VMS Software

VSI OpenVMS

Guidelines for OpenVMS Cluster Configurations

Document Number: DO-DGDCLU-01A

Publication Date: May 2024

Operating System and Version: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher

VMS Software, Inc. (VSI)
Boston, Massachusetts, USA

Guidelines for OpenVMS Cluster Configurations

Legal Notice

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.

Note

This manual is applicable only for a combination of Integrity server systems and Alpha or systems. For Alpha and VAX or Alpha systems combination, see the previous version of the manual.

---

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

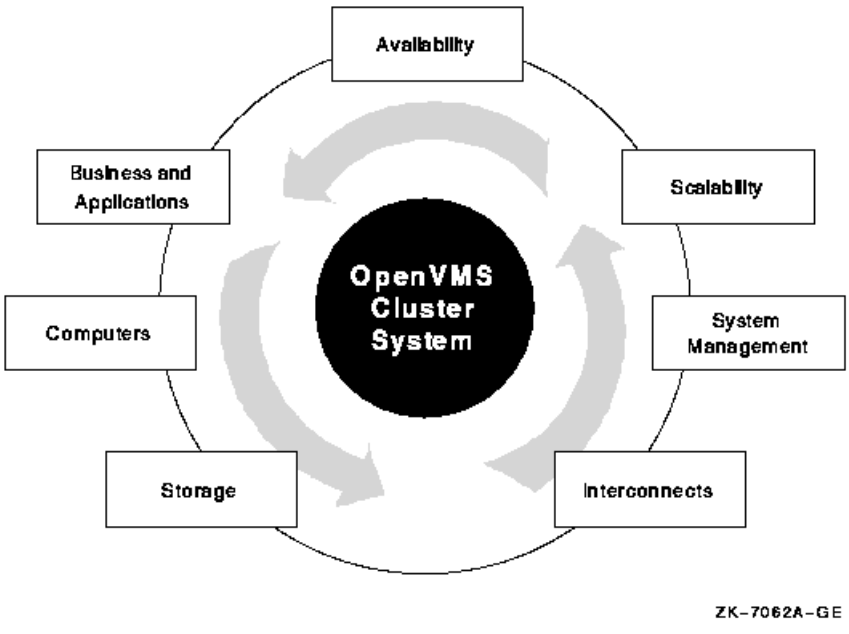
2. Intended Audience

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.

3. 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. *Guidelines for OpenVMS Cluster Configurations* explains these cluster concepts in the following chapters and appendixes.

Chapter 1 provides information on OpenVMS Cluster hardware, software, and general concepts.

Chapter 2 explains how to determine your OpenVMS Cluster business and application requirements.

Chapter 3 contains information to help you select systems for your OpenVMS Cluster to satisfy your business and application requirements.

Chapter 4 describes cluster interconnects that OpenVMS Cluster systems support.

Chapter 5 describes how to design a storage subsystem.

Chapter 6 provides guidelines for configuring multiple paths to storage using Parallel SCSI or Fibre Channel interconnects, thereby increasing availability.

Chapter 7 provides guidelines for configuring an OpenVMS Cluster with Fibre Channel as a storage interconnect.

Chapter 8 provides information on how to increase the availability of a cluster system.

Chapter 9 explains how to expand an OpenVMS Cluster system in all of its dimensions while understanding the tradeoffs.

Chapter 10 explains how to deal effectively with some of the issues involved in managing an OpenVMS Cluster system.

Appendix A provides guidelines for configuring multiple hosts and storage on a single SCSI bus so that multiple hosts can share access to SCSI devices directly.

Appendix B explains why, when, and how to use the MEMORY CHANNEL interconnect.

Appendix C discusses benefits, the configuration options and requirements, and the management of multiple-site OpenVMS Cluster systems.

4. Related Documents

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

- VSI OpenVMS Cluster Software *Software Product Description* (DO-VIBHAA-032)
- *VSI OpenVMS Cluster Systems Manual*
- *VSI OpenVMS Volume Shadowing Guide*
- *VSI OpenVMS Alpha Partitioning and Galaxy Guide*
- *Availability Manager User's Guide* [https://docs.vmssoftware.com/vsi-availability-manager-user-s-guide/]
- *VSI OpenVMS DECnet Networking Manual*
- *VSI OpenVMS Guide to OpenVMS File Applications*
- *VSI OpenVMS Guide to System Security*

- *VSI OpenVMS System Manager's Manual*

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have VSI OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

6. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at https://docs.vmssoftware.com.

7. Typographical Conventions

The following conventions are used in this manual:

Convention	Meaning
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*	A sequence such as **PF1** *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
...	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 you must enclose choices in parentheses if you specify more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
|	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold text	This typeface represents the name of an argument, an attribute, or a reason.

Convention	Meaning
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.
`Example`	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
-	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.

Chapter 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. OpenVMS Cluster Configurations

An OpenVMS Cluster systems is a group of OpenVMS systems, storage subsystems, interconnects, and software that work together as one virtual system. An OpenVMS Cluster system can be homogeneous, that is, all systems are the same architecture all running OpenVMS. An OpenVMS Cluster can be heterogeneous, that is, a combination of two architectures with all systems running OpenVMS. The two valid combinations are Alpha and VAX or Alpha and Integrity server systems.

In an OpenVMS Cluster system, each system:

- 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.

Note

In a heterogeneous cluster, only one architecture is supported per system disk and per system boot block.

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.
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 multiprocessor 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

Benefit	Description
	only once. OpenVMS Clusters automatically balance user, batch, and print work loads.
Open systems	Adherence to IEEE, POSIX, OSF/1, 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. See the VSI OpenVMS Cluster Software *Software Product Description* for the complete list of supported components.

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 OpenVMS systems.	OpenVMS Cluster systems can contain any supported Integrity sever, Alpha, VAX, or x86-64 system.
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: • Small Computer Systems Interface (SCSI) (node-to-storage only). • Serial Attached SCSI (SAS) (node-to-storage only). • Fibre Channel (node-to-storage only). • Ethernet. See the Software Product Description for the complete list of supported adapters.
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 or SAS disks and tapes • Storage arrays • SCSI or SAS storage controllers • InfoServer systems

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 adapters used on Peripheral Component Interconnect (PCI) and PCI-Express (PCIe) systems include the following:
		
		• KZPSA (SCSI)
		• The following Ethernet adapters:
		  • DE435
		  • DEGPA
		  • DEGXA
		  • AB287A
		  • AD385A
		  • AD339A (PCIe)
		  • NC360M (PCIe)

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 with an OpenVMS Cluster license.

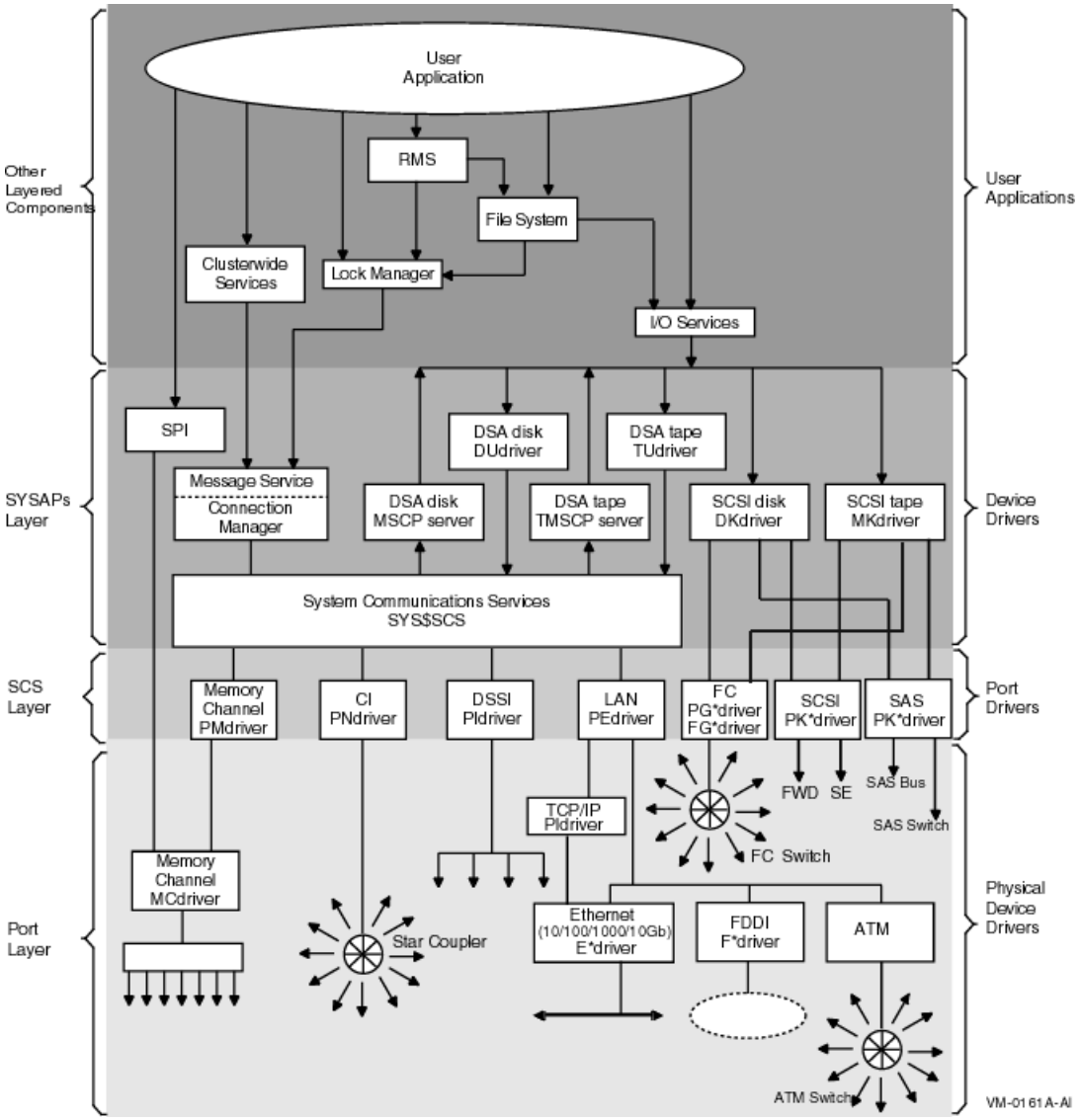
Table 1.3. Operating System Components

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.

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 the OpenVMS 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.

Figure 1.1. Hardware and Operating System Components



Note

Not all interconnects are supported on all three architectures of OpenVMS. The CI, DSSI, and FDDI interconnects are supported on Alpha and VAX systems. Memory Channel and ATM interconnects are supported only on Alpha systems.

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/DECnet	A network transport is necessary for internode communication.
Distributed File Service (DFS)	Software to let you communicate and share resources among systems over extended distances.
Advanced Server for OpenVMS (Advanced Server) and Common Internet File System (CIFS) for OpenVMS	Client and server networking software that links PC systems into OpenVMS Cluster systems. Advanced Server can be used on only Alpha systems and CIFS can be used for both Integrity servers and Alpha systems.
VSI TCP/IP for OpenVMS	Provides Network File System (NFS) server capabilities for OpenVMS and supports Internet networking protocols. Cluster over IP requires VSI TCP/IP stack for cluster communication.

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)
- RAID Software for OpenVMS (supports RAID level 0 arrays (disk striping) and RAID level 5 arrays (disk striping with parity)
- Hierarchical Storage Manager (HSM)

1.3.4. System Management Software

System management software helps you manage your OpenVMS Cluster system.

Examples include:

- VSI Archive/Backup System for OpenVMS
- *Code Management System for OpenVMS*
- VSI WBEM Services for OpenVMS Integrity servers

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:

- An OpenVMS Cluster system consisting of Integrity server systems and Alpha cannot contain more than 96 (combined total) systems.
- An Alpha and an Integrity server system cannot boot from the same system disk. System disks are architecture specific and can be shared only by systems of the same architecture.
- Each 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 or the Fibre Channel.

  Reference: See Section 4.8 for more information about the SCSI interconnect and Section 4.6 for more information about the Fibre Channel interconnect.

  Configurations that use SAS interconnect must also be configured with any of the other supported OpenVMS Cluster interconnects.

  Reference: See Section 4.9 for more information about the SAS interconnect.
- An OpenVMS Cluster node 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.
- Cluster over IP requires TCP/IP Services. If any node requires IP for cluster communication, all the other members must be enabled for IP.

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.

Chapter 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.

Availability Requirements	Description
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.

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.

HPE offers a full range of OpenVMS Alpha and Integrity systems, with each supporting 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 9 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 9.3.8 for information about multiple and extended local area network (LAN) configurations.

For information about clusters formed using IP at different geographical locations, see Section C.2.

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 *VSI 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: See the OpenVMS Cluster Software *Software Product Description* for the memory requirements.

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 time sharing	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. System Management 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 that are supplied with the OpenVMS operating system.

Table 2.4. System Management Tools

Tool	Function
Accounting utility	Tracks how resources are being used.
AUTOGEN command procedure	Optimizes system parameter settings based on usage.
Availability Manager	VSI Availability Manager is a system management tool that enables one or more OpenVMS Integrity server or Alpha nodes to be monitored extended LAN or wide area network (WAN). That is, the nodes for which you are collecting the information must be in the same extended LAN and there should be an interface that communicates with the collector nodes as well as the WAN analyzer. The Availability Manager collects system and process data from multiple OpenVMS nodes simultaneously, and then analyzes the data and displays the output using a native Java GUI.
Insight Management Agents for OpenVMS	Enables you to look at devices on your OpenVMS systems. With the installation of Insight Manager on a Microsoft Windows server, you can manage all your platforms from a single Windows server.
VSI WBEM Services for OpenVMS	WBEM (Web-Based Enterprise Management) enables management applications to retrieve system information and request system operations wherever and whenever required. It allows customers to manage their systems consistently across multiple platforms and operating systems, providing integrated solutions that optimize your infrastructure for greater operational efficiency.
Monitor utility	Provides basic performance data.
OpenVMS Data Collector and Performance Analyzer (ECP Data Collector and ECP Performance Analyzer)	The rights to use these ECP products are provided at no additional costs. The ECP Data Collector provides performance-data collection, archiving, reporting, and file display. The ECP Performance Analyzer assists in performance analysis and capacity planning of OpenVMS Cluster systems by identifying bottlenecks and recommending ways to fix problems. An API is provided

Tool	Function
	within the Data Collector to gain direct access to the data it collects.
Performance Data Collector for OpenVMS (TDC V2)	Used to gather performance data for an AlphaServer system running OpenVMS Version 7.3-2, or later. By default, TDC periodically collects and stores data in a file. Subsequently, user applications can retrieve data from the file.
Show Cluster utility	Monitors activity and performance in a OpenVMS Cluster configuration.
Systems Communications Architecture Control Program(SCACP)	Designed to monitor and manage LAN cluster communications.

For more information about the Accounting utility, the AUTOGEN command procedure, the Monitor utility, the Show Cluster utility, and the System Communications Architecture Control Program (SCACP), refer to the *VSI OpenVMS System Management Utilities Reference Manual*.

Chapter 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. Integrity servers and Alpha Systems

An OpenVMS cluster can include systems running OpenVMS Integrity servers or a combination of systems running OpenVMS Integrity servers and OpenVMS Alpha. See the OpenVMS *Software Product Description* for a listing of the models currently supported.

- OpenVMS Integrity servers operating system

  Based on the Intel® Itanium® architecture, OpenVMS Integrity servers provide the price or performance, reliability, and scalability benefits of OpenVMS on HPE Integrity systems.

- OpenVMS Alpha operating system

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

3.2. Types of Systems

HPE Integrity systems span a range of computing environments, including:

- Entry
- Standalone servers
- Scalable blades
- Integrity VM guest 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. See the *Software Product Description*or visit for the complete list of supported Integrity server systems.

3.4. 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 for more information about these failover mechanisms and about availability.

Chapter 4. Choosing OpenVMS Cluster Interconnects

An interconnect is a physical path that connects computers to other computers, and to storage subsystems. OpenVMS Cluster systems support a variety of interconnects (also referred to as buses) so that members can communicate with each other and with storage, using the most appropriate and effective method available.

The software that enables OpenVMS Cluster systems to communicate over an interconnect is the System Communications Services (SCS). An interconnect that supports node-to-node SCS communications is called a **cluster interconnect**. An interconnect that provides node-to-storage connectivity within a cluster is called a **shared-storage interconnect**.

OpenVMS supports the following types of interconnects:

- Ethernet for cluster interconnects (node-to-node only)
- Shared-storage interconnects (node-to-storage only)
  - Fibre Channel
  - Small Computer Systems Interface (SCSI) (Integrity servers or Alpha, Integrity servers limited to specific configurations)
  - Serial Attached SCSI (SAS) (Integrity servers only)
- Ethernet for both node-to-node and node-to-storage interconnects

Note

Cluster over IP is supported on Ethernet.

Note

The CI, DSSI, and FDDI interconnects are supported on Alpha and VAX systems. Memory Channel and ATM interconnects are supported only on Alpha systems. For documentation related to these interconnects, see the previous version of the manual.

4.1. Characteristics

The 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.

Characteristic	Description
	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 4.2shows key statistics for a variety of interconnects.

Table 4.2. Comparison of Cluster Interconnect Types

Interconnect	Maximum Throughput (Mb/s)	Hardware-Assisted Data Link[1]	Storage Connection	Topology	Maximum Nodes per Cluster	Maximum Length
General-purpose						
Ethernet	10/100/1000	No	MSCP served	Linear or radial to a hub or switch	96[2]	100 m[4]/ 100 m[4]/ 550 m[3]
Shared-storage only						
Fibre Channel	1000	No	Direct[5]	Radial to a switch	96[2]	10 km[6]/ 100 km[7]
SCSI	160	No	Direct[5]	Bus or radial to a hub	8-12[8]	25 m
SAS	6000	No	Direct	Point to Point, Radial to a switch	96[2]	6 m

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

[2]OpenVMS Cluster computers.

[4]Based on unshielded twisted-pair wiring (UTP). Longer distances can be achieved by bridging between this interconnect and WAN interswitch links (ISLs), using common carriers such as [D]WDM and so on.

[3]Based on multimode fiber (MMF). Longer distances can be achieved by bridging between this interconnect and WAN interswitch links using common carriers such as [D]WDM and so on.

[5]Direct-attached SCSI and Fibre Channel storage can be MSCP served over any of the general-purpose cluster interconnects.

[6]Based on single-mode fiber, point-to-point link.

[7]Support for longer distances (up to 100 km) based on inter-switch links (ISLs) using single-mode fiber. In addition, DRM configurations provide longer distance ISLs using the Open Systems Gateway and Wave Division Multiplexors.

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

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 9.3.3 for an example of dynamic MSCP load balancing.

4.4. Mixed Interconnects

You can use two or more different types of interconnects in an OpenVMS Cluster system. You can use different types of interconnects to combine the advantages of each type and to expand your OpenVMS Cluster system.

Note

If any one node in a cluster requires IP for cluster communication, all the other members in the cluster must be enabled for IP cluster communication.

4.5. Interconnect Support

For the latest information on supported interconnects, see the most recent OpenVMS Cluster Systems *Software Product Description*.

Reference: For detailed information about the interconnect sand adapters supported on each Integrity server system and AlphaServer system, visit the VSI OpenVMS web page at: .

4.6. Fibre Channel Interconnect

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 over 100 megabytes per second. Fibre Channel supports 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.

4.6.1. Advantages

The Fibre Channel interconnect offers the following advantages:

- High-speed transmission, 2 Gb/s, 4 Gb/s, 8 Gb/s (depending on adapter)
- Scalable configuration to support department to enterprise configurations.
- Long-distance interconnects

  Fibre Channel supports multi mode fiber at 500 meters per link. Fibre Channel supports longer-distance interswitch links (ISLs) — up to 100 kilometers per link, using single-mode fiber and up to 600 kilometers per link with FC/ATM links.

In addition, SANworks Data Replication Manager (DRM) configurations provide long distance ISLs through the use of the Open Systems Gateway and Wave Division Multiplexors.

- High availability

  Multipath support is available to provide configurations with no single point of failure.

4.6.2. Throughput

The Fibre Channel interconnect transmits up to 2 Gb/s, 4 Gb/s, 8 Gb/s (depending on adapter). It is a full-duplex serial interconnect that can simultaneously transmit and receive over 100 MB/s.

4.7. MEMORY CHANNEL Interconnect (Alpha Only)

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 or 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 MEMORY CHANNEL adapter, CCMAA–BA, provides improved performance over the earlier adapter.

Reference: For information about the CCMAA-BA adapter support on AlphaServer systems, go to the VSI OpenVMS web page at: .

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 computers on a single SCSI interconnect can simultaneously access SCSI disks. This type of configuration is called multihost SCSI connectivity or shared SCSI storage and is restricted to certain adapters and limited configurations. A second type of interconnect is required for node-to-node communication.

Shared SCSI storage in an OpenVMS Cluster system enables computers connected to a single SCSI bus to share access to SCSI storage devices directly. This capability makes it possible to build highly available servers using shared access to SCSI storage.

4.8.1. OpenVMS Alpha Configurations

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

- SCSI host adapter that is supported in a multihost configuration (see Table 4.5)

- 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 VSI OpenVMS Cluster *Software Product Description*)

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

---

Note

This support is restricted to certain adapters. OpenVMS does *not* provide this support for the newest SCSI adapters, including the Ultra SCSI adapters KZPEA, KZPDC, A6828A, A6829A, and A7173A.

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

4.8.2. OpenVMS Integrity servers Two-Node Shared SCSI Configuration

Shared SCSI storage support for two-node OpenVMS Integrity servers Cluster systems was introduced in OpenVMS Version 8.2-1. Prior to this release, shared SCSI storage was supported on OpenVMS Alpha systems only, using an earlier SCSI host-based adapter (HBA).

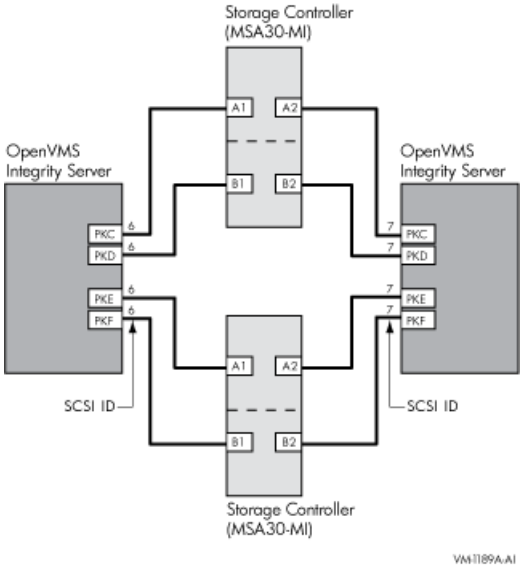
Shared SCSI storage in an OpenVMS Integrity servers Cluster system is subject to the following restrictions:

- A maximum of two OpenVMS Integrity server systems can be connected to a single SCSI bus.
- A maximum of four shared-SCSI buses can be connected to each system.
- Systems supported are the rx1600 family, the rx2600 family, and the rx4640 system.
- The A7173A HBA is the only supported HBA.
- MSA30-MI storage enclosure is the only supported SCSI storage type.
- Ultra320 SCSI disk family is the only supported disk family.

Figure 4.1 illustrates two-node shared SCSI configuration. Note that a second interconnect, a LAN, is required for host-to-host OpenVMS Cluster communications.(OpenVMS Cluster communications are also known as System Communications Architecture (SCA) communications).

Note, the SCSI IDs of 6 and 7 are required in this configuration. One of the systems must have a SCSI ID of 6 for each A7173A adapter port connected to a shared SCSI bus, instead of the factory-set default of 7. You use the U320_SCSI pscsi.efi utility, included on the IPF Offline Diagnostics and Utilities CD, to change the SCSIID. The procedure for doing this is documented in the *HP A7173APCI-X Dual Channel Ultra320 SCSI Host Bus Adapter Installation Guide*.

Figure 4.1. Two-Node OpenVMS Integrity servers Cluster System



4.8.3. 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)

4.8.4. Throughput

Table 4.3 show throughput for the SCSI interconnect.

Table 4.3. Maximum Data Transfer Rates in Megabytes per Second

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

4.8.5. 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 4.4 summarizes how the type of signaling method affects SCSI interconnect distances.

Table 4.4. 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[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 1m apart, and the interconnect cannot exceed 4 m.

[3]More than two devices can be supported.

4.8.6. Supported Adapters, Bus Types, and Computers

Table 4.5 shows SCSI adapters with the internal buses and computers they support.

Table 4.5. SCSI Adapters

Adapter	Internal Bus	Supported Computers
Embedded (NCR-810 based)/KZPAA[1]	PCI	See the options specifications for your system.
KZPSA[2]	PCI	Supported on all Alpha computers that support KZPSA in single-host configurations.[3]
KZTSA[2]	TURBO channel	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 information about the SCSI adapters supported on each OpenVMS Integrity server or Alpha system, go to the OpenVMS web page at: .

Reference: For information about the SCSI adapters supported on each AlphaServer system, go to the OpenVMS web page at: .

4.9. SAS Interconnect (Integrity servers Only)

SAS is a point-to-point architecture that transfers data to and from SCSI storage devices by using serial communication (one bit at a time). SAS uses the SAS devices and differential signaling method to achieve reliable, high-speed serial communication.

SAS combines high-end features from fiber channel (such as multi-initiator support and full duplex communication) and the physical interface leveraged from SATA (for better compatibility and investment protection), with the performance, reliability and ease of use of traditional SCSI technology.

SAS Devices

There are three types of SAS devices: initiators, targets, and expanders. An initiator device is a HBA, or controller. The initiator is attached to one or more targets - SAS hard disk drives, SATA hard disk drives, and SAS tape drives - to form a SAS domain. Expanders are low-cost, high-speed switches that scale the number of targets attached to an initiator, thereby creating a larger SAS domain. Each SAS device has a unique worldwide name (SAS address) assigned at manufacturing to simplify its identification in a domain.

Differential signaling

All SAS devices have connection points called ports. One or more transceiver mechanisms, called phys, are located in the port of each SAS device. A physical link, consisting of two wire pairs, connects the transmitter of each phy in one device's port to the receiver of a phy in another device's port. The SAS interface allows the combination of multiple physical links to create two (2x), 3x, 4x, or 8x connections per port for scalable bandwidth. A port that has one phy is described as "narrow" while a port with two to four phys is described as "wide".

SAS uses differential signaling to transfer data over a physical link, which reduces the effects of capacitance, inductance, and noise experienced by parallel SCSI at higher speeds. SAS communication is full duplex, which means that each phy can send and receive information simultaneously over the two wire pairs.

4.9.1. Advantages

A multi-node cluster using SAS provides an alternative to clustered Fibre Channel local loop topologies. This highly scalable SAS architecture enables topologies that provide high performance and high availability with no single point of failure.

SAS solutions accommodate both low cost bulk storage (SATA) or performance and reliability for mission critical applications (SAS)allowing for maximum configuration flexibility and simplicity.

4.9.2. Throughput

SAS is designed to work with speeds greater than Parallel SCSI, that is, greater than 320 Mb/s. Table 4.6 shows the throughput for the SAS interconnects. For SCSI interconnect throughput, see Table 4.3.

Table 4.6. Maximum Data Transfer Rates in Megabytes per Second

Mode	Speed in Megabytes per Second
SAS 1 (also called 3Gig SAS)	300 MB/s or 3 Gb/s
SAS 2 (also called 6Gig SAS)	600MB/s or 6 Gb/s

4.9.3. Supported Adapters, Bus Types, and Computers

Table 4.7 shows SAS adapters with the internal buses and computers they support.

Table 4.7. SAS Adapters

Adapter	Internal Bus	Supported Computers
8p SAS HBA	PCI-X	Core I/O on Integrity servers rx3600, rx6600
SC44Ge Host Bus Adapter	PCIe	Supported on Integrity servers with PCIe backplane (rx2660, rx3600, rx6600)

4.10. LAN Interconnects

Ethernet interconnectors are LAN-based.

These interconnects provide the following features:

- Single-path connections within an OpenVMS Cluster system and a LAN
- Support for LAN failover
- Support for multiple paths using multiple adapters
- Long-distance interconnect

  In addition to the maximum length specific to each LAN type, as shown in Table 4.2, longer distances can be achieved by bridging between LANs and WAN interswitch links.
- Extended physical distribution of nodes
- Support for multiple clusters (up to 96 nodes each) on a single interconnect

LANs are supported as OpenVMS Cluster interconnects on each OpenVMS platform (Integrity servers, Alpha, and x86-64).

Following the discussion of multiple LAN adapters, information specific to supported LAN interconnect, Ethernet, is provided.

4.10.1. Multiple LAN Adapters

Multiple LAN adapters are supported. The adapters can be for different LAN types or for different adapter models for the same LAN type.

Multiple LAN adapters can be used to provide the following:

- Increased node-to-node throughput by distributing the load across multiple LAN paths.
- Increased availability of node-to-node LAN communications.

4.10.1.1. Multiple LAN Path Load Distribution

When multiple node-to-node LAN paths are available, the OpenVMS Cluster software chooses the set of paths to use based on the following criteria, which are evaluated in strict precedence order:

1. Recent history of packet loss on the path

   Paths that have recently been losing packets at a high rate are termed **lossy** and will be excluded from consideration. Channels that have an acceptable loss history are termed **tight** and will be further considered for use.

2. Priority

   Management priority values can be assigned to both individual LAN paths and to local LAN devices. A LAN path's priority value is the sum of these priorities. Only tight LAN paths with a priority value equal to, or one less than, the highest priority value of any tight path will be further considered for use.

3. Maximum packet size

   Tight, equivalent-priority channels whose maximum packet size is equivalent to that of the largest packet size of any tight equivalent-priority channel will be further considered for use.

4. Equivalent latency

   LAN paths that meet the preceding criteria will be used if their latencies(computed network delay) are closely matched to that of the fastest such channel. The delay of each LAN path is measured using cluster communications traffic on that path. If a LAN path is excluded from cluster communications use because it does not meet the preceding criteria, its delay will be measured at intervals of a few seconds to determine if its delay, or packet loss rate, has improved enough so that it then meets the preceding criteria.

Packet transmissions are distributed in round-robin fashion across all communication paths between local and remote adapters that meet the preceding criteria.

4.10.1.2. Increased LAN Path Availability

Because LANs are ideal for spanning great distances, you may want to supplement an intersite link's throughput with high availability. You can do this by configuring critical nodes with multiple LAN adapters, each connected to a different intersite LAN link.

A common cause of intersite link failure is mechanical destruction of the intersite link. This can be avoided by path diversity, that is, physically separating the paths of the multiple intersite links. Path diversity helps to ensure that the configuration is unlikely to be affected by disasters affecting an intersite link.

4.10.2. Configuration Guidelines for LAN-Based Clusters

The following guidelines apply to all LAN-based OpenVMS Cluster systems:

- OpenVMS Integrity servers and OpenVMS Alpha or x86-64 systems can be configured with any mix of LAN adapters supported on those architectures. See the latest Software Product Description for the complete list of supported adapters.
- All LAN paths used for OpenVMS Cluster communication must operate with a minimum of 10 Mb/s throughput and low latency. You must use translating bridges or switches when connecting nodes on one type of LAN to nodes on another LAN type. 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.

4.10.3. Ethernet Advantages

The Ethernet interconnect is typically the lowest cost of all OpenVMS Cluster interconnects.

In addition to the advantages listed in Section 4.10, the Ethernet interconnects offer the following advantages:

- Very high throughput
- Support of jumbo frames (7552 bytes per frame) for cluster communications

4.10.4. Ethernet Ethernet Throughput

Ethernet adapters do not provide hardware assistance, so processor overhead is higher than for CI or DSSI.

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. Multiple Ethernet adapters can be used to improve cluster performance by offloading general network traffic.

Reference: For LAN configuration guidelines, see Section 4.10.2.

4.10.5. Configuration Guidelines for 10 Gigabit Ethernet Clusters

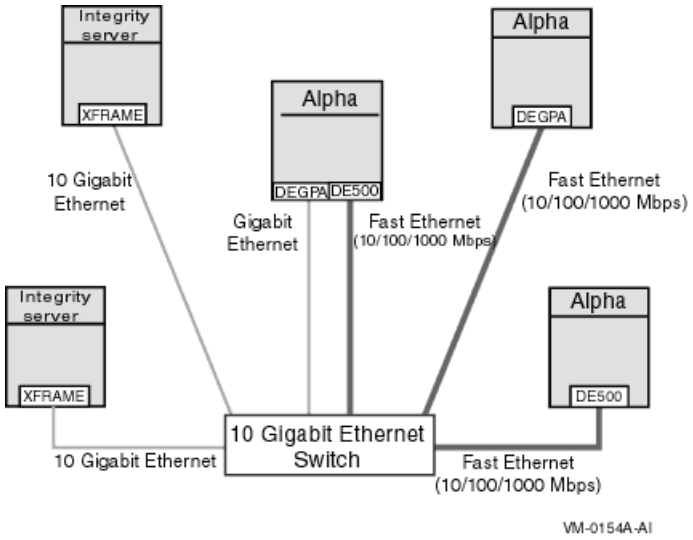
Use the following guidelines when configuring systems in a 10 Gigabit Ethernet cluster:

- Two-node 10 Gigabit Ethernet clusters do not require a switch. They can be connected point to point, as shown in Figure 4.2.

Figure 4.2. Point-to-Point 10 Gigabit Ethernet OpenVMS Cluster

- Most 10 Gigabit Ethernet switches can be configured with Gigabit Ethernet or a combination of Gigabit Ethernet and Fast Ethernet (100 Mb/s).
- Each node can have a single connection to the switch or can be configured with multiple paths, thereby increasing availability, as shown in Figure 4.3.

Figure 4.3. Switched 10 Gigabit Ethernet OpenVMS Cluster



- Support for jumbo frames (7552 bytes each) is available starting with OpenVMS Version 7.3. (Prior to the introduction of jumbo-frame support, the only frame size supported for cluster communications was the standard 1518-byte maximum Ethernet frame size.)
- The DEGPA cannot be used as the boot device, but satellites can be booted over standard 10/100 Ethernet network adapters configured on a Gigabit switch. The DEGXA Gigabit Ethernet adapter, is a Broadcom BCM5703 chip (TIGON3) network interface card (NIC). The introduction of the DEGXA Gigabit Ethernet adapter continues the existing Gigabit Ethernet support as both a LAN device as well as a cluster interconnect device. The DEGXA can also be used as a boot device.

4.11. Cluster over IP

OpenVMS cluster can also use Internet Protocol for cluster communication. The basic cluster rule is that all nodes in a cluster must be able to communicate with all other nodes in a cluster by means of direct communication. The nodes in a cluster cab situated in a same LAN in a data centre or the nodes can be distributed geographically apart.

If the nodes are within the same LAN, LAN is preferred for cluster communication. When nodes are located in multiple sites or multiple LANs, IP is preferred for cluster communication.

In a scenario, where Layer 2 service is not available or if the service is expensive, cluster communication between two different sites can use cluster over IP.

Note

It is also possible to create an extended LAN or VLAN between two sites and use LAN for cluster communication between the nodes in two different sites.

Cluster protocol (SCS aka SCA, System Communication Architecture) over LAN is provided by Port Emulator driver (PEDRIVER), the PEDRIVER also provides SCS communication using TCP/IP in

addition to LAN for cluster communication as shown in Figure 1.1. PEDRIVER uses UDP to transport SCS packets between two different nodes.

Cluster over IP provides the following features:

- IP multicast and IP unicast enables you to discover nodes beyond a LAN segment in an IP only network. The multicast and unicast node discovery mechanism is used to trace new nodes in a cluster. The file based unicast node discovery can be used alternately using a configuration file.

  SYS$SYSTEM:PE$IP_CONFIG.DAT includes the optional IP multicast and IP unicast addresses of the nodes of the cluster. IP multicast messages are used for discovering a node within the same IP multicast domain and remote nodes in a different IP multicast domain can use IP unicast messaging technique to join the cluster.

  SYS$SYSTEM:TCPIPCLUSTER.DAT contains the IP interface address names and IP addresses on which cluster communication is enabled. It also includes the TCP/IP route information.

- Enables you to add and remove nodes dynamically without any disruption to cluster. It ensures a problem free and transparent migration without a cluster reboot. You can perform rolling upgrades to the new version without a cluster reboot.

- You can perform cluster state transitions and support failover with minimal latency. PEDRIVER provides delay probing and delay measuring technique that helps to reduce latency in IP network by selecting the path with least latency.

- Provides interoperability with servers running prior versions of OpenVMS clusters which are LAN based.

- Helps dynamically load balance between all the available healthy interfaces.

- Enables the system managers to detect and remove faulty interface from the set of healthy interfaces transparent to higher layers. Availability Manager can be used for managing cluster communication over IP.

- Existing corporate policies restrict the use of non-IP protocols and the non-availability of LAN bridging from all switch and Telco vendors.

4.11.1. Configuration Guidelines

The following guidelines apply to all IP based OpenVMS Clusters:

- Ensure that the TCP/IP software is configured prior to configuring cluster over IP. To ensure that network and TCP/IP is configured properly, use the PING utility and ping the node from outside the subnet.

- The maximum number of nodes that you can connect using IP for cluster communication is 96 and this feature also supports mixed architecture configuration. Adheres to OpenVMS support matrix for mixed version cluster.

- Starting with OpenVMS Version 8.4, cluster over IP is supported and it requires VSI OpenVMS TCP/IP Version 5.7 for cluster communication.

- Alpha satellite node and its disk server must be in the same LAN.

- OpenVMS cluster using IP for cluster communication requires a minimum of 100 Mbps throughput. OpenVMS supports clusters with up to 500 miles (qualified for 60,000 miles)

4.11.2. IP Availability

Logical LAN failover can be used and configured with IP addresses for cluster communication. This logical LAN failover feature provides high availability in case of any link failure drops. For more information about logical LAN failover, see Section 8.7.

4.11.3. IP Advantages

Cluster over IP provides the following advantages:

- Cluster over IP can be deployed in sites with non-availability of Layer 2 intersite service between intersites from all switch and Telco vendors.
- Corporate policies do not restrict the use of IP protocol, whereas some of the non-IP protocols are restricted.
- Lower cost of high speed IP services.

4.11.4. IP Performance

A key challenge is to have comparable performance levels when using IP for cluster traffic. For long distance cluster the speed-of-light delay when dealing with geographically distant sites quickly becomes the dominant factor for latency, overshadowing any delays associated with traversing the IP stacks within the cluster member hosts. There may be a tradeoff between the latency of failover and steady-state performance. Localization of cluster traffic in the normal (non-failover) case as vital to optimizing system performance as the distance between sites is stretched to supported limits is considered.

Chapter 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:
    - SAS based storage
    - SCSI based storage
    - Fibre Channel 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:

- LSI 1068 and LSI Logic 1068e (on SAS)
- HSZ and RZ series (on SCSI)
- HSG and HSV controllers (on 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
SCSI	HSZ controllers and SCSI storage
Fibre Channel	HSG and HSV controllers and SCSI storage
SAS	LSI 1068 and LSI Logic 1068e 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.
- 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 *VSI 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.
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.

Software Component	Description
	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.

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

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]

Optimizer	Description
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 then dividing the application data into “chunks” that are spread equally across the disks in the stripe set in a round-robin fashion. By reducing access time, disk striping can 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 HSG 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, using RAID for OpenVMS, 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, and you can shadow host-based disk stripe sets.
Extended file cache (XFC)	OpenVMS Alpha supports host-based caching with extended file cache (XFC),which can replace and can coexist with virtual I/O cache (VIOC). XFC is a clusterwide, file-system data cache that offers several features not available with VIOC, including read-ahead caching and automatic resizing of the cache to improve performance. OpenVMS Integrity servers also supports XFC but does not support VIOC.
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 HSZ and HSG 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 9.5 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 *VSI OpenVMS Volume Shadowing Guide* 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 10.2.
Database redundancy	Keep redundant copies of certain files or partitions of databases that are, for example, updated overnight by batch jobs. Rather than

Availability Optimizer	Description
	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.
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.
Comprehensive backup strategies	Frequent and regular backups are the most effective way to ensure the availability of your data. **Reference:** For information about Fibre Channel tape support, see Section 7.5. For information about backup strategies and OpenVMS Backup, refer to the *VSI OpenVMS System Manager's Manual*.

5.5. SAS Based Storage

SAS is a point-to-point architecture that transfers data to and from SCSI storage devices by using serial communication (one bit at a time).

5.5.1. Storage Devices

Dual-domain SAS creates an additional domain to address the single SAS domain pathway failure. The additional domain uses an open port on an Smart Array controller that is capable of supporting dual-domain SAS. The second port on the dual-domain capable Smart Array controller generates a unique identifier and can support its own domain.

The following SAS controllers are supported:

- LSI 1068
- LSI Logic 1068e

The following SMART Arrays, that are supported have a SAS backplane but cannot be considered as SAS HBA:

- P400i
- P411
- P700m
- P800

There are no external controllers supported on SAS, you can only connect JABODs such as MSA60/70 and internal disks to SAS HBA. However, P700m can be connected to MSA2000SA (SAS version of MSA2000).

5.6. 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.

5.6.1. 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

5.7. Fibre Channel Based Storage

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

5.7.1. Storage Devices

The HSG and HSV storage controllers can connect to a single host or to a multihost Fibre Channel interconnect. For more information about Fibre Channel hardware support, see Section 7.2.

5.8. 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 anode 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.

5.8.1. Internal Buses

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

- PCI
- PCI-X
- PCI-Express
- EISA
- ISA
- XMI
- SCSI
- TURBOchannel

- Futurebus+

For more information about the buses supported, see the *VSI OpenVMS I/O User's Reference Manual*.

5.8.2. 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)

For the list of supported internal buses and local adapters, see the Software Product Description.

Chapter 6. Configuring Multiple Paths to SCSI and Fibre Channel Storage

This chapter describes multipath SCSI support, and applies to disks and tapes except where noted. The SCSI protocol is used on both the parallel SCSI interconnect and the Fibre Channel interconnect. The term **SCSI** is used to refer to either parallel SCSI, or Fibre Channel (FC) devices throughout the chapter.

Note

OpenVMS Alpha Version 7.3-1 introduced support for failover between local and MSCP served paths to SCSI disk devices. This capability is enabled by the MPDEV_REMOTE system parameter setting of 1, which is the default setting. This type of failover *does not* apply to tape devices.

This SCSI multipath feature may be incompatible with some third-party disk caching, disk shadowing, or similar products. Specifically, third-party products that rely on altering the Driver Dispatch Table (DDT) of either the OpenVMS Alpha SCSI disk class driver (SYS$DKDRIVER.EXE), the OpenVMS Alpha SCSI tape class driver (SYS$MKDRIVER.EXE), or the SCSI generic class driver(SYS$GKDRIVER) may need to be modified in order to function correctly with the SCSI multipath feature. VSI advises that you not use such software on SCSI devices that are configured for multipath failover (for example, SCSI devices that are connected to HSZ70 and HSZ80 controllers in multibus mode) until this feature is supported by the producer of the software. VSI offers driver dispatch table (DDT) routines for modifying these third-party products to make them compatible with SCSI multipath failover.

See Section 6.2 for important requirements and restrictions for using the multipath SCSI function.

Note that the Fibre Channel and parallel SCSI interconnects are shown generically 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*, HSG *x*, or HSV *x* controllers serve as a logical unit to a host, are shown in the figures as a single logical unit.

The following topics are presented in this chapter:

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

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 for disk devices can use either parallel SCSI or Fibre Channel as the storage interconnect, as illustrated by Figure 6.1.Multipath SCSI configurations for tape devices can use only Fibre Channel as the storage interconnect.

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 the following types of failover:

- Direct SCSI to direct SCSI
- Direct SCSI to MSCP served (disks only)
- MSCP served to direct SCSI (disks only)

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, the HSG80, the dual MDR, and the HSV110 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 for disk devices since Version 6.2.

OpenVMS Version 7.2 introduced support for a new failover mode in which the multiported disk 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 6.3.

Figure 6.1 is a generic illustration of a multibus failover configuration.

Note

Configure multiple direct SCSI paths to a disk device only when multipath support is enabled on all connected nodes, and the HSZ/G is in multibus failover mode.

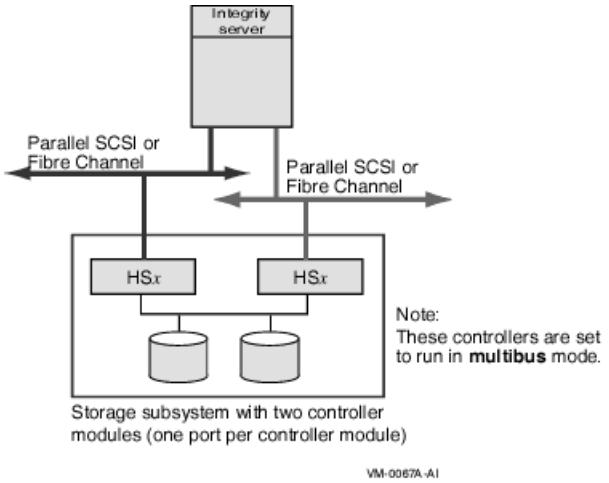
The two logical disk devices shown in Figure 6.1 represent virtual storage units that are presented to the host by the HS *x*controller modules. Each logical storage unit is "on line" to one of the two HS *x* controller modules at a time. When there are multiple logical units, they can be on line to 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, as shown in Figure 6.1, a logical unit switches 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 and issues a command to cause a switch.
- The OpenVMS system manager issues a command to cause a switch.

