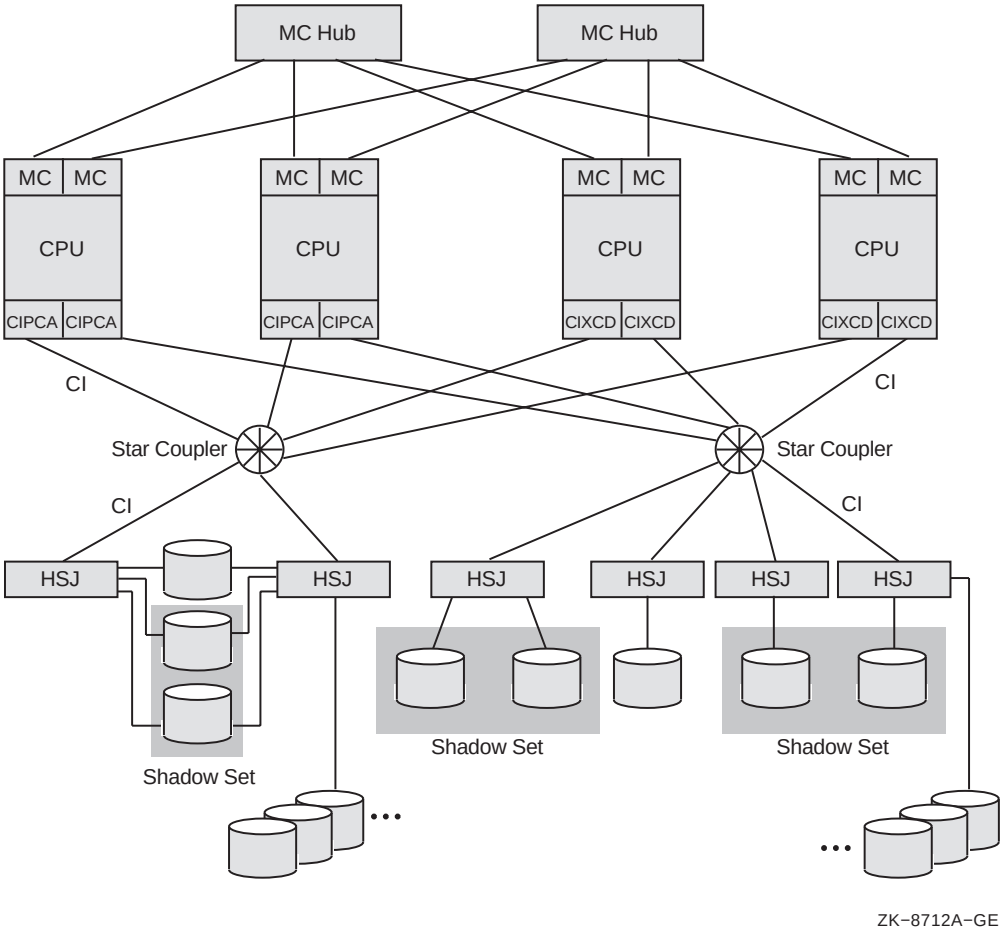
- The amount of storage that is directly accessible to all nodes is limited.

If the configuration in Figure 10–12 required more storage, this could lead to a configuration like the one shown in Figure 10–13.

10.5.3 Four-Node MEMORY CHANNEL OpenVMS Cluster

Figure 10–13, each node is connected by a MEMORY CHANNEL interconnect as well as by a CI interconnect.

Figure 10–13 MEMORY CHANNEL Cluster with a CI Cluster



The advantages and disadvantages of the configuration shown in Figure 10–13 include:

Advantages

- All nodes have shared, direct access to all of the storage.
- This configuration has more than double the storage and processing capacity as the one shown in Figure 10–12.
- If the MEMORY CHANNEL interconnect fails, the CI can take over internode communication.
- The CIPCA adapters on two of the nodes enable the addition of Alpha systems to a CI cluster that formerly comprised VAX (CIXCD-based) systems.
- Multiple CIs between processors and storage provide twice the performance of one path. Bandwidth further increases because MEMORY CHANNEL offloads internode traffic from the CI, enabling the CI to be devoted only to storage traffic. This improves the performance of the entire cluster.

- Volume shadowed, dual-ported disks increase data availability.

Disadvantage

- This configuration is complex and requires the care of an experienced system manager.

10.6 Scalability in SCSI OpenVMS Clusters

SCSI-based OpenVMS Clusters allow commodity-priced storage devices to be used directly in OpenVMS Clusters. Using a SCSI interconnect in an OpenVMS Cluster offers you variations in distance, price, and performance capacity. This SCSI clustering capability is an ideal starting point when configuring a low-end, affordable cluster solution. SCSI clusters can range from desktop to deskside to departmental and larger configurations.

Note the following general limitations when using the SCSI interconnect:

- Because the SCSI interconnect handles only storage traffic, it must always be paired with another interconnect for node-to-node traffic. In the figures shown in this section, MEMORY CHANNEL is the alternate interconnect; but CI, DSSI, Ethernet, and FDDI could also be used.
- Total SCSI cable lengths must take into account the system's internal cable length. For example, an AlphaServer 1000 rackmount uses 1.6 m of internal cable to connect the internal adapter to the external connector. Two AlphaServer 1000s joined by a 2 m SCSI cable would use 1.6 m within each system, resulting in a total SCSI bus length of 5.2 m.

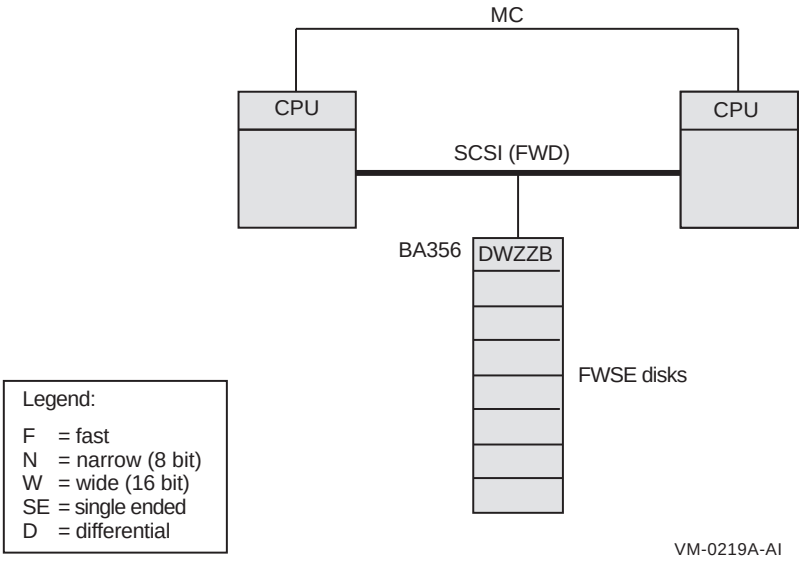
  Reference: For more information about internal SCSI cable lengths as well as highly detailed information about clustering SCSI devices, see Appendix A.

The figures in this section show a progression from a two-node SCSI configuration with modest storage to a four-node SCSI hub configuration with maximum storage and further expansion capability.

10.6.1 Two-Node Fast-Wide SCSI Cluster

In Figure 10–14, two nodes are connected by a 25-m, fast-wide differential (FWD) SCSI bus, with MEMORY CHANNEL (or any) interconnect for internode traffic. The BA356 storage cabinet contains a power supply, a DWZZB single-ended to differential converter, and six disk drives. This configuration can have either narrow or wide disks.

Figure 10–14 Two-Node Fast-Wide SCSI Cluster



The advantages and disadvantages of the configuration shown in Figure 10–14 include:

Advantages

- Low cost SCSI storage is shared by two nodes.
- With the BA356 cabinet, you can use a narrow (8 bit) or wide (16 bit) SCSI bus.
- The DWZZB converts single-ended signals to differential.
- The fast-wide SCSI interconnect provides 20 MB/s performance.
- MEMORY CHANNEL handles internode traffic.
- The differential SCSI bus can be 25 m.

Disadvantage

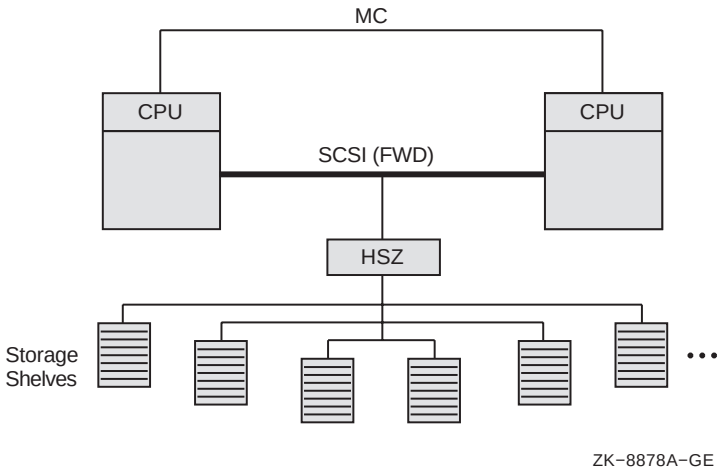
- Somewhat limited storage capability.

If the configuration in Figure 10–14 required even more storage, this could lead to a configuration like the one shown in Figure 10–15.

10.6.2 Two-Node Fast-Wide SCSI Cluster with HSZ Storage

In Figure 10–15, two nodes are connected by a 25-m, fast-wide differential (FWD) SCSI bus, with MEMORY CHANNEL (or any) interconnect for internode traffic. Multiple storage shelves are within the HSZ controller.

Figure 10–15 Two-Node Fast-Wide SCSI Cluster with HSZ Storage



The advantages and disadvantages of the configuration shown in Figure 10–15 include:

Advantages

- Costs slightly more than the configuration shown in Figure 10–14, but offers significantly more storage. (The HSZ controller enables you to add more storage.)
- Cache in the HSZ, which also provides RAID 0, 1, and 5 technologies. The HSZ is a differential device; no converter is needed.
- MEMORY CHANNEL handles internode traffic.
- The FWD bus provides 20 MB/s throughput.
- Includes a 25 m differential SCSI bus.

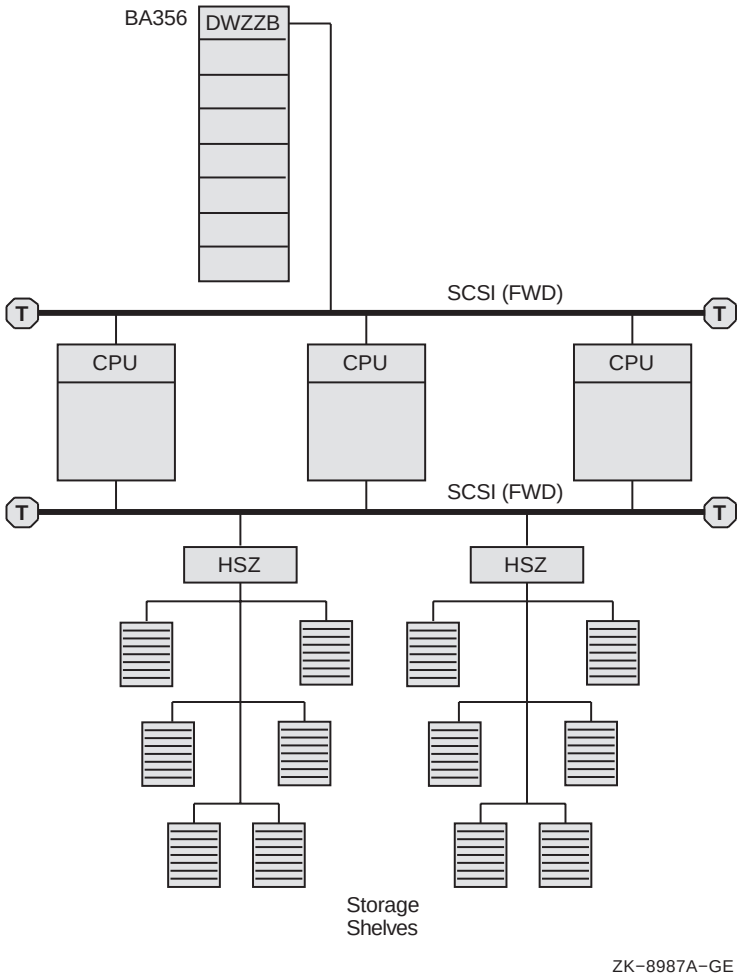
Disadvantage

- This configuration is more expensive than the one shown in Figure 10–14.

10.6.3 Three-Node Fast-Wide SCSI Cluster

In Figure 10–16, three nodes are connected by two 25-m, fast-wide (FWD) SCSI interconnects. Multiple storage shelves are contained in each HSZ controller, and more storage is contained in the BA356 at the top of the figure.

Figure 10–16 Three-Node Fast-Wide SCSI Cluster



The advantages and disadvantages of the configuration shown in Figure 10–16 include:

Advantages

- Combines the advantages of the configurations shown in Figure 10–14 and Figure 10–15:
  - Significant (25 m) bus distance and scalability.
  - Includes cache in the HSZ, which also provides RAID 0, 1, and 5 technologies. The HSZ contains multiple storage shelves.
  - FWD bus provides 20 MB/s throughput.
  - With the BA356 cabinet, you can use narrow (8 bit) or wide (16 bit) SCSI bus.

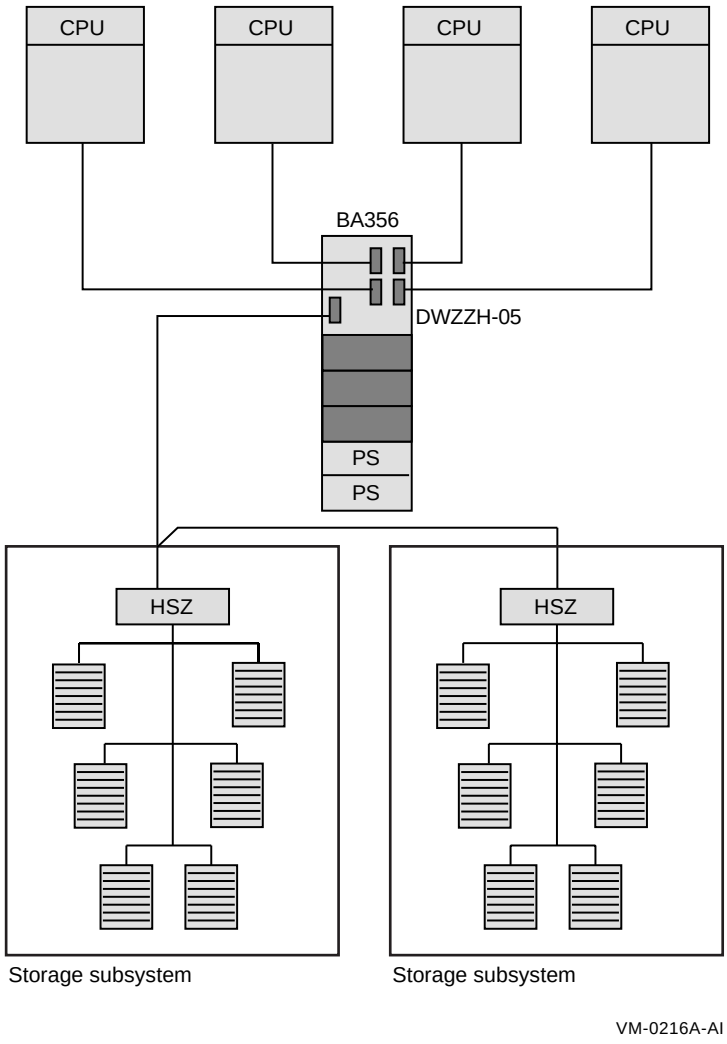
Disadvantage

- This configuration is more expensive than those shown in previous figures.

10.6.4 Four-Node Ultra SCSI Hub Configuration

Figure 10–17 shows four nodes connected by a SCSI hub. The SCSI hub obtains power and cooling from the storage cabinet, such as the BA356. The SCSI hub does not connect to the SCSI bus of the storage cabinet.

Figure 10–17 Four-Node Ultra SCSI Hub Configuration



The advantages and disadvantages of the configuration shown in Figure 10–17 include:

Advantages

- Provides significantly more bus distance and scalability than the configuration shown in Figure 10–15.
- The SCSI hub provides fair arbitration on the SCSI bus. This provides more uniform, predictable system behavior. Four CPUs are allowed only when fair arbitration is enabled.
- Up to two dual HSZ controllers can be daisy-chained to the storage port of the hub.

- Two power supplies in the BA356 (one for backup).
- Cache in the HSZs, which also provides RAID 0, 1, and 5 technologies.
- Ultra SCSI bus provides 40 MB/s throughput.

Disadvantage

- You cannot add CPUs to this configuration by daisy-chaining a SCSI interconnect from a CPU or HSZ to another CPU.
- This configuration is more expensive than those shown in Figure 10–14 and Figure 10–15.
- Only HSZ storage can be connected. You cannot attach a storage shelf with disk drives directly to the SCSI hub.

10.7 Scalability in OpenVMS Clusters with Satellites

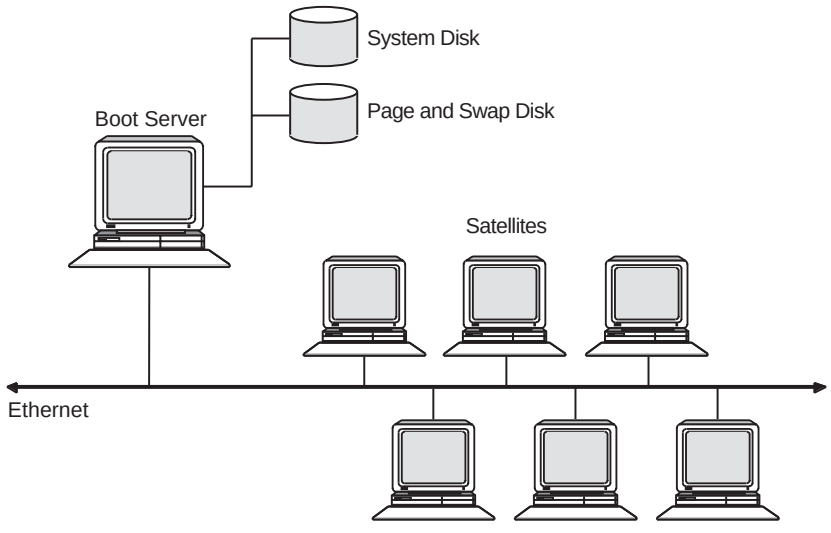
The number of satellites in an OpenVMS Cluster and the amount of storage that is MSCP served determine the need for the quantity and capacity of the servers. Satellites are systems that do not have direct access to a system disk and other OpenVMS Cluster storage. Satellites are usually workstations, but they can be any OpenVMS Cluster node that is served storage by other nodes in the OpenVMS Cluster.

Each Ethernet LAN segment should have only 10 to 20 satellite nodes attached. Figure 10–18, Figure 10–19, Figure 10–20, and Figure 10–21 show a progression from a 6-satellite LAN to a 45-satellite LAN.

10.7.1 Six-Satellite OpenVMS Cluster

In Figure 10–18, six satellites and a boot server are connected by Ethernet.

Figure 10–18 Six-Satellite LAN OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 10–18 include:

Advantages

- The MSCP server is enabled for adding satellites and allows access to more storage.
- With one system disk, system management is relatively simple.

  Reference: For information about managing system disks, see Section 11.2.

Disadvantage

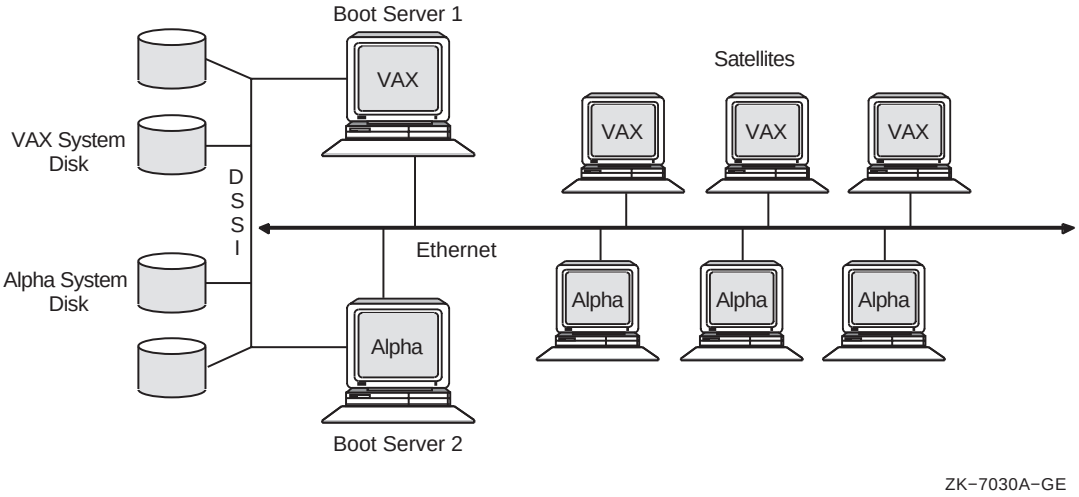
- The Ethernet is a potential bottleneck and a single point of failure.

If the boot server in Figure 10–18 became a bottleneck, a configuration like the one shown in Figure 10–19 would be required.

10.7.2 Six-Satellite OpenVMS Cluster with Two Boot Nodes

Figure 10–19 shows six satellites and two boot servers connected by Ethernet. Boot server 1 and boot server 2 perform MSCP server dynamic load balancing: they arbitrate and share the work load between them and if one node stops functioning, the other takes over. MSCP dynamic load balancing requires shared access to storage.

Figure 10–19 Six-Satellite LAN OpenVMS Cluster with Two Boot Nodes



The advantages and disadvantages of the configuration shown in Figure 10–19 include:

Advantages

- The MSCP server is enabled for adding satellites and allows access to more storage.
- Two boot servers perform MSCP dynamic load balancing.

Disadvantage

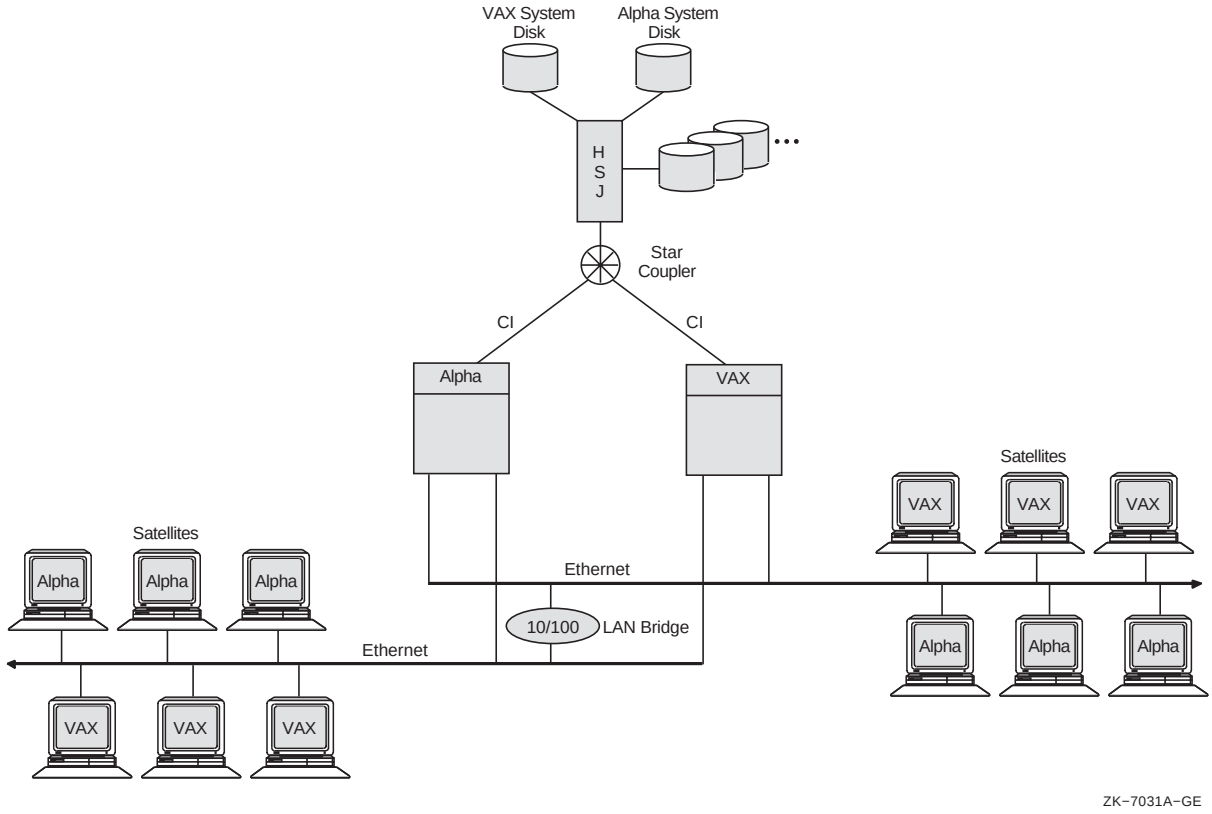
- The Ethernet is a potential bottleneck and a single point of failure.

If the LAN in Figure 10–19 became an OpenVMS Cluster bottleneck, this could lead to a configuration like the one shown in Figure 10–20.

10.7.3 Twelve-Satellite LAN OpenVMS Cluster with Two LAN Segments

Figure 10–20 shows 12 satellites and 2 boot servers connected by two Ethernet segments. These two Ethernet segments are also joined by a LAN bridge. Because each satellite has dual paths to storage, this configuration also features MSCP dynamic load balancing.

Figure 10–20 Twelve-Satellite OpenVMS Cluster with Two LAN Segments



The advantages and disadvantages of the configuration shown in Figure 10–20 include:

Advantages

- The MSCP server is enabled for adding satellites and allows access to more storage.
- Two boot servers perform MSCP dynamic load balancing.

  From the perspective of a satellite on the Ethernet LAN, the dual paths to the Alpha and VAX nodes create the advantage of MSCP load balancing.
- Two LAN segments provide twice the amount of LAN capacity.

Disadvantages

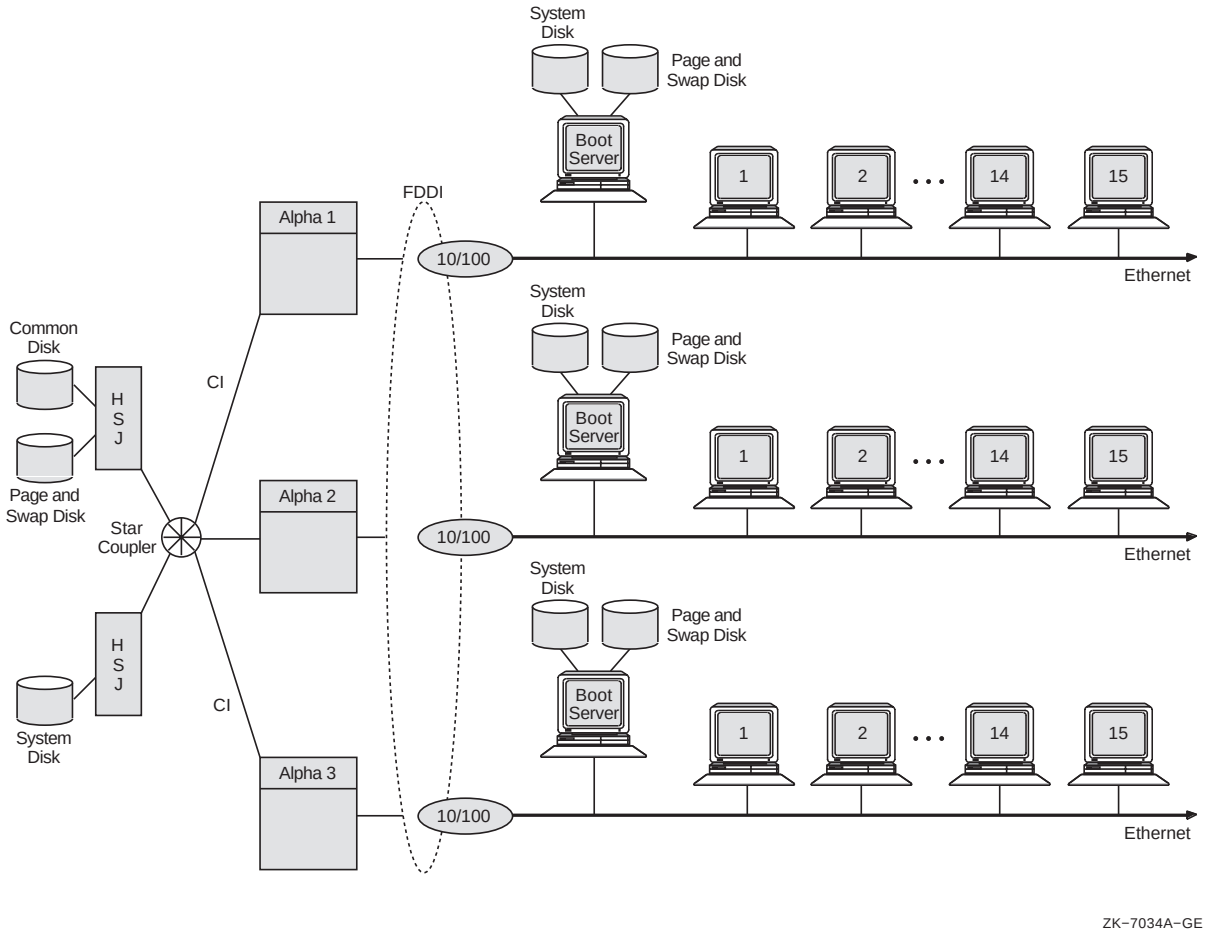
- This OpenVMS Cluster configuration is limited by the number of satellites that it can support.
- The single HSJ controller is a potential bottleneck and a single point of failure.

If the OpenVMS Cluster in Figure 10–20 needed to grow beyond its current limits, this could lead to a configuration like the one shown in Figure 10–21.

10.7.4 Forty-Five Satellite OpenVMS Cluster with FDDI Ring

Figure 10–21 shows a large, 51-node OpenVMS Cluster. The three boot servers, Alpha 1, Alpha 2, and Alpha 3, have two disks: a common disk and a system disk. The FDDI ring has three LAN segments attached. Each segment has 15 workstation satellites as well as its own boot node.

Figure 10–21 Forty-Five Satellite OpenVMS Cluster with FDDI Ring



The advantages and disadvantages of the configuration shown in Figure 10–21 include:

Advantages

- Decreased boot time, especially for an OpenVMS Cluster with such a high node count.

  Reference: For information about booting an OpenVMS Cluster like the one in Figure 10–21 see Section 11.2.4.

- The MSCP server is enabled for satellites to access more storage.
- Each boot server has its own page and swap disk, which reduces I/O activity on the system disks.

- All of the environment files for the entire OpenVMS Cluster are on the common disk. This frees the satellite boot servers to serve only root information to the satellites.

  Reference: For more information about common disks and page and swap disks, see Section 11.2.

- The FDDI ring provides 10 times the capacity of one Ethernet interconnect.

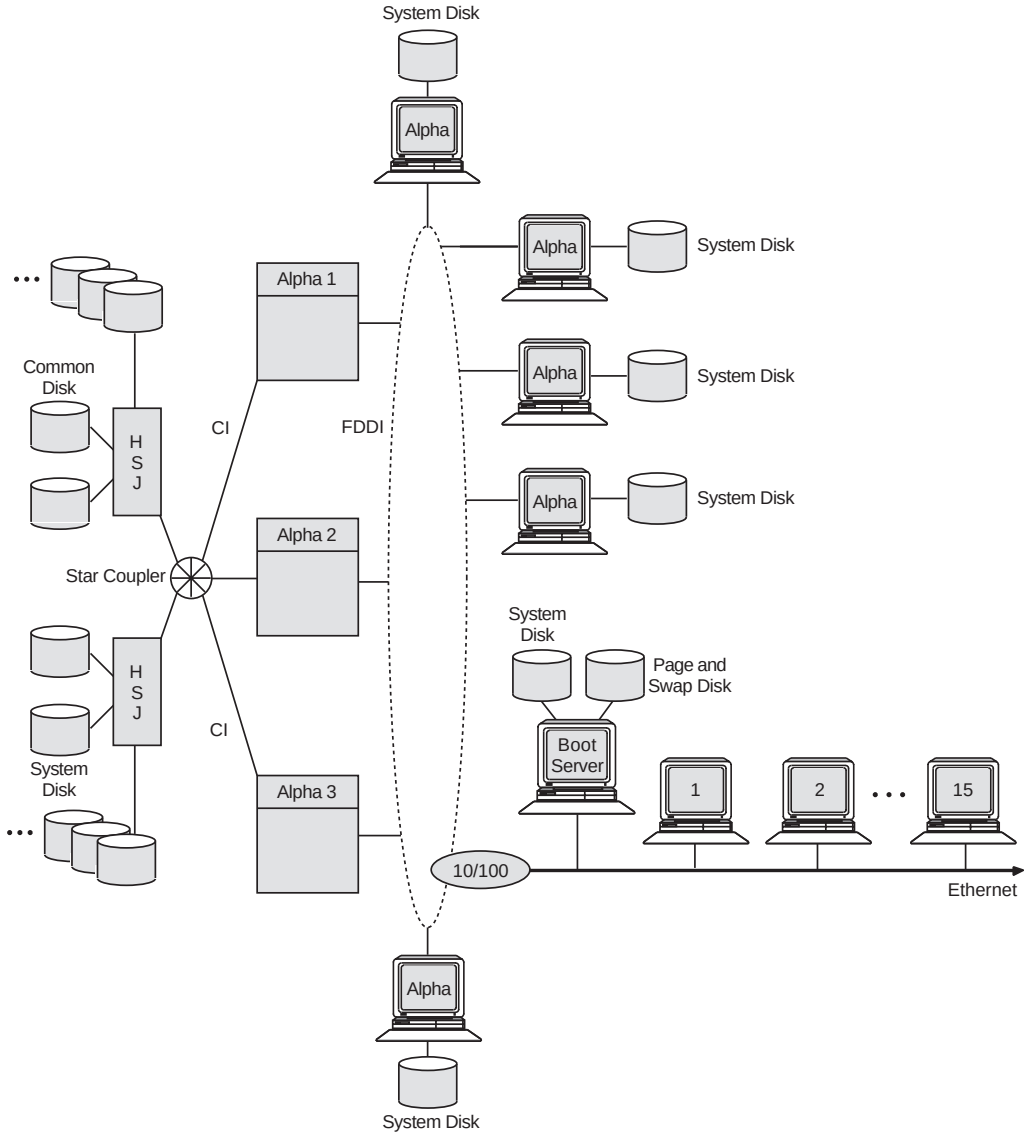
Disadvantages

- The satellite boot servers on the Ethernet LAN segments can boot satellites only on their own segments.

