EMC²
where information lives®

Preventative Monitoring in the NAS Environment

EMC Proven Professional Knowledge Sharing 2009

Robert Wittig
Infrastructure Specialist
EDS, an HP company
bob.wittig@eds.com

EMC² PROVEN PROFESSIONAL

Preventative Monitoring in the NAS Environment

EMC Proven Professional Knowledge Sharing 2009

Robert Wittig
Infrastructure Specialist
EDS, an HP company
bob.wittig@eds.com

Table of Contents

Disclaimer: The views, processes or methodologies published in this compilation are those of the authors. They do not necessarily reflect EMC Corporation's views, processes, or methodologies

Abstract

As the EMC Celerra® environment expands to larger capacities and more installed frames, it becomes increasingly important to assess the health of each frame and verify that the configuration fully utilizes the Celerra's redundant capabilities. Verification, testing and preventative monitoring are critical to maintain reliability and availability.

EMC Celerra monitoring starts with a properly configured system including the ConnectHome capability, notifications, redundant connectivity to the network and backend storage, properly configured data mover standby relationships and optionally, redundant control stations. Once configured, you must test and monitor the Celerra environment to verify that it will properly handle faults and continue to function. Do not stop testing once the system is in production, rather continue at regular intervals. Make non-intrusive checks at regular intervals to verify that changes have not adversely impacted the environment. You should also perform periodic intrusive tests of the redundant configuration where possible.

My objective is to provide Celerra Storage Administrators with a set of steps to check the status, verify redundant operation, and confirm that failure notifications are functioning properly. Finally, this article suggests methods that you can apply to gather these checks into a single automatic process. This process can be regularly executed to provide evidence and identify potential problems before they impact the Celerra's availability.

Introduction

Whether you are setting up your Celerra for the first time, or managing a Celerra that has already been installed and configured, your first concern is that your systems continue to provide the services that they were designed to provide. A key part of Celerra operations is ensuring that the impact of any problem in your environment is minimized and is completely transparent to users.

The primary goal of this article is to build your confidence in your Celerra environments' ability to continually provide the services for which it was installed. EMC built your Celerra to continue running in the event of a hardware failure, and your environment is configured to eliminate any single point of failure. But, how do you know this and how do you verify that no single points of failure have been accidentally introduced into the environment as changes have been made to the Celerra? More importantly, how do you demonstrate to your management and perhaps to your customer that your configuration is redundant, stable and functioning properly?

This article assumes that you have already configured your Celerra to eliminate all single points of failure, that you intend to maintain this state, and that you will fix any problems as they are discovered. You may be able to identify opportunities to improve the redundancy in your Celerra by following the techniques discussed in this article. However, there are no guarantees against every potential failure. Regular testing and validation are key contributors to reliable operation.

This article is based on a Celerra Gateway environment but the concepts also apply to integrated Celerra installations. For instance, concepts applicable to a Symmetrix® backend will be indicated as well as specific details applicable to a Celerra with a CLARiiON® backend. In most instances, concepts for an integrated Celerra will be similar or identical to a gateway Celerra with a CLARiiON backend.

Call for Help

Your first line of defense to maintain your Celerra environment is to ensure that you and EMC are aware of any hardware failures. It is essential that the correct groups are notified and can work to restore or replace the failed component whether it's a Celerra cooling fan or an entire data mover.

As of Celerra DART software version 5.5, EMC has provided multiple methods to send ConnectHome messages. In the past, many systems used a modem to dial home to EMC. The ConnectHome facility adds the option to send ConnectHome messages to EMC over Email or directly to an FTP server. You can configure your Celerra to use one or more of these transports, so you can specify that ConnectHome messages be sent first by Email and then if the Email send fails, the modem or FTP should be used to send out the message.

Using Email as the primary transport offers the advantage of multiple destination addresses as well as using an established corporate Email infrastructure. If you put a second Email address of a Celerra administrator or administrator distribution list in the recipient address of the ConnectHome configuration screen, those recipients will get a copy of the same Email that is sent to EMC. You will find, however, that since the actual ConnectHome message is encrypted, you will not be able to read the actual details of the failure. However, the serial number of the Celerra sending the ConnectHome message is included in plain text in the subject of the Email, so you can determine which Celerra you should review to identify the problem.

Even if your Celerra does not use the modem to send ConnectHome messages to EMC, you may still need to have the modem available for EMC to dial-in to your Celerra to diagnose and resolve problems. So in this case, it is important to regularly test and verify that the modem is connected to a phone line and is functioning properly.