Figure 6.1. Multibus Failover Configuration



Note the following about Figure 6.1:

- Host has two adapters.
- Interconnects can both be parallel SCSI (HSZ70 or HSZ80) or both be Fibre Channel (HSG *x* or HSV *x*) 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.

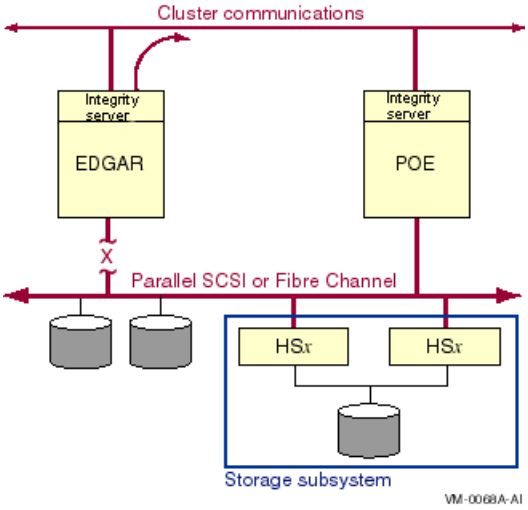
In a multibus failover configuration, SAS and its controllers (LSI 1068 and LSI 1068e) can also be used.

6.1.2. Direct SCS to MSCP Served Failover (Disks Only)

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).

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) 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.

Multipath support in such a multihost SCSI OpenVMS Cluster system also enables failover from MSCP served SCSI storage to directly attached SCSI storage. For example, the following sequence of events can occur on the configuration shown in Figure 6.2:

- Node POE is using node EDGAR as an MSCP server to access some storage device on the shared storage interconnect.
- On node EDGAR, the direct connection to the shared storage fails, or node EDGAR is shut down, or node EDGAR becomes unreachable via the cluster interconnect.
- Node POE switches to using its direct path to the shared storage.

Note

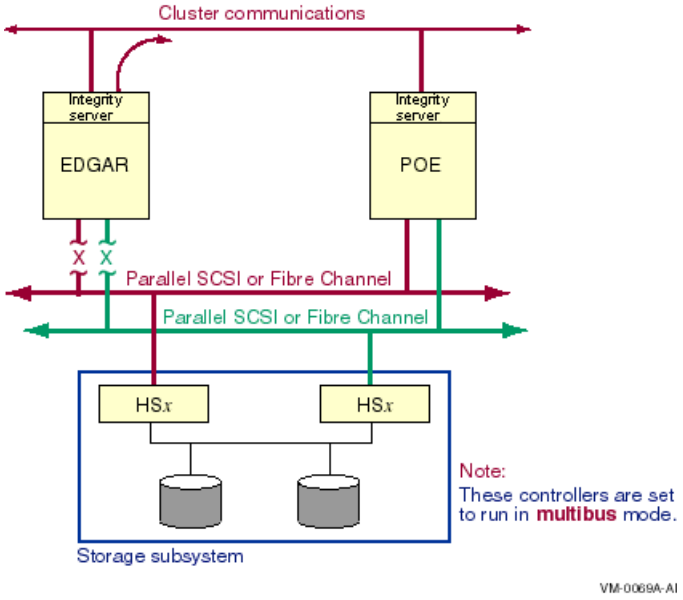
In this document, the capability to fail over from direct SCSI to MSCP served paths implies the ability to fail over in either direction between direct and served paths.

In a direct SCSI to MSCP served failover configuration, SAS and its controllers (LSI 1068 and LSI 1068e) can also be used.

6.1.3. Configurations Combining Both Types of Multipath Failover

In a multihost SCSI OpenVMS cluster system, you can increase disk 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.

In this type of configuration, SAS and its controllers (LSI 1068 and LSI 1068e) can also be used.

6.2. Configuration Requirements and Restrictions

The requirements for multipath SCSI and FC configurations are presented in Table 6.1.

Table 6.1. Multipath SCSI and FC Configuration Requirements

Component	Description
Host adapter	For parallel SCSI, the KZPBA-CB must be used. It is the only SCSI host adapter that supports disk multipath failover on OpenVMS. For Fibre Channel, both the KGPSA-BC and the KGPSA-CA support multipath failover on OpenVMS.
Alpha console firmware	For systems with HSZ70 and HSZ80, the minimum revision level is 5.3 or 5.4, depending on your AlphaServer. 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. For MDR, the minimum revision level is 1170.

Component	Description
Controller module mode	Must be set to multibus mode (disks only). 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 6.7.11), or by shutting down the partially connected controller. • Disconnect the partially connected host from both controllers.
Allocation classes	For parallel SCSI, a valid HSZ allocation class is required (see Section 6.5.3). 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 for that bus. That is, the adapters require neither a matching port allocation class nor a matching node allocation class and matching OpenVMS adapter device names. However, if there are non-HSZ devices on the bus, or HSZ controllers without an HSZ allocation class, you must follow the standard rules for shared SCSI buses for node and port allocation class assignments and for controller device names. Booting from devices with an HSZ allocation class is supported on all AlphaServers that support the KZPBA-CB except for the AlphaServer 2 *x*00(A). The controller allocation class is not used for FC devices. **Note**: If you are using Volume Shadowing for OpenVMS, every disk must have a nonzero allocation class. All FC disks attached to HSG and HSV controllers are automatically assigned the allocation class value of 1, which satisfies the volume shadowing requirement.

The restrictions for multipath FC and SCSI configurations are presented in Table 6.2.

Table 6.2. Multipath FC and SCSI Configuration Restrictions

<table>
<tr><th>Capability</th><th colspan="2">Description</th></tr>
<tr><td>Devices supported</td><td colspan="2">DKDRIVER disk devices attached to HSZ70, HSZ80,HSG60, HSG80, and HSV110 controller modules are supported. MKDRIVER tapes and GKDRIVER tape robots attached to an MDR are supported. Other device types, such as tapes, and generic class drivers, such as GKDRIVER, are not supported.

Note that under heavy load, a host-initiated manual or automatic switch from one controller to another may fail on an HSZ70 or HSZ80 controller. Testing has shown this to occur infrequently. This problem has been fixed for the HSZ70with the firmware HSOF V7.7 (and later). The problem will be fixed for the HSZ80 in a future release. This problem does not occur on HSG x or HSV xcontrollers, nor on the MDR.</td></tr>
<tr><td>Mixed-version and mixed-architecture clusters</td><td colspan="2">All hosts that are connected to an HSZ, HSG, or HSV in multibus mode must be running OpenVMS Version 7.2 or higher.

To use multipath failover to a served path, MPDEV_REMOTE must be enabled on all systems that have direct access to shared SCSI or Fibre Channel devices.</td></tr>
<tr><td>SCSI to MSCP failover
MSCP to SCSI failover</td><td colspan="2">Multiple hosts must be attached to SCSI disk devices via a shared SCSI bus (either parallel SCSI or Fibre Channel). The MPDEV_REMOTE system parameter must be set to 1 on these hosts.</td></tr>
<tr><td>Volume Shadowing for OpenVMS</td><td colspan="2">The use of default settings for certain system parameters may lead to the occasional removal of shadow set members that are configured for multipath support. The shadow set members where this has been observed are using Volume Shadowing for OpenVMS.

Therefore, when configuring multipath shadow sets using Volume Shadowing for OpenVMS, observe the following recommended setting for these system parameters:
<table>
<tr><th>System Parameter</th><th>Recommended Setting</th></tr>
<tr><td>MSCP_CMD_TMO</td><td>60 as a minimum. The value of 60 is appropriate for most configurations. Some configurations may require a higher setting.</td></tr>
<tr><td>SHADOW_MBR_TMO</td><td>At least 3 x MSCP_CMD_TMO.</td></tr>
<tr><td>SHADOW_SYS_TMO</td><td>At least 3 x MSCP_CMD_TMO.</td></tr>
<tr><td>MVTIMEOUT</td><td>At least 4 x SHADOW_MBR_TMO.</td></tr>
</table></td></tr>
</table>

6.3. HS *x* Failover Modes

The HSZ70, HSZ80, and HSG *x* implement two modes of failover operation when they are in a dual-redundant configuration, transparent failover mode, and multibus failover mode. HSV *x* supports multibus failover only.

Note

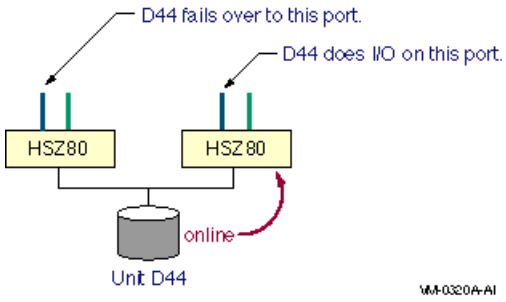
Starting with OpenVMS Alpha Version 7.3, transparent failover mode for the HSG *x* is supported.

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, as described in the following sections.

6.3.1. 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. As shown in Figure 6.4, when the HSZ detects that a controller module has failed, it moves the logical unit to the corresponding port on the surviving controller.

Figure 6.4. Storage Subsystem in Transparent Mode



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. OpenVMS has supported transparent failover for HSZ controllers since Version 6.2.

To select transparent failover mode on the HSZ or HSG, enter one of the following commands at the console, depending on your configuration:

```
HSZ> SET FAILOVER COPY=THIS_CONTROLLER
```

or

```
HSZ> SET FAILOVER COPY=OTHER_CONTROLLER
```

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

```
z70_A => SHOW THIS_CONTROLLER
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
```

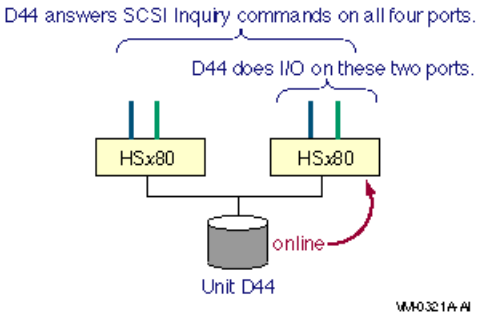
6.3.2. Multibus Failover Mode (Disks Only)

In multibus failover mode, the HS *x* responds to SCSI Inquiry commands from 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 not 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 on line and thus capable of doing I/O on the ports of only one controller at a time. Different logical units may be on line to different controllers, but an individual logical unit is on line to only one controller at a time, as shown in Figure 6.5.

Figure 6.5. Storage Subsystem in Multibus Mode



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

```
z70_A => SHOW 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, then different hosts can access the same logical unit over different ports of the controller to which the logical unit is on line.

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 one of the following commands at the HS *x*: console, whichever is appropriate to your configuration:

```
HSZ> SET MULTIBUS_FAILOVER COPY=THIS_CONTROLLER
```

or

```
HSZ> SET MULTIBUS_FAILOVER COPY=OTHER_CONTROLLER
```

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

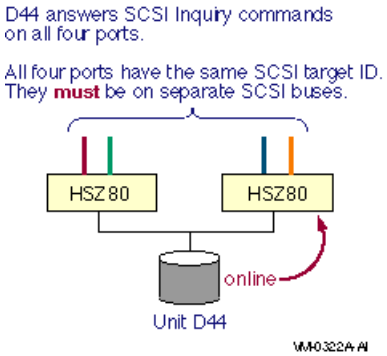
```
z70_B => SHOW THIS_CONTROLLER
Controller:
        HSZ70 ZG64100136 Firmware XB32-0, Hardware CX25
        Configured for MULTIBUS_FAILOVER with ZG64100160
            In dual-redundant configuration
        Device Port SCSI address 6
        Time: NOT SET
Host port:
        SCSI target(s) (0, 2, 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
```

6.3.3. Port Addressing for Controllers in Multibus Mode

There is a difference between parallel SCSI and FC in the way that the ports on multibus controllers are addressed. In parallel SCSI (the HSZ70 and the HSZ80), all the ports are assigned the same SCSI target IDs. This is noted for the HSZ80 configuration shown in Figure 6.6.

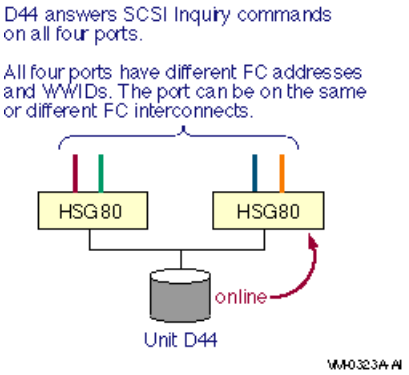
Figure 6.6. Port Addressing for Parallel SCSI Controllers in Multibus Mode



The reason all the ports have the same target ID is that the target ID is part of the OpenVMS device name (for example, the 6 in 4DKC600),and the device name must be the same for all paths. This means that each port must be on a separate SCSI bus or there will bean address conflict.

In Fibre Channel configurations with the HSG *x* or the HSV *x*, all the ports have their own FC address and WWID, as noted in Figure 6.7. The same is true for the MDR.

Figure 6.7. Port Addressing for Fibre Channel Controllers in Multibus Mode



The ports on the HSG *x* and the HSV *x* have separate FC addresses and WWIDs because these items are not used in the OpenVMS FC device name. This means that any number of ports can be connected to the same FC interconnect. In fact, all the ports of the HSG *x* or HSV *x* in multibus mode should be connected, even if there is just one interconnect, because this can improve availability and performance.

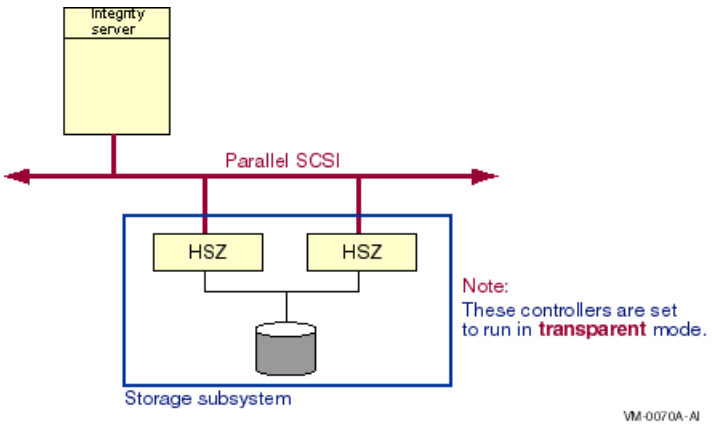
6.4. Parallel SCSI Multipath Configurations (Disks Only)

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, such as the HSZ80, are also described.

6.4.1. Transparent Failover

Transparent failover in a parallel SCSI configuration, as shown in Figure 6.8, requires that both controller modules be on the same SCSI bus.

Figure 6.8. Parallel SCSI Configuration With Transparent Failover



In this configuration:

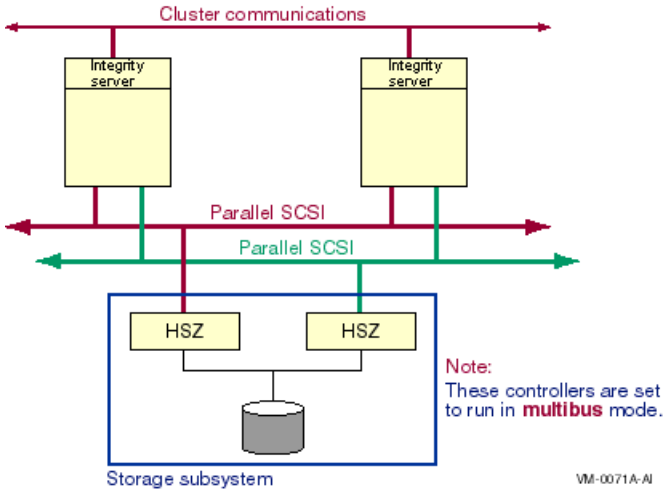
- Each logical unit is visible to the host on only one controller module at a time. The other controller module does not answer at the same SCSI address, but it can be used for other SCSI addresses.

- 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.

6.4.2. 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 6.9 shows this configuration.

Figure 6.9. 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.

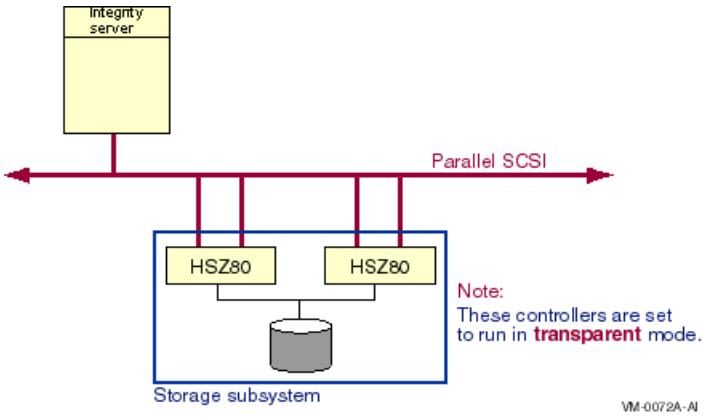
6.4.3. Configurations Using Multiported Storage Controllers

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 6.10 shows a single host with a single interconnect, using an HSZ80 in transparent mode.

Figure 6.10. Multiported Parallel SCSI Configuration With Single Interconnect in Transparent Mode

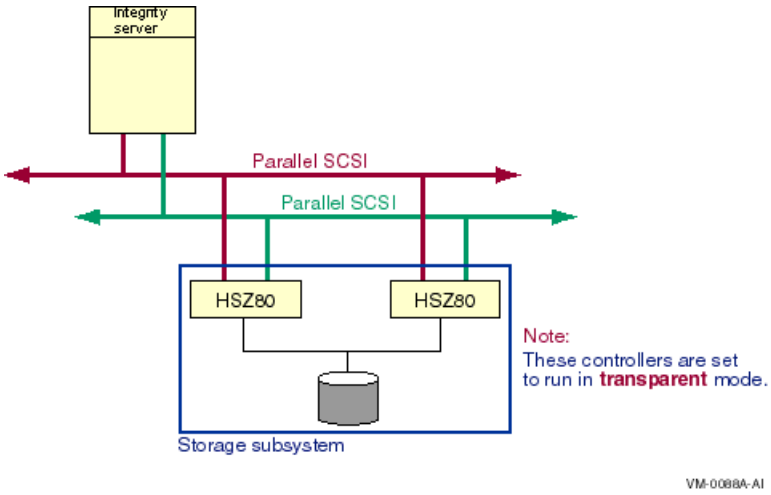


Note the following about this configuration:

- 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 6.11 shows a system configured in transparent mode using two paths from the host.

Figure 6.11. Multiported Parallel SCSI Configuration With Multiple Paths in Transparent Mode



In this configuration:

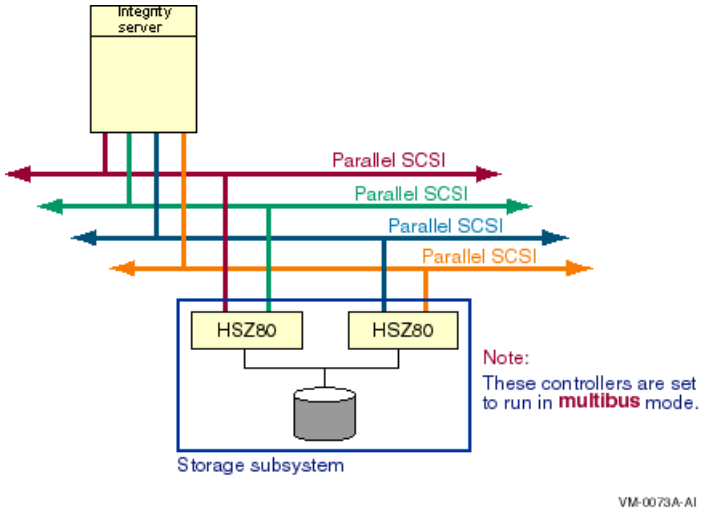
- Physically corresponding ports must be on the same SCSI bus.
- A maximum of two buses can be connected to each storage controller.

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 6.10 because both SCSI buses can be simultaneously active. This configuration does not have higher availability, however, because there is still only one path from the host to the logical unit.

Figure 6.12 shows a system using the multiported HSZ80 storage controller configured in multibus mode.

Figure 6.12. 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.

6.5. Disk 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 toaccess 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.

6.5.1. Review of Node Allocation Classes

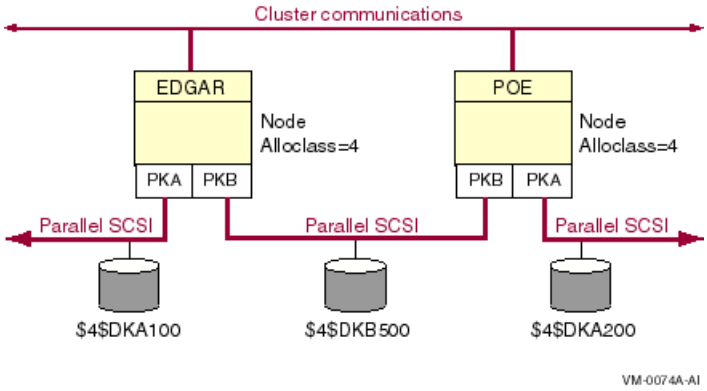
A node allocation class is used in a device name in place of a node name. Anode allocation class is needed to produce a unique device name when multiple nodes have a direct connection to the same SCSI device.

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 6.13 shows a configuration whose devices are named using a node allocation class.

Figure 6.13. Devices Named Using a Node Allocation Class



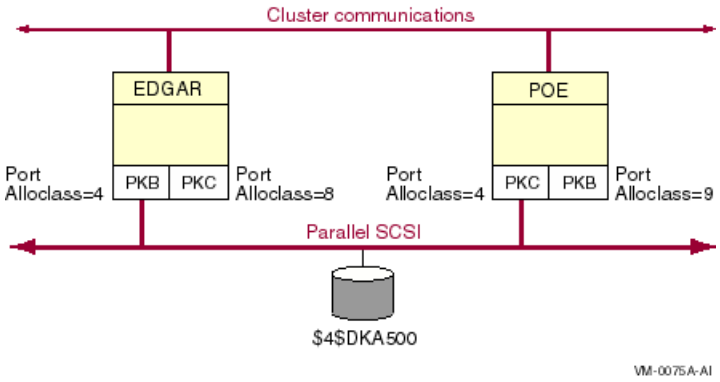
6.5.2. Review of Port Allocation Classes

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 6.14 shows a configuration whose devices are named using a port allocation class.

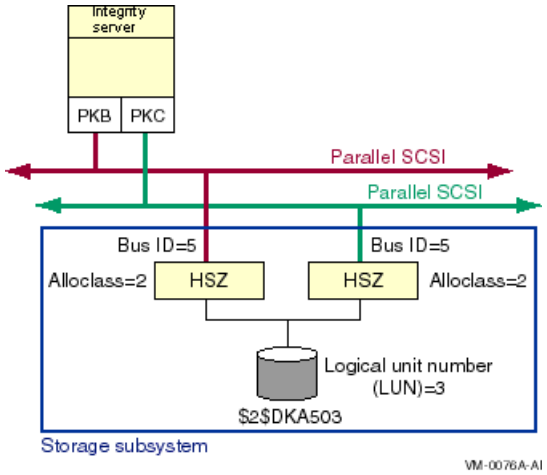
Figure 6.14. Devices Named Using a Port Allocation Class



6.5.3. 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 6.15 must be used.

Figure 6.15. 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 one of the following commands, as appropriate to the configuration:

```
HSZ> SET THIS_CONTROLLER ALLOCATION_CLASS = n
```

or

```
HSZ> SET 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 ALLOCATION_CLASS=199
z70_B => SHOW THIS_CONTROLLER
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 => SHOW OTHER_CONTROLLER
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
```

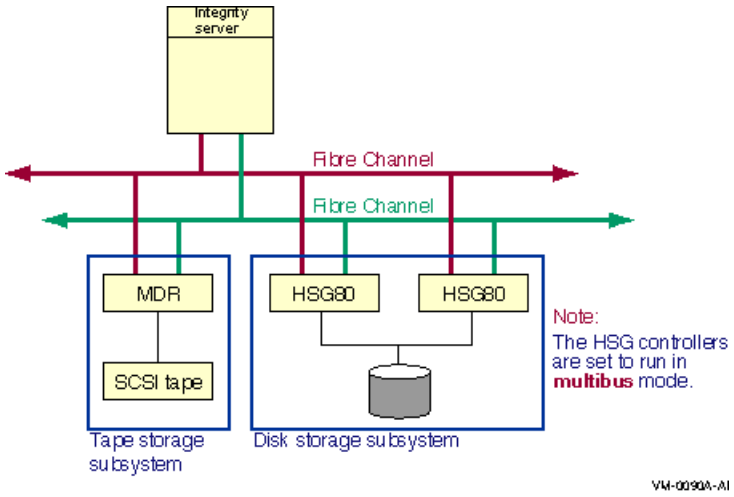
The following rules pertain to the use of an HSZ allocation class in SCSI device names:

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 HSZ allocation class number must be unique among all types of allocation classes throughout the cluster.
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.

6.6. Fibre Channel Multipath Configurations

Figure 6.16 shows a multipath configuration with both a tape storage subsystem and a disk storage subsystem. Note that the disk storage controllers are configured in multibus mode.

Figure 6.16. Single Host With Two Dual-Ported Storage Controllers, One Dual-Ported MDR, and Two Buses



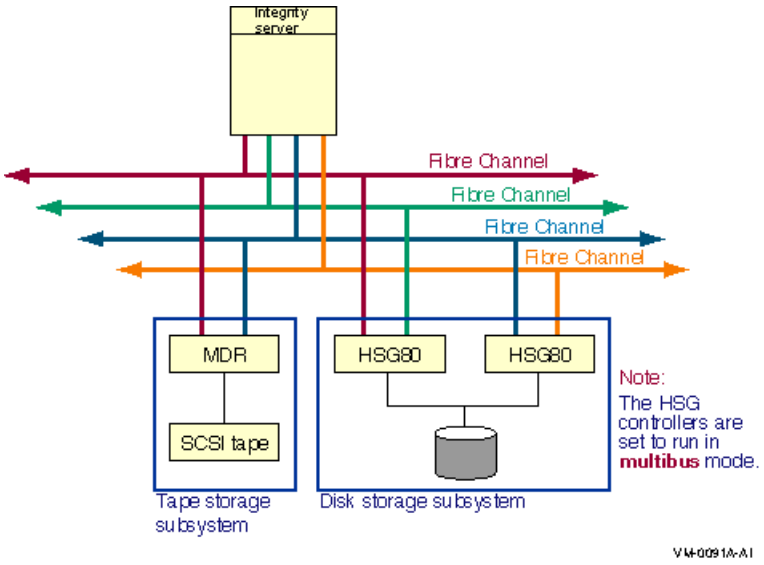
Note the following about this configuration:

- Host has two adapters, each attached to a different bus.
- Each port on each HSG *x* or HSV *x* storage controller is attached to a different interconnect.
- Each port on the Modular Data Router (MDR) or the Network Storage Router(NSR) is attached to a different interconnect.
- Both storage controllers can access the same disk.
- Host has four paths to the same logical unit.

Note that each HSG80 port has its own Fibre Channel address and Fibre Channel port WWID. This is different from an HSZ80 in multibus mode where all the ports respond to the same SCSI address and must, therefore, be connected to different SCSI buses. The separate FC addresses enable both ports of the dual HSG80 to be on the same FC.

Figure 6.17 is similar to Figure 6.16, except it has two additional Fibre Channel interconnects.

Figure 6.17. Single Host With Two Dual-Ported Storage Controllers, One Dual-Ported MDR, and Four Buses

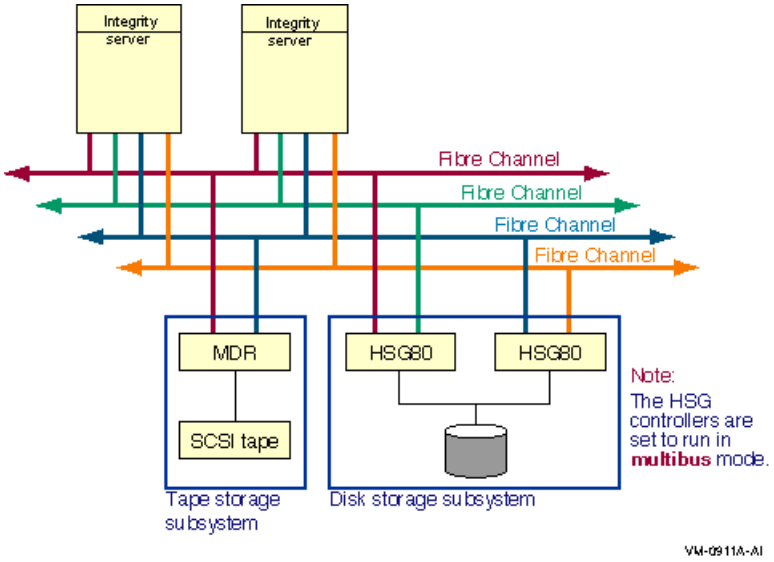


Note the following about this configuration:

- Host has four adapters, each attached to a different interconnect.
- Each port on each HSG *x* or HSV *x* storage controller is attached to a different interconnect.
- Each port on the Modular Data Router (MDR) or the Network Storage Router(NSR) is attached to a different interconnect.
- Host has four paths to the same logical unit.

Figure 6.18 builds on the previous two figures. Instead of a single host, it has two hosts.

Figure 6.18. Two Hosts With Two Dual-Ported Storage Controllers, One Dual-Ported MDR, and Four Buses



Note the following about this configuration:

- Each host has four adapters, each attached to a different interconnect.
- Each port on each HSG *x* or HSV *x* storage controller is attached to a different interconnect.
- Each port on the Modular Data Router (MDR) or the Network Storage Router(NSR) is attached to a different interconnect.
- Each host has four paths to the same logical unit of the disk storage subsystem and two paths to the tape storage subsystem.

6.7. Implementing Multipath Configurations

Parallel SCSI, SAS, 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.

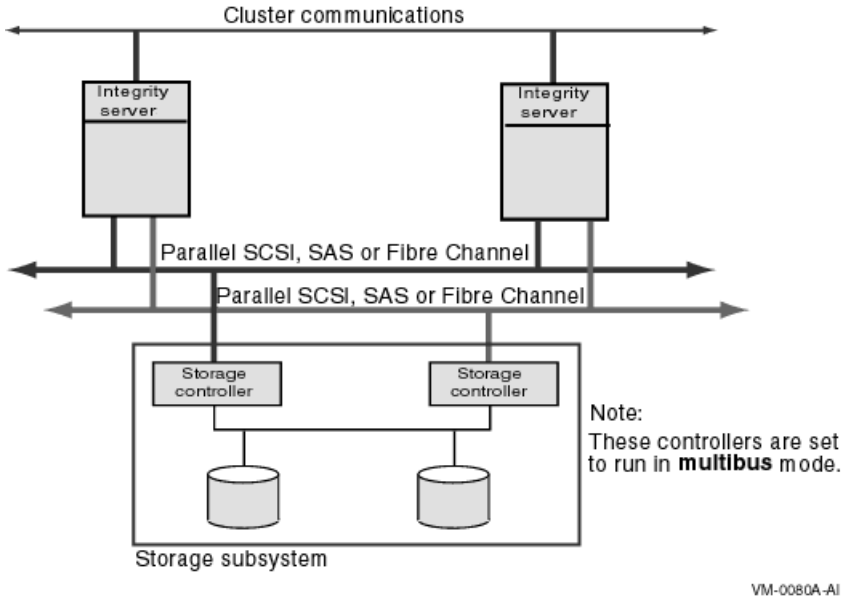
Implementing multiple paths to devices consists of the following steps:

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 (disks only).
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*.

6.7.1. Valid Multipath Configurations

Figure 6.19 shows a valid multipath, multihost configuration.

Figure 6.19. Two Hosts With Shared Buses and Shared Storage Controllers



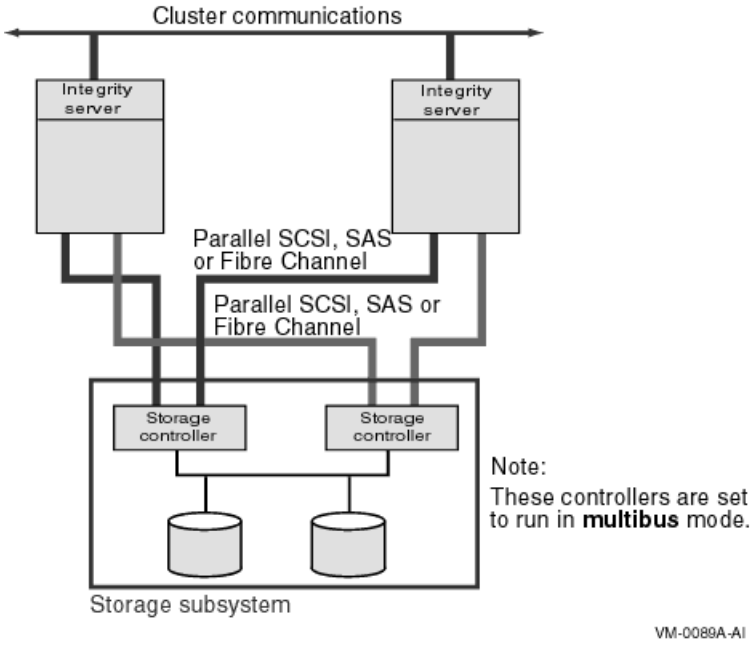
Note the following about this configuration:

- 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 6.20 shows a valid multipath configuration for systems that are not configured on the same bus.

Figure 6.20. Two Hosts With Shared, Multiported Storage Controllers



Note the following about this configuration:

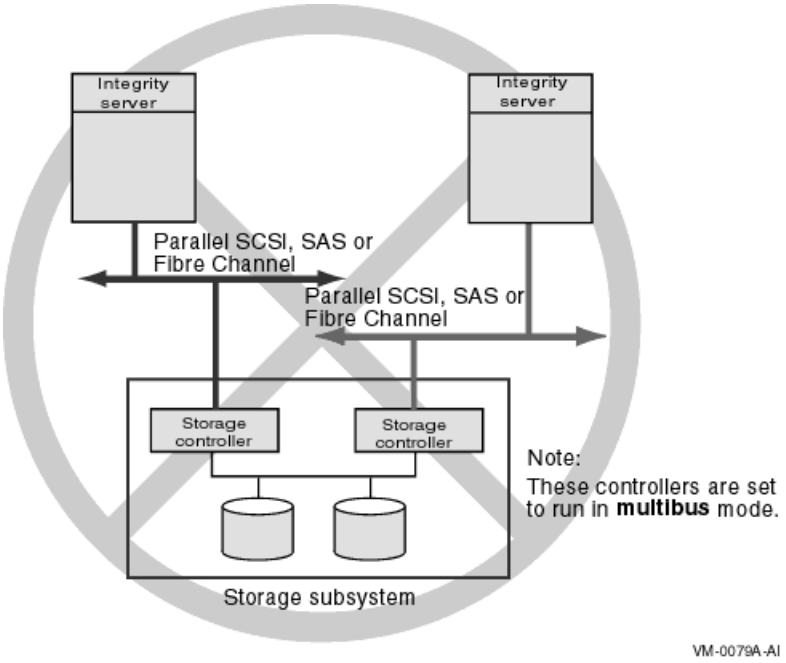
- Each host has two adapters.
- Each host is connected to two buses but the hosts do not share a bus.
- Both hosts share the storage.
- Each storage controller has two connections, one to each host.
- The two storage controllers are connected to the same disks.

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

6.7.2. Invalid Multipath Configuration

Figure 6.21 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 6.21. Invalid Multipath Configuration



6.7.3. Multipath System Parameters

Multipath support is enabled and qualified by the use of the system parameters described in Table 6.3. (Certain multipath system parameters are 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). When set to OFF (0), the formation of additional multipath sets and the addition of new paths to existing multipath sets is disabled. However, existing multipath sets remain in effect. The default is ON. MPDEV_REMOTE and MPDEV_AFB_INTVL have no effect when MPDEV_ENABLE is set to OFF.
MPDEV_LCRETRIES	Controls the number of times the system retries direct paths to the controller that the logical unit is on line to, before moving on to direct paths to the other controller, or to an MSCP served path to the device (MSCP paths apply only to disks). 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. When set to OFF (0), multipath polling is disabled. The default is ON. Note that this parameter must be set to ON to use the automatic failback feature.
MPDEV_REMOTE (disks only)	Enables MSCP served paths to become members of a multipath set when set to ON (1). When set to OFF (0),only local paths to a SCSI or Fibre Channel device are used in the formation of additional multipath sets. MPDEV_REMOTE is enabled by default. However, setting this parameter to OFF has no effect on existing multipath sets that have remote paths.

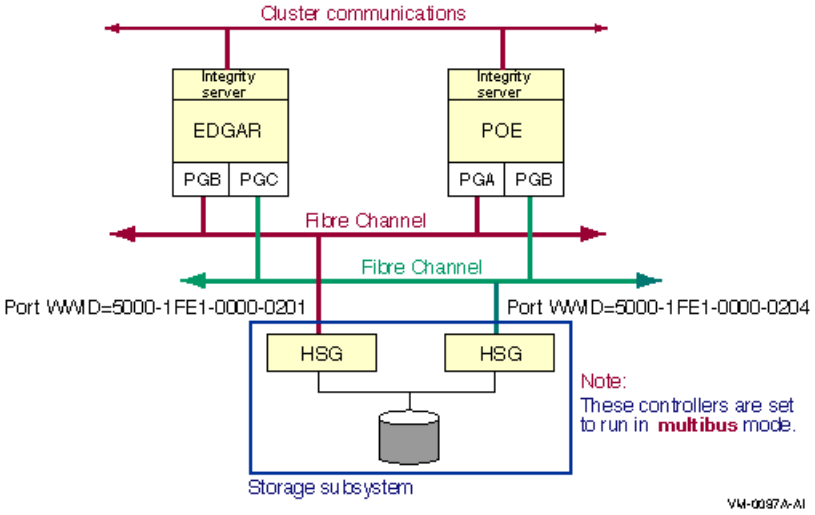
Parameter	Description
	To use multipath failover to a served path, MPDEV_REMOTE must be enabled on all systems that have direct access to shared SCSI or Fibre Channel devices. The first release to provide this feature is OpenVMS Alpha Version 7.3–1. Therefore, all nodes on which MPDEV_REMOTE is enabled must be running OpenVMS Alpha Version 7.3–1 (or later). If MPDEV_ENABLE is set to OFF (0), the setting of MPDEV_REMOTE has no effect because the addition of all new paths to multipath sets is disabled. The default is ON.
MPDEV_AFB_INTVL (disks only)	Specifies the automatic failback interval in seconds. The automatic failback interval is the minimum number of seconds that must elapse before the system will attempt another failback from an MSCP path to a direct path on the same device. MPDEV_POLLER must be set to ON to enable automatic failback. You can disable automatic failback without disabling the poller by setting MPDEV_AFB_INTVL to 0. The default is 300 seconds.
MPDEV_D1	Reserved for use by the operating system.
MPDEV_D2	Reserved for use by the operating system.
MPDEV_D3	Reserved for use by the operating system.
MPDEV_D4	Reserved for use by the operating system.

6.7.4. 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 6.22, the path identifier for the path from the host adapter on the left to the HSG storage controller on the left is PGB0.5000-1FE1-0000-0201. (The second port on each HSG is omitted for convenience). You can obtain the WWID for a storage controller from its console.

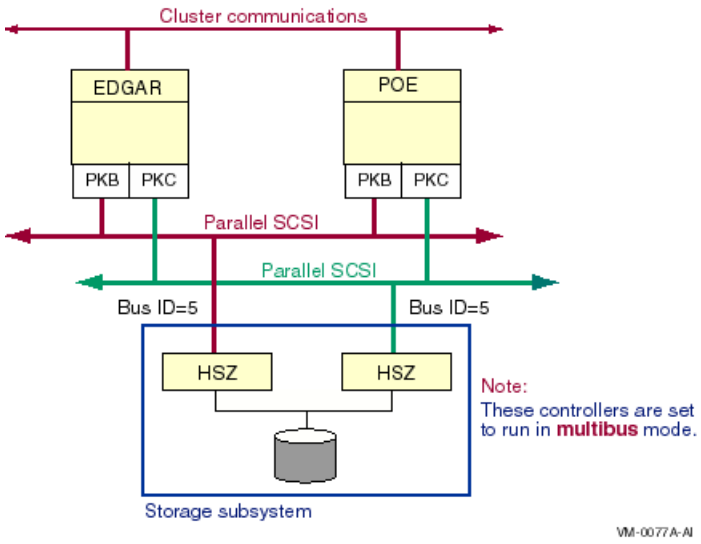
Figure 6.22. Fibre Channel Path Naming



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 6.23, 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 6.23. Configuration With Multiple Direct Paths



6.7.5. 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 DEVICEDCL 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 6.7.4).

The SHOW DEVICE/MULTIPATH_SET *device-name* command lists only some brief multipath information about devices that have multiple paths.

Multipath information is displayed only on nodes that are directly connected to the multipath device.

6.7.5.1. 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 the counts for each path.

```
$ SHOW DEVICE/FULL $1$DGA23:

Disk $1$DGA23: (WILD8), device type HSG80, 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                    3    Operations completed          32814199
  Owner process                 ""    Owner UIC                     [SYSTEM]
```

```
 Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
 Reference count                9    Default buffer size                 512
 WWID   01000010:6000-1FE1-0000-0D10-0009-8090-0677-0034
 Total blocks            17769177    Sectors per track                   169
 Total cylinders             5258    Tracks per cylinder                  20
 Host name                "WILD8"    Host type, avail HP AlphaServer GS160
6/731, yes
 Alternate host name     "W8GLX1"    Alt. type, avail HP AlphaServer GS160
6/731, yes
 Allocation class               1

 Volume label      "S5SH_V72_SSS"    Relative volume number               0
 Cluster size                  18    Transaction count                    8
 Free blocks             12812004    Maximum files allowed           467609
 Extend quantity                5    Mount count                          8
 Mount status              System    Cache name          "_$1$DGA8:XQPCACHE"
 Extent cache size             64    Maximum blocks in extent cache  1281200
 File ID cache size            64    Blocks currently in extent cache      0
 Quota cache size               0    Maximum buffers in FCP cache       1594
 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 XQP caching enabled, write-through XFC caching enabled. Volume is
also mounted on H2OFRD, FIBRE3, NORLMN, SISKO, BOOLA, FLAM10, W8GLX1.

 I/O paths to device              5
 Path PGA0.5000-1FE1-0000-0D12  (WILD8), primary path.
   Error count                    2    Operations completed        130666
 Path PGA0.5000-1FE1-0000-0D13  (WILD8), current path.
   Error count                    1    Operations completed        30879310
 Path PGA0.5000-1FE1-0000-0D11  (WILD8).
   Error count                    0    Operations completed        130521
 Path PGA0.5000-1FE1-0000-0D14  (WILD8).
   Error count                    0    Operations completed        130539
 Path MSCP (W8GLX1).
   Error count                    0    Operations completed        1543163
```

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
WWID	The worldwide ID of the SCSI logical unit.
Host name	The name of the system that is being used by the current path. The host name is displayed if there is an MSCP path to a multipath device.
Alternate host name	The name of another system that can also provide access to the device. If the current path is a direct path, this will be the host currently associated with the MSCP path. If the current path is an MSCP path, this will be the name of the local system. The alternate host name is displayed if there is an MSCP path to a multipath disk device.
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.

Term	Description
Polling disabled	The DCL command SET DEVICE/NOPOLL has been executed for this path.
Not responding	This path to the device was unusable the last time it was checked. Typically, the multipath poller checks every 60 seconds if the path is good and every 30 seconds if the path is bad.
Unavailable	The path is unavailable because the software driver has disconnected from the path.

6.7.5.2. 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 number of accessible paths can be less than the total number of paths for two reasons:

- A system manager disabled the path with the command SETDEVICE/PATH= *pathname*/ NOENABLE.
- If a path is designated as Not Responding, the operating system decrements the total number of paths. This action was introduced in OpenVMS Alpha Version 7.3–1.

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.

```
$ SHOW DEVICE/MULTIPATH
Device                  Device           Error  Paths  Current
 Name                   Status           Count Avl/Tot   path
$1$DGA8:      (H2OFRD)  Mounted              3   5/ 5  PGA0.5000-1FE1-0000-0D12
$1$DGA10:     (H2OFRD)  ShadowSetMember      1   5/ 5  PGA0.5000-1FE1-0000-0D14
$1$DGA11:      (WILD8)  ShadowSetMember      3   3/ 3  MSCP
$1$DGA23:     (H2OFRD)  Mounted              6   5/ 5  PGA0.5000-1FE1-0000-0D13
$1$DGA30:     (H2OFRD)  ShadowSetMember      8   5/ 5  PGA0.5000-1FE1-0000-0D13
$1$DGA31:      (WILD8)  ShadowMergeMbr       5   3/ 3  MSCP
$1$DGA33:     (H2OFRD)  Online               0   5/ 5  PGA0.5000-1FE1-0000-0D12
$1$DGA40:     (H2OFRD)  Mounted              2   5/ 5  PGA0.5000-1FE1-0000-0D13
$1$DGA41:     (H2OFRD)  ShadowMergeMbr       8   5/ 5  PGA0.5000-1FE1-0000-0D12
$70$DKA100:   (H2OFRD)  Mounted              0   3/ 3  PKD0.1
$70$DKA104:   (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.1
$70$DKA200:   (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.2
$70$DKA300:   (H2OFRD)  ShadowSetMember      0   3/ 3  PKC0.3
$80$DKA1104:  (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.11
$80$DKA1200:  (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.12
$80$DKA1204:  (H2OFRD)  ShadowSetMember      0   3/ 3  PKC0.12
$80$DKA1207:  (H2OFRD)  Mounted              0   3/ 3  PKD0.12
$80$DKA1300:  (H2OFRD)  Mounted              0   3/ 3  PKD0.13
$80$DKA1307:  (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.13
$80$DKA1500:  (H2OFRD)  Mounted              0   3/ 3  PKD0.15
$80$DKA1502:  (H2OFRD)  ShadowSetMember      0   3/ 3  PKD0.15
```

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.

6.7.6. 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 or to an MDR 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 messages, to the console and to the operator's log:

```
All multipath devices on path PKB0.5 are either disabled or not reachable.
```

or

```
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.

