Data General

DG/UX Technical Brief

January 27, 1993

**Information About AViiON® Systems
from Data General's UNIX® Development Group**

In This Issue:

Machine Initiated Failover in the DG/UX™ 5.4 R2.01 Operating System

Contents

Data General's introduction of dual-initiator configuration disk subsystems such as the CLARiiON™ Disk Array Storage System has been well received by customers. Customers need and appreciate the ability to transfer physical disks to a secondary system when a primary system fails.

To help customers better use the dual-initiator configuration disk subsystem features, Data General developed and released Operator Initiated Failover software (OIF) in the DG/UX 5.4 R2.00 operating system.

The OIF software provided customers with a mechanism to transfer physical disks when a system failed. The detection of a system failure was left up to the system administrator. When the administrator detected a system failure, he or she performed the appropriate action on the secondary system to transfer the disks and restart the application.

To eliminate the need for administrator-detection of a system failure, Data General has developed *Machine Initiated Failover* software. This technical brief describes how Machine Initiated Failover (MIF) software automates the detection and response to failover situations. The following topics are discussed:

- ❑ The hardware and software requirements for using MIF software
- ❑ An overview of MIF software
- ❑ An example of Machine Initiated Failover

Produced on a Data General
AViiON AV4000 with FrameMaker® 3.1X.

Terminology

Here are some terms that are used in this technical brief.

Array

A collection of one or more of disk modules and one or more SCSI busses that participate in a Redundant Array of Inexpensive Disks (RAID) redundancy scheme.

Disk module (or spindle)

A self contained disk-drive unit—as opposed to the generic term "disk," which could refer to a logical disk or a physical disk.

Dual-initiator configuration

A configuration in which a physical disk is connected to a SCSI bus that can have two initiators.

Failover

The transfer of one or more dual-initiator configuration disk modules and zero or more applications from one machine to another machine sharing the dual-initiator configuration disk module.

Initiator

A SCSI device that has the capability to initiate operations with a SCSI target, such as a host bus adapter.

Logical disk

A software abstraction that enables the DG/UX operating system to manage files the same way, regardless of how the files are stored physically. Logical disks are built on physical disks and a logical disk can use pieces from as many as 32 physical disks.

Physical disk

What the operating system recognizes as a single disk. A physical disk can be a single disk module or a group of disk modules in a CLARiiON Disk Array Storage System.

RAID

Redundant Array of Inexpensive Disks. RAID technology provides redundant disk resources. RAID level 5 (RAID 5) distributes user and parity data among all of the disk modules in an array.

Target

A SCSI device, typically a disk or tape drive, which can be selected as the target of a given SCSI bus operation.

Hardware/Software Requirements

Here are the requirements for using the MIF software:

- ❑ DG/UX 5.4 R2.01 or later revisions
- ❑ A communication link between each host in the dual-initiator configuration
- ❑ A Dual-initiator configuration CLARiiON disk array