Understanding the importance of the ConnectHome service leads to the first verification procedure for the Celerra environment. You can check the date and time of the last successful ConnectHome by looking at the file `/nas/log/ConnectHome_log`

```
[nasadmin]$ more /nas/log/connecthome_log
Nov 30 10:29:32 2008 CallHome:6:50 ConnectHome Test of the
Primary Email initiated
Nov 30 10:30:34 2008 CallHome:3:52 ConnectHome Test of the
Primary Email failed
Dec  1 04:02:50 2008 CallHome:3:2 Primary Email failed to
transfer CallHome event file : Description: email_1, retries: 1
Dec  1 12:40:43 2008 CallHome:6:50 ConnectHome Test of the
Primary Modem initiated
Dec  1 12:42:11 2008 CallHome:5:51 ConnectHome Test of the
Primary Modem succeeded
Dec  2 18:59:39 2008 CallHome:6:50 ConnectHome Test of the
Primary Email initiated
Dec  2 18:59:41 2008 CallHome:5:51 ConnectHome Test of the
Primary Email succeeded
Jan  1 04:03:59 2009 CallHome:7:25 Callhome successfully
transferred
```

Figure 1: ConnectHome Log Output Sample

This example log shows that the ConnectHome service initially had a problem with a manual test of the Email transport. The primary modem transport was successfully tested. Once the problem with Email was resolved, the test of the Email transport was successful and is reflected in the log. Finally, the last line indicates that an actual Callhome was triggered and was successfully transferred.

You can initiate ConnectHome tests from the command line or from the Celerra Manager Graphical User Interface, (GUI) when logged on as the root user.

To test Email ConnectHome from the command line, use the command:
`/nas/sbin/nas_connecthome -test -email_1`

To test modem ConnectHome from the command line, use the command:
`/nas/sbin/nas_connecthome -test -modem_1`

Both of these commands will provide feedback on the screen and in the ConnectHome_log file. Issuing the command **`/nas/sbin/nas_connecthome`** without parameters will provide syntax and help information.

Regularly checking this log allows you to track when each transport was last used, and when you should perform a test of your ConnectHome transports.

Perform these same tests from the Celerra Manager GUI if the GUI is logged on with the root user id.

To get to the ConnectHome screen in the GUI, login to the Celerra GUI as root, click on the Support folder, (at the bottom of the folder list on the left hand side), and click on the ConnectHome tab a the top of the Support screen. You will see the following screen:

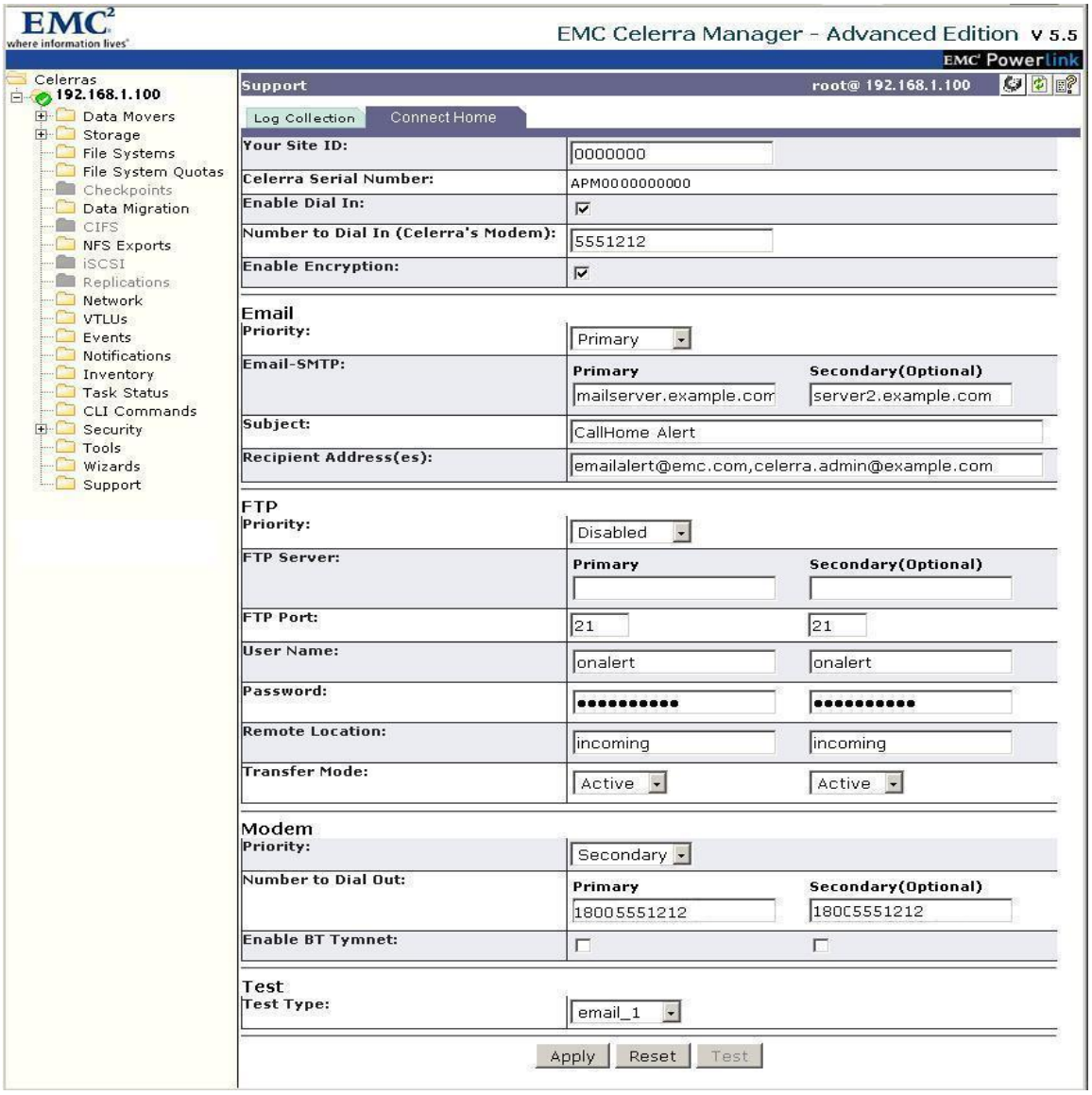


Figure 2: ConnectHome Configuration Page

Add your Email address to the Recipient Address(es) field so the resulting tests will be sent to your Email as well. If you are adding your Email address for testing purposes, please remove the EMC email address, "emailalert@emc.com," before generating tests so that the EMC support staff does not have to acknowledge multiple test Emails.

Capture an image of the ConnectHome screen before you make any changes so that you can put things back the way they were if you encounter any problems. This can be done with a screen capture from the GUI or by capturing the output of the **/nas/sbin/nas_connecthome -info** command:

```
[nasadmin]$ /nas/sbin/nas_connecthome -info
ConnectHome Configuration:


Encryption Enabled         = yes

Dial In :
   Enabled                 = yes
   Modem phone number      = 8005551212
   Site ID                 = 0000000
   Serial number           = APM00000000000

Email :
   Priority                = 1
   Sender Address          = connectemc@emc.com
   Recipient Address(es)   =
emailalert@emc.com,celerra.admin@example.com
   Subject                 = CallHome Alert
   Primary :
     Email Server          = mailserver.example.com
   Secondary :
     Email Server          = server2.example.com

FTP :
   Priority                = Disabled
   Primary :
     FTP Server            =
     FTP Port              = 21
     FTP User Name         = onalert
     FTP Password          = **********
     FTP Remote Folder     = incoming
     FTP Transfer Mode     = active
   Secondary :
     FTP Server            =
     FTP Port              = 21
     FTP User Name         = onalert
     FTP Password          = **********
     FTP Remote Folder     = incoming
     FTP Transfer Mode     = active
```

```
Modem :
   Priority                = 2
   Primary :
     Phone Number          = 1800#######
     BT Tymnet             = no
   Secondary :
     Phone Number          = 1800#######
     BT Tymnet             = no
```

Figure 3: Listing Current ConnectHome Configuration

Once you have added your Email address to the Recipient Address field and saved the changes with the "Apply" button, you can test the ConnectHome transport by selecting "Test Type" at the bottom of the screen. In this example, a test will be performed on the email_1 connection method. You will receive a test Email from the Celerra as long as everything is properly configured. I suppose you could also change the phone number and have it dial your phone, but always make sure you leave ConnectHome configured properly once your testing has been completed.

There are also other Celerra monitoring options configured in the Celerra Manager GUI under the Notifications folder. These can be configured to use Email or SNMP traps for automatic notification of problems and even file system utilization reports. We will look at configuring these notifications next.

Alerts and notifications

Now that we have configured the ConnectHome facility and know that it is working properly, we will cover some simple notification configurations. You can configure notifications from the command line or from Celerra Manager GUI. We will work with the GUI in this article.

Select the Notifications folder on the left hand side of the screen, then select the Events tab at the top of the Notifications page to configure Event notifications from the Celerra Manager GUI. The Notifications page shows the Celerra events currently configured.