6.7.7. 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, MDRs, and buses.

The command syntax for switching the current path is:

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

This command requires the OPER privilege. Additionally, if the device is currently allocated by another process, as tape devices often are, the SHARE privilege is needed.

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

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

Note that this command initiates the process of switching the path and then returns to the DCL prompt immediately. A delay may occur between when the DCL prompt reappears and when the path switch is complete.

A manual path switch of a mounted device takes place within mount verification, which is triggered by the path switch command. It is accompanied by the usual mount verification messages and a path switch message, as shown in Example 6.1.

Example 6.1. Messages Resulting from Manual Path Switch

```
%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:23.05  %%%%%%%%%%
Device $1$DGA23: (H2OFRD PGA) is offline.
Mount verification is in progress.
```

```
%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:25.76  %%%%%%%%%%
09:04:25.76 Multipath access to device $1$DGA23: has been manually
switched
from path PGA0.5000-1FE1-0000-0D11 to path PGA0.5000-1FE1-0000-0D14

%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:25.79  %%%%%%%%%%
Mount verification has completed for device $1$DGA23: (H2OFRD PGA)
```

You can check for completion of a path switch by issuing the SHOWDEVICE/FULL command, or the SHOW DEVICE/MULTIPATH command.

Note that if the path that is designated in a manual path switch fails during the switch operation, then automatic path switching takes over. This can result in a switch to a path different from the one designated in the command.

If a manual path switch causes a logical unit to switch from one HSG80 controller to another controller, then the command can affect other nodes in the cluster. These nodes will experience a mount verification on their current path, causing an automatic switch to a path on the other HSG80 controller. Example 6.2 shows the messages that indicate this event.

Example 6.2. Messages Displayed When Other Nodes Detect a Path Switch

```
%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:26.48  %%%%%%%%%%
Device $1$DGA23: (WILD8 PGA, H20FRD) is offline.
Mount verification is in progress.

%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:26.91  %%%%%%%%%%
09:04:29.91 Multipath access to device $1$DGA23: has been auto switched
from
path PGA0.5000-1FE1-0000-0D12 (WILD8) to path PGA0.5000-1FE1-0000-0D13
(WILD8)

%%%%%%%%%%  OPCOM  15-JUN-2001 09:04:27.12  %%%%%%%%%%
Mount verification has completed for device $1$DGA23: (WILD8 PGA, H20FRD)
```

The WILD8 node name is displayed for each path because each path is a direct path on node WILD8. The node name field in the mount verification `in progress` and `completed` messages shows both a local path and an MSCP alternative. In this example, the `WILD8 PGA, H20FRD` name shows that a local PGA path on WILD8 is being used and that an MSCP path via node H20FRD is an alternative.

6.7.8. Path Selection by OpenVMS

The selection of the current path to a multipath device is determined by the device type as well as by the event that triggered path selection.

Path Selection for Initial Configuration at System Startup

When a new path to a multipath disk (DG, DK) or tape device (MG) is configured, the path chosen automatically as the current path is the direct path with the fewest devices. No operator messages are displayed when this occurs. (This type of path selection is introduced in OpenVMS Alpha Version 7.3-1.) A DG, DK, or MG device is eligible for this type of path selection until the device's first use after a system boot or until a manual path switch is performed by means of the SET DEVICE/SWITCH command.

When a new path to a generic multipath SCSI device (GG, GK) is configured, the path chosen automatically as the current path is the first path discovered, which is also known as the primary path.

For GG and GK devices, the primary path remains the current path even as new paths are configured. GG and GK devices are typically the console LUNs for HSG or HSV controller LUNs or for tape media robots.

Path Selection When Mounting a Disk Device

The current path to a multipath disk device can change as a result of a MOUNT command. The I/O performed by the MOUNT command triggers a search for a direct path that does not require the disk device to fail over from one HS *x* controller to another.

The path selection initiated by a MOUNT command on a DG or DK disk device proceeds as follows:

1. If the current path is a direct path and access to the device on this path does not require a controller failover, the current path is used.

2. If the current path is an MSCP path and it was selected by a manual path switch command, the current path is used.

3. All direct paths are checked, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If a path is found on which access to the device does not require a controller failover, that path is selected. If the selected path is not the current path, an automatic path switch is performed and an OPCOM message is issued.

4. All direct paths are tried, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If necessary, an attempt is made to fail over the device to the HS *x* controller on the selected path. If the selected path is not the current path, an automatic path switch is performed and an OPCOM message is issued.

5. The MSCP served path is tried. If the MSCP path is not the current path, an automatic path switch is performed and an OPCOM message is issued.

The MOUNT utility might trigger this path selection algorithm a number of times until a working path is found. The exact number of retries depends on both the time elapsed for the prior attempts and the qualifiers specified with the MOUNT command.

This path selection process, introduced in OpenVMS Alpha Version 7.3-1, has the following benefits:

- Minimizes the disruption on other hosts in the cluster.

- Tends to preserve any static load balancing that has been manually set up on other nodes in the cluster.

- Enables the use of HS *x* console commands to set up an initial default distribution of devices between the two HS *x* controllers.

- Tends to balance the use of available paths from this host to the disk devices.

- Prefers direct paths over MSCP served paths.

Note that this selection process allows devices to be distributed between the two HS *x* controllers. You can accomplish this by using HS *x* console commands, such as the following:

```
HSG> SET UNIT PREFERRED_PATH=THIS_CONTROLLER
HSG> SET UNIT PREFERRED_PATH=OTHER_CONTROLLER
```

In addition, you can use the DCL commands for manual path switching described in Section 6.7.7, to select a different host bus adapter or a different port on the same HS *x* controller, or to force the device to fail over to a different HS *x* controller.

Path Selection When Mounting Tape Drive Device

Support for multipath tape drives and this type of path selection was introduced in OpenVMS Alpha Version 7.3-1.Path selection when the MOUNT command is issued differs somewhat between multipath tape drives and disk devices for several reasons:

- Tape drives are not concurrently shareable by multiple hosts.
- Tape drives do not present the same concerns as disks do for disrupting I/O being performed by another host.
- There is no failover between direct and MSCP served paths to multipath tape devices.

The path selection initiated by a MOUNT command on an MG tape drive device proceeds as follows:

1. The current path is used if possible, even if a controller failover is required.
2. The direct paths are checked, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If a path is found on which access to the device does not require a controller failover, that path is selected. If the selected path is not the current path, an automatic path switch is performed and an OPCOM message is issued.
3. The direct paths are checked again, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If the selected path is useable and is not the current path an automatic paths witch is performed and an OPCOM message is issued.

6.7.9. How OpenVMS Performs Multipath Failover

When an I/O operation fails on a device that is subject to mount verification, and the failure status suggests that retries are warranted, mount verification is invoked. If the device is a multipath device or a shadow set that includes multipath devices, alternate paths to the device are automatically tried during mount verification. This allows the system to recover transparently from cases in which the device has failed over from one HS *x*controller or MDR to another, and to recover transparently from failures in the path to the device.

The following devices are subject to mount verification:

- Disk devices that are mounted as Files-11 volumes, including host-based volume shadowing sets
- Tape devices that are mounted as ANSI tape volumes
- Tape devices that are mounted as foreign volumes

Note that foreign mounted disk volumes and generic SCSI devices (GG and GK) are not subject to mount verification and, therefore, are not eligible for automatic multipath failover.

Path selection during mount verification proceeds as follows:

1. If the current path is a direct path and access to the device on this path does not require controller failover, the current path is tried.

2. If the current path is an MSCP path and it was selected by a manual path switch command, the current path is tried (disks only).

3. All direct paths are checked, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If a path is found on which access to the device does not require a controller failover, that path is selected.

4. Step 3 is repeated the number of times specified by the MPDEV_LCRETRIES system parameter. This provides additional bias toward selection of a path that does not require an HS *x* or MDR controller failover. The default value for MPDEV_LCRETRIES is 1.

5. All direct paths are tried, starting with the path that is the current path for the fewest devices. The direct paths are considered in order of increasing use as a current path for other devices. If necessary, an attempt is made to fail over the device to the HS *x* or MDR controller on the selected path.

6. If present, the MSCP served path is tried (disks only).

Steps 1 through 6 are repeated until either a working path is found or mount verification times out. If a working path is found and the path is different, the current path is automatically switched to the new path and an OPCOM message is issued. Mount verification then completes, the failed I/O is restarted, and new I/O is allowed to proceed. This path selection procedure attempts to avoid unnecessary failover of devices from one HS *x* controller 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.

This path selection procedure prefers direct paths over MSCP served paths because the use of an MSCP served path imposes additional CPU and I/O overhead on the server system. In contrast, the use of a direct path imposes no additional CPU or I/O overhead on the MSCP-server system. This procedure selects an MSCP served path only if none of the available direct paths are working. Furthermore, this path selection procedure tends to balance the use of available direct paths, subject to the constraint of avoiding unnecessary controller failovers.

6.7.10. Automatic Failback to a Direct Path (Disks Only)

Multipath failover, as described in Section 6.7.9, applies to MSCP served paths as well. That is, if the current path is via an MSCP served path and the served path fails, mount verification can trigger an automatic failover back to a working direct path.

However, an I/O error on the MSCP path is required to trigger the failback to a direct path. Consider the following sequence of events:

- A direct path is being used to a device.

- All direct paths fail and the next I/O to the device fails.

- Mount verification provokes an automatic path switch to the MSCP served path.

- Sometime later, a direct path to the device is restored.

In this case, the system would continue to use the MSCP served path to the device, even though the direct path is preferable. This is because no error occurred on the MSCP served path to provoke the path selection procedure.

The automatic failback feature is designed to address this situation. Multipath polling attempts to fail back the device to a direct path when it detects that all of the following conditions apply:

- A direct path to a device is responsive.
- The current path to the device is an MSCP served path.
- The current path was not selected by a manual path switch.
- An automatic failback has not been attempted on this device within the last MPDEV_AFB_INTVL seconds.

The automatic failback is attempted by triggering mount verification and, as a result, the automatic failover procedure on the device.

The main purpose of multipath polling is to test the status of unused paths in order to avoid situations such as the following:

- A system has paths A and B to a device.
- The system is using path A.
- If path B becomes inoperative, it goes unnoticed.
- Much later, and independently, path A breaks.
- The system attempts to fail over to path B but finds that it is broken.

The poller would detect the failure of path B within 1 minute of its failure and would issue an OPCOM message. An alert system manager can initiate corrective action immediately.

Note that a device might successfully respond to the SCSI INQUIRY commands that are issued by the path poller but might fail to complete a path switch or mount verification successfully on that path. A system manager or operator can control automatic failback in three ways:

1. Specify the minimum interval between automatic failback attempts on any given device by the MPDEV_AFB_INTVL system parameter. The default value is 300 seconds.
2. Prevent automatic failback by setting the value of MPDEV_AFB_INTVL to 0. If set to 0, no automatic failback is attempted on any device on this system.
3. Temporarily disable automatic failback on a specific device by manually switching the device to the MSCP served path. You can do this even if the current path is an MSCP served path.

Because of the path selection procedure, the automatic failover procedure, and the automatic failback feature, the current path to a mounted device is usually a direct path when there are both direct and MSCP served paths to that device. The primary exceptions to this are when the path has been manually switched to the MSCP served path or when there are no working direct paths.

6.7.11. 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 that 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 6.21.

6.7.12. Performance Considerations

The presence of an MSCP served path in a disk multipath set has no measurable effect on steady-state I/O performance when the MSCP path is not the current path.

Note that the presence of an MSCP served path in a multipath set might increase the time it takes to find a working path during mount verification under certain, unusual failure cases. Because direct paths are tried first, the presence of an MSCP path should not affect recovery time.

However, the ability to switch dynamically from a direct path to an MSCP served path might significantly increase the I/O serving load on a given MSCP server system with a direct path to the multipath disk storage. Because served I/O takes precedence over almost all other activity on the MSCP server, failover to an MSCP served path can affect the responsiveness of other applications on that MSCP server, depending on the capacity of the server and the increased rate of served I/O requests.

For example, a given OpenVMS Cluster configuration might have sufficient CPU and I/O bandwidth to handle an application work load when all the shared SCSI storage is accessed by direct SCSI paths. Such a configuration might be able to work acceptably as failures force a limited number of devices to switch over to MSCP served paths. However, as more failures occur, the load on the MSCP served paths could approach the capacity of the cluster and cause the performance of the application to degrade to an unacceptable level.

The MSCP_BUFFER and MSCP_CREDITS system parameters allow the system manager to control the resources allocated to MSCP serving. If the MSCP server does not have enough resources to serve all incoming I/O requests, performance will degrade on systems that are accessing devices on the MSCP path on this MSCP server.

You can use the MONITOR MSCP command to determine whether the MSCP server is short of resources. If the Buffer Wait Rate is nonzero, the MSCP server has had to stall some I/O while waiting for resources.

It is not possible to recommend correct values for these parameters. However, note that, starting with OpenVMS Alpha Version 7.2-1, the default value for MSCP_BUFFER has been increased from 128 to 1024.

As noted in the online help for the SYSGEN utility, MSCP_BUFFER specifies the number of pagelets to be allocated to the MSCP server's local buffer area, and MSCP_CREDITS specifies the number of outstanding I/O requests that can be active from one client system. For example, a system with many disks being served to several OpenVMS systems might have MSCP_BUFFER set to a value of 4000 or higher and MSCP_CREDITS set to 128 or higher.

For information about modifying system parameters, see the *VSI OpenVMS System Manager's Manual*.

VSI recommends that you test configurations that rely on failover to MSCP served paths at the worst-case load level for MSCP served paths. If you are configuring a multiple-site cluster that uses a multiple-site SAN, consider the possible failures that can partition the SAN and force the use of MSCP served paths. In a symmetric dual-site configuration, VSI recommends that you provide capacity for 50 percent of the SAN storage to be accessed by an MSCP served path.

You can test the capacity of your configuration by using manual path switching to force the use of MSCP served paths.

6.7.13. Console Considerations

This section describes how to use the console with parallel SCSI multipath disk devices. See Section 7.6 for information on using the console with FC multipath devices.

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.

```
>>>SHOW DEVICE
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
```

Chapter 7. Configuring Fibre Channel as an OpenVMS Cluster Storage Interconnect

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 support for Fibre Channel as a storage interconnect for single systems and as a shared storage interconnect for multihost OpenVMS Cluster systems. With few exceptions, as noted, this chapter applies equally to OpenVMS Alpha systems and OpenVMS Integrity server systems.

The following topics are discussed:

- Overview of OpenVMS Fibre Channel support (Section 7.1)
- Fibre Channel configuration support (Section 7.2)
- Example configurations (Section 7.3)
- Fibre Channel addresses, WWIDs, and device names (Section 7.4)
- Fibre Channel tape support (Section 7.5)
- Using the AlphaServer console for configuring Fibre Channel (Section 7.6)
- Booting on a Fibre Channel Storage Device on OpenVMS Integrity server Systems (Section 7.7)
- Setting up a storage controller for use with OpenVMS (Section 7.8)
- Creating a cluster with a shared Fibre Channel system disk (Section 7.9)
- Using interrupt coalescing for I/O performance gains (Section 7.10)
- Using Fast Path in your configuration (Section 7.11)
- Using FIBRE_SCAN for displaying devices (Section 7.12)

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

Note

The Fibre Channel interconnect is shown generically in the figures in this chapter. It 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.

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 HSG*x* controllers serve as a logical unit to a host, are shown in the figures as a single logical unit.

For ease of reference, the term HSG, which represents a Fibre Channel hierarchical storage controller, is used throughout this chapter to represent both an HSG60 and an HSG80, except where it is important to note any difference, as in the Software Product Description for Fibre Channel Hardware Components.

7.1. Overview of OpenVMS Fibre Channel Support

Fibre Channel is an ANSI standard network and storage interconnect that offers many advantages over other interconnects. Its most important features and the support offered by OpenVMS for these features are shown in Table 7.1.

Table 7.1. Fibre Channel Features and OpenVMS Support

Feature	OpenVMS Support
High-speed transmission	OpenVMS supports 2 Gb/s, full-duplex, serial interconnect (can simultaneously transmit and receive 200 MB of data per second).
Choice of media	OpenVMS supports fiber-optic media.
Long interconnect distances	OpenVMS supports multimode fiber-optic media at 500 meters per link and single-mode fiber-optic media (for interswitch links [ISLs]) for distances up to 100 kilometers per link.
Multiple protocols	OpenVMS supports SCSI–3. Possible future support for IP.
Numerous topologies	OpenVMS supports switched FC (highly scalable, multiple concurrent communications) and multiple switches (fabric). For more information about hardware and storage supported by VSI, please refer to our hardware support web page: https://vmssoftware.com/products/supported-hardware/.

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)

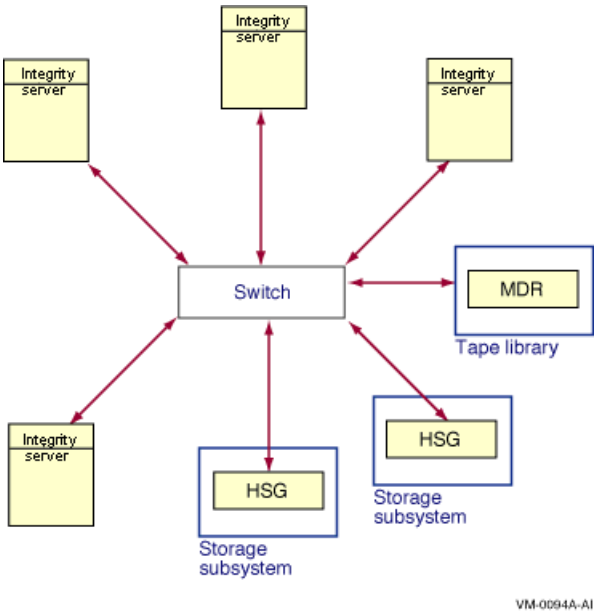


Figure 7.2 shows a physical view of a Fibre Channel switched topology. The configuration in Figure 7.2 is simplified for clarity. Typical configurations will have multiple Fibre Channel interconnects for high availability, as shown in Section 7.3.4.

Figure 7.2. Switched Topology (Physical View)

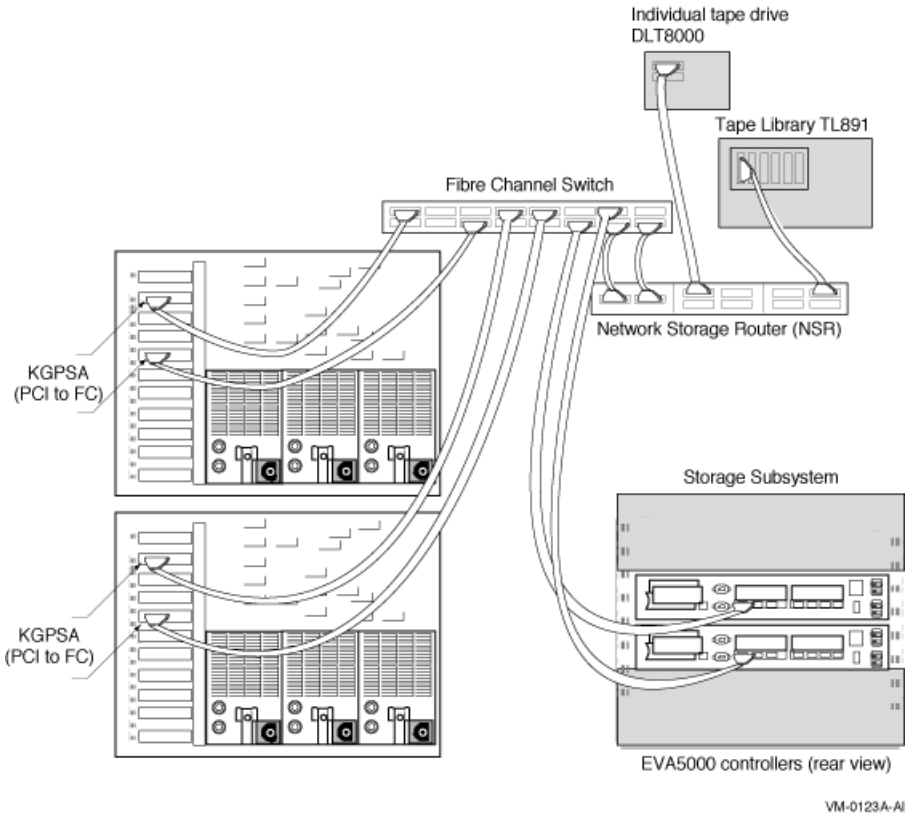
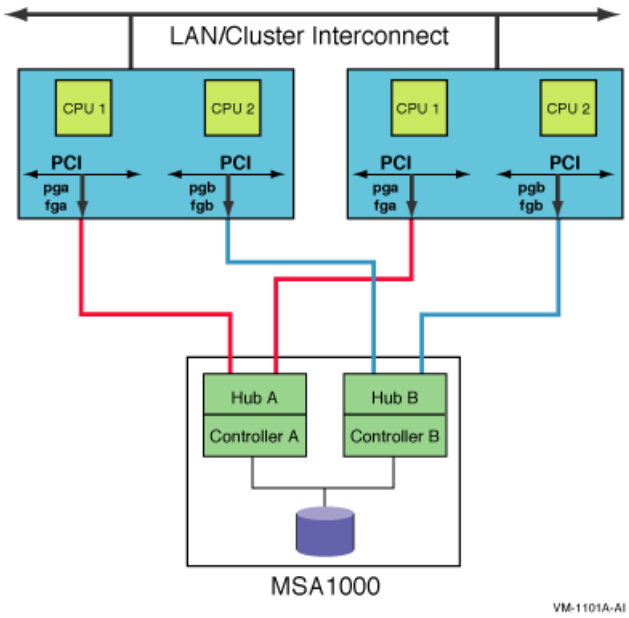


Figure 7.3 shows an arbitrated loop topology. Two hosts are connected to a dual-ported StorageWorks MSA 1000 storage system. OpenVMS supports an arbitrated loop topology only on this storage system.

Figure 7.3. Arbitrated Loop Topology Using MSA 1000



7.2. Fibre Channel Configuration Support

OpenVMS supports the Fibre Channel devices listed in the Software Product Description. For Fibre Channel fabric components supported by OpenVMS, refer to the latest version of *HP StorageWorks SAN Design Reference Guide*.

Note that Fibre Channel hardware names typically use the letter G to designate hardware that is specific to Fibre Channel. Fibre Channel configurations with other Fibre Channel equipment are not supported. To determine the required minimum versions of the operating system and firmware, see the release notes.

VSI recommends that all OpenVMS Fibre Channel configurations use the latest update kit for the OpenVMS version they are running.

The root name of these kits is FIBRE_SCSI, a change from the earlier naming convention of FIBRECHAN.

OpenVMS supports the Fibre Channel SAN configurations described in the latest *HP StorageWorks SAN Design Reference Guide* and the Data Replication Manager (DRM) user documentation. This includes support for:

- Multiswitch FC fabrics.
- Support for up to 500 meters of multimode fiber, and support for up to 100-kilometer interswitch links (ISLs) using single-mode fiber. In addition, DRM configurations provide longer-distance ISLs through the use of the Open Systems Gateway and Wave Division Multiplexors.
- Sharing of the fabric and the HSG storage with non-OpenVMS systems.

Within the configurations described in the StorageWorks documentation, OpenVMS provides the following capabilities and restrictions:

- All OpenVMS disk functions are supported: system disk, dump disks, shadow set member, quorum disk, and MSCP served disk. Each virtual disk must be assigned an identifier that is unique clusterwide.
- OpenVMS provides support for the number of hosts, switches, and storage controllers specified in the StorageWorks documentation. In general, the number of hosts and storage controllers is limited only by the number of available fabric connections.
- The number of Fibre Channel host bus adapters per platform depends on the platform type. Currently, the largest platforms support up to 26 adapters (independent of the number of OpenVMS instances running on the platform).
- OpenVMS requires that the HSG operate in SCSI-3 mode, and if the HSG is in a dual redundant configuration, then the HSG must be in multibus failover mode. The HSG can only be shared with other systems that operate in these modes.
- The OpenVMS Fibre Channel host bus adapter must be connected directly to the FC switch. The host bus adapter is not supported on a Fibre Channel loop, nor in a point-to-point connection to another Fibre Channel end node.
- Neither the KGPSA-BC nor the KGPSA-CA can be connected to the same PCI bus as the S3 Trio 64V+ Video Card (PB2GA-JC/JD). On the AlphaServer 800, the integral S3 Trio must be disabled when the KGPSA is installed.

- Hosts on the fabric can be configured as a single cluster or as multiple clusters and/or nonclustered nodes. It is critical to ensure that each cluster and each nonclustered system has exclusive access to its storage devices. HSG/HSV selective storage presentation and FC switch zoning or both can be used to ensure that each HSG/HSV storage device is accessible to only one cluster or one nonclustered system.

- The HSG supports a limited number of connections. A connection is a nonvolatile record of a particular host bus adapter communicating with a particular port on the HSG. (Refer to the HSG CLI command SHOW CONNECTIONS.) The HSG ACS V8.6 supports a maximum of 96 connections, whereas HSG ACS V8.5 allows a maximum of 64 connections, and HSG ACS V8.4 allows a maximum of 32 connections. The connection limit is the same for both single and dual redundant controllers.

  If your FC fabric is large, and the number of active connections exceeds the HSG limit, then you must reconfigure the fabric, or use FC switch zoning to "hide" some of the adapters from some of the HSG ports, in order to reduce the number of connections.

  The HSG does not delete connection information from the connection table when a host bus adapter is disconnected. Instead, the user must prevent the table from becoming full by explicitly deleting the connection information using a CLI command.

This configuration support is in effect as of the revision date of this document. OpenVMS plans to increase these limits in future releases.

In addition to the configurations already described, OpenVMS also supports the SANworks Data Replication Manager. This is a remote data vaulting solution that enables the use of Fibre Channel over longer distances.

7.2.1. Mixed-Version and Mixed-Architecture Cluster Support

Shared Fibre Channel OpenVMS Cluster storage is supported in both mixed-version and mixed-architecture OpenVMS Cluster systems. Mixed-version support is described in the Software Product Description. Mixed-architecture support means a combination of OpenVMS Alpha systems with OpenVMS Integrity server systems. In an OpenVMS mixed-architecture cluster, each architecture requires a minimum of one system disk.

The following configuration requirements must be observed:

- All hosts configured for shared access to the same storage devices must be in the same OpenVMS Cluster.

- All hosts in the cluster require a common cluster communication interconnect, such as a LAN, IP network, or MEMORY CHANNEL.

- All hosts with a direct connection to the FC must be running one of the supported OpenVMS Integrity servers or OpenVMS Alpha.

- All hosts must have the remedial kits for mixed-version clusters installed, as documented in the Release Notes.

- If you use DECevent for error tracing, Version 2.9 or later is required. Earlier versions of DECevent do not support Fibre Channel.

7.3. Example Configurations

This section presents example Fibre Channel configurations.

Note

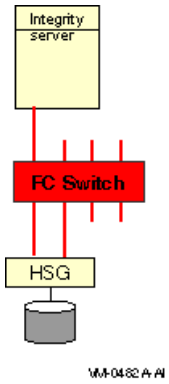
These configurations are valid for HSG storage controllers and for HSV storage controllers, except for Section 7.3.1 and Section 7.3.2, which apply only to HSG storage controllers.

The configurations build on each other, starting with the smallest valid configuration and adding redundant components for increasing levels of availability, performance, and scalability.

7.3.1. Single Host with Dual-Ported Storage

Figure 7.4 shows a single system using Fibre Channel as a storage interconnect.

Figure 7.4. Single Host With One Dual-Ported Storage Controller



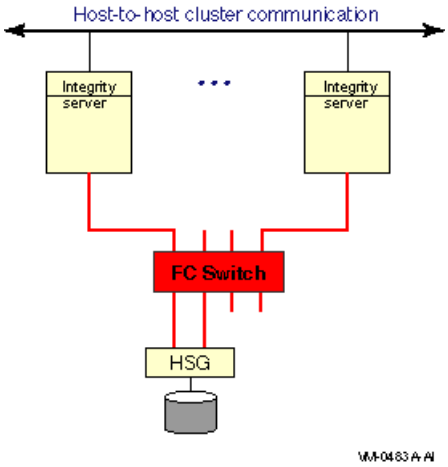
Note the following about this configuration:

- Dual ports of the HSG or HSV storage controller increase the availability and performance of the storage subsystem.
- Extra ports on the switch enable system growth.
- To maximize performance, logical units can be spread over the two HSG or HSV ports.
- The switch and the HSG or HSV are single points of failure. To provide higher availability, Volume Shadowing for OpenVMS can be used to replicate the data to another Fibre Channel switch and HSG or HSV controller.

7.3.2. Multiple Hosts With One Dual-Ported Storage Controller

Figure 7.5 shows multiple hosts connected to a dual-ported storage subsystem.

Figure 7.5. Multiple Hosts With One Dual-Ported Storage Controller



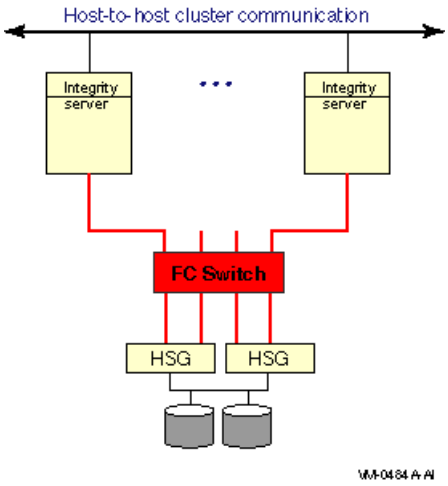
Note the following about this configuration:

- Multiple hosts increase availability of the entire system.
- Extra ports on the switch enable system growth.
- The switch and the HSG or HSV are single points of failure. To provide higher availability, Volume Shadowing for OpenVMS can be used to replicate the data to another Fibre Channel switch and HSG or HSV controller.

7.3.3. Multiple Hosts With Storage Controller Redundancy

Figure 7.6 shows multiple hosts connected to two dual-ported storage controllers.

Figure 7.6. Multiple Hosts With Storage Controller Redundancy



This configuration offers the following advantages:

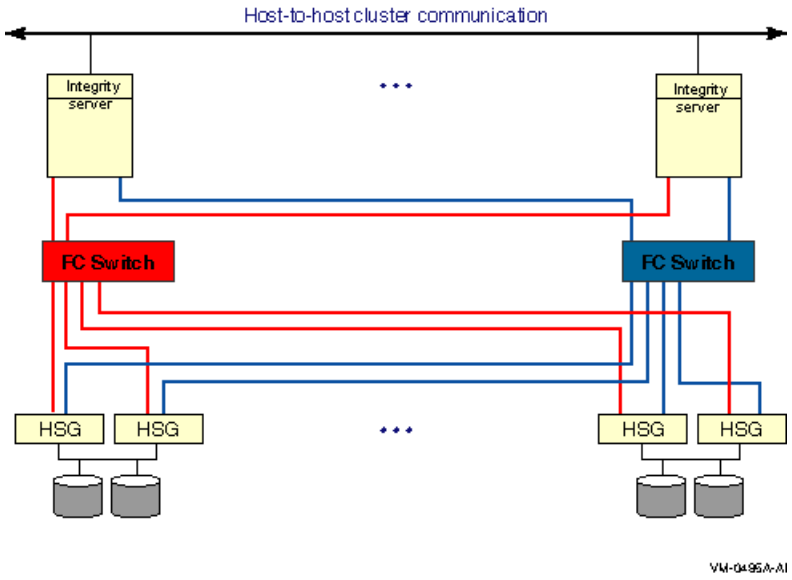
- Logical units can be spread over the four HSG or HSV ports, offering higher performance.
- HSGs or HSVs can be configured in multibus failover mode, even though there is just one Fibre Channel “bus.”

- The switch is still a single point of failure. To provide higher availability, Volume Shadowing for OpenVMS can be used to replicate the data to another Fibre Channel switch and HSG or HSV controller.

7.3.4. Multiple Hosts With Multiple Independent Switches

Figure 7.7 shows multiple hosts connected to two switches, each of which is connected to a pair of dual-ported storage controllers.

Figure 7.7. Multiple Hosts With Multiple Independent Switches



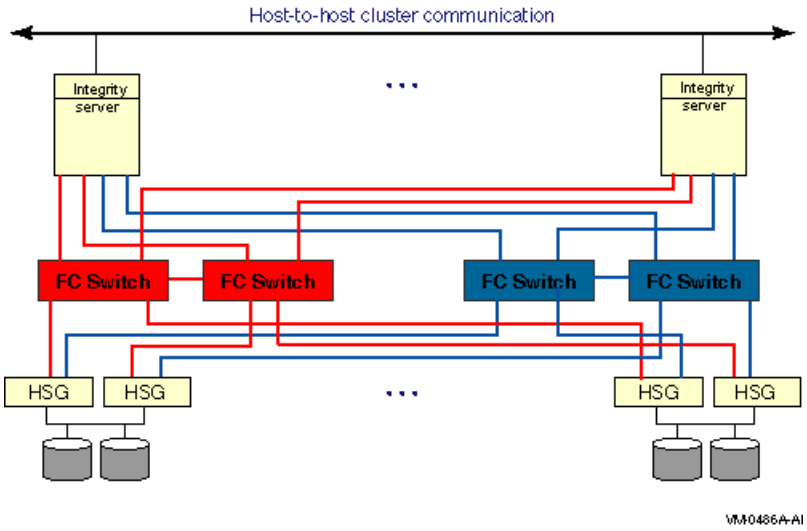
This two-switch configuration offers the advantages of the previous configurations plus the following:

- Higher level of availability afforded by two switches. There is no single point of failure.
- Better performance because of the additional host bus adapter.
- Each host has multiple independent paths to a storage subsystem. The two switches are not connected to each other to ensure that the paths are completely independent.

7.3.5. Multiple Hosts With Dual Fabrics

Figure 7.8 shows multiple hosts connected to two fabrics; each fabric consists of two switches.

Figure 7.8. Multiple Hosts With Dual Fabrics



This dual-fabric configuration offers the advantages of the previous configurations plus the following advantages:

- More ports are available per fabric for connecting to additional hosts and storage subsystems.
- Each host has four host bus adapters, one for each switch. Only two adapters are required, one per fabric. The additional adapters increase availability and performance.

7.3.6. Multiple Hosts With Larger Fabrics

The configurations shown in this section offer even higher levels of performance and scalability.

Figure 7.9 shows multiple hosts connected to two fabrics. Each fabric has four switches.

Figure 7.9. Multiple Hosts With Larger Dual Fabrics

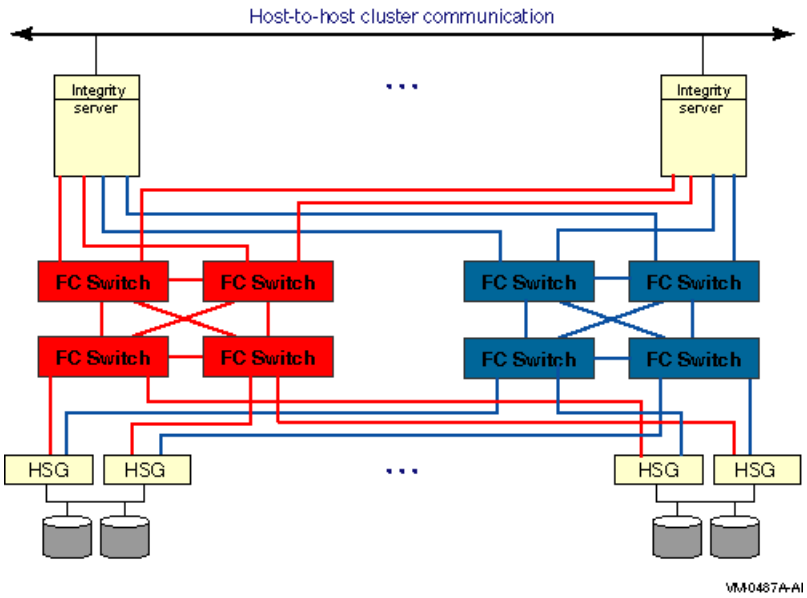
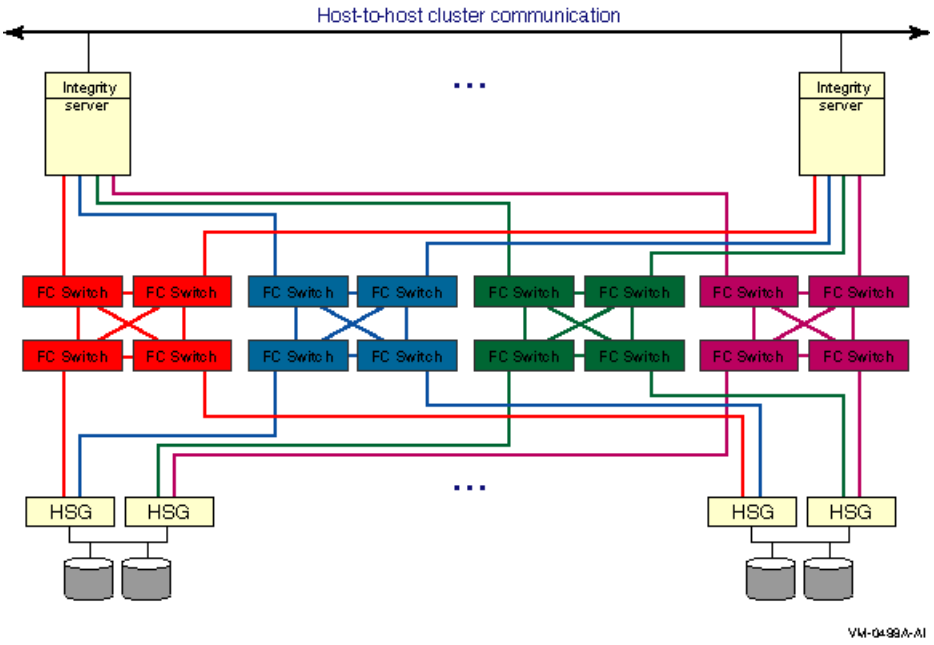


Figure 7.10 shows multiple hosts connected to four fabrics. Each fabric has four switches.

Figure 7.10. Multiple Hosts With Four Fabrics



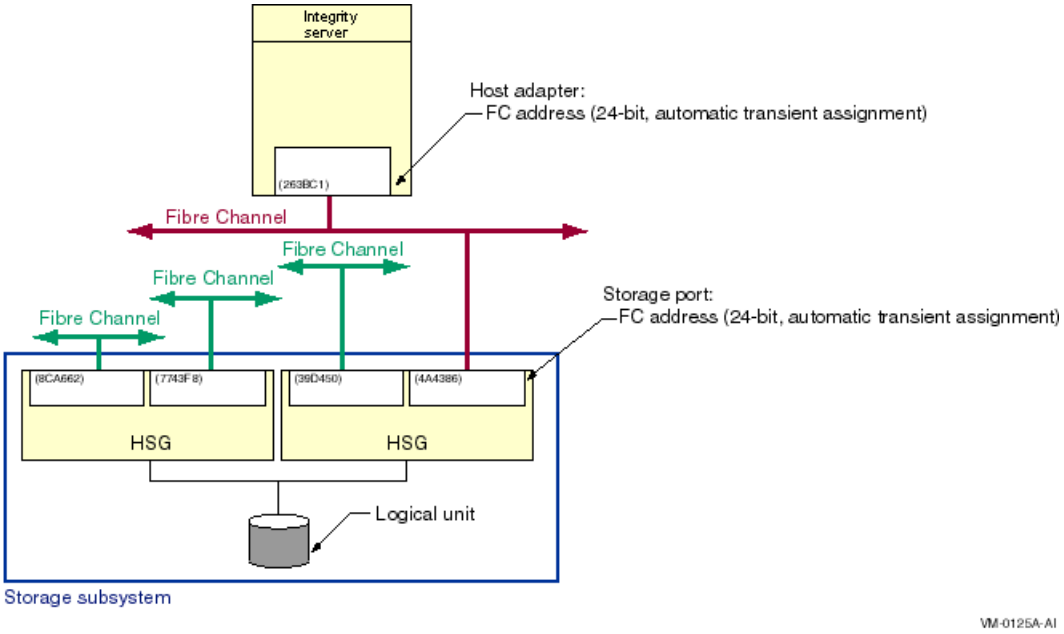
7.4. Fibre Channel Addresses, WWIDs, and Device Names

Fibre Channel devices for disk and tape storage 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 addresses are displayed, for example, by the Alpha console and by the HSG or HSV 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 7.11.

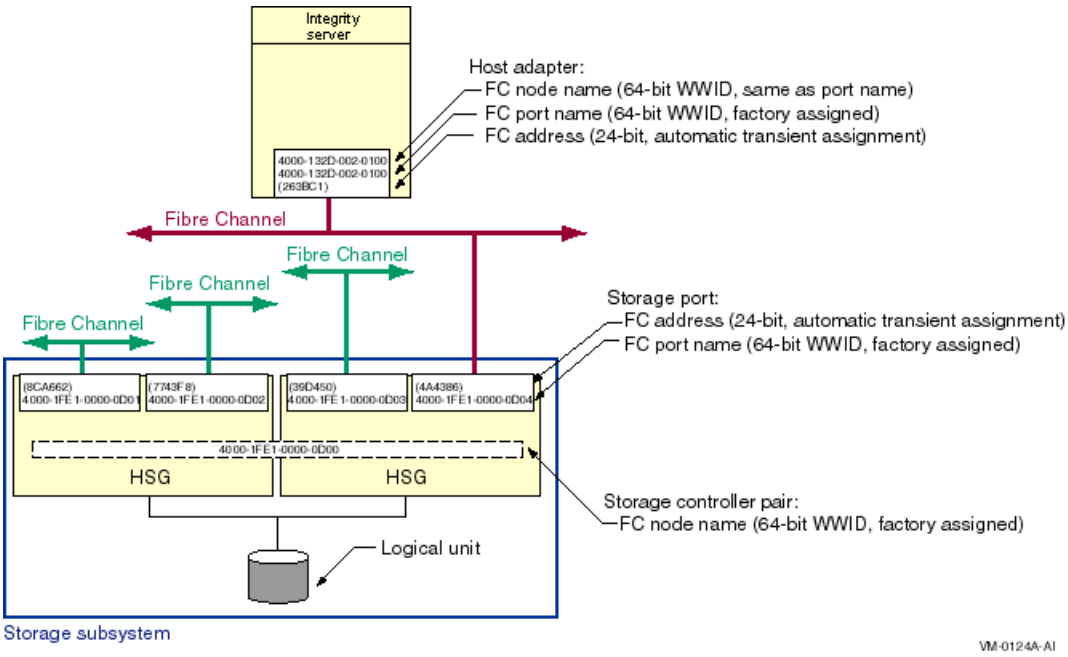
Figure 7.11. Fibre Channel Host and Port Addresses



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 7.12 shows Fibre Channel components with their factory-assigned WWIDs and their Fibre Channel addresses.

Figure 7.12. Fibre Channel Host and Port WWIDs and Addresses



Note the following about this figure:

- Host adapter's port name and node name are each a 64-bit, factory-assigned WWID.

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

You can display the FC node name and FC port name for a Fibre Channel host bus adapter with the SHOW DEVICE/FULL command. For example:

```
$ SHOW DEVICE/FULL FGA0

Device FGA0:, device type KGPSA Fibre Channel, is online, shareable, error
    logging is enabled.

    Error count                    0    Operations completed                0
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot              S:RWPL,O:RWPL,G,W
    Reference count                0    Default buffer size                 0
    FC Port Name 1000-0000-C923-0E48    FC Node Name        2000-0000-C923-0E48
```

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 sections apply to both disk devices and tape devices, except for Section 7.4.2.3, which is specific to disk devices. Tape device names are described in Section 7.5.