10.7.5 High-Powered Workstation OpenVMS Cluster

Figure 10–22 shows an OpenVMS Cluster configuration that provides high performance and high availability on the FDDI ring.

Figure 10–22 High-Powered Workstation Server Configuration



In Figure 10–22, several Alpha workstations, each with its own system disk, are connected to the FDDI ring. Putting Alpha workstations on the FDDI provides high performance because each workstation has direct access to its system disk. In addition, the FDDI bandwidth is higher than that of the Ethernet. Because Alpha workstations have FDDI adapters, putting these workstations on an FDDI is a useful alternative for critical workstation requirements. FDDI is 10 times faster than Ethernet, and Alpha workstations have processing capacity that can take advantage of FDDI's speed.

10.7.6 Guidelines for OpenVMS Clusters with Satellites

The following are guidelines for setting up an OpenVMS Cluster with satellites:

- Extra memory is required for satellites of large LAN configurations because each node must maintain a connection to every other node.
- Place only 10 to 20 satellites on each LAN segment.
- Maximize resources with MSCP dynamic load balancing, as shown in Figure 10–19 and Figure 10–20.
- Keep the number of nodes that require MSCP serving minimal for good performance.

  Reference: See Section 10.8.1 for more information about MSCP overhead.
- To save time, ensure that the booting sequence is efficient, particularly when the OpenVMS Cluster is large or has multiple segments. See Section 11.2.4 for more information about how to reduce LAN and system disk activity and how to boot separate groups of nodes in sequence.
- Use two or more LAN adapters per host (up to four adapters are supported for OpenVMS Cluster communications), and connect to independent LAN paths. This enables simultaneous two-way communication between nodes and allows traffic to multiple nodes to be spread over the available LANs.

10.7.7 Extended LAN Configuration Guidelines

You can use bridges between LAN segments to form an extended LAN (ELAN). This can increase availability, distance, and aggregate bandwidth as compared with a single LAN. However, an ELAN can increase delay and can reduce bandwidth on some paths. Factors such as packet loss, queuing delays, and packet size can also affect ELAN performance. Table 10–3 provides guidelines for ensuring adequate LAN performance when dealing with such factors.

Table 10–3 ELAN Configuration Guidelines

Factor	Guidelines
Propagation delay	The amount of time it takes a packet to traverse the ELAN depends on the distance it travels and the number of times it is relayed from one link to another by a bridge or a station on the FDDI ring. If responsiveness is critical, then you must control these factors. When an FDDI is used for OpenVMS Cluster communications, the ring latency when the FDDI ring is idle should not exceed 400 ms. FDDI packets travel at 5.085 microseconds/km and each station causes an approximate 1-ms delay between receiving and transmitting. You can calculate FDDI latency by using the following algorithm: $Latency = (distance\ in\ km) * (5.085\ ms/km) + (number\ of\ stations) * (1\ ms/station)$ For high-performance applications, limit the number of bridges between nodes to two. For situations in which high performance is not required, you can use up to seven bridges between nodes.

(continued on next page)

Table 10–3 (Cont.) ELAN Configuration Guidelines

Factor	Guidelines
Queuing delay	Queuing occurs when the instantaneous arrival rate at bridges and host adapters exceeds the service rate. You can control queuing by: • Reducing the number of bridges between nodes that communicate frequently. • Using only high-performance bridges and adapters. • Reducing traffic bursts in the LAN. In some cases, for example, you can tune applications by combining small I/Os so that a single packet is produced rather than a burst of small ones. • Reducing LAN segment and host processor utilization levels by using faster processors and faster LANs, and by using bridges for traffic isolation.
Packet loss	Packets that are not delivered by the ELAN require retransmission, which wastes network resources, increases delay, and reduces bandwidth. Bridges and adapters discard packets when they become congested. You can reduce packet loss by controlling queuing, as previously described. Packets are also discarded when they become damaged in transit. You can control this problem by observing LAN hardware configuration rules, removing sources of electrical interference, and ensuring that all hardware is operating correctly. Packet loss can also be reduced by using VMS Version 5.5–2 or later, which has PEDRIVER congestion control. The retransmission timeout rate, which is a symptom of packet loss, must be less than 1 timeout in 1000 transmissions for OpenVMS Cluster traffic from one node to another. ELAN paths that are used for high-performance applications should have a significantly lower rate. Monitor the occurrence of retransmission timeouts in the OpenVMS Cluster. **Reference:** For information about monitoring the occurrence of retransmission timeouts, see *OpenVMS Cluster Systems*.
Bridge recovery delay	Choose bridges with fast self-test time and adjust bridges for fast automatic reconfiguration. **Reference:** Refer to *OpenVMS Cluster Systems* for more information about LAN bridge failover.
Bandwidth	All LAN paths used for OpenVMS Cluster communication must operate with a nominal bandwidth of at least 10 Mb/s. The average LAN segment utilization should not exceed 60% for any 10-second interval. Use FDDI exclusively on the communication paths that have the highest performance requirements. Do not put an Ethernet LAN segment between two FDDI segments. FDDI bandwidth is significantly greater, and the Ethernet LAN will become a bottleneck. This strategy is especially ineffective if a server on one FDDI must serve clients on another FDDI with an Ethernet LAN between them. A more appropriate strategy is to put a server on an FDDI and put clients on an Ethernet LAN, as Figure 10–21 shows.
Traffic isolation	Use bridges to isolate and localize the traffic between nodes that communicate with each other frequently. For example, use bridges to separate the OpenVMS Cluster from the rest of the ELAN and to separate nodes within an OpenVMS Cluster that communicate frequently from the rest of the OpenVMS Cluster. Provide independent paths through the ELAN between critical systems that have multiple adapters.

(continued on next page)

Table 10–3 (Cont.) ELAN Configuration Guidelines

Factor	Guidelines
Packet size	You can adjust the NISCS_MAX_PKTSZ system parameter to use the full FDDI packet size. Ensure that the ELAN path supports a data field of at least 4474 bytes end to end. Some failures cause traffic to switch from an ELAN path that supports 4474-byte packets to a path that supports only smaller packets. It is possible to implement automatic detection and recovery from these kinds of failures. This capability requires that the ELAN set the value of the priority field in the FDDI frame-control byte to zero when the packet is delivered on the destination FDDI link. Ethernet-to-FDDI bridges that conform to the IEEE 802.1 bridge specification provide this capability.

10.7.8 System Parameters for OpenVMS Clusters

In an OpenVMS Cluster with satellites and servers, specific system parameters can help you manage your OpenVMS Cluster more efficiently. Table 10–4 gives suggested values for these system parameters.

Table 10–4 OpenVMS Cluster System Parameters

System Parameter	Value for Satellites	Value for Servers
LOCKDIRWT	0	1–4
SHADOW_MAX_COPY	0	1–4
MSCP_LOAD	0	1 or 2
NPAGEDYN	Higher than for standalone node	Higher than for satellite node
PAGEDYN	Higher than for standalone node	Higher than for satellite node
VOTES	0	1
EXPECTED_VOTES	Sum of OpenVMS Cluster votes	Sum of OpenVMS Cluster votes
RECNXINTERVL[1]	Equal on all nodes	Equal on all nodes

[1]Correlate with bridge timers and LAN utilization.

Reference: For a more in-depth description of these parameters, see *OpenVMS Cluster Systems*.

10.8 Scaling for I/Os

The ability to scale I/Os is an important factor in the growth of your OpenVMS Cluster. Adding more components to your OpenVMS Cluster requires high I/O throughput so that additional components do not create bottlenecks and decrease the performance of the entire OpenVMS Cluster. Many factors can affect I/O throughput:

- Direct access or MSCP served access to storage
- File system technologies, such as Files–11
- Disk technologies, such as magnetic disks, solid-state disks, and DECram
- Read/write ratio

- I/O size
- Caches and cache “hit” rate
- “Hot file” management
- RAID striping and host-based striping
- Volume shadowing

These factors can affect I/O scalability either singly or in combination. The following sections explain these factors and suggest ways to maximize I/O throughput and scalability without having to change in your application.

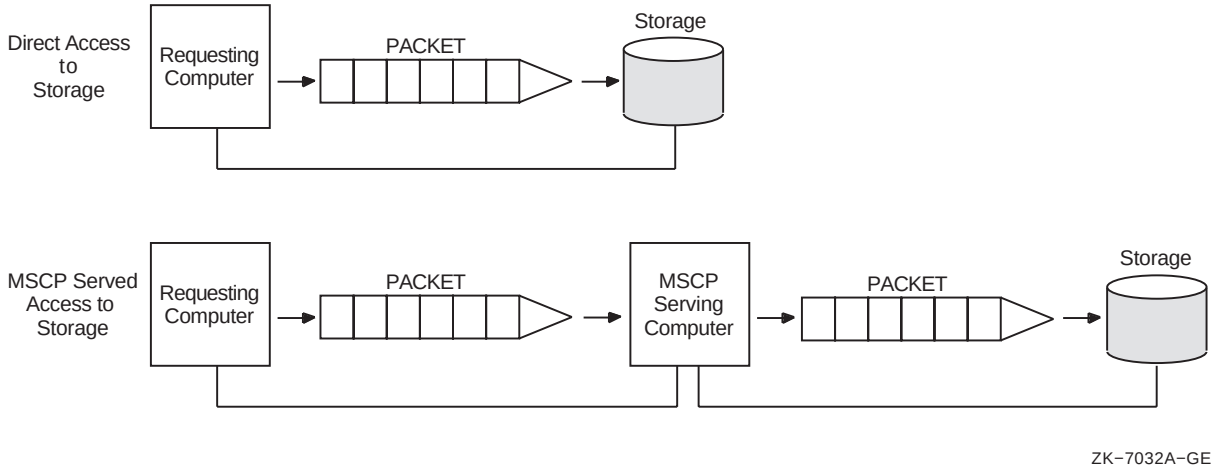
Additional factors that affect I/O throughput are types of interconnects and types of storage subsystems.

Reference: See Chapter 4 for more information about interconnects and Chapter 5 for more information about types of storage subsystems.

10.8.1 MSCP Served Access to Storage

MSCP server capability provides a major benefit to OpenVMS Clusters: it enables communication between nodes and storage that are not directly connected to each other. However, MSCP served I/O does incur overhead. Figure 10–23 is a simplification of how packets require extra handling by the serving system.

Figure 10–23 Comparison of Direct and MSCP Served Access



In Figure 10–23, an MSCP served packet requires an extra “stop” at another system before reaching its destination. When the MSCP served packet reaches the system associated with the target storage, the packet is handled as if for direct access.

In an OpenVMS Cluster that requires a large amount of MSCP serving, I/O performance is not as efficient and scalability is decreased. The total I/O throughput is approximately 20% less when I/O is MSCP served than when it has direct access. Design your configuration so that a few large nodes are serving many satellites rather than satellites serving their local storage to the entire OpenVMS Cluster.

10.8.2 Disk Technologies

In recent years, the ability of CPUs to process information has far outstripped the ability of I/O subsystems to feed processors with data. The result is an increasing percentage of processor time spent waiting for I/O operations to complete.

Solid-state disks (SSDs), DECram, and RAID level 0 bridge this gap between processing speed and magnetic-disk access speed. Performance of magnetic disks is limited by seek and rotational latencies, while SSDs and DECram use memory, which provides nearly instant access.

RAID level 0 is the technique of spreading (or "striping") a single file across several disk volumes. The objective is to reduce or eliminate a bottleneck at a single disk by partitioning heavily accessed files into stripe sets and storing them on multiple devices. This technique increases parallelism across many disks for a single I/O.

Table 10–5 summarizes disk technologies and their features.

Table 10–5 Disk Technology Summary

Disk Technology	Characteristics
Magnetic disk	Slowest access time. Inexpensive. Available on multiple interconnects.
Solid-state disk	Fastest access of any I/O subsystem device. Highest throughput for write-intensive files. Available on multiple interconnects.
DECram	Highest throughput for small to medium I/O requests. Volatile storage; appropriate for temporary read-only files. Available on any Alpha or VAX system.
RAID level 0	Available on HSJ and HSD controllers.

Note: Shared, direct access to a solid-state disk or to DECram is the fastest alternative for scaling I/Os.

10.8.3 Read/Write Ratio

The read/write ratio of your applications is a key factor in scaling I/O to shadow sets. MSCP writes to a shadow set are duplicated on the interconnect.

Therefore, an application that has 100% (100/0) read activity may benefit from volume shadowing because shadowing causes multiple paths to be used for the I/O activity. An application with a 50/50 ratio will cause more interconnect utilization because write activity requires that an I/O be sent to each shadow member. Delays may be caused by the time required to complete the slowest I/O.

To determine I/O read/write ratios, use the DCL command MONITOR IO.

10.8.4 I/O Size

Each I/O packet incurs processor and memory overhead, so grouping I/Os together in one packet decreases overhead for all I/O activity. You can achieve higher throughput if your application is designed to use bigger packets. Smaller packets incur greater overhead.

10.8.5 Caches

Caching is the technique of storing recently or frequently used data in an area where it can be accessed more easily—in memory, in a controller, or in a disk. Caching complements solid-state disks, DECram, and RAID. Applications automatically benefit from the advantages of caching without any special coding. Caching reduces current and potential I/O bottlenecks within OpenVMS Cluster systems by reducing the number of I/Os between components.

Table 10–6 describes the three types of caching.

Table 10–6 Types of Caching

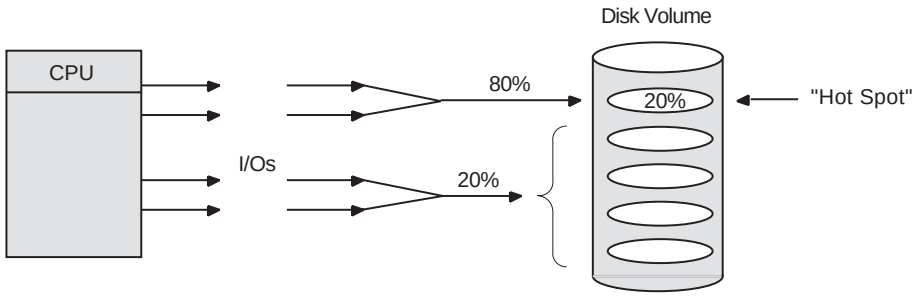
Caching Type	Description
Host based	Cache that is resident in the host system's memory and services I/Os from the host.
Controller based	Cache that is resident in the storage controller and services data for all hosts.
Disk	Cache that is resident in a disk.

Host-based disk caching provides different benefits from controller-based and disk-based caching. In host-based disk caching, the cache itself is not shareable among nodes. Controller-based and disk-based caching are shareable because they are located in the controller or disk, either of which is shareable.

10.8.6 Managing "Hot" Files

A "hot" file is a file in your system on which the most activity occurs. Hot files exist because, in many environments, approximately 80% of all I/O goes to 20% of data. This means that, of equal regions on a disk drive, 80% of the data being transferred goes to one place on a disk, as shown in Figure 10–24.

Figure 10–24 Hot-File Distribution



To increase the scalability of I/Os, focus on hot files, which can become a bottleneck if you do not manage them well. The activity in this area is expressed in I/Os, megabytes transferred, and queue depth.

RAID level 0 balances hot-file activity by spreading a single file over multiple disks. This reduces the performance impact of hot files.

Use the following DCL commands to analyze hot-file activity:

- MONITOR IO command—Monitors hot disks.

- MONITOR MSCP command—Monitors MSCP servers.

The MONITOR IO and the MONITOR MSCP commands enable you to find out which disk and which server are hot.

10.8.7 Volume Shadowing

The Volume Shadowing for OpenVMS product ensures that data is available to applications and end users by duplicating data on multiple disks. Although volume shadowing provides data redundancy and high availability, it can affect OpenVMS Cluster I/O on two levels:

Factor	Effect
Geographic distance	Host-based volume shadowing enables shadowing of any disk volumes in an OpenVMS Cluster system, including those served by MSCP servers. This ability can allow great distances along with MSCP overhead. For example, OpenVMS Cluster systems using FDDI can be located up to 25 miles apart. Both the distance and the MSCP involvement can slow I/O throughput.
Read/write ratio	Because shadowing writes data to multiple volumes, applications that are write intensive may experience reduced throughput. In contrast, read-intensive applications may experience increased throughput because the shadowing software selects one disk member from which it can retrieve the data most efficiently.

11

OpenVMS Cluster System Management Strategies

This chapter suggests some key system management strategies that you can use to get the most out of your OpenVMS Cluster. It is not intended to be a comprehensive discussion of the most common OpenVMS Cluster system management practices; see *OpenVMS Cluster Systems* for that information.

This chapter also assumes that the reader has some familiarity with basic system management concepts, such as system disks, quorum disks, and OpenVMS Cluster transitions.

The following information is contained in this chapter:

- System disk strategies
- Common and multiple environment strategies
- Quorum strategies
- State transition strategies
- Multiple OpenVMS versions in the same OpenVMS Cluster
- Alpha and VAX systems in the same OpenVMS Cluster
- Backup and storage management strategies

11.1 Simple and Complex Configurations

OpenVMS Cluster software makes a system manager's job easier because many system management tasks need to be done only once. This is especially true if business requirements call for a simple configuration rather than for every feature that an OpenVMS Cluster can provide. The simple configuration is appealing to both new and experienced system managers and is applicable to small OpenVMS Clusters—those with 3 to 7 nodes, 20 to 30 users, and 100 GB of storage.

Reference: See Figure 11–1 for an example of a simple OpenVMS Cluster configuration.

More complex OpenVMS Cluster configurations may require a more sophisticated system management strategy to deliver more availability, scalability, and performance.

Reference: See Figure 11–3 for an example of a complex OpenVMS Cluster configuration.

Choose system management strategies that balance simplicity of system management with the additional management tasks required by more complex OpenVMS Clusters.

11.2 System Disk Strategies

System disks contain system files and environment files.

System files are primarily read-only images and command procedures, such as run-time libraries, and are accessed clusterwide.

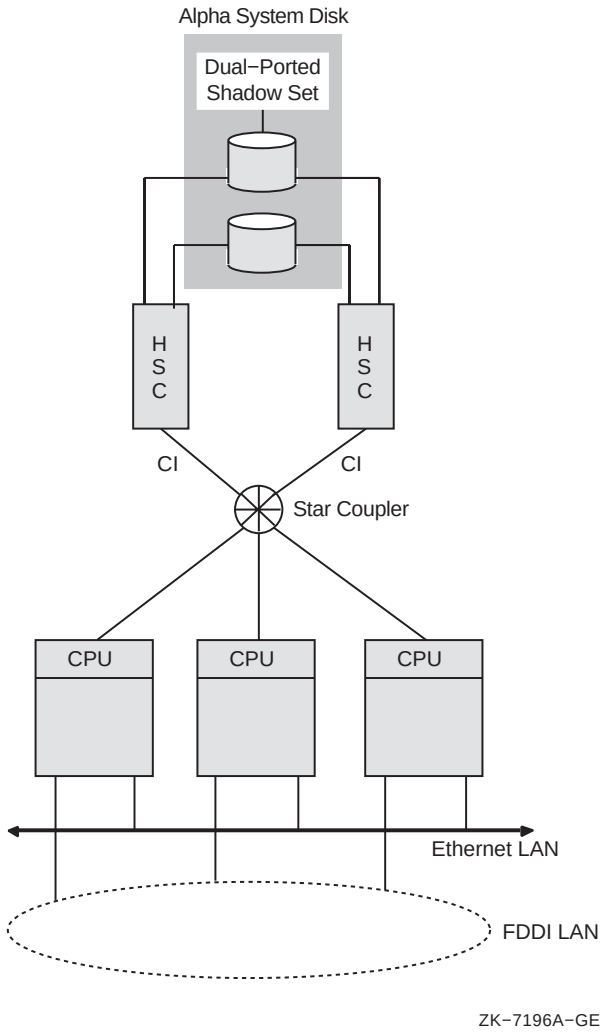
Environment files create specific working environments for users. You can create a common environment by making all environment files accessible clusterwide, or you can create multiple environments by making specific environment files accessible to only certain users or systems.

11.2.1 Single System Disk

System management is easiest for a simple configuration that has a single system disk and a common environment. Most procedures need to be performed only once, and both system files and environment files are located on the same disk. Page and swap files are also located on the system disk.

Figure 11–1 shows an example of a simple OpenVMS Cluster with a single system disk and a common environment.

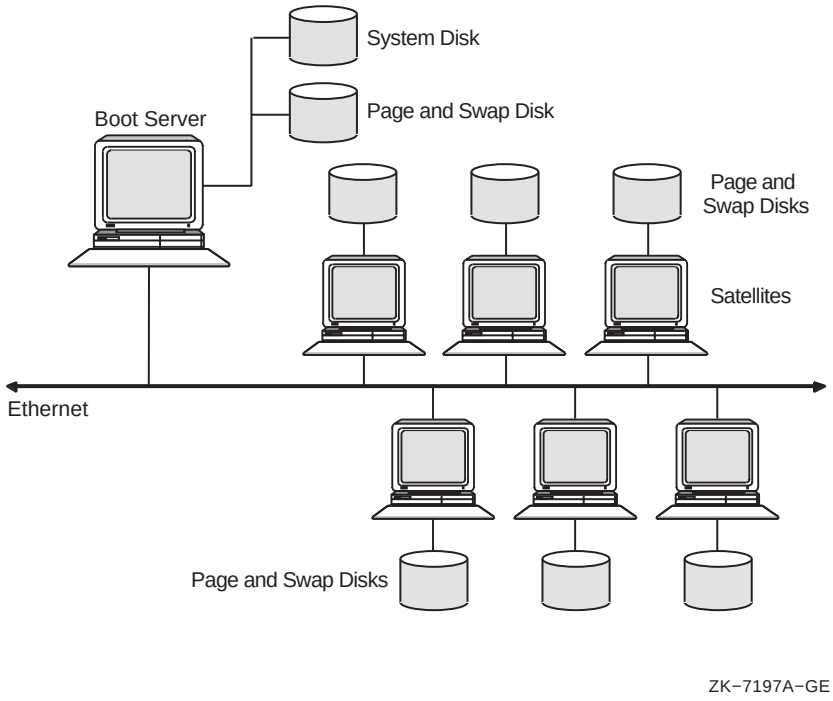
Figure 11–1 Common Environment with a Single System Disk



In Figure 11–1, a simple CI OpenVMS Cluster contains a single, shadowed system disk. This system disk contains system files, environment files, and page and swap files. Because there is one set of environment files, this is a common environment.

Figure 11–2 shows another variation of a simple OpenVMS Cluster with a common environment.

Figure 11–2 Simple LAN OpenVMS Cluster with a Single System Disk



In Figure 11–2, six satellites and one boot server are connected by Ethernet. Each satellite has its own page and swap disk, which saves system disk space and removes the I/O activity of page and swap files from the Ethernet. Removing page and swap files from the system disk improves performance for the OpenVMS Cluster.

Although the single-system-disk configuration works well for many OpenVMS Cluster requirements, multiple system disks can offer several advantages.

11.2.2 Multiple System Disks

OpenVMS Clusters that include both Alpha and VAX systems require multiple system disks: a VAX system disk and an Alpha system disk. Table 11–1 gives some additional reasons (not related to architecture) why a system manager might want more than one system disk in a OpenVMS Cluster.

Table 11–1 Advantages of Multiple System Disks

Advantage	Description
Decreased boot times	A single system disk can be a bottleneck when booting three or more systems simultaneously. Boot times are highly dependent on: • LAN utilization • Speed of the system disk • Number of disks mounted • Number of applications installed • Proximity of boot node to satellites • Boot node's processing power • Whether environment files are on the system disk • Whether the system disk is shadowed Volume Shadowing for OpenVMS software can help disk read performance, assuming that environment files that experience high write activity (such as SYSUAF.DAT) are not on the system disk.
Increased system and application performance	If your OpenVMS Cluster has many different applications that are in constant use, it may be advantageous to have either a local system disk for every node or a system disk that serves fewer systems. The benefits are shorter image-activation times and fewer files being served over the LAN. Alpha workstations benefit from a local system disk because the powerful Alpha processor does not have to wait as long for system disk access. **Reference:** See Section 10.7.5 for more information.
Reduced LAN utilization	More system disks reduce LAN utilization because fewer files are served over the LAN. Isolating LAN segments and their boot servers from unnecessary traffic outside the segments decreases LAN path contention. **Reference:** See Section 11.2.4 for more information.
Increased OpenVMS Cluster availability	A single system disk can become a single point of failure. Increasing the number of boot servers and system disks increases availability by reducing the OpenVMS Cluster's dependency on a single resource.

11.2.3 Multiple System-Disk OpenVMS Cluster

Arranging system disks as shown in Figure 11–3 can reduce booting time and LAN utilization.

Figure 11–3 Multiple System Disks in a Common Environment

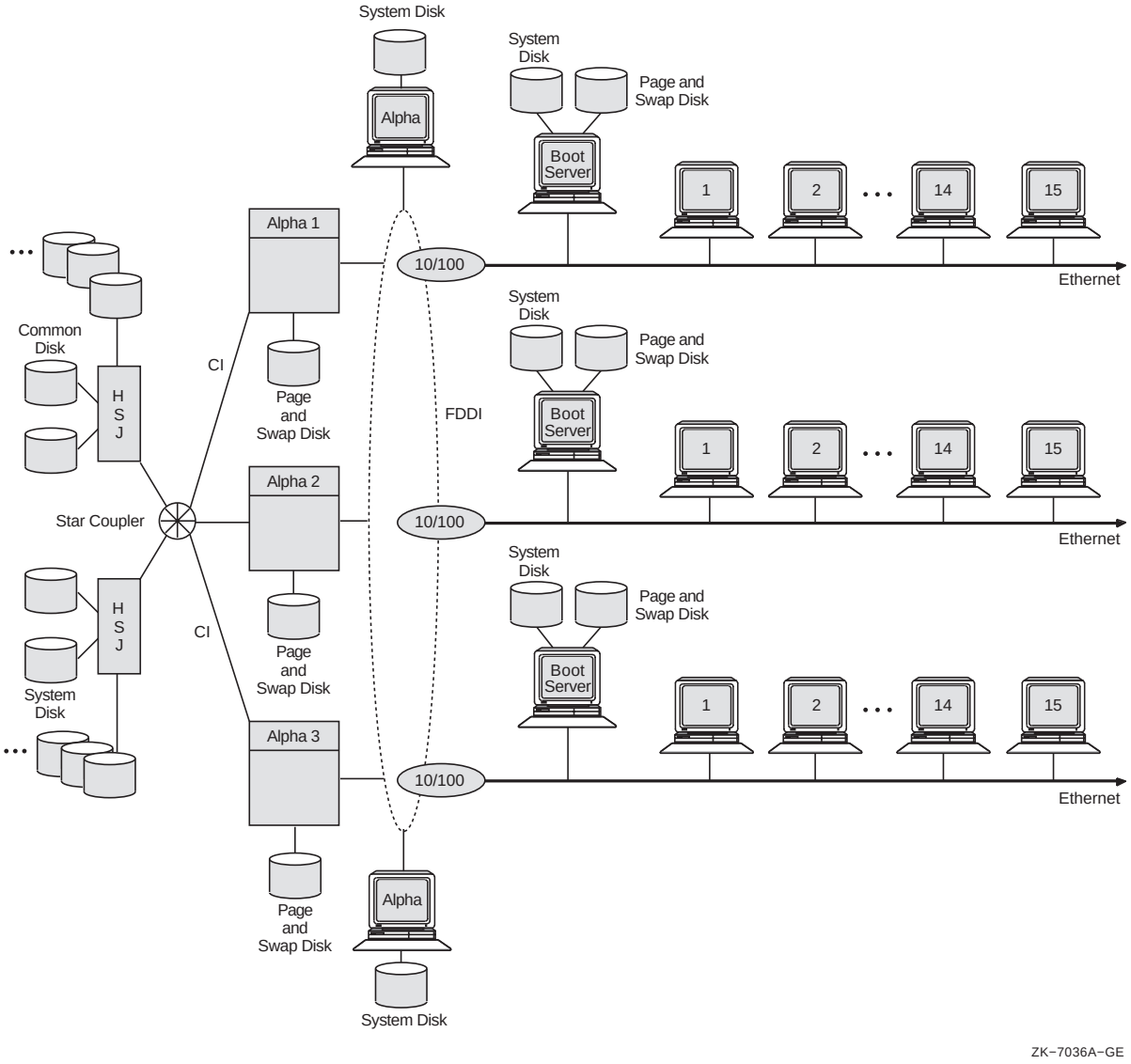


Figure 11–3 is an OpenVMS Cluster with multiple system disks:

- One for Alpha 1, Alpha 2, and Alpha 3
- One for each boot server on the LAN segments

The use of multiple system disks in this configuration and the way that the LAN segments are divided enable the booting sequence to be efficient and timely.

11.2.4 Dividing an OpenVMS Cluster System

In the workstation server examples shown in Section 10.7, OpenVMS Cluster reboots after a failure are relatively simple because of the small number of satellites per server. However, reboots in the larger, OpenVMS Cluster configuration shown in Figure 11–3 require careful planning. Dividing this OpenVMS Cluster and arranging the system disks as described in this section can reduce booting time significantly. Dividing the OpenVMS Cluster can

also reduce the satellite utilization of the LAN segment and increase satellite performance.

The disks in this OpenVMS Cluster have specific functions, as described in Table 11–2.

Table 11–2 How Multiple System Disks Are Used

Disk	Contents	Purpose
Common disk	All environment files for the entire OpenVMS Cluster	Environment files such as SYSUAF.DAT, NETPROXY.DAT, QMAN$MASTER.DAT are accessible to all nodes—including satellites—during booting. This frees the satellite boot servers to serve only system files and root information to the satellites. To create a common environment and increase performance for all system disks, see Section 11.3.
System disk	System roots for Alpha 1, Alpha 2, and Alpha 3	High performance for server systems. Make this disk as read-only as possible by taking environment files that have write activity off the system disk. The disk can be mounted clusterwide in SYLOGICALS.COM during startup.
Satellite boot servers' system disks	System files or roots for the satellites	Frees the system disk attached to Alpha 1, Alpha 2, and Alpha 3 from having to serve satellites, and divide total LAN traffic over individual Ethernet segments.
Page and swap disks	Page and swap files for one or more systems	Reduce I/O activity on the system disks, and free system disk space for applications and system roots.

In a booting sequence for the configuration in Figure 11–3, make sure that nodes Alpha 1, Alpha 2, and Alpha 3 are entirely booted before booting the LAN Ethernet segments so that the files on the common disk are available to the satellites. Enable filtering of the Maintenance Operations Protocol (MOP) on the Ethernet-to-FDDI (10/100) bridges so that the satellites do not try to boot from the system disks for Alpha 1, Alpha 2, and Alpha 3. The order in which to boot this OpenVMS Cluster is:

1. Boot Alpha 1, Alpha 2, and Alpha 3.
2. Boot the satellite boot servers.
3. Boot all satellites.

Reference: See Section 10.7.6 for information about extended LANs.

11.2.5 Summary: Single Versus Multiple System Disks

Use the information in Table 11–3 to determine whether you need a system disk for the entire OpenVMS Cluster or multiple system disks.

Table 11–3 Comparison of Single and Multiple System Disks

Single System Disk	Multiple System Disks
Node may have to wait longer for access to a file on the system disk.	Node does not have to wait for access to the system disk and has faster processor performance.
Contention for a single resource increases.	Contention for a single resource decreases.
Boot time for satellites increases.	Boot time for satellites decreases.
Only one system disk to manage.	More than one system disk to manage.
Less complex system management.	More complex system management, such as coordinating system parameters and files clusterwide.
Less hardware and software expense.	More hardware and software expense, especially if disks are shadowed.
Less expense for system management time and experience.	More expense for system management time and experience.

11.3 OpenVMS Cluster Environment Strategies

Depending on your processing needs, you can prepare either a common environment, in which all environment files are shared clusterwide, or a multiple environment, in which some files are shared clusterwide and others are accessible only by certain OpenVMS Cluster members.

The following are the most frequently used and manipulated OpenVMS Cluster environment files:

SYS$SYSTEM:SYSUAF.DAT
SYS$SYSTEM:NETPROXY.DAT
SYS$SYSTEM:VMSMAIL_PROFILE.DATA
SYS$SYSTEM:NETNODE_REMOTE.DAT
SYS$MANAGER:NETNODE_UPDATE.COM
SYS$SYSTEM:RIGHTSLIST.DAT
SYS$SYSTEM:QMAN$MASTER.DAT

Reference: For more information about managing these files, see *OpenVMS Cluster Systems*.

11.3.1 Common Environment

A common OpenVMS Cluster environment is an operating environment that is identical on all nodes in the OpenVMS Cluster. A common environment is easier to manage than multiple environments because you use a common version of each system file. The environment is set up so that:

- All nodes run the same programs, applications, and utilities.
- All users have the same type of accounts, and the same logical names are defined.
- All users can have common access to storage devices and queues.
- All users can log in to any node in the configuration and can work in the same environment as all other users.

The simplest and most inexpensive environment strategy is to have one system disk for the OpenVMS Cluster with all environment files on the same disk, as shown in Figure 11–1. The benefits of this strategy are:

- Software products need to be installed only once.

- All environment files are on the system disk and are easier to locate and manage.
- Booting dependencies are clear.

11.3.2 Putting Environment Files on a Separate, Common Disk

For an OpenVMS Cluster in which every node share the same system disk and environment, most common environment files are located in the SYS$SYSTEM directory.

However, you may want to move environment files to a separate disk so that you can improve OpenVMS Cluster performance. Because the environment files typically experience 80% of the system-disk activity, putting them on a separate disk decreases activity on the system disk. Figure 11–3 shows an example of a separate, common disk.

If you move environment files such as SYSUAF.DAT to a separate, common disk, SYSUAF.DAT will not be located in its default location of SYS$SYSTEM:SYSUAF.DAT.

Reference: See *OpenVMS Cluster Systems* for procedures to ensure that every node in the OpenVMS Cluster can access SYSUAF.DAT in its new location.