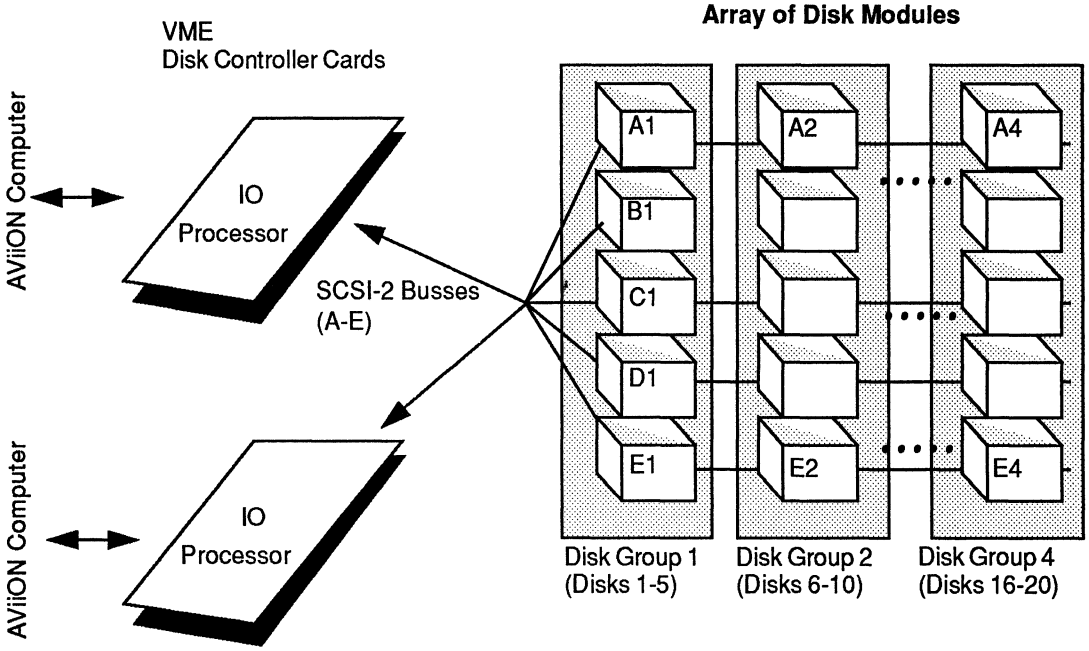

Figure 1 A High Availability Disk Dual-Initiator Configuration

The MIF software will be released in DG/UX 5.4 R2.01 to support monitoring of a remote host via redundant LANs. The software will also be released in DG/UX 5.4 R3.00 and will be capable of monitoring via alternate communication media, and detecting system "hangs."

MIF Operation

You set up and run the MIF software on the backup system. After you have set up the failover databases for OIF you should set up the failover databases for MIF. Unlike OIF, MIF does not require a synchronization or exchange of failover database information. The information is only required on the machine that is doing the monitoring.

Machine Initiated Failover

The MIF software is designed to automatically transfer physical disks and restart applications when a system fails. The MIF software provides a monitoring process that detects a system failure and initiates the OIF functionality. The MIF monitor process can use multiple communication paths to ensure that the monitored host has actually failed before taking action.

Figure 2 shows a configuration consisting of two LAN communication paths.

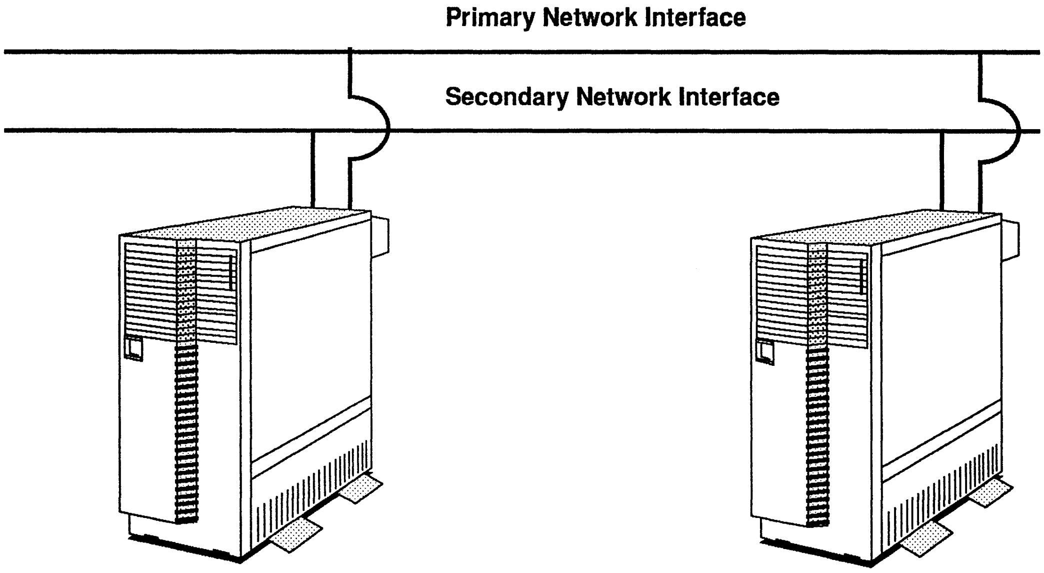

Figure 2 MIF Configuration With Multiple LANs

MIF Commands

The MIF software uses two failover databases, a daemon monitor process, and a sysadm interface for ease of use. The software also provides two new administrative commands:

- **admfailoveraltcommpath**—used to maintain the failover altcommpath database and check alternate communication path accessibility
- **failovermon**—used to maintain the failover-monitors database and perform failover monitoring

MIF Operation

You set up and run the MIF software on the backup system. After you have set up the failover databases for OIF you should set up the failover databases for MIF. Unlike OIF, MIF does not require a synchronization or exchange of failover database information. The information is required only on the machine that is doing the monitoring.