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:`

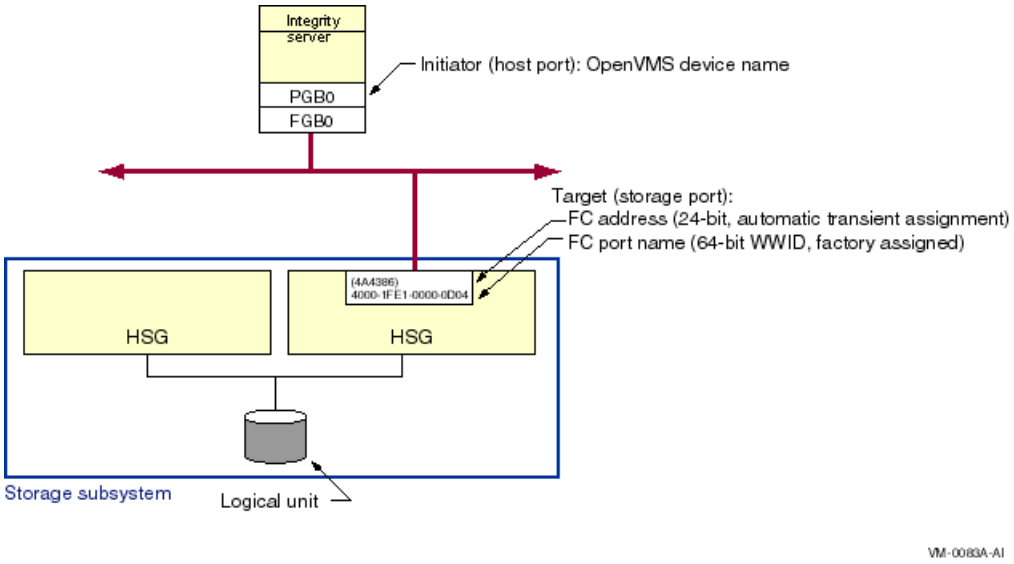
- `FG` represents Fibre Channel.
- *x* represents the unit letter, from A to Z.
- `0` is a constant.

The naming design places a limit of 26 adapters per system. 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 7.13.

Figure 7.13. Fibre Channel Initiator and Target Names



7.4.2.2. Fibre Channel Path Names

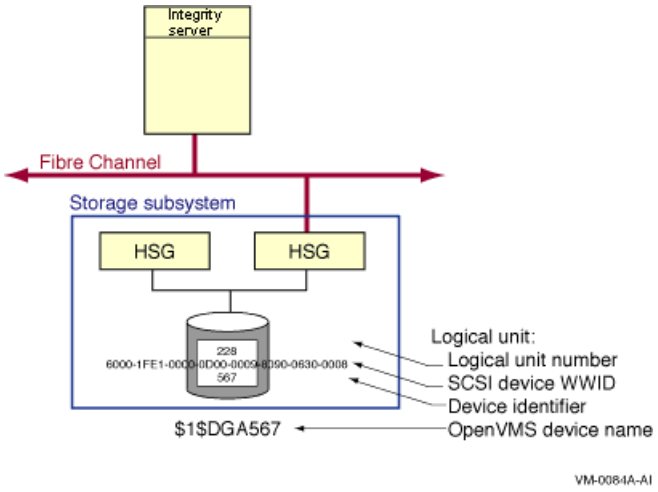
With the introduction of multipath SCSI support, as described in Chapter 6, it is necessary to identify specific paths from the host to the storage subsystem. This is done by concatenating the SCSI pseudodevice name, a decimal point (.), and the WWID of the storage subsystem port that is being accessed. For example, the Fibre Channel path shown in Figure 7.13 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 Disk Device Identification

The four identifiers associated with each FC disk device are shown in Figure 7.14.

Figure 7.14. Fibre Channel Disk 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 HSG or HSV console by the system manager. It can also be displayed by the OpenVMS SDA utility.

Each Fibre Channel disk device also has a WWID to provide permanent, unique identification of the device. The HSG or HSV device WWID is 128 bits. Half of this identifier is the WWID of the HSG or HSV that created the logical storage device, and the other half is specific to the logical device. The device WWID is displayed by the SHOW DEVICE/FULL command, the HSG or HSV console and the AlphaServer console.

The third identifier associated with the storage device is a user-assigned device identifier. A device identifier has the following attributes:

- User assigned at the HSG or HSV console.
- User must ensure it is cluster unique.
- Moves with the device.
- Can be any decimal number from 0 to 32767.

The device identifier has a value of 567 in Figure 7.14. This value is used by OpenVMS to form the device name so it must be unique throughout the cluster. (It may be convenient to set the device identifier to the same value as the logical unit number (LUN). This is permitted as long as the device identifier is unique throughout the cluster).

A Fibre Channel storage disk device name is formed by the operating system from the constant 1DGA and a device identifier, *nnnnn*. Note that Fibre Channel disk device names use an allocation class value of 1 whereas Fibre Channel tape device names use an allocation class value of 2, as described in Section 7.5.2.1. The only variable part of the name is its device identifier, which you assign at the HSG or HSV console. Figure 7.14 shows a storage device that is known to the host as 1DGA567.

Note

A device identifier of 0 is not supported on the HSV.

The following example shows the output of the SHOW DEVICE/FULL display for this device:

```
$ SHOW DEVICE/FULL $1$DGA567:

Disk $1$DGA567: (WILD8), device type HSG80, 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                    14    Operations completed            6896599
    Owner process                  ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                 9    Default buffer size                512
    WWID   01000010:6000-1FE1-0000-0D00-0009-8090-0630-0008
    Total blocks             17769177    Sectors per track                  169
    Total cylinders              5258    Tracks per cylinder                 20
    Host name                 "WILD8"    Host type, avail Compaq AlphaServer
                                         GS160 6/731, yes
    Alternate host name      "H2OFRD"    Alt. type, avail AlphaServer 1200 5/533
                                         4MB, yes
    Allocation class                1

    Volume label       "S5SH_V72_SSS"    Relative volume number                0
    Cluster size                   18    Transaction count                     9
    Free blocks              12811860    Maximum files allowed            467609
    Extend quantity                 5    Mount count                           6
    Mount status               System    Cache name          "_$1$DGA8:XQPCACHE"
```

```
  Extent cache size              64    Maximum blocks in extent cache  1281186
  File ID cache size             64    Blocks currently in extent cache1260738
  Quota cache size                0    Maximum buffers in FCP cache       1594
  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 XQP caching enabled, write-through XFC caching enabled.
Volume is also mounted on H2OFRD, FIBRE3, NORLMN, BOOLA, FLAM10.

I/O paths to device              5
Path PGA0.5000-1FE1-0000-0D02  (WILD8), primary path.
  Error count                    0    Operations completed            14498
Path PGA0.5000-1FE1-0000-0D03  (WILD8), current path.
  Error count                   14    Operations completed          6532610
Path PGA0.5000-1FE1-0000-0D01  (WILD8).
  Error count                    0    Operations completed            14481
Path PGA0.5000-1FE1-0000-0D04  (WILD8).
  Error count                    0    Operations completed            14481
Path MSCP (H2OFRD).
  Error count                    0    Operations completed           320530
```

7.5. Fibre Channel Tape Support

This section describes the configuration requirements and user commands necessary to utilize the Fibre Channel tape functionality. Fibre Channel tape functionality refers to the support of SCSI tapes and SCSI tape libraries in an OpenVMS Cluster system with shared Fibre Channel storage. The SCSI tapes and libraries are connected to the Fibre Channel by a Fibre-to-SCSI bridge. Currently, two bridges are available: the Modular Data Router (MDR) and the Network Storage Router (NSR).

7.5.1. Minimum Hardware Configuration

Following is the minimum Fibre Channel tape hardware configuration:

- Alpha or Integrity system with supported FC HBA
- Fibre-to-SCSI bridge:
  - Network Storage Router (NSR)

    The NSR must also be connected to a switch and not directly to an Alpha system.

    VSI recommends that the NSR be set to indexed mode.

    The indexed map should be populated in Target/Bus priority order to ensure that the controller LUN is mapped to LUN 0. Also, be careful to avoid conflicting IDs, as documented in the *HP StorageWorks Network Storage Router M2402* user guide.
  - Modular Data Router (MDR), minimum firmware revision 1170

    The MDR must be connected to a switch and not directly to an Alpha system. Furthermore, the MDR must be in SCSI Command Controller (SCC) mode, which is normally the default. If the MDR is not in SCC mode, use the command `SetSCCmode On` at the MDR console.

    Tape devices and tape library robots must not be set to SCSI target ID 7, since that ID is reserved for use by the MDR.
- Fibre Channel switch

- Tape library, for example:
  - MSL5000 series
  - ESL9000 series
  - TL891
  - TL895
- Individual tapes, for example:
  - SDLT 160/320
  - SDLT 110/220
  - HP Ultrium 460
  - HP Ultrium 448c
  - DLT8000
  - TZ89
- SAS tape blade - HP StorageWorks Ultrium 488c and Ultrium 920c for the C-Class Integrity BladeSystem.

Note

Tapes are not supported in an HSG*xx* storage subsystem nor behind a Fibre Channel Tape Controller II (FCTC-II).

A tape library robot is an example of a **medium changer device**, the term that is used throughout this section.

7.5.2. Overview of Fibre Channel Tape Device Naming

This section provides detailed background information about Fibre Channel Tape device naming.

Tape and medium changer devices are automatically named and configured using the SYSMAN IO FIND and IO AUTOCONFIGURE commands described in Section 7.5.3. System managers who configure tapes on Fibre Channel should refer directly to this section for the tape configuration procedure.

7.5.2.1. Tape and Medium Changer Device Names

Fibre Channel tapes and medium changers are named using a scheme similar to Fibre Channel disk naming.

On parallel SCSI, the device name of a directly attached tape implies the physical location of the device; for example, MKB301 resides on bus B, SCSI target ID 3, and LUN 1. Such a naming scheme does not scale well for Fibre Channel configurations, in which the number of targets or nodes can be very large.

Fibre Channel tape names are in the form \$2\$MGA*n*. The letter for the controller is always A, and the prefix is \$2\$. The device mnemonic is MG for tapes and GG for medium changers. The device unit *n* is automatically generated by OpenVMS.

The name creation algorithm chooses the first free unit number, starting with zero. The first tape discovered on the Fibre Channel is named 2MGA0, the next is named 2MGA1, and so forth. Similarly, the first medium changer detected on the Fibre Channel is named 2GGA0. The naming of tapes and medium changers on parallel SCSI buses remains the same.

Note the use of allocation class 2. Allocation class 1 is already used by devices whose name is keyed by a user-defined identifier (UDID), as with HSG Fibre Channel disks (1DGA*nnnn*) and HSG console command LUNs (1GGA*nnnn*).

An allocation class of 2 is used by devices whose names are obtained from the file, SYS $DEVICES.DAT. The names are based on a worldwide identifier (WWID) key, as described in the following sections. Also note that, while GG is the same mnemonic used for both medium changers and HSG Command Console LUNs (CCLs), medium changers always have an allocation class of 2 and HSG CCLs an allocation class of 1.

Tape and medium changer names are automatically kept consistent within a single OpenVMS Cluster system. Once a tape device is named by any node in the cluster, all other nodes in the cluster automatically choose the same name for that device, even if this overrides the first free unit number algorithm. The chosen device name remains the same through all subsequent reboot operations in the cluster.

If multiple nonclustered Integrity server systems exist on a SAN and need to access the same tape device on the Fibre Channel, then the upper-level application must provide consistent naming and synchronized access.

7.5.2.2. Use of Worldwide Identifiers (WWIDs)

For each Fibre Channel tape device name, OpenVMS must uniquely identify the physical device that is associated with that name.

In parallel SCSI, directly attached devices are uniquely identified by their physical path (port/target/LUN). Fibre Channel disks are uniquely identified by user-defined identifiers (UDIDs). These strategies are either unscalable or unavailable for Fibre Channel tapes and medium changers.

Therefore, the identifier for a given Fibre Channel tape or medium changer device is its worldwide identifier (WWID). The WWID resides in the device firmware and is required to be unique by the Fibre Channel standards.

WWIDs can take several forms, for example:

- IEEE registered WWID (64-bit binary)
- Vendor ID plus product ID plus serial number (ASCII)

The overall WWID consists of the WWID data prefixed by a binary WWID header, which is a longword describing the length and type of WWID data.

In general, if a device reports an IEEE WWID, OpenVMS chooses this as the unique identifying WWID for the device. If the device does not report such a WWID, then the ASCII WWID is used. If the device reports neither an IEEE WWID nor serial number information, then OpenVMS does not configure the device. During the device discovery process, OpenVMS rejects the device with the following message:

```
%SYSMAN-E-NOWWID, error for device Product-ID, no valid WWID found.
```

The WWID structures can be a mix of binary and ASCII data. These formats are displayable and are intended to be consistent with those defined by the console WWIDMGR utility.

Note that if the data following the WWID header is pure ASCII data, it must be enclosed in double quotation marks.

The displayable format of a 64-bit IEEE WWID consists of an 8-digit hexadecimal number in ASCII (the WWID header), followed by a colon (:) and then the IEEE WWID data. For example:

```
0C000008:0800-4606-8010-CD3C
```

The displayable format of an ASCII WWID consists of an 8-digit WWID header, followed by a colon (:) and then the concatenation of the 8-byte vendor ID plus the 16-byte product ID plus the serial number. For example:

```
04100022:"COMPAQ  DLT8000         JF71209240"
```

Note

Occasionally, an ASCII WWID may contain nonprintable characters in the serial number. In a displayable format, such a character is represented by*nn*, where *nn* is the 2-digit ASCII hexadecimal value of the character. For example, a null is represented by\00.

7.5.2.3. File-Based Device Naming

Fibre Channel tape and medium changer devices are configured according to information found in the SYS$SYSTEM:SYS$DEVICES.DAT file. This is an ASCII file consisting of two consecutive records per device, where the two records are in the following form:

```
[Device $2$devnam]
WWID = displayable_identifier
```

During autoconfiguration, the Fibre Channel is probed and the WWIDs are fetched for all devices. If the fetched WWID matches an entry in the memory-resident copy of the SYS$DEVICES.DAT file, then the device is configured using the device name that has been paired with that WWID.

Note

The SYS$DEVICES.DAT file is also used for port allocation class (PAC) information. Fibre Channel tape-naming is a second use of this same file, even though PACs and Fibre Channel tapes are not related, other than their common need to access file-based device information at boot time.

By default, the SYS$DEVICES.DAT file is created in the cluster common directory, SYS$COMMON:[SYSEXE].

As an example, the following portion of SYS$DEVICES.DAT causes the eventual configuration of devices named 2MGA300 and 2MGA23:

```
!
[Device $2$MGA300]
WWID = 04100022:"COMPAQ  DLT8000         JF71209240"
!
[Device $2$mga23]
WWID = 04100022:"DEC     TZ89     (C) DECJL01164302"
```

Although the file is typically read and written only by OpenVMS utilities, in rare instances you may need to edit the file. You can change only the unit number of the device, as described in Section 7.5.5. The internal syntax rules governing the file are summarized as follows:

- Comment lines (beginning with !) and blank lines are permitted.

- Any white space (or none) can separate `[Device` from the device name represented by `$2$xxx`].
- Failure to supply the 2 prefix will result in a console warning.

Similarly, on the line containing `WWID =`, any white space (or none) can appear on either side of the equals sign. All lines must be left-justified, and all lines must be less than 512 characters.

The parsing of this file is not case sensitive, with one important exception: all characters enclosed within double quotation marks are taken literally, so that characters such as spaces and lowercase letters are significant. In the case of ASCII data enclosed by double quotation marks, there must be no space between the colon and the double quotation mark.

Also, if more than one `WWID =` line follows a single `[Device devnam]` line, the last `WWID =` value takes precedence. Normally, however, there is exactly one `WWID =` line per `[Device devnam]` line.

Similarly, if two or more `[Device devnam]` lines specify the same device name but different WWIDs, only the last device name and WWID specified in the file is used.

This file is read at boot time, and it is also read from and written to by the SYSMAN IO FIND_WWID command. If there are additional system-specific copies of the SYS$DEVICES.DAT file, their tape naming records become automatically compatible as a result of running SYSMAN IO FIND_WWID on each system. The SYSMAN IO FIND_WWID command is described in more detail in the following section. The SYS$DEVICES.DAT file may also be modified by the SYSMAN IO CREATE_WWID and REPLACE_WWID commands which are described below.

7.5.3. Management Support for Fibre Channel Tape Devices

The following System Management utility (SYSMAN) commands are provided for managing Fibre Channel tape devices:

- IO FIND_WWID

  Probes all ports on the Fibre Channel and detects all previously undiscovered tapes and medium changers, and assigns a name to each. Displays a list of the devices and their assigned device names, and automatically records this information in the SYS$SYSTEM:SYS$DEVICES.DAT file. Also updates relevant local and clusterwide memory structures. It should be executed clusterwide.

  Use this command prior to running the SYSMAN command IO AUTOCONFIGURE.

  Requires the CMKRNL privilege.

- IO LIST_WWID

  Lists all tape device WWIDs that are not yet configured on Fibre Channel. Use this command prior to running the SYSMAN command IO CREATE_WWID.

  Requires the CMKRNL privilege.

- IO CREATE_WWID

  Enables the user to assign a specific (and previously unused) device name to a specific (and previously unused) WWID from the SYSMAN IO LIST_WWID display. It should be executed clusterwide. The command should then be followed by a clusterwide SYSMAN IO AUTO command to actually configure the device.

This command offers an alternative to the SYSMAN IO FIND_WWID command, which chooses system-generated device names for the discovered WWIDs. The IO CREATE command should not be used after the IO FIND command as a means of redefining WWID correlations. The device and WWID strings specified in IO CREATE_WWID should not be in use elsewhere in the cluster.

Requires the CMKRNL privilege.

- IO REPLACE_WWID

  Updates appropriate file and memory data structures in case one tape drive must be physically replaced by another tape drive at the same FC LUN location.

  Requires the CMKRNL privilege.

The following DCL support for Fibre Channel tape devices is available:

- The SHOW DEVICE/FULL command displays the WWID for Fibre Channel tape devices.
- The F$GETDVI lexical function supports the keyword `WWID`, which returns the Fibre Channel tape device's WWID.

7.5.4. Configuring a Fibre Channel Tape Device

This section lists the steps required to configure a new tape or medium changer on the Fibre Channel.

7.5.4.1. Basic Configuration Steps: Summary

The basic steps for configuring new Fibre Channel tape devices in a cluster are as follows:

1. Power on the new tape device or devices.
2. If you are using the MDR, power cycle the MDR to update MDR mapping information.

   If you are using the NSR, use Visual Manager to update the mapping information as follows:

   - Click on the Mapping submenu. Username is root, password is password.
   - Ensure the Select Map box indicates Indexed mode.
   - Click on the Edit/View box that is next to the Select Map box.
   - This brings up an empty indexed map. Under the Priority option, select Target/Bus Priority and then click on Fill Map. Note that the Target/Bus priority ensures that the controller LUN, also known as the Active Fabric (AF) LUN, is mapped to LUN 0.
   - The new map appears. Close that window, which then returns you to the Mapping submenu.
   - If the NSR has additional FC ports besides FC port 0, click on FC Port 1 and repeat the mapping process for FC Port 1, and any other FC ports.
   - Click on Reboot to make the updates to the maps take effect.

     Further details on the Visual Manager are documented in the *HP StorageWorks Network Storage Router M2402* user guide.
3. Run SYSMAN to assign device names and configure the devices:

```
$ MC SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER        ! Execute on all nodes
SYSMAN> IO FIND_WWID                   ! Assign names
SYSMAN> IO AUTOCONFIGURE/LOG           ! Configure devices
SYSMAN> EXIT
```

You need to perform these steps only once for the initial configuration. After any subsequent system reboot, the devices will appear automatically.

7.5.4.2. Basic Configuration Steps: Details

Prior to configuring a tape device on Fibre Channel, the worldwide identifier (WWID) of the device must be detected and stored, along with a device name, in the text file SYS$SYSTEM:SYS $DEVICES.DAT. This is usually accomplished by using the SYSMAN command IO FIND_WWID. However, some users prefer to choose their own devices name for the tape devices, rather than using the system-generated names assigned by FIND_WWID. In that case, the user will execute the IO CREATE_WWID command instead of IO FIND_WWID. IO CREATE_WWID will be described in the next section, while this current section documents the use of IO FIND_WWID.

The IO FIND_WWID command probes all ports on the Fibre Channel and locates all tape and medium changer devices connected to an MDR or NSR. For tapes and medium changers that have not been detected by a previous IO FIND_WWID command, IO FIND_WWID assigns a device name, retrieves the WWID of the device, stores the device name and WWID data in the SYS$SYSTEM:SYS $DEVICES.DAT file, and updates memory structures.

Since the primary goal of IO FIND_WWID is to populate the SYS$DEVICES.DAT file, you need to invoke the IO FIND_WWID command only once for each device. IO FIND_WWID does not configure the 2MGA*nnnn*: device for use by an application.

Once the information is stored in the file, subsequent use of the IO AUTOCONFIGURE command reads a memory-resident copy of the file and configures the tape and medium changer devices automatically, loading or connecting the device drivers as needed. The SYS$DEVICES.DAT file is read into memory during each system reboot; this action initiates the automatic configuration of tapes and medium changers on the Fibre Channel.

Note that running the IO FIND_WWID command for the first time detects all existing tape and medium changer devices on the system. If you add additional Fibre Channel tape devices to the system at a later time, you must first powercycle the MDR to update internal mapping information, and then run the IO FIND_WWID command again to append the new device information to the SYS$DEVICES.DAT file. On an NSR, edit the indexed map to update mapping information.

In an OpenVMS Cluster environment, various data structures in memory must be updated on each system when a new Fibre Channel tape device is added. To accomplish this, VSI recommends that you run the SYSMAN IO FIND_WWID command on each Alpha node in the cluster. Alternatively, you can run IO FIND_WWID on one node, and then reboot the other nodes that share that same system disk, because the SYS$DEVICES.DAT file is read at boot time and causes memory structures to be correctly initialized.

In the case of multiple system disks in the cluster, ensure that all copies of the SYS$DEVICES.DAT file are kept consistent, preferably by running the IO FIND_WWID command on all nodes. Alternatively, you can run IO FIND_WWID to update just one SYS$DEVICES.DAT file, and then manually edit the remaining SYS$DEVICES.DAT files by cutting and pasting the appropriate device name and WWID records from the original file to the target files. If this second alternative is used, however, the remaining nodes must be rebooted in order for the memory-resident copy of SYS$DEVICES.DAT to be updated.

VSI recommends that you refrain from copying the entire original file to another system disk. The SYS $DEVICES.DAT file is also used to define port allocation classes (PACs), and PAC entries could be transferred inadvertently to the target system.

Following is a configuration example using a TL891 tape library on a single node.

First, the SYSMAN command IO FIND_WWID displays a list of all previously undiscovered tape devices and their proposed device names.

```
$ MCR SYSMAN IO FIND_WWID

%SYSMAN-I-OUTPUT, command execution on node SAMPLE
On port _SAMPLE$PGA0:, the following tape WWIDs and their proposed device
names have been found but not yet configured:

      [Device $2$GGA0]
      WWID=04100024:"DEC     TL800    (C) DEC3G9CCR82A017"

      [Device $2$MGA0]
      WWID=04100022:"DEC     TZ89     (C) DECCX939S2777"

      [Device $2$MGA1]
      WWID=04100022:"DEC     TZ89     (C) DECCX942S6295"
```

Note that the overall WWID consists of everything to the right of the equals sign. Each such WWID is unique; however, the header portion may not be unique, because the header reflects only the basic type and length of the WWID data.

The IO FIND_WWID command automatically records the information about the new tape devices in SYS$SYSTEM:SYS$DEVICES.DAT:

```
$ TYPE SYS$SYSTEM:SYS$DEVICES.DAT
!
! Updated 23-OCT-2000 14:17:41.85:  DEC TL800
!
[Device $2$GGA0]
WWID=04100024:"DEC     TL800    (C) DEC3G9CCR82A017"
!
!
! Updated 23-OCT-2000 14:17:41.93:  DEC TZ89
!
[Device $2$MGA0]
WWID=04100022:"DEC     TZ89     (C) DECCX939S2777"
!
!
! Updated 23-OCT-2000 14:17:42.01:  DEC TZ89
!
[Device $2$MGA1]
WWID=04100022:"DEC     TZ89     (C) DECCX942S6295"
!
```

Next, the SYSMAN command IO AUTOCONFIGURE configures the tape device.

```
$ MCR SYSMAN IO AUTOCONFIGURE/LOG

%SYSMAN-I-OUTPUT, command execution on node SAMPLE
%IOGEN-I-PREFIX, searching for ICBM with prefix SYS$
%IOGEN-I-PREFIX, searching for ICBM with prefix DECW$
%IOGEN-I-SCSIPOLL, scanning for devices through SCSI port PKA0
```

```
%IOGEN-I-SCSIPOLL, scanning for devices through SCSI port PKB0
%IOGEN-I-FIBREPOLL, scanning for devices through FIBRE port PGA0
%IOGEN-I-CONFIGURED, configured device GGA0
%IOGEN-I-CONFIGURED, configured device MGA0
%IOGEN-I-CONFIGURED, configured device MGA1
```

Finally, the SHOW DEVICE/FULL command displays the WWID of the tape device.

```
$ SHOW DEVICE/FULL $2$MG

Magtape $2$MGA0: (SAMPLE), device type TZ89, is online, file-oriented device,
   available to cluster, error logging is enabled, controller supports
   compaction (compaction  disabled), device supports fastskip.

   Error count                    0    Operations completed                0
   Owner process                 ""    Owner UIC                      [SYSTEM]
   Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
   Reference count                0    Default buffer size                2048
   WWID   04100022:"DEC     TZ89     (C) DECCX939S2777"
   Density                  default    Format                      Normal-11
   Allocation class               2

   Volume status:  no-unload on dismount, position lost, odd parity.

Magtape $2$MGA1: (SAMPLE), device type TZ89, is online, file-oriented device,
  available to cluster, error logging is enabled, controller supports
  compaction (compaction  disabled), device supports fastskip.

   Error count                    0    Operations completed                0
   Owner process                 ""    Owner UIC                      [SYSTEM]
   Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
   Reference count                0    Default buffer size                2048
   WWID   04100022:"DEC     TZ89     (C) DECCX942S6295"
   Density                  default    Format                      Normal-11
   Allocation class               2

   Volume status:  no-unload on dismount, position lost, odd parity.
```

The F$GETDVI lexical function also retrieves the displayable WWID:

```
$ write sys$output f$getdvi("$2$MGA0","WWID")
04100022:"DEC     TZ89     (C) DECCX939S2777"
```

Once the device is named and configured, you can use the device in the same way that you use parallel SCSI tapes with DCL commands such as INITIALIZE, MOUNT, BACKUP, and COPY. Refer to the installation guide for individual tape layered products for details on product-specific support of Fibre Channel tapes.

Note that while medium changers on Fibre Channel are autoconfigured, the medium changers on parallel SCSI continue to require the IO CONNECT command to load the device driver. It is impossible to manually connect a Fibre Channel medium changer by the SYSMAN IO CONNECT command because the device name does not imply the device's physical location, as it does in parallel SCSI.

7.5.4.3. Creating User-Specified Device Names

If you prefer to choose specific names for the tape devices instead of using the default names generated by IO FIND_WWID, you can use the IO CREATE_WWID command. For example:

```
SYSMAN> IO CREATE_WWID $2$MGA3/WWID=04100022:"DEC   TZ89   (C)
 DECCX939S2341"
```

The selected name must be of the form 2GGAn for medium changers and 2MGA*n* for tapes, where *n* is less than or equal to 32767. The name must not be in use elsewhere in the cluster. The WWID should be cut and pasted from the output of the IO LIST_WWID display. The IO CREATE_WWID command is intended only for naming new devices; it should not be used to rename existing devices. (Renaming existing devices is discussed in Section 7.5.5).

The following configuration example uses IO CREATE_WWID to create user-specified device names for two tapes and a medium changer within an ESL library. The commands are executed clusterwide on a 2-node cluster consisting of nodes SYSTM1 and SYSTM2. Each node has two Fibre Channel host bus adapters, PGA0 and PGB0, so multiple paths to the tape are configured.

First, the SYSMAN command IO LIST_WWID displays a list of all previously undiscovered tape devices.

```
Systm1> mcr sysman
SYSMAN> set env/clus
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username SYSTEM       will be used on nonlocal nodes

SYSMAN> io list_wwid

%SYSMAN-I-OUTPUT, command execution on node SYSTM2
On port _SYSTM2$PGA0:, the following tape WWIDs are not yet configured:

Target 8, LUN 1, HP       ESL9000 Series
WWID=0C000008:0050-8412-9DA1-0026

Target 8, LUN 2, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-84D1

Target 8, LUN 3, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-4E4E

On port _SYSTM2$PGB0:, the following tape WWIDs are not yet configured:

Target 6, LUN 1, HP       ESL9000 Series
WWID=0C000008:0050-8412-9DA1-0026

Target 6, LUN 2, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-84D1

Target 6, LUN 3, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-4E4E

%SYSMAN-I-OUTPUT, command execution on node SYSTM1
On port _SYSTM1$PGA0:, the following tape WWIDs are not yet configured:


Target 6, LUN 1, HP       ESL9000 Series
WWID=0C000008:0050-8412-9DA1-0026

Target 6, LUN 2, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-84D1

Target 6, LUN 3, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-4E4E
```

```
On port _SYSTM1$PGB0:, the following tape WWIDs are not yet configured:


Target 5, LUN 1, HP       ESL9000 Series
WWID=0C000008:0050-8412-9DA1-0026

Target 5, LUN 2, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-84D1

Target 5, LUN 3, COMPAQ   SDLT320
WWID=02000008:500E-09E0-0009-4E4E


%SYSMAN-I-NODERR, error returned from node SYSTM1
-SYSTEM-W-NOMORENODE, no more nodes
SYSMAN>
```

The previous NOMORENODE error is normal, because the command has completed on all existing nodes. Next, still in the same SYSMAN session, the user executes IO CREATE_WWID to choose device names 2GGA40, 2MGA40, 2MGA41.

```
SYSMAN> io create_wwid $2$GGA40/WWID=0C000008:0050-8412-9DA1-0026
%SYSMAN-I-NODERR, error returned from node SYSTM1
-SYSTEM-W-NOMORENODE, no more nodes
SYSMAN> io create_wwid $2$mga40/WWID=02000008:500E-09E0-0009-84D1
%SYSMAN-I-NODERR, error returned from node SYSTM1
-SYSTEM-W-NOMORENODE, no more nodes
SYSMAN> io create_wwid $2$mga41/WWID=02000008:500E-09E0-0009-4E4E
%SYSMAN-I-NODERR, error returned from node SYSTM1
-SYSTEM-W-NOMORENODE, no more nodes
SYSMAN>
```

The user now executes IO AUTOCONFIGURE to configure the devices. Note that both the PGA path and the PGB path are configured for each node.

```
SYSMAN> io auto/lo

%SYSMAN-I-OUTPUT, command execution on node SYSTM2
%IOGEN-I-PREFIX, searching for ICBM with prefix SYS$
%IOGEN-I-PREFIX, searching for ICBM with prefix DECW$
%IOGEN-I-FIBREPOLL, scanning for devices through FIBRE port PGA0
%IOGEN-I-CONFIGURED, configured device GGA40
%IOGEN-I-CONFIGURED, configured device MGA40
%IOGEN-I-CONFIGURED, configured device MGA41
%IOGEN-I-FIBREPOLL, scanning for devices through FIBRE port PGB0
%IOGEN-I-CONFIGURED, configured device GGA40
%IOGEN-I-CONFIGURED, configured device MGA40
%IOGEN-I-CONFIGURED, configured device MGA41

%SYSMAN-I-OUTPUT, command execution on node SYSTM1
%IOGEN-I-PREFIX, searching for ICBM with prefix SYS$
%IOGEN-I-PREFIX, searching for ICBM with prefix DECW$
%IOGEN-I-FIBREPOLL, scanning for devices through FIBRE port PGA0
%IOGEN-I-CONFIGURED, configured device GGA40
%IOGEN-I-CONFIGURED, configured device MGA40
%IOGEN-I-CONFIGURED, configured device MGA41
%IOGEN-I-FIBREPOLL, scanning for devices through FIBRE port PGB0
%IOGEN-I-CONFIGURED, configured device GGA40
```

```
%IOGEN-I-CONFIGURED, configured device MGA40
%IOGEN-I-CONFIGURED, configured device MGA41
%SYSMAN-I-NODERR, error returned from node SYSTM1
-SYSTEM-W-NOMORENODE, no more nodes
SYSMAN> exit
Systm1>
Systm1>SHOW DEVICE/FULL $2$GG

Device $2$GGA40:, device type Generic SCSI device, is online, shareable, device
    has multiple I/O paths.

    Error count                    0    Operations completed                0
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot    S:RWPL,O:RWPL,G:RWPL,W:RWPL
    Reference count                0    Default buffer size                 0
    WWID   0C000008:0050-8412-9DA1-0026

  I/O paths to device              2
  Path PGA0.1000-00E0-0242-86ED (SYSTM1), primary path, current path.
    Error count                    0    Operations completed                0
  Path PGB0.1000-00E0-0222-86ED (SYSTM1).
    Error count                    0    Operations completed                0

Systm1> sho dev/fu $2$MG

Magtape $2$MGA40: (SYSTM1), device type COMPAQ SDLT320, is online, file-oriented
    device, available to cluster, device has multiple I/O paths, error logging
    is enabled, device supports fastskip (per_io).

    Error count                    0    Operations completed                2
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                0    Default buffer size              2048
    WWID   02000008:500E-09E0-0009-84D1
    Density                  default    Format                        Normal-11
    Host name               "SYSTM1"    Host type, avail AlphaServer DS10
                                        466 MHz, yes
    Alternate host name     "SYSTM2"    Alt. type, avail AlphaServer DS10
                                        466 MHz,  no
    Allocation class               2

  Volume status:  no-unload on dismount, position lost, odd parity.

  I/O paths to device              2
  Path PGA0.1000-00E0-0242-86ED (SYSTM1), primary path, current path.
    Error count                    0    Operations completed                1
  Path PGB0.1000-00E0-0222-86ED (SYSTM1).
    Error count                    0    Operations completed                1


Magtape $2$MGA41: (SYSTM1), device type COMPAQ SDLT320, is online, file-oriented
    device, available to cluster, device has multiple I/O paths, error logging
    is enabled, device supports fastskip (per_io).

    Error count                    0    Operations completed                0
    Owner process                 ""    Owner UIC                      [SYSTEM]
    Owner process ID        00000000    Dev Prot            S:RWPL,O:RWPL,G:R,W
    Reference count                0    Default buffer size              2048
    WWID   02000008:500E-09E0-0009-4E4E
    Density                  default    Format                        Normal-11
    Host name               "SYSTM1"    Host type, avail AlphaServer DS10
                                        466 MHz, yes
```

```
    Alternate host name     "SYSTM2"    Alt. type, avail AlphaServer DS10
                                        466 MHz,  no
    Allocation class              2

  Volume status:  no-unload on dismount, position lost, odd parity.

  I/O paths to device             2
  Path PGA0.1000-00E0-0242-86ED (SYSTM1), primary path, current path.
    Error count                   0    Operations completed                0
  Path PGB0.1000-00E0-0222-86ED (SYSTM1).
    Error count                   0    Operations completed                0

Systm1>
```

7.5.5. Changing the Name of an Existing Fibre Channel Tape Device

Because SYS$SYSTEM:SYS$DEVICES.DAT is a text file, you can edit it but only to change the unit number of a Fibre Channel tape or medium changer device. However, as stated earlier, Fibre Channel tape and medium changer device information is stored internally by OpenVMS using clusterwide data structures, specifically clusterwide logical names. To clean up these data structures, you must do a complete cluster shutdown. A rolling reboot (leaving at least one node up during the reboot of other nodes) is inadequate to clean up the structures.

The specific steps for changing an existing device name follow:

1. Edit the SYS$SYSTEM:SYS$DEVICES.DAT file to change the unit number of the chosen device. In the basic 2MGA*nnnn* or 2GGA*nnnn* format, only the *nnnn* portion can be edited. The maximum allowed value for *nnnn* is 32767 and must be decimal. Be sure to choose a unit number that is not already in use by another device for that device type.

   For example, if 2MGA26 is already associated with the WWID of another tape, then choose a unit number other than 26; if 2GGA4 is already associated with the WWID of another medium changer, then choose a unit number other than 4.

2. If there are multiple system disks in the cluster, edit each of the SYS$DEVICES.DAT files in the same way.

3. Shut down the entire cluster to clean up existing cluster data structures.

4. Reboot the cluster. The new device names will automatically appear.

7.5.6. Moving a Physical Tape Device on Fibre Channel

When you move a tape or medium changer device without changing its name, rebooting is not required. However, you must ensure that the NSR or MDR has assigned a FC LUN to the device at its new location, and you must then run SYSMAN IO AUTOCONFIGURE to configure the new physical path to the device. For changers only, you must also manually switch the changer to the new path using the SET DEVICE/SWITCH/PATH=*new_path* command. The previous paths will still show up in the SHOW DEV/FULL display, but those paths will be stale and unused, with no harmful side effects; after the next reboot the stale paths will disappear.

7.5.7. Swapping Out an NSR on Fibre Channel

You can swap out an NSR without rebooting the Alpha system or Integrity server system.

After attaching the new NSR, use the Mapping submenu in the Visual Manager to populate the Indexed map on each Fibre Channel port of the NSR and reboot the NSR. An alternative way to map the new NSR is to copy the .cfg file from the previous NSR via the NSR's FTP utility.

Once the Indexed map is populated, run SYSMAN IO AUTOCONFIGURE to configure the new physical paths to the tape. For changers only, you must also manually switch the changer to the new path using the SET DEVICE/SWITCH/PATH=*new_path* command. The previous paths will still show up in the SHOW DEV/FULL display, but those paths will be stale and unused, with no harmful side effects; after the next reboot the stale paths will disappear.

7.5.8. Serving a Fibre Channel Tape Device

In general, all OpenVMS Alpha or Integrity server nodes in an OpenVMS Cluster have a direct path to Fibre Channel tape devices if the nodes are connected to the same Fibre Channel fabric as the NSR or MDR.

Medium changers, whether connected to Fibre Channel or to parallel SCSI, cannot be TMSCP served.

7.5.9. Replacing a Fibre Channel Tape Device

If one tape drive must be physically replaced by another tape drive at the same FC LUN location within the MDR or NSR, update the appropriate data structures with the IO REPLACE_WWID command.

For example, you may need to replace a defective tape drive with a new drive without rebooting the cluster, and that drive may need to retain the device name of the previous tape at that location.

The replacement device should have the same SCSI target ID as the original device. Cease all activity on the device, then type the following command to update all the necessary file and memory data structures with the WWID of the new tape drive:

```
$ MCR SYSMAN IO REPLACE_WWID $2$MGA1
```

Execute this command on each Alpha node in the cluster environment. You can accomplish this with the following commands:

```
$ MCR SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> IO REPLACE_WWID $2$MGA1
```

In some cases, this command may fail because the device name 2MGA1 no longer exists in the SHOW DEVICE display. This happens when the system has been rebooted some time after the drive has malfunctioned. In such a case, you must specify both the device name and the WWID, as shown in the following example.

The WWID must be the WWID of the new device that resides at the same Port/Target/LUN location as the replaced device. (To determine the value of the WWID that resides at a particular Port/Target/LUN location, use the SYSMAN IO LIST_WWID command).

```
$ MCR SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> IO REPLACE_WWID $2$MGA1/WWID=02000008:500E-09E0-0009-4E44
```

Note

This command should *not* be used to rename devices or redefine WWID correlations. The specified WWID must not be associated with any other device name in the cluster.

7.5.10. Determining the Physical Location of a Fibre Channel Tape Device

Given the name of a Fibre Channel tape device, it is helpful to know how to locate the Fibre Channel tape device. To do so, follow these steps:

1. From the system manager's account, run ANALYZE/SYSTEM.
2. At the SDA prompt, type CLUE SCSI/SUMMARY.
3. Search for the name of the device (for example, MGA3) in the column labeled DEVICE.
4. Note the corresponding value in the column labeled SCSI-LUN. This SCSI LUN value is the same value used by the MDR or NSR as the FC LUN. Also note the columns labeled Port and SCSI-ID for the device; all devices associated with that same port and SCSI-ID are attached to the same physical Fibre Channel port of the same MDR or NSR.
5. For the NSR, enter the Mapping submenu of the Visual Manager and click on Edit/View next to the Select Map box to display the Indexed Map for the current port.

   In the column labeled FC LUN, locate the value noted in step 4. Once you find the FC LUN value, note (on the same line) the corresponding values for SCSI Bus, SCSI Target ID, and SCSI LUN. This B:T:L information describes the physical location of the device within the NSR. Additional device information is available by clicking on 'Report' in the main menu of the Visual Manager.

   If an MDR is being used, at the MDR console's AMC prompt, similar information is available by typing `ShowFcScsiMap`.

7.5.11. Accessing a Fibre Channel Tape Device in a Standalone Environment

Fibre Channel tape devices can be configured in the context of booting from the CDROM distribution kit. The configuration steps are the same as the steps described in Section 7.5.4. Specifically, you must use the SYSMAN IO FIND_WWID and IO AUTOCONFIGURATION commands to configure the tape devices prior to use.

The file, SYS$DEVICES.DAT, is not created in this environment; therefore all pertinent naming information is stored in the memory data structures. Each time the CDROM is booted, you must repeat the IO FIND_WWID and IO AUTOCONFIGURE commands to name and configure the tape devices.

Note that the name of a Fibre Channel tape device in the CDROM boot environment does not persist through reboots, and may differ from the name that is assigned when booting from a read/write system disk

7.5.12. Multipath Tape Support

In a Fibre Channel configuration with SCSI tape devices attached to the Fibre Channel by means of an NSR or MDR, multiple paths can exist from an Alpha or an Integrity server system host to a SCSI tape. For example, an AlphaServer host with four KGPSA adapters has four distinct paths to a tape on the Fibre Channel. Furthermore, the NSR itself can be dual ported, allowing two paths into the NSR. An AlphaServer system with four KGPSAs leading to a dual-ported NSR actually has eight different paths from the AlphaServer system to a given tape drive.

OpenVMS systems configure and makes available all possible paths from an Alpha or an Integrity server system to the SCSI tape. You can specify a particular path with the DCL command SET DEVICE/SWITCH. Moreover, in the event of a broken connection, automatic failover takes place.

Note

Multipath failover between direct and MSCP-served paths is not supported for tape devices (unlike multipath failover between direct and MSCP-served paths for SCSI and Fibre Channel disks).

However, there is support for *TMSCP clients* of multipath sets, in which all members of the serving multipath set must be directly connected to the Fibre Channel. If one member of the set fails, another member will provide the local path to the device for use by the client.

7.6. Using the AlphaServer Console for Configuring FC (Alpha Only)

The AlphaServer console can be used to view the status of an FC interconnect. This allows you to confirm that the interconnect is set up properly before booting. If you plan to use an FC disk device for booting or dumping, you must perform some additional steps to set up those FC disk devices at the console. These topics are discussed in the next sections.

7.6.1. Viewing the FC Configuration from the Console

Console SHOW commands can be used to display information about the devices that the console detected when it last probed the system's I/O adapters. Unlike other interconnects, however, FC disk devices are not automatically included in the SHOW DEVICE output. This is because FC devices are identified by their WWIDs, and WWIDs are too large to be included in the SHOW DEVICE output. Instead, the console provides a command for managing WWIDs, named the `wwidmgr` command. This command enables you to display information about FC devices and to define appropriate device names for the FC devices that will be used for booting and dumping.

Note the following points about using the `wwidmgr` command:

- To use the `wwidmgr` command, if your system is an AlphaServer model 8*x*00, 4*x*00, or 1200, you must first enter diagnostic mode. On all other platforms, the `wwidmgr` command can be issued at any time.
- The changes made by the `wwidmgr` command do not take effect until after the next system initialization. After using the `wwidmgr` command, you must issue the `initialize` command.

Refer to the *Wwidmgr Users' Manual* for a complete description of the wwidmgr command. (The *Wwidmgr Users' Manual* is available in the [.DOC] directory of the Alpha Systems Firmware Update CD-ROM).

The following examples, produced on an AlphaServer 4100 system, show some typical uses of the wwidmgr command. Other environments may require additional steps to be taken, and the output on other systems may vary slightly.

Note the following about Example 7.1:

- The `wwidmgr -show wwid` command displays a summary of the FC devices on the system. This command does not display information about device connectivity.

- There are two FC adapters and five disks. (All the disks are listed at the end, independent of the adapters to which they are connected). In this example, each of the disks was assigned a device identifier at the HSG80 console. The console refers to this identifier as a user-assigned device identifier (UDID).

Example 7.1. Using wwidmgr -show wwid

```
P00>>>SET MODE DIAG
Console is in diagnostic mode
P00>>>wwidmgr -show wwid
polling kgpsa0 (KGPSA-B) slot 2, bus 0 PCI, hose 1
kgpsaa0.0.0.2.1            PGA0        WWN 1000-0000-c920-a7db
polling kgpsa1 (KGPSA-B) slot 3, bus 0 PCI, hose 1
kgpsab0.0.0.3.1            PGB0        WWN 1000-0000-c920-a694
[0] UDID:10 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016 (ev:none)
[1] UDID:50 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026 (ev:none)
[2] UDID:51 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027 (ev:none)
[3] UDID:60 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021 (ev:none)
[4] UDID:61 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022 (ev:none)
```

Example 7.2 shows how the `wwidmgr show wwid -full` command displays information about FC devices and how they are connected. The display has two parts:

- The first part lists each path from an adapter to an FC port. Adapters are identified by console device names, such as KGPSAA. FC ports are identified by their WWID, such as 5000-1FE1-0000-0D14. If any FC disks are found on a path, they are listed after that path. FC disks are identified by their current console device name, followed by their WWID.
- The second part of the display lists all the FC disks and the paths through which they are reachable. In this part, which begins with `[0]UDID:10...`, you will see there are four paths to each disk with two paths through each adapter, KGPSAA and KGPSAB. Each path through an adapter goes to a different port on the HSG or HSV. The column titled `Con` indicates whether the FC disk unit is currently online to the HSG or HSV controller that this path uses.