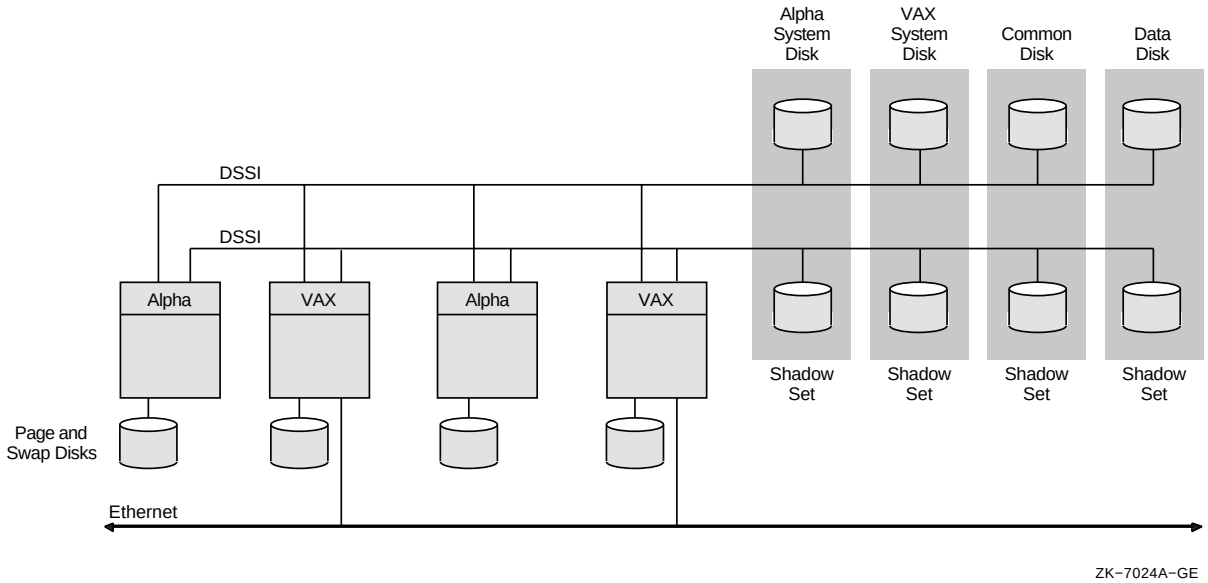
11.3.3 Multiple Environments

Multiple environments can vary from node to node. You can set up an individual node or a subset of nodes to:

- Provide multiple access according to the type of tasks users perform and the resources they use.
- Share a set of resources that are not available on other nodes.
- Perform specialized functions using restricted resources while other processors perform general timesharing work.
- Allow users to work in environments that are specific to the node where they are logged in.

Figure 11–4 shows an example of a multiple environment.

Figure 11–4 Multiple-Environment OpenVMS Cluster



In Figure 11–4, the multiple-environment OpenVMS Cluster consists of two system disks: one for VAX nodes and one for Alpha nodes. The common disk contains environment files for each node or group of nodes. Although many OpenVMS Cluster system managers prefer the simplicity of a single (common) environment, duplicating environment files is necessary for creating multiple environments that do not share resources across every node. Each environment can be tailored to the types of tasks users perform and the resources they use, and the configuration can have many different applications installed.

Each of the four DSSI nodes has its own page and swap disk, offloading the Alpha and VAX system disks on the DSSI interconnect from page and swap activity. All of the disks are shadowed across the DSSI interconnects, which protects the disks if a failure occurs.

11.4 Additional Multiple-Environment Strategies

This section describes additional multiple-environment strategies, such as using multiple SYSUAF.DAT files and multiple queue managers.

11.4.1 Using Multiple SYSUAF.DAT Files

Most OpenVMS Clusters are managed with one user authorization (SYSUAF.DAT) file, but you can use multiple user authorization files to limit access for some users to certain systems. In this scenario, users who need access to all systems also need multiple passwords.

Be careful about security with multiple SYSUAF.DAT files. The OpenVMS VAX and OpenVMS Alpha operating systems do not support multiple security domains.

Reference: See *OpenVMS Cluster Systems* for the list of fields that need to be the same for a single security domain, including SYSUAF.DAT entries.

Because Alpha systems require higher process quotas, system managers often respond by creating multiple SUSUAF.DAT files. This is not an optimal solution. Multiple SYSUAF.DAT files are intended only to vary environments from node to node, not to increase process quotas. To increase process quotas, Compaq recommends that you have one SYSUAF.DAT file and that you use system parameters to override process quotas in the SYSUAF.DAT file with system parameters to control resources for your Alpha systems.

11.4.2 Using Multiple Queue Managers

If the number of batch and print transactions on your OpenVMS Cluster is causing congestion, you can implement multiple queue managers to distribute the batch and print loads between nodes.

Every OpenVMS Cluster has only one QMAN$MASTER.DAT file. Multiple queue managers are defined through multiple *.QMAN$QUEUES and *.QMAN$JOURNAL files. Place each pair of queue manager files on different disks. If the QMAN$MASTER.DAT file has contention problems, place it on a solid-state disk to increase the number of batch and print transactions your OpenVMS Cluster can process. For example, you can create separate queue managers for batch queues and print queues.

Reference: See *OpenVMS System Manager's Manual: Essentials* for examples and commands to implement multiple queue managers.

11.5 Quorum Strategies

OpenVMS Cluster systems use a quorum algorithm to ensure synchronized access to storage. The quorum algorithm is a mathematical method for determining whether a majority of OpenVMS Cluster members exists so that they can "vote" on how resources can be shared across an OpenVMS Cluster system. The connection manager, which calculates quorum as a dynamic value, allows processing to occur only if a majority of the OpenVMS Cluster members are functioning.

Quorum votes are contributed by:

- Systems with the parameter VOTES set to a number greater than zero
- A designated disk, called a quorum disk

Each OpenVMS Cluster system can include only one quorum disk. The disk cannot be a member of a shadow set, but it can be the system disk.

The connection manager knows about the quorum disk from "quorum disk watchers," which are any systems that have a direct, active connection to the quorum disk.

11.5.1 Quorum Strategy Options

At least two systems should have a direct connection to the quorum disk. This ensures that the quorum disk votes are accessible if one of the systems fails.

When you consider quorum strategies, you must decide under what failure circumstances you want the OpenVMS Cluster to continue. Table 11–4 describes four options from which to choose.

Table 11–4 Quorum Strategies

Strategy Option[1]	Description
Continue if the majority of the maximum "expected" nodes still remain.	Give every node a vote and do not use a quorum disk. This strategy requires three or more nodes.
Continue with only one node remaining (of three or more nodes).	This strategy requires a quorum disk. By increasing the quorum disk's votes to one less than the total votes from all systems (and by increasing the value of the EXPECTED_VOTES system parameter by the same amount), you can boot and run the cluster with only one node as a quorum disk watcher. This prevents having to wait until more than half the voting systems are operational before you can start using the OpenVMS Cluster system.
Continue with only one node remaining (two-node OpenVMS Cluster).	Give each node and quorum disk a vote. The two-node OpenVMS Cluster is a special case of this alternative. By establishing a quorum disk, you can increase the availability of a two-node OpenVMS Cluster. Such configurations can maintain quorum and continue to operate in the event of failure of either the quorum disk or one node. This requires CI or DSSI nodes for both nodes to both be quorum disk watchers.
Continue with only critical nodes in the OpenVMS Cluster.	Generally, this strategy gives servers votes and gives satellites none. This assumes three or more servers and no quorum disk.

[1]These strategies are mutually exclusive; choose only one.

Reference: For more information about quorum disk management, see *OpenVMS Cluster Systems*.

11.6 State Transition Strategies

OpenVMS Cluster state transitions occur when a system joins or leaves an OpenVMS Cluster system and when the OpenVMS Cluster recognizes a quorum-disk state change. The connection manager handles these events to ensure the preservation of data integrity throughout the OpenVMS Cluster.

State transitions should be a concern only if systems are joining or leaving an OpenVMS Cluster system frequently enough to cause disruption.

A state transition's duration and effect on users and applications is determined by the reason for the transition, the configuration, and the applications in use. By managing transitions effectively, system managers can control:

- Detection of failures and how long the transition takes
- Side effects of the transition, such as volume shadowing copy and merge operations

11.6.1 Dealing with State Transitions

The following guidelines describe effective ways of dealing with transitions so that you can minimize the actual transition time as well as the side effects after the transition.

- Be proactive in preventing nodes from leaving an OpenVMS Cluster by:
  - Providing interconnect redundancy between all systems.
  - Preventing resource exhaustion of disks and memory as well as saturation of interconnects, processors, and adapters.
  - Using an uninterruptible power supply (UPS).

  - Informing users that shutting off a workstation in a large OpenVMS Cluster disrupts the operation of all systems in the cluster.

- Do not use a quorum disk unless your OpenVMS Cluster has only two nodes.
- Where possible, ensure that shadow set members reside on shared buses to increase availability.
- The time to detect the failure of nodes, disks, adapters, interconnects, and virtual circuits is controlled by system polling parameters. Reducing polling time makes the cluster react quickly to changes, but it also results in lower tolerance to temporary outages. When setting timers, try to strike a balance between rapid recovery from significant failures and "nervousness" resulting from temporary failures.

  Table 11–5 describes OpenVMS Cluster polling parameters that you can adjust for quicker detection time. Compaq recommends that these parameters be set to the same value in each OpenVMS Cluster.

Table 11–5 OpenVMS Cluster Polling Parameters

Parameter	Description
QDSKINTERVAL	Specifies the quorum disk polling interval.
RECNXINTERVL	Specifies the interval during which the connection manager attempts to restore communication to another system.
TIMVCFAIL	Specifies the time required for detection of a virtual circuit failure.

- Include application recovery in your plans. When you assess the effect of a state transition on application users, consider that the application recovery phase includes activities such as replaying a journal file, cleaning up recovery units, and users logging in again.

Reference: For more detailed information about OpenVMS Cluster transitions and their phases, system parameters, quorum management, see *OpenVMS Cluster Systems*.

11.7 Migration and Warranted Support for Multiple Versions

Compaq provides two levels of support for mixed-version and mixed-architecture OpenVMS Cluster systems: warranted support and migration support. Warranted support means that Compaq has fully qualified the two versions coexisting in an OpenVMS Cluster and will answer all problems identified by customers using these configurations.

Migration support is a superset of the Rolling Upgrade support provided in earlier releases of OpenVMS and is available for mixes that are not warranted. Migration support means that Compaq has qualified the versions for use together in configurations that are migrating in a staged fashion to a newer version of OpenVMS VAX or to OpenVMS Alpha. Problem reports submitted against these configurations will be answered by Compaq. However, in exceptional cases, Compaq may request that you move to a warranted configuration as part of answering the problem.

Migration support helps customers move to warranted OpenVMS Cluster version mixes with minimal impact on their cluster environments. Table 11–6 shows the level of support provided for all possible version pairings.

Table 11–6 OpenVMS Cluster Warranted and Migration Support

	Alpha V6.2–*xxx*	Alpha V7.1–*xxx*	Alpha V7.2
VAX V6.2–*xxx*	Warranted	Migration	Migration
VAX V7.1–*xxx*	Migration	Warranted	Migration
VAX V7.2	Migration	Migration	Warranted

For OpenVMS Version 6.2 nodes to participate in a cluster with systems running either Version 7.1 or Version 7.2, the cluster compatibility kit must be installed on each Version 6.2 node. For more information about this kit, see the *OpenVMS Version 7.2 Release Notes*.

Compaq does not support the use of Version 7.2 with Version 6.1 (or earlier versions) in an OpenVMS Cluster. In many cases, mixing Version 7.2 with versions prior to Version 6.2 will operate successfully, but Compaq cannot commit to resolving problems experienced with such configurations.

Note

Nodes running OpenVMS VAX Version 5.5–2 or earlier, or OpenVMS Alpha Version 1.0 or 1.5, cannot participate in a cluster with one or more OpenVMS Version 7.2 nodes. For more information, see the *OpenVMS Version 7.2 Release Notes*.

11.8 Alpha and VAX Systems in the Same OpenVMS Cluster

OpenVMS Alpha and OpenVMS VAX systems can work together in the same OpenVMS Cluster to provide both flexibility and migration capability. You can add Alpha processing power to an existing VAXcluster, enabling you to utilize applications that are system specific or hardware specific.

Table 11–6 depicts the OpenVMS version pairs for which Compaq provides migration and warranted support.

11.8.1 OpenVMS Cluster Satellite Booting Across Architectures

OpenVMS Alpha Version 7.1 and OpenVMS VAX Version 7.1 enable VAX boot nodes to provide boot service to Alpha satellites and Alpha boot nodes to provide boot service to VAX satellites. This support, called cross-architecture booting, increases configuration flexibility and higher availability of boot servers for satellites.

Two configuration scenarios make cross-architecture booting desirable:

- You want the Alpha system disk configured in the same highly available and high-performance area as your VAX system disk.
- Your Alpha boot server shares CI or DSSI storage with the VAX boot server. If your only Alpha boot server fails, you want to be able to reboot an Alpha satellite before the Alpha boot server reboots.

11.8.2 Restrictions

You cannot perform OpenVMS operating system and layered product installations and upgrades across architectures. For example, you must install and upgrade OpenVMS Alpha software using an Alpha system. When you configure OpenVMS Cluster systems that take advantage of cross-architecture booting, ensure that at least one system from each architecture is configured with a disk that can be used for installations and upgrades.

System disks can contain only a single version of the OpenVMS operating system and are architecture specific. For example, OpenVMS VAX Version 7.1 cannot coexist on a system disk with OpenVMS Alpha Version 7.1.

11.9 Determining Backup and Storage Management Strategies

In any system, hardware and electrical failures as well as human errors occur. All important data must be backed up to limit the effects of these errors. You can do this in a number of ways, depending on the time and resources available.

11.9.1 Steps for Determining a Backup Strategy

Follow these steps to determine a backup strategy:

Step	Description
1	Decide how much lost work is acceptable in the event of a failure. This determines how often the data needs to be backed up.
2	Decide how long the data can remain unavailable while it is being backed up. This determines the methods of backup.
3	Establish a backup schedule, including the frequency and times of the day and week that backups will occur. Determine: • How much data will be backed up daily, weekly, and monthly? • Will you conduct full or incremental backups? How often for each?
4	Make sure that sufficient backup media are available. Determine both the initial amount of backup media needed and its growth rate.
5	Determine if your backup strategy requires backup media to be stored off site.

11.10 Disk Backup

Table 11–7 describes ways to provide a copy of data for backup.

Table 11–7 Backup Methods for Data

Type of Data	Backup Method
Database is continually changing; transactions cannot be lost.	Use a combination of database backup (at a time when it is known to be static) and journaling transactions to the database. **Reference:** See the following manuals for additional information: • *RMS Journaling for OpenVMS Manual* • *Guide to OpenVMS File Applications* • *DEC Rdb Guide to Database Design and Definition* • *DEC DBMS Database Design Guide* • *DEC DBMS Database Maintenance and Performance Guide*
Data must be accessible at all times, including nights and weekends.	Use Volume Shadowing for OpenVMS software to accomplish rapid disk backup. Remove a member from a three-member shadow set by dismounting the shadow set, remounting the shadow set with two members, and copying the third disk to magnetic tape. After this, the third disk can be included again in the shadow set.
Data can be unavailable for an extended period of time for backup.	Use the OpenVMS Backup utility (BACKUP) to make an image backup of a volume or a file-by-file copy of specified sets of files. BACKUP can make a copy to another disk (or set of disks) or to magnetic tape. Restoring from an image copy requires that the entire image be written to a disk. When you restore specific files, they are copied from the restored disk to the intended destination. On the other hand, an image copy is faster than a file-by-file copy, which copies files one at a time. Restoring a single file from the backup copy is easy. Also, a file-by-file restore greatly reduces fragmentation of the restored disk.
Data is static.	Archiving copies of the data on magnetic tape and excluding the online files from other backup procedures may be sufficient. Examples are program sources, documentation files, and distribution kits.
Scratch files and intermediate files.	You can choose not to provide any backup for these files.

11.11 Tape Backup

Backup tape storage provides the least expensive storage medium. Tapes are the most common medium for offline storage and provide a range of capacities, cost, and shelf life. In general, tape storage is removable and generally off line.

11.11.1 For More Information

Backup procedures are described in detail in the following manuals:

- *OpenVMS System Manager's Manual*
- *OpenVMS System Management Utilities Reference Manual*

11.11.2 Benefits of Unattended Backup

With current tape-drive technology, you can initiate a large backup operation that completes without operator intervention (that is, changing tapes). Such unattended backups can save significant time and reduce staffing costs. Cartridge tape loaders with tape magazines, such as the T*x*8*x*7 or the TA91, allow unattended backups of up to nearly 42 GB of online storage. Backups can also be performed on robot-accessible media, such as the StorageTek® 4400 ACS through the TC44 interconnect adapter, which provides terabyte capacity for backup archives.

11.11.3 Archive/Backup System for OpenVMS

Archive/Backup System for OpenVMS is a replacement for the Storage Library System (SLS). Archive/Backup provides lower system management costs, reduced equipment costs, and data security. It uses the POLYCENTER Media Library Manager (MLM) and the POLYCENTER Media Robot Manager (MRM) to move data to inexpensive tapes, and allows you to find and restore backed up and archived data easily. POLYCENTER MLM and MRM are the first Compaq products to provide OpenVMS users secure, highly reliable, fully automated access to tape and optical removable media through cost-effective media robots, such as the Odetics 5480 and the T*x*8*x*7 family.

11.11.4 StorageTek 4400 ACS

You can attach the StorageTek 4400 ACS, a storage silo, to either an HSC using the TC44 adapter or directly to the XMI bus of a system using a KCM44 adapter. The StorageTek Silo automates access to a library of IBM® 3480 compatible cartridge tapes. The library can contain up to 16 library storage modules. Each module can hold up to 1.2 TB of data in 6000 tape cartridges. A robotic arm can find and mount a requested tape within 45 to 90 seconds. Data movement for tape applications, such as the OpenVMS Backup utility, is performed the same way as with a TA90 tape drive.

11.11.5 Tape-Drive Performance and Capacity

Table 11–8 describes the performance and capacity of various tape drives and the interconnects to which they attach.

Table 11–8 Tape-Drive Performance and Capacity

Interconnect	Description
CI (STI tapes)	The TA92 can transfer at a rate of 2.6 MB/s. Its magazine of IBM 3480 compatible cartridge tapes lets it back up 38 GB unattended. To achieve highest performance, connect the TA92 through a KDM70 controller or configure it with multiple CI adapters, so that the path to the tape drives is separate from the path to the disk drives.
DSSI	The TF867 offers the best tape performance. Its magazine of half-inch cartridge tapes can hold up to 42 GB of data for unattended backup. Its transfer rate is 0.8 MB/s. The TF857 can read TK50 and TK70 tapes, and its magazine can hold up to 18 GB of data.
SCSI	The TSZ07 allows SCSI configurations to access 9-track reel-to-reel tapes. It has a capacity of 140 MB per reel and a 750 KB/s transfer rate. The TZK10 offers a less expensive but slower-performing tape solution for SCSI configurations. It uses a quarter-inch cartridge that holds 525 MB and can transfer at a rate of 200 KB/s.

A

SCSI as an OpenVMS Cluster Interconnect

One of the benefits of OpenVMS Cluster systems is that multiple computers can simultaneously access storage devices connected to a OpenVMS Cluster storage interconnect. Together, these systems provide high performance and highly available access to storage.

This appendix describes how OpenVMS Cluster systems support the Small Computer Systems Interface (SCSI) as a storage interconnect. Multiple Alpha computers, also referred to as hosts or nodes, can simultaneously access SCSI disks over a SCSI interconnect. Such a configuration is called a SCSI multihost OpenVMS Cluster. A SCSI interconnect, also called a SCSI bus, is an industry-standard interconnect that supports one or more computers, peripheral devices, and interconnecting components.

The discussions in this chapter assume that you already understand the concept of sharing storage resources in an OpenVMS Cluster environment. OpenVMS Cluster concepts and configuration requirements are also described in the following OpenVMS Cluster documentation:

- *OpenVMS Cluster Systems*
- OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*)

This appendix includes two primary parts:

- Section A.1 through Section A.6.6 describe the fundamental procedures and concepts that you would need to plan and implement a SCSI multihost OpenVMS Cluster system.
- Section A.7 and its subsections provide additional technical detail and concepts.

A.1 Conventions Used in This Appendix

Certain conventions are used throughout this appendix to identify the ANSI Standard and for elements in figures.

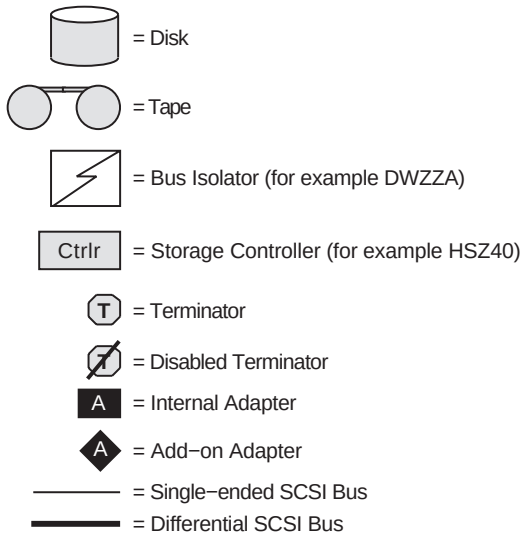
A.1.1 SCSI ANSI Standard

OpenVMS Cluster systems configured with the SCSI interconnect must use standard SCSI–2 or SCSI–3 components. The SCSI–2 components must be compliant with the architecture defined in the *American National Standards Institute (ANSI) Standard SCSI–2*, X3T9.2, Rev. 10L. The SCSI–3 components must be compliant with approved versions of the SCSI–3 Architecture and Command standards. For ease of discussion, this appendix uses the term SCSI to refer to both SCSI–2 and SCSI–3.

A.1.2 Symbols Used in Figures

Figure A–1 is a key to the symbols used in figures throughout this appendix.

Figure A–1 Key to Symbols Used in Figures



ZK−7759A−GE

A.2 Accessing SCSI Storage

In OpenVMS Cluster configurations, multiple VAX and Alpha hosts can directly access SCSI devices in any of the following ways:

- CI interconnect with HSJ or HSC controllers
- Digital Storage Systems Interconnect (DSSI) with HSD controller
- SCSI adapters directly connected to VAX or Alpha systems

You can also access SCSI devices indirectly using the OpenVMS MSCP server.

The following sections describe single-host and multihost access to SCSI storage devices.

A.2.1 Single-Host SCSI Access in OpenVMS Cluster Systems

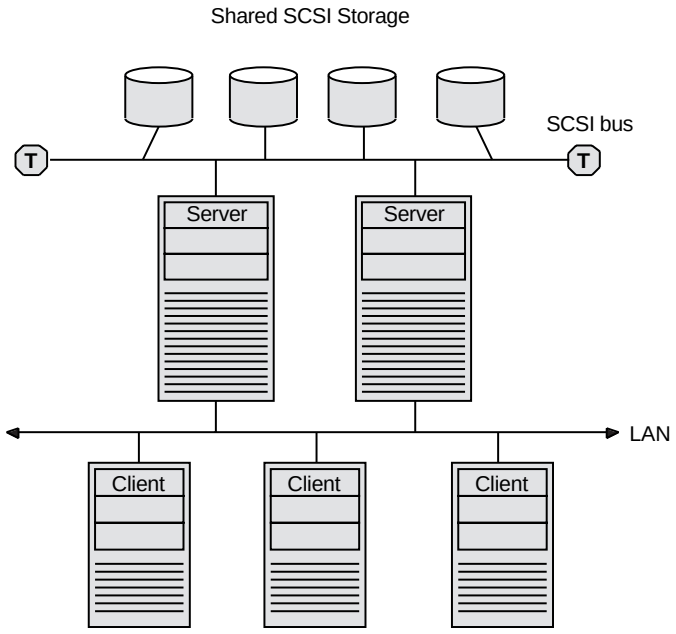
Prior to OpenVMS Version 6.2, OpenVMS Cluster systems provided support for SCSI storage devices connected to a single host using an embedded SCSI adapter, an optional external SCSI adapter, or a special-purpose RAID (redundant arrays of independent disks) controller. Only one host could be connected to a SCSI bus.

A.2.2 Multihost SCSI Access in OpenVMS Cluster Systems

Beginning with OpenVMS Alpha Version 6.2, multiple Alpha hosts in an OpenVMS Cluster system can be connected to a single SCSI bus to share access to SCSI storage devices directly. This capability allows you to build highly available servers using shared access to SCSI storage.

Figure A–2 shows an OpenVMS Cluster configuration that uses a SCSI interconnect for shared access to SCSI devices. Note that another interconnect (for example, a local area network [LAN]) is required for host-to-host OpenVMS Cluster (System Communications Architecture [SCA]) communications.

Figure A–2 Highly Available Servers for Shared SCSI Access



You can build a three-node OpenVMS Cluster system using the shared SCSI bus as the storage interconnect, or you can include shared SCSI buses within a larger OpenVMS Cluster configuration. A quorum disk can be used on the SCSI bus to improve the availability of two- or three-node configurations. Host-based RAID (including host-based shadowing) and the MSCP server are supported for shared SCSI storage devices.

A.3 Configuration Requirements and Hardware Support

This section lists the configuration requirements and supported hardware for multihost SCSI OpenVMS Cluster systems.

A.3.1 Configuration Requirements

Table A–1 shows the requirements and capabilities of the basic software and hardware components you can configure in a SCSI OpenVMS Cluster system.

Table A–1 Requirements for SCSI Multihost OpenVMS Cluster Configurations

Requirement	Description
Software	All Alpha hosts sharing access to storage on a SCSI interconnect must be running: • OpenVMS Alpha Version 6.2 or later • OpenVMS Cluster Software for OpenVMS Alpha Version 6.2 or later
Hardware	Table A–2 lists the supported hardware components for SCSI OpenVMS Cluster systems. See also Section A.7.7 for information about other hardware devices that might be used in a SCSI OpenVMS Cluster configuration.
SCSI tape, floppies, and CD–ROM drives	You cannot configure SCSI tape drives, floppy drives, or CD–ROM drives on multihost SCSI interconnects. If your configuration requires SCSI tape, floppy, or CD–ROM drives, configure them on single-host SCSI interconnects. Note that SCSI tape, floppy, or CD–ROM drives may be MSCP or TMSCP served to other hosts in the OpenVMS Cluster configuration.
Maximum hosts on a SCSI bus	You can connect up to three hosts on a multihost SCSI bus. You can configure any mix of the hosts listed in Table A–2 on the same shared SCSI interconnect.
Maximum SCSI buses per host	You can connect each host to a maximum of six multihost SCSI buses. The number of nonshared (single-host) SCSI buses that can be configured is limited only by the number of available slots on the host bus.
Host-to-host communication	All members of the cluster must be connected by an interconnect that can be used for host-to-host (SCA) communication; for example, DSSI, CI, Ethernet, FDDI, or MEMORY CHANNEL.
Host-based RAID (including host-based shadowing)	Supported in SCSI OpenVMS Cluster configurations.
SCSI device naming	The name of each SCSI device must be unique throughout the OpenVMS Cluster system. When configuring devices on systems that include a multihost SCSI bus, adhere to the following requirements: • A host can have, at most, one adapter attached to a particular SCSI interconnect. • All host controllers attached to a given SCSI interconnect must have the same OpenVMS device name (for example, PKA0), unless port allocation classes are used (see *OpenVMS Cluster Systems*). • Each system attached to a SCSI interconnect must have the same nonzero node allocation class, regardless of whether port allocation classes are used (see *OpenVMS Cluster Systems*).

A.3.2 Hardware Support

Table A–2 shows the supported hardware components for SCSI OpenVMS Cluster systems; it also lists the minimum required revision for these hardware components. That is, for any component, you must use either the version listed in Table A–2 or a subsequent version. For host support information, refer to the *DIGITAL Systems and Options Catalog* on the World Wide Web at the following address:

`http://www.digital.com:80/info/soc/`

You can also access the *DIGITAL Systems and Options Catalog* from the OpenVMS web site by selecting Publications and then selecting this catalog.

For disk support information, refer to StorageWorks documentation. You can access the StorageWorks web site at the following address:

http://www.storage.digital.com/

You can also access the StorageWorks web site from the OpenVMS web site by selecting Hardware and then selecting Storage.

The SCSI interconnect configuration and all devices on the SCSI interconnect must meet the requirements defined in the *ANSI Standard SCSI–2* document, or the SCSI–3 Architecture and Command standards, and the requirements described in this appendix. See also Section A.7.7 for information about other hardware devices that might be used in a SCSI OpenVMS Cluster configuration.

Table A–2 Supported Hardware for SCSI OpenVMS Cluster Systems

Component	Supported Item	Minimum Firmware (FW) Version[1]
Controller	HSZ40–B	2.5 (FW)
	HSZ50	
	HSZ70	
	HSZ80	8.3 (FW)
Adapters[2]	Embedded (NCR-810 based)	
	KZPAA (PCI to SCSI)	
	KZPSA (PCI to SCSI)	A11 (FW)
	KZPBA-CB (PCI to SCSI)	5.53 (FW)
	KZTSA (TURBOchannel to SCSI)	A10-1 (FW)

[1]Unless stated in this column, the minimum firmware version for a device is the same as required for the operating system version you are running. There are no additional firmware requirements for a SCSI multihost OpenVMS Cluster configuration.

[2]You can configure other types of SCSI adapters in a system for single-host access to local storage.

A.4 SCSI Interconnect Concepts

The SCSI standard defines a set of rules governing the interactions between initiators (typically, host systems) and SCSI targets (typically, peripheral devices). This standard allows the host to communicate with SCSI devices (such as disk drives, tape drives, printers, and optical media devices) without having to manage the device-specific characteristics.

The following sections describe the SCSI standard and the default modes of operation. The discussions also describe some optional mechanisms you can implement to enhance the default SCSI capabilities in areas such as capacity, performance, availability, and distance.

A.4.1 Number of Devices

The SCSI bus is an I/O interconnect that can support up to 16 devices. A narrow SCSI bus supports up to 8 devices; a wide SCSI bus support up to 16 devices. The devices can include host adapters, peripheral controllers, and discrete peripheral devices such as disk or tape drives. The devices are addressed by a unique ID number from 0 through 15. You assign the device IDs by entering console commands, or by setting jumpers or switches, or by selecting a slot on a StorageWorks enclosure.

Note

In order to connect 16 devices to a wide SCSI bus, the devices themselves must also support wide addressing. Narrow devices do not talk to hosts above ID 7. Presently, the HSZ40 does not support addresses above 7. Host adapters that support wide addressing are KZTSA, KZPSA, and the QLogic wide adapters (KZPBA, KZPDA, ITIOP, P1SE, and P2SE). Only the KZPBA–CB is supported in a multihost SCSI OpenVMS Cluster configuration.

When configuring more devices than the previous limit of eight, make sure that you observe the bus length requirements (see Table A–4).

To configure wide IDs on a BA356 box, refer to the BA356 manual *StorageWorks Solutions BA356-SB 16-Bit Shelf User's Guide* (order number EK-BA356-UG). Do not configure a narrow device in a BA356 box that has a starting address of 8.

To increase the number of devices on the SCSI interconnect, some devices implement a second level of device addressing using logical unit numbers (LUNs). For each device ID, up to eight LUNs (0–7) can be used to address a single SCSI device as multiple units. The maximum number of LUNs per device ID is eight.

Note

When connecting devices to a SCSI interconnect, each device on the interconnect must have a unique device ID. You may need to change a device's default device ID to make it unique. For information about setting a single device's ID, refer to the owner's guide for the device.

A.4.2 Performance

The default mode of operation for all SCSI devices is 8-bit asynchronous mode. This mode, sometimes referred to as narrow mode, transfers 8 bits of data from one device to another. Each data transfer is acknowledged by the device receiving the data. Because the performance of the default mode is limited, the SCSI standard defines optional mechanisms to enhance performance. The following list describes two optional methods for achieving higher performance:

- Increase the amount of data that is transferred in parallel on the interconnect. The 16-bit and 32-bit wide options allow a doubling or quadrupling of the data rate, respectively. Because the 32-bit option is seldom implemented, this appendix discusses only 16-bit operation and refers to it as **wide**.
- Use synchronous data transfer. In synchronous mode, multiple data transfers can occur in succession, followed by an acknowledgment from the device receiving the data. The standard defines a slow mode (also called standard mode) and a fast mode for synchronous data transfers:
  - In standard mode, the interconnect achieves up to 5 million transfers per second.
  - In fast mode, the interconnect achieves up to 10 million transfers per second.

- In ultra mode, the interconnect achieves up to 20 million transfers per second.

Because all communications on a SCSI interconnect occur between two devices at a time, each pair of devices must negotiate to determine which of the optional features they will use. Most, if not all, SCSI devices implement one or more of these options.

Table A–3 shows data rates when using 8- and 16-bit transfers with standard, fast, and ultra synchronous modes.

Table A–3 Maximum Data Transfer Rates (MB/s)

Mode	Narrow (8-bit)	Wide (16-bit)
Standard	5	10
Fast	10	20
Ultra	20	40

A.4.3 Distance

The maximum length of the SCSI interconnect is determined by the signaling method used in the configuration and by the data transfer rate. There are two types of electrical signaling for SCSI interconnects:

- Single-ended signaling

  The single-ended method is the most common and the least expensive. The distance spanned is generally modest.

- Differential signaling

  This method provides higher signal integrity, thereby allowing a SCSI bus to span longer distances.

Table A–4 summarizes how the type of signaling method affects SCSI interconnect distances.

Table A–4 Maximum SCSI Interconnect Distances

Signaling Technique	Rate of Data Transfer	Maximum Cable Length
Single ended	Standard	6 m[1]
Single ended	Fast	3 m
Single ended	Ultra	20.5 m[2]
Differential	Standard or fast	25 m
Differential	Ultra	25.5 m[2]

[1]The SCSI standard specifies a maximum length of 6 m for this type of interconnect. However, where possible, it is advisable to limit the cable length to 4 m to ensure the highest level of data integrity.

[2]For more information, refer to the *StorageWorks UltraSCSI Configuration Guidelines*, order number EK–ULTRA–CG.

The **DWZZA, DWZZB, and DWZZC converters** are single-ended to differential converters that you can use to connect single-ended and differential SCSI interconnect segments. The DWZZA is for narrow (8-bit) SCSI buses, the DWZZB is for wide (16-bit) SCSI buses, and the DWZZC is for wide Ultra SCSI buses.

The differential segments are useful for the following:

- Overcoming the distance limitations of the single-ended interconnect
- Allowing communication between single-ended and differential devices

Because the DWZZA, the DWZZB, and the DWZZC are strictly signal converters, you can not assign a SCSI device ID to them. You can configure a maximum of two DWZZA or two DWZZB converters in the path between any two SCSI devices. Refer to the *StorageWorks UltraSCSI Configuration Guidelines* for information on configuring the DWZZC.

A.4.4 Cabling and Termination

Each single-ended and differential SCSI interconnect must have two terminators, one at each end. The specified maximum interconnect lengths are measured from terminator to terminator.

The interconnect terminators are powered from the SCSI interconnect line called TERMPWR. Each StorageWorks host adapter and enclosure supplies the TERMPWR interconnect line, so that as long as one host or enclosure is powered on, the interconnect remains terminated.