Setting Up the Communication Path Database

You should use sysadm to set up the failover **altcommpath** database with the alternate communication paths (such as secondary LANs) that you want the failovermon monitor to use. The parameters required when executing the Device ➤ Disk ➤ Failover ➤ Alternate Paths ➤ Add operation are:

❑ The primary network interface name of the host to monitor. This name is used to look up alternate communication path database entries.

❑ The secondary network interface name of the host to monitor. This name is used to establish communication to the host via the secondary network interface.

Setting Up the Failover Monitor Process

Once the alternate communication paths have been added and checked, you should add and start the failovermon monitor. The monitor uses several parameters that you can configure. These parameters enable you to configure the monitor to your level of comfort. The parameters required when executing the Device ➤ Disk ➤ Failover ➤ Monitors ➤ Add operation are:

❑ The primary network interface name of the host to monitor. This name is used to communicate via the primary network and as a key to look up alternate communication paths to this host.

❑ The time interval in seconds that the monitor will sleep between message cycles. Lower interval values will make the monitor a more active process on the system. This may affect CPU usage and LAN performance. The default is zero.

❑ The number of times to retry a message cycle before declaring the host to be failed. On systems that have only one LAN or experience sporadic network outages you will want to configure this to a non-zero value. The default is zero.

❑ The user-defined action script that will execute when the monitored host is declared failed. This script will most likely contain an **admfailoverdisk(1M)** command to take control of the physical disks and restart the applications. Since this command line will also include the trespass option, you should first verify that the disks are not registered on the system before taking control of them. The default is **/etc/failover/failovermon_lost_pulse**, which can be modified but contains no executable statements.

- ❑ The user-defined action script that will execute when the monitored host is reachable again. This script can be used to gracefully shut down the application and return the physical disks to the primary system. The default is **/etc/failover/failovermon_regain_pulse**, which can be modified but contains no executable statements.
- ❑ Optionally, a flag indicating whether or not the monitor should be started when the system is rebooted. The rc.failover script, which is executed at run level changes, will start any monitors that have this flag set when the system goes to run level 3. The default is to not start the monitor on reboot.
- ❑ Optionally, a flag indicating whether or not the monitor should be started at the completion of this operation. The default is to not start the monitor when the operation completes.

Running the Failover Monitor Process

When you start a failovermon monitor, you are creating a daemon process that will continue to run until you stop it or the system comes down. After the initial lookup and validation of information, the failovermon process creates a child process that performs the tasks shown in Figure 3.

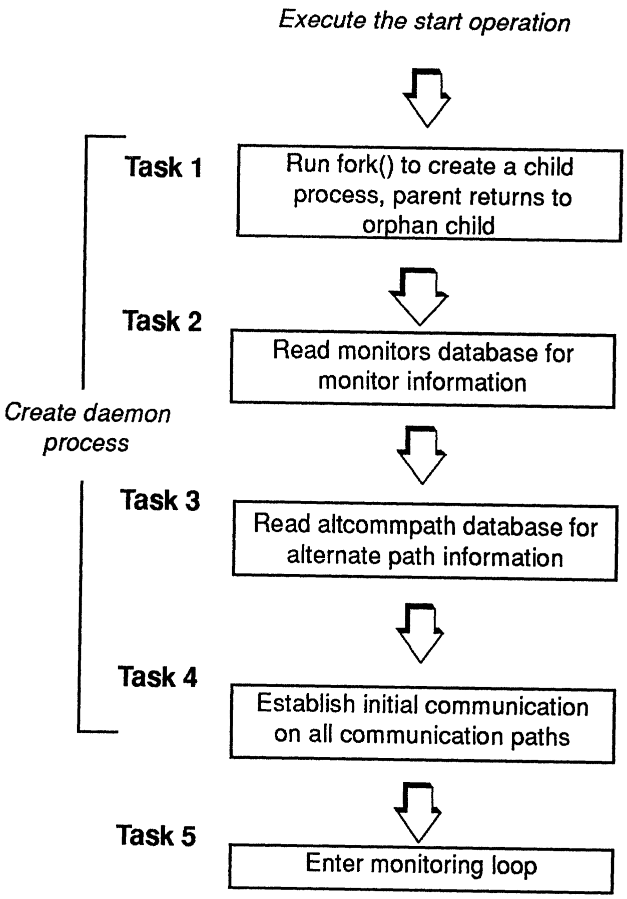
Execute the start operation
Task 1
Run fork() to create a child process, parent returns to orphan child
Task 2
Read monitors database for monitor information
Create daemon process
Task 3
Read altcommpath database for alternate path information
Task 4
Establish initial communication on all communication paths
Task 5
Enter monitoring loop