In this example, there is a single event configured for BoxMonitor warnings to be sent to a SNMP trap at 192.168.1.200:

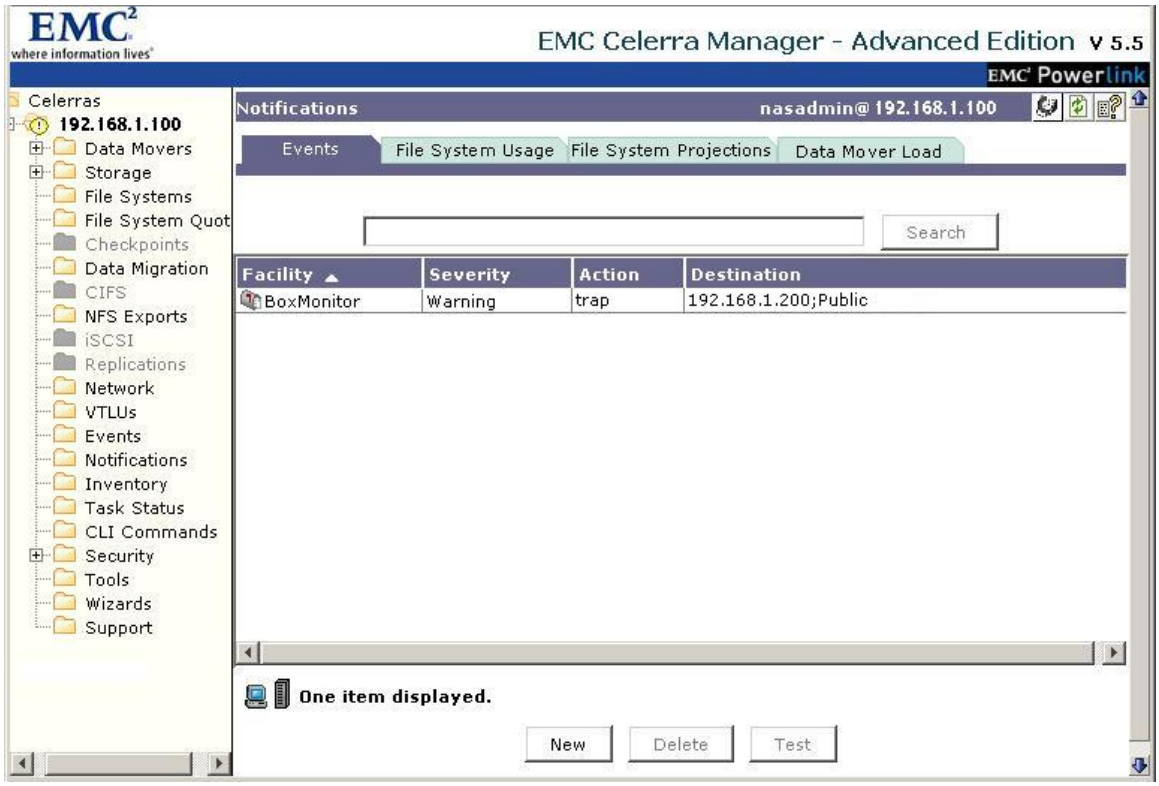


Figure 4: Configure Notifications - Events screen

The IP address in this example is the address of the EMC ControlCenter® server that receives and handles the BoxMonitor warning. (Please make sure that you enter the SNMP server's correct IP address so that you do not inadvertently send information to the wrong server or one which is not configured as a SNMP server). You can find a full list of BoxMonitor events in the document titled: Configuring Celerra Events and Notifications, available on Powerlink.

To add another event, select the New button and fill in the New Notification screen. We will configure an Email notification for all Critical BoxMonitor events (please see next page).