Devices attach to the interconnect by short cables (or etch) called stubs. Stubs must be short in order to maintain the signal integrity of the interconnect. The maximum stub lengths allowed are determined by the type of signaling used by the interconnect, as follows:

- For single-ended interconnects, the maximum stub length is .1 m.
- For differential interconnects, the maximum stub length is .2 m.

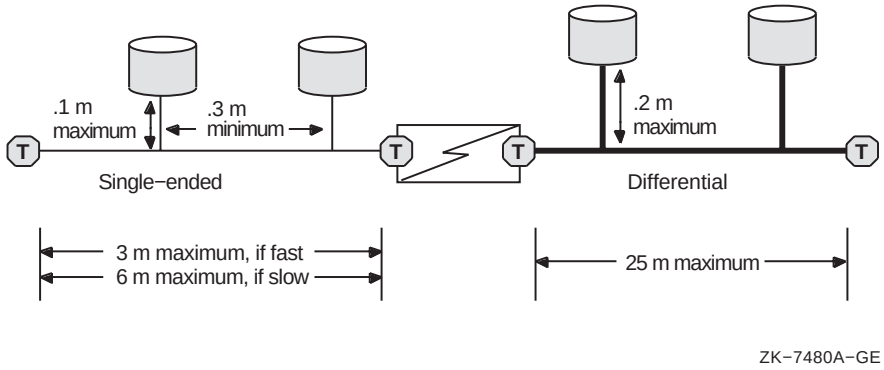
Additionally, the minimum distance between stubs on a single-ended interconnect is .3 m. Refer to Figure A–3 for an example of this configuration.

Note

Terminate single-ended and differential buses individually, even when using DWZZ*x* converters.

When you are extending the SCSI bus beyond an existing terminator, it is necessary to disable or remove that terminator.

Figure A–3 Maximum Stub Lengths



A.5 SCSI OpenVMS Cluster Hardware Configurations

The hardware configuration that you choose depends on a combination of factors:

- Your computing needs—for example, continuous availability or the ability to disconnect or remove a system from your SCSI OpenVMS Cluster system
- Your environment—for example, the physical attributes of your computing facility
- Your resources—for example, your capital equipment or the available PCI slots

Refer to the OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*) for configuration limits.

The following sections provide guidelines for building SCSI configurations and describe potential configurations that might be suitable for various sites.

A.5.1 Systems Using Add-On SCSI Adapters

Shared SCSI bus configurations typically use optional add-on KZPAA, KZPSA, KZPBA, and KZTSA adapters. These adapters are generally easier to configure than internal adapters because they do not consume any SCSI cable length. Additionally, when you configure systems using add-on adapters for the shared SCSI bus, the internal adapter is available for connecting devices that cannot be shared (for example, SCSI tape, floppy, and CD–ROM drives).

When using add-on adapters, storage is configured using BA350, BA353, or HSZ*xx* StorageWorks enclosures. These enclosures are suitable for all data disks, and for shared OpenVMS Cluster system and quorum disks. By using StorageWorks enclosures, it is possible to shut down individual systems without losing access to the disks.

The following sections describe some SCSI OpenVMS Cluster configurations that take advantage of add-on adapters.

A.5.1.1 Building a Basic System Using Add-On SCSI Adapters

Figure A–4 shows a logical representation of a basic configuration using SCSI adapters and a StorageWorks enclosure. This configuration has the advantage of being relatively simple, while still allowing the use of tapes, floppies, CD–ROMs, and disks with nonshared files (for example, page files and swap files) on internal buses. Figure A–5 shows this type of configuration using AlphaServer 1000 systems and a BA350 enclosure.

The BA350 enclosure uses 0.9 m of SCSI cabling, and this configuration typically uses two 1-m SCSI cables. (A BA353 enclosure also uses 0.9 m, with the same total cable length.) The resulting total cable length of 2.9 m allows fast SCSI mode operation.

Although the shared BA350 storage enclosure is theoretically a single point of failure, this basic system is a very reliable SCSI OpenVMS Cluster configuration. When the quorum disk is located in the BA350, you can shut down either of the AlphaStation systems independently while retaining access to the OpenVMS Cluster system. However, you cannot physically remove the AlphaStation system, because that would leave an unterminated SCSI bus.

If you need the ability to remove a system while your OpenVMS Cluster system remains operational, build your system using DWZZ*x* converters, as described in Section A.5.1.2. If you need continuous access to data if a SCSI interconnect fails, you should do both of the following:

- Add a redundant SCSI interconnect with another BA350 shelf.
- Shadow the data.

In Figure A–4 and the other logical configuration diagrams in this appendix, the required network interconnect is not shown.

Figure A–4 Conceptual View: Basic SCSI System

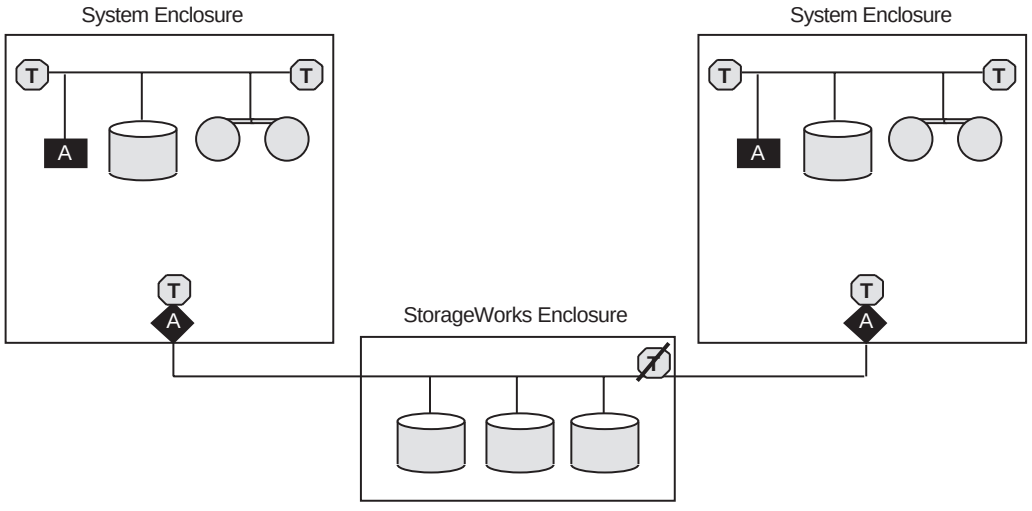
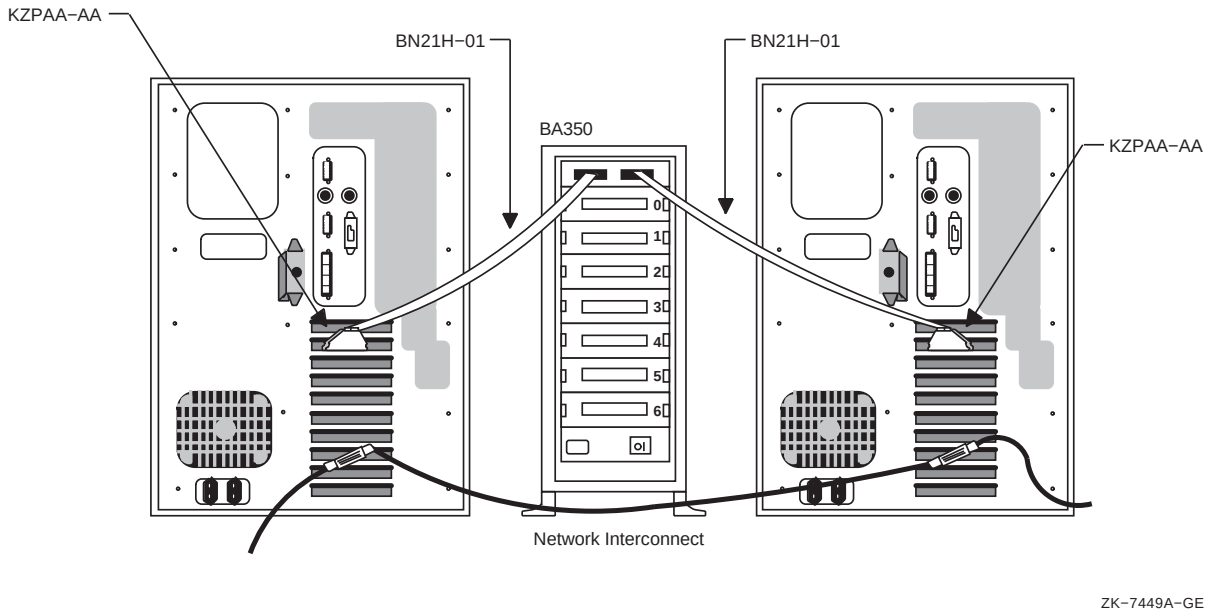


Figure A–5 Sample Configuration: Basic SCSI System Using AlphaServer 1000, KZPAA Adapter, and BA350 Enclosure



A.5.1.2 Building a System with More Enclosures or Greater Separation or with HSZ Controllers

If you need additional enclosures, or if the needs of your site require a greater physical separation between systems, or if you plan to use HSZ controllers, you can use a configuration in which DWZZ*x* converters are placed between systems with single-ended signaling and a differential-cabled SCSI bus.

DWZZ*x* converters provide additional SCSI bus length capabilities, because the DWZZ*x* allows you to connect a single-ended device to a bus that uses differential signaling. As described in Section A.4.3, SCSI bus configurations that use differential signaling may span distances up to 25 m, whereas single-ended configurations can span only 3 m when fast-mode data transfer is used.

DWZZ*x* converters are available as standalone, desktop components or as StorageWorks compatible building blocks. DWZZ*x* converters can be used with the internal SCSI adapter or the optional KZPAA adapters.

The HSZ40 is a high-performance differential SCSI controller that can be connected to a differential SCSI bus, and supports up to 72 SCSI devices. An HSZ40 can be configured on a shared SCSI bus that includes DWZZ*x* single-ended to differential converters. Disk devices configured on HSZ40 controllers can be combined into RAID sets to further enhance performance and provide high availability.

Figure A–6 shows a logical view of a configuration that uses additional DWZZAs to increase the potential physical separation (or to allow for additional enclosures and HSZ40s), and Figure A–7 shows a sample representation of this configuration.

Figure A–6 Conceptual View: Using DWZZAs to Allow for Increased Separation or More Enclosures

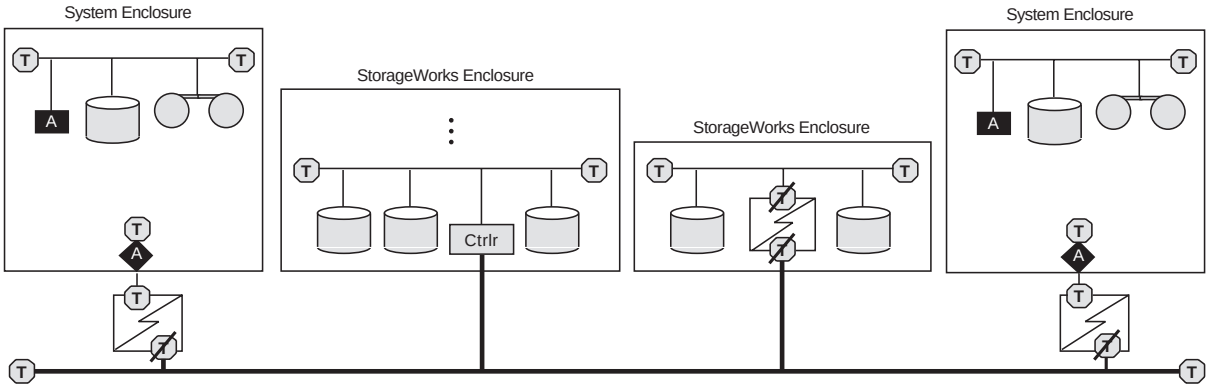


Figure A–7 Sample Configuration: Using DWZZAs to Allow for Increased Separation or More Enclosures

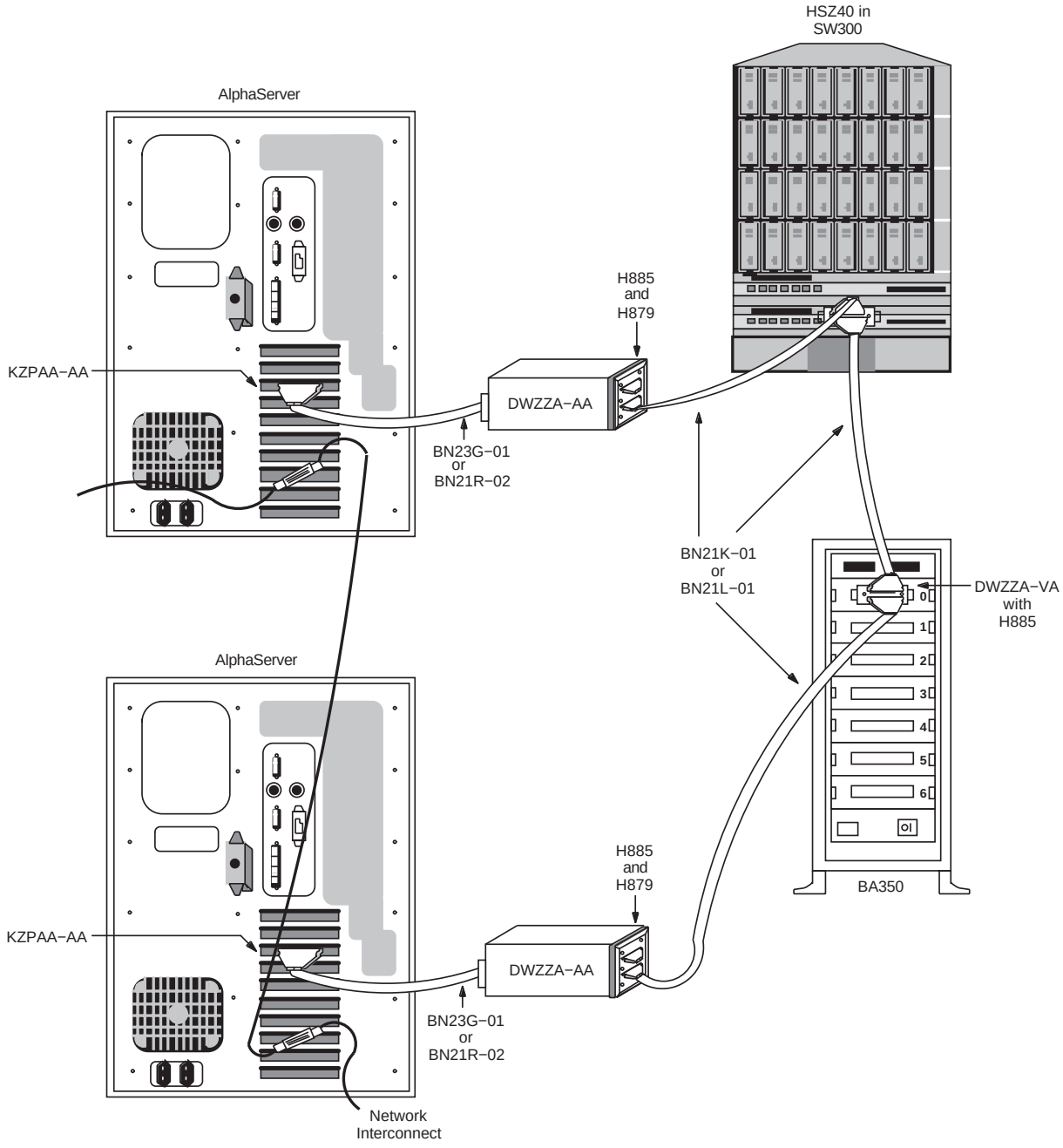
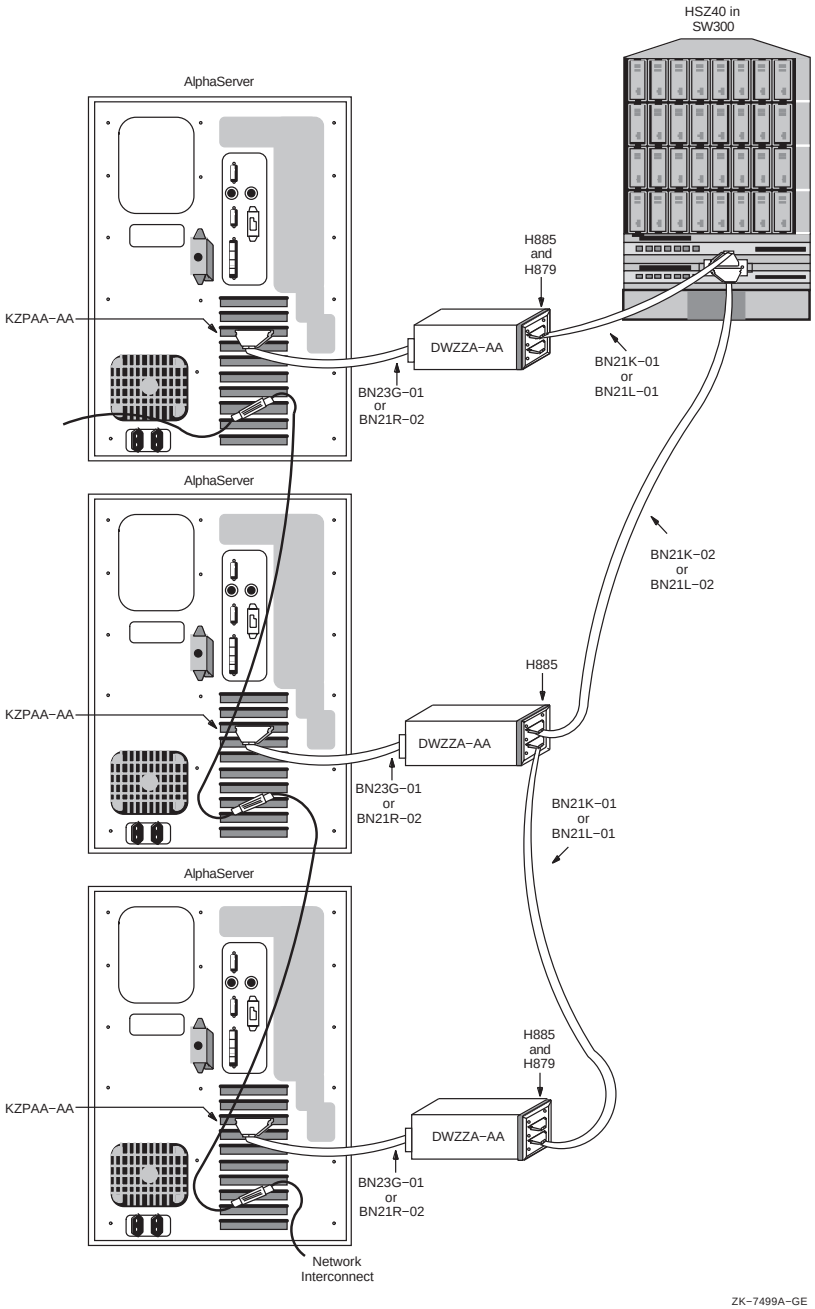


Figure A–8 shows how a three-host SCSI OpenVMS Cluster system might be configured.

Figure A–8 Sample Configuration: Three Hosts on a SCSI Bus



A.5.1.3 Building a System That Uses Differential Host Adapters

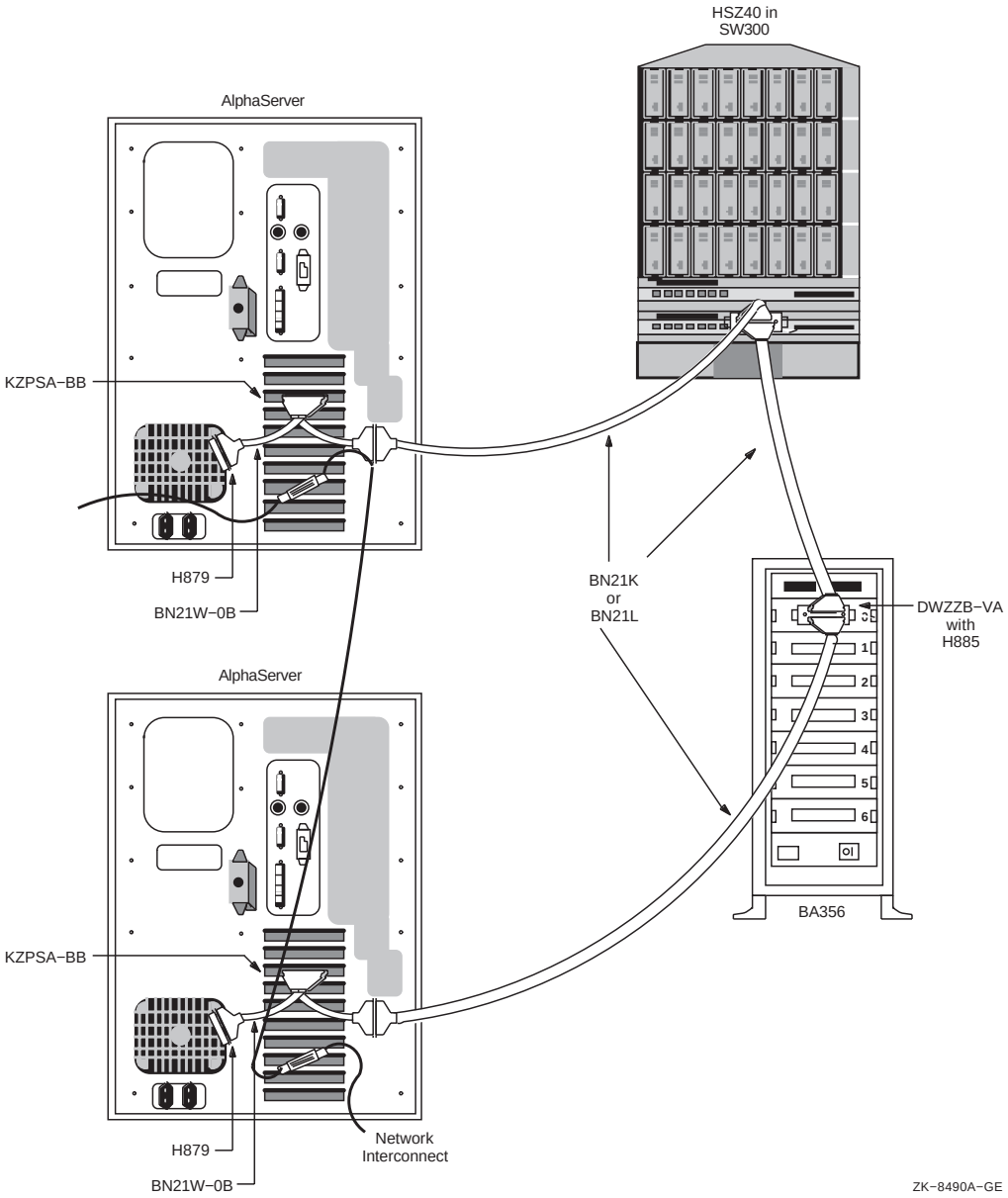
Figure A–9 is a sample configuration with two KZPSA adapters on the same SCSI bus. In this configuration, the SCSI termination has been removed from the KZPSA, and external terminators have been installed on "Y" cables. This allows you to remove the KZPSA adapter from the SCSI bus without rendering the SCSI bus inoperative. The capability of removing an individual system from your SCSI OpenVMS Cluster configuration (for maintenance or repair) while the

other systems in the cluster remain active gives you an especially high level of availability.

Please note the following about Figure A–9:

- Termination is removed from the host adapter.
- Termination for the single-ended bus inside the BA356 is provided by the DWZZB in slot 0 and by the automatic terminator on the personality module. (No external cables or terminators are attached to the personality module.)
- The DWZZB's differential termination is removed.

Figure A–9 Sample Configuration: SCSI System Using Differential Host Adapters (KZPSA)



The differential SCSI bus in the configuration shown in Figure A–9 is chained from enclosure to enclosure and is limited to 25 m in length. (The BA356 does not add to the differential SCSI bus length. The differential bus consists only of the BN21W-0B "Y" cables and the BN21K/BN21L cables.) In configurations where this cabling scheme is inconvenient or where it does not provide adequate distance, an alternative radial scheme can be used.

The radial SCSI cabling alternative is based on a SCSI hub. Figure A–10 shows a logical view of the SCSI hub configuration, and Figure A–11 shows a sample representation of this configuration.

Figure A–10 Conceptual View: SCSI System Using a SCSI Hub

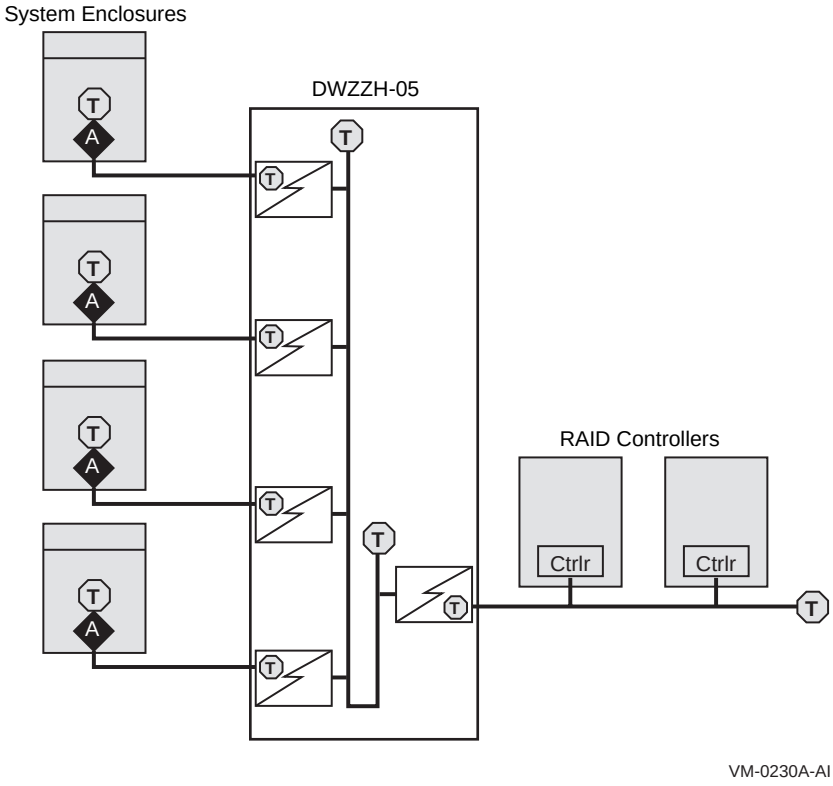
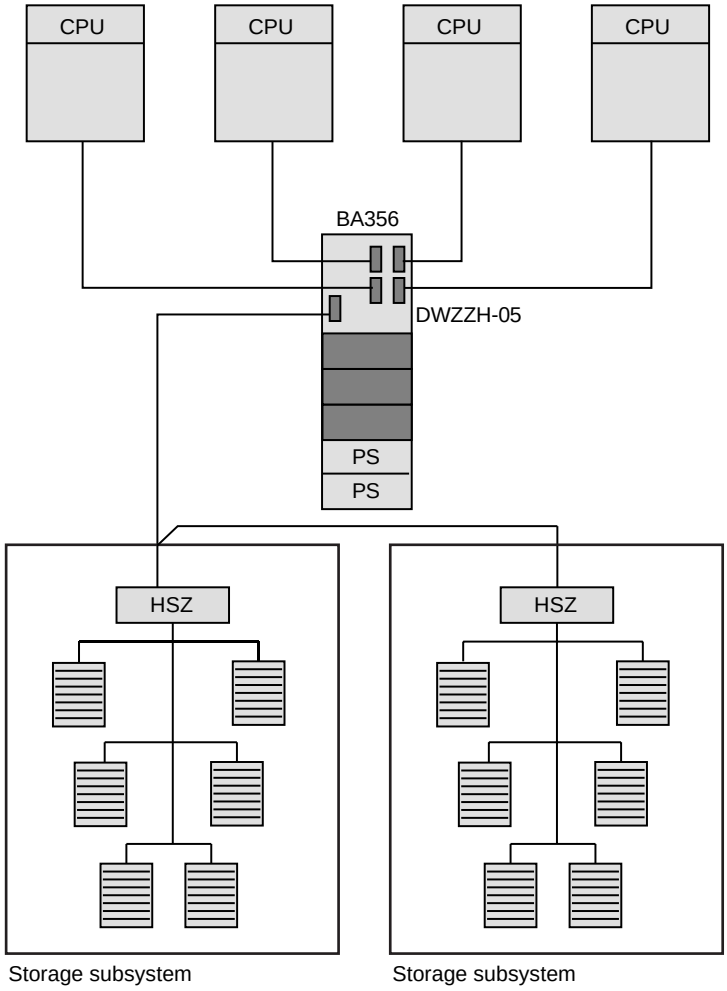


Figure A–11 shows a sample representation of a SCSI hub configuration.

Figure A–11 Sample Configuration: SCSI System with SCSI Hub Configuration



A.6 Installation

This section describes the steps required to set up and install the hardware in a SCSI OpenVMS Cluster system. The assumption in this section is that a new OpenVMS Cluster system, based on a shared SCSI bus, is being created. If, on the other hand, you are adding a shared SCSI bus to an existing OpenVMS Cluster configuration, then you should integrate the procedures in this section with those described in *OpenVMS Cluster Systems* to formulate your overall installation plan.

Table A–5 lists the steps required to set up and install the hardware in a SCSI OpenVMS Cluster system.

Table A–5 Steps for Installing a SCSI OpenVMS Cluster System

Step	Description	Reference
1	Ensure proper grounding between enclosures.	Section A.6.1 and Section A.7.8
2	Configure SCSI host IDs.	Section A.6.2
3	Power up the system and verify devices.	Section A.6.3
4	Set SCSI console parameters.	Section A.6.4
5	Install the OpenVMS operating system.	Section A.6.5
6	Configure additional systems.	Section A.6.6

A.6.1 Step 1: Meet SCSI Grounding Requirements

You must ensure that your electrical power distribution systems meet local requirements (for example, electrical codes) prior to installing your OpenVMS Cluster system. If your configuration consists of two or more enclosures connected by a common SCSI interconnect, you must also ensure that the enclosures are properly grounded. Proper grounding is important for safety reasons and to ensure the proper functioning of the SCSI interconnect.

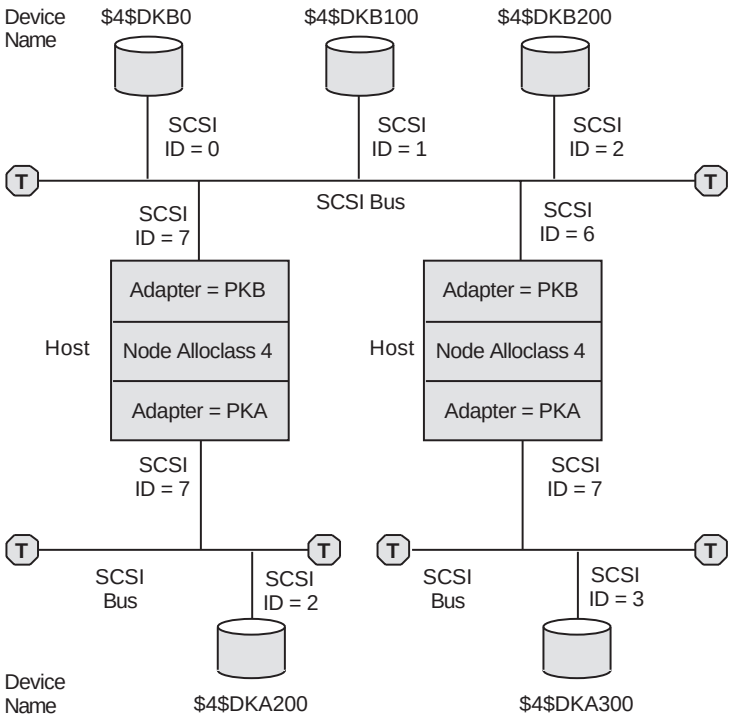
Electrical work should be done by a qualified professional. Section A.7.8 includes details of the grounding requirements for SCSI systems.

A.6.2 Step 2: Configure SCSI Node IDs

This section describes how to configure SCSI node and device IDs. SCSI IDs must be assigned separately for multihost SCSI buses and single-host SCSI buses.

Figure A–12 shows two hosts; each one is configured with a single-host SCSI bus and shares a multihost SCSI bus. (See Figure A–1 for the key to the symbols used in this figure.)

Figure A–12 Setting Allocation Classes for SCSI Access



The following sections describe how IDs are assigned in this type of multihost SCSI configuration. For more information about this topic, see *OpenVMS Cluster Systems*.

A.6.2.1 Configuring Device IDs on Multihost SCSI Buses

When configuring multihost SCSI buses, adhere to the following rules:

- Set each host adapter on the multihost bus to a different ID. Start by assigning ID 7, then ID 6, and so on, using decreasing ID numbers.

  If a host has two multihost SCSI buses, allocate an ID to each SCSI adapter separately. There is no requirement that you set the adapters to the same ID, although using the same ID may simplify configuration management. (Section A.6.4 describes how to set host IDs for the internal adapter using SCSI console parameters.)

- When assigning IDs to devices and storage controllers connected to multihost SCSI buses, start at ID 0 (zero), assigning the highest ID numbers to the disks that require the fastest I/O response time.

- Devices connected to a multihost SCSI bus must have the same name as viewed from each host. To achieve this, you must do one of the following:

  – Ensure that all hosts connected to a multihost SCSI bus are set to the same node allocation class, and all host adapters connected to a multihost SCSI bus have the same controller letter, as shown in Figure A–12.

  – Use port allocation classes (see *OpenVMS Cluster Systems*) or HSZ allocation classes (see *Guidelines for OpenVMS Cluster Configurations*).

A.6.2.2 Configuring Device IDs on Single-Host SCSI Buses

The device ID selection depends on whether you are using a node allocation class or a port allocation class. The following discussion applies to node allocation classes. Refer to *OpenVMS Cluster Systems* for a discussion of port allocation classes.

In multihost SCSI configurations, device names generated by OpenVMS use the format $*allocation_class*$DKA300. You set the allocation class using the ALLOCLASS system parameter. OpenVMS generates the controller letter (for example, A, B, C, and so forth) at boot time by allocating a letter to each controller. The unit number (for example, 0, 100, 200, 300, and so forth) is derived from the SCSI device ID.

When configuring devices on single-host SCSI buses that are part of a multihost SCSI configuration, take care to ensure that the disks connected to the single-host SCSI buses have unique device names. Do this by assigning different IDs to devices connected to single-host SCSI buses with the same controller letter on systems that use the same allocation class. Note that the device names must be different, even though the bus is not shared.

For example, in Figure A–12, the two disks at the bottom of the picture are located on SCSI bus A of two systems that use the same allocation class. Therefore, they have been allocated different device IDs (in this case, 2 and 3).