Figure 3 Tasks Performed in Starting a failovermon Monitor

The first four tasks create a daemon process that has all the necessary information to prepare for monitoring. Task 5 is a loop that the process enters to perform the actual monitoring. This loop is designed to run until the process is terminated.

The Failover Monitor Process' Monitoring Loop

The actual communication messages are sent and received from within the monitoring loop. In addition, the decision whether or not to execute the lost-pulse and regain-pulse scripts is made. The tasks performed in the monitoring loop are shown in Figure 4.

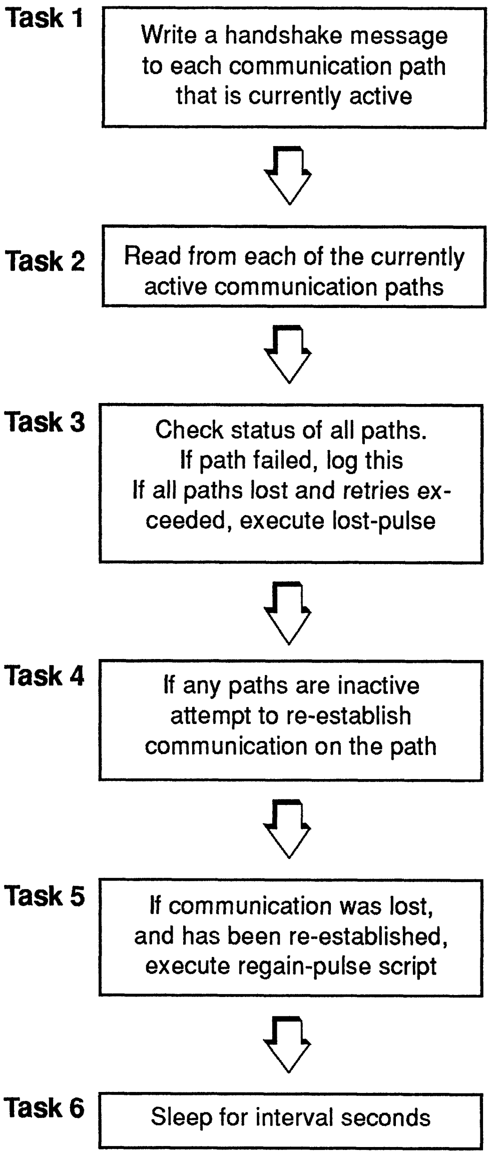
Task 1
Write a handshake message to each communication path that is currently active
Task 2
Read from each of the currently active communication paths
Task 3
Check status of all paths. If path failed, log this If all paths lost and retries exceeded, execute lost-pulse
Task 4
If any paths are inactive attempt to re-establish communication on the path
Task 5
If communication was lost, and has been re-established, execute regain-pulse script
Task 6
Sleep for interval seconds

Figure 4 Tasks Performed in Monitoring Loop.

These six tasks are performed when a failovermon monitor is started, to ensure that the monitoring is accurate. The following section looks more closely at these tasks to show how this is achieved.

Task 1

The first task within the loop writes handshake message to all active communication paths. The handshake message contains information that identifies the message type and what host it is from. This message is sent to the **failoverd(1M)** process on the system that is being monitored. The **failoverd(1M)** process then returns an acknowledgment message to the monitor to indicate it is reachable.

The message is written to all active communication paths so that the status of the paths can be monitored. Using all the paths for each message cycle reduces the time required to detect a system failure and provides diagnostic monitoring of the communication paths.