Example 7.2. Using wwidmgr -show wwid -full

```
P00>>>wwidmgr -show wwid -full

kgpsaa0.0.0.2.1
- Port: 1000-0000-c920-a7db

kgpsaa0.0.0.2.1
- Port: 2007-0060-6900-075b

kgpsaa0.0.0.2.1
- Port: 20fc-0060-6900-075b

kgpsaa0.0.0.2.1
- Port: 5000-1fe1-0000-0d14
 - dga12274.13.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016
 - dga15346.13.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026
 - dga31539.13.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027
 - dga31155.13.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021
 - dga30963.13.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022

kgpsaa0.0.0.2.1
- Port: 5000-1fe1-0000-0d11
 - dga12274.14.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016
```

```
 - dga15346.14.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026
 - dga31539.14.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027
 - dga31155.14.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021
 - dga30963.14.0.2.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022

kgpsab0.0.0.3.1
- Port: 1000-0000-c920-a694

kgpsab0.0.0.3.1
- Port: 2007-0060-6900-09b8

kgpsab0.0.0.3.1
- Port: 20fc-0060-6900-09b8

kgpsab0.0.0.3.1
- Port: 5000-1fe1-0000-0d13
 - dgb12274.13.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016
 - dgb15346.13.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026
 - dgb31539.13.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027
 - dgb31155.13.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021
 - dgb30963.13.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022

kgpsab0.0.0.3.1
- Port: 5000-1fe1-0000-0d12
 - dgb12274.14.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016
 - dgb15346.14.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026
 - dgb31539.14.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027
 - dgb31155.14.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021
 - dgb30963.14.0.3.1 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022


[0] UDID:10 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0016 (ev:none)
 - current_unit:12274 current_col: 0 default_unit:12274
          via adapter       via fc_nport       Con     DID     Lun
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d14   Yes   210013     10
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d11   No    210213     10
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d13   Yes   210013     10
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d12   No    210213     10

[1] UDID:50 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0026 (ev:none)
 - current_unit:15346 current_col: 0 default_unit:15346
          via adapter       via fc_nport       Con     DID     Lun
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d14   Yes   210013     50
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d11   No    210213     50
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d13   Yes   210013     50
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d12   No    210213     50

[2] UDID:51 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0027 (ev:none)
 - current_unit:31539 current_col: 0 default_unit:31539
          via adapter       via fc_nport       Con     DID     Lun
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d14   Yes   210013     51
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d11   No    210213     51
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d13   Yes   210013     51
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d12   No    210213     51

[3] UDID:60 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0021 (ev:none)
 - current_unit:31155 current_col: 0 default_unit:31155
          via adapter       via fc_nport       Con     DID     Lun
```

```
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d14   Yes   210013     60
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d11   No    210213     60
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d13   Yes   210013     60
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d12   No    210213     60

[4] UDID:61 WWID:01000010:6000-1fe1-0000-0d10-0009-8090-0677-0022 (ev:none)
 - current_unit:30963 current_col: 0 default_unit:30963
          via adapter       via fc_nport       Con     DID     Lun
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d14   Yes   210013     61
 -      kgpsaa0.0.0.2.1  5000-1fe1-0000-0d11   No    210213     61
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d13   Yes   210013     61
 -      kgpsab0.0.0.3.1  5000-1fe1-0000-0d12   No    210213     61
```

7.6.2. Setting Up FC Disks for Booting and Dumping

You must use the `wwidmgr` command to set up each device that you will use for booting or dumping. Once a device is set up, the console retains the information it requires to access the device in nonvolatile memory. You only have to rerun the `wwidmgr` command if the system configuration changes and the nonvolatile information is no longer valid.

The console provides a simplified setup command, called `wwidmgr -quickset`. This command can be used in either of the following cases:

- You are setting up just one device.
- All the devices you are setting up are accessed through the same ports on the HSG or HSV.

If neither description applies to your configuration, refer to the *Wwidmgr Users' Manual* for additional instructions.

Example 7.3 illustrates the `wwidmgr-quickset` command. Note the following:

- The command `wwidmgr -quickset -udid 10` sets up the FC disk whose HSG or HSV device identifier is equal to 10.
- The console device names are path dependent. Each path used to access an FC disk has a different name. In this example, the `wwidmgr -quickset` command establishes four console device names corresponding to the four paths from the host to the FC disk:
  - dga10.1001.0.2.1
  - dga10.1002.0.2.1
  - dgb10.1003.0.3.1
  - dgb10.1004.0.3.1
- The second command, `wwidmgr -quickset -udid 50`, sets up the FC disk whose HSG or HSV identifier is equal to 50.
- The changes made by the `wwidmgr` command do not take effect until after the next system initialization, so the next step is to issue an `initialize` command.
- After the initialization, the console `show device` command displays each FC adapter, followed by the paths through that adapter to each of the defined FC disks. The path-independent OpenVMS device name for each FC disk is displayed in the second column.

Example 7.3. Using widmgr -quickset

```
P00>>>wwidmgr -quickset -udid 10

Disk assignment and reachability after next initialization:


6000-1fe1-0000-0d10-0009-8090-0677-0016
                          via adapter:         via fc nport:
 connected:
dga10.1001.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d14      Yes
dga10.1002.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d11      No
dgb10.1003.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d13      Yes
dgb10.1004.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d12      No
P00>>>wwidmgr -quickset -udid 50

Disk assignment and reachability after next initialization:


6000-1fe1-0000-0d10-0009-8090-0677-0016
                          via adapter:         via fc nport:
 connected:
dga10.1001.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d14      Yes
dga10.1002.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d11      No
dgb10.1003.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d13      Yes
dgb10.1004.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d12      No

6000-1fe1-0000-0d10-0009-8090-0677-0026
                          via adapter:         via fc nport:
 connected:
dga50.1001.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d14      Yes
dga50.1002.0.2.1         kgpsaa0.0.0.2.1      5000-1fe1-0000-0d11      No
dgb50.1003.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d13      Yes
dgb50.1004.0.3.1         kgpsab0.0.0.3.1      5000-1fe1-0000-0d12      No
P00>>>initialize
Initializing...
P00>>>show device
polling ncr0 (NCR 53C810) slot 1, bus 0 PCI, hose 1   SCSI Bus ID 7
dka500.5.0.1.1     DKA500                    RRD45  1645
polling kgpsa0 (KGPSA-B) slot 2, bus 0 PCI, hose 1
kgpsaa0.0.0.2.1            PGA0        WWN 1000-0000-c920-a7db
dga10.1001.0.2.1   $1$DGA10                  HSG80  R024
dga50.1001.0.2.1   $1$DGA50                  HSG80  R024
dga10.1002.0.2.1   $1$DGA10                  HSG80  R024
dga50.1002.0.2.1   $1$DGA50                  HSG80  R024
polling kgpsa1 (KGPSA-B) slot 3, bus 0 PCI, hose 1
kgpsab0.0.0.3.1            PGB0        WWN 1000-0000-c920-a694
dgb10.1003.0.3.1   $1$DGA10                  HSG80  R024
dgb50.1003.0.3.1   $1$DGA50                  HSG80  R024
dgb10.1004.0.3.1   $1$DGA10                  HSG80  R024
dgb50.1004.0.3.1   $1$DGA50                  HSG80  R024
polling isp0 (QLogic ISP1020) slot 4, bus 0 PCI, hose 1   SCSI Bus ID 15
dkb0.0.0.4.1       DKB0                      RZ1CB-CS  0844
dkb100.1.0.4.1     DKB100                    RZ1CB-CS  0844
polling floppy0 (FLOPPY) PCEB - XBUS hose 0
dva0.0.0.1000.0    DVA0                       RX23
polling ncr1 (NCR 53C810) slot 4, bus 0 PCI, hose 0   SCSI Bus ID 7
dkc0.0.0.4.0       DKC0                      RZ29B  0007
```

```
polling tulip0 (DECchip 21040-AA) slot 3, bus 0 PCI, hose 0
ewa0.0.0.3.0       00-00-F8-21-09-74 Auto-Sensing
```

Example 7.4 shows a boot sequence from an FC system disk. Note the following:

- The boot device is `$1$DGA50`. The user has elected to enter all four paths to the device in the `bootdef_dev` string. This ensures that the system will be able to boot even if a path has failed.
- The first path on the boot command string, `dga50.1002.0.2.1`, is not currently connected (that is, the disk is not on line to the HSG80 on that path). The console indicates this fact, retries a few times, then moves on to the next path in the bootdef_dev string. This path is currently connected, and the boot succeeds.
- After booting, the OpenVMS SHOW DEVICE command confirms that OpenVMS has configured all five of the FC devices that were displayed by the `wwidmgr -show wwid` command, not only the two FC disks that were set up using the console `wwidmgr -quickset` command. The OpenVMS SHOW DEV/MULTIPATH command confirms that OpenVMS has configured all four paths to each disk.

Example 7.4. Boot Sequence from an FC System Disk

```
P00>>>set bootdef_dev dga50.1002.0.2.1,dga50.1001.0.2.1,dgb50.1003.0.3.1,
dgb50.1004.0.3.1
P00>>>b
(boot dga50.1002.0.2.1 -flags 0,0)
dga50.1002.0.2.1 is not connected
dga50.1002.0.2.1 is not connected
dga50.1002.0.2.1 is not connected
dga50.1002.0.2.1 is not connected
failed to open dga50.1002.0.2.1
(boot dga50.1001.0.2.1 -flags 0,0)
block 0 of dga50.1001.0.2.1 is a valid boot block
reading 919 blocks from dga50.1001.0.2.1
bootstrap code read in
Building FRU table
base = 200000, image_start = 0, image_bytes = 72e00
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code


    OpenVMS (TM) Alpha Operating System, Version V7.2
...

$ SHOW DEVICE

Device                  Device           Error    Volume         Free
 Trans Mnt
 Name                   Status           Count     Label        Blocks
 Count Cnt
$1$DGA10:     (FCNOD1)  Online               0
$1$DGA50:     (FCNOD1)  Mounted              0  V72_SSB        4734189
 303   1
$1$DGA51:     (FCNOD1)  Online               0
$1$DGA60:     (FCNOD1)  Online               0
```

```
$1$DGA61:      (FCNOD1)  Online               0

$ SHOW LOGICAL SYS$SYSDEVICE
   "SYS$SYSDEVICE" = "$1$DGA50:" (LNM$SYSTEM_TABLE)

$ SHO DEV/MULTI

Device                  Device           Error         Current
 Name                   Status           Count  Paths    path
$1$DGA10:      (FCNOD1)  Online               0   4/ 4
 PGB0.5000-1FE1-0000-0D11
$1$DGA50:      (FCNOD1)  Mounted              0   4/ 4
 PGA0.5000-1FE1-0000-0D12
$1$DGA51:      (FCNOD1)  Online               0   4/ 4
 PGA0.5000-1FE1-0000-0D13
$1$DGA60:      (FCNOD1)  Online               0   4/ 4
 PGB0.5000-1FE1-0000-0D14
$1$DGA61:      (FCNOD1)  Online               0   4/ 4
 PGB0.5000-1FE1-0000-0D11
Device                  Device           Error         Current
 Name                   Status           Count  Paths    path
$1$GGA42:               Online               0   4/ 4
 PGB0.5000-1FE1-0000-0D11
```

7.7. Booting on a Fibre Channel Storage Device on OpenVMS Integrity server Systems

This section describes how to boot the fibre channel (FC) storage device on OpenVMS Integrity server systems. FC storage is supported on all storage arrays that are supported on OpenVMS systems.

OpenVMS Integrity servers Version 8.2 supports the HP A6826A, a PCI-X dual-channel, 2-Gb Fibre Channel host-based adapter (HBA) and its variants. The A6826A HBA requires the following software and firmware:

- EFI driver Version 1.40
- RISC firmware Version 3.03.001

Fibre channel device booting supports point-to-point topology. There is no plan to support FC arbitrated loop topology.

7.7.1. Installing the Bootable Firmware

Before you can boot on a FC device on OpenVMS Integrity server systems, you must update the EFI bootable firmware of the flash memory of the FC HBA.

To flash the memory of the FC HBA, update the firmware of the following components:

- EFI driver firmware
- RISC firmware
- NVRAM resident in the FLASH ROM on the HBA

To update the firmware, use the efiutil.efi utility, which is located on the IPF Offline Diagnostics and Utilities CD.

To perform these firmware updates, complete the following steps:

1. Insert the IPF Offline Diagnostics and Utilities IPF CD.

2. To flash all adapters found on the system in batch mode, select the EFI Shell from the Boot Options list on the EFI Boot Manager menu.

   At the EFI console, enter the following commands (where `fs0:` represents the bootable partition on the CD-ROM):

   a. `fs0:\efi\hp\tools\io_cards\fc2p2g\ efiutil all info`

      This command provides the current EFI driver and RISC firmware version on all adapters in the system.

   b. `fs0:\efi\hp\tools\io_cards\fc2p2g\ efiutil all efi_write`

      This command updates the EFI driver.

   c. `s0:\efi\hp\tools\io_cards\fc2p2g\ efiutil all risc_fw_write`

      This command updates the RISC firmware.

   d. `s0:\efi\hp\tools\io_cards\fc2p2g\ efiutil all nvram_write`

      This command updates the NVRAM.

   e. `fs0:> reset`

      This command resets the system.

3. Alternatively, you can flash each adapter separately by specifying the adapter ID and firmware file name to write to the ROM, as follows:

   a. Boot the entry that corresponds to the DVD-ROM from the Boot Options list; or specify the CD Media by selecting the “Boot Option Maintenance Menu,” then selecting “Boot from a File,” then selecting “Removable Media Boot.”

   b. From the CD main menu, select “View I/O Cards FW Update and Configuration Utilities, and MCA Menu”, then select “2Gb Fibre Channel HBA Utility”. This invokes the efiutil CLI utility and displays a list of fibre channel adapters found in the system.

   c. Select the fibre channel adapter by specifying the index number. Update the EFI driver, RISC firmware driver, and the NVRAM. Repeat this step until all adapters have been updated. For example:

      ```
      efiutil.efi>  adapter
       Adapter index number [0]?
      efiutil.efi> efi_write
      efiutil.efi> risc_fw_write
      efiutil.efi> nvram_write
      ```

   d. Exit the efiutil CLI by typing Quit from the utility. This will bring you to the “I/O Cards Firmware and Configuration Menu.” Type q to return to the Main Menu. From the Main Menu, select X to exit and reboot the system.

7.7.2. Checking the Firmware Version

You can check the installation of the firmware version in two ways: from the console during system initialization, or by using the efiutil utility:

- The firmware version is shown in the booting console message that is displayed during system initialization, as shown in the following example:

```
HP 2 Port 2Gb Fibre Channel Adapter (driver 1.40, firmware 3.03.001)
```

- The firmware version number is also shown in the display of the efiutil info command:

```
fs0:\efi\hp\tools\io_cards\fc2p2g\efiutil info

Fibre Channel Card Efi Utility  1.20  (1/30/2003)

 2 Fibre Channel Adapters found:

 Adapter        Path                    WWN           Driver (Firmware)
  A0  Acpi(000222F0,200)/Pci(1|0)  50060B00001CF2DC  1.40  (3.03.001)
  A1  Acpi(000222F0,200)/Pci(1|1)  50060B00001CF2DE  1.40  (3.03.001)
```

7.7.3. Configuring the Boot Device Paths on the FC

For configuration booting on a Fibre Channel storage device, VSI recommends that you use the OpenVMS Integrity servers Boot Manager (BOOT_OPTIONS.COM) after completing the installation of VSI OpenVMS. Follow these steps:

1. From the OpenVMS Installation Menu, choose Option 7 “Execute DCL commands and procedures” to access the DCL prompt.

2. At the DCL prompt, enter the following command to invoke the OpenVMS Integrity servers Boot Manager utility:

```
$$$ @SYS$MANAGER:BOOT_OPTIONS
```

3. When the utility is invoked, the main menu is displayed. To add your system disk as a boot option, enter 1 at the prompt, as shown in the following example:

```
OpenVMS Integrity server Boot Manager Boot Options List Management
 Utility

(1) ADD an entry to the Boot Options list
(2) DISPLAY the Boot Options list
(3) REMOVE an entry from the Boot Options list
(4) MOVE the position of an entry in the Boot Options list
(5) VALIDATE boot options and fix them as necessary
(6) Modify Boot Options TIMEOUT setting

(B) Set to operate on the Boot Device Options list
(D) Set to operate on the Dump Device Options list
(G) Set to operate on the Debug Device Options list

(E) EXIT from the Boot Manager utility

You can also enter Ctrl-Y at any time to abort this utility
```

```
Enter your choice: 1
```

Note

While using this utility, you can change a response made to an earlier prompt by typing the "^" character as many times as needed. To abort and return to the DCL prompt, press **Ctrl/Y**.

4. The utility prompts you for the device name. Enter the system disk device you are using for this installation, as in the following example where the device is a multipath Fibre Channel device 1DGA1: (press **Return**):

```
Enter the device name (enter "?" for a list of devices): $1$DGA1:
```

5. The utility prompts you for the position you want your entry to take in the EFI boot option list. Enter 1 as in the following example:

```
Enter the desired position number (1,2,3,,,) of the entry.
To display the Boot Options list, enter "?" and press [Return].
Position [1]: 1
```

6. The utility prompts you for OpenVMS boot flags. By default, no flags are set. Enter the OpenVMS flags (for example, 0,1) followed by a Return, or press **Return** to set no flags as in the following example:

```
Enter the value for VMS_FLAGS in the form n,n.
VMS_FLAGS [NONE]:
```

7. The utility prompts you for a description to include with your boot option entry. By default, the device name is used as the description. You can enter more descriptive information as in the following example.

```
Enter a short description (do not include quotation marks).
Description ["$1$DGA1"]: $1$DGA1 OpenVMS V8.2 System

efi$bcfg: $1$dga1 (Boot0001) Option successfully added

efi$bcfg: $1$dga1 (Boot0002) Option successfully added

efi$bcfg: $1$dga1 (Boot0003) Option successfully added
```

8. When you have successfully added your boot option, exit from the utility by entering E at the prompt.

```
Enter your choice: E
```

9. Log out from the DCL prompt and shut down the Integrity server System.

For more information on this utility, refer to the *VSI OpenVMS System Manager's Manual*.

7.8. Storage Array Controllers for Use with VSI OpenVMS

Visit the VSI OpenVMS web page at for information about storage arrays that are supported by VSI OpenVMS.

7.9. Creating a Cluster with a Shared FC System Disk

To configure nodes in an OpenVMS Cluster system, you must execute the CLUSTER_CONFIG.COM (or CLUSTER_CONFIG_LAN.COM) command procedure. (You can run either the full version, which provides more information about most prompts, or the brief version).

For the purposes of CLUSTER_CONFIG, a shared Fibre Channel (FC) bus is treated like a shared SCSI bus, except that the allocation class parameters do not apply to FC. The rules for setting node allocation class and port allocation class values remain in effect when parallel SCSI storage devices are present in a configuration that includes FC storage devices.

To configure a new OpenVMS Cluster system, you must first enable clustering on a single, or standalone, system. Then you can add additional nodes to the cluster.

Example 7.5 shows how to enable clustering using brief version of CLUSTER_CONFIG_LAN.COM on a standalone node called FCNOD1. At the end of the procedure, FCNOD1 reboots and forms a one-node cluster.

Example 7.6 shows how to run the brief version of CLUSTER_CONFIG_LAN.COM on FCNOD1 to add a second node, called FCNOD2, to form a two-node cluster. At the end of the procedure, the cluster is configured to allow FCNOD2 to boot off the same FC system disk as FCNOD1.

The following steps are common to both examples:

1. Select the default option [1] for ADD.
2. Answer Yes when CLUSTER_CONFIG_LAN.COM asks whether there will be a shared SCSI bus. SCSI in this context refers to FC as well as to parallel SCSI.

   The allocation class parameters are not affected by the presence of FC.
3. Answer No when the procedure asks whether the node will be a satellite.

Example 7.5. Enabling Clustering on a Standalone FC Node

```
$ @CLUSTER_CONFIG_LAN BRIEF

                  Cluster Configuration Procedure
                   Executing on an Alpha System

    DECnet Phase IV is installed on this node.

    The LAN, not DECnet, will be used for MOP downline loading.
    This Alpha node is not currently a cluster member


MAIN MENU

   1. ADD FCNOD1 to existing cluster, or form a new cluster.
   2. MAKE a directory structure for a new root on a system disk.
   3. DELETE a root from a system disk.
   4. EXIT from this procedure.

Enter choice [1]: 1
```

```
Is the node to be a clustered node with a shared SCSI or Fibre Channel bus
 (Y/N)? Y

    Note:
Every cluster node must have a direct connection to every other node in
 the cluster.  Since FCNOD1 will be a clustered node with a shared SCSI or
 FC bus, and Memory Channel, CI, and DSSI are not present, the LAN will be
 used for cluster communication.

Enter this cluster's group number: 511
Enter this cluster's password:
Re-enter this cluster's password for verification:

Will FCNOD1 be a boot server [Y]? Y
    Verifying LAN adapters in LANACP database...
    Updating LANACP LAN server process volatile and permanent databases...
    Note:
The LANACP LAN server process will be used by FCNOD1 for boot serving
 satellites. The following LAN devices have been found:
    Verifying LAN adapters in LANACP database...

    LAN TYPE    ADAPTER NAME    SERVICE STATUS
    ========    ============    ==============
    Ethernet    EWA0            ENABLED


  CAUTION: If you do not define port allocation classes later in this
 procedure for shared SCSI buses, all nodes sharing a SCSI bus must have
 the same non-zero ALLOCLASS value. If multiple nodes connect to a shared
 SCSI bus without the same allocation class for the bus, system booting
 will halt due to the error or IO AUTOCONFIGURE after boot will keep the
 bus offline.

Enter a value for FCNOD1's ALLOCLASS parameter [0]: 5
Does this cluster contain a quorum disk [N]? N
Each shared SCSI bus must have a positive allocation class value. A shared
 bus uses a PK adapter. A private bus may use: PK, DR, DV.

When adding a node with SCSI-based cluster communications, the shared SCSI
 port allocation classes may be established in SYS$DEVICES.DAT.
Otherwise, the system's disk allocation class will apply.

A private SCSI bus need not have an entry in SYS$DEVICES.DAT. If it has an
 entry, its entry may assign any legitimate port allocation class value:

    n   where n = a positive integer, 1 to 32767 inclusive
    0   no port allocation class and disk allocation class does not apply
   -1   system's disk allocation class applies (system parameter ALLOCLASS)

When modifying port allocation classes, SYS$DEVICES.DAT must be updated for
 all affected nodes, and then all affected nodes must be rebooted.
The following dialog will update SYS$DEVICES.DAT on FCNOD1.

There are currently no entries in SYS$DEVICES.DAT for FCNOD1.
After the next boot, any SCSI controller on FCNOD1 will use FCNOD1's disk
 allocation class.
```

```
Assign port allocation class to which adapter [RETURN for none]: PKA
Port allocation class for PKA0: 10

        Port Alloclass   10    Adapter FCNOD1$PKA

Assign port allocation class to which adapter [RETURN for none]: PKB
Port allocation class for PKB0: 20

        Port Alloclass   10    Adapter FCNOD1$PKA
        Port Alloclass   20    Adapter FCNOD1$PKB

  WARNING: FCNOD1 will be a voting cluster member. EXPECTED_VOTES for
 this and every other cluster member should be adjusted at a convenient
 time before a reboot. For complete instructions, check the section on
 configuring a cluster in the "OpenVMS Cluster Systems" manual.

  Execute AUTOGEN to compute the SYSGEN parameters for your configuration
 and reboot FCNOD1 with the new parameters. This is necessary before FCNOD1
 can become a cluster member.

Do you want to run AUTOGEN now [Y]? Y

    Running AUTOGEN -- Please wait.

The system is shutting down to allow the system to boot with the generated
 site-specific parameters and installed images.

The system will automatically reboot after the shutdown and the upgrade
 will be complete.
```

Example 7.6. Adding a Node to a Cluster with a Shared FC System Disk

```
$ @CLUSTER_CONFIG_LAN BRIEF

                  Cluster Configuration Procedure
                   Executing on an Alpha System

    DECnet Phase IV is installed on this node.

The LAN, not DECnet, will be used for MOP downline loading.
FCNOD1 is an Alpha system and currently a member of a cluster so the
 following functions can be performed:

MAIN MENU

   1. ADD an Alpha node to the cluster.
   2. REMOVE a node from the cluster.
   3. CHANGE a cluster member's characteristics.
   4. CREATE a duplicate system disk for FCNOD1.
   5. MAKE a directory structure for a new root on a system disk.
   6. DELETE a root from a system disk.
   7. EXIT from this procedure.

Enter choice [1]: 1

This ADD function will add a new Alpha node to the cluster.
```

```
  WARNING: If the node being added is a voting member, EXPECTED_VOTES for
 every cluster member must be adjusted.  For complete instructions check
 the section on configuring a cluster in the "OpenVMS Cluster Systems"
 manual.

  CAUTION: If this cluster is running with multiple system disks and common
 system files will be used, please, do not proceed unless appropriate
 logical names are defined for cluster common files in SYLOGICALS.COM. For
 instructions, refer to the "OpenVMS Cluster Systems" manual.

Is the node to be a clustered node with a shared SCSI or Fibre Channel bus
 (Y/N)? Y
Will the node be a satellite [Y]? N
What is the node's SCS node name? FCNOD2
What is the node's SCSSYSTEMID number? 19.111
    NOTE: 19.111 equates to an SCSSYSTEMID of 19567
Will FCNOD2 be a boot server [Y]? Y
What is the device name for FCNOD2's system root
[default DISK$V72_SSB:]? Y
What is the name of FCNOD2's system root [SYS10]?
    Creating directory tree SYS10 ...
    System root SYS10 created

  CAUTION: If you do not define port allocation classes later in this
 procedure for shared SCSI buses, all nodes sharing a SCSI bus must have
 the same non-zero ALLOCLASS value. If multiple nodes connect to a shared
 SCSI bus without the same allocation class for the bus, system booting
 will halt due to the error or IO AUTOCONFIGURE after boot will keep the
 bus offline.

Enter a value for FCNOD2's ALLOCLASS parameter [5]:
Does this cluster contain a quorum disk [N]? N
Size of pagefile for FCNOD2 [RETURN for AUTOGEN sizing]?

A temporary pagefile will be created until resizing by AUTOGEN. The default
 size below is arbitrary and may or may not be appropriate.

Size of temporary pagefile [10000]?
Size of swap file for FCNOD2 [RETURN for AUTOGEN sizing]?

A temporary swap file will be created until resizing by AUTOGEN. The
 default size below is arbitrary and may or may not be appropriate.

Size of temporary swap file [8000]?
Each shared SCSI bus must have a positive allocation class value. A shared
 bus uses a PK adapter. A private bus may use: PK, DR, DV.

When adding a node with SCSI-based cluster communications, the shared SCSI
 port allocation classes may be established in SYS$DEVICES.DAT.
Otherwise, the system's disk allocation class will apply.

    A private SCSI bus need not have an entry in SYS$DEVICES.DAT. If it has
 an entry, its entry may assign any legitimate port allocation class value:

    n   where n = a positive integer, 1 to 32767 inclusive
    0   no port allocation class and disk allocation class does not apply
   -1   system's disk allocation class applies (system parameter ALLOCLASS)
```

```
When modifying port allocation classes, SYS$DEVICES.DAT must be updated for
 all affected nodes, and then all affected nodes must be rebooted.
The following dialog will update SYS$DEVICES.DAT on FCNOD2.

Enter [RETURN] to continue:

$20$DKA400:<VMS$COMMON.SYSEXE>SYS$DEVICES.DAT;1 contains port allocation
 classes for FCNOD2. After the next boot, any SCSI controller not assigned
 in SYS$DEVICES.DAT will use FCNOD2's disk allocation class.


Assign port allocation class to which adapter [RETURN for none]: PKA
Port allocation class for PKA0: 11

     Port Alloclass   11    Adapter FCNOD2$PKA

Assign port allocation class to which adapter [RETURN for none]: PKB
Port allocation class for PKB0: 20

     Port Alloclass   11    Adapter FCNOD2$PKA
     Port Alloclass   20    Adapter FCNOD2$PKB

Assign port allocation class to which adapter [RETURN for none]:

  WARNING: FCNOD2 must be rebooted to make port allocation class
 specifications in SYS$DEVICES.DAT take effect.
Will a disk local only to FCNOD2 (and not accessible at this time to
 FCNOD1) be used for paging and swapping (Y/N)? N

If you specify a device other than DISK$V72_SSB: for FCNOD2's page
 and swap files, this procedure will create PAGEFILE_FCNOD2.SYS and
 SWAPFILE_FCNOD2.SYS in the [SYSEXE] directory on the device you specify.

What is the device name for the page and swap files [DISK$V72_SSB:]?
%SYSGEN-I-CREATED, $20$DKA400:[SYS10.SYSEXE]PAGEFILE.SYS;1 created
%SYSGEN-I-CREATED, $20$DKA400:[SYS10.SYSEXE]SWAPFILE.SYS;1 created
    The configuration procedure has completed successfully.

    FCNOD2 has been configured to join the cluster.

The first time FCNOD2 boots, NETCONFIG.COM and AUTOGEN.COM will run
 automatically.
```

7.9.1. Configuring Additional Cluster Nodes to Boot with a Shared FC Disk (Integrity servers Only)

For configuring additional nodes to boot with a shared FC Disk on an OpenVMS Cluster system, VSI requires that you execute the OpenVMS Integrity servers Boot Manager (BOOT_OPTIONS.COM).

After you have enabled clustering on a single or standalone system, you can add additional Integrity server nodes to boot on a shared FC Disk, as follows:

1. Boot the VSI OpenVMS Installation Disk on the target node.

2. From the OpenVMS Installation Menu, choose Option 7 “Execute DCL commands and procedures.”

3. Follow the instructions in Section 7.7.3. Make sure that you set the correct system root when asked to enter the OpenVMS boot flags.

Note

The OpenVMS Integrity servers Boot Manager (BOOT_OPTIONS.COM) utility requires the shared FC disk to be mounted. If the shared FC disk is not mounted cluster-wide, the utility will try to mount the disk with a /NOWRITE option. If the shared FC disk is already mounted cluster-wide, user intervention is required. For more information on this utility, refer to the *VSI OpenVMS System Manager's Manual*.

7.9.2. Online Reconfiguration

The FC interconnect can be reconfigured while the hosts are running OpenVMS. This includes the ability to:

- Add, move, or remove FC switches and HSGs.
- Add, move, or remove HSG virtual disk units.
- Change the device identifier or LUN value of the HSG virtual disk units.
- Disconnect and reconnect FC cables. Reconnection can be to the same or different adapters, switch ports, or HSG ports.

OpenVMS does not automatically detect most FC reconfigurations. You must use the following procedure to safely perform an FC reconfiguration, and to ensure that OpenVMS has adjusted its internal data structures to match the new state:

1. Dismount all disks that are involved in the reconfiguration.
2. Perform the reconfiguration.
3. Enter the following commands on each host that is connected to the Fibre Channel:

```
SYSMAN> IO SCSI_PATH_VERIFY
SYSMAN> IO AUTOCONFIGURE
```

The purpose of the SCSI_PATH_VERIFY command is to check each FC path in the system's IO database to determine whether the attached device has been changed. If a device change is detected, then the FC path is disconnected in the IO database. This allows the path to be reconfigured for a new device by using the IO AUTOCONFIGURE command.

Note

In the current release, the SCSI_PATH_VERIFY command only operates on FC disk devices. It does not operate on generic FC devices, such as the HSG command console LUN (CCL). (Generic FC devices have names such as `$1$GGA`*`nnnnn`*`.` This means that once the CCL of an HSG has been configured by OpenVMS with a particular device identifier, its device identifier should not be changed.

7.9.3. HSG Host Connection Table and Devices Not Configured

When a Fibre Channel host bus adapter is connected (through a Fibre Channel switch) to an HSG controller, the HSG controller creates an entry in the HSG connection table. There is a separate

connection for each host bus adapter, and for each HSG port to which the adapter is connected. (Refer to the HSG CLI command SHOW CONNECTIONS for more information).

Once an HSG connection exists, you can modify its parameters by using commands that are described in the *HSG Array Controller ACS Configuration and CLI Reference Guide*. Since a connection can be modified, the HSG does not delete connection information from the table when a host bus adapter is disconnected. Instead, when the user is done with a connection, the user must explicitly delete the connection using a CLI command.

The HSG controller supports a limited number of connections: ACS V8.5 allows a maximum of 64 connections and ACS V8.4 allows a maximum of 32 connections. The connection limit is the same for both single- and dual-redundant controllers. Once the maximum number of connections is reached, then new connections will not be made. When this happens, OpenVMS will not configure disk devices, or certain paths to disk devices, on the HSG.

The solution to this problem is to delete old connections that are no longer needed. However, if your Fibre Channel fabric is large and the number of active connections exceeds the HSG limit, then you must reconfigure the fabric or use FC switch zoning to “hide” some adapters from some HSG ports to reduce the number of connections.

7.10. Using Interrupt Coalescing for I/O Performance Gains (Alpha Only)

Starting with OpenVMS Alpha Version 7.3-1, interrupt coalescing is supported for the KGPSA host adapters and is off by default. Interrupt coalescing can improve performance in environments with high I/O work loads by enabling the adapter to reduce the number of interrupts seen by a host. This feature is implemented in the KGPSA firmware.

You can read and modify the current settings for interrupt coalescing by means of the Fibre Channel Control Program (FC$CP). You must have the CMKRNL privilege to use FC$CP.

If you specify a response count and a delay time (in milliseconds) with FC$CP, the adapter defers interrupting the host until that number of responses is available or until that amount of time has passed, whichever occurs first.

Interrupt coalescing may cause a performance degradation to an application that does synchronous I/O. If no other I/O is going through a given KGPSA, the latency for single writes is an average of 900 microseconds longer with interrupt coalescing enabled (or higher depending on the selected response interval).

Interrupt coalescing is set on a per KGPSA basis. You should have an average of at least 2000 I/Os per second through a given KGPSA before enabling interrupt coalescing.

The format of the command is:

```
RUN SYS$ETC:FC$CP FGx enable-value [delay][response-count]
```

In this format

- For FG*x*, the valid range of *x* is A to Z.
- *enable-value* is a bit mask, with bit 1 controlling response coalescing and bit 0 controlling interrupt coalescing. The possible decimal values are:

  1=interrupt coalescing

2=response coalescing
3=interrupt coalescing and response coalescing

- *delay* (in milliseconds) can range from 0 to 255 decimal.
- *response-count* can range from 0 to 63 decimal.
- Any negative value leaves a parameter unchanged.
- Values returned are those that are current after any changes.

OpenVMS recommends the following settings for the FC$CP command:

```
$  RUN SYS$ETC:FC$CP FGx 2 1 8
```

7.11. Using Fast Path in Your Configuration

Fast Path support was introduced for Fibre Channel in OpenVMS Alpha Version 7.3 and is enabled by default. It is designed for use in a symmetric multiprocessor system (SMP). When Fast Path is enabled, the I/O completion processing can occur on all the processors in the SMP system instead of only on the primary CPU. Fast Path substantially increases the potential I/O throughput on an SMP system, and helps to prevent the primary CPU from becoming saturated.

The Fast Path also support the following features, which includes additional optimizations, preallocating of resources, and providing an optimized code path for mainline code:

- Reduced contention for the SCS/IOLOCK8 spinlock. The LAN drivers now synchronize using a LAN port-specific spinlock where possible.
- Offload of the primary CPU. The LAN drivers may be assigned to a secondary CPU so that I/O processing can be initiated and completed on the secondary CPU. This offloads the primary CPU and reduces cache contention between processors.

You can manage Fast Path programmatically using Fast Path system services. You can also manage Fast Path with DCL commands and by using the system parameters FAST_PATH and FAST_PATH_PORTS. For more information about using Fast Path, see the *VSI OpenVMS I/O User's Reference Manual*.

7.12. FIBRE_SCAN Utility for Displaying Device Information

FIBRE_SCAN.EXE displays information about all storage devices attached to Fibre Channel on the system; both configured and nonconfigured devices are included. The displayed information includes such data as the Fibre Channel target and LUN values, the vendor and product ID, device type, port and device worldwide identifiers (WWIDs), serial number, firmware revision level, and port login state. While the program primarily describes disk and tape devices, some limited information is also displayed for controller and other generic ($*n*$GGA*n*) devices.

FIBRE_SCAN can be invoked in two modes:

```
$ MCR SYS$ETC:FIBRE_SCAN        ! Scans all ports on the Fibre Channel.
$ MCR SYS$ETC:FIBRE_SCAN  PGx ! Scans only port x on the Fibre Channel.
```

FIBRE_SCAN requires CMKRNL and LOG_IO privilege.

To capture the FIBRE_SCAN output in a file, use a command such as the following before invoking FIBRE_SCAN:

```
$ DEFINE/USER SYS$OUTPUT xxx.log
```

FIBRE_SCAN is a display-only utility and is not capable of loading device drivers nor otherwise configuring devices on the Fibre Channel. To configure devices, use the SYSMAN IO AUTOCONFIGURE command.

7.13. SDA FC PERFORMANCE Command

The System Dump Analyzer (SDA) command FC PERFORMANCE was introduced in OpenVMS Version 8.2–1. FC PERFORMANCE is used to display I/O performance characteristics of DGA devices.

FC PERFORMANCE is also available in Version 8.2 of OpenVMS Alpha and OpenVMS Integrity servers. Furthermore, this command is available for OpenVMS Alpha Version 7.3-2 in the FIBRE_SCSI_V400 Fibre Channel patch kit. The Fibre Channel drivers of these supported versions keep a performance array for every configured disk.

You can use this SDA command to display I/O performance characteristics of a named DGA device or of all DGA devices that are configured on the system. If you omit the device name, then the performance data of all DGA disks with any nonzero data is displayed, regardless of whether the devices are currently mounted. FC PERFORMANCE arrays keep counts of I/O latency across LUNs, I/O size, and I/O direction (read/write).

I/O latency is measured from the time a request is queued to the host FC adapter until the completion interrupt occurs. Modern disk drives have an average latency in the range of 5-10 ms. Caches located in disk controllers (HSG/EVA/MSA/XP) and in the physical disk drives can occasionally produce significantly lower access times.

By default, the FC PERFORMANCE command uses the /SYSTIME qualifier to measure latency. The /SYSTIME qualifier uses EXE$GQ_SYSTIME, which is updated every millisecond. If I/O completes in less than 1 ms, it appears to have completed in zero time. When /SYSTIME is used, I/O operations that complete in less than 1 ms are shown in the display in the <2us column, where us represents microsecond.

To achieve greater accuracy, you can use the /RSCC qualifier, which uses the System Cycle Counter timer. The command qualifiers, including the timers, are described in Table 7.2.

Table 7.2. FC PERFORMANCE Command Qualifiers

Qualifier	Description
[*device-name*]	Device whose performance characteristics are to be displayed. You can specify only one DGA device name. If you omit the name, then the performance data for every DGA device configured on the system is displayed, provided the array contains nonzero data. This includes the performance data for DGA devices that are not currently mounted.
[/RSCC|/SYSTIME]	Two time qualifiers are available. The /RSCC qualifier uses a PAL call, Read System Cycle Counter, to get a CPU cycle counter. This is highly accurate but incurs the cost, in time, of two expensive PAL calls per I/O operation. To display time resolution below 1 ms, use this qualifier. The /SYSTIME qualifier is the OpenVMS system time which is updated every millisecond; it is the default.
[/COMPRESS]	Suppresses the screen display of columns that contain only zeroes.
[/CSV]	Causes output to be written to a comma-separated values (CSV) file, which can be read into a Microsoft® Excel spreadsheet or can be graphed.

Qualifier	Description
[/CLEAR]	Clears the performance array. This qualifier is useful when you are testing I/O performance on your system. Before starting a new test, you can clear the performance array for the device whose performance you are measuring and then immediately dump the contents of its performance array when the test is completed.

The following example shows the write and read output for device 1DGA4321 resulting from the use of the SDA command FC PERFORMANCE with the /COMPRESS qualifier.

In the display, LBC (first column heading) represents logical block count. Notice that the LBCs in the rows are powers of 2. A particular row contains all counts up to the count in the next row. A particular row contains all LBC counts up to the count that begins the next row; for example, the LBC 8 row shows the count for all I/Os with LBC 8 through LCB 15. I/Os with an LBC greater than 256 are not shown in the matrix, but they are included in the "total blocks" value at the beginning of the matrix.

The counts in each column represent the number of I/Os that completed in less than the time shown in the column header. For example the <2ms column means that the I/Os in this column took less than 2ms but more than 1 ms to complete. Similarly, the I/Os in the <4ms column took less than 4ms but more than 2 ms to complete. The one exception is the <2us column when you use the /SYSTIME qualifier; all I/O that completes in less than a millisecond are included in the <2us count.

The columns headings of <2us, <2ms, <4ms, and so on, are shown in this display. If there are no values for some of the headings, those columns are not displayed because the /COMPRESS qualifier was used. If the /RSCC qualifier was used instead of the default /SYSTIME qualifier, additional headings for <4us, <8us, <16us, and <256us would be displayed.

```
SDA> FC PERFORMANCE $1$DGA4321/COMPRESS

Fibre Channel Disk Performance Data
-----------------------------------

$1$dga4321 (write)

Using EXE$GQ_SYSTIME to calculate the I/O time

    accumulated write time = 2907297312us

    writes = 266709

    total blocks = 1432966


I/O rate is less than 1 mb/sec


LBC  <2us  <2ms  <4ms  <8ms <16ms <32ms <64ms <128ms <256ms <512ms <1s

===  ===== ====  ====  ==== ===== ===== ===== ====   =====  ===== ====

  1 46106 20630 12396 13605 13856 15334 14675 8101  777     8    - 145488

  2    52    21     8     9     5     5     6    1    2     -    -    109

  4 40310 13166  3241  3545  3423  3116  2351  977   88     -    -  70217
```

```
  8  2213  1355   360   264   205   225   164   82    5     -     -   4873

 16 16202  6897  3283  3553  3184  2863  2323 1012  108     -     1  39426

 32   678   310    36    39    47    44    33   27    6     -     -   1220

 64   105    97    18    26    41    43    42   24    7     -     -    403

128   592  3642   555    60    43    31    23    9    2     -     -   4957

256     -     9     7     -     -     -     -    -    -     -     -     16


   106258 46127 19904  21101 20804 21661 19617 10233 995     8     1 266709

Fibre Channel Disk Performance Data

---------------------------------

$1$dga4321 (read)

Using EXE$GQ_SYSTIME to calculate the I/O time

    accumulated read time = 1241806687us

    reads = 358490

    total blocks = 1110830


I/O rate is less than 1 mb/sec

LBC  <2us  <2ms  <4ms  <8ms  <16ms <32ms <64ms <128ms <256ms <512ms <2s
=== ===== =====  ===== ===== ===== =====  ===== ===== ====== ====== ====

  1  46620 12755  6587  7767  3758  2643 1133   198   5    -     -  81466

  2    574   134    66   158    82    20   21     4   1    -     -   1060

  4 162060 35896 20059 18677 15851 11298 5527  1300  25    2     1 270696

  8    355    79    46    97    59    36   28    10   -    -     -    710

 16    241   103    32   150    77    24   13     1   -    -     -    641

 32    916   355    76   302   316    61   25    10   -    -     -   2061

 64    725   380    64   248   140    17   10     3   -    -     -   1587

128     13    22    13    36    21     6    -     -   -    -     -    111

256     10    41    28    15    49    13    2     -   -    -     -    158

    211514 49765 26971 27450 20353 14118 6759  1526  31    2     1 358490

SDA>
```

Chapter 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.

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.

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 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.
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.	Automatic.

Mechanism	What Happens if a Failure Occurs	Type of Recovery
	Failure of a boot server does not affect nodes that have already booted, providing they have an alternate path to access MSCP served disks.	
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 aliases, generic queues, and autostart queues, refer to the *VSI OpenVMS Cluster Systems Manual* manual.

8.2.4. Related Software Products

Table 8.3 shows a variety of related OpenVMS Cluster software products that you can use to increase availability.

Table 8.3. Products That Increase Availability

Product	Description
Availability Manager	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.
RTR	Provides continuous and fault-tolerant transaction delivery services in a distributed environment with scalability and location transparency. In-

Product	Description
	flight transactions are guaranteed with the two-phase commit protocol, and databases can be distributed worldwide and partitioned for improved performance.
Volume Shadowing for OpenVMS	Makes any disk in an OpenVMS Cluster system a redundant twin of any other same-size disk (same number of physical blocks) in the OpenVMS Cluster.