For a given allocation class, SCSI device type, and controller letter (in this example, 4DKA), there can be up to eight devices in the cluster, one for each SCSI bus ID. To use all eight IDs, it is necessary to configure a disk on one SCSI bus at the same ID as a processor on another bus. See Section A.7.5 for a discussion of the possible performance impact this can have.

SCSI bus IDs can be effectively "doubled up" by configuring different SCSI device types at the same SCSI ID on different SCSI buses. For example, device types DK and MK could produce 4DKA100 and 4MKA100.

A.6.3 Step 3: Power Up and Verify SCSI Devices

After connecting the SCSI cables, power up the system. Enter a console SHOW DEVICE command to verify that all devices are visible on the SCSI interconnect.

If there is a SCSI ID conflict, the display may omit devices that are present, or it may include nonexistent devices. If the display is incorrect, then check the SCSI ID jumpers on devices, the automatic ID assignments provided by the StorageWorks shelves, and the console settings for host adapter and HSZ*xx* controller IDs. If changes are made, type INIT, then SHOW DEVICE again. If problems persist, check the SCSI cable lengths and termination.

Example A–1 is a sample output from a console SHOW DEVICE command. This system has one host SCSI adapter on a private SCSI bus (PKA0), and two additional SCSI adapters (PKB0 and PKC0), each on separate, shared SCSI buses.

Example A–1 SHOW DEVICE Command Sample Output

```
>>>SHOW DEVICE
dka0.0.0.6.0              DKA0                    RZ26L  442D
dka400.4.0.6.0            DKA400                  RRD43  2893
dkb100.1.0.11.0           DKB100                  RZ26  392A
dkb200.2.0.11.0           DKB200                  RZ26L  442D
dkc400.4.0.12.0           DKC400                  HSZ40   V25
dkc401.4.0.12.0           DKC401                  HSZ40   V25
dkc500.5.0.12.0           DKC500                  HSZ40   V25
dkc501.5.0.12.0           DKC501                  HSZ40   V25
dkc506.5.0.12.0           DKC506                  HSZ40   V25
dva0.0.0.0.1              DVA0
jkb700.7.0.11.0           JKB700                  OpenVMS  V62
jkc700.7.0.12.0           JKC700                  OpenVMS  V62
mka300.3.0.6.0            MKA300                  TLZ06  0389
era0.0.0.2.1              ERA0                    08-00-2B-3F-3A-B9
pka0.7.0.6.0              PKA0                    SCSI Bus ID 7
pkb0.6.0.11.0             PKB0                    SCSI Bus ID 6
pkc0.6.0.12.0             PKC0                    SCSI Bus ID 6
```

The following list describes the device names in the preceding example:

- DK devices represent SCSI disks. Disks connected to the SCSI bus controlled by adapter PKA are given device names starting with the letters DKA. Disks on additional buses are named according to the host adapter name in a similar manner (DKB devices on adapter PKB, and so forth).

  The next character in the device name represents the device's SCSI ID. Make sure that the SCSI ID for each device is unique for the SCSI bus to which it is connected.

- The last digit in the DK device name represents the LUN number. The HSZ40 virtual DK device in this example is at SCSI ID 4, LUN 1. Note that some systems do not display devices that have nonzero LUNs.

- JK devices represent nondisk or nontape devices on the SCSI interconnect. In this example, JK devices represent other processors on the SCSI interconnect that are running the OpenVMS operating system. If the other system is not running, these JK devices do not appear in the display. In this example, the other processor's adapters are at SCSI ID 7.

- MK devices represent SCSI tapes. The A in device MKA300 indicates that it is attached to adapter PKA0, the private SCSI bus.

- PK devices represent the local SCSI adapters. The SCSI IDs for these adapters is displayed in the rightmost column. Make sure this is different from the IDs used by other devices and host adapters on its bus.

  The third character in the device name (in this example, a) is assigned by the system so that each adapter has a unique name on that system. The fourth character is always zero.

A.6.4 Step 4: Show and Set SCSI Console Parameters

When creating a SCSI OpenVMS Cluster system, you need to verify the settings of the console environment parameters shown in Table A–6 and, if necessary, reset their values according to your configuration requirements.

Table A–6 provides a brief description of SCSI console parameters. Refer to your system-specific documentation for complete information about setting these and other system parameters.

Note

The console environment parameters vary, depending on the host adapter type. Refer to the Installation and User's Guide for your adapter.

Table A–6 SCSI Environment Parameters

Parameter	Description
bootdef_dev *device_name*	Specifies the default boot device to the system.
boot_osflags *root_ number,bootflag*	The boot_osflags variable contains information that is used by the operating system to determine optional aspects of a system bootstrap (for example, conversational bootstrap).
pk*0_disconnect	Allows the target to disconnect from the SCSI bus while the target acts on a command. When this parameter is set to 1, the target is allowed to disconnect from the SCSI bus while processing a command. When the parameter is set to 0, the target retains control of the SCSI bus while acting on a command.
pk*0_fast	Enables SCSI adapters to perform in fast SCSI mode. When this parameter is set to 1, the default speed is set to fast mode; when the parameter is 0, the default speed is standard mode.
pk*0_host_id	Sets the SCSI device ID of host adapters to a value between 0 and 7.
scsi_poll	Enables console polling on all SCSI interconnects when the system is halted.
control_scsi_term	Enables and disables the terminator on the integral SCSI interconnect at the system bulkhead (for some systems).

Note

If you need to modify any parameters, first change the parameter (using the appropriate console SET command). Then enter a console INIT command or press the Reset button to make the change effective.

Examples

Before setting boot parameters, display the current settings of these parameters, as shown in the following examples:

1.

```
>>>SHOW *BOOT*

boot_osflags            10,0
boot_reset              OFF
bootdef_dev             dka200.2.0.6.0
>>>
```

The first number in the boot_osflags parameter specifies the system root. (In this example, the first number is 10.) The boot_reset parameter controls the boot process. The default boot device is the device from which the OpenVMS operating system is loaded. Refer to the documentation for your specific system for additional booting information.

Note that you can identify multiple boot devices to the system. By doing so, you cause the system to search for a bootable device from the list of devices that you specify. The system then automatically boots from the first device on which it finds bootable system software. In addition, you can override the default boot device by specifying an alternative device name on the boot command line.

Typically, the default boot flags suit your environment. You can override the default boot flags by specifying boot flags dynamically on the boot command line with the -flags option.

2.

```
>>>SHOW *PK*

pka0_disconnect         1
pka0_fast               1
pka0_host_id            7
```

The pk*0_disconnect parameter determines whether or not a target is allowed to disconnect from the SCSI bus while it acts on a command. On a multihost SCSI bus, the pk*0_disconnect parameter *must* be set to 1, so that disconnects can occur.

The pk*0_fast parameter controls whether fast SCSI devices on a SCSI controller perform in standard or fast mode. When the parameter is set to 0, the default speed is set to standard mode; when the pk*0_fast parameter is set to 1, the default speed is set to fast SCSI mode. In this example, devices on SCSI controller pka0 are set to fast SCSI mode. This means that both standard and fast SCSI devices connected to this controller will automatically perform at the appropriate speed for the device (that is, in either fast or standard mode).

The pk*0_host_id parameter assigns a bus node ID for the specified host adapter. In this example, pka0 is assigned a SCSI device ID of 7.

3.

```
>>>SHOW *POLL*
scsi_poll               ON
```

Enables or disables polling of SCSI devices while in console mode.

Set polling ON or OFF depending on the needs and environment of your site. When polling is enabled, the output of the SHOW DEVICE is always up to date. However, because polling can consume SCSI bus bandwidth (proportional to the number of unused SCSI IDs), you might want to disable polling if one system on a multihost SCSI bus will be in console mode for an extended time.

Polling *must* be disabled during any hot-plugging operations. For information about hot plugging in a SCSI OpenVMS Cluster environment, see Section A.7.6.

4.

```
>>>SHOW *TERM*
control_scsi_term       external
```

Used on some systems (such as the AlphaStation 400) to enable or disable the SCSI terminator next to the external connector. Set the control_scsi_term parameter to external if a cable is attached to the bulkhead. Otherwise, set the parameter to internal.

A.6.5 Step 5: Install the OpenVMS Operating System

Refer to the OpenVMS Alpha or VAX upgrade and installation manual for information about installing the OpenVMS operating system. Perform the installation once for each system disk in the OpenVMS Cluster system. In most configurations, there is a single system disk. Therefore, you need to perform this step once, using any system.

During the installation, when you are asked if the system is to be a cluster member, answer Yes. Then, complete the installation according to the guidelines provided in *OpenVMS Cluster Systems*.

A.6.6 Step 6: Configure Additional Systems

Use the CLUSTER_CONFIG command procedure to configure additional systems. Execute this procedure once for the second host that you have configured on the SCSI bus. (See Section A.7.1 for more information.)

A.7 Supplementary Information

The following sections provide supplementary technical detail and concepts about SCSI OpenVMS Cluster systems.

A.7.1 Running the OpenVMS Cluster Configuration Command Procedure

You execute either the CLUSTER_CONFIG.COM or the CLUSTER_CONFIG_LAN.COM command procedure to set up and configure nodes in your OpenVMS Cluster system. Your choice of command procedure depends on whether you use DECnet or the LANCP utility for booting. CLUSTER_CONFIG.COM uses DECnet; CLUSTER_CONFIG_LAN.COM uses the LANCP utility. (For information about using both procedures, see *OpenVMS Cluster Systems*.)

Typically, the first computer is set up as an OpenVMS Cluster system during the initial OpenVMS installation procedure (see Section A.6.5). The CLUSTER_CONFIG procedure is then used to configure additional nodes. However, if you originally installed OpenVMS without enabling clustering, the first time you run CLUSTER_CONFIG, the procedure converts the standalone system to a cluster system.

To configure additional nodes in a SCSI cluster, execute CLUSTER_CONFIG.COM for each additional node. Table A–7 describes the steps to configure additional SCSI nodes.

Table A–7 Steps for Installing Additional Nodes

Step	Procedure
1	From the first node, run the CLUSTER_CONFIG.COM procedure and select the default option [1] for ADD.
2	Answer Yes when CLUSTER_CONFIG.COM asks whether you want to proceed.
3	Supply the DECnet name and address of the node that you are adding to the existing single-node cluster.
4	Confirm that this will be a node with a shared SCSI interconnect.
5	Answer No when the procedure asks whether this node will be a satellite.
6	Configure the node to be a disk server if it will serve disks to other cluster members.
7	Place the new node's system root on the default device offered.
8	Select a system root for the new node. The first node uses SYS0. Take the default (SYS10 for the first additional node), or choose your own root numbering scheme. You can choose from SYS1 to SYS*n*, where *n* is hexadecimal FFFF.
9	Select the default disk allocation class so that the new node in the cluster uses the same ALLOCLASS as the first node.
10	Confirm whether or not there is a quorum disk.
11	Answer the questions about the sizes of the page file and swap file.
12	When CLUSTER_CONFIG.COM completes, boot the new node from the new system root. For example, for SYSFF on disk DKA200, enter the following command: `BOOT -FL FF,0 DKA200` In the BOOT command, you can use the following flags: • -FL indicates boot flags. • FF is the new system root. • 0 means there are no special boot requirements, such as conversational boot.

You can run the CLUSTER_CONFIG.COM procedure to set up an additional node in a SCSI cluster, as shown in Example A–2.

Example A–2 Adding a Node to a SCSI Cluster

```
$ @SYS$MANAGER:CLUSTER_CONFIG
          Cluster Configuration Procedure

    Use CLUSTER_CONFIG.COM to set up or change an OpenVMS Cluster configuration.
    To ensure that you have the required privileges, invoke this procedure
    from the system manager's account.

    Enter ? for help at any prompt.

           1. ADD a node to a cluster.
           2. REMOVE a node from the cluster.
           3. CHANGE a cluster member's characteristics.
           4. CREATE a duplicate system disk for CLU21.
           5. EXIT from this procedure.

    Enter choice [1]:

    The ADD function adds a new node to a cluster.
```

(continued on next page)

Example A–2 (Cont.) Adding a Node to a SCSI Cluster

```
    If the node being added is a voting member, EXPECTED_VOTES in
    every cluster member's MODPARAMS.DAT must be adjusted, and the
    cluster must be rebooted.

    WARNING - If this cluster is running with multiple system disks and
              if common system files will be used, please, do not
              proceed unless you have defined appropriate logical
              names for cluster common files in SYLOGICALS.COM.
              For instructions, refer to the OpenVMS Cluster Systems
              manual.

              Do you want to continue [N]? y

    If the new node is a satellite, the network databases on CLU21 are
    updated. The network databases on all other cluster members must be
    updated.

    For instructions, refer to the OpenVMS Cluster Systems manual.

What is the node's DECnet node name? SATURN
What is the node's DECnet node address? 7.77
Is SATURN to be a clustered node with a shared SCSI bus (Y/N)? y
Will SATURN be a satellite [Y]? N
Will SATURN be a boot server [Y]?

    This procedure will now ask you for the device name of SATURN's system root.
    The default device name (DISK$BIG_X5T5:) is the logical volume name of
    SYS$SYSDEVICE:.

What is the device name for SATURN's system root [DISK$BIG_X5T5:]?
What is the name of SATURN's system root [SYS10]? SYS2
    Creating directory tree SYS2 ...
    System root SYS2 created

    NOTE:
        All nodes on the same SCSI bus must be members of the same cluster
        and must all have the same non-zero disk allocation class or each
        will have a different name for the same disk and data corruption
        will result.

Enter a value for SATURN's ALLOCLASS parameter [7]:
Does this cluster contain a quorum disk [N]?
Updating network database...
Size of pagefile for SATURN [10000 blocks]?
   .
   .
   .
```

A.7.2 Error Reports and OPCOM Messages in Multihost SCSI Environments

Certain common operations, such as booting or shutting down a host on a multihost SCSI bus, can cause other hosts on the SCSI bus to experience errors. In addition, certain errors that are unusual in a single-host SCSI configuration may occur more frequently on a multihost SCSI bus.

These errors are transient errors that OpenVMS detects, reports, and recovers from without losing data or affecting applications that are running. This section describes the conditions that generate these errors and the messages that are displayed on the operator console and entered into the error log.

A.7.2.1 SCSI Bus Resets

When a host connected to a SCSI bus first starts, either by being turned on or by rebooting, it does not know the state of the SCSI bus and the devices on it. The ANSI SCSI standard provides a method called BUS RESET to force the bus and its devices into a known state. A host typically asserts a RESET signal one or more times on each of its SCSI buses when it first starts up and when it shuts down. While this is a normal action on the part of the host asserting RESET, other hosts consider this RESET signal an error because RESET requires that the hosts abort and restart all I/O operations that are in progress.

A host may also reset the bus in the midst of normal operation if it detects a problem that it cannot correct in any other way. These kinds of resets are uncommon, but they occur most frequently when something on the bus is disturbed. For example, an attempt to hot plug a SCSI device while the device is still active (see Section A.7.6) or halting one of the hosts with Ctrl/P can cause a condition that forces one or more hosts to issue a bus reset.

A.7.2.2 SCSI Timeouts

When a host exchanges data with a device on the SCSI bus, there are several different points where the host must wait for the device or the SCSI adapter to react. In an OpenVMS system, the host is allowed to do other work while it is waiting, but a timer is started to make sure that it does not wait too long. If the timer expires without a response from the SCSI device or adapter, this is called a timeout.

There are three kinds of timeouts:

- Disconnect timeout—The device accepted a command from the host and disconnected from the bus while it processed the command but never reconnected to the bus to finish the transaction. This error happens most frequently when the bus is very busy. See Section A.7.5 for more information. The disconnect timeout period varies with the device, but for most disks, it is about 20 seconds.
- Selection timeout—The host tried to send a command to a device on the SCSI bus, but the device did not respond. This condition might happen if the device did not exist or if it were removed from the bus or powered down. (This failure is not more likely with a multi-initiator system; it is mentioned here for completeness.) The selection timeout period is about 0.25 seconds.
- Interrupt timeout—The host expected the adapter to respond for any other reason, but it did not respond. This error is usually an indication of a busy SCSI bus. It is more common if you have initiator unit numbers set low (0 or 1) rather than high (6 or 7). The interrupt timeout period is about 4 seconds.

Timeout errors are not inevitable on SCSI OpenVMS Cluster systems. However, they are more frequent on SCSI buses with heavy traffic and those with two initiators. They do not necessarily indicate a hardware or software problem. If they are logged frequently, you should consider ways to reduce the load on the SCSI bus (for example, adding an additional bus).

A.7.2.3 Mount Verify

Mount verify is a condition declared by a host about a device. The host declares this condition in response to a number of possible transient errors, including bus resets and timeouts. When a device is in the mount verify state, the host suspends normal I/O to it until the host can determine that the correct device is there, and that the device is accessible. Mount verify processing then retries outstanding I/Os in a way that insures that the correct data is written or read. Application programs are unaware that a mount verify condition has occurred as long as the mount verify completes.

If the host cannot access the correct device within a certain amount of time, it declares a mount verify timeout, and application programs are notified that the device is unavailable. Manual intervention is required to restore a device to service after the host has declared a mount verify timeout. A mount verify timeout usually means that the error is not transient. The system manager can choose the timeout period for mount verify; the default is one hour.

A.7.2.4 Shadow Volume Processing

Shadow volume processing is a process similar to mount verify, but it is for shadow set members. An error on one member of a shadow set places the set into the volume processing state, which blocks I/O while OpenVMS attempts to regain access to the member. If access is regained before shadow volume processing times out, then the outstanding I/Os are reissued and the shadow set returns to normal operation. If a timeout occurs, then the failed member is removed from the set. The system manager can select one timeout value for the system disk shadow set, and one for application shadow sets. The default value for both timeouts is 20 seconds.

Note

The SCSI disconnect timeout and the default shadow volume processing timeout are the same. If the SCSI bus is heavily utilized so that disconnect timeouts may occur, it may be desirable to increase the value of the shadow volume processing timeout. (A recommended value is 60 seconds.) This may prevent shadow set members from being expelled when they experience disconnect timeout errors.

A.7.2.5 Expected OPCOM Messages in Multihost SCSI Environments

When a bus reset occurs, an OPCOM message is displayed as each mounted disk enters and exits mount verification or shadow volume processing.

When an I/O to a drive experiences a timeout error, an OPCOM message is displayed as that drive enters and exits mount verification or shadow volume processing.

If a quorum disk on the shared SCSI bus experiences either of these errors, then additional OPCOM messages may appear, indicating that the connection to the quorum disk has been lost and regained.

A.7.2.6 Error Log Basics

In the OpenVMS system, the Error Log utility allows device drivers to save information about unusual conditions that they encounter. In the past, most of these unusual conditions have happened as a result of errors such as hardware failures, software failures, or transient conditions (for example, loose cables).

If you type the DCL command SHOW ERROR, the system displays a summary of the errors that have been logged since the last time the system booted. For example:

```
$ SHOW ERROR

Device                            Error Count
SALT$PKB0:                              6
$1$DKB500:                             10
PEA0:                                   1
SALT$PKA0:                              9
$1$DKA0:                                0
```

In this case, 6 errors have been logged against host SALT's SCSI port B (PKB0), 10 have been logged against disk 1DKB500, and so forth.

To see the details of these errors, you can use the command ANALYZE/ERROR /SINCE=*dd-mmm-yyyy:hh:mm:ss* at the DCL prompt. The output from this command displays a list of error log entries with information similar to the following:

```
******************************* ENTRY    2337. *******************************
 ERROR SEQUENCE 6.                              LOGGED ON:  CPU_TYPE 00000002
 DATE/TIME 29-MAY-1995 16:31:19.79                           SYS_TYPE 0000000D

<identification information>

       ERROR TYPE            03
                                      COMMAND TRANSMISSION FAILURE
       SCSI ID               01
                                      SCSI ID = 1.
       SCSI LUN              00
                                      SCSI LUN = 0.
       SCSI SUBLUN           00
                                      SCSI SUBLUN = 0.
       PORT STATUS     00000E32
                                      %SYSTEM-E-RETRY, RETRY OPERATION

<additional information>
```

For this discussion, the key elements are the ERROR TYPE and, in some instances, the PORT STATUS fields. In this example, the error type is 03, COMMAND TRANSMISSION FAILURE, and the port status is 00000E32, SYSTEM-E-RETRY.

A.7.2.7 Error Log Entries in Multihost SCSI Environments

The error log entries listed in this section are likely to be logged in a multihost SCSI configuration, and you usually do not need to be concerned about them. You should, however, examine any error log entries for messages other than those listed in this section.

- ERROR TYPE 0007, BUS RESET DETECTED

  Occurs when the other system asserts the SCSI bus reset signal. This happens when:

  – A system's power-up self-test runs.

  - A console INIT command is executed.
  - The EISA Configuration Utility (ECU) is run.
  - The console BOOT command is executed (in this case, several resets occur).
  - System shutdown completes.
  - The system detects a problem with an adapter or a SCSI bus (for example, an interrupt timeout).

  This error causes all mounted disks to enter mount verification.

- ERROR TYPE 05, EXTENDED SENSE DATA RECEIVED

  When a SCSI bus is reset, an initiator must get "sense data" from each device. When the initiator gets this data, an EXTENDED SENSE DATA RECEIVED error is logged. This is expected behavior.

- ERROR TYPE 03, COMMAND TRANSMISSION FAILURE
  PORT STATUS E32, SYSTEM-E-RETRY

  Occasionally, one host may send a command to a disk while the disk is exchanging error information with the other host. Many disks respond with a SCSI "BUSY" code. The OpenVMS system responds to a SCSI BUSY code by logging this error and retrying the operation. You are most likely to see this error when the bus has been reset recently. This error does not always happen near resets, but when it does, the error is expected and unavoidable.

- ERROR TYPE 204, TIMEOUT

  An interrupt timeout has occurred (see Section A.7.2.2). The disk is put into mount verify when this error occurs.

- ERROR TYPE 104, TIMEOUT

  A selection timeout has occurred (see Section A.7.2.2). The disk is put into mount verify when this error occurs.

A.7.3 Restrictions and Known Problems

The OpenVMS Cluster software has the following restrictions when multiple hosts are configured on the same SCSI bus:

- For versions prior to OpenVMS Alpha Version 7.2, a node's access to a disk will not fail over from a direct SCSI path to an MSCP served path.

  There is also no failover from an MSCP served path to a direct SCSI path. Normally, this type of failover is not a consideration, because when OpenVMS discovers both a direct and a served path, it chooses the direct path permanently. However, you must avoid situations in which the MSCP served path becomes available first and is selected by OpenVMS before the direct path becomes available. To avoid this situation, observe the following rules:

  - A node that has a direct path to a SCSI system disk must boot the disk directly from the SCSI port, not over the LAN.
  - If a node is running the MSCP server, then a SCSI disk must not be added to the multihost SCSI bus after a second node boots (either by physically inserting it or by reconfiguring an HSZ*xx*).

If you add a device after two nodes boot and then configure the device using SYSMAN, the device might become visible to one of the systems through the served path before the direct path is visible. Depending upon the timing of various events, this problem can sometimes be avoided by using the following procedure:

```
$ MCR SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> IO AUTOCONFIGURE
```

To ensure that the direct path to a new device is used (including HSZ*xx* virtual devices), reboot each node after a device is added.

- For versions prior to OpenVMS Alpha Version 7.2, if there are two paths to a device, the $DEVICE_SCAN system service and the F$DEVICE lexical function list each device on a shared bus twice. Devices on the shared bus are also listed twice in the output from the DCL command SHOW DEVICE if you boot a non-SCSI system disk. These double listings are errors in the display programs. They do not indicate a problem or imply that the MSCP served path is being used instead of the direct SCSI path.
- When a system powers up, boots, or shuts down, it resets the SCSI bus. These resets cause other hosts on the SCSI bus to experience I/O errors. For Files–11 volumes, the Mount Verification facility automatically recovers from these errors and completes the I/O. As a result, the user's process continues to run without error.

  This level of error recovery is not possible for volumes that are mounted with the /FOREIGN qualifier. Instead, the user's process receives an I/O error notification if it has I/O outstanding when a bus reset occurs.

  If possible, avoid mounting foreign devices on multihost SCSI buses. If foreign devices are mounted on the shared bus, make sure that systems on that bus do not assert a SCSI bus reset while I/O is being done to foreign devices.
- When the ARC console is enabled on a multihost SCSI bus, it sets the SCSI target ID for all local host adapters to 7. This setting causes a SCSI ID conflict if there is already a host or device on a bus at ID 7. A conflict of this type typically causes the bus, and possibly all the systems on the bus, to hang.

  The ARC console is used to access certain programs, such as the KZPSA configuration utilities. If you must run the ARC console, first disconnect the system from multihost SCSI buses and from buses that have a device at SCSI ID 7.
- Any SCSI bus resets that occur when a system powers up, boots, or shuts down cause other systems on the SCSI bus to log errors and display OPCOM messages. This is expected behavior and does not indicate a problem.
- Abruptly halting a system on a multihost SCSI bus (for example, by pressing Ctrl/P on the console) may leave the KZPAA SCSI adapter in a state that can interfere with the operation of the other host on the bus. You should initialize, boot, or continue an abruptly halted system as soon as possible after it has been halted.
- All I/O to a disk drive must be stopped while its microcode is updated. This typically requires more precautions in a multihost environment than are needed in a single-host environment. Refer to Section A.7.6.3 for the necessary procedures.

- The EISA Configuration Utility (ECU) causes a large number of SCSI bus resets. These resets cause the other system on the SCSI bus to pause while its I/O subsystem recovers. It is suggested (though not required) that both systems on a shared SCSI bus be shut down when the ECU is run.

OpenVMS Cluster systems also place one restriction on the SCSI quorum disk, whether the disk is located on a single-host SCSI bus or a multihost SCSI bus. The SCSI quorum disk must support tagged command queuing (TCQ). This is required because of the special handling that quorum I/O receives in the OpenVMS SCSI drivers.

This restriction is not expected to be significant, because all disks on a multihost SCSI bus must support tagged command queuing (see Section A.7.7), and because quorum disks are normally not used on single-host buses.

A.7.4 Troubleshooting

The following sections describe troubleshooting tips for solving common problems in an OpenVMS Cluster system that uses a SCSI interconnect.

A.7.4.1 Termination Problems

Verify that two terminators are on every SCSI interconnect (one at each end of the interconnect). The BA350 enclosure, the BA356 enclosure, the DWZZ*x*, and the KZ*xxx* adapters have internal terminators that are not visible externally (see Section A.4.4.)

A.7.4.2 Booting or Mounting Failures Caused by Incorrect Configurations

OpenVMS automatically detects configuration errors described in this section and prevents the possibility of data loss that could result from such configuration errors, either by bugchecking or by refusing to mount a disk.

A.7.4.2.1 Bugchecks During the Bootstrap Process For versions prior to OpenVMS Alpha Version 7.2, there are three types of configuration errors that can cause a bugcheck during booting. The bugcheck code is `VAXCLUSTER, Error detected by OpenVMS Cluster software`.

When OpenVMS boots, it determines which devices are present on the SCSI bus by sending an inquiry command to every SCSI ID. When a device receives the inquiry, it indicates its presence by returning data that indicates whether it is a disk, tape, or processor.

Some processor devices (host adapters) answer the inquiry without assistance from the operating system; others require that the operating system be running. The adapters supported in OpenVMS Cluster systems require the operating system to be running. These adapters, with the aid of OpenVMS, pass information in their response to the inquiry that allows the recipient to detect the following configuration errors:

- Different controller device names on the same SCSI bus

  Unless a port allocation class is being used, the OpenVMS device name of each adapter on the SCSI bus must be identical (for example, all named PKC0). Otherwise, the OpenVMS Cluster software cannot coordinate the host's accesses to storage (see Section A.6.2 and Section A.6.3).

  OpenVMS can check this automatically because it sends the controller letter in the inquiry response. A booting system receives this response, and it compares the remote controller letter with the local controller letter. If a mismatch is detected, then an OPCOM message is printed, and the system stops with an VAXCLUSTER bugcheck to prevent the possibility of data

loss. See the description of the NOMATCH error in the Help Message utility. (To use the Help Message utility for NOMATCH, enter HELP/MESSAGE NOMATCH at the DCL prompt.)

- Different or zero allocation class values.

  Each host on the SCSI bus must have the same nonzero disk allocation class value, or matching port allocation class values. Otherwise, the OpenVMS Cluster software cannot coordinate the host's accesses to storage (see Section A.6.2 and Section A.6.3).

  OpenVMS is able to automatically check this, because it sends the needed information in the inquiry response. A booting system receives this response, and compares the remote value with the local value. If a mismatch or a zero value is detected, then an OPCOM message is printed, and the system stops with a VAXCLUSTER bugcheck to prevent the possibility of data loss. See the description of the ALLODIFF and ALLOZERO errors in the Help Message utility.

- Unsupported processors

  There may be processors on the SCSI bus that are not running OpenVMS or that do not return the controller name or allocation class information needed to validate the configuration. If a booting system receives an inquiry response and the response does not contain the special OpenVMS configuration information, then an OPCOM message is printed and an VAXCLUSTER bugcheck occurs. See the description of the CPUNOTSUP error in the Help Message utility.

  If your system requires the presence of a processor device on a SCSI bus, then refer to the CPUNOTSUP message description in the Help Message utility for instructions on the use of a special SYSGEN parameter, SCSICLUSTER_P*n* for this case.

A.7.4.2.2 Failure to Configure Devices In OpenVMS Alpha Version 7.2, SCSI devices on a misconfigured bus (as described in Section A.7.4.2.1) are not configured. Instead, error messages that describe the incorrect configuration are displayed.

A.7.4.2.3 Mount Failures There are two types of configuration error that can cause a disk to fail to mount.

First, when a system boots from a disk on the shared SCSI bus, it may fail to mount the system disk. This happens if there is another system on the SCSI bus that is already booted, and the other system is using a different device name for the system disk. (Two systems will disagree about the name of a device on the shared bus if their controller names or allocation classes are misconfigured, as described in the previous section.) If the system does not first execute one of the bugchecks described in the previous section, then the following error message is displayed on the console:

```
%SYSINIT-E- error when mounting system device, retrying..., status = 007280B4
```

The decoded representation of this status is:

```
VOLALRMNT,  another volume of same label already mounted
```

This error indicates that the system disk is already mounted in what appears to be another drive in the OpenVMS Cluster system, so it is not mounted again. To solve this problem, check the controller letters and allocation class values for each node on the shared SCSI bus.

Second, SCSI disks on a shared SCSI bus will fail to mount on both systems unless the disk supports tagged command queuing (TCQ). This is because TCQ provides a command-ordering guarantee that is required during OpenVMS Cluster state transitions.

OpenVMS determines that another processor is present on the SCSI bus during autoconfiguration, using the mechanism described in Section A.7.4.2.1. The existence of another host on a SCSI bus is recorded and preserved until the system reboots.

This information is used whenever an attempt is made to mount a non-TCQ device. If the device is on a multihost bus, the mount attempt fails and returns the following message:

```
%MOUNT-F-DRVERR, fatal drive error.
```

If the drive is intended to be mounted by multiple hosts on the same SCSI bus, then it must be replaced with one that supports TCQ.

Note that the first processor to boot on a multihost SCSI bus does not receive an inquiry response from the other hosts because the other hosts are not yet running OpenVMS. Thus, the first system to boot is unaware that the bus has multiple hosts, and it allows non-TCQ drives to be mounted. The other hosts on the SCSI bus detect the first host, however, and they are prevented from mounting the device. If two processors boot simultaneously, it is possible that they will detect each other, in which case neither is allowed to mount non-TCQ drives on the shared bus.

A.7.4.3 Grounding

Having excessive ground offset voltages or exceeding the maximum SCSI interconnect length can cause system failures or degradation in performance. See Section A.7.8 for more information about SCSI grounding requirements.

A.7.4.4 Interconnect Lengths

Adequate signal integrity depends on strict adherence to SCSI bus lengths. Failure to follow the bus length recommendations can result in problems (for example, intermittent errors) that are difficult to diagnose. See Section A.4.3 for information on SCSI bus lengths.

A.7.5 SCSI Arbitration Considerations

Only one initiator (typically, a host system) or target (typically, a peripheral device) can control the SCSI bus at any one time. In a computing environment where multiple targets frequently contend for access to the SCSI bus, you could experience throughput issues for some of these targets. This section discusses control of the SCSI bus, how that control can affect your computing environment, and what you can do to achieve the most desirable results.

Control of the SCSI bus changes continually. When an initiator gives a command (such as READ) to a SCSI target, the target typically disconnects from the SCSI bus while it acts on the command, allowing other targets or initiators to use the bus. When the target is ready to respond to the command, it must regain control of the SCSI bus. Similarly, when an initiator wishes to send a command to a target, it must gain control of the SCSI bus.

If multiple targets and initiators want control of the bus simultaneously, bus ownership is determined by a process called arbitration, defined by the SCSI standard. The default arbitration rule is simple: control of the bus is given to the requesting initiator or target that has the highest unit number.

The following sections discuss some of the implications of arbitration and how you can respond to arbitration situations that affect your environment.

A.7.5.1 Arbitration Issues in Multiple-Disk Environments

When the bus is not very busy, and bus contention is uncommon, the simple arbitration scheme is adequate to perform I/O requests for all devices on the system. However, as initiators make more and more frequent I/O requests, contention for the bus becomes more and more common. Consequently, targets with lower ID numbers begin to perform poorly, because they are frequently blocked from completing their I/O requests by other users of the bus (in particular, targets with the highest ID numbers). If the bus is sufficiently busy, low-numbered targets may never complete their requests. This situation is most likely to occur on systems with more than one initiator because more commands can be outstanding at the same time.

The OpenVMS system attempts to prevent low-numbered targets from being completely blocked by monitoring the amount of time an I/O request takes. If the request is not completed within a certain period, the OpenVMS system stops sending new requests until the tardy I/Os complete. While this algorithm does not ensure that all targets get equal access to the bus, it does prevent low-numbered targets from being totally blocked.

A.7.5.2 Solutions for Resolving Arbitration Problems

If you find that some of your disks are not being serviced quickly enough during periods of heavy I/O, try some or all of the following, as appropriate for your site:

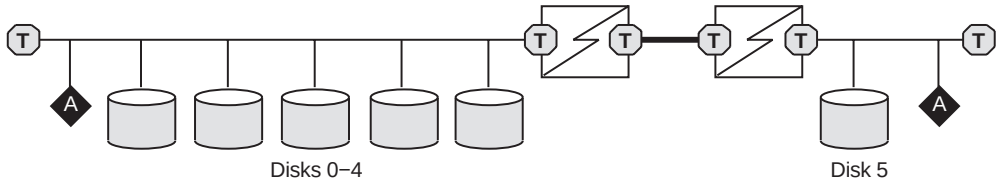
- Obtain the DWZZH–05 SCSI hub and enable its fair arbitration feature.
- Assign the highest ID numbers to those disks that require the fastest response time.
- Spread disks across more SCSI buses.
- Keep disks that need to be accessed only by a single host (for example, page and swap disks) on a nonshared SCSI bus.