Task 2

The second task reads each of the active communication paths to see if there is a response to the handshake message. This task is performed with a time-out value so that the reads will not wait forever. If the time-out occurs before the read completes, the read will be retried one time before declaring the line inactive. If the read completes before the time-out, the return message will be checked to ensure that the correct information is returned.

Task 3

The third task determines if the lost-pulse action script is required. If all of the communication paths are inactive and the user-specified retry value has been exceeded, the monitor declares the host to be failed. The lost-pulse action script is then executed.

The monitor does not wait for the lost-action script to complete, nor does it check its exit status. The lost-action script is executed in the background.

Task 4

The fourth task attempts to re-establish communication on all inactive communication paths. The monitor will continue trying to re-establish connections to the host on any paths that are inactive.

If a single path was temporarily lost, this task will detect its return and make it available for the next message cycle. This task detects a host that returns to the network after the host has been restarted.

Task 5

The fifth task determines if the regain-pulse action script is required. If a connection is re-established after all paths were lost, the monitor declares the host to be returned. The regain-pulse script is then executed.

The monitor does not wait for the regain-action script to complete, nor does it check its exit status. The regain-pulse script is executed in the background.

Task 6

The sixth task sleeps for the specified interval. The monitor will sleep for the specified number of seconds before repeating these tasks.

New Failover Databases

MIF uses two new failover databases, which are added to the **/etc/failover directory** as part of the DG/UX 5.4 R2.01 update procedure. The two new failover database files are:

altcommpath	contains entries for alternate communication paths
monitors	contains entries for failovermon monitors

Altcommpath

The failover altcommpath database stores information about alternate communication paths to the host that is being monitored. An alternate communication path is any medium capable of supporting TCP/IP.

The entries in the altcommpath database contain the primary hostname of the host to be monitored and the hostname associated with the alternate communication path.

The alternate communication paths are used by the failovermon monitor process to ensure that the system being monitored is unreachable before the monitor executes the lost-pulse action script.

Monitors

The failover-monitors database stores information about the failovermon monitors that will run on this system to monitor other systems. Before a system can be monitored, an entry must exist in the monitors database. The monitors database entries contain the following information:

- ❑ the name of the host to monitor
- ❑ the process ID of the currently running monitor
- ❑ a flag indicating whether or not to start the monitor on system reboot
- ❑ the interval in seconds that the monitor will sleep between message cycles
- ❑ the number of times the monitor will retry communication before declaring the host failed

- ❑ the user-defined lost-pulse-script to be executed when the host is declared failed
- ❑ the user-defined regain-pulse-script to be executed when the host has returned

Machine Initiated Failover Examples

Figure 5 illustrates an example MIF set up. For this example, assume a configuration with two AViiON AV6225 (rack-mount, dual processor) servers named HostA and HostB. Both servers are running DG/UX 5.4 R2.01. Each server has two LAN controllers, and each server connects to two LANs.

The list that follows the figure highlights the other configuration information for the example.