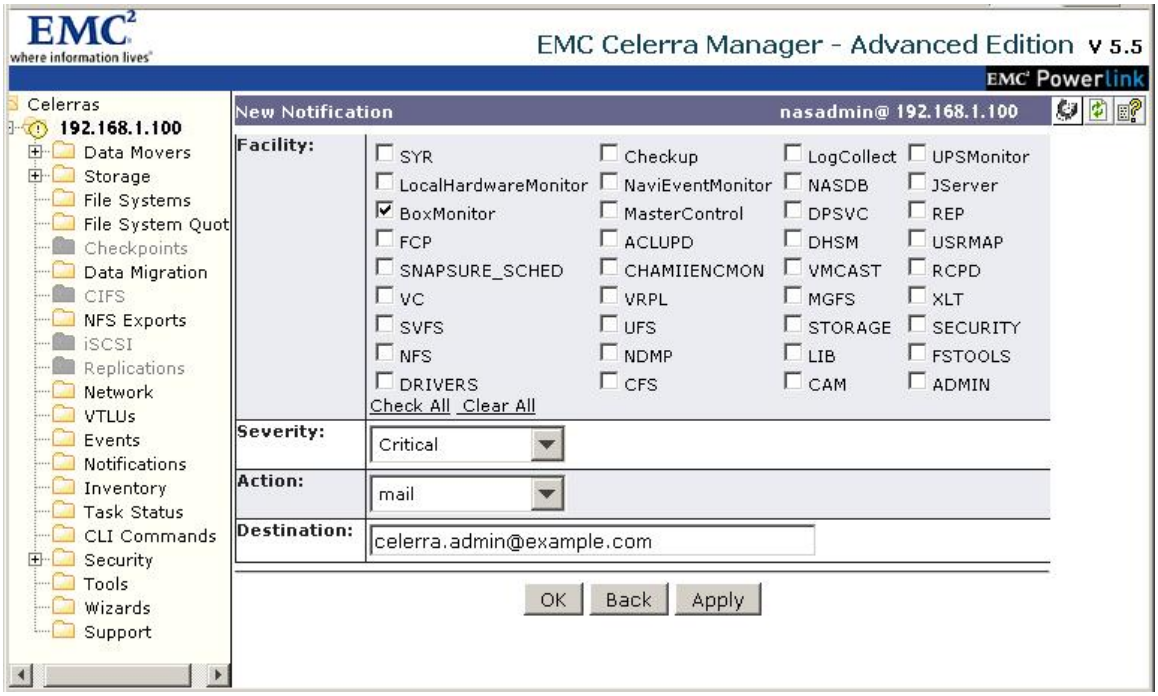


Figure 5: Configure Notifications - BoxMonitor Email

Fill in the check box for BoxMonitor, select the “Critical” severity level, select the “mail” action, and enter the Email address in the destination field. This will send an Email for any Critical notification from the BoxMonitor.

There are additional notification tabs available, the File System usage tab is one of the most useful. It allows the Celerra administrator to configure notifications to be sent whenever a specified threshold is exceeded for all file systems or for individual file systems. It is commonly used to notify someone when any file system goes above 90% utilization, a level which could impact file system performance. Another possibility is to configure a threshold for each file system and have that notification go directly to the owner of that file system, so that they can decide on a course of action. They may clean up the file system or request additional space with a file system extend.

In the next screen, we will configure a single file system notification event to check every 24 hours to see if the file system has exceeded 80% utilization or if the file system has used up 90% or more of the available inodes for the file system. This event will send an Email message to the file system owner, "claims.alerts@example.com." You can specify multiple Email addresses by separating them with a comma.

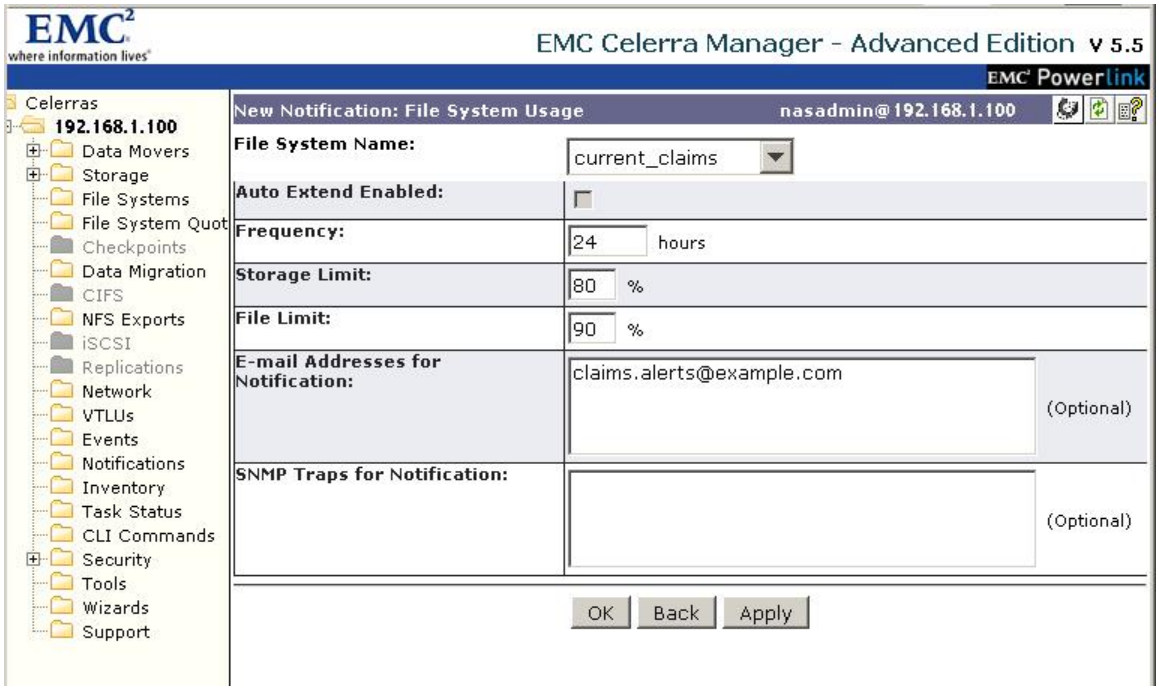


Figure 6: Configure Notifications - File System Usage

The file system has a greater likelihood of being cleaned or extended before it starts to impact the Celerra's performance when you notify the file system owner before it reaches a critical threshold of 90% space utilization.