Another method that might provide for more equal servicing of lower and higher ID disks is to set the host IDs to the lowest numbers (0 and 1) rather than the highest. When you use this method, the host cannot gain control of the bus to send new commands as long as any disk, including those with the lowest IDs, need the bus. Although this option is available to improve fairness under some circumstances, this configuration is less desirable in most instances, for the following reasons:

- It can result in lower total throughput.
- It can result in timeout conditions if a command cannot be sent within a few seconds.
- It can cause physical configuration difficulties. For example, StorageWorks shelves such as the BA350 have no slot to hold a disk with ID 7, but they do have a slot for a disk with ID 0. If you change the host to ID 0, you must remove a disk from slot 0 in the BA350, but you cannot move the disk to ID 7. If you have two hosts with IDs 0 and 1, you cannot use slot 0 or 1 in the BA350. (Note, however, that you *can* have a disk with ID 7 in a BA353.)

A.7.5.3 Arbitration and Bus Isolators

Any active device, such as a DWZZ*x*, that connects bus segments introduces small delays as signals pass through the device from one segment to another. Under some circumstances, these delays can be another cause of unfair arbitration. For example, consider the following configuration, which could result in disk servicing problems (starvation) under heavy work loads:



ZK–7913A–GE

Although disk 5 has the highest ID number, there are some circumstances under which disk 5 has the lowest access to the bus. This can occur after one of the lower-numbered disks has gained control of the bus and then completed the operation for which control of the bus was needed. At this point, disk 5 does not recognize that the bus is free and might wait before trying to arbitrate for control of the bus. As a result, one of the lower-numbered disks, having become aware of the free bus and then submitting a request for the bus, will gain control of the bus.

If you see this type of problem, the following suggestions can help you reduce its severity:

- Try to place all disks on the same bus segment.
- If placing all disks on the same bus segment is not possible (for example if you have both some RZ28 disks by themselves and an HSZ*xx*, try to use a configuration that has only one isolator between any pair of disks.
- If your configuration requires two isolators between a pair of disks (for example, to meet distance requirements), try to balance the number of disks on each bus segment.
- Follow the suggestions in Section A.7.5.2 to reduce the total traffic on the logical bus.

A.7.6 Removal and Insertion of SCSI Devices While the OpenVMS Cluster System is Operating

With proper procedures, certain SCSI devices can be removed from or inserted onto an active SCSI bus without disrupting the ongoing operation of the bus. This capability is referred to as **hot plugging**. Hot plugging can allow a suitably configured OpenVMS Cluster system to continue to run while a failed component is replaced. Without hot plugging, it is necessary to make the SCSI bus inactive and remove power from all the devices on the SCSI bus before any device is removed from it or inserted onto it.

In a SCSI OpenVMS Cluster system, hot plugging requires that all devices on the bus have certain electrical characteristics and be configured appropriately on the SCSI bus. Successful hot plugging also depends on strict adherence to the procedures described in this section. These procedures ensure that the hot-plugged device is inactive and that active bus signals are not disturbed.

Hot Plugging for SCSI Buses Behind a Storage Controller

This section describes hot-plugging procedures for devices that are on the same SCSI bus as the host that is running OpenVMS. The procedures are different for SCSI buses that are behind a storage controller, such as the HSZ*xx*. Refer to the storage controller documentation for the procedures to hot plug devices that they control.

A.7.6.1 Terminology for Describing Hot Plugging

The terms shown in bold in this section are used in the discussion of hot plugging rules and procedures.

- A SCSI bus **segment** consists of two terminators, the electrical path forming continuity between them, and possibly, some attached stubs. Bus segments can be connected together by bus isolators (for example, DWZZ*x*), to form a **logical SCSI bus** or just a **SCSI bus**.
- There are two types of connections on a segment: **bussing connections**, which break the path between two terminators, and **stubbing connections**, which disconnect all or part of a stub.
- A device is **active** on the SCSI bus when it is asserting one or more of the bus signals. A device is **inactive** when it is not asserting any bus signals.

  The segment attached to a bus isolator is inactive when all devices on that segment, except possibly the bus isolator, are inactive.
- A port on a bus isolator has **proper termination** when it is attached to a segment that is terminated at both ends and has TERMPWR in compliance with SCSI–2 requirements.

A.7.6.2 Rules for Hot Plugging

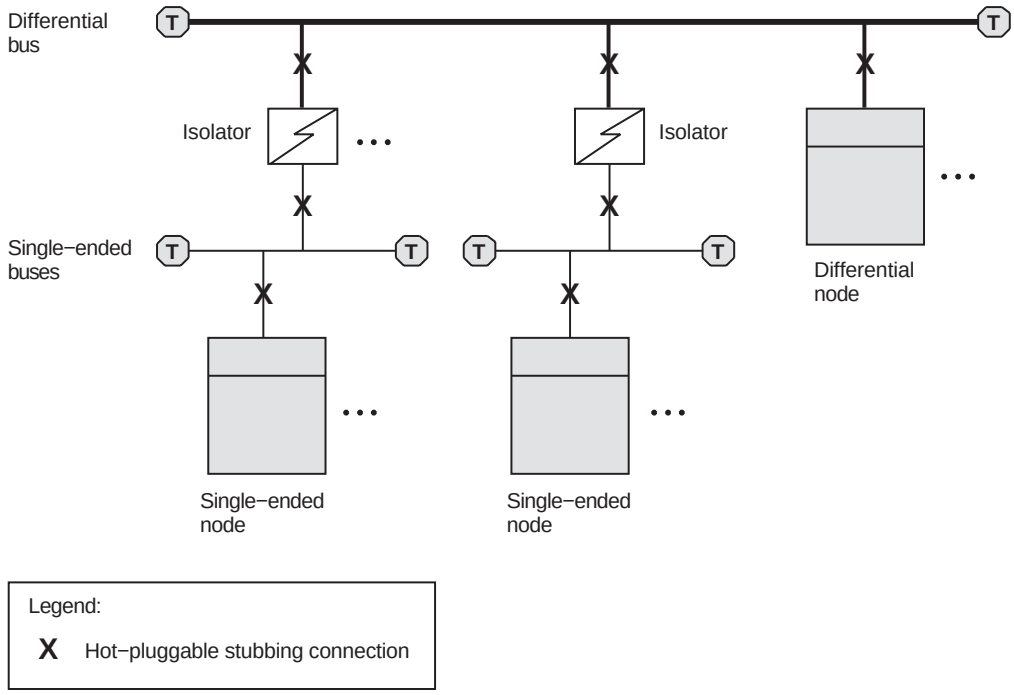
Follow these rules when planning for and performing hot plugging:

- The device to be hot plugged, and all other devices on the same segment, must meet the electrical requirements described in Annex A, Section A.4, of the SCSI-3 Parallel Interface (SPI) Standard, working draft X3T10/855D.[1] The SPI document places requirements on the receivers and terminators on the segment where the hot plugging is being performed, and on the transceivers, TERMPWR, termination, and power/ground/signal sequencing, of the device that is being hot plugged.
- Hot plugging must occur only at a stubbing connection.

  This implies that a hot-plugged device can make only one connection to the SCSI bus, the device must not provide termination for the SCSI bus, and the device's connection must not exceed the maximum stub length, as shown in Figure A–3. An example of a SCSI bus topology showing the valid hot plugging connections is illustrated in Figure A–13.
- Take precautions to ensure that electrostatic discharge (ESD) does not damage devices or disrupt active signals on the SCSI bus. You should take such precautions during the process of disconnecting and connecting, as well as during the time that SCSI bus conductors are exposed.

[1] Referring to this draft standard is necessary because the SCSI–2 standard does not adequately specify the requirements for hot plugging.

Figure A–13 SCSI Bus Topology



ZK−8842A−GE

- Take precaution to ensure that ground offset voltages do not pose a safety hazard and will not interfere with SCSI bus signaling, especially in single-ended configurations. The procedures for measuring and eliminating ground offset voltages are described in Section A.7.8.
- The device that is hot plugged must be inactive during the disconnection and connection operations. Otherwise, the SCSI bus may hang.[2]

  Note

  Ideally, a device will also be inactive whenever its power is removed, for the same reason.

  The procedures for ensuring that a device is inactive are described in Section A.7.6.3.
- A quorum disk must not be hot plugged. This is because there is no mechanism for stopping the I/O to a quorum disk, and because the replacement disk will not contain the correct quorum file.

  The OpenVMS Cluster system must be reconfigured to remove a device as a quorum disk before that device is removed from the bus. The procedure for accomplishing this is described in *OpenVMS Cluster Systems*.

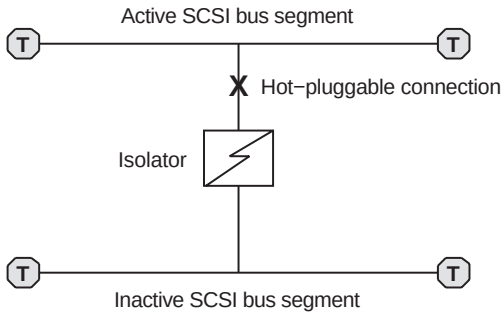
[2] OpenVMS will eventually detect a hung bus and reset it, but this problem may first temporarily disrupt OpenVMS Cluster operations.

An alternate method for increasing the availability of the quorum disk is to use an HSZ*xx* mirror set as the quorum disk. This would allow a failed member to be replaced while maintaining the quorum disk functionality.

- Disks must be dismounted logically before removing or replacing them in a hot-plugging operation. This is required to ensure that the disk is inactive and to ensure the integrity of the file system.
- The DWZZ*x* must be powered up when it is inserted into an active SCSI bus and should remain powered up at all times while it is attached to the active SCSI bus. This is because the DWZZ*x* can disrupt the operation of the attached segments when it is powering up or down.
- The segment attached to a bus isolator must be maintained in the inactive state whenever the other port on the bus isolator is terminated improperly. This is required because an improperly terminated bus isolator port may pass erroneous signals to the other port.

  Thus, for a particular hot-plugging operation, one of the segments attached to a bus isolator must be designated as the (potentially) active segment, and the other must be maintained in the inactive state, as illustrated in Figure A–14. The procedures for ensuring that a segment is inactive are described in Section A.7.6.3.

Figure A–14 Hot Plugging a Bus Isolator



ZK−8843A−GE

Note that, although a bus isolator may have more than one stubbing connection and thus be capable of hot plugging on each of them, only one segment can be the active segment for any particular hot-plugging operation.

- Take precautions to ensure that the only electrical conductor that contacts a connector pin is its mate. These precautions must be taken during the process of disconnecting and connecting as well as during the time the connector is disconnected.
- Devices must be replaced with devices of the same type. That is, if any system in the OpenVMS Cluster configures a SCSI ID as a DK or MK device, then that SCSI ID must contain only DK or MK devices, respectively, for as long as that OpenVMS Cluster member is running.

Different implementations of the same device type can be substituted (for example, an RZ26L can be replaced with an RZ28B). Note that the system will not recognize the change in device type until an attempt is made to mount the new device. Also, note that host-based shadowing continues to require that all members of a shadow set be the same device type.

- SCSI IDs that are empty when a system boots must remain empty as long as that system is running. This rule applies only if there are multiple processors on the SCSI bus and the MSCP server is loaded on any of them. (The MSCP server is loaded when the MSCP_LOAD system parameter is set to 1).

  This is required to ensure that nodes on the SCSI bus use their direct path to the disk rather than the served path. When the new device is configured on a system (using SYSMAN IO commands), that system serves it to the second system on the shared SCSI bus. The second system automatically configures the new device by way of the MSCP served path. Once this occurs, the second system will be unable to use its direct SCSI path to the new device because failover from an MSCP served path to a direct SCSI path is not implemented.

A.7.6.3 Procedures for Ensuring That a Device or Segment is Inactive

Use the following procedures to ensure that a device or a segment is inactive:

- To ensure that a disk is inactive:
  1. Dismount the disk on all members of the OpenVMS Cluster system.
  2. Ensure that any I/O that can occur to a dismounted disk is stopped, for example:
     - Disable the disk as a quorum disk.
     - Allocate the disk (using the DCL command ALLOCATE) to block further mount or initialization attempts.
     - Disable console polling by all halted hosts on the logical SCSI bus (by setting the console variable SCSI_POLL to OFF and entering the INIT command).
     - Ensure that no host on the logical SCSI bus is executing power-up or initialization self-tests, booting, or configuring the SCSI bus (using SYSMAN IO commands).
- To ensure that an HSZ*xx* controller is inactive:
  1. Dismount all of the HSZ*xx* virtual disks on all members of the OpenVMS Cluster system.
  2. Shut down the controller, following the procedures in the *HS Family of Array Controllers User's Guide.*
  3. Power down the HSZ*xx* (optional).
- To ensure that a host adapter is inactive:
  1. Halt the system.
  2. Power down the system, or set the console variable SCSI_POLL to OFF and then enter the INIT command on the halted system. This ensures that the system will not poll or respond to polls.
- To ensure that a segment is inactive, follow the preceding procedures for every device on the segment.

A.7.6.4 Procedure for Hot Plugging StorageWorks SBB Disks

To remove an SBB (storage building block) disk from an active SCSI bus, use the following procedure:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance.
2. Follow the procedure in Section A.7.6.3 to make the disk inactive.
3. Squeeze the clips on the side of the SBB, and slide the disk out of the StorageWorks shelf.

To plug an SBB disk into an active SCSI bus, use the following procedure:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance.
2. Ensure that the SCSI ID associated with the device (either by jumpers or by the slot in the StorageWorks shelf) conforms to the following:
   - The SCSI ID is unique for the logical SCSI bus.
   - The SCSI ID is already configured as a DK device on all of the following:
     - Any member of the OpenVMS Cluster system that already has that ID configured
     - Any OpenVMS processor on the same SCSI bus that is running the MSCP server
3. Slide the SBB into the StorageWorks shelf.
4. Configure the disk on OpenVMS Cluster members, if required, using SYSMAN IO commands.

A.7.6.5 Procedure for Hot Plugging HSZ*xx*

To remove an HSZ*xx* controller from an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance.
2. Follow the procedure in Section A.7.6.3 to make the HSZ*xx* inactive.
3. The HSZ*xx* can be powered down, but it must remain plugged in to the power distribution system to maintain grounding.
4. Unscrew and remove the differential triconnector from the HSZ*xx*.
5. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.

To plug an HSZ*xx* controller into an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance. Also, ensure that the ground offset voltages between the HSZ*xx* and all components that will be attached to it are within the limits specified in Section A.7.8.
2. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.

3. Power up the HSZ*xx* and ensure that the disk units associated with the HSZ*xx* conform to the following:
   - The disk units are unique for the logical SCSI bus.
   - The disk units are already configured as DK devices on the following:
     - Any member of the OpenVMS Cluster system that already has that ID configured
     - Any OpenVMS processor on the same SCSI bus that is running the MSCP server
4. Ensure that the HSZ*xx* will make a legal stubbing connection to the active segment. (The connection is legal when the triconnector is attached directly to the HSZ*xx* controller module, with no intervening cable.)
5. Attach the differential triconnector to the HSZ*xx*, using care to ensure that it is properly aligned. Tighten the screws.
6. Configure the HSZ*xx* virtual disks on OpenVMS Cluster members, as required, using SYSMAN IO commands.

A.7.6.6 Procedure for Hot Plugging Host Adapters

To remove a host adapter from an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance.
2. Verify that the connection to be broken is a stubbing connection. If it is not, then do not perform the hot plugging procedure.
3. Follow the procedure in Section A.7.6.3 to make the host adapter inactive.
4. The system can be powered down, but it must remain plugged in to the power distribution system to maintain grounding.
5. Remove the "Y" cable from the host adapter's single-ended connector.
6. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.
7. Do *not* unplug the adapter from the host's internal bus while the host remains powered up.

   At this point, the adapter has disconnected from the SCSI bus. To remove the adapter from the host, first power down the host, then remove the adapter from the host's internal bus.

To plug a host adapter into an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance. Also, ensure that the ground offset voltages between the host and all components that will be attached to it are within the limits specified in Section A.7.8.
2. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.
3. Ensure that the host adapter will make a legal stubbing connection to the active segment (the stub length must be within allowed limits, and the host adapter must not provide termination to the active segment).

4. Plug the adapter into the host (if it is unplugged).
5. Plug the system into the power distribution system to ensure proper grounding. Power up, if desired.
6. Attach the "Y" cable to the host adapter, using care to ensure that it is properly aligned.

A.7.6.7 Procedure for Hot Plugging DWZZ*x* Controllers

Use the following procedure to remove a DWZZ*x* from an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance.
2. Verify that the connection to be broken is a stubbing connection. If it is not, then do not perform the hot plugging procedure.
3. Do not power down the DWZZ*x*. This can disrupt the operation of the attached SCSI bus segments.
4. Determine which SCSI bus segment will remain active after the disconnection. Follow the procedure in Section A.7.6.3 to make the other segment inactive.

   When the DWZZ*x* is removed from the active segment, the inactive segment must remain inactive until the DWZZ*x* is also removed from the inactive segment, or until proper termination is restored to the DWZZ*x* port that was disconnected from the active segment.
5. The next step depends on the type of DWZZ*x* and the segment that is being hot plugged, as follows:

DWZZ*x* Type	Condition	Action
SBB[1]	Single-ended segment will remain active.	Squeeze the clips on the side of the SBB, and slide the DWZZ*x* out of the StorageWorks shelf.
SBB[1]	Differential segment will remain active.	Unscrew and remove the differential triconnector from the DWZZ*x*.
Table top	Single-ended segment will remain active.	Remove the "Y" cable from the DWZZ*x*'s single-ended connector.
Table top	Differential segment will remain active.	Unscrew and remove the differential triconnector from the DWZZ*x*.

[1]SSB is the StorageWorks abbreviation for storage building block.

6. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.

To plug a DWZZ*x* into an active SCSI bus:

1. Use an ESD grounding strap that is attached either to a grounding stud or to unpainted metal on one of the cabinets in the system. Refer to the system installation procedures for guidance. Also, ensure that the ground offset voltages between the DWZZ*x* and all components that will be attached to it are within the limits specified in Section A.7.8.
2. Protect all exposed connector pins from ESD and from contacting any electrical conductor while they are disconnected.

3. Ensure that the DWZZ*x* will make a legal stubbing connection to the active segment (the stub length must be within allowed limits, and the DWZZ*x* must not provide termination to the active segment).
4. The DWZZ*x* must be powered up. The SCSI segment that is being added must be attached and properly terminated. All devices on this segment must be inactive.
5. The next step depends on the type of DWZZ*x*, and which segment is being hot plugged, as follows:

DWZZ*x* Type	Condition	Action
SBB[1]	Single-ended segment is being hot plugged.	Slide the DWZZ*x* into the StorageWorks shelf.
SBB[1]	Differential segment is being hot plugged.	Attach the differential triconnector to the DWZZ*x*, using care to ensure that it is properly aligned. Tighten the screws.
Table top	Single-ended segment is being hot plugged.	Attach the "Y" cable to the DWZZ*x*, using care to ensure that it is properly aligned.
Table top	Differential segment is being hot plugged.	Attach the differential triconnector to the DWZZ*x*, using care to ensure that it is properly aligned. Tighten the screws.

[1]SSB is the StorageWorks abbreviation for storage building block.

6. If the newly attached segment has storage devices on it, then configure them on OpenVMS Cluster members, if required, using SYSMAN IO commands.

A.7.7 OpenVMS Requirements for Devices Used on Multihost SCSI OpenVMS Cluster Systems

At this time, the only devices approved for use on multihost SCSI OpenVMS Cluster systems are those listed in Table A–2. While not specifically approved for use, other disk devices might be used in a multihost OpenVMS Cluster system when they conform to the following requirements:

- Support for concurrent multi-initiator I/O.
- Proper management for the following states or conditions on a per-initiator basis:
  - Synchronous negotiated state and speed
  - Width negotiated state
  - Contingent Allegiance and Unit Attention conditions
- Tagged command queuing. This is needed to provide an ordering guarantee used in OpenVMS Cluster systems to ensure that I/O has been flushed. The drive must implement queuing that complies with Section 7.8.2 of the SCSI–2 standard, which says (in part):

  "...All commands received with a simple queue tag message prior to a command received with an ordered queue tag message, *regardless of initiator*, shall be executed before that command with the ordered queue tag message." (Emphasis added.)

- Support for command disconnect.
- A reselection timeout procedure compliant with Option b of Section 6.1.4.2 of the SCSI–2 standard. Furthermore, the device shall implement a reselection retry algorithm that limits the amount of bus time spent attempting to reselect a nonresponsive initiator.
- Automatic read reallocation enabled (ARRE) and automatic write reallocation enabled (AWRE) (that is, drive-based bad block revectoring) to prevent multiple hosts from unnecessarily revectoring the same block. To avoid data corruption, it is essential that the drive comply with Section 9.3.3.6 of the SCSI–2 Standard, which says (in part):

  "...The automatic reallocation shall then be performed only if the target *successfully recovers the data*." (Emphasis added.)
- Storage devices should not supply TERMPWR. If they do, then it is necessary to apply configuration rules to ensure that there are no more than four sources of TERMPWR on a segment.

Finally, if the device or any other device on the same segment will be hot plugged, then the device must meet the electrical requirements described in Section A.7.6.2.

A.7.8 Grounding Requirements

This section describes the grounding requirements for electrical systems in a SCSI OpenVMS Cluster system.

Improper grounding can result in voltage differentials, called ground offset voltages, between the enclosures in the configuration. Even small ground offset voltages across the SCSI interconnect (as shown in step 3 of Table A–8) can disrupt the configuration and cause system performance degradation or data corruption.

Table A–8 describes important considerations to ensure proper grounding.

Table A–8 Steps for Ensuring Proper Grounding

Step	Description
1	Ensure that site power distribution meets all local electrical codes.
2	Inspect the entire site power distribution system to ensure that: • All outlets have power ground connections. • A grounding prong is present on all computer equipment power cables. • Power-outlet neutral connections are not actual ground connections. • All grounds for the power outlets are connected to the same power distribution panel. • All devices that are connected to the same circuit breaker as the computer equipment are UL® or IEC approved.

(continued on next page)

Table A–8 (Cont.) Steps for Ensuring Proper Grounding

Step	Description
3	If you have difficulty verifying these conditions, you can use a hand-held multimeter to measure the ground offset voltage between any two cabinets. To measure the voltage, connect the multimeter leads to unpainted metal on each enclosure. Then determine whether the voltage exceeds the following allowable ground offset limits: • Single-ended signaling: 50 millivolts (maximum allowable offset) • Differential signaling: 800 millivolts (maximum allowable offset) The multimeter method provides data for only the moment it is measured. The ground offset values may change over time as additional devices are activated or plugged into the same power source. To ensure that the ground offsets remain within acceptable limits over time, Compaq recommends that you have a power survey performed by a qualified electrician.
4	If you are uncertain about the grounding situation or if the measured offset exceeds the allowed limit, Compaq recommends that a qualified electrician correct the problem. It may be necessary to install grounding cables between enclosures to reduce the measured offset.
5	If an unacceptable offset voltage was measured and a ground cable was installed, then measure the voltage again to ensure it is less than the allowed limits. If not, an electrician must determine the source of the ground offset voltage and reduce or eliminate it.

B

MEMORY CHANNEL Technical Summary

This appendix contains information about MEMORY CHANNEL, a high-performance cluster interconnect technology. MEMORY CHANNEL, which was introduced in OpenVMS Alpha Version 7.1, supports several configurations.

This chapter contains the following sections:

Section	Content
Product Overview	High-level introduction to the MEMORY CHANNEL product and its benefits, hardware components, and configurations.
Technical Overview	More in-depth technical information about how MEMORY CHANNEL works.

B.1 Product Overview

MEMORY CHANNEL is a high-performance cluster interconnect technology for PCI-based Alpha systems. With the benefits of very low latency, high bandwidth, and direct memory access, MEMORY CHANNEL complements and extends the unique ability of an OpenVMS Cluster to work as a single, virtual system.

MEMORY CHANNEL offloads internode cluster traffic (such as lock management communication) from existing interconnects—CI, DSSI, FDDI, and Ethernet—so that they can process storage and network traffic more effectively. MEMORY CHANNEL significantly increases throughput and decreases the latency associated with traditional I/O processing.

Any application that must move large amounts of data among nodes will benefit from MEMORY CHANNEL. It is an optimal solution for applications that need to pass data quickly, such as real-time and transaction processing. MEMORY CHANNEL also improves throughput in high-performance databases and other applications that generate heavy OpenVMS Lock Manager traffic.

B.1.1 MEMORY CHANNEL Features

MEMORY CHANNEL technology provides the following features:

- **Offers excellent price/performance.**

  With several times the CI bandwidth, MEMORY CHANNEL provides a 100 MB/s interconnect with minimal latency. MEMORY CHANNEL architecture is designed for the industry-standard PCI bus.

- **Requires no change to existing applications.**

  MEMORY CHANNEL works seamlessly with existing cluster software, so that no change is necessary for existing applications. The new MEMORY CHANNEL drivers, PMDRIVER and MCDRIVER, integrate with the Systems Communication Services layer of OpenVMS Clusters in the same way as existing port drivers. Higher layers of cluster software are unaffected.

- **Offloads CI, DSSI, and the LAN in SCSI clusters.**

  You cannot connect storage directly to MEMORY CHANNEL.

  While MEMORY CHANNEL is not a replacement for CI and DSSI, when used in combination with those interconnects, it offloads their node-to-node traffic. This enables them to be dedicated to storage traffic, optimizing communications in the entire cluster.

  When used in a cluster with SCSI and LAN interconnects, MEMORY CHANNEL offloads node-to-node traffic from the LAN, enabling it to handle more TCP/IP or DECnet traffic.

- **Provides fail-separately behavior.**

  When a system failure occurs, MEMORY CHANNEL nodes behave like any failed node in an OpenVMS Cluster. The rest of the cluster continues to perform until the failed node can rejoin the cluster.

B.1.2 MEMORY CHANNEL Version 2.0 Features

When first introduced in OpenVMS Version 7.1, MEMORY CHANNEL supported a maximum of four nodes in a 10-foot radial topology. Communication occurred between one sender-receiver pair at a time. MEMORY CHANNEL Version 1.5 introduced support for eight nodes, a new adapter (CCMAA–BA), time stamps on all messages, and more robust performance.

MEMORY CHANNEL Version 2.0 provides the following new capabilities:

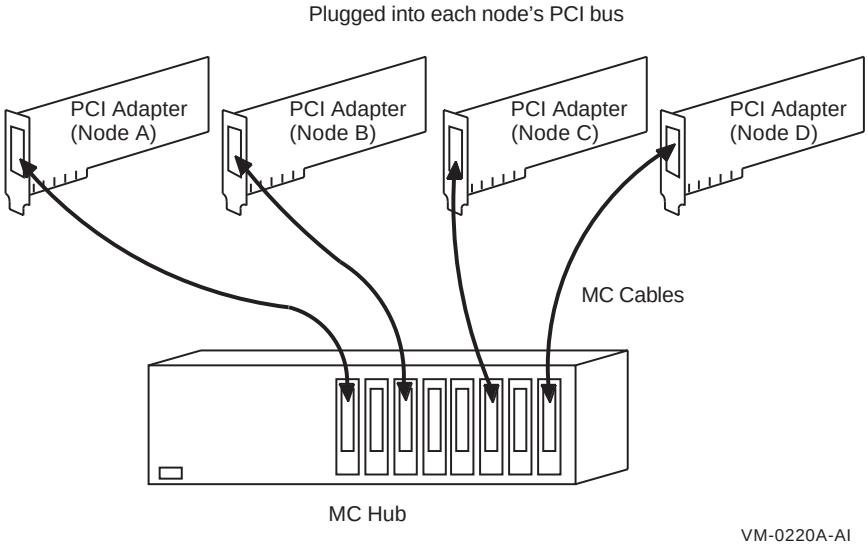
- Support for a new adapter (CCMAB-AA) and new hubs (CCMHB-AA and CCMHB-BA)
- Support for simultaneous communication between four sender-receiver pairs
- Support for longer cables for a radial topology up to 3 km

B.1.3 Hardware Components

A MEMORY CHANNEL cluster is joined together by a hub, a desktop-PC sized unit which provides a connection among systems. The hub is connected to a system's PCI adapter by a link cable. Figure B–1 shows all three hardware components required by a node to support MEMORY CHANNEL:

- A PCI-to-MEMORY CHANNEL adapter
- A link cable
- A port in a MEMORY CHANNEL hub (except for a two-node configuration in which the cable connects just two PCI adapters.)

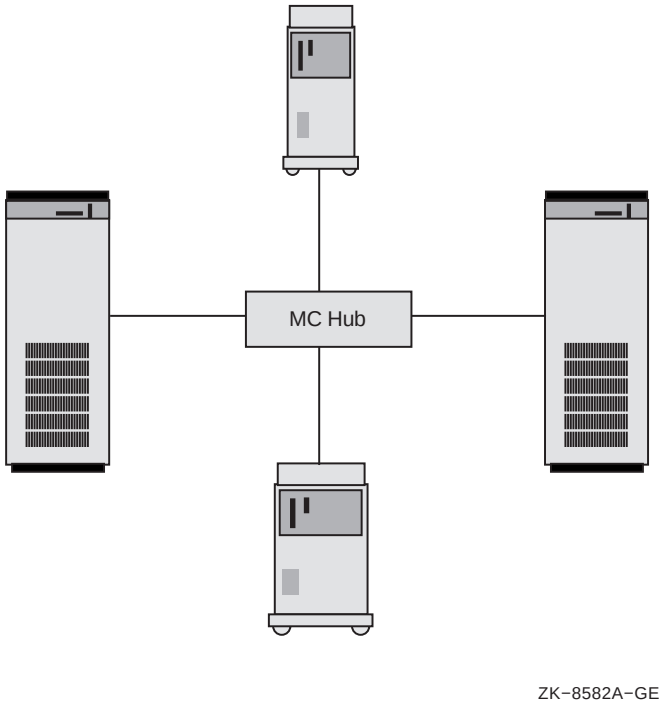
Figure B–1 MEMORY CHANNEL Hardware Components



The PCI adapter pictured in Figure B–1 has memory mapping logic that enables each system to communicate with the others in the MEMORY CHANNEL cluster.

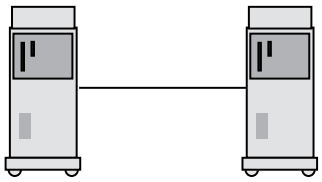
Figure B–2 shows an example of four-node MEMORY CHANNEL cluster with a hub at its center.

Figure B–2 Four-Node MEMORY CHANNEL Cluster



A MEMORY CHANNEL hub is not required in clusters that contain only two nodes. In a two-node configuration like the one shown Figure B–3, the same adapters and cable are used, and one of the PCI adapters serves as a virtual hub. You can continue to use the adapters and cable if you expand to a larger configuration later.

Figure B–3 Virtual Hub MEMORY CHANNEL Cluster



B.1.4 Backup Interconnect for High-Availability Configurations

MEMORY CHANNEL requires a central hub in configurations of three or more nodes. The MEMORY CHANNEL hub contains active, powered electronic components. In the event of a hub failure, resulting from either a power shutdown or component failure, the MEMORY CHANNEL interconnect ceases operation. This type of failure does not occur with the other cluster interconnects, such as CI, DSSI, and most LAN configurations.

Compaq therefore recommends that customers with MEMORY CHANNEL configurations who have high availability requirements consider using one of the following configurations to provide a second backup interconnect:

- In most cases a second interconnect can easily be configured by enabling the LAN (Ethernet or FDDI) for clustering. FDDI and 100 Mb/s Ethernet usually provide acceptable interconnect performance in the event of MEMORY CHANNEL failure. (See *OpenVMS Cluster Systems* and *Guidelines for OpenVMS Cluster Configurations* for details about how to enable the LAN for clustering.)
- CI and DSSI interconnects automatically act as a backup for MEMORY CHANNEL.
- A configuration with two MEMORY CHANNEL interconnects provides the highest possible performance as well as continued operation if one MEMORY CHANNEL interconnect fails.

B.1.5 Software Requirements

The use of MEMORY CHANNEL imposes certain requirements on memory and on your choice of diagnostic tools.

B.1.5.1 Memory Requirements

MEMORY CHANNEL consumes memory during normal operations. Each system in your MEMORY CHANNEL cluster must have at least 128 MB of memory.

B.1.5.2 Large-Memory Systems' Use of NPAGEVIR Parameter

On systems containing very large amounts of nonpaged pool memory, MEMORY CHANNEL may be unable to complete initialization. If this happens, the console displays the following message repeatedly:

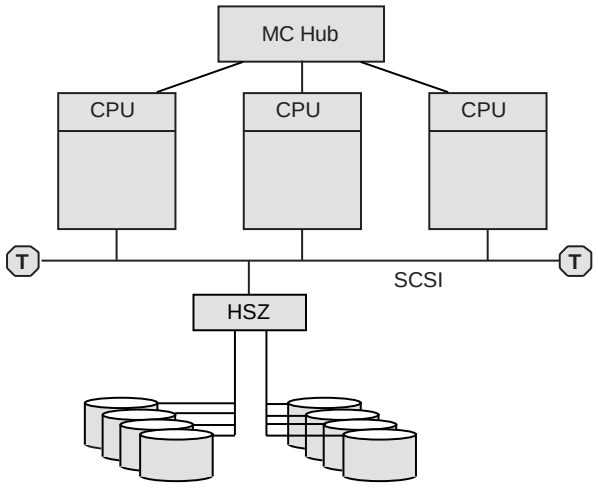
```
Hub timeout - reinitializing adapter
```

To fix this problem, examine the value of the SYSGEN parameter NPAGEVIR. If its value is greater than 1 gigabyte, consider lowering it to about half of that. Thereafter, a reboot of your system should allow the MEMORY CHANNEL to complete initialization.

B.1.6 Configurations

Figure B–4 shows a basic MEMORY CHANNEL cluster that uses the SCSI interconnect for storage. This configuration provides two advantages: high performance on the MEMORY CHANNEL interconnect and low cost on the SCSI interconnect.