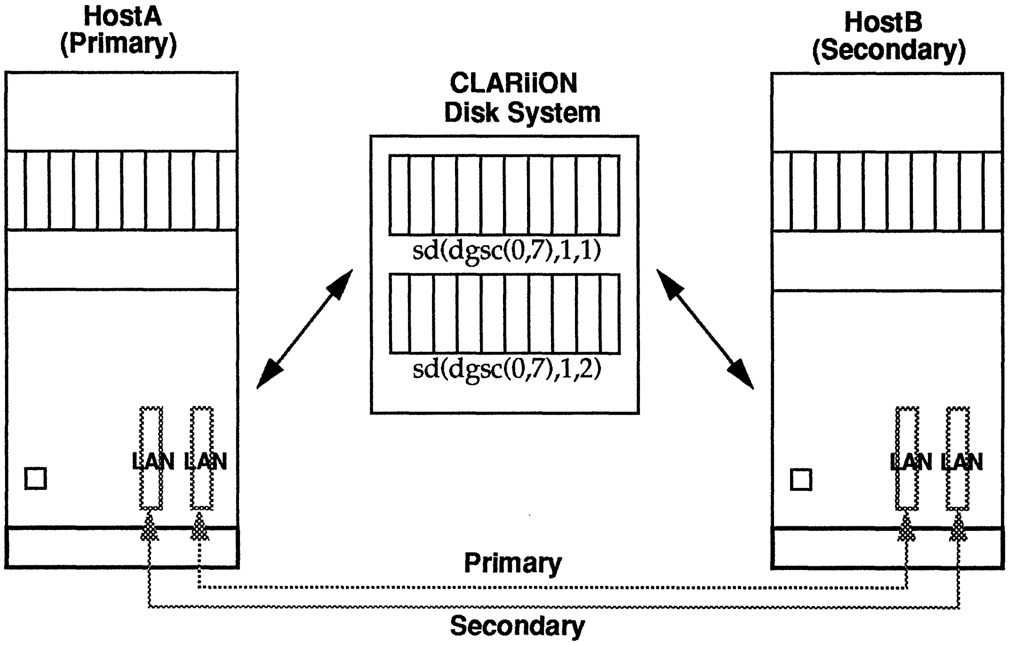

Figure 5 Example MIF Configuration

Other Configuration Information

- ❑ There is one CLARiiON disk system connected to the two servers.
- ❑ The CLARiiON disk system has two RAID-5 disk arrays that contain the application and are to be failed over.
- ❑ HostA currently owns the RAID-5 arrays and will serve as the primary system.
- ❑ HostB serves as the secondary system.

- ❑ The first array "**sd(dgsc(0,7),1,1)**" is used by the Acme Database Manager, which performs raw I/O on a logical disk that spans the entire array.
- ❑ The Acme Database Manager is started with a script called acmeup, which is installed in **/usr/bin**.
- ❑ The second array "**sd(dgsc(0,7),1,2)**" contains three file systems, all of which are set up as fast recovery file systems. One file system will not be exported; the other two file systems will be exported without restrictions.
- ❑ The monitor sends messages every 15 seconds.
- ❑ The monitor retries the message cycle one time before declaring the host to be failed.
- ❑ The monitor executes **/usr/bin/HostA_lost** as the lost-pulse action script.
- ❑ The monitor executes **/usr/bin/HostA_regain** as the regain-pulse action script.
- ❑ The monitor should be restarted when HostB is rebooted.
- ❑ The monitor should be started as part of the add operation.

Preparing for Machine Initiated Failover

Here are the general steps that you would take to protect against a system crash or communication failure.

Preparation (Set up Disk Arrays and Network Interfaces)

Set up the disk arrays on HostA by using gridman to bind disk modules into arrays and by using sysadm to create logical disks and file systems. Set up the network interfaces for both systems to both networks.

OIF Set Up (Set up Failover Databases)

Use the Add operation of the **admfailoverdisk(1M)** command to set up the failover databases so that the disks can be transferred between the two systems (OIF). This procedure is described in the **admfailoverdisk(1M)** manual page, "Managing the DG/UX System," and in the "Operator Initiated Failover in the DG/UX 5.4.2 Operating System" Technical Brief.

MIF Set Up (Set up the Secondary Network Interface)

Set up the secondary network interface in the failover altcommpath database. Select the Sysadm ➤ Device ➤ Disk ➤ Failover ➤ Alternate Paths ➤ Add option on HostB. This operation will ask you for the "Alternate Remote Host Name" of the system to be monitored. This is the name to access HostA by the secondary network. You can optionally check this path to ensure that it is accessible.

Set up and Start the Monitor on HostB

When the alternate communication path has been added, you can set up and start the monitor. To do that, on HostB, select the Sysadm➤Device➤ Disk➤Failover➤Monitors➤Add option. The add operation requires that you enter the following information (highlighted in bold).

□ The primary host name of the system to monitor (**HostA**).

□ The number of seconds that the monitor should sleep between message cycles (**15**).

□ The number of times that the monitor should retry the message cycle before declaring the host to be failed (**1**).

□ The full pathname to a user-defined script that will be executed when the monitored host has failed (**/usr/bin/HostA_lost**).

□ The full pathname to a user-defined script that will be executed when the monitored host has returned (**/usr/bin/HostA_regain**).

□ A flag indicating whether or not the monitor should be started when the system is rebooted (**yes**).