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 10.5 and *VSI OpenVMS Cluster Systems Manual*.
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%

Strategy	Description
	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.
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

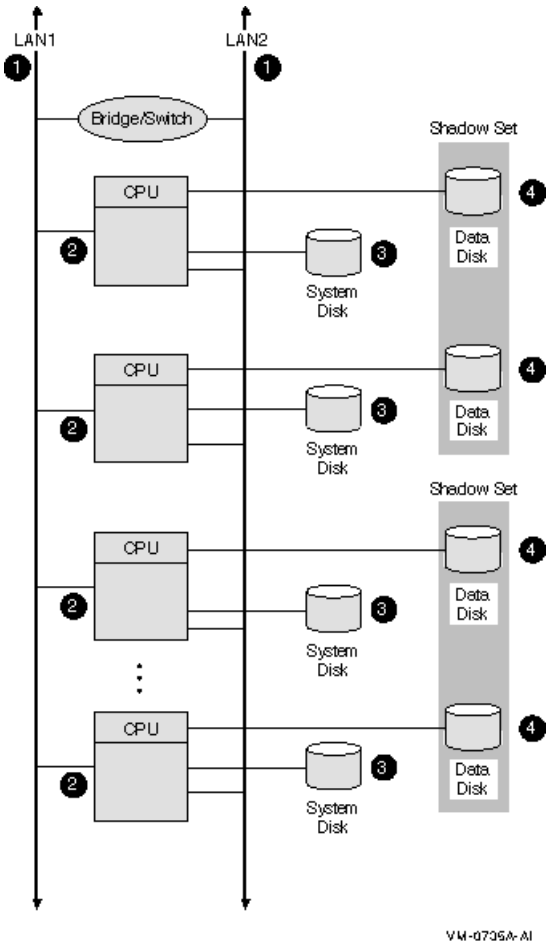
Strategy	Description
	change, test it in a noncritical environment before applying it to your production environment.
Reduce size and complexity	After you have achieved redundancy, reduce the number of components and the complexity of the configuration. A simple configuration minimizes the potential for user and operator errors as well as hardware and software errors.
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, refer to *VSI OpenVMS Cluster Systems Manual*.
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:

Component	Description
1	Two Ethernet interconnects. **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 8.9.

Component	Description
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 10.2.4 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.
- Use the Local Area OpenVMS Cluster Network Failure Analysis Program to monitor and maintain network availability. (See *VSI OpenVMS Cluster Systems Manual* for more information).
- Use the troubleshooting suggestions in *VSI OpenVMS Cluster Systems Manual* to diagnose performance problems with the SCS layer and the NISCA transport protocol.
- Keep LAN average utilization below 50%.

Reference: See Section 9.3.8 for information about extended LANs (ELANs). For differences between Alpha and Integrity server satellites, see the *VSI OpenVMS Cluster Systems Manual* manual.

8.6.1. Selecting MOP Servers

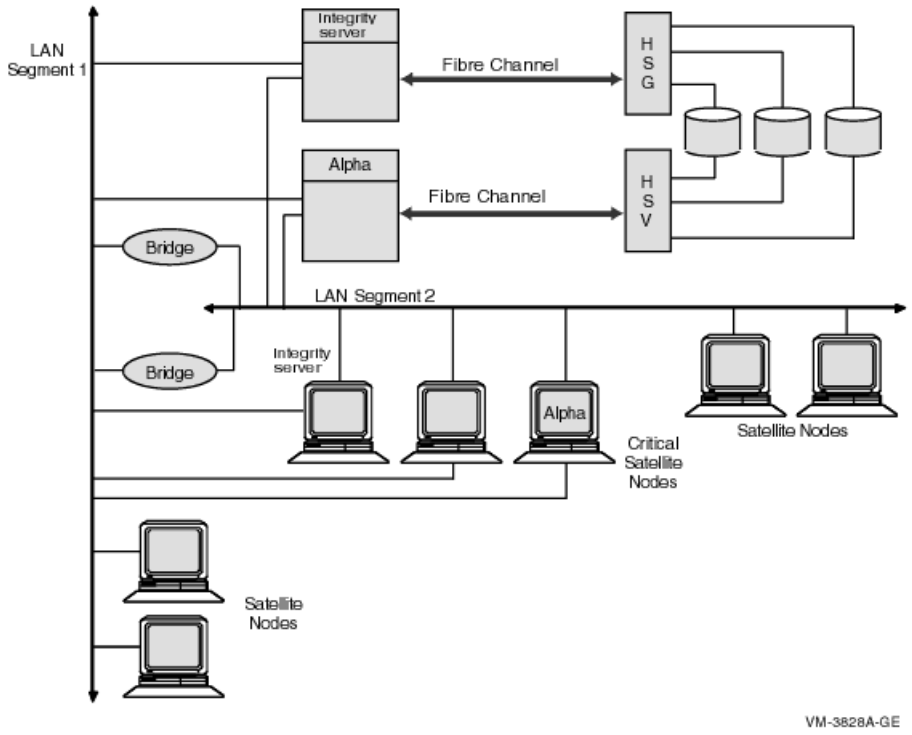
When using multiple LAN adapters with multiple LAN segments, distribute the connections to LAN segments that provide MOP service. The distribution allows MOP servers to downline load satellites even when network component failures occur.

It is important to ensure sufficient MOP servers for both Alpha and Integrity server 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 Integrity server and Alpha nodes, satellites, and two bridges.

Figure 8.2. Two-LAN Segment OpenVMS Cluster Configuration



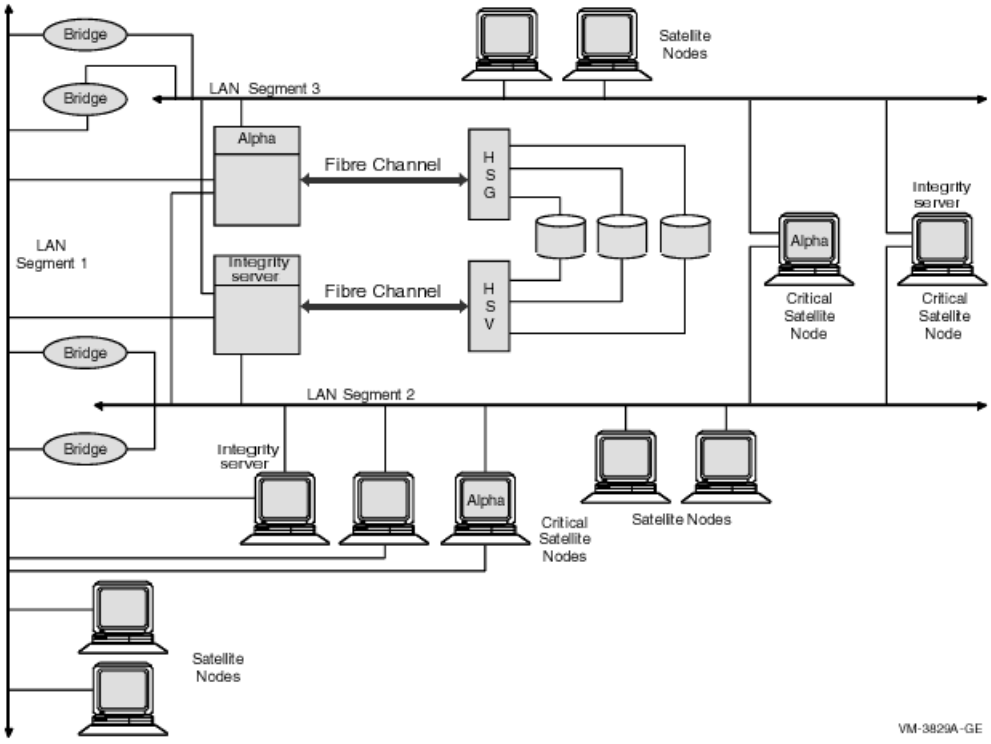
The figure illustrates the following points:

- 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) 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 Integrity server nodes, 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 for Alpha systems is provided for each LAN segment.

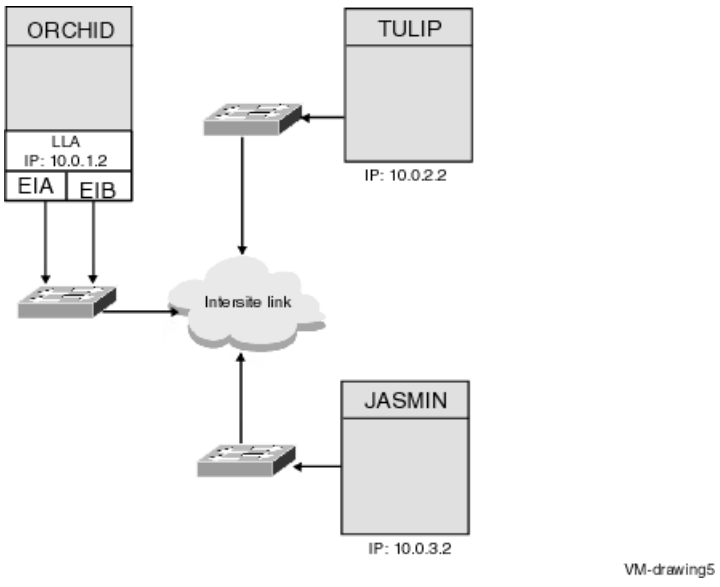
Reference: See Section 10.2.4 for more information about boot order and satellite dependencies in a LAN. See *VSI OpenVMS Cluster Systems Manual* for information about LAN bridge failover.

8.7. Availability in a Cluster over IP

Figure 8.4 shows an optimal configuration for a medium-capacity, highly available Logical LAN Failover IP 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. Logical LAN Failover IP OpenVMS Cluster System



8.7.1. Components

The IP OpenVMS Cluster configuration in Figure 8.4 has the following components:

Part	Description
1	EIA and EIB are the two IP interfaces connected to the node **Rationale:** Both the interfaces must be used to connect to the node for availability. EIA and EIB are the two LAN interfaces connected to the node. The two LAN interfaces

Part	Description
	can be used to create a Logical LAN failover set. Execute the following command to create a logical LAN failover set: `$ MC LANCPLANCP> DEFINE DEVICE LLB/ENABLE/FAILOVER=(EIA0, EIB0)` IP addresses can be configured on the Logical LAN failover device and can be used for cluster communication.
2	Intersite Link **Rationale:** Multiple intersite links can be obtained from either the same vendor or two different vendors ensuring site-to-site high availability.

8.7.2. Advantages

The configuration in Figure 8.4 offers the following advantages:

- High availability between site-to-site because of the multiple intersite links.
- Uses LLA, which ensures failure of one component to be taken over by the other node. Failure of LAN adapter is made transparent because of the availability of the other node.

8.7.3. Key Availability and Performance Strategies

The configuration in Figure 8.4 incorporates the following strategies, which are critical to its success:

- Define the device to enable logical LAN failover, a node will be able to survive if the local LAN card fails, it will switchover to other interface configured in the logical LAN failover set.
- All nodes are accessible by other nodes. To configure the logical LAN failover set, execute the following command:

  ```
  $ MC LANCPLANCP> DEFINE DEVICE LLB/ENABLE/FAILOVER=(EIA0, EIB0)
  ```

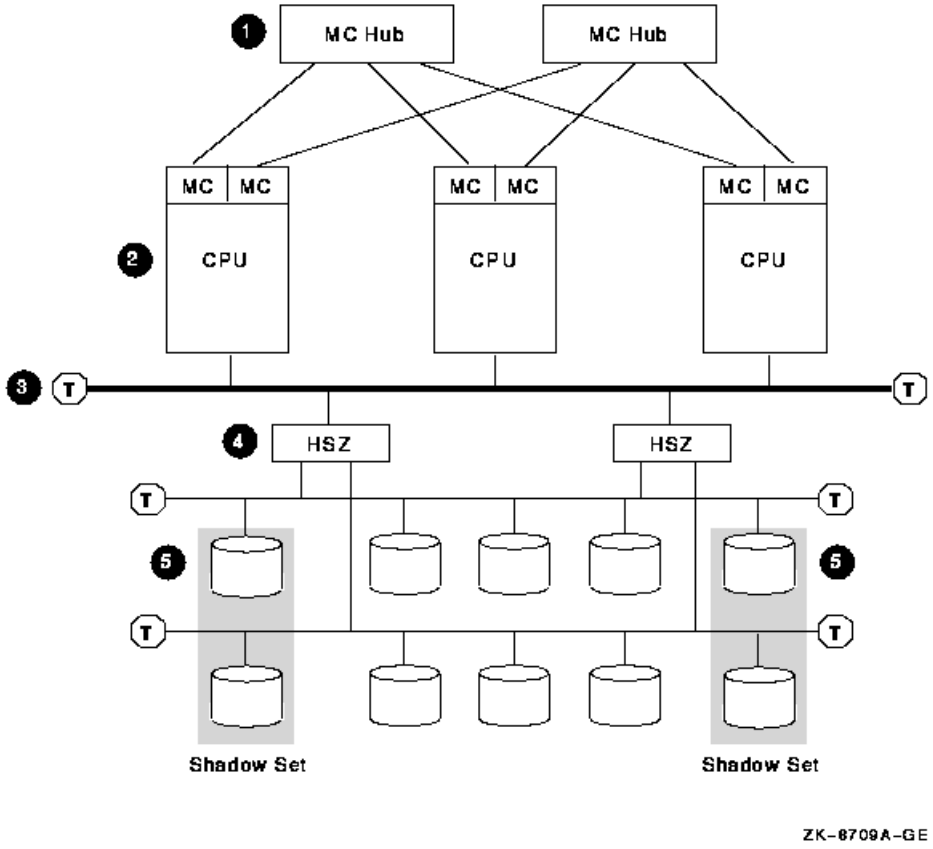
  See the example from Chapter 8 of the *VSI OpenVMS Cluster Systems Manual* manual that explains you how to add node ORCHID to an existing 2 node cluster with JASMIN and TULIP. The logical LAN failover set will be created and configured on ORCHID. ORCHID survives if the local LAN card fails, and it will switchover to other interface configured in the logical LAN failover set.

8.8. Availability in a MEMORY CHANNEL OpenVMS Cluster

Figure 8.5 shows a highly available MEMORY CHANNEL (MC) cluster configuration. 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. MEMORY CHANNEL Cluster



8.8.1. Components

The MEMORY CHANNEL configuration shown in Figure 8.5 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.
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.

Part	Description
	Rationale: Shadow essential disks and place shadow set members on different SCSI buses 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.
- 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.

8.8.3. 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 failover to the other. To use this functionality, the systems must be running OpenVMS Version 7.2 or higher and have multipath support enabled.

8.8.4. Key Availability Strategies

The configuration in Figure 8.5 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.

8.9. 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.

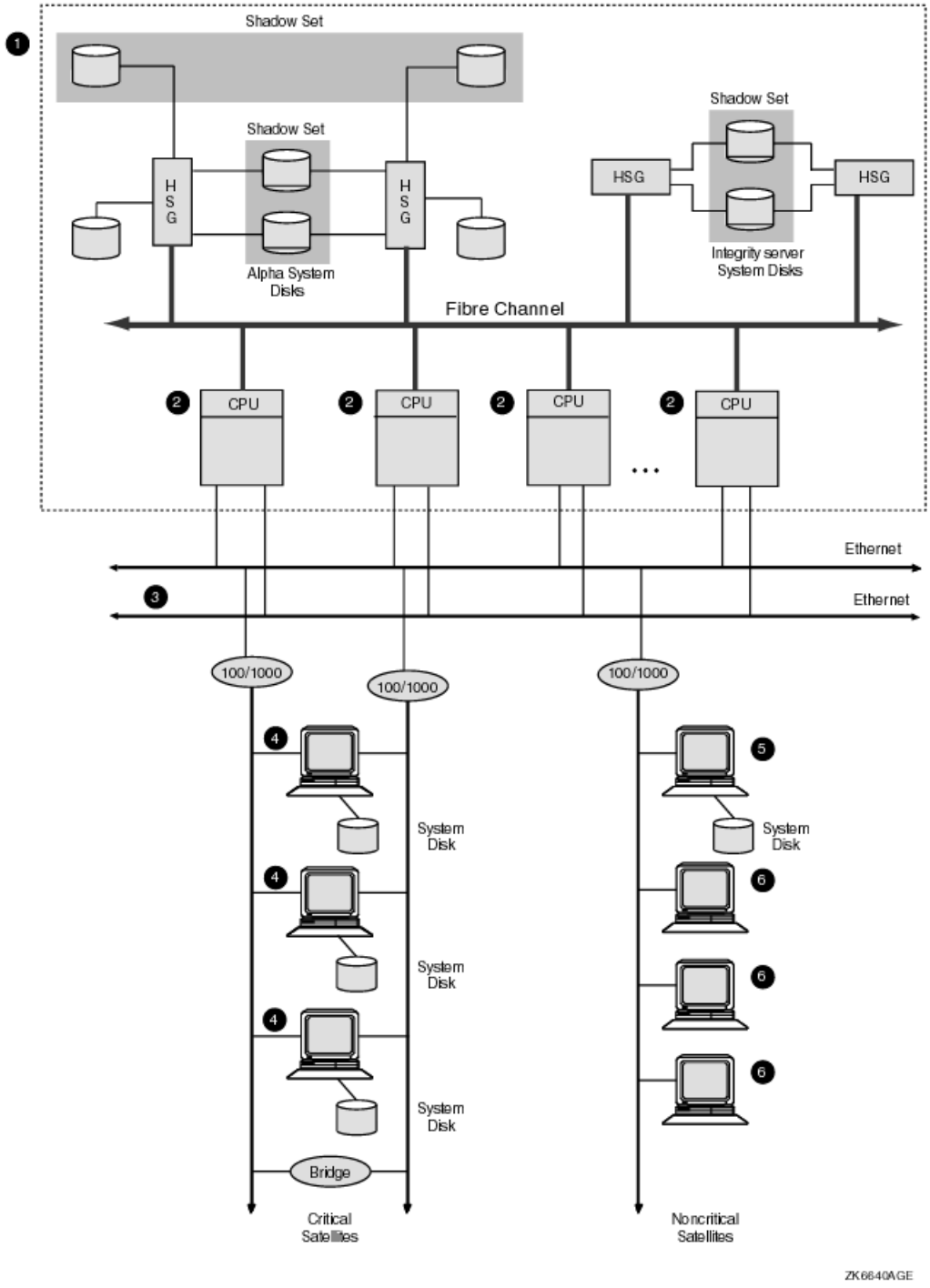
Because satellite nodes are highly dependent on server nodes for availability, the sample configurations presented earlier in this chapter do not include satellite nodes. However, because satellite/server configurations provide important advantages, you may decide to trade off some availability to include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

Figure 8.6 shows an optimal configuration for an OpenVMS Cluster system with satellites. 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. OpenVMS Cluster with Satellites



8.9.1. Components

This satellite/server configuration in Figure 8.6 has the following components:

Part	Description
1	Base configuration.

Part	Description
	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	Two Ethernet segments between base server nodes and satellites. **Rationale:** Provides high availability.
4	Two Ethernet segments attached to 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.

8.9.2. 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.

8.9.3. 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.

8.9.4. Key Availability Strategies

The configuration in Figure 8.6 incorporates the following strategies, which are critical to its success:

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

8.10. Multiple-Site OpenVMS Cluster System

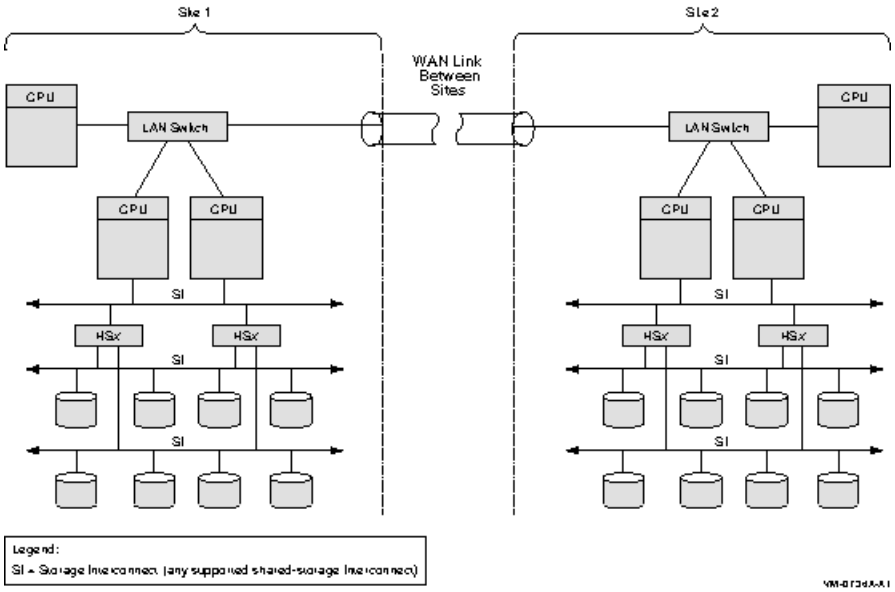
Multiple-site OpenVMS Cluster configurations contain nodes that are located at geographically separated sites. Depending on the technology used, the distances between sites can be as great as 150

miles. FDDI, and DS3 are used to connect these separated sites to form one large cluster. Available from most common telephone service carriers and DS3 services provide long-distance, point-to-point communications for multiple-site clusters.

Figure 8.7 shows a typical configuration for a multiple-site OpenVMS Cluster system. Figure 8.7 is followed by an analysis of the configuration that includes:

- Analysis of components
- Advantages

Figure 8.7. Multiple-Site OpenVMS Cluster Configuration Connected by WAN Link



8.10.1. Components

Although Figure 8.7 does not show all possible configuration combinations, a multiple-site OpenVMS Cluster can include:

- Two data centers with an intersite link (FDDI, 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 Integrity servers and Alpha (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 4.10.2. For ELAN configuration guidelines, see Section 9.3.8.

8.10.2. 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 phototype setters.
- 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 *VSI OpenVMS Cluster Systems Manual*.

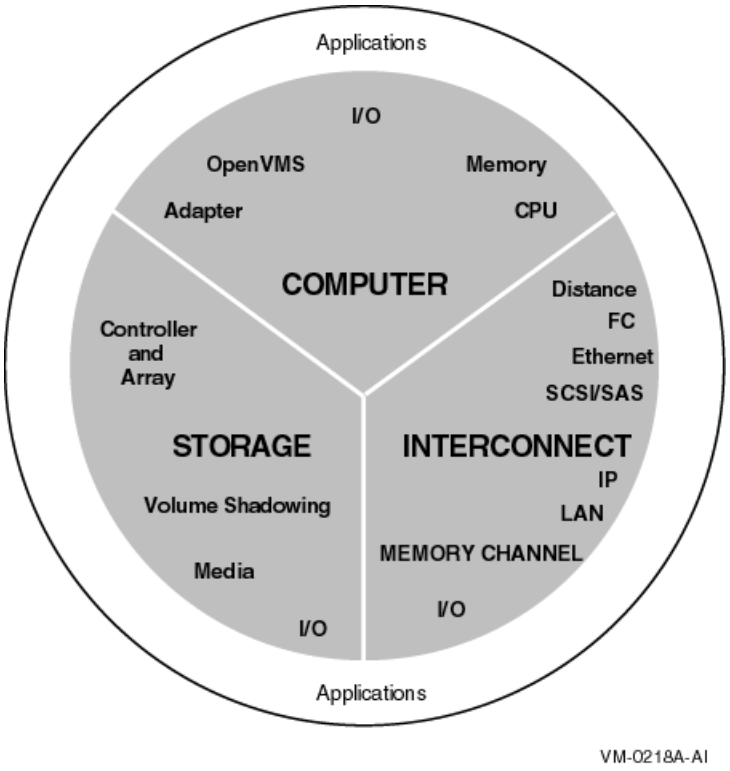
Chapter 9. Configuring OpenVMS Clusters for Scalability

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

9.1. 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 9.1. Each dimension also enables your applications to expand.

Figure 9.1. OpenVMS Cluster Growth Dimensions



9.1.1. Scalable Dimensions

Table 9.1 describes the growth dimensions for systems, storage, and interconnects in OpenVMS Clusters.

Table 9.1. 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 Alpha systems to Integrity servers.

This Dimension	Grows by...
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 Integrity servers.
Adapter	Adding storage 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 segments. Adding redundant segments and bridging segments.
Fibre Channel, SCSI, SAS, and MEMORY CHANNEL	Adding Fibre Channel, SCSI, SAS, 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 9.1 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. 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 9.2 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.

9.2. 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.

9.2.1. Scalability Strategies

Table 9.2 lists strategies in order of importance that ensure scalability. This chapter contains many figures that show how these strategies are implemented.

Table 9.2. Scalability Strategies

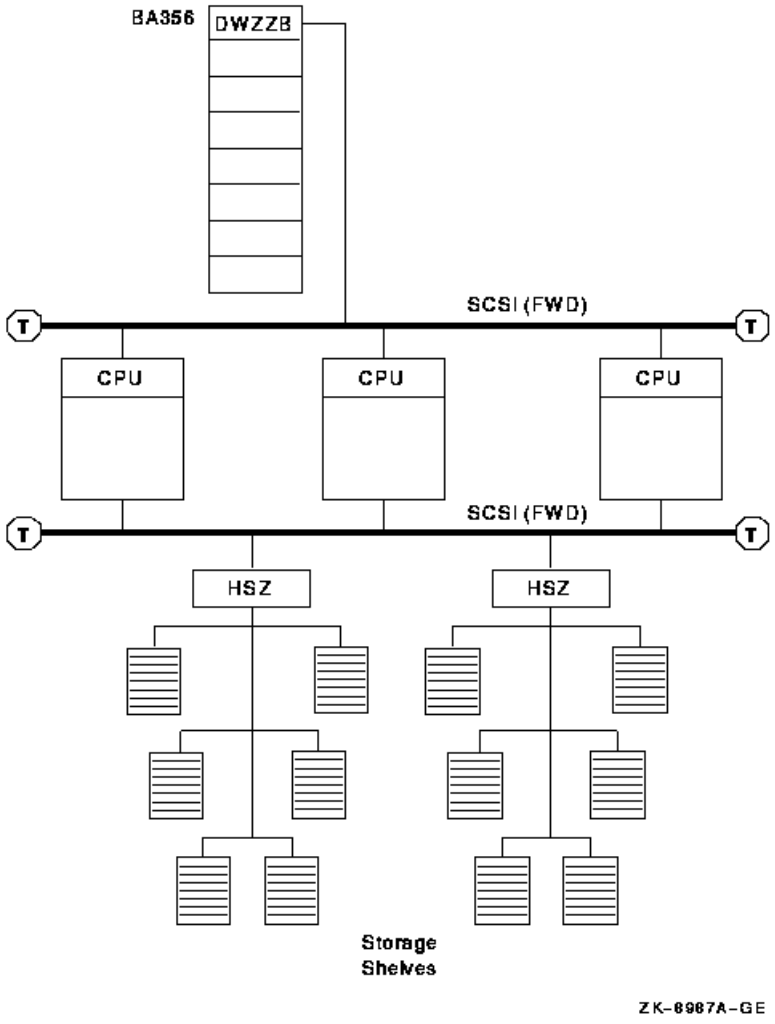
Strategy	Description
Capacity planning	Running a system above 80% capacity (near performance saturation) limits the amount of future growth possible. Understand whether your business and applications will grow. Try to anticipate future requirements for processor, memory, and I/O.
Shared, direct access 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 FC and LAN 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 9.5.1.
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.

Strategy	Description
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 10.2.4 for more information.
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.

9.2.2. Three-Node Fast-Wide SCSI Cluster

In Figure 9.2, three nodes are connected by two25-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 9.2. Three-Node Fast-Wide SCSI Cluster



The advantages and disadvantages of the configuration shown in Figure 9.2 include:

Advantages

- Combines the advantages of the configurations of two-node fast-wide SCSI cluster and two-node fast-wide SCSI cluster with HSZ storage:
  - 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.

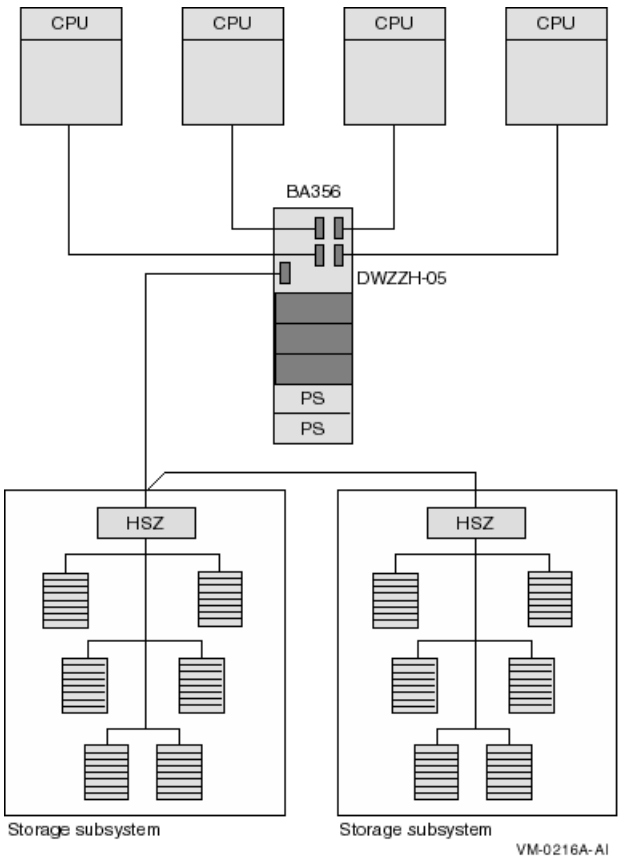
Disadvantage

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

9.2.3. Four-Node Ultra SCSI Hub Configuration

Figure 9.3 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 9.3. Four-Node Ultra SCSI Hub Configuration



The advantages and disadvantages of the configuration shown in Figure 9.3 include:

Advantages

- Provides significantly more bus distance and scalability.
- 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 the two-node fast-wide SCSI cluster and two-node fast-wide SCSI cluster HSZ storage.
- Only HSZ storage can be connected. You cannot attach a storage shelf with disk drives directly to the SCSI hub.

9.3. Scalability in OpenVMS Clusters with Satellites

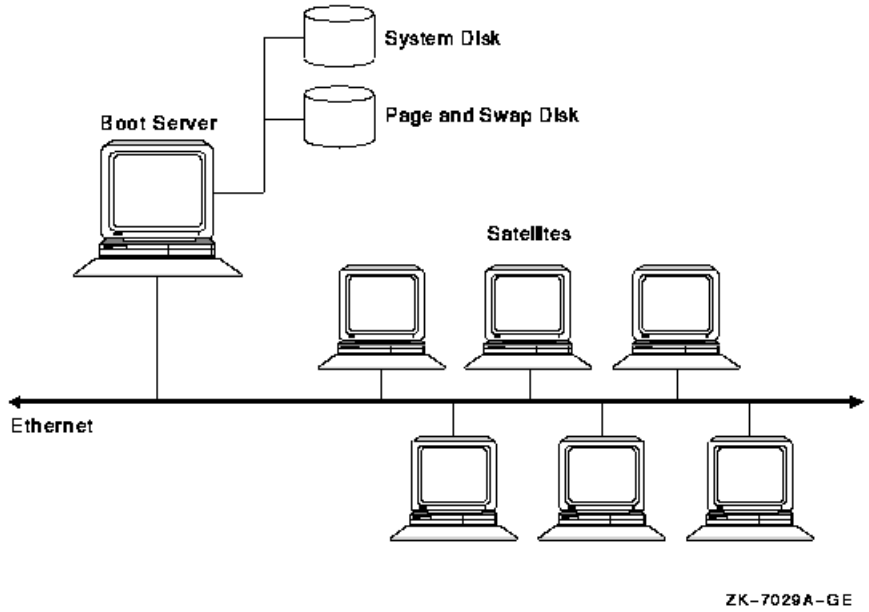
The number of satellites in an OpenVMS Cluster and the amount of storage that is MSCP served determine the need for the quantity and capacity of the servers. Satellites are systems that do not have direct access to a system disk and other OpenVMS Cluster storage. Satellites are usually workstations, 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 9.4, Figure 9.5, Figure 9.6, and Figure 9.7 show a progression from a 6-satellite LAN to a 45-satellite LAN.

9.3.1. Six-Satellite OpenVMS Cluster

In Figure 9.4, six satellites and a boot server are connected by Ethernet.

Figure 9.4. Six-Satellite LAN OpenVMS Cluster



The advantages and disadvantages of the configuration shown in Figure 9.4 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 10.2.

Disadvantage

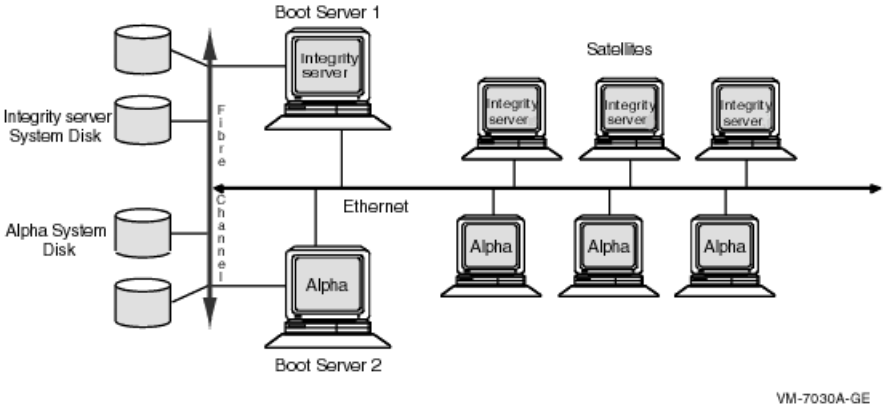
- The Ethernet is a potential bottleneck and a single point of failure.

If the boot server in Figure 9.4 became a bottleneck, a configuration like the one shown in Figure 9.5would be required.

9.3.2. Six-Satellite OpenVMS Cluster with Two Boot Nodes

Figure 9.5 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 9.5. Six-Satellite LAN OpenVMS Cluster with Two Boot Nodes



The advantages and disadvantages of the configuration shown in Figure 9.5 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

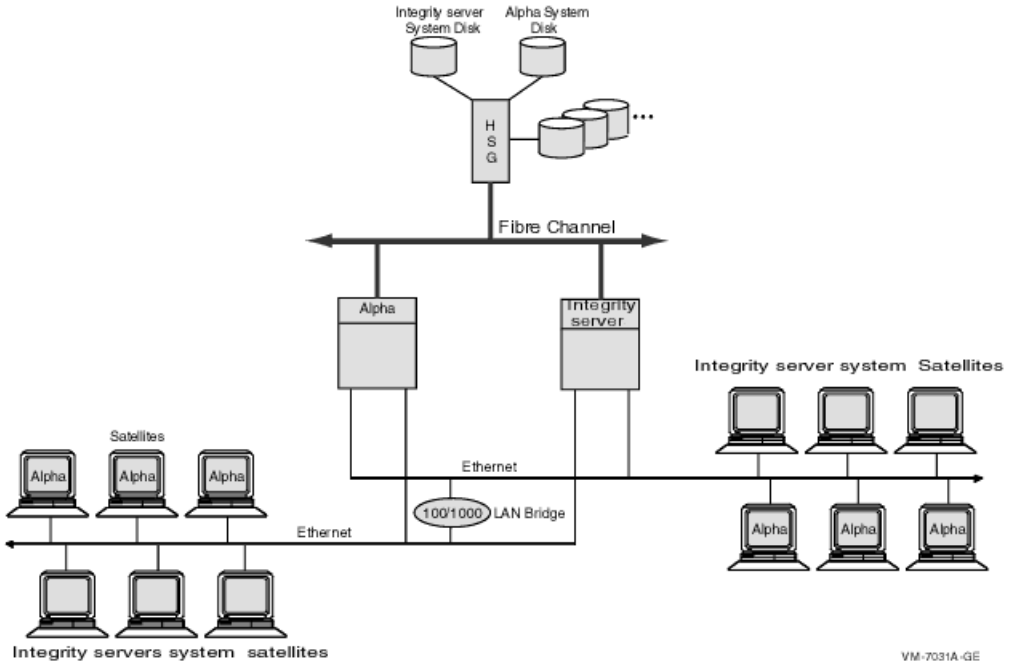
- The Ethernet is a potential bottleneck and a single point of failure.

If the LAN in Figure 9.5 became an OpenVMS Cluster bottleneck, this could lead to a configuration like the one shown in Figure 9.6.

9.3.3. Twelve-Satellite LAN OpenVMS Cluster with Two LAN Segments

Figure 9.6 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 9.6. Twelve-Satellite OpenVMS Cluster with Two LAN Segments



The advantages and disadvantages of the configuration shown in Figure 9.6 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 Integrity server nodes create the advantage of MSCP load balancing.
- Two LAN segments provide twice the amount of LAN capacity.

Disadvantages

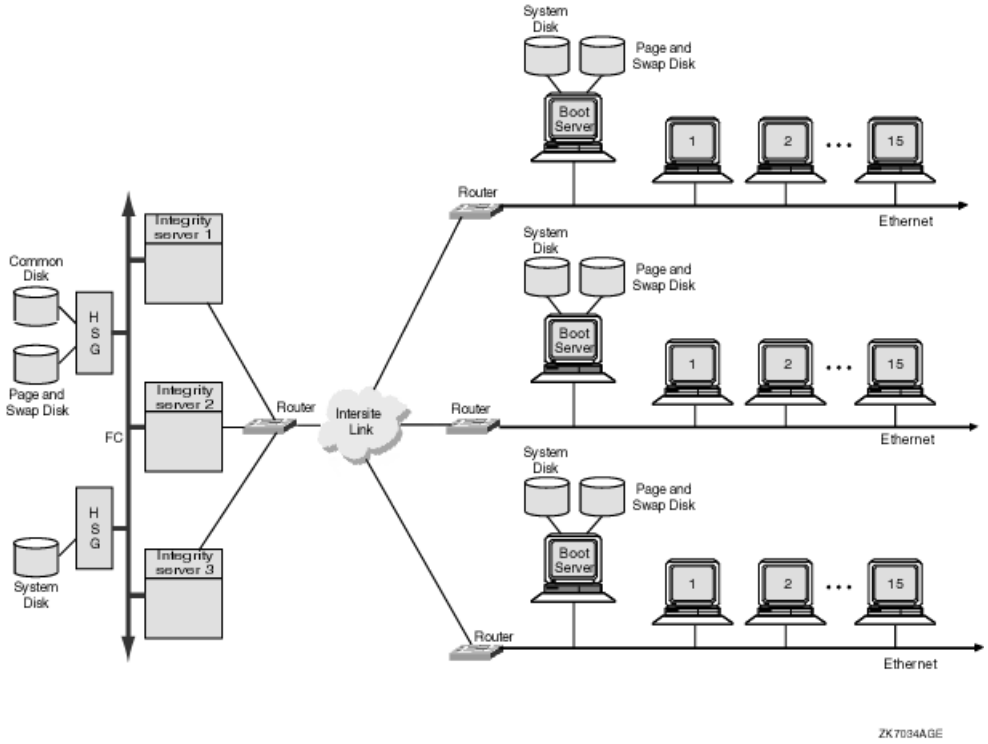
- This OpenVMS Cluster configuration is limited by the number of satellites that it can support.
- The single HSG controller is a potential bottleneck and a single point of failure.

If the OpenVMS Cluster in Figure 9.6 needed to grow beyond its current limits, this could lead to a configuration like the one shown in Figure 9.7.

9.3.4. Forty-Five Satellite OpenVMS Cluster with Intersite Link

Figure 9.7 shows a large, 51-node OpenVMS Cluster that includes 45 satellite nodes. The three boot servers, Integrity server 1, Integrity server 2, and Integrity server 3, share three disks: a common disk, a page and swap disk, and a system disk. The intersite link is connected to routers and has three LAN segments attached. Each segment has 15 workstation satellites as well as its own boot node.

Figure 9.7. Forty-Five Satellite OpenVMS Cluster with Intersite Link



The advantages and disadvantages of the configuration shown in Figure 9.7 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 9.7 see Section 10.2.4.

- 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 10.2.

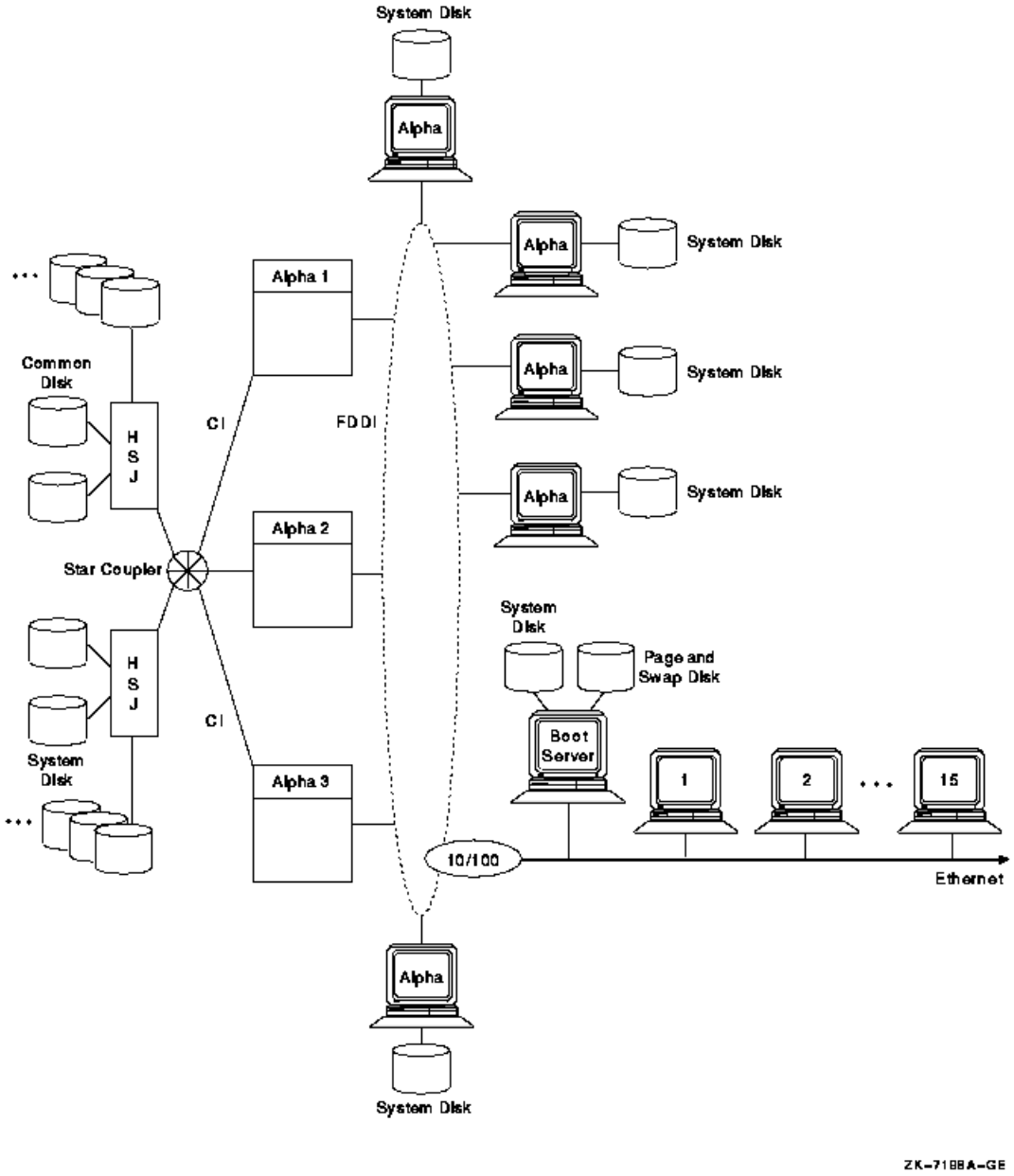
Disadvantages

- The satellite boot servers on the Ethernet LAN segments can boot satellites only on their own segments.

9.3.5. High-Powered Workstation OpenVMS Cluster (1995 Technology)

Figure 9.8 shows an OpenVMS Cluster configuration that provides high performance and high availability on the FDDI ring.