Use the same process to set alerts on the File System Projection tab, where the Celerra can be configured to estimate if a file system will fill up in a configurable number of days, based upon the growth measured at hourly intervals over the past number of days. This allows you to receive alerts if a file system is growing fast enough to reach its capacity in the next month, months or even year. Projections are made from samples taken every H number of hours over D days and will send an alert if the file system is growing at a rate that would reach capacity in the next W days. So if the Frequency H was 1 hour, the Interval D was 1 day, and the number of days to Warn before the file system filled up was 90 days, any file system growing fast enough over the last day such that it would reach capacity before the next 90 days, would generate an alert to the destination Email or SNMP trap.

The Data Mover Load tab is the last Notification screen. It allows you to configure notifications based upon the CPU utilization and Memory utilization of one or all of the data movers. When first displayed, the Data Mover Load tab shows the previously configured notifications.

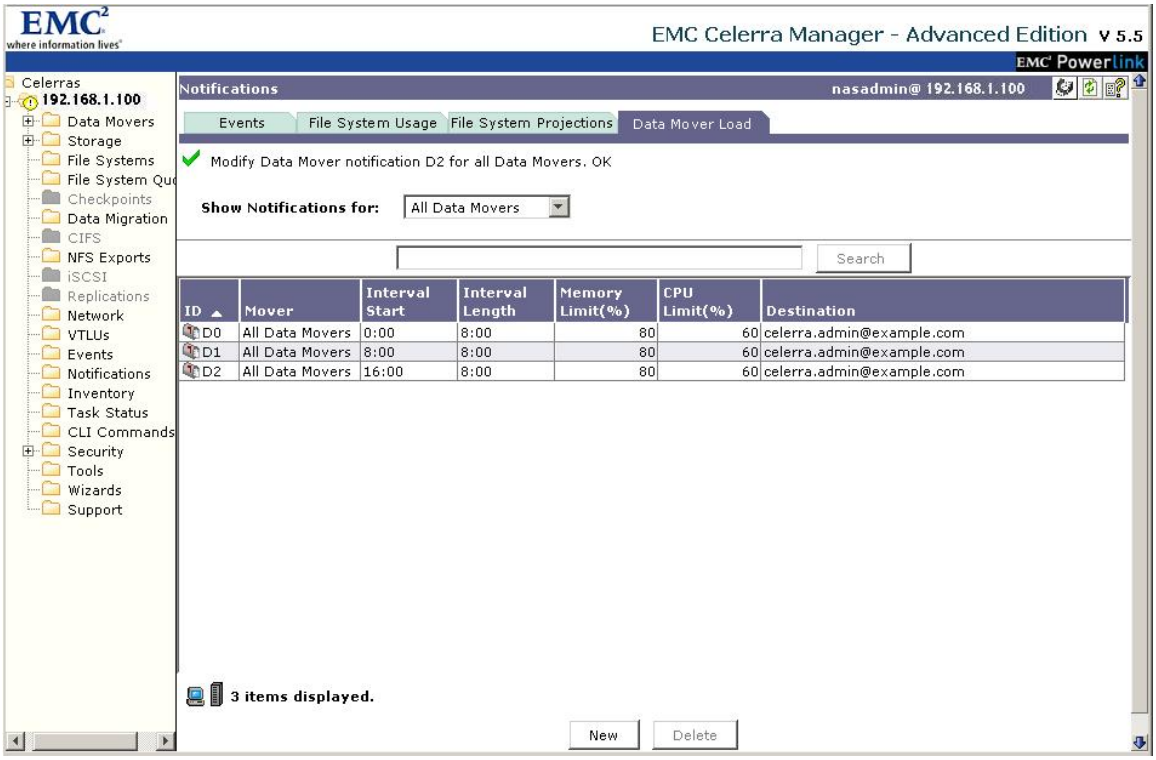


Figure 7: Configuring Notifications - Data Mover Load screen

Figure 7 shows separate notifications configured. Each notification applies to All Data Movers in the Celerra, and will send an alert if memory utilization exceeds 80%, or if CPU utilization exceeds 60% for the 8 hour period specified. We configured three separate entries so that each shift in the day is alerted separately should the thresholds be exceeded. Each of these notifications sends an Email to the Celerra Administrator group if the notification is triggered.