□ A flag indicating whether or not the monitor should be started after the successful completion of the add operation (**yes**).

For this example the following failovermon command is formatted and executed by sysadm:

```
failovermon -o add -i 15 -r 1 -l /usr/bin/HostA_lost -g /usr/bin/HostA_regain -b -s HostA
```

This command creates the failovermon monitor process, starts the process running on HostB, which starts monitoring HostA. Running on HostB, the failovermon monitor sends messages to HostA via the primary and secondary network interfaces. The responses are received, and the monitor sleeps for 15 seconds before repeating the monitoring loop.

What Happens if HostA Fails?

Let's assume that HostA panics after you've started the monitor running on HostB (Figure 6). At the beginning of the next monitoring loop, the monitor sends its handshake messages to HostA. The monitor attempts to read the responses, but both attempts time out.

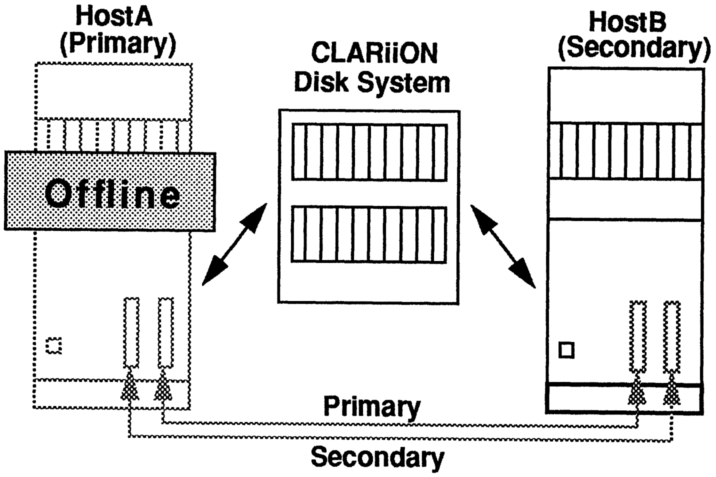

Figure 6 HostA Offline

The monitor then looks at its retry value, which is one. This tells the monitor to attempt to re-connect to HostA on both of the network interfaces. The connections fail and the monitor declares HostA to be failed. The monitor then executes the lost-pulse action script (**/usr/bin/HostA_lost**).

The Lost-Pulse Script

The lost-pulse script uses the **admfailoverdisk(1M)** command to take control of the physical disks and re-start the application on HostB. The monitor continues trying to re-establish communication with HostA.

The Regain-Pulse Script

When HostA is rebooted, communication will be re-established and the monitor will execute the regain-pulse action script (**/usr/bin/HostA_regain**). The **admfailoverdisk(1M)** command in the regain-pulse action script will return the physical disks to HostA.

What Happens If There's a Communication Failure?

Assume that both HostA and HostB are running normally. The monitor on HostB is sending and receiving messages over both network interfaces. All is well until someone inadvertently disconnects the primary network interface cable from the back of HostA (Figure 7).

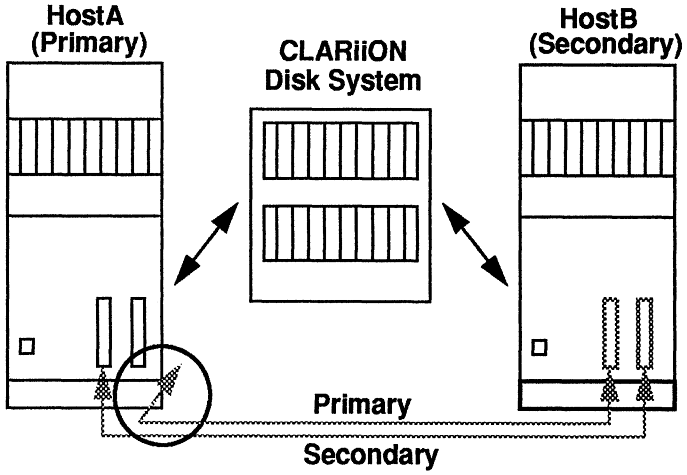

Figure 7 Communication Failure at HostA

The users who are accessing HostA directly do not notice that the primary network is down. The monitor running on HostB detects the loss of the primary network interface, but continues to receive responses on the secondary network interface.