Figure 9.8. High-Powered Workstation Server Configuration 1995

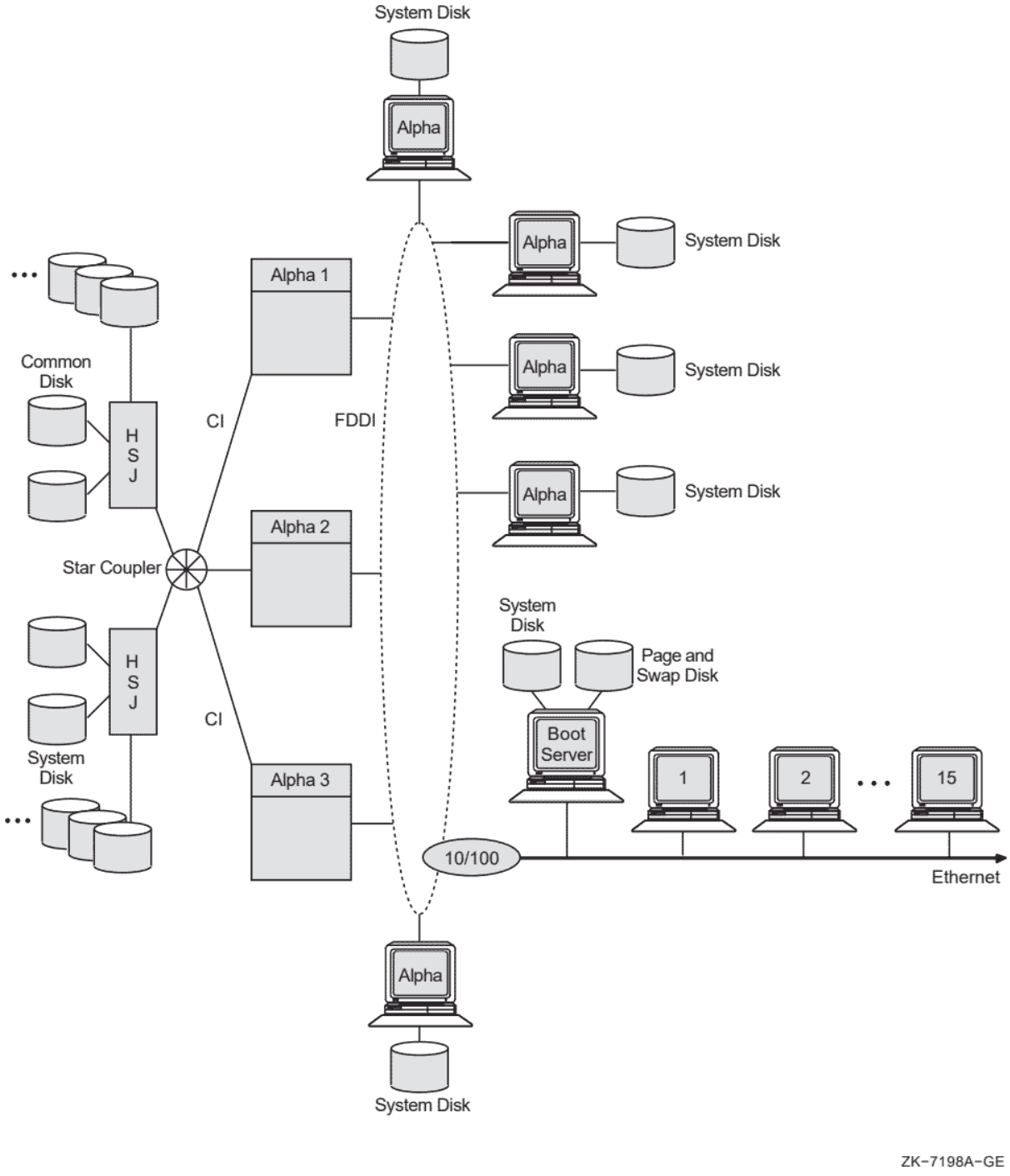


In Figure 9.8, several Alpha workstations, each with its own system disk, are connected to the FDDI ring. Putting Alpha workstations on the FDDI provides high performance because each workstation has direct access to its system disk. In addition, the FDDI bandwidth is higher than that of the Ethernet. Because Alpha workstation shave 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.

9.3.6. High-Powered Workstation OpenVMS Cluster (2004Technology)

Figure 9.9 shows an OpenVMS Cluster configuration that provides high performance and high availability using Gigabit Ethernet for the LAN and Fibre Channel for storage.

Figure 9.9. High-Powered Workstation Server Configuration 2004



In Figure 9.9, several Alpha workstations, each with its own system disk, are connected to the Ethernet LAN. Putting Alpha workstations on the Ethernet LAN provides high performance because each workstation has direct access to its system disk.

9.3.7. 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.

- Configure network to eliminate bottlenecks (that is, allocate sufficient bandwidth within the network cloud and on server connections).
- Maximize resources with MSCP dynamic load balancing, as shown in Figure 9.5 and Figure 9.6.
- Keep the number of nodes that require MSCP serving minimal for good performance.

  Reference: See Section 9.5.1 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 10.2.4 for more information about how to reduce LAN and system disk activity and how to boot separate groups of nodes in sequence.
- Use multiple LAN adapters per host, 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. In addition, multiple LAN adapters increase failover capabilities.

9.3.8. Extended LAN Configuration Guidelines

You can use bridges and switches between LAN segments to form an extended LAN. This can increase availability, distance, and aggregate bandwidth as compared with a single LAN. However, an extended LAN can increase delay and can reduce bandwidth on some paths. Factors such as packet loss, queuing delays, and packet size can also affect network performance. Table 9.3 provides guidelines for ensuring adequate LAN performance when dealing with such factors.

Table 9.3. Extended LAN Configuration Guidelines

Factor	Guidelines
Propagation delay	The amount of time it takes a packet to traverse the LAN depends on the distance it travels and the number of times it is relayed from one link to another through a switch or bridge. If responsiveness is critical, then you must control these factors. For high-performance applications, limit the number of switches between nodes to two. For situations in which high performance is not required, you can use up to seven switches or bridges between nodes.
Queuing delay	Queuing occurs when the instantaneous arrival rate at switches or bridges and host adapters exceeds the service rate. You can control queuing by: • Reducing the number of switches or bridges between nodes that communicate frequently. • Using only high-performance switches or 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 switches or bridges for traffic isolation.
Packet loss	Packets that are not delivered by the LAN require retransmission, which wastes system and network resources, increases delay, and reduces

Factor	Guidelines
	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. 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. LAN 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 *VSI OpenVMS Cluster Systems Manual*.
Switch or bridge recovery delay	Choose switches or bridges with fast self-test time and adjust them for fast automatic reconfiguration. This includes adjusting spanning tree parameters to match network requirements. **Reference:** Refer to *VSI OpenVMS Cluster Systems Manual* 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. For configurations that allow for faster speeds, enable jumbo frames where possible.
Traffic isolation	Use switches or bridges to isolate and localize the traffic between nodes that communicate with each other frequently. For example, use switches or bridges to separate the OpenVMS Cluster from the rest of the LAN and to separate nodes within an OpenVMS Cluster that communicate frequently from the rest of the OpenVMS Cluster. Provide independent paths through the LAN between critical systems that have multiple adapters.
Packet size	Ensure that the LAN path supports a data field of at least 4474 bytes end to end. For faster Ethernet devices using jumbo frames, set NISCS_MAX_PTKSZ to 8192 bytes. Some failures cause traffic to switch from an LAN path that supports a large packet size to a path that supports only smaller packets. It is possible to implement automatic detection and recovery from these kinds of failures.

9.3.9. 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 9.4 gives suggested values for these system parameters.

Table 9.4. OpenVMS Cluster System Parameters

System Parameter	Value for Satellites	Value for Servers
LOCKDIRWT	0	1-4. The setting of LOCKDIRWT influences a node's willingness to serve as a resource directory node and also may be used to determine mastership of resource trees. In general, a setting greater than 1 is determined after careful examination of a cluster node's specific workload and application mix and is beyond the scope of this document.
SHADOW_MAX_COPY	0	4, where a significantly higher setting may be appropriate for your environment
MSCP_LOAD	0	1
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 more information about these parameters, see *VSI OpenVMS Cluster Systems Manual* and *VSI OpenVMS Volume Shadowing Guide*.

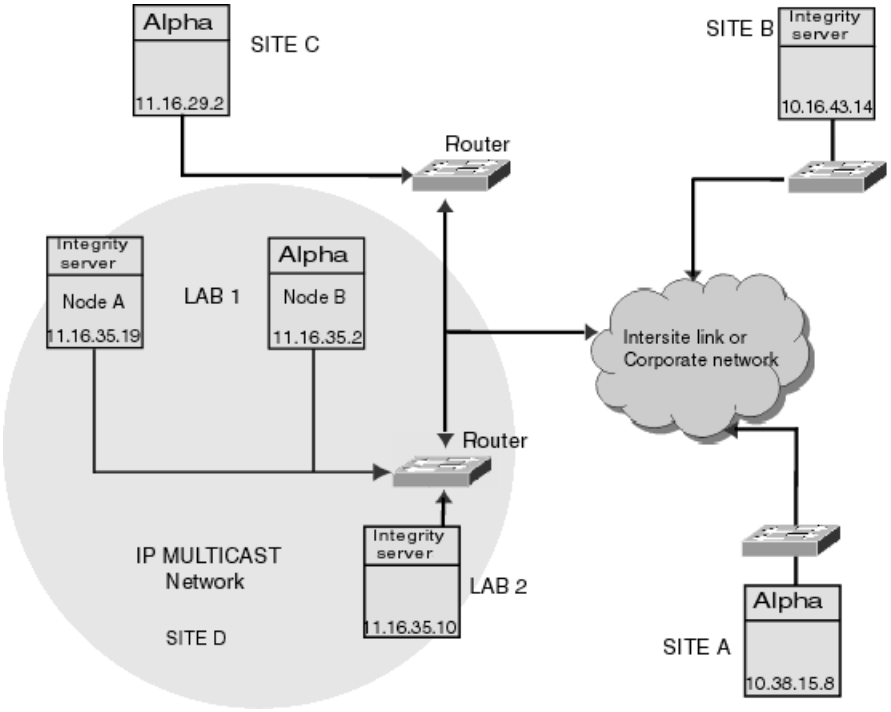
9.4. Scalability in a Cluster over IP

Cluster over IP allows a maximum of 96 nodes to be connected across geographical locations along with the support for storage. The usage of extended LAN configuration can be replaced by IP cluster communication. The LAN switches and bridges are replaced by the routers, thus overcoming the disadvantages of the LAN components. The routers can be used for connecting two or more logical subnets, which do not necessarily map one-to-one to the physical interfaces of the router.

9.4.1. Multiple node IP based Cluster System

Figure 9.10 shows an IP based cluster system that has multiple nodes connected to the system. The nodes can be located across different geographical locations thus, enabling high availability.

Figure 9.10. Multiple node IP based Cluster System



Advantages

- Cluster communication on IP supports Ethernet
- Easy to configure
- All nodes can access the other nodes and can have shared direct access to storage

9.4.2. Guidelines for Configuring IP based Cluster

The following are the guidelines for setting up a cluster using IP cluster communication:

- Requires the IP unicast address for remote node discovery
- Requires IP multicast address, which is system administrator scoped and is computed dynamically using the cluster group number. See *VSI OpenVMS Cluster Systems Manual* for information on cluster configuration.
- IP address of the local machine is required along with the network mask address
- Requires the local LAN adapter on which the IP address will be configured and is used for SCS.

9.5. 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

- Settings of the MSCP_BUFFER and MSCP_CREDITS system parameters
- 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.

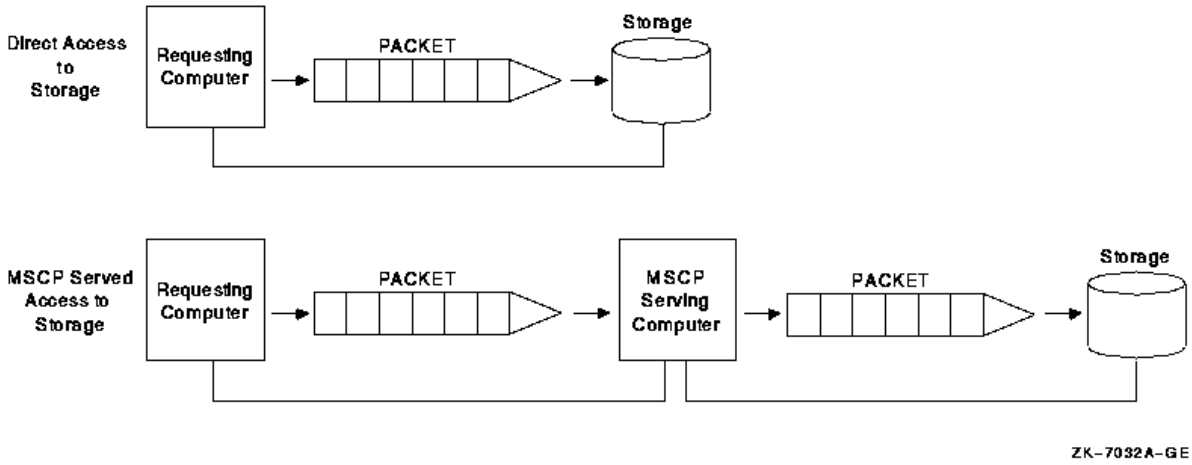
Additional factors that affect I/O throughput are types of interconnects and types of storage subsystems.

Reference: For more information about interconnects, see Chapter 4. For more information about types of storage subsystems, see Chapter 5.For more information about MSCP_BUFFER and MSCP_CREDITS, see *VSI OpenVMS Cluster Systems Manual*.

9.5.1. 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 9.11 is a simplification of how packets require extra handling by the serving system.

Figure 9.11. Comparison of Direct and MSCP Served Access



In Figure 9.11, 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.

9.5.2. 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 9.5 summarizes disk technologies and their features.

Table 9.5. 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 HSD, HSJ, and HSG controllers.

Note: Shared, direct access to a solid-state disk or to DECram is the fastest alternative for scaling I/Os.

9.5.3. 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 maybe caused by the time required to complete the slowest I/O.

To determine I/O read/write ratios, use the DCL command MONITOR IO.

9.5.4. 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.

9.5.5. 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 9.6 describes the three types of caching.

Table 9.6. Types of Caching

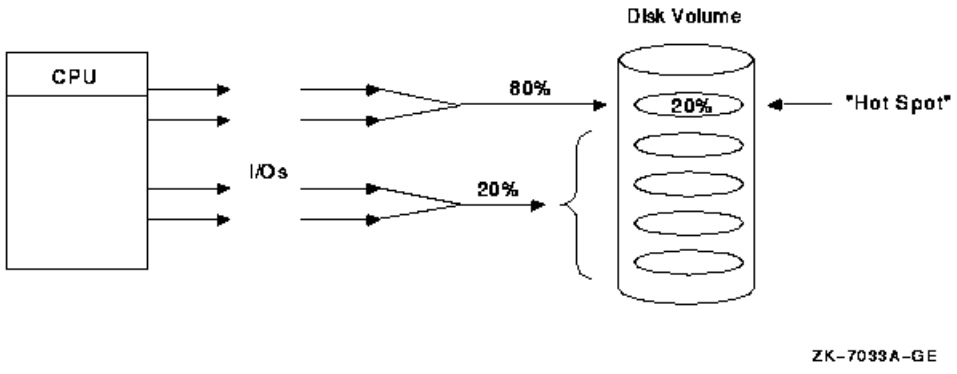
Caching Type	Description
Host based	Cache that is resident in the host system's memory and services I/Os from the host.
Controller based	Cache that is resident in the storage controller and services data for all hosts.
Disk	Cache that is resident in a disk.

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.

9.5.6. 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 9.12.

Figure 9.12. 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.

9.5.7. 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 devices 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 IP can be located up to 500 miles apart. Using Fibre Channel, they can be located up to 62 miles (100 kilometers) 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.

Chapter 10. 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 *VSI OpenVMS Cluster Systems Manual* 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
- Multiple platforms (Integrity servers and Alpha systems in the same OpenVMS Cluster)

10.1. 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.

Complex OpenVMS Cluster configurations may require a more sophisticated system management strategy to deliver more availability, scalability, and performance.

Reference: See Figure 10.2 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.

10.2. 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.

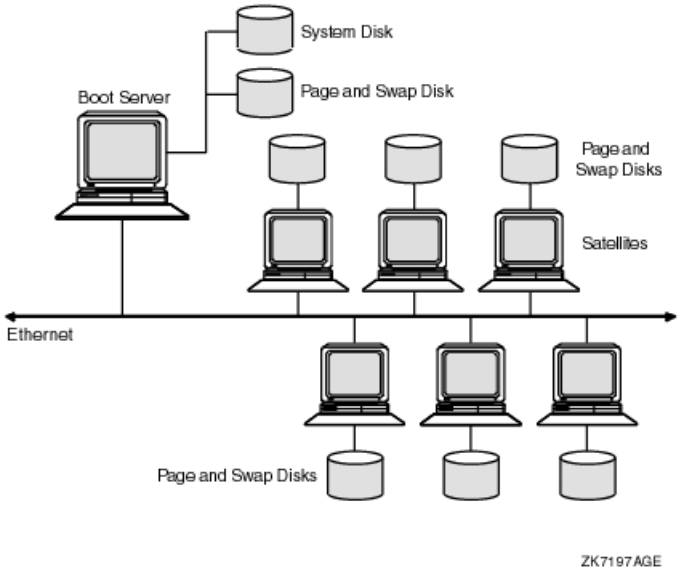
Environment files create specific working environments for users. You can create a common environment by making all environment files accessible clusterwide, or you can create multiple environments by making specific environment files accessible to only certain users or systems.

10.2.1. 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 10.1 shows another variation of a simple OpenVMS Cluster with a common environment.

Figure 10.1. Simple LAN OpenVMS Cluster with a Single System Disk



In Figure 10.1, 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.

10.2.2. Multiple System Disks

OpenVMS Clusters that include both Integrity servers and Alpha systems require multiple system disks: an Integrity server system disk and an Alpha system disk. Table 10.1 gives some additional reasons (not related to architecture)why a system manager might want more than one system disk in a OpenVMS Cluster.

Table 10.1. 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

Advantage	Description
	• 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 9.3.5 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 10.2.4 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.

10.2.3. Multiple System-Disk OpenVMS Cluster

Arranging system disks as shown in Figure 10.2 can reduce booting time and LAN utilization.

Figure 10.2. Multiple System Disks in a Common Environment

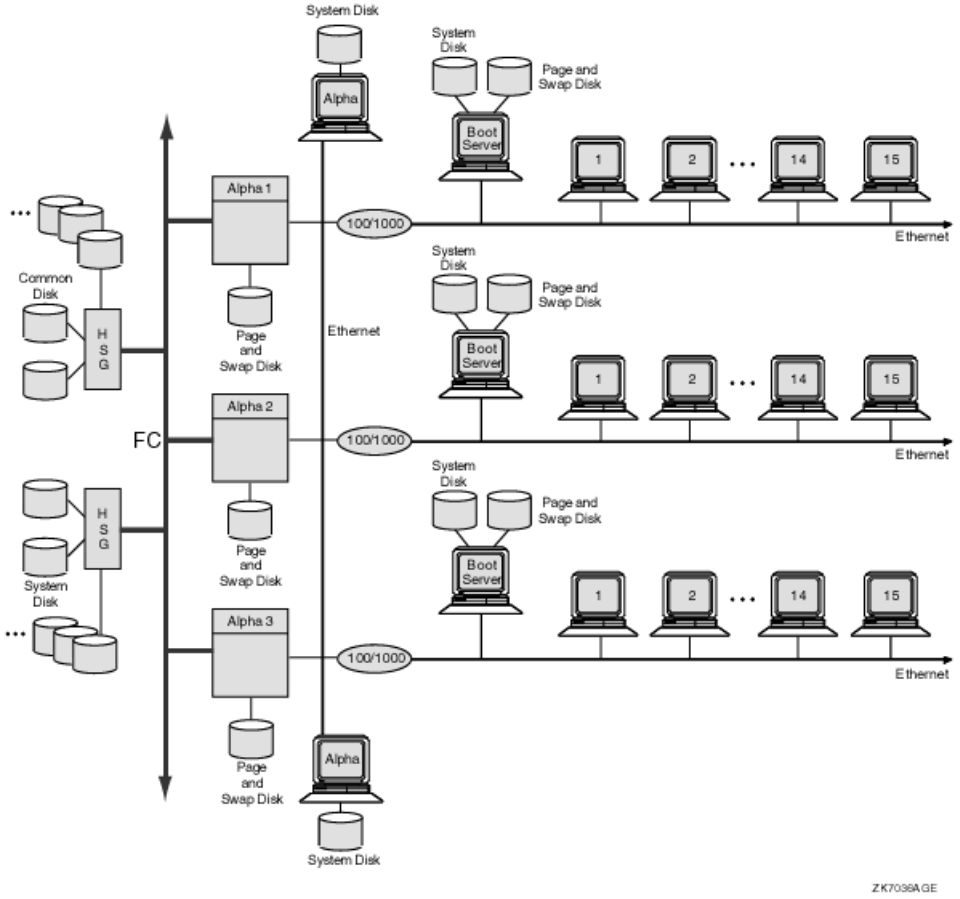


Figure 10.2 is an OpenVMS Cluster with multiple system disks:

- 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.

10.2.4. Dividing an OpenVMS Cluster System

In the workstation server examples shown in Section 9.3,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 10.2 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 10.2.

Table 10.2. 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—

Disk	Contents	Purpose
		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 10.3.
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 10.2, 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. For FDDI configuration, 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 9.3.7 for information about extended LANs.

10.2.5. Summary: Single Versus Multiple System Disks

Use the information in Table 10.3 to determine whether you need a system disk for the entire OpenVMS Cluster or multiple system disks.

Table 10.3. 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.

Single System Disk	Multiple System Disks
Lower hardware and software costs.	Higher hardware and software costs, especially if disks are shadowed.
Lower cost of system management because less time and experience required to manage a single system disk.	Higher cost of system management because more time and experience required to manage multiple system disks.

10.3. 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 *VSI OpenVMS Cluster Systems Manual*.

10.3.1. 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. 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.

10.3.2. 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 10.2 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 *VSI OpenVMS Cluster Systems Manual* for procedures to ensure that every node in the OpenVMS Cluster can access SYSUAF.DAT in its new location.

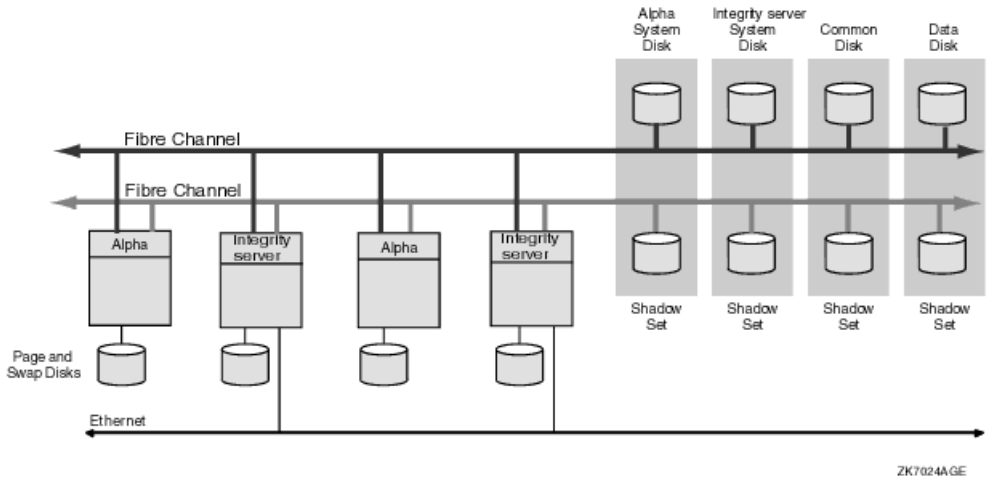
10.3.3. Multiple Environments

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 time sharing work.
- Allow users to work in environments that are specific to the node where they are logged in.

Figure 10.3 shows an example of a multiple environment.

Figure 10.3. Multiple-Environment OpenVMS Cluster



In Figure 10.3, the multiple-environment OpenVMS Cluster consists of two system disks: one for Integrity server 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 Fibre Channel nodes has its own page and swap disk, offloading the Alpha and Integrity server system disks on the FC interconnect from page and swap activity. All of the disks are shadowed across the FC interconnects, which protects the disks if a failure occurs.

10.4. Additional Multiple-Environment Strategies

This section describes additional multiple-environment strategies, such as using multiple SYSUAF.DAT files and multiple queue managers.

10.4.1. 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 Integrity servers and OpenVMS Alpha operating systems do not support multiple security domains.

Reference: See *VSI OpenVMS Cluster Systems Manual* 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 SYSUAF.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, VSI 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.

10.4.2. 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 *VSI OpenVMS System Manager's Manual* for examples and commands to implement multiple queue managers.

10.5. 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.

10.5.1. 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 10.4 describes four options from which to choose.

Table 10.4. 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 the 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 that both nodes be directly connected to storage (by CI, DSSI, SCSI, or Fibre Channel) for both to 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 *VSI OpenVMS Cluster Systems Manual*.

10.6. 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

10.6.1. 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 10.5 describes OpenVMS Cluster polling parameters that you can adjust for quicker detection time. VSI recommends that these parameters be set to the same value in each OpenVMS Cluster member.

Table 10.5. 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.

Parameter	Description
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 *VSI OpenVMS Cluster Systems Manual*.

10.7. Migration and Warranted Support for Multiple Versions

VSI provides two levels of support, warranted and migration, for mixed-version and mixed-architecture OpenVMS Cluster systems.

Warranted support means that VSI has fully qualified different versions coexisting in an OpenVMS Cluster and will answer all problems identified by customers using these configurations. See the Software Product Description for the complete list of warranted configurations for this release.

Migration support helps customers move to warranted OpenVMS Cluster configurations with minimal impact on their cluster environments. Migration support means that VSI has qualified the versions for use together in configurations that are migrating in a staged fashion to a newer version of OpenVMS Alpha or OpenVMS Integrity servers. Problem reports submitted against these configurations will be answered by VSI. However, in exceptional cases, VSI might request that you move to a warranted configuration as part of the solution.

In a mixed-version cluster, you must install remedial kits on earlier versions of OpenVMS.

Appendix 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 an 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.

Note

VSI does not support the latest SCSI adapters on OpenVMS Alpha systems.

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:

- *VSI OpenVMS Cluster Systems Manual*
- OpenVMS Cluster Software *Software Product Description*

This appendix includes two primary parts:

- Section A.1 through Section A.6.6describe 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.

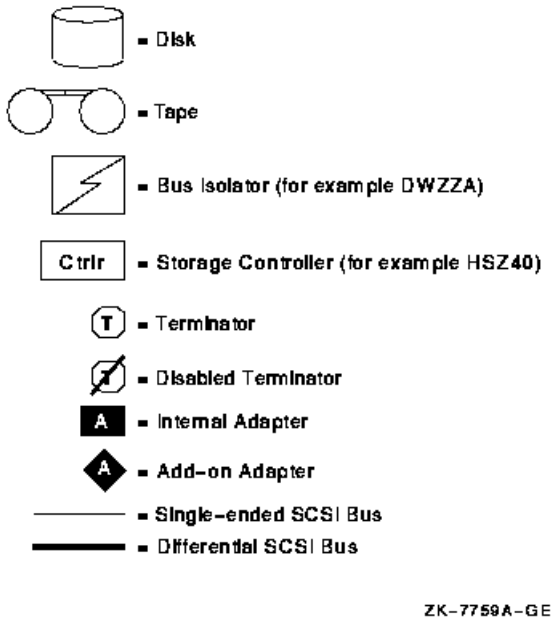
A.1.1. SCSI ANSI Standard

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



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

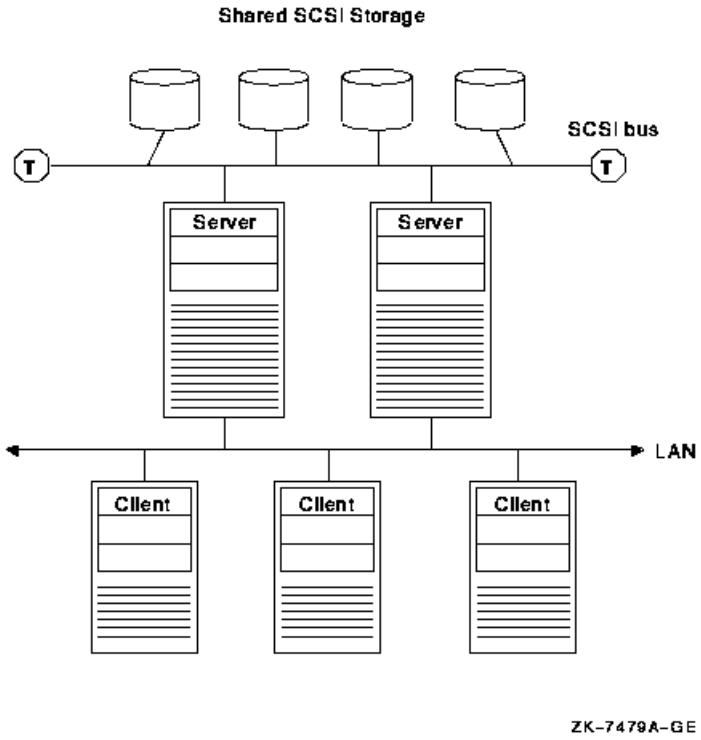
Prior to OpenVMS Version 6.2, OpenVMS Cluster systems provided support for SCSI storage devices connected to a single host using an embedded SCSI adapter, an optional external SCSI adapter, or a special-purpose RAID (redundant arrays of independent disks) controller. Only one host could be connected 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.2shows 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

Requirement	Description
	• 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 adapters attached to a given SCSI interconnect must have the same OpenVMS device name (for example, PKA0), unless port allocation classes are used (see *VSI OpenVMS Cluster Systems Manual*).
	• Each system attached to a SCSI interconnect must have a nonzero node disk allocation class value. These node disk allocation class values may differ as long as either of the following conditions is true:
	  • The SCSI interconnect has a positive, non-zero port allocation class
	  • The only devices attached to the SCSI interconnect are accessed by HSZ70 or HSZ80 controllers that have a non-zero controller allocation class.

Requirement	Description
	If you have multiple SCSI interconnects, you must consider all the SCSI interconnects to determine whether you can chose a different value for the node disk allocation class on each system. Note, also, that the addition of a SCSI device to an existing SCSI interconnect requires a revaluation of whether the node disk allocation classes can still be different. Therefore, it is recommended to use the same node disk allocation class value for all systems attached to the same SCSI interconnect. For more information about allocation classes, see *VSI OpenVMS Cluster Systems Manual*.

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.

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*. 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 Ultra SCSI Configuration Guidelines*.

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 Ultra SCSI 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.

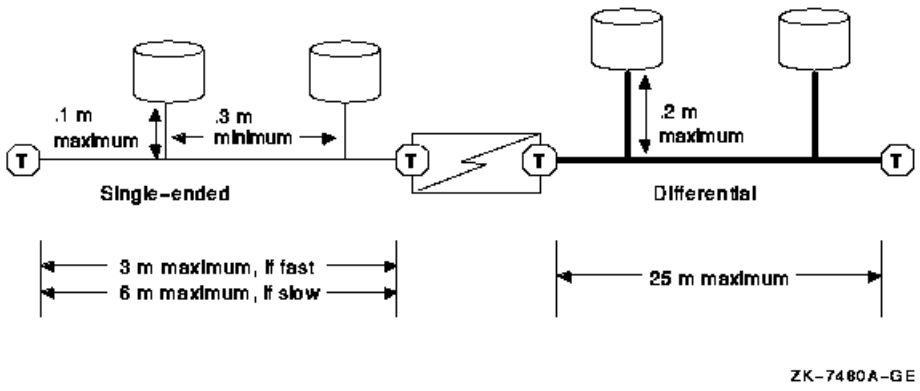
Additionally, the minimum distance between stubs on a single-ended interconnect is .3 m. Refer to Figure A.3 for an example of this configuration.

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* 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 two1-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

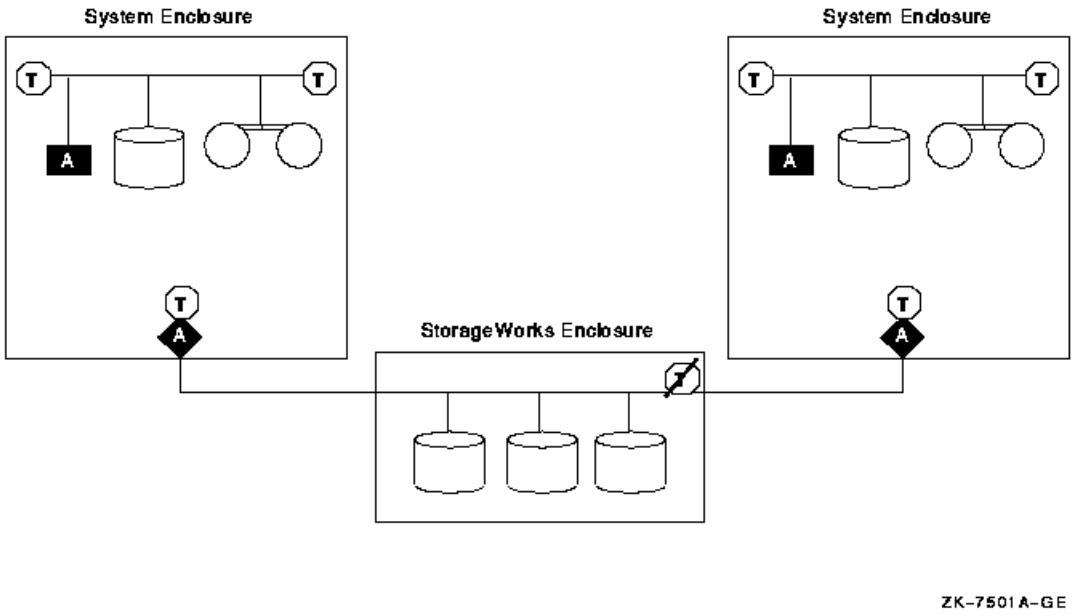
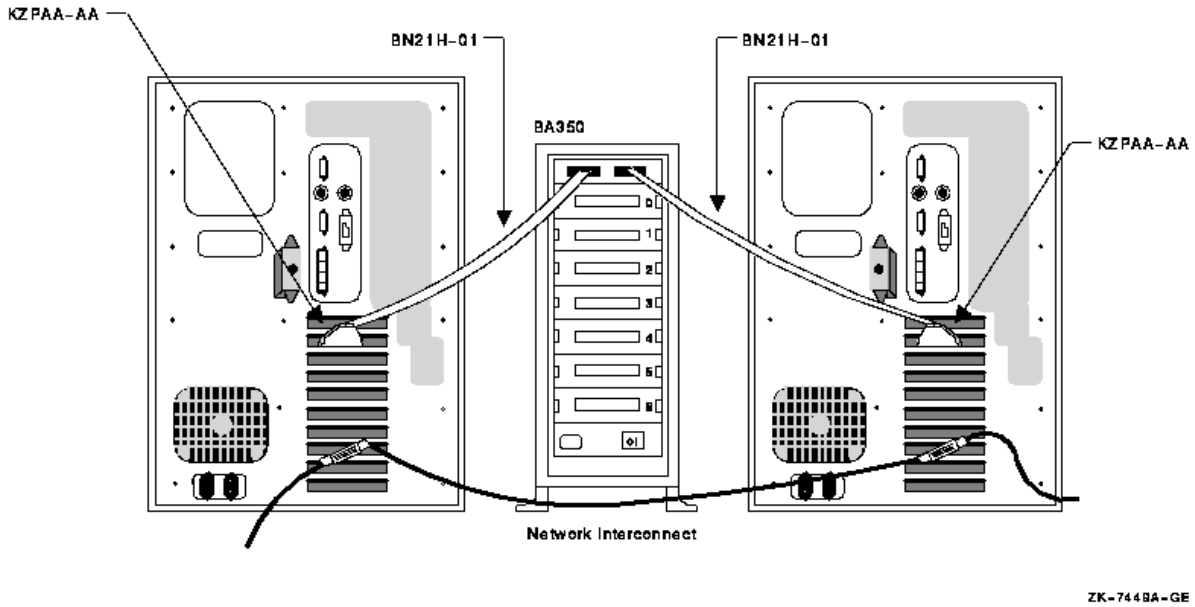


Figure A.5. Sample Configuration: Basic SCSI System Using AlphaServer 1000, KZPAA Adapter, and BA350 Enclosure



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

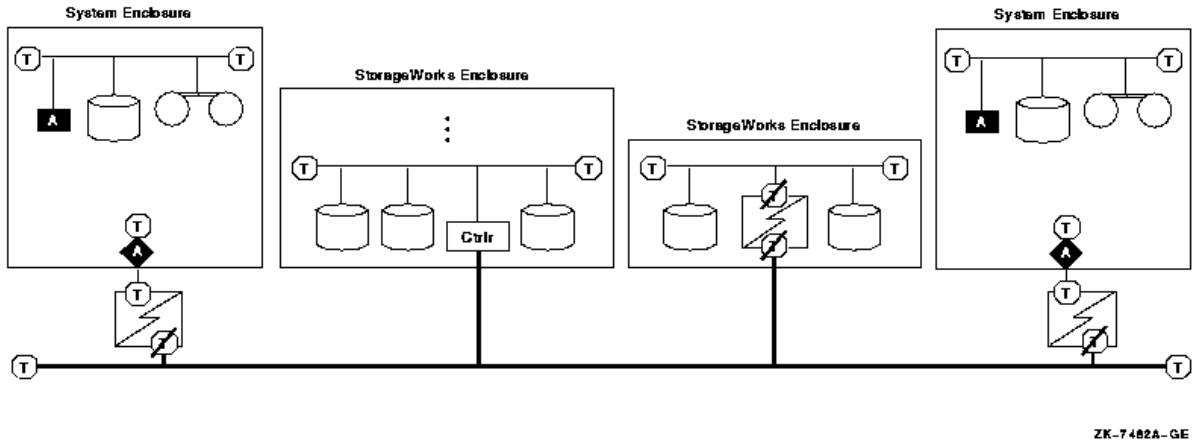


Figure A.7. Sample Configuration: Using DWZZAs to Allow for Increased Separation or More Enclosures

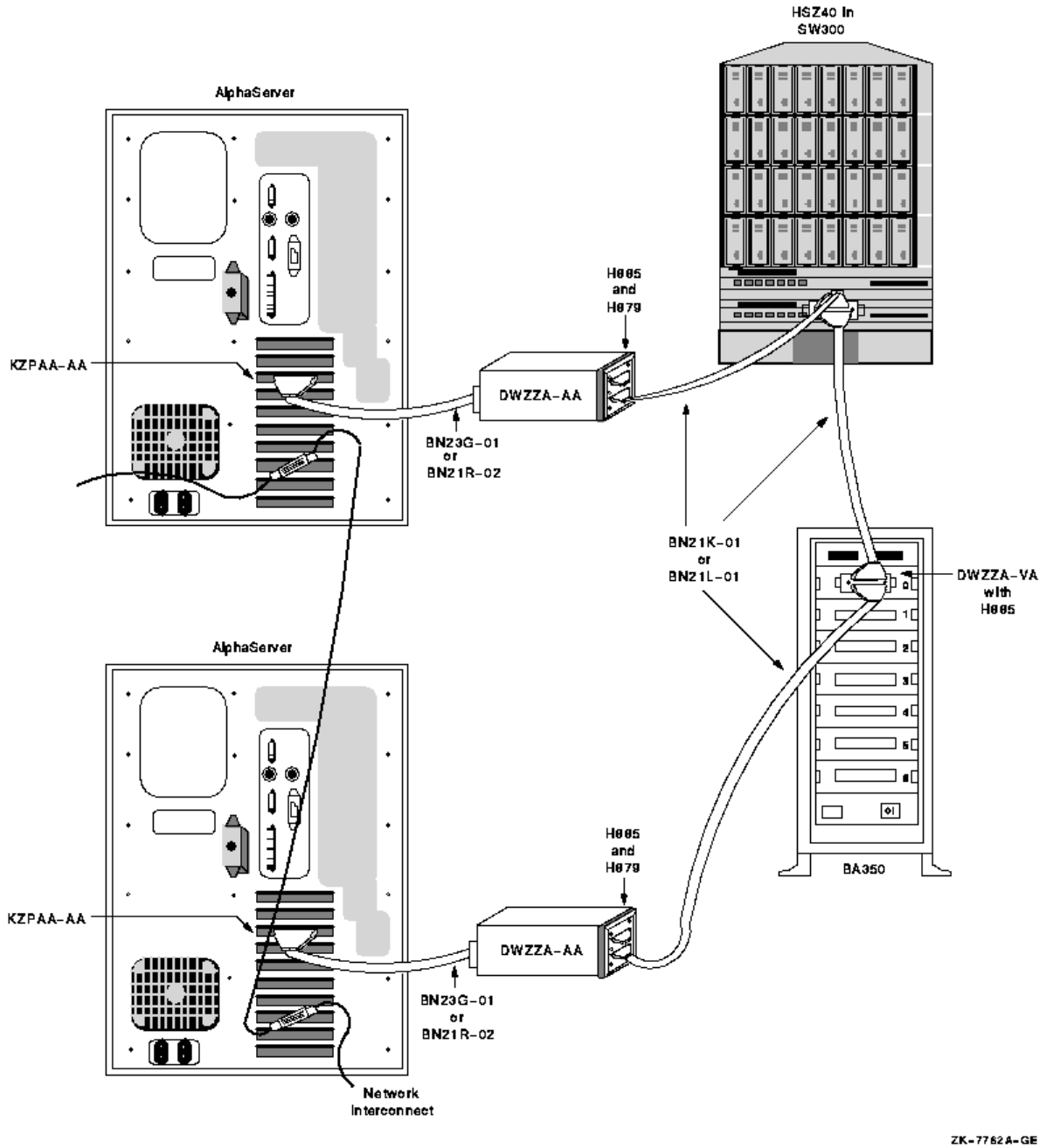
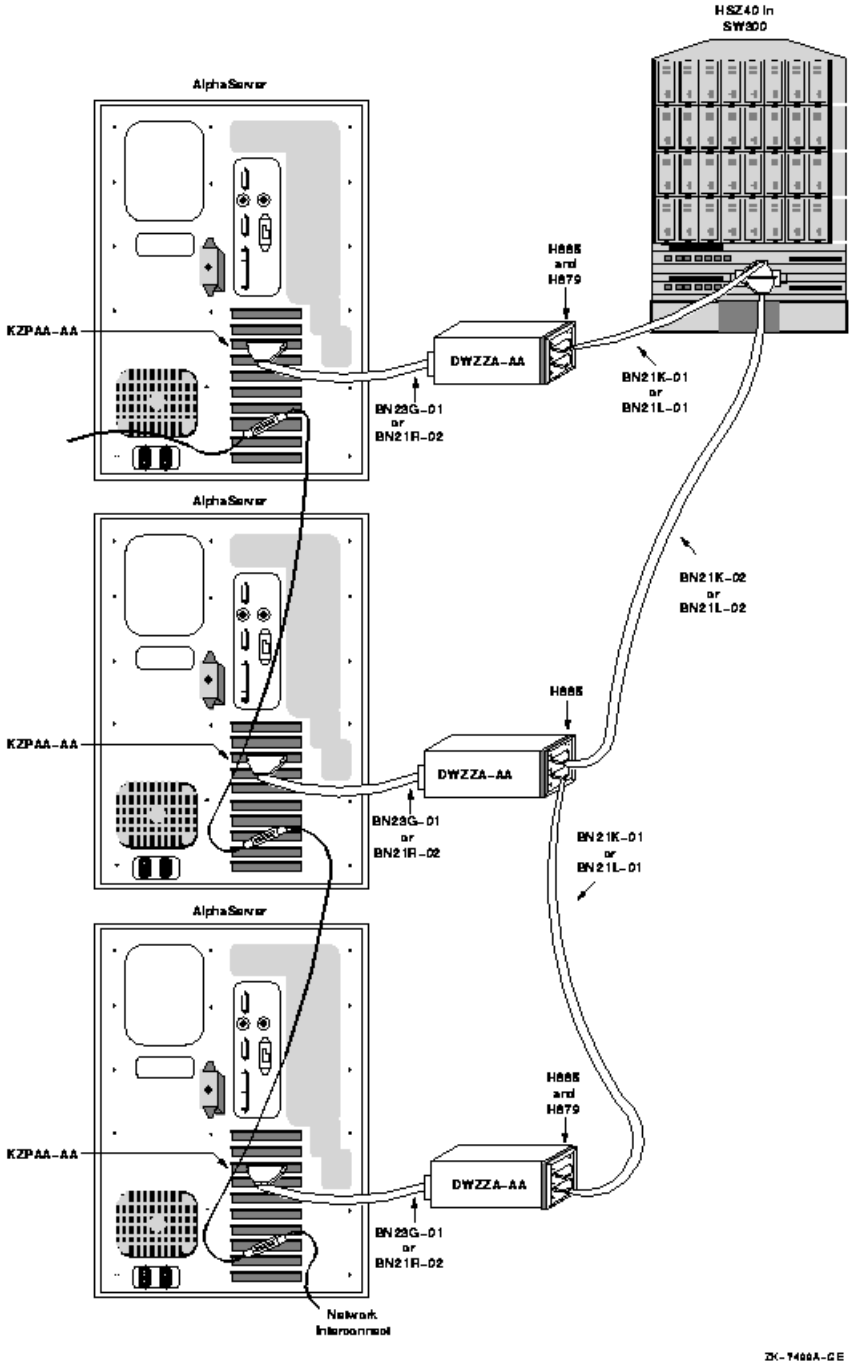


Figure A.8shows how a three-host SCSI OpenVMS Cluster system might be configured.