Figure B–4 MEMORY CHANNEL- and SCSI-Based Cluster

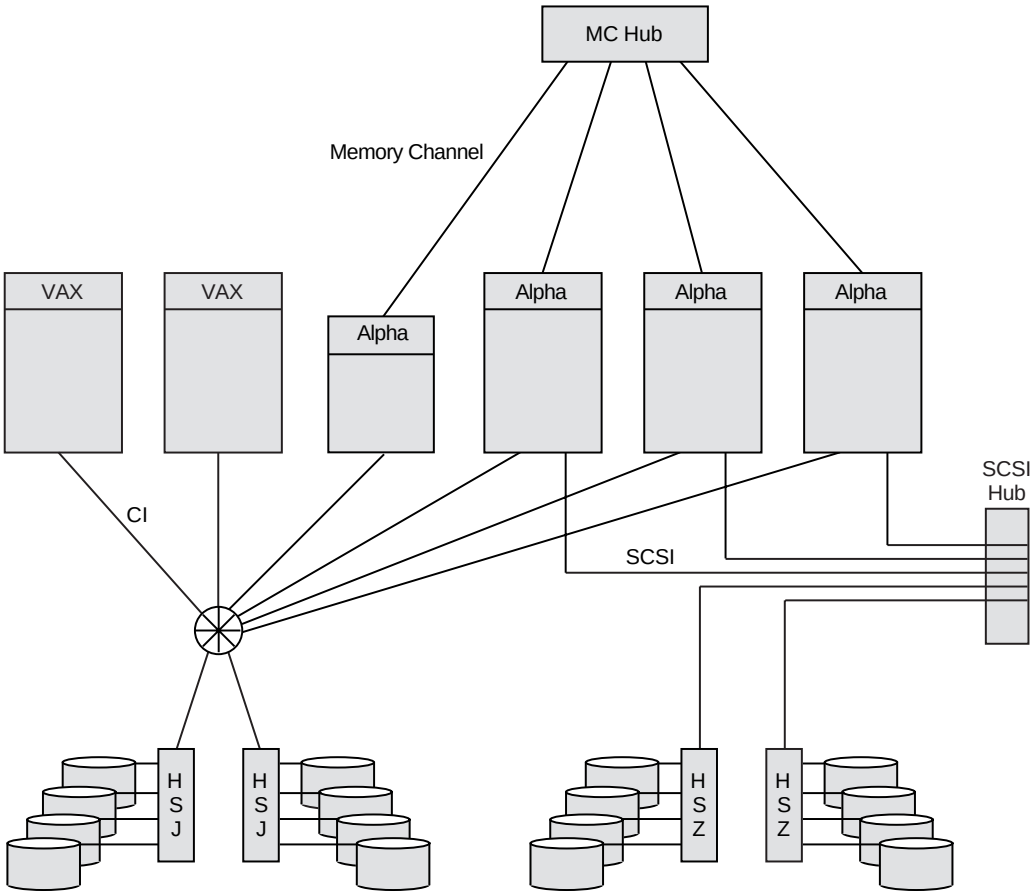


ZK–8777A–GE

In a configuration like the one shown in Figure B–4, the MEMORY CHANNEL interconnect handles internode communication while the SCSI bus handles storage communication.

You can integrate MEMORY CHANNEL with your current systems. Figure B–5 shows an example of how to add MEMORY CHANNEL to a mixed-architecture CI- and SCSI-based cluster. In this example, the BI- and XMI-based VAX systems are joined in the same CI cluster with the PCI-based Alpha MEMORY CHANNEL systems.

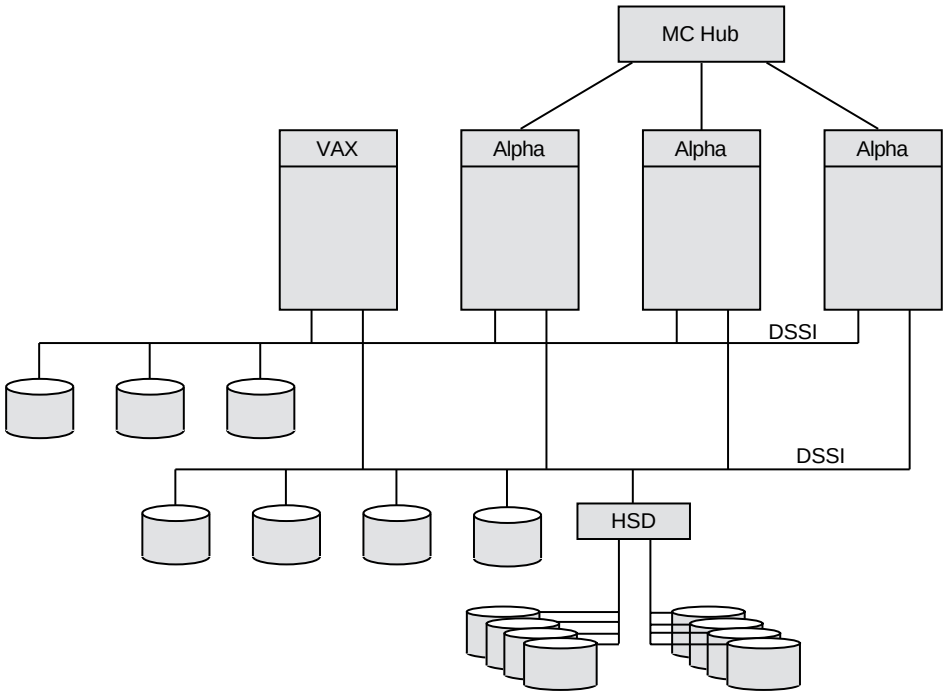
Figure B–5 MEMORY CHANNEL CI- and SCSI-Based Cluster



Because the MEMORY CHANNEL interconnect is not used for storage and booting, you must provide access to a boot device through one of the other interconnects. To use Figure B–5 as an example, one of the CI-based disks would be a good choice for a boot device because all nodes have direct access to it over the CI.

MEMORY CHANNEL can also be integrated into an existing DSSI cluster, as shown in Figure B–6.

Figure B–6 MEMORY CHANNEL DSSI-Based Cluster



As Figure B–6 shows, the three MEMORY CHANNEL systems and the VAX system have access to the storage that is directly connected to the DSSI interconnect as well as to the SCSI storage attached to the HSD controller. In this configuration, MEMORY CHANNEL handles the Alpha internode traffic, while the DSSI handles the storage traffic.

B.1.6.1 Configuration Support

MEMORY CHANNEL supports the platforms and configurations shown in Table B–1.

Table B–1 MEMORY CHANNEL Configuration Support

Requirement	Description
Configuration	MEMORY CHANNEL supports the following configurations: • Up to eight nodes per MEMORY CHANNEL hub. • For two-hub configurations, up to two PCI adapters per node; each adapter must be connected to a different hub. • For two-node configurations, no hub is required.

(continued on next page)

Table B–1 (Cont.) MEMORY CHANNEL Configuration Support

Requirement	Description
Cables	MEMORY CHANNEL supports the following cables: • Copper cables up to a 10-m (32.8 ft) radial topology • Fiber-optic cables from Compaq up to a 30-m (98.4 ft) radial topology; fiber-optic cables from other vendors, up to a 3-km (1.8 miles) radial topology
Host systems	MEMORY CHANNEL supports the following systems: • AlphaServer 8400 • AlphaServer 8200 • AlphaServer 4100 • AlphaServer 2100A • AlphaServer 1200 • AlphaServer 800

Note

You can configure a computer in an OpenVMS Cluster system with both a MEMORY CHANNEL Version 1.5 hub and a MEMORY CHANNEL Version 2.0 hub. However, the version number of the adapter and the cables must match the hub's version number for MEMORY CHANNEL to function properly.

In other words, you must use MEMORY CHANNEL Version 1.5 adapters with the MEMORY CHANNEL Version 1.5 hub and MEMORY CHANNEL Version 1.5 cables. Similarly, you must use MEMORY CHANNEL Version 2.0 adapters with the MEMORY CHANNEL Version 2.0 hub and MEMORY CHANNEL Version 2.0 cables.

B.2 Technical Overview

This section describes in more technical detail how MEMORY CHANNEL works.

B.2.1 Comparison With Traditional Networks and SMP

You can think of MEMORY CHANNEL as a form of "stretched SMP bus" that supports enough physical distance to interconnect up to eight systems. However, MEMORY CHANNEL differs from an SMP environment where multiple CPUs can directly access the same physical memory. MEMORY CHANNEL requires each node to maintain its own physical memory, even though the nodes share MEMORY CHANNEL global address space.

MEMORY CHANNEL fills a price/performance gap between the high performance of SMP systems and traditional packet-based networks. Table B–2 shows a comparison among the characteristics of SMP, MEMORY CHANNEL, and standard networks.

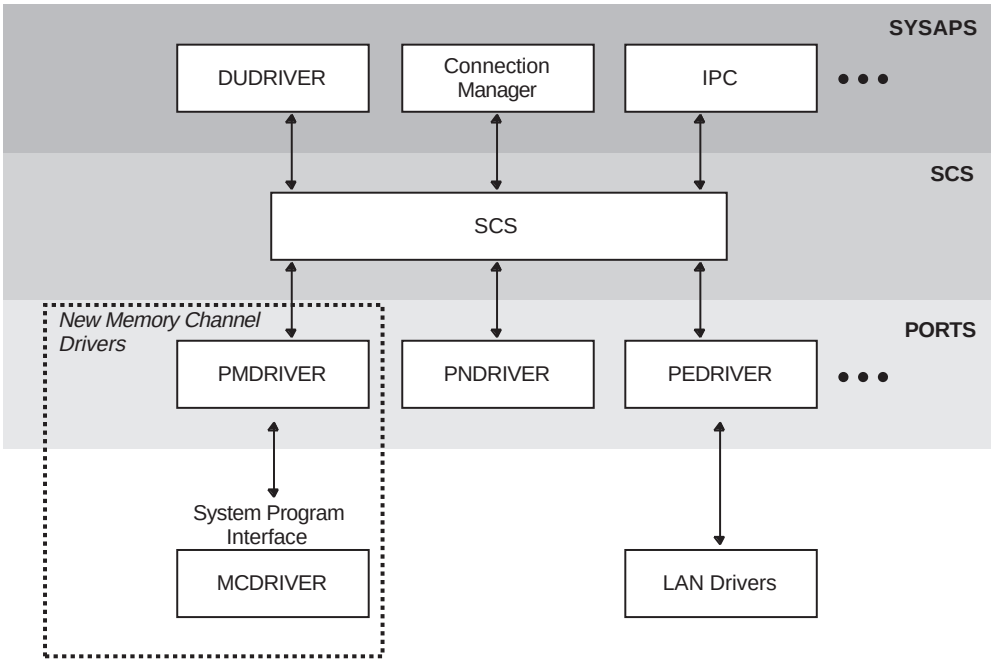
Table B–2 Comparison of SMP, MEMORY CHANNEL, and Standard Networks

Characteristics	SMP	MEMORY CHANNEL	Standard Networking
Bandwidth (MB/s)	1000+	100+	10+
Latency (ms/simplest message)	0.5	Less than 5	About 300
Overhead (ms/simplest message)	0.5	Less than 5	About 250
Hardware communication model	Shared memory	Memory-mapped	Message passing
Hardware communication primitive	Store to memory	Store to memory	Network packet
Hardware support for broadcast	n/a	Yes	Sometimes
Hardware support for synchronizaton	Yes	Yes	No
Hardware support for node hot swap	No	Yes	Yes
Software communication model	Shared memory	Fast messages, shared memory	Messages
Communication model for errors	Not recoverable	Recoverable	Recoverable
Supports direct user mode communication	Yes	Yes	No
Typical physical interconnect technology	Backplane etch	Parallel copper cables	Serial fiber optics
Physical interconnect error rate	Extremely low order: less than one per year	Extremely low order: less than one per year	Low order: several per day
Hardware interconnect method	Special purpose connector and logic	Standard I/O bus adapter (PCI)	Standard I/O bus adapter (PCI and others)
Distance between nodes (m)	0.3	20 (copper) or 60 (fiber-optic) in a hub configuration and 10 (copper) or 30 (fiber-optic) in a two-node configuration	50–1000
Number of nodes	1	8	Hundreds
Number of processors	6–12	8 times the maximum number of CPUs in an SMP system	Thousands
Failure model	Fail together	Fail separately	Fail separately

B.2.2 MEMORY CHANNEL in the OpenVMS Cluster Architecture

As Figure B–7 shows, MEMORY CHANNEL functionality has been implemented in the OpenVMS Cluster architecture just below the System Communication Services layer. This design ensures that no changes are required to existing applications because higher layers of OpenVMS Cluster software are unchanged.

Figure B–7 OpenVMS Cluster Architecture and MEMORY CHANNEL



MEMORY CHANNEL software consists of two new drivers:

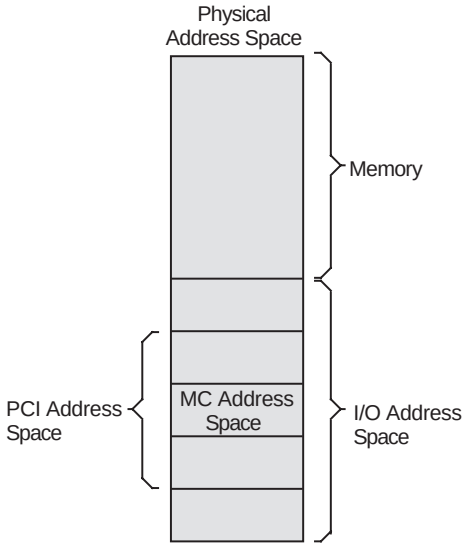
Driver	Description
PMDRIVER	Emulates a cluster port driver.
MCDRIVER	Provides MEMORY CHANNEL services and an interface to MEMORY CHANNEL hardware.

B.2.3 MEMORY CHANNEL Addressing

In a MEMORY CHANNEL configuration, a section of system physical address space is shared among all nodes. When a system writes data to this address space, the MEMORY CHANNEL hardware also performs a global write so that this data is stored in the memories of other systems. In other words, when a node's CPU writes data to the PCI address space occupied by the MEMORY CHANNEL adapter, the data is sent across the MEMORY CHANNEL interconnect to the other nodes. The other nodes' PCI adapters map this data into their own memory. This infrastructure enables a write to an I/O address on one system to get mapped to a physical address on the other system. The next two figures explain this in more detail.

Figure B–8 shows how MEMORY CHANNEL global address space is addressed in physical memory.

Figure B–8 Physical Memory and I/O Address Space



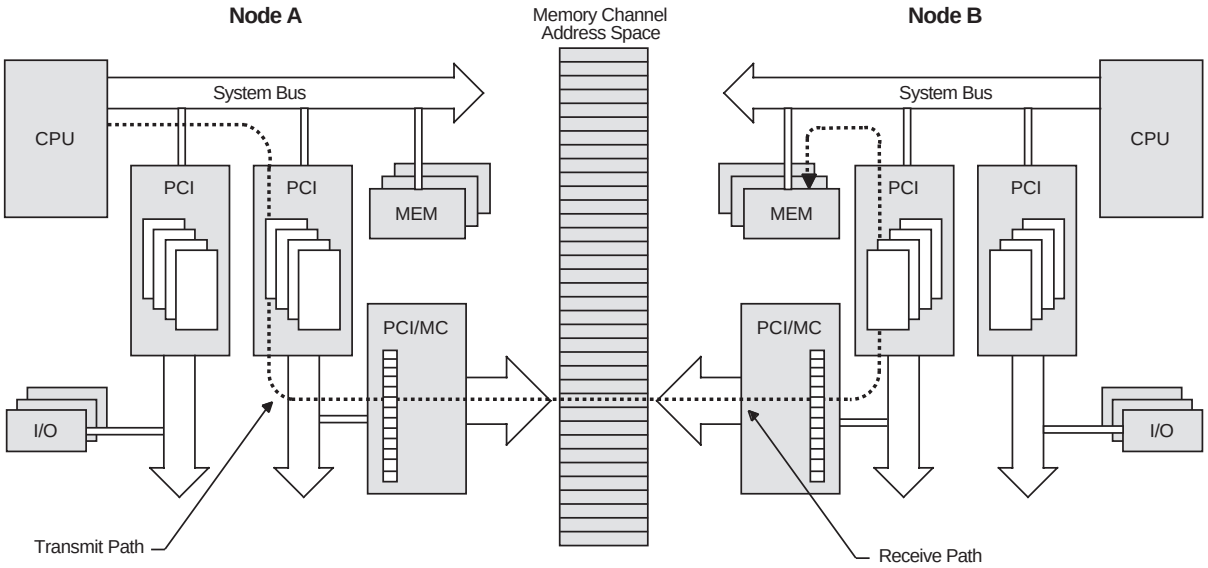
ZK–8778A–GE

Figure B–8 shows the typical address space of a system, divided into physical memory and I/O address space. Within the PCI I/O address space, MEMORY CHANNEL consumes 128 to 512 MB of address space. Therefore, the MEMORY CHANNEL PCI adapter can be addressed within this space, and the CPU can write data to it.

Every system in a MEMORY CHANNEL cluster allocates this address space for MEMORY CHANNEL data and communication. By using this address space, a CPU can perform global writes to the memories of other nodes.

To explain global writes more fully, Figure B–9 shows the internal bus architecture of two nodes, node A and node B.

Figure B–9 MEMORY CHANNEL Bus Architecture



ZK–8776A–GE

In the example shown in Figure B–9, node A is performing a global write to node B's memory, in the following sequence:

1. Node A's CPU performs a write to MEMORY CHANNEL address space, which is part of PCI address space. The write makes its way through the PCI bus to the PCI/MEMORY CHANNEL adapter and out on the MEMORY CHANNEL interconnect.
2. Node B's PCI adapter receives the data, which is picked up by its PCI bus and DMA-mapped to memory.

If all nodes in the cluster agree to address MEMORY CHANNEL global address space in the same way, they can virtually "share" the same address space and the same data. This is why MEMORY CHANNEL address space is depicted as a common, central address space in Figure B–9.

MEMORY CHANNEL global address space is divided into pages of 8 KB (8,192 bytes). These are called MC pages. These 8 KB pages can be mapped similarly among systems.

The "shared" aspect of MEMORY CHANNEL global address space is set up using the page control table, or PCT, in the PCI adapter. The PCT has attributes that can be set for each MC page. Table B–3 explains these attributes.

Table B–3 MEMORY CHANNEL Page Attributes

Attribute	Description
Broadcast	Data is sent to all systems or, with a node ID, data is sent to only the specified system.
Loopback	Data that is sent to the other nodes in a cluster is also written to memory by the PCI adapter in the transmitting node. This provides message order guarantees and a greater ability to detect errors.
Interrupt	Specifies that if a location is written in this MC page, it generates an interrupt to the CPU. This can be used for notifying other nodes.
Suppress transmit/receive after error	Specifies that if an error occurs on this page, transmit and receive operations are not allowed until the error condition is cleared.
ACK	A write to a page causes each receiving system's adapter to respond with an ACK (acknowledge), ensuring that a write (or other operation) has occurred on remote nodes without interrupting their hosts. This is used for error checking and error recovery.

B.2.4 MEMORY CHANNEL Implementation

MEMORY CHANNEL software comes bundled with the OpenVMS Cluster software. After setting up the hardware, you configure the MEMORY CHANNEL software by responding to prompts in the CLUSTER_CONFIG.COM procedure. A prompt asks whether you want to enable MEMORY CHANNEL for node-to-node communications for the local computer. By responding "Yes", MC_SERVICES_P2, the system parameter that controls whether MEMORY CHANNEL is in effect, is set to 1. This setting causes the driver, PMDRIVER, to be loaded and the default values for the other MEMORY CHANNEL system parameters to take effect.

For a description of all the MEMORY CHANNEL system parameters, refer to the *OpenVMS Cluster Systems* manual.

For more detailed information about setting up the MEMORY CHANNEL hub, link cables, and PCI adapters, see the *MEMORY CHANNEL User's Guide*, order number EK-PCIRM-UG.

C

CI-to-PCI Adapter (CIPCA) Support

This appendix describes the CI-to-PCI adapter (CIPCA) which was introduced in OpenVMS Alpha Version 6.2–1H2 and is supported on all subsequent versions, except OpenVMS Version 7.0. The CIPCA adapter supports specific Alpha servers and OpenVMS Cluster configurations.

This appendix contains the following sections:

- CIPCA overview (Section C.1)
- Technical specifications (Section C.2)
- Configuration support and restrictions (Section C.3)
- Installation requirements (Section C.4)
- DECevent for Analyzing CIPCA Errors (Section C.5)
- Performance recommendations (Section C.6)

C.1 CIPCA Overview

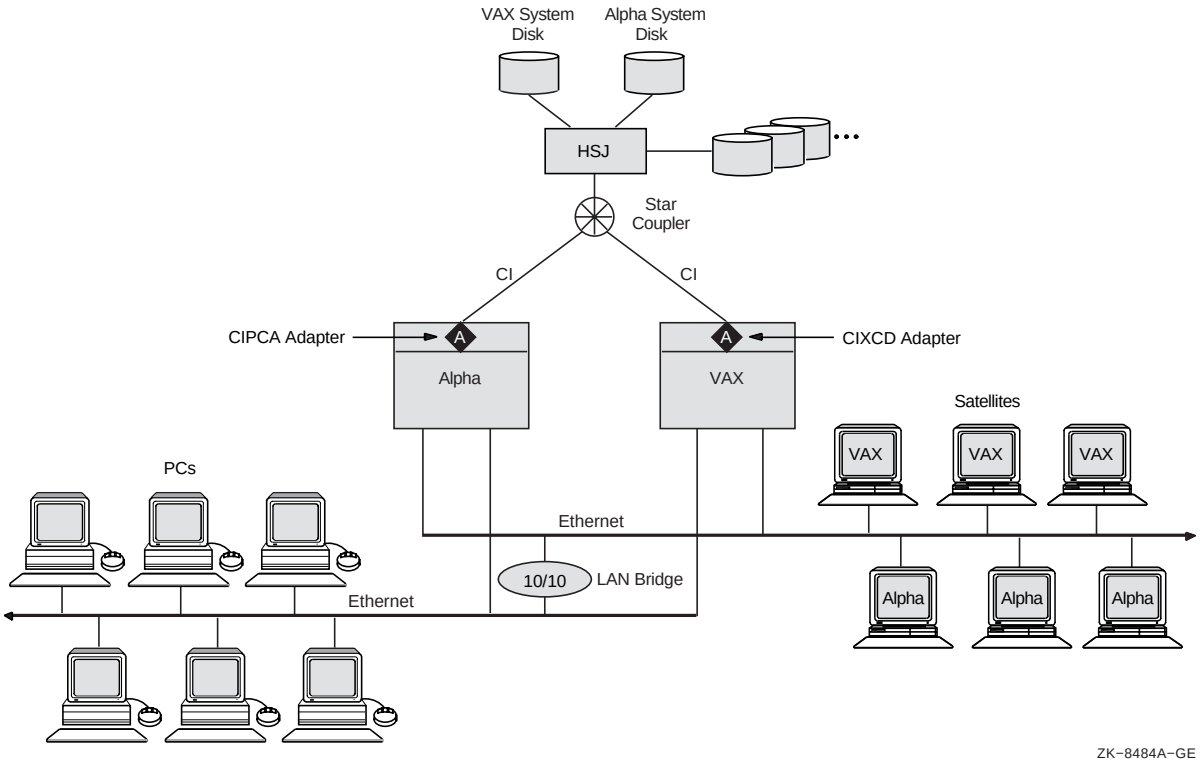
The CIPCA adapter, developed in partnership with CMD Technologies, enables Alpha servers with PCI buses or with a PCI bus and an EISA bus to connect to the CI. The CIPCA adapter provides the following features and benefits:

Feature	Benefit
Lower entry cost and more configuration choices	If you require midrange compute power for your business needs, CIPCA enables you to integrate midrange Alpha servers into your existing CI cluster.
High-end Alpha speed and power	If you require maximum compute power, you can use the CIPCA with both the AlphaServer 8200 systems and AlphaServer 8400 systems that have PCI and EISA I/O subsystems.
Cost-effective Alpha migration path	If you want to add Alpha servers to an existing CI VAXcluster, CIPCA provides a cost-effective way to start migrating to a mixed-architecture cluster in the price/performance range that you need.

Feature	Benefit
Advantages of the CI	The CIPCA connects to the CI, which offers the following advantages: • High speed to accommodate larger processors and I/O-intensive applications. • Efficient, direct access to large amounts of storage. • Minimal CPU overhead for communication. CI adapters are intelligent interfaces that perform much of the communication work in OpenVMS Cluster systems. • High availability through redundant, independent data paths, because each CI adapter is connected with two pairs of CI cables. • Multiple access paths to disks and tapes.

Figure C–1 shows an example of a mixed-architecture CI OpenVMS Cluster that has two servers: an Alpha and a VAX.

Figure C–1 CIPCA in a Mixed-Architecture OpenVMS Cluster



As Figure C–1 shows, you can use the CIPCA adapter to connect an Alpha server to a CI OpenVMS Cluster that contains a VAX server with a CIXCD (or CIBCA-B) adapter. This enables you to smoothly integrate an Alpha server into a cluster that previously comprised only high-end VAX systems.

Figure C–2 shows another example of a configuration that uses the CIPCA to connect systems with the CI. In this example, each Alpha has two CIPCA adapters that allow connectivity to multiple CI star couplers and HSJ storage controllers for I/O load balancing or for OpenVMS shadow-set member isolation. Also, the Alpha systems are connected to a high-speed FDDI interconnect that provides additional connectivity for PC clients and OpenVMS satellites.

Figure C–2 CIPCA in an Alpha OpenVMS Cluster

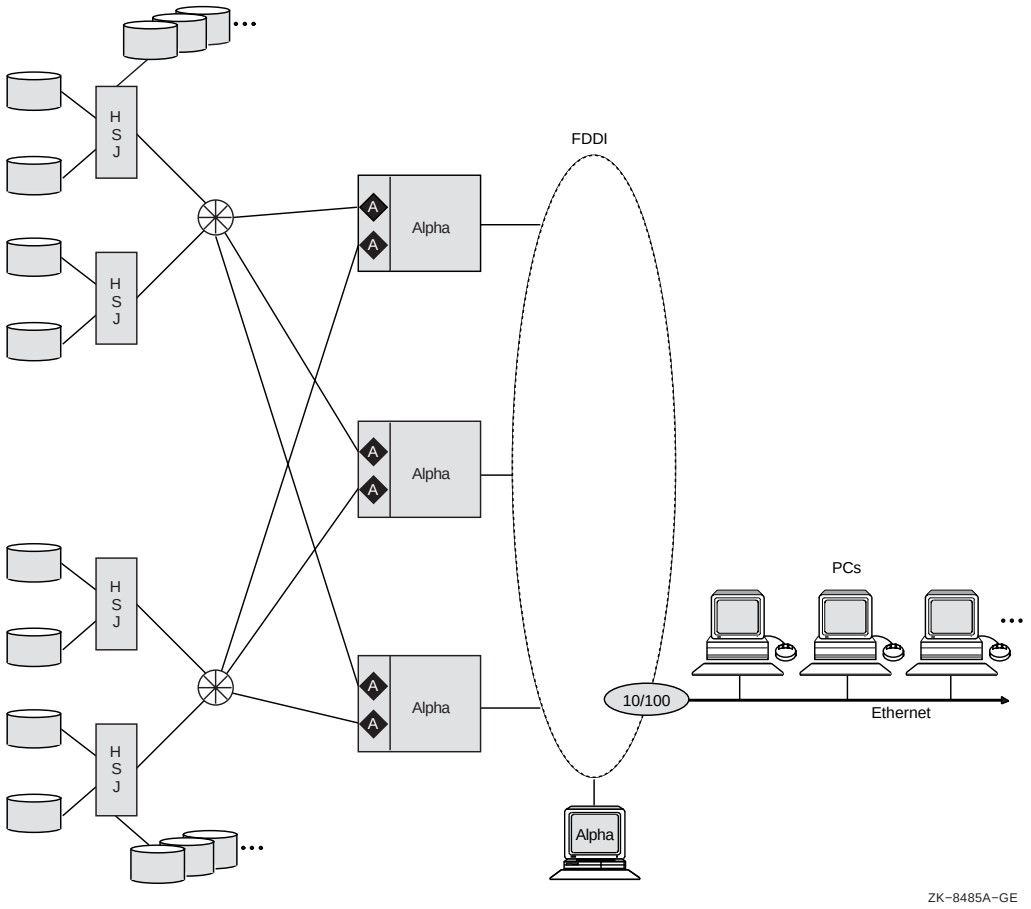


Figure C–1 and Figure C–2 show that the CIPCA makes the performance, availability, and large storage access of the CI available to a wide variety of users. The CI has a high maximum throughput. Both the PCI-based CIPCA and the XMI based CIXCD are highly intelligent microprocessor-controlled adapters that consume minimal CPU overhead.

Because the effective throughput of the CI bus is high, the CI interconnect is not likely to be a bottleneck. In large configurations like the one shown in Figure C–2, multiple adapters and CI connections provide excellent availability and throughput.

Although not shown in Figure C–1 and Figure C–2, you can increase availablity by placing disks on a SCSI interconnect between a pair of HSJ controllers and connecting each HSJ to the CI.

C.2 Technical Specifications

The CIPCA is a two-slot optional adapter. Two CIPCA models are available, the CIPCA-AA and the CIPCA-BA.

The CIPCA-AA was introduced first. It requires one PCI backplane slot and one EISA backplane slot. The EISA slot supplies only power (not bus signals) to the CIPCA. The CIPCA-AA is suitable for older systems with a limited number of PCI slots.

The CIPCA-BA requires two PCI slots and is intended for newer systems with a limited number of EISA slots.

The CIPCA driver is named the SYS$PCAdriver. It is included in the OpenVMS operating system software.

Table C–1 shows the performance of the CIPCA in relation to the CIXCD adapter.

Table C–1 CIPCA and CIXCD Performance

Performance Metric	CIPCA	CIXCD
Read request rate (I/Os)	4900	5500
Read data Rate (MB/s)	10.6	10.5
Write request rate (I/Os)	4900	4500
Write data rate (MB/s)	9.8	5.8
Mixed request rate (I/Os)	4800	5400
Mixed data rate (MB/s)	10.8	9.2

For information about installing and operating the CIPCA, refer to the hardware manual that came with your CIPCA adapter: *CIPCA PCI-CI Adapter User's Guide*.

C.3 Configuration Support and Restrictions

The CIPCA adapter is supported by AlphaServers with PCI buses, by CI-connected VAX host systems, by storage controllers, and by the CI star coupler expander.

C.3.1 AlphaServer Support

Table C–2 describes CIPCA support on AlphaServer systems with PCI buses, including the maximum number of CIPCAs supported on each system.

Table C–2 AlphaServer Support for CIPCAs

System	Maximum CIPCAs	Comments
AlphaServer 8400	26	Can use a combination of CIPCA and CIXCD adapters, not to exceed 26. Prior to OpenVMS Version 7.1, the maximum is 10.

(continued on next page)

Table C–2 (Cont.) AlphaServer Support for CIPCAs

System	Maximum CIPCAs	Comments
AlphaServer 8200	26	Prior to OpenVMS Version 7.1, the maximum is 10.
AlphaServer 4000, 4100	3	When using three CIPCAs, one must be a CIPCA-AA and two must be CIPCA-BA.
AlphaServer 4000 plus I/O expansion module	6	When using six CIPCAs, only three can be CIPCA-AA.
AlphaServer 1200	2	First supported in OpenVMS Version 7.1-1H1.
AlphaServer 2100A	3	
AlphaServer 2000, 2100	2	Only one can be a CIPCA-BA.

C.3.2 CI-Connected Host System Compatibility

For CI-connected host systems, CIPCA is supported by any OpenVMS VAX host using CIXCD or CIBCA-B as well as by any OpenVMS Alpha server host using CIPCA or CIXCD. This means that an Alpha server using the CIPCA adapter can coexist on a CI bus with VAX systems using CIXCD and CIBCA-B CI adapters.

The maximum number of systems supported in an OpenVMS Cluster system, 96, is not affected by the use of one or more CIPCAs, although the maximum number of CI nodes is limited to 16 (see Section C.3.4).

C.3.3 Storage Controller Support

The CIPCA adapter can coexist on a CI bus with all variants of the HSC/HSJ controllers except the HSC50. Certain controllers require specific firmware and hardware, as shown in Table C–3.

Table C–3 Controller Requirements for Supporting CIPCA

Controller	Requirement
HSJ30, HSC40	HSOF Version 2.5 (or higher) firmware
HSC40, HSC70	Revision F (or higher) L109 module

C.3.4 Star Coupler Expander Support

A CI star coupler expander (CISCE) can be added to any star coupler to increase its connection capacity to 32 ports. The maximum number of CPUs that can be connected to a star coupler is 16, regardless of the number of ports.

C.3.5 Configuration Restrictions

Note the following configuration restrictions:

CIPCA-AA with EISA-Slot Link Module Rev. A01

For the CIPCA-AA adapter with the EISA-slot link module Rev. A01, use the DIP switch settings described here to prevent arbitration timeout errors. Under heavy CI loads, arbitration timeout errors can cause CI path errors and CI virtual circuit closures.

The DIP switch settings on the CIPCA-AA link module are used to specify cluster size and the node address. Follow these instructions when setting the DIP switches for link module Rev. A01 only:

- If the cluster size is set to 16, do not set a CI adapter to node address 15 on that star coupler.
- If the cluster size is set to 32, do not set a CI adapter to node address 31 on that star coupler. Also, do not set any CIPCA to node address 0 *or* do not set any CI adapter to node address 16.

These restrictions do not apply to the EISA slot link module Rev. B01 and higher or to the PCI-slot link module of the CIPCA-BA.

HSJ50 Firmware Requirement for Use of 4K CI Packets

Do not attempt to enable the use of 4K CI packets by the HSJ50 controller unless the HSJ50 firmware is Version 5.0J–3 or higher. If the HSJ50 firmware version is less than Version 5.0J–3 and 4K CI packets are enabled, data can become corrupted. If your HSJ50 firmware does not meet this requirement, contact your Compaq support representative.

C.4 Installation Requirements

When installing CIPCA adapters in your cluster, observe the following version-specific requirements.