The loss of the primary network interface is logged by the monitor, but there is no need for the lost-pulse action script to be executed. However, the monitor continues trying to re-establish communication to HostA via the primary network interface.

When the primary network interface is reconnected, the monitor will be able to reconnect to HostA. The regain-pulse action script will not be executed, since all communication was not lost.

An Example Lost-Pulse Action Script

The lost-pulse action script is executed when the monitor determines that the host it is monitoring has failed. The contents of this script are user defined, and the script should contain the actions that you want executed when the monitored host fails.

We provide a default script (**failovermon_lost_pulse**) in **/etc/failover**. The default script, shown below, has no executable commands but documents some actions that will most likely be taken. This script illustrates the use of the **admpdisk(1M)** command to determine whether the physical disks that you want to transfer are already registered. This check is important to avoid trespassing on physical disks that this system is currently using.

This script can also be used to perform any system preparation, such as terminal or printer set up, that will be required when the disks are transferred, or to send mail alerting designated individuals of the system failure.

```
#!/bin/sh

####################################################################
# Copyright (C) Data General Corporation, 1984 - 1992
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
#     $What: $
#
####################################################################

####################################################################
#
# The failovermon_lost_pulse script is executed when the failovermon
# monitor has determined that the host it is monitoring has failed.
# This script should be used to perform any actions required when
# the system being monitored fails.
#
# This script will most likely be used to take control of the failed
# systems physical disks to restart the application.
#
# Any additional set up (enabling of tty lines or printers) can also
# be performed at this time to prepare for users switching systems.
#
# This script will most likely contain the following command to take
# control of the physical disks:
#
#     admfailoverdisk -o take -h <hostname> -T <diskname ...>
#
# Before executing this command it is recommended that you use the
# admpdisk(1M) command to check if the disks have already been
# transferred to this system. If the disks are already registered
# on this system, the previous admfailoverdisk(1M) command line
# will cause any applications accessing the disks to be terminated
# and restarted.
#
# The following sample commands should avoid this problem:
#
#     DISKLIST="sd(ncsc(),2) sd(ncsc(),5)"
#
#     admpdisk -o list -q "$DISKLIST" > /dev/null 2>&1
#
#     if [ $? -ne 0 ]
#     then
#            admfailoverdisk -o take -h <hostname> -T "$DISKLIST"
#     else
#            echo "Disks $DISKLIST are already registered!"
#     fi
#
####################################################################
```

Regain-Pulse Action Scripts

The regain-pulse action script is executed when the monitor determines that the host it is monitoring has returned. This script should contain the actions that you want executed when this condition occurs. The contents of the script are user defined.

This script can be used to send mail to designated individuals alerting them that the primary host has returned. In addition, the script can be used to shut down the application, return the physical disks to the primary host, and restart the application there.

The following default script documents some actions that are likely to be performed when a host returns.

```
#!/bin/sh

######################################################################
# Copyright (C) Data General Corporation, 1984 - 1992
# All Rights Reserved.
# Licensed Material-Property of Data General Corporation.
# This software is made available solely pursuant to the
# terms of a DGC license agreement which governs its use.
#
#     $What: $
#
######################################################################

######################################################################
#
# The failovermon_regain_pulse script is executed when the failovermon
# monitor has determined that the host it is monitoring is reachable
# again. This script should be used to perform any actions required
# when the failed system being monitored returns.
#
# This script will most likely be used either to send mail to a
# system administrator to notify him of the systems return, or to
# relinquish control of the physical disks it took over when the
# system being monitored failed.
#
# If this script is to relinquish control of the disks that it took,
# it will have a command similar to the following, to give the disks
# back to the host it took them from:
#
#     admfailoverdisk -o give -h <hostname> -T <diskname ...>
#
# Before executing this command it is recommended that you gracefully
# shut down your application.
#
######################################################################
```

For More Information

The following articles discuss disk failover and CLARiiON disk systems in more detail.

DG/UX™ Technical Brief: *Operator Initiated Failover in the DG/UX™ Operating System* (012-004188-01), January 27, 1993, Data General Corporation

DG/UX™ Technical Brief: *A Look at High Availability Disk Systems* (012-004035), July 31, 1991, Data General Corporation