These notifications are configured by filling out the screen displayed with the New button. Suppose we want to set up a new alert just for our backup window that starts at 18:00. We would press the New button and fill in following screen (see next page):

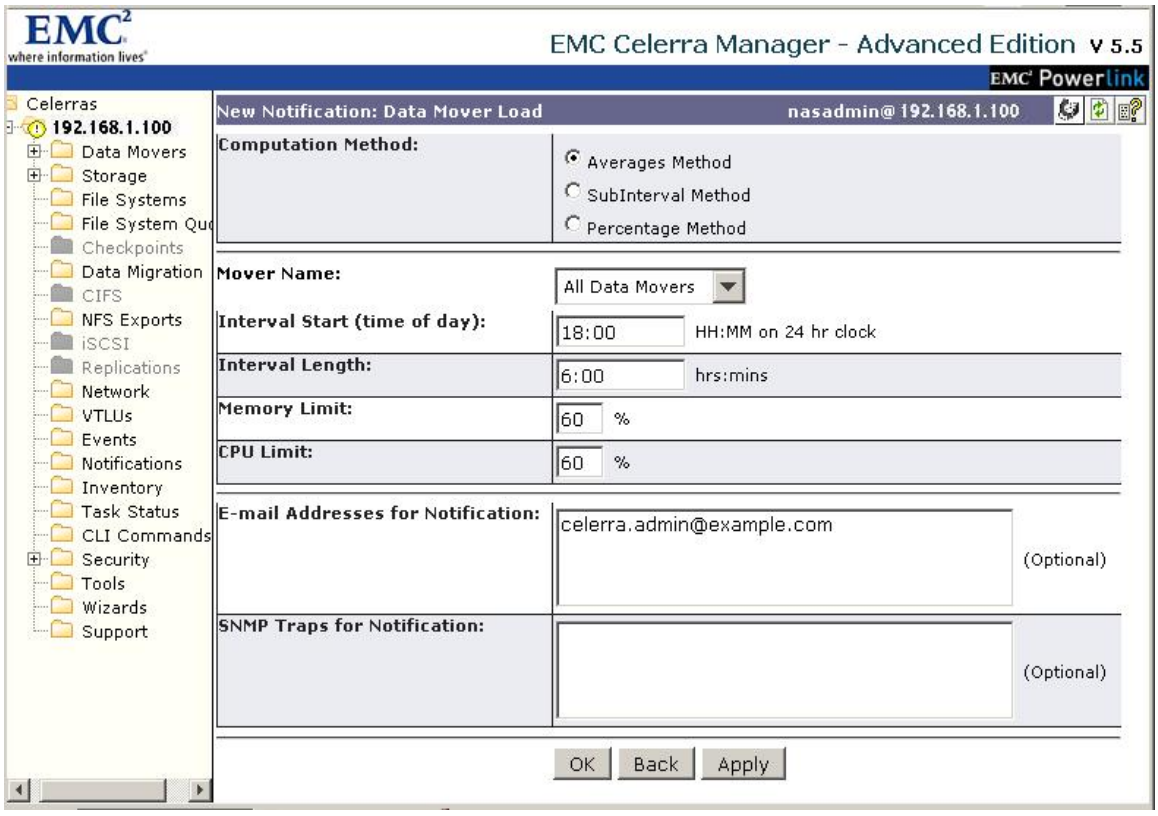
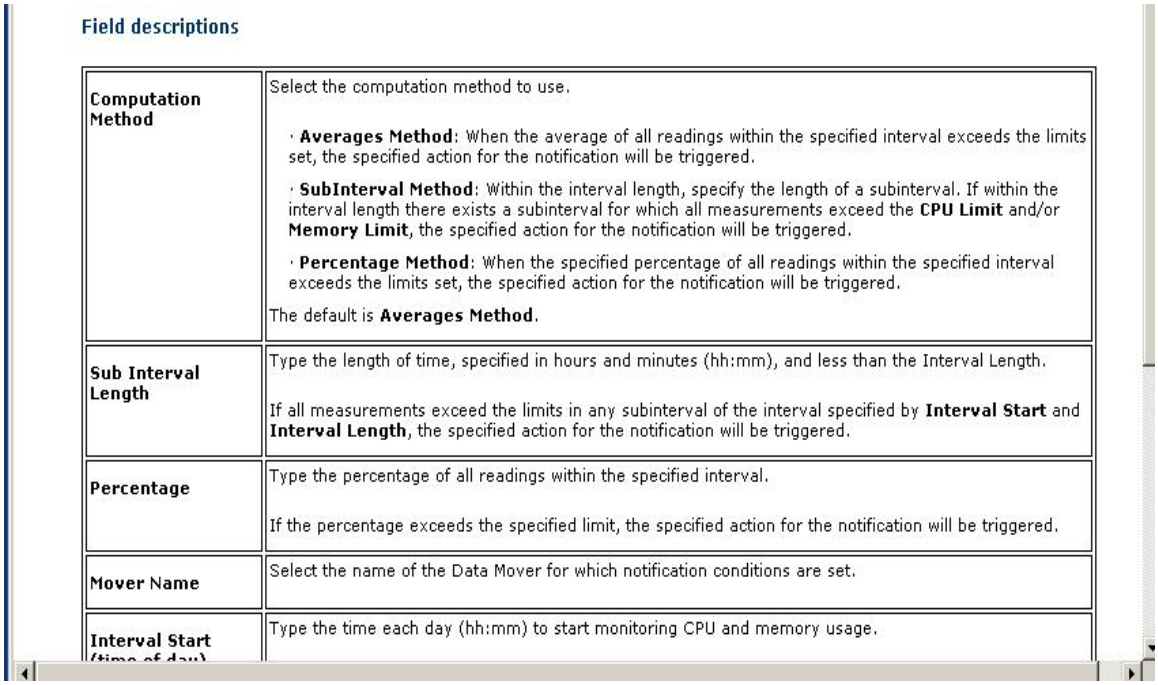