Figure A.8. Sample Configuration: Three Hosts on a SCSI Bus



A.5.1.3. Building a System That Uses Differential Host Adapters

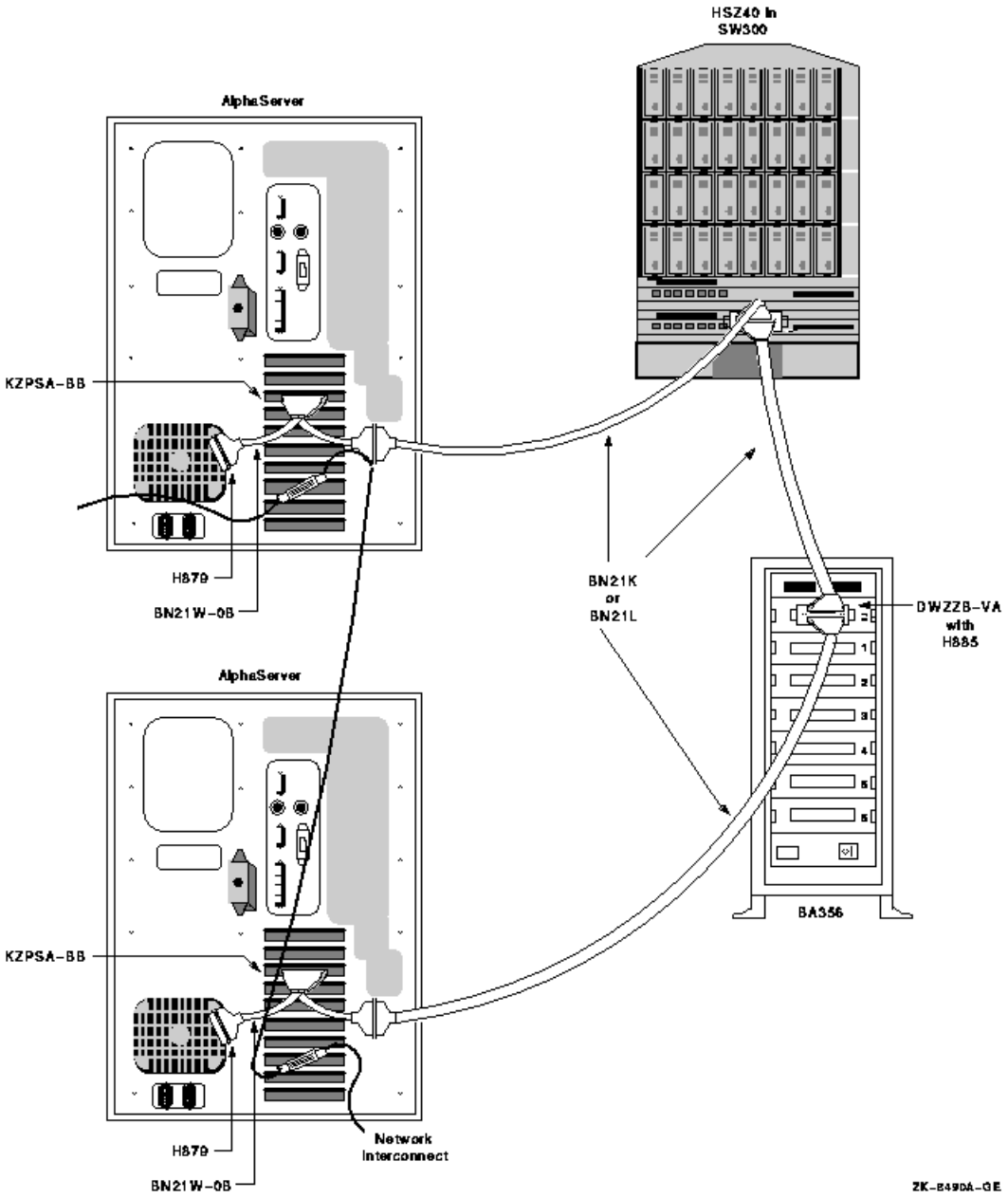
Figure A.9 is a sample configuration with two KZPSA adapters on the same SCSI bus. In this configuration, the SCSI termination has been removed from the KZPSA, and external terminators have been installed on “Y” cables. This allows you to remove the KZPSA adapter from the SCSI bus without rendering the 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.10shows 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

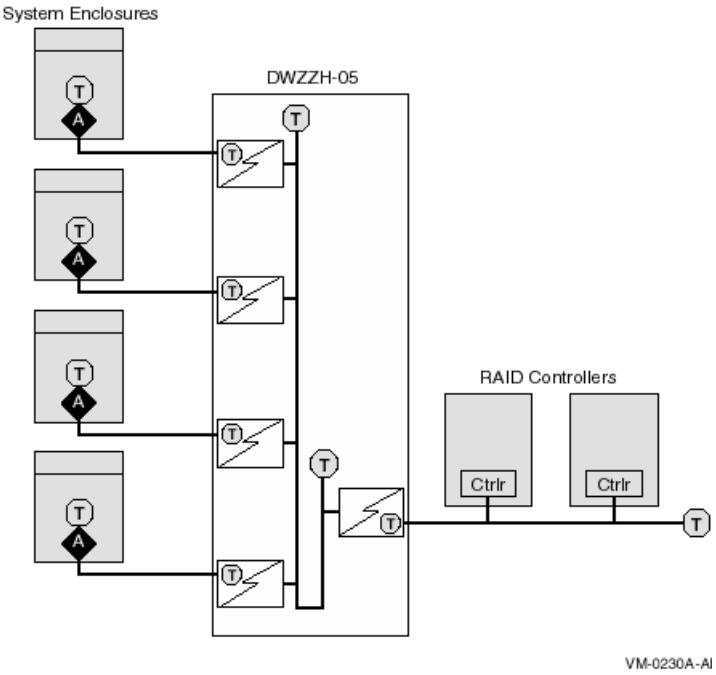
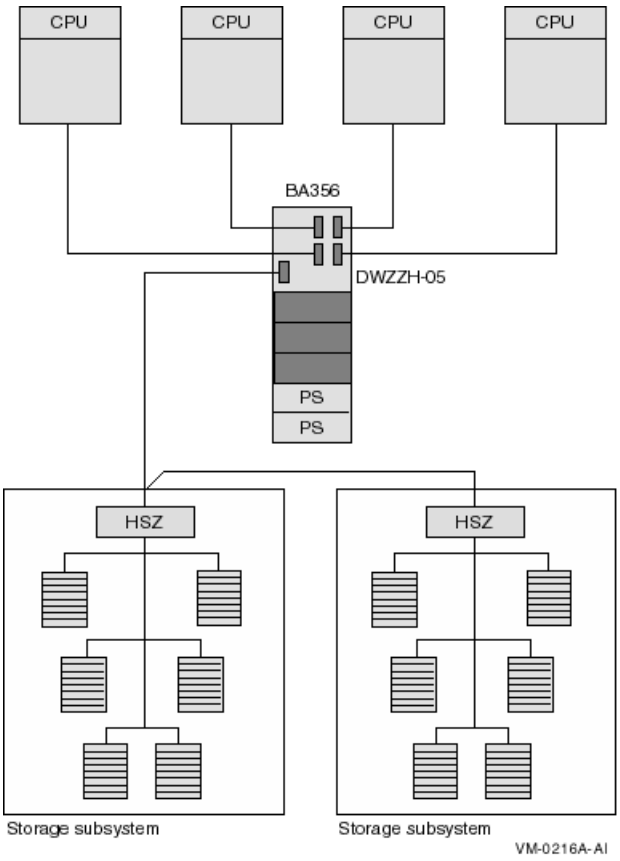


Figure A.11 shows a sample representation of a SCSI hub configuration.

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 *VSI OpenVMS Cluster Systems Manual* 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.

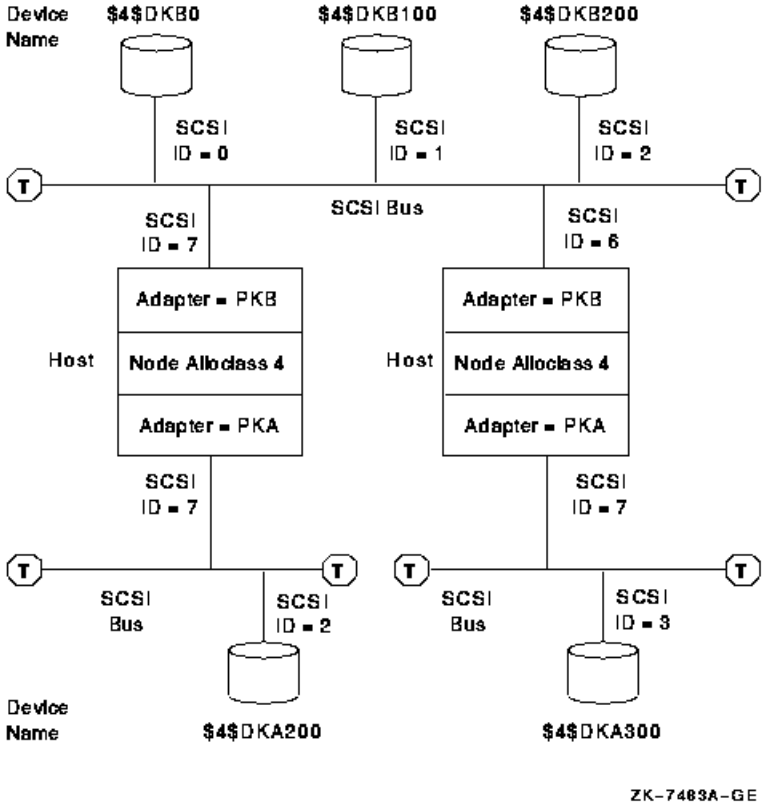
Electrical work should be done by a qualified professional. Section A.7.8includes details of the grounding requirements for SCSI systems.

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.12shows 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 *VSI OpenVMS Cluster Systems Manual*.

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 *VSI OpenVMS Cluster Systems Manual*) 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 *VSI OpenVMS Cluster Systems Manual* 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 *VSI OpenVMS Cluster Systems Manual*.

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 *VSI OpenVMS Cluster Systems Manual*).

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.

Step	Procedure
	• 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.

    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 SCSIBUSY 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 busby 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.4.3for 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.

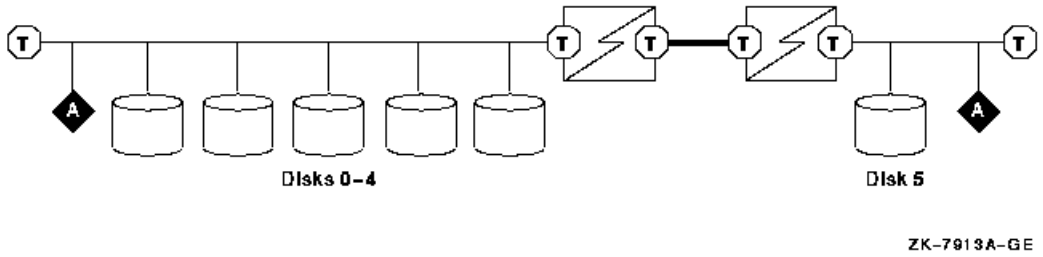
Another method that might provide for more equal servicing of lower and higher ID disks is to set the host IDs to the lowest numbers (0 and 1) rather than the highest. When you use this method, the host cannot gain control of the bus to send new commands as long as any disk, including those

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 ID0. 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:



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

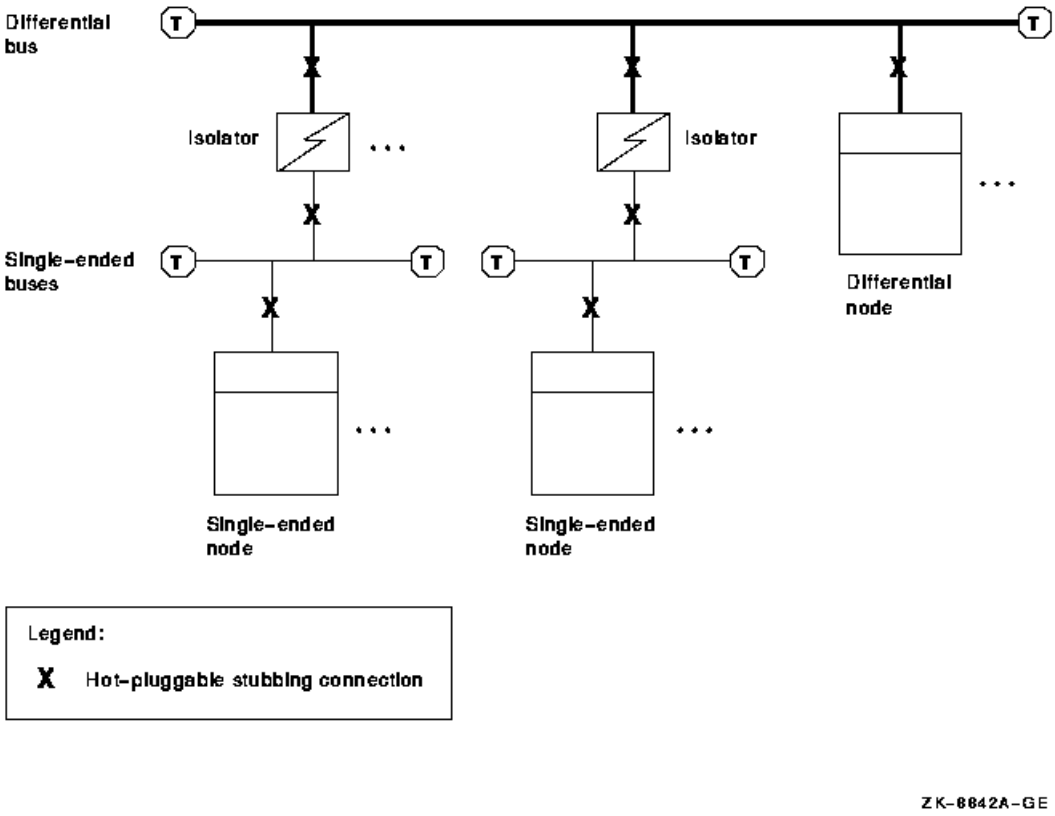
Follow these rules when planning for and performing hot plugging:

- 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. Referring to this draft standard is necessary because the SCSI-2 standard does not adequately specify the requirements for hot plugging. 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.

Figure A.13. SCSI Bus Topology



- 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.
- 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. OpenVMS will eventually detect a hung bus and reset it, but this problem may first temporarily disrupt OpenVMS Cluster operations.

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.

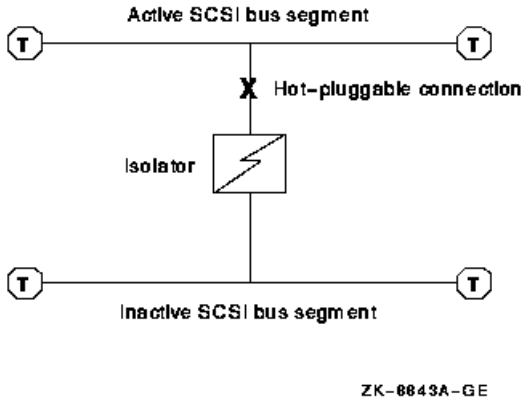
  The OpenVMS Cluster system must be reconfigured to remove a device as a quorum disk before that device is removed from the bus. The procedure for accomplishing this is described in *VSI OpenVMS Cluster Systems Manual*.

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



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.
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:

Step	Description
	• 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, VSI 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, VSI 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.

Appendix 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:

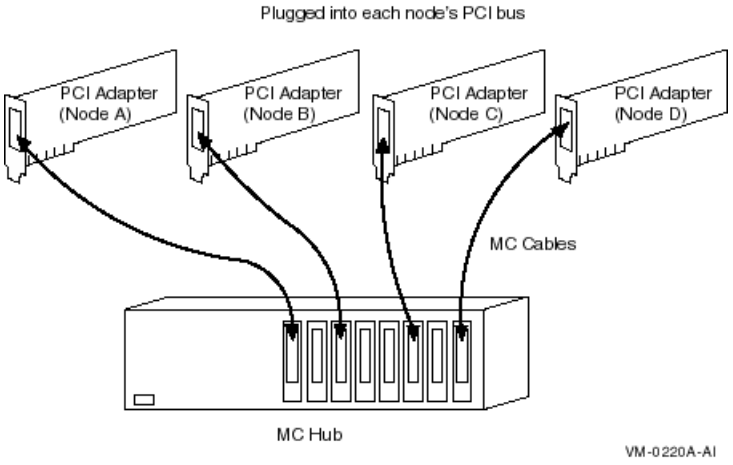
- Support for a new adapter (CCMAB-AA) and new hubs (CCMHB-AA and CCMHB-BA)
- Support for simultaneous communication between four sender-receiver pairs
- Support for longer cables for a radial topology up to 3 km

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).

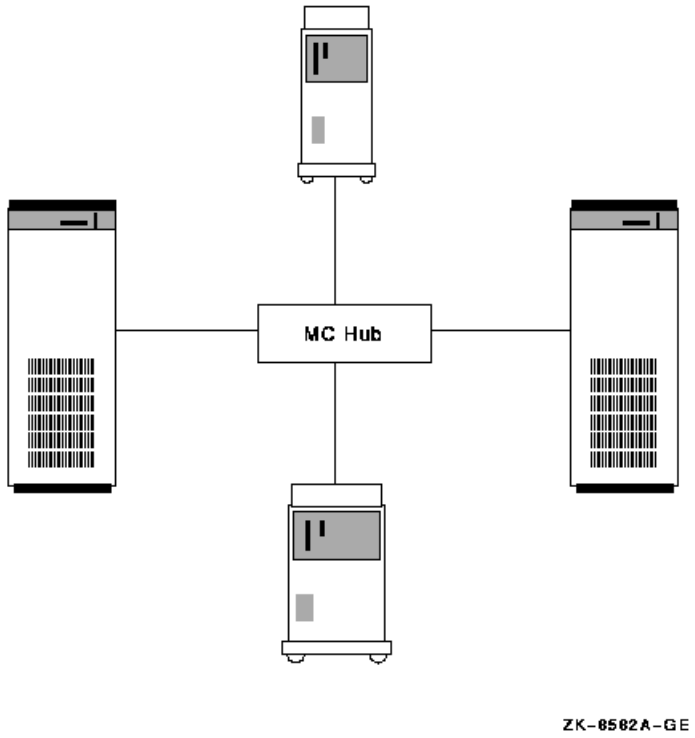
Figure B.1. MEMORY CHANNEL Hardware Components



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.

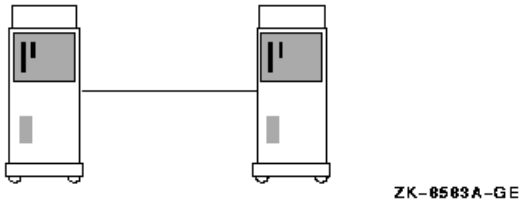
Figure B.2 shows an example of a four-node MEMORY CHANNEL cluster with a hub at its center.

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.

VSI 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 Ethernet usually provide acceptable interconnect performance in the event of MEMORY CHANNEL failure. See *VSI OpenVMS Cluster Systems Manual* 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, MEMORYCHANNEL may be unable to complete initialization. If this happens, the console displays the following message repeatedly:

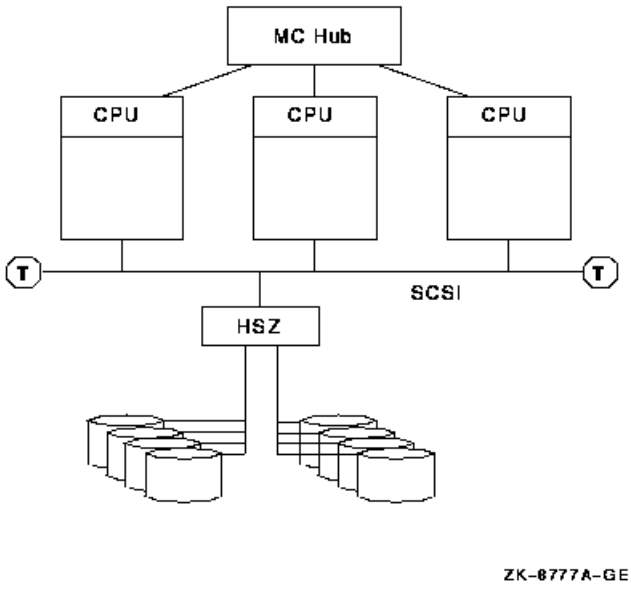
```
Hub timeout - reinitializing adapter
```

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.

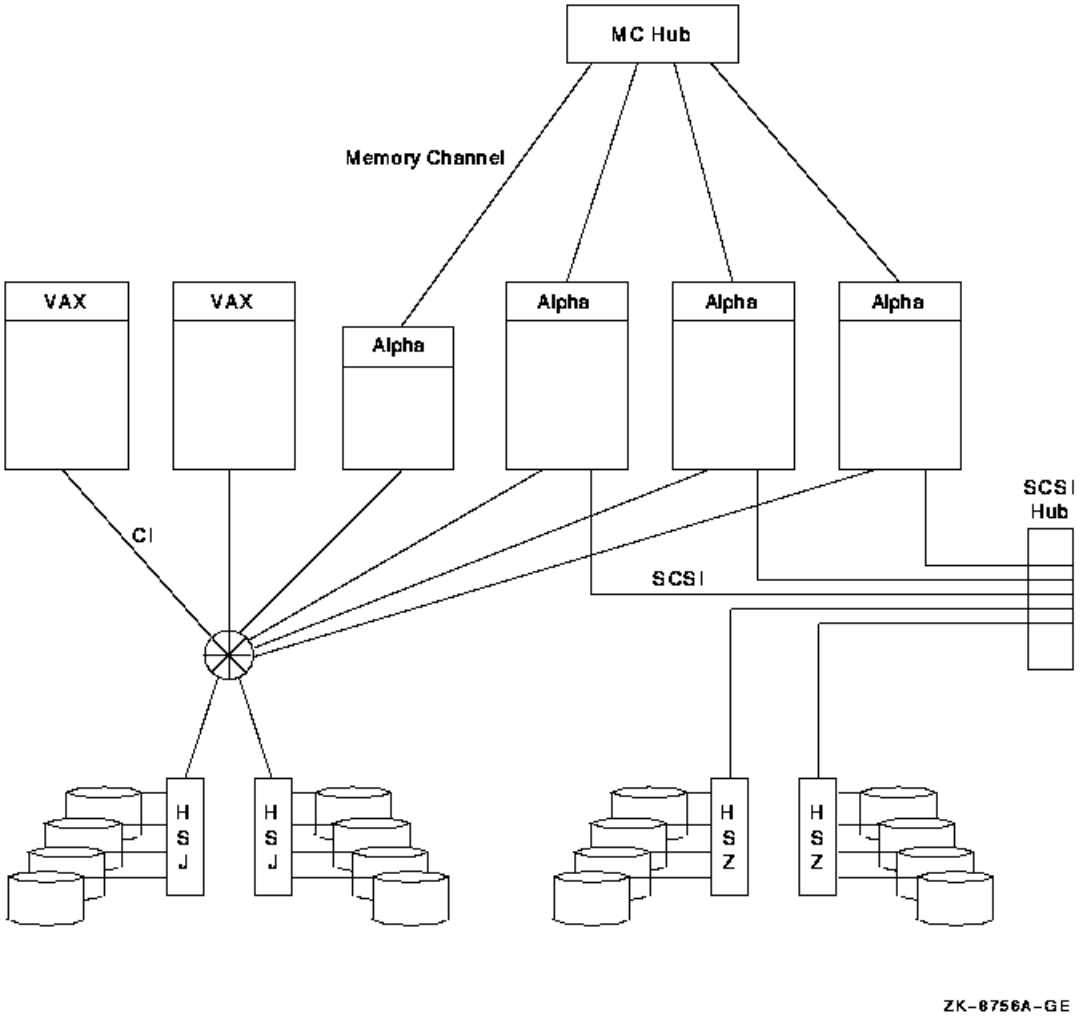
Figure B.4. MEMORY CHANNEL- and SCSI-Based Cluster



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.

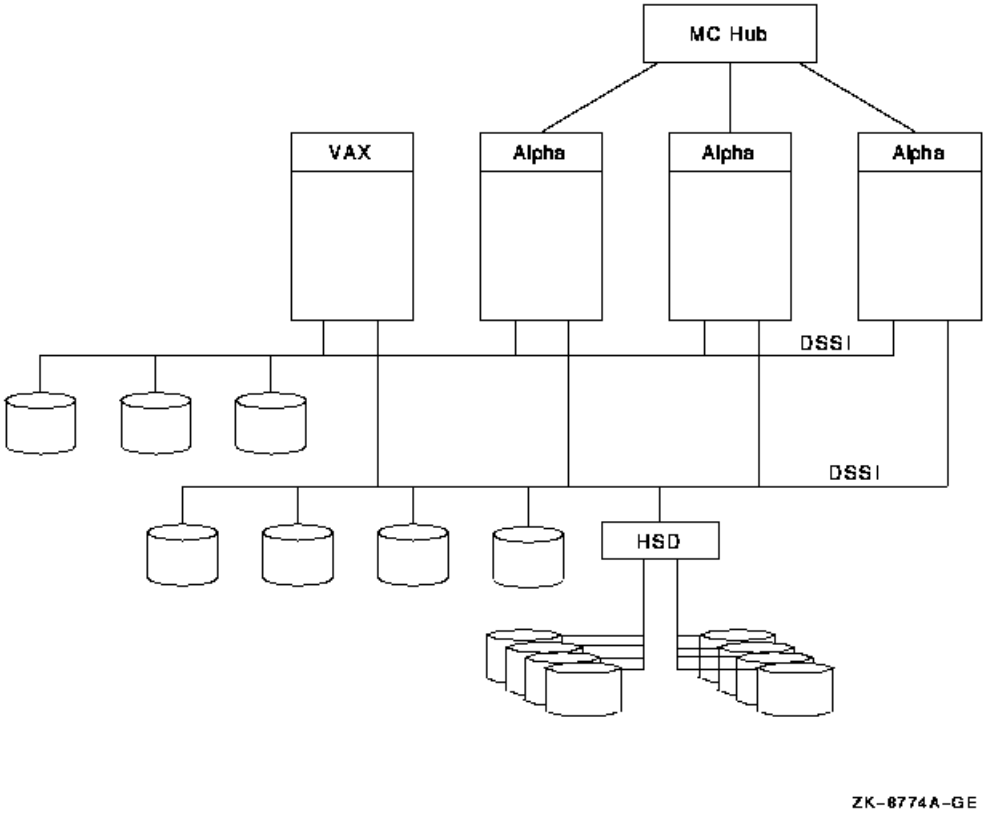
Figure B.5. MEMORY CHANNEL CI- and SCSI-Based Cluster



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.5as 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 MEMORYCHANNEL 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.
Cables	MEMORY CHANNEL supports the following cables: • Copper cables up to a 10-m (32.8 ft) radial topology • Fiber-optic cables 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:

Requirement	Description
	• 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 MEMORYCHANNEL 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 MEMORYCHANNEL 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

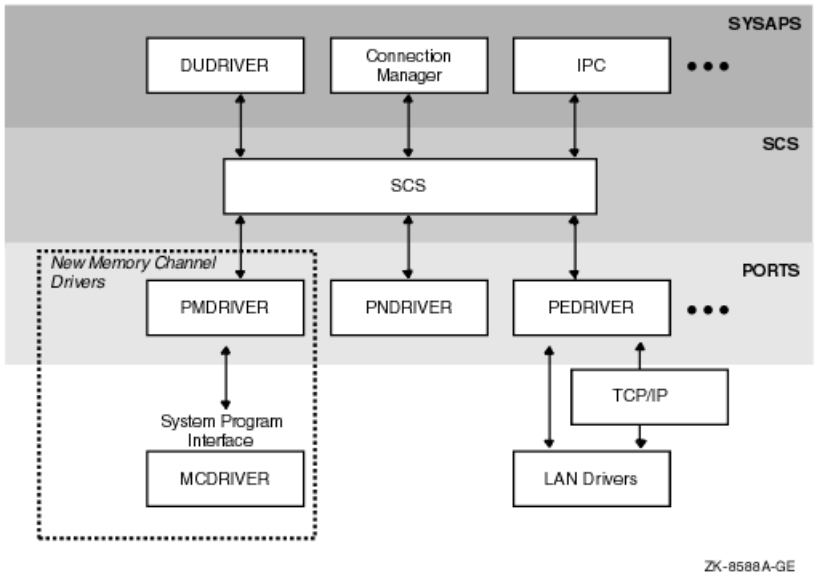
Characteristics	SMP	MEMORY CHANNEL	Standard Networking
Bandwidth (MB/s)	1000+	100+	10+
Latency (ms/simplest message)	0.5	Less than 5	About 300
Overhead (ms/simplest message)	0.5	Less than 5	About 250
Hardware communication model	Shared memory	Memory-mapped	Message passing
Hardware communication primitive	Store to memory	Store to memory	Network packet

Characteristics	SMP	MEMORY CHANNEL	Standard Networking
Hardware support for broadcast	n/a	Yes	Sometimes
Hardware support for synchronization	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 al 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

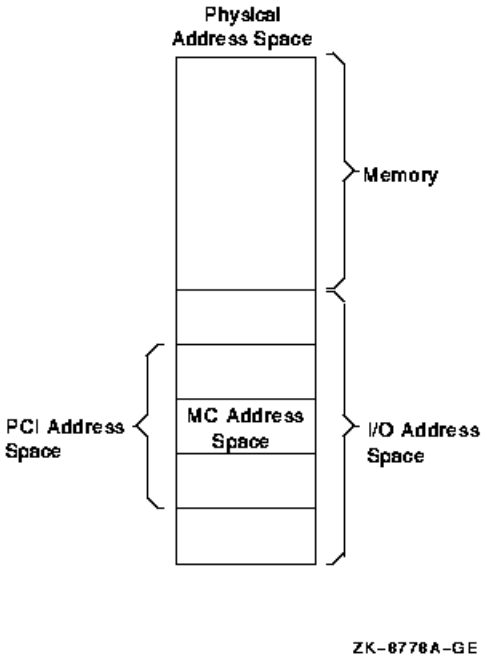
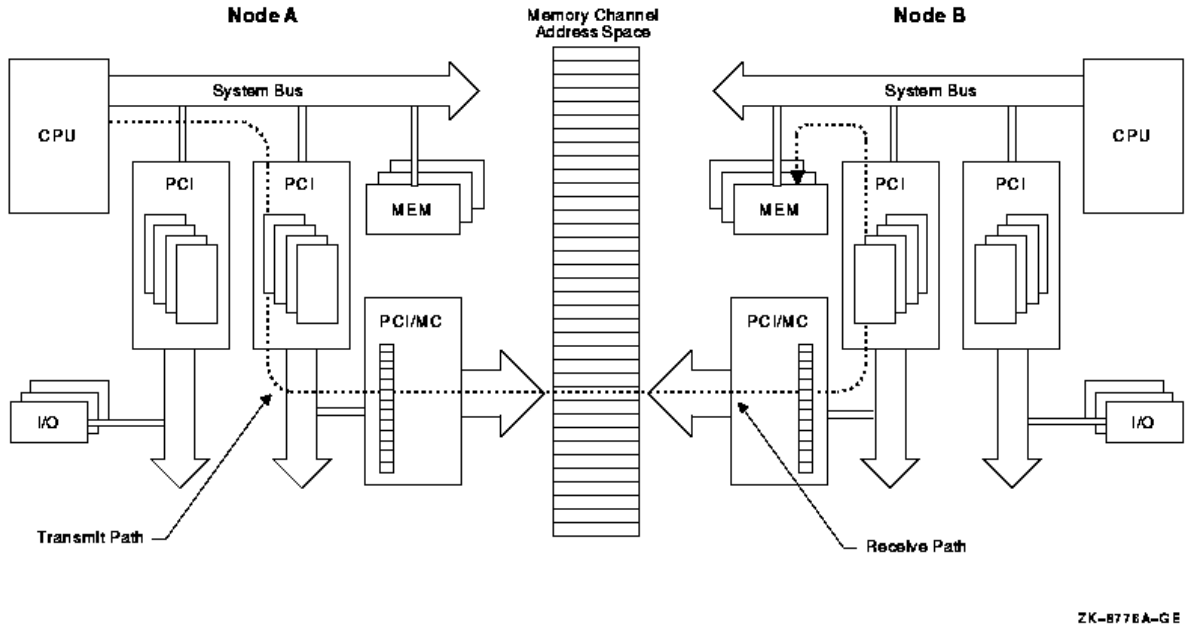


Figure B.8 shows the typical address space of a system, divided into physical memory and I/O address space. Within the PCI I/O address space, MEMORY CHANNEL consumes 128 to 512 MB of address space. Therefore, the MEMORY CHANNEL PCI adapter can be addressed within this space, and the CPU can write data 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.9shows the internal bus architecture of two nodes, node A and node B.

Figure B.9. MEMORY CHANNEL Bus Architecture



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.3explains 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 to1. 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 *VSI OpenVMS Cluster Systems Manual* manual.

For more detailed information about setting up the MEMORY CHANNEL hub, link cables, and PCI adapters, see the *MEMORYCHANNEL User's Guide*.

Appendix C. 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 VMS cluster Systems* addendum manual.

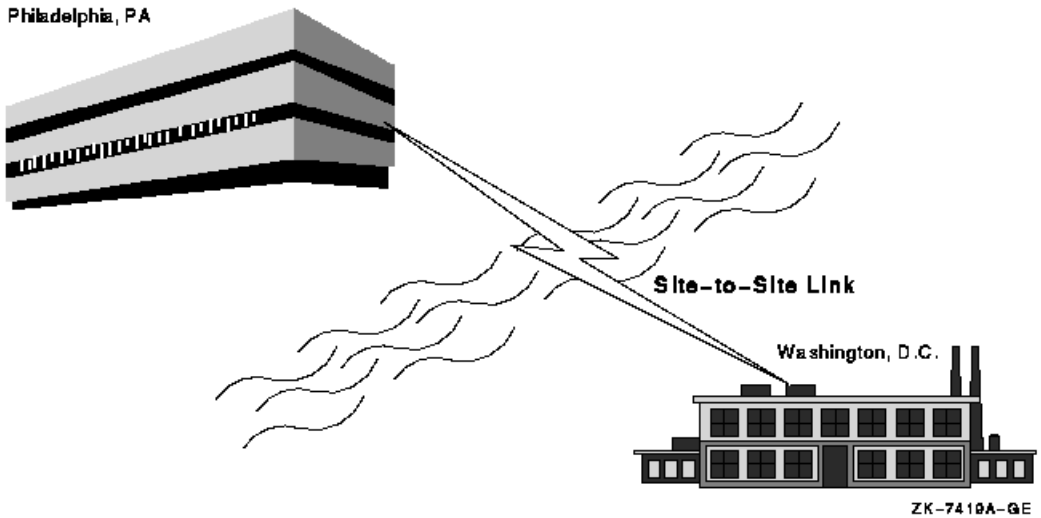
C.1. What is a Multiple-Site OpenVMS Cluster System?

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 500 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 C.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 C.1. Site-to-Site Link Between Philadelphia and Washington



C.1.1. ATM, DS3, FDDI, and [D]WDM Intersite Links

The following link technologies between sites are approved for OpenVMS VAX and OpenVMS Alpha systems:

- Asynchronous transfer mode (ATM)
- DS3
- FDDI
- [D]WDM

High-performance local area network (LAN) technology combined with the ATM, DS3, FDDI, and [D]WDM 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 C.4 describes OpenVMS Cluster systems and the WAN communications services in more detail.

C.1.2. Benefits of Multiple-Site OpenVMS Cluster Systems

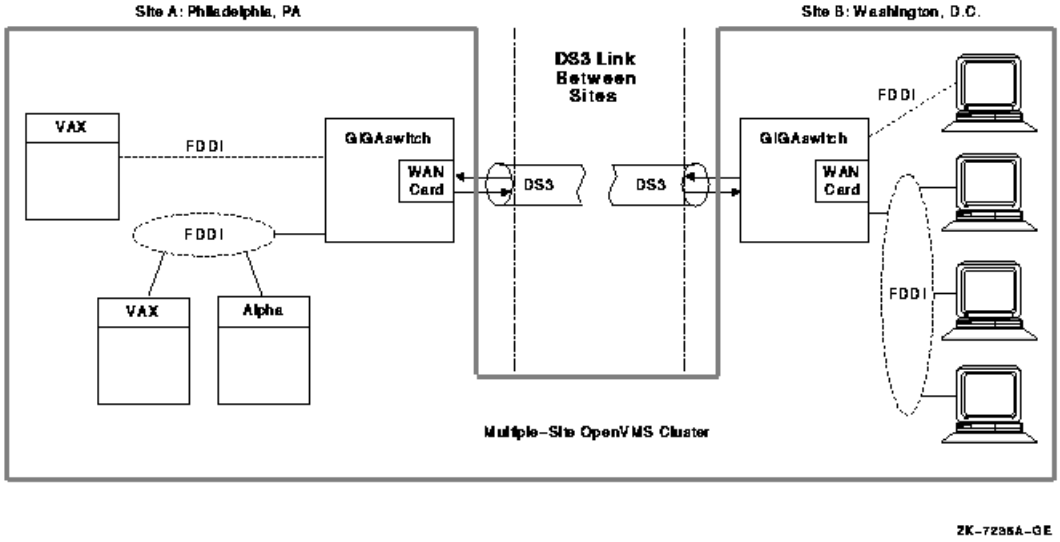
Some of the benefits you can realize with a multiple-site OpenVMS Cluster system include the following:

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 C.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 phototype setters.
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.

Benefit	Description
	• 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.

Figure C.2 shows an OpenVMS Cluster system with satellites accessible from a remote site.

Figure C.2. Multiple-Site OpenVMS Cluster Configuration with Remote Satellites



C.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:

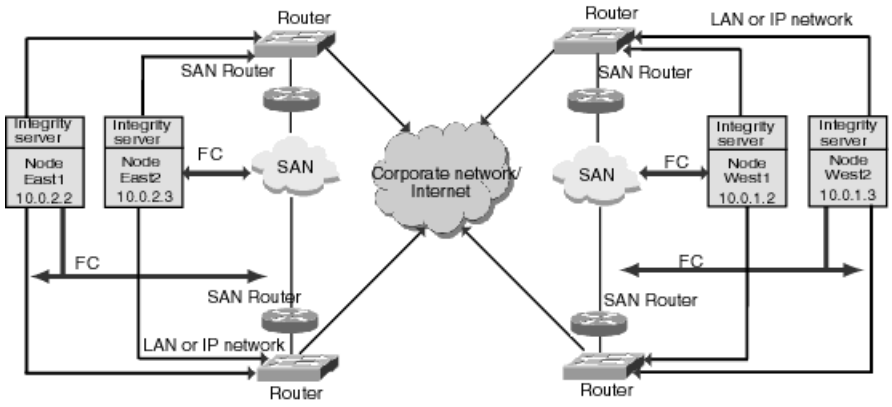
- OpenVMS Cluster Software *Software Product Description*
- Chapter 8 of this manual

Some configuration guidelines are unique to multiple-site OpenVMS Clusters;these guidelines are described in Section C.4.4.

C.2. Using Cluster over IP to Configure Multiple-Site OpenVMS Cluster Systems

When nodes are located in multiple sites or multiple LANs, IP is preferred for cluster communication. Using cluster over IP, you can connect sites with an intersite distance of up to 500 miles. Figure C.3 illustrates the multiple-site OpenVMS cluster using IP interconnect with storage configured.

Figure C.3. Multiple-Site OpenVMS Cluster Configuration with Cluster over IP



Node East1, East2, West1, and West2 can be part of the same or different LAN. Cluster traffic is routable. Nodes in the east use cluster over IP to communicate with nodes in the west, which are at geographically distant sites. Node East1 and Node West1 forms a Virtual Circuit (VC). The VC consists of IP channels for SCS traffic. However, nodes in the east use LAN between themselves for cluster communication. Node East1 will form a virtual circuit using a LAN channel with Node East2.

C.3. 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), the cable route distance between sites.

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.
- 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.

Additional OpenVMS Cluster configuration guidelines and system management information can be found in this manual and in *VSI OpenVMS Cluster Systems Manual*.

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.

C.4. 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:

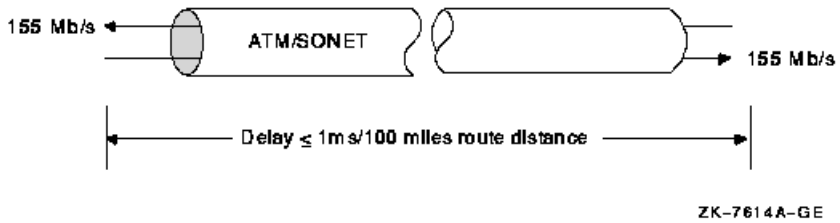
Service	Approved Versions of OpenVMS
ATM	OpenVMS Version 6.2 or later
DS3	OpenVMS Version 6.1 or later

The following sections describe the ATM and DS3 communication services and how to configure these services into multiple-site OpenVMS Cluster systems.

C.4.1. 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 C.4). 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 C.4. ATM/SONET OC-3 Service

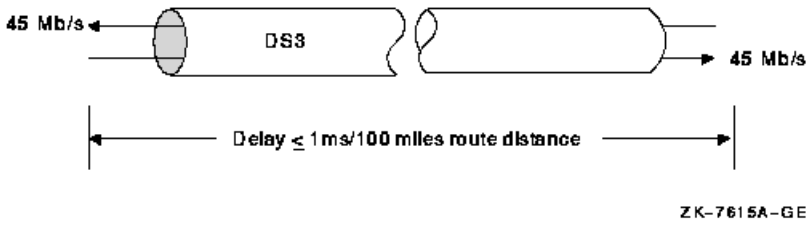


To transmit data, ATM frames (packets) are broken into **cells** for transmission by the ATM service. Each cell has 53 bytes, of which 5 bytes are reserved for header information and48 bytes are available for data. At the destination of the transmission, the cells are reassembled into ATM frames. The 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.

C.4.2. The DS3 Communications Service (T3 Communications Service)

The DS3 communications service provides full-duplex communications as shown in Figure C.5. 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.

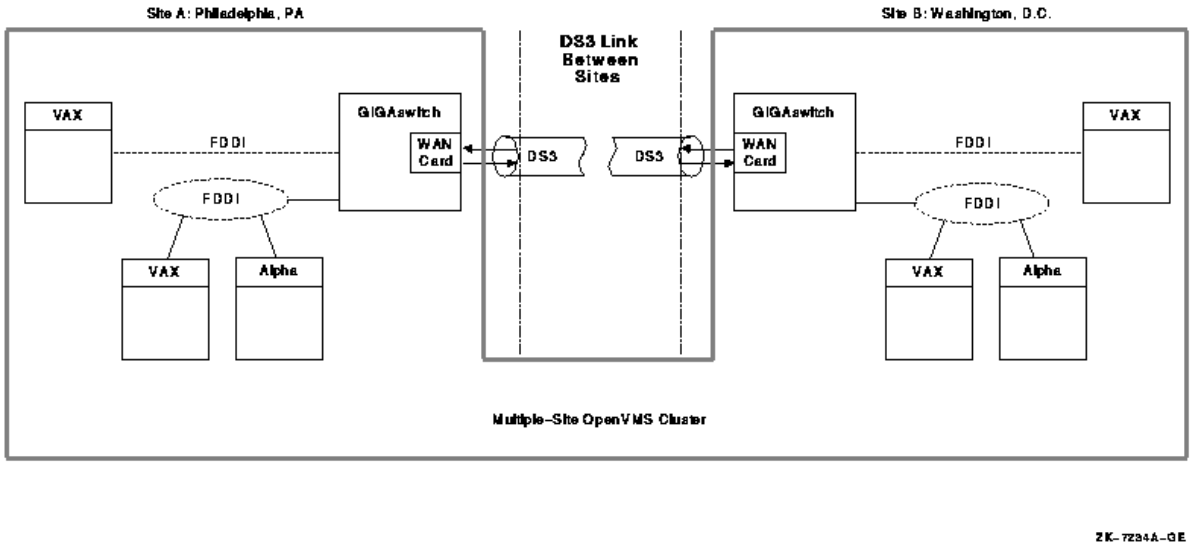
Figure C.5. DS3 Service



C.4.3. 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 C.6. 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 C.6. Multiple-Site OpenVMS Cluster Configuration Connected by DS3



In Figure C.6, 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.

It is recommended 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.

C.4.4. 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.

C.4.4.1. Requirements

To be a configuration approved by VSI, 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.
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 C.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 C.1.3.

C.4.4.2. 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 XX.XX.*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, 155Mb/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 C.1.

Table C.1. DS3 Protocol Options

Protocol Option	Link Capacity
ATM[1] AAL–5[2] mode with PLCP[3]disabled.	39 Mb/s

Protocol Option	Link Capacity
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, VSI 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, VSI 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. VSI 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 C.4.6). 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 C.4.5).

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 C.4.6 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.

C.4.5. 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.

C.4.6. 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 C.2 describes terminology that will help you understand the Bellcore and OpenVMS Cluster requirements and goals used in Table C.3.

Use the Bellcore and OpenVMS Cluster requirements for ATM/SONET - OC3 and DS3 service error performance (quality) specified in Table C.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 C.3.

Table C.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 VSI to approve a configuration, parameters must meet or exceed the values shown in the OpenVMS Cluster Requirements column in Table C.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 through put. OpenVMS	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

Specification	Requirements	Goals
	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]	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 C.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:				

Table Key

- **Availability—The long-term fraction or percentage of time that at rans mission 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}."**

Parameter	Bellcore Requirement	Bellcore Goal	OpenVMS Cluster Requirement[1]	OpenVMS Cluster Goal[1]	Units
	<864	<345	<864	<24	ES per 24-hr period
Burst errored seconds (BES)[2]	≤4	–	≤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	≤97 @ 250 miles, linearly decreasing to 24 @ ≤25 miles	None	Bellcore Goal	Min/yr
SONET channel unavailability	None	≤105 @ 250 miles, linearly decreasing	None	Bellcore Goal	Min/yr

Table Key

- **Availability—The long-term fraction or percentage of time that at rans mission 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}."**

Parameter	Bellcore Requirement	Bellcore Goal	OpenVMS Cluster Requirement[1]	OpenVMS Cluster Goal[1]	Units
		to 21 @ ≤50 miles			
Channel-unavailable event[4]	None	None	None	1 to 2	Events/year

Table Key

- **Availability—The long-term fraction or percentage of time that at rans mission 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}."**

[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.

C.5. 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.

C.5.1. 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)
  - 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.

C.5.2. 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 *VSI OpenVMS Cluster Systems Manual*.
- 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.
- You can also use Availability Manager, HP ECP (CP/Collect and CP/Analyze), and Unicenter Performance Management for OpenVMS (formerly Polycenter Performance Solution Data Collector and Performance Analyzer, formerly SPM and VPA) to monitor the cluster performance.