C.4.1 Managing Bus Addressable Pool (BAP) Size

The CIPCA, CIXCD, and KFMSB adapters use bus-addressable pool (BAP). Starting with OpenVMS Version 7.1, AUTOGEN controls the allocation of BAP. After installing or upgrading the operating system, you must run AUTOGEN with the FEEDBACK qualifier. When you run AUTOGEN in this way, the following four system parameters are set:

- NPAG_BAP_MIN
- NPAG_BAP_MAX
- NPAG_BAP_MIN_PA
- NPAG_BAP_MAX_PA

The BAP allocation amount depends on the adapter type, the number of adapters, and the version of the operating system. The size of physical memory determines whether the BAP remains separate or is merged with normal, nonpaged dynamic memory (NPAGEDYN), as shown in the following table:

Table C–4 BAP Allocation by Adapter Type and OpenVMS Version

Adapter	Version 7.1	Version 7.2	Separate BAP or Merged
CIPCA	4 MB	2 MB	Separate if physical memory >1 GB; otherwise merged
CIXCD	4 MB	2 MB	Separate if physical memory >4 GB; otherwise merged
KFMSB	8 MB	4 MB	Separate if physical memory >4 GB; otherwise merged

For systems whose BAP is merged with nonpaged pool, the initial amount and maximum amount of nonpaged pool (as displayed by the DCL command SHOW MEMORY/POOL/FULL) do not match the value of the SYSGEN parameters NPAGEDYN and NPAGEVIR. Instead, the value of SYSGEN parameter NPAG_BAP_MIN is added to NPAGEDYN to determine the initial size, and the value of NPAG_BAP_MAX is added to NPAGEVIR to determine the maximum size.

Your OpenVMS system may not require as much merged pool as the sum of these SYSGEN parameters. After your system has been running a few days, use AUTOGEN with the FEEDBACK qualifier to fine-tune the amount of memory allocated for the merged, nonpaged pool.

C.4.2 AUTOCONFIGURE Restriction for OpenVMS Version 6.2-1H2 and OpenVMS Version 6.2-1H3

When you perform a normal installation boot, AUTOCONFIGURE runs automatically. AUTOCONFIGURE is run from SYS$STARTUP:VMS$DEVICE_STARTUP.COM (called from SYS$SYSTEM:STARTUP.COM), unless disabled by SYSMAN. If you are running OpenVMS Version 6.2-1H2 or OpenVMS Version 6.2-1H3 and you have customized your booting sequence, make sure that AUTOCONFIGURE runs or that you explicitly configure all CIPCA devices before SYSTARTUP_VMS.COM exits.

C.5 DECevent for Analyzing CIPCA Errors

To analyze error log files for CIPCA errors, use DECevent. The DCL command ANALYZE/ERROR_LOG has not been updated to support CIPCA and other new devices; using that command will result in improperly formatted error log entries.

Install the DECevent kit supplied on the OpenVMS Alpha CD–ROM. Then use the following DCL commands to invoke DECevent to analyze dump files:

- DIAGNOSE — Analyzes the current system error log file
- DIAGNOSE *filename* — Analyzes the error log file named *filename.sys*

For more information about using DECevent, use the DCL HELP DIAGNOSE command.

C.6 Performance Recommendations

To enhance performance, follow the recommendations that pertain to your configuration.

C.6.1 Synchronous Arbitration

CIPCA uses a new, more optimal CI arbitration algorithm called synchronous arbitration instead of the older asynchronous arbitration algorithm. The two algorithms are completely compatible. Under CI saturation conditions, both the old and new algorithms are equivalent and provide equitable round-robin access to all nodes. However, with less traffic, the new algorithm provides the following benefits:

- Reduced packet transmission latency due to reduced average CI arbitration time.
- Increased node-to-node throughput.
- Complete elimination of CI collisions that waste bandwidth and increase latency in configurations containing only synchronous arbitration nodes.

- Reduced CI collision rate in configurations with mixed synchronous and asynchronous arbitration CI nodes. The reduction is roughly proportional to the fraction of CI packets being sent by the synchronous arbitration CI nodes.

Support for synchronous arbitration is latent in the HSJ controller family. In configurations containing both CIPCAs and HSJ controllers, enabling the HSJs to use synchronous arbitration is recommended.

The HSJ CLI command to do this is:

```
CLI> SET THIS CI_ARB = SYNC
```

This command will take effect upon the next reboot of the HSJ.

C.6.2 Maximizing CIPCA Performance With an HSJ50

To maximize the performance of the CIPCA adapter with an HSJ50 controller, it is advisable to enable the use of 4K CI packets by the HSJ50. To do this, your HSJ50 firmware revision level must be at Version 5.0J–3 or higher.

Caution

Do not attempt to do this if your HSJ50 firmware revision level is not Version 5.0J–3 or higher, because data can become corrupted.

To enable the use of 4K CI packets, specify the following command at the HSJ50 console prompt:

```
CLI> SET THIS_CONTROLLER CI_4K_PACKET_CAPABILITY
```

This command takes effect when the HSJ50 is rebooted.

D

Multiple-Site OpenVMS Clusters

This appendix describes multiple-site OpenVMS Cluster configurations in which multiple nodes are located at sites separated by relatively long distances, from approximately 25 to 125 miles, depending on the technology used. This configuration was introduced in OpenVMS Version 6.2. General configuration guidelines are provided and the three technologies for connecting multiple sites are discussed. The benefits of multiple site clusters are cited and pointers to additional documentation are provided.

The information in this appendix supersedes the *Multiple-Site VMScluster Systems* addendum manual.

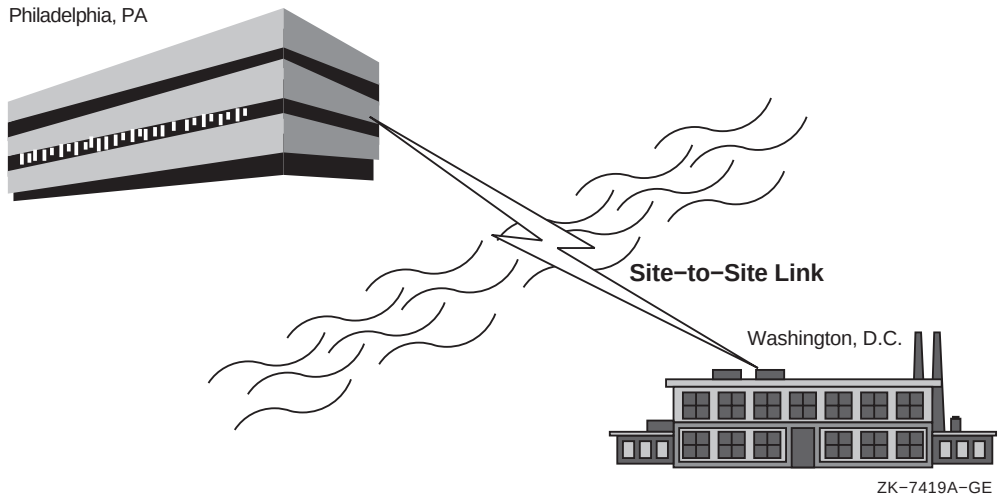
D.1 What is a Multiple-Site OpenVMS Cluster System?

A **multiple-site OpenVMS Cluster system** is an OpenVMS Cluster system in which the member nodes are located in geographically separate sites. Depending on the technology used, the distances can be as great as 150 miles.

When an organization has geographically dispersed sites, a multiple-site OpenVMS Cluster system allows the organization to realize the benefits of OpenVMS Cluster systems (for example, sharing data among sites while managing data center operations at a single, centralized location).

Figure D–1 illustrates the concept of a multiple-site OpenVMS Cluster system for a company with a manufacturing site located in Washington, D.C., and corporate headquarters in Philadelphia. This configuration spans a geographical distance of approximately 130 miles (210 km).

Figure D–1 Site-to-Site Link Between Philadelphia and Washington



D.1.1 ATM, DS3, and FDDI Intersite Links

The following link technologies between sites are approved for OpenVMS VAX and OpenVMS Alpha systems:

- Asynchronous transfer mode (ATM)
- DS3
- FDDI

High-performance local area network (LAN) technology combined with the ATM, DS3, and FDDI interconnects allows you to utilize wide area network (WAN) communication services in your OpenVMS Cluster configuration. OpenVMS Cluster systems configured with the GIGAswitch crossbar switch and ATM, DS3, or FDDI interconnects approve the use of nodes located miles apart. (The actual distance between any two sites is determined by the physical intersite cable-route distance, and not the straight-line distance between the sites.) Section D.3 describes OpenVMS Cluster systems and the WAN communications services in more detail.

Note

To gain the benefits of disaster tolerance across a multiple-site OpenVMS Cluster, use Disaster Tolerant Cluster Services for OpenVMS, a system management and software package from Compaq.

Consult your Compaq Services representative for more information.

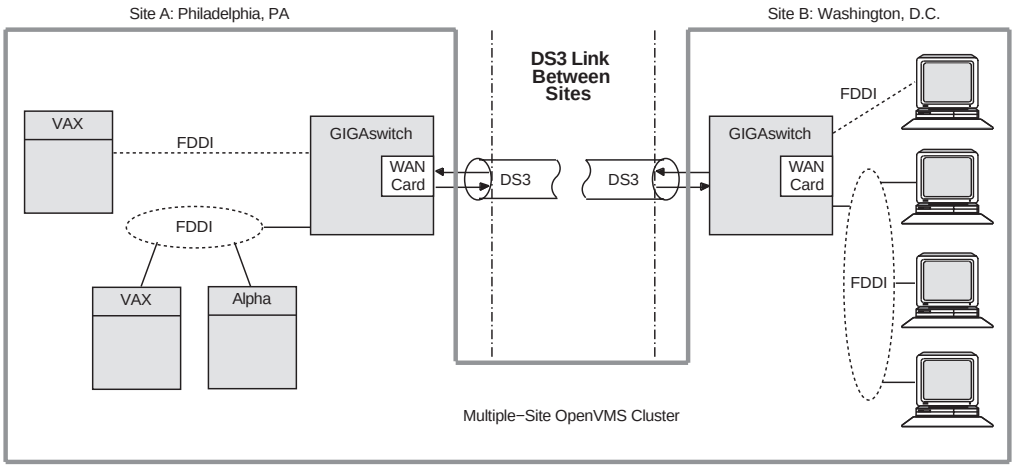
D.1.2 Benefits of Multiple-Site OpenVMS Cluster Systems

Some of the benefits you can realize with a multiple-site OpenVMS Cluster system include the following:

Benefit	Description
Remote satellites and nodes	A few systems can be remotely located at a secondary site and can benefit from centralized system management and other resources at the primary site, as shown in Figure D–2. For example, a main office data center could be linked to a warehouse or a small manufacturing site that could have a few local nodes with directly attached site-specific devices. Alternatively, some engineering workstations could be installed in an office park across the city from the primary business site.
Data center management consolidation	A single management team can manage nodes located in data centers at multiple sites.
Physical resource sharing	Multiple sites can readily share devices such as high-capacity computers, tape libraries, disk archives, or phototypesetters.
Remote archiving	Backups can be made to archival media at any site in the cluster. A common example would be to use disk or tape at a single site to back up the data for all sites in the multiple-site OpenVMS Cluster. Backups of data from remote sites can be made transparently (that is, without any intervention required at the remote site).
Increased availability	In general, a multiple-site OpenVMS Cluster provides all of the availability advantages of a LAN OpenVMS Cluster. Additionally, by connecting multiple, geographically separate sites, multiple-site OpenVMS Cluster configurations can increase the availability of a system or elements of a system in a variety of ways: • Logical volume/data availability—Volume shadowing or redundant arrays of independent disks (RAID) can be used to create logical volumes with members at both sites. If one of the sites becomes unavailable, data can remain available at the other site. • Site failover—By adjusting the VOTES system parameter, you can select a preferred site to continue automatically if the other site fails or if communications with the other site are lost. • Disaster tolerance—When combined with the software, services, and management procedures provided by the Disaster Tolerant Cluster Services for OpenVMS, you can achieve a high level of disaster tolerance. Consult your Compaq Services representative for further information.

Figure D–2 shows an OpenVMS Cluster system with satellites accessible from a remote site.

Figure D–2 Multiple-Site OpenVMS Cluster Configuration with Remote Satellites



D.1.3 General Configuration Guidelines

The same configuration rules that apply to OpenVMS Cluster systems on a LAN also apply to a multiple-site OpenVMS Cluster configuration that includes ATM, DS3, or FDDI intersite interconnect. General LAN configuration rules are stated in the following documentation:

- OpenVMS Cluster Software *Software Product Description* (SPD 29.78.*xx*)
- Chapter 8 of this manual

Some configuration guidelines are unique to multiple-site OpenVMS Clusters; these guidelines are described in Section D.3.4.

D.2 Using FDDI to Configure Multiple-Site OpenVMS Cluster Systems

Since VMS Version 5.4–3, FDDI has been the most common method to connect two distant OpenVMS Cluster sites. Using high-speed FDDI fiber-optic cables, you can connect sites with an intersite cable-route distance of up to 25 miles 40 km.[3]

You can connect sites using these FDDI methods:

- To obtain maximum performance, use a full-duplex FDDI link at 100 Mb/s both ways between GIGAswitch/FDDI bridges at each site for maximum intersite bandwidth.
- To obtain maximum availability, use a dual FDDI ring at 100 Mb/s between dual attachment stations (DAS) ports of wiring concentrators or GIGAswitch /FDDI bridges for maximum link availability.

[3] The cable route distance between sites.

- For maximum performance and availability, use two disjoint FDDI LANs, each with dedicated host adapters and full-duplex FDDI intersite links connected to GIGAswitch/FDDI bridges at each site.

Refer to the *GIGAswitch/FDDI ATM Linecard Reference Manual* for configuration information. Additional OpenVMS Cluster configuration guidelines and system management information can be found in this manual and in *OpenVMS Cluster Systems*. See the *Overview of OpenVMS Documentation* for information about ordering the current version of these manuals.

The inherent flexibility of OpenVMS Cluster systems and improved OpenVMS Cluster LAN protocols also allow you to connect multiple OpenVMS Cluster sites using the ATM or DS3 or both communications services.

D.3 Using WAN Services to Configure Multiple-Site OpenVMS Cluster Systems

This section provides an overview of the ATM and DS3 wide area network (WAN) services, describes how you can bridge an FDDI interconnect to the ATM or DS3 or both communications services, and provides guidelines for using these services to configure multiple-site OpenVMS Cluster systems.

The ATM and DS3 services provide long-distance, point-to-point communications that you can configure into your OpenVMS Cluster system to gain WAN connectivity. The ATM and DS3 services are available from most common telephone service carriers and other sources.

Note

DS3 is not available in Europe and some other locations. Also, ATM is a new and evolving standard, and ATM services might not be available in all localities.

ATM and DS3 services are approved for use with the following OpenVMS versions:

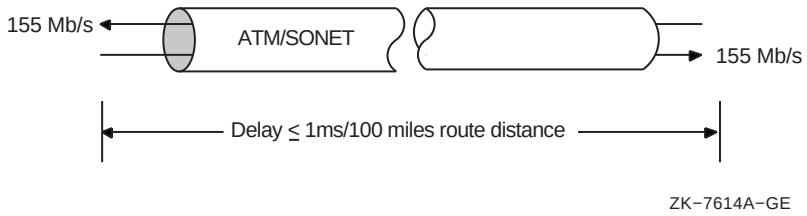
Service	Approved Versions of OpenVMS
ATM	OpenVMS Version 6.2 or later
DS3	OpenVMS Version 6.1 or later

The following sections describe the ATM and DS3 communication services and how to configure these services into multiple-site OpenVMS Cluster systems.

D.3.1 The ATM Communications Service

The ATM communications service that uses the SONET physical layer (ATM /SONET) provides full-duplex communications (that is, the bit rate is available simultaneously in both directions as shown in Figure D–3). ATM/SONET is compatible with multiple standard bit rates. The SONET OC-3 service at 155 Mb/s full-duplex rate is the best match to FDDI's 100 Mb/s bit rate. ATM/SONET OC-3 is a standard service available in most parts of the world. In Europe, ATM/SONET is a high performance alternative to the older E3 standard.

Figure D–3 ATM/SONET OC-3 Service

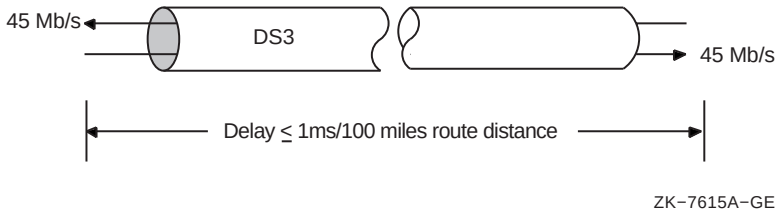


To transmit data, ATM frames (packets) are broken into **cells** for transmission by the ATM service. Each cell has 53 bytes, of which 5 bytes are reserved for header information and 48 bytes are available for data. At the destination of the transmission, the cells are reassembled into ATM frames. The use of cells permits ATM suppliers to multiplex and demultiplex multiple data streams efficiently at differing bit rates. This conversion of frames into cells and back is transparent to higher layers.

D.3.2 The DS3 Communications Service (aka T3 Communications Service)

The DS3 communications service provides full-duplex communications as shown in Figure D–4. DS3 (also known as T3) provides the T3 standard bit rate of 45 Mb/s. T3 is the standard service available in North America and many other parts of the world.

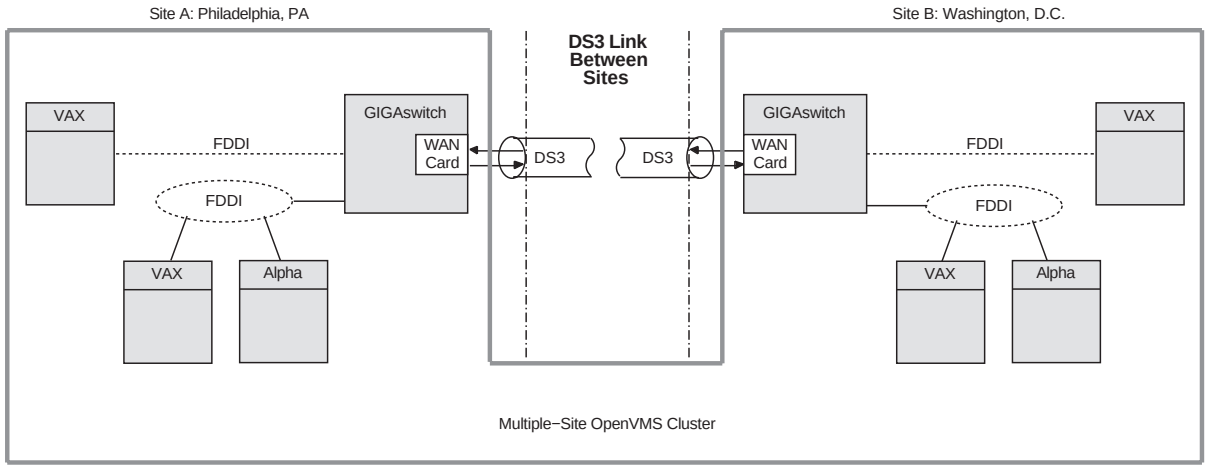
Figure D–4 DS3 Service



D.3.3 FDDI-to-WAN Bridges

You can use FDDI-to-WAN (for example, FDDI-to-ATM or FDDI-to-DS3 or both) bridges to configure an OpenVMS Cluster with nodes in geographically separate sites, such as the one shown in Figure D–5. In this figure, the OpenVMS Cluster nodes at each site communicate as though the two sites are connected by FDDI. The FDDI-to-WAN bridges make the existence of ATM and DS3 transparent to the OpenVMS Cluster software.

Figure D–5 Multiple-Site OpenVMS Cluster Configuration Connected by DS3



ZK−7234A−GE

In Figure D–5, the FDDI-to-DS3 bridges and DS3 operate as follows:

1. The local FDDI-to-DS3 bridge receives FDDI packets addressed to nodes at the other site.
2. The bridge converts the FDDI packets into DS3 packets and sends the packets to the other site via the DS3 link.
3. The receiving FDDI-to-DS3 bridge converts the DS3 packets into FDDI packets and transmits them on an FDDI ring at that site.

Compaq recommends using the GIGAswitch/FDDI system to construct FDDI-to-WAN bridges. The GIGAswitch/FDDI, combined with the DEFGT WAN T3 /SONET option card, was used during qualification testing of the ATM and DS3 communications services in multiple-site OpenVMS Cluster systems.

D.3.4 Guidelines for Configuring ATM and DS3 in an OpenVMS Cluster System

When configuring a multiple-site OpenVMS Cluster, you must ensure that the intersite link's delay, bandwidth, availability, and bit error rate characteristics meet application needs. This section describes the requirements and provides recommendations for meeting those requirements.

D.3.4.1 Requirements

To be a configuration approved by Compaq, a multiple-site OpenVMS Cluster must comply with the following rules:

Maximum intersite link route distance	The total intersite link cable route distance between members of a multiple-site OpenVMS Cluster cannot exceed 150 miles (242 km). You can obtain exact distance measurements from your ATM or DS3 supplier. *This distance restriction can be exceeded when using Disaster Tolerant Cluster Services for OpenVMS, a system management and software package for configuring and managing OpenVMS disaster tolerant clusters.*

Maximum intersite link utilization	Average intersite link utilization in either direction must be less than 80% of the link's bandwidth in that direction for any 10-second interval. Exceeding this utilization is likely to result in intolerable queuing delays or packet loss.
Intersite link specifications	The intersite link must meet the OpenVMS Cluster requirements specified in Table D–3.
OpenVMS Cluster LAN configuration rules	Apply the configuration rules for OpenVMS Cluster systems on a LAN to a configuration. Documents describing configuration rules are referenced in Section D.1.3.

D.3.4.2 Recommendations

When configuring the DS3 interconnect, apply the configuration guidelines for OpenVMS Cluster systems interconnected by LAN that are stated in the OpenVMS Cluster Software SPD (SPD 29.78.*nn*) and in this manual. OpenVMS Cluster members at each site can include any mix of satellites, systems, and other interconnects, such as CI and DSSI.

This section provides additional recommendations for configuring a multiple-site OpenVMS Cluster system.

DS3 link capacity/protocols

The GIGAswitch with the WAN T3/SONET option card provides a full-duplex, 155 Mb/s ATM/SONET link. The entire bandwidth of the link is dedicated to the WAN option card. However, the GIGAswitch/FDDI's internal design is based on full-duplex extensions to FDDI. Thus, the GIGAswitch/FDDI's design limits the ATM/SONET link's capacity to 100 Mb/s in each direction.

The GIGAswitch with the WAN T3/SONET option card provides several protocol options that can be used over a DS3 link. Use the DS3 link in clear channel mode, which dedicates its entire bandwidth to the WAN option card. The DS3 link capacity varies with the protocol option selected. Protocol options are described in Table D–1.

Table D–1 DS3 Protocol Options

Protocol Option	Link Capacity
ATM[1] AAL–5[2] mode with PLCP[3] disabled.	39 Mb/s
ATM AAL–5 mode with PLCP enabled.	33 Mb/s
HDLC[4] mode (not currently available).	43 Mb/s

[1]Asynchronous transfer mode
[2]ATM Adaptation Layer
[3]Physical Layer Convergence Protocol
[4]High-Speed Datalink Control

For maximum link capacity, Compaq recommends configuring the WAN T3/SONET option card to use ATM AAL–5 mode with PLCP disabled.

Intersite bandwidth

The intersite bandwidth can limit application locking and I/O performance (including volume shadowing or RAID set copy times) and the performance of the lock manager.

To promote reasonable response time, Compaq recommends that average traffic in either direction over an intersite link not exceed 60% of the link's bandwidth in that direction for any 10-second interval. Otherwise, queuing delays within the FDDI-to-WAN bridges can adversely affect application performance.

Remember to account for both OpenVMS Cluster communications (such as locking and I/O) and network communications (such as TCP/IP, LAT, and DECnet) when calculating link utilization.

Intersite delay

An intersite link introduces a one-way delay of up to 1 ms per 100 miles of intersite cable route distance plus the delays through the FDDI-to-WAN bridges at each end. Compaq recommends that you consider the effects of intersite delays on application response time and throughput.

For example, intersite link one-way path delays have the following components:

- Cable route one-way delays of 1 ms/100 miles (0.01 ms/mile) for both ATM and DS3.
- FDDI-to-WAN bridge delays (approximately 0.5 ms per bridge, and two bridges per one-way trip)

Calculate the delays for a round trip as follows:

WAN round-trip delay = 2 x (*N* miles x 0.01 ms per mile + 2 x 0.5 ms per FDDI-WAN bridge)

An I/O write operation that is MSCP served requires a minimum of two round-trip packet exchanges:

WAN I/O write delay = 2 x WAN round-trip delay

Thus, an I/O write over a 100-mile WAN link takes at least 8 ms longer than the same I/O write over a short, local FDDI.

Similarly, a lock operation typically requires a round-trip exchange of packets:

WAN lock operation delay = WAN round-trip delay

An I/O operation with *N* locks to synchronize it incurs the following delay due to WAN:

WAN locked I/O operation delay = (*N* x WAN lock operation delay) + WAN I/O delay

Bit error ratio

The bit error ratio (BER) parameter is an important measure of the frequency that bit errors are likely to occur on the intersite link. You should consider the effects of bit errors on application throughput and responsiveness when configuring a multiple-site OpenVMS Cluster. Intersite link bit errors can result in packets being lost and retransmitted with consequent delays in application I/O response time (see Section D.3.6). You can expect application delays ranging from a few hundred milliseconds to a few seconds each time a bit error causes a packet to be lost.

Intersite link availability

Interruptions of intersite link service can result in the resources at one or more sites becoming unavailable until connectivity is restored (see Section D.3.5).

System disks

Sites with nodes contributing quorum votes should have a local system disk or disks for those nodes.

System management

A large, multiple-site OpenVMS Cluster requires a system management staff trained to support an environment that consists of a large number of diverse systems that are used by many people performing varied tasks.

Microwave DS3 links

You can provide portions of a DS3 link with microwave radio equipment. The specifications in Section D.3.6 apply to any DS3 link. The BER and availability of microwave radio portions of a DS3 link are affected by local weather and the length of the microwave portion of the link. Consider working with a microwave consultant who is familiar with your local environment if you plan to use microwaves as portions of a DS3 link.

D.3.5 Availability Considerations

If the FDDI-to-WAN bridges and the link that connects multiple sites become temporarily unavailable, the following events could occur:

- Intersite link failures can result in the resources at one or more sites becoming unavailable until intersite connectivity is restored.
- Intersite link bit errors (and ATM cell losses) and unavailability can affect:
  - System responsiveness
  - System throughput (or bandwidth)
  - Virtual circuit (VC) closure rate
  - OpenVMS Cluster transition and site failover time

Many communication service carriers offer availability-enhancing options, such as path diversity, protective switching, and other options that can significantly increase the intersite link's availability.

D.3.6 Specifications

This section describes the requirements for successful communications and performance with the WAN communications services.

To assist you in communicating your requirements to a WAN service supplier, this section uses WAN specification terminology and definitions commonly used by telecommunications service providers. These requirements and goals are derived from a combination of Bellcore Communications Research specifications and a Digital analysis of error effects on OpenVMS Clusters.

Table D–2 describes terminology that will help you understand the Bellcore and OpenVMS Cluster requirements and goals used in Table D–3.

Use the Bellcore and OpenVMS Cluster requirements for ATM/SONET - OC3 and DS3 service error performance (quality) specified in Table D–3 to help you assess the impact of the service supplier's service quality, availability, down time, and service-interruption frequency goals on the system.

Note

To ensure that the OpenVMS Cluster system meets your application response-time requirements, you might need to establish WAN requirements that exceed the Bellcore and OpenVMS Cluster requirements and goals stated in Table D–3.

Table D–2 Bellcore and OpenVMS Cluster Requirements and Goals Terminology

Specification	Requirements	Goals
Bellcore Communications Research	Bellcore specifications are the recommended "generic error performance requirements and objectives" documented in the Bellcore Technical Reference TR–TSY–000499 *TSGR: Common Requirements*. These specifications are adopted by WAN suppliers as their service guarantees. The FCC has also adopted them for tariffed services between common carriers. However, some suppliers will contract to provide higher service-quality guarantees at customer request. Other countries have equivalents to the Bellcore specifications and parameters.	These are the recommended minimum values. Bellcore calls these goals their "objectives" in the *TSGR: Common Requirements* document.
OpenVMS Cluster	In order for Compaq to approve a configuration, parameters must meet or exceed the values shown in the OpenVMS Cluster Requirements column in Table D–3. If these values are not met, OpenVMS Cluster performance will probably be unsatisfactory because of interconnect errors/error recovery delays, and VC closures that may produce OpenVMS Cluster state transitions or site failover or both. If these values are met or exceeded, then interconnect bit error–related recovery delays will not significantly degrade average OpenVMS Cluster throughput. OpenVMS Cluster response time should be generally satisfactory. Note that if the requirements are only being met, there may be several application pauses **per hour**.[1]	For optimal OpenVMS Cluster operation, all parameters should meet or exceed the OpenVMS Cluster Goal values. Note that if these values are met or exceeded, then interconnect bit errors and bit error recovery delays should not significantly degrade average OpenVMS Cluster throughput. OpenVMS Cluster response time should be generally satisfactory, although there may be brief application pauses a few times **per day**.[2]

[1] Application pauses may occur every hour or so (similar to what is described under OpenVMS Cluster Requirements) because of packet loss caused by bit error.

[2] Pauses are due to a virtual circuit retransmit timeout resulting from a lost packet on one or more NISCA transport virtual circuits. Each pause might last from a few hundred milliseconds to a few seconds.

Table D–3 OpenVMS Cluster DS3 and SONET OC3 Error Performance Requirements

Parameter	Bellcore Requirement	Bellcore Goal	OpenVMS Cluster Requirement[1]	OpenVMS Cluster Goal[1]	Units
Errored seconds (% ES)	<1.0%	<0.4%	<1.0%	<0.028%	% ES/24 hr
	The ES parameter can also be expressed as a count of errored seconds, as follows:				
	<864	<345	<864	<24	ES per 24-hr period
Burst errored seconds (BES)[2]	$\leq$4	–	$\leq$4	Bellcore Goal	BES/day
Bit error ratio (BER)[3]	1×10^{-9}	2×10^{-10}	1×10^{-9}	6×10^{-12}	Errored bits/bit
DS3 channel unavailability	None	$\leq$97 @ 250 miles, linearly decreasing to 24 @ $\leq$25 miles	None	Bellcore Goal	Min/yr
SONET channel unavailability	None	$\leq$105 @ 250 miles, linearly decreasing to 21 @ $\leq$50 miles	None	Bellcore Goal	Min/yr
Channel-unavailable event[4]	None	None	None	1 to 2	Events/year

[1]Application requirements might need to be more rigorous than those shown in the OpenVMS Cluster Requirements column.

[2]Averaged over many days.

[3]Does not include any burst errored seconds occurring in the measurement period.

[4]The average number of channel down-time periods occurring during a year. This parameter is useful for specifying how often a channel might become unavailable.

Table Key

- Availability—The long-term fraction or percentage of time that a transmission channel performs as intended. Availability is frequently expressed in terms of unavailability or down time.
- BER (bit error ratio)—"The BER is the ratio of the number of bits in error to the total number of bits transmitted during a measurement period, excluding all burst errored seconds (defined below) in the measurement period. During a burst errored second, neither the number of bit errors nor the number of bits is counted."
- BES (burst errored second)—"A burst errored second is any errored second containing at least 100 errors."
- Channel—The term for a link that is used in the Bellcore *TSGR: Common Requirements* document for a SONET or DS3 link.
- Down time—The long-term average amount of time (for example, minutes) that a transmission channel is not available during a specified period of time (for example, 1 year).

  "...unavailability or downtime of a channel begins when the first of 10 [or more] consecutive Severely Errored Seconds (SESs) occurs, and ends when the first of 10 consecutive non-SESs occurs."

  The unavailable time is counted from the first SES in the 10–SES sequence.

  "The time for the end of unavailable time is counted from the first fault-free second in the [non-SES] sequence."
- ES (errored second)—"An errored second is any one-second interval containing at least one error."
- SES (severely errored second)—"...an SES is a second in which the BER is greater than 10^{-3}."

D.4 Managing OpenVMS Cluster Systems Across Multiple Sites

In general, you manage a multiple-site OpenVMS Cluster using the same tools and techniques that you would use for any OpenVMS Cluster interconnected by a LAN. The following sections describe some additional considerations and recommends some system management tools and techniques.

The following table lists system management considerations specific to multiple-site OpenVMS Cluster systems.

Problem	Possible Solution
Multiple-site configurations present an increased probability of the following failure modes: • OpenVMS Cluster quorum loss resulting from site-to-site communication link failure. • Site loss resulting from power failure or other breakdown can affect all systems at that site.	Assign votes so that one preferred site has sufficient votes to maintain quorum and to continue operation if the site-to-site communication link fails or if the other site is unavailable. Select the site with the most critical applications as the primary site. Sites with a few noncritical systems or satellites probably should not have sufficient votes to continue.
Users expect that the local resources will either continue to be available or will rapidly become available after such a failure. This might not always be the case.	Consider the following options for setting user expectations: • Set management and user expectations regarding the likely effects of failures, and consider training remote users in the procedures to be followed at a remote site when the system becomes unresponsive because of quorum loss or other problems. • Develop management policies and procedures for what actions will be taken to identify and handle these failure modes. These procedures may include manually adjusting quorum to allow a site to continue.

D.4.1 Methods and Tools

You can use the following system management methods and tools to manage both remote and local nodes:

- There are two options for remote-site console access when you use an intersite link through a DECserver in reverse LAT mode.
  - + Use the following tools to connect remote consoles:
    - – SET HOST/LAT command
    - – POLYCENTER Console Manager
    - – OpenVMS Cluster Console System (VCS)
    - – Disaster Tolerant Cluster Services for OpenVMS, a Compaq system management and software package
  - + Use a modem to dial up the remote system consoles.
- An alternative to remote-site console access is to have a system manager at each site.
- To enable device and processor control commands to take effect across all nodes in an OpenVMS Cluster system, use the System Management utility (SYSMAN) that is supplied with the OpenVMS operating system.

D.4.2 Shadowing Data

Volume Shadowing for OpenVMS allows you to shadow data volumes across multiple sites. System disks can be members of a volume shadowing or RAID set within a site; however, use caution when configuring system disk shadow set members in multiple sites. This is because it may be necessary to boot off a remote system disk shadow set member after a failure. If your system does not support FDDI booting, it will not be possible to do this.

See the Software Product Descriptions (SPDs) for complete and up-to-date details about Volume Shadowing for OpenVMS (SPD 27.29.*xx*) and StorageWorks RAID for OpenVMS (SPD 46.49.*xx*).

D.4.3 Monitoring Performance

Monitor performance for multiple-site OpenVMS Cluster systems as follows:

- Monitor the virtual circuit (VC) packet-loss count and round-trip time values using the System Dump Analyzer (SDA). The procedures for doing this are documented in *OpenVMS Cluster Systems*.
- Monitor the intersite link bit error ratio (BER) and packet loss using network management tools. You can use tools such as POLYCENTER NetView or DECmcc to access the GIGAswitch and WAN T3/SONET option card's management information and to set alarm thresholds. See the GIGAswitch, WAN T3/SONET card, POLYCENTER, and DECmcc documentation, as appropriate.

Index

A

B

C

D

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W