Figure 8: Configuring Data Mover Load

We are using the Average utilization across the entire interval computation method. Again, we select All Data Movers to be monitored. Our backup window starts at 18:00 and we select an interval length of 6 hours. We are limited to 6 hours in this example because the start time plus the interval should not span midnight. (As we are reminded when we hover the mouse over the filed name Interval Length and bring up the help bubble associated with the field on this screen.) We select a lower value of 60% as the threshold for both of these fields since we want to know when the backup process starts to have a noticeable impact on the Celerra CPU and Memory Utilization.

Select the Email box for the Celerra Admin as the target for any generated notifications. We could instead direct the alert to be sent as an SNMP trap instead of an Email address. This allows us to integrate the notifications into a central SNMP server and monitoring setup. Also notice that the Email address is an optional field, so it is possible to have the alert sent to the Celerra log rather than an Email box or SNMP trap. Press the OK button to save and activate the new alert once you have filled in the screen.

We could have also used the "SubInterval Method" that, according to the Celerra Manager's associated help page, allows us to specify a Sub Interval Length. The Sub Interval Length is a shorter interval within the entire interval monitored over which any measurement exceeding the threshold will send a notification. The help page is the only place where the separate Computation Methods are explained, I could not find any information on them in any other Celerra manual.



Field descriptions

Computation Method	Select the computation method to use. · **Averages Method**: When the average of all readings within the specified interval exceeds the limits set, the specified action for the notification will be triggered. · **SubInterval Method**: Within the interval length, specify the length of a subinterval. If within the interval length there exists a subinterval for which all measurements exceed the **CPU Limit** and/or **Memory Limit**, the specified action for the notification will be triggered. · **Percentage Method**: When the specified percentage of all readings within the specified interval exceeds the limits set, the specified action for the notification will be triggered. The default is **Averages Method**.
Sub Interval Length	Type the length of time, specified in hours and minutes (hh:mm), and less than the Interval Length. If all measurements exceed the limits in any subinterval of the interval specified by **Interval Start** and **Interval Length**, the specified action for the notification will be triggered.
Percentage	Type the percentage of all readings within the specified interval. If the percentage exceeds the specified limit, the specified action for the notification will be triggered.
Mover Name	Select the name of the Data Mover for which notification conditions are set.
Interval Start	Type the time each day (hh:mm) to start monitoring CPU and memory usage.

Figure 9: Configure Notifications - Data Mover Load help page

You can set up notifications to be as detailed or as broad as your needs dictate. Some of these notifications can be configured from the command line with configuration files that will facilitate duplicating a standard notification configuration across multiple Celerras. You can find additional details about configuration files and utilizing them from the command line in the Configuring Celerra Events and Notifications document on Powerlink.

Data Mover Standby Configuration

A properly configured Primary and Standby data mover relation is one of the simplest preventative measures to ensure that your Celerra continues to provide the services for which it was installed. Verifying you Data Movers' configuration is the next task.

Step one is to verify that your Celerra has a standby Data Mover configured to take over for every primary Data Mover in the environment. This is done with the command:

```
[nasadmin]$ nas_server -info -all
id        = 1
name      = server_2
acl       = 1211, owner=nasadmin, ID=201
type      = nas
slot      = 2
member_of =
standby   = server_4, policy=auto
status    :
  defined = enabled
   actual = online, active

id        = 2
name      = server_3
acl       = 1211, owner=nasadmin, ID=201
type      = nas
slot      = 3
member_of =
standby   = server_4, policy=auto
status    :
  defined = enabled
   actual = online, active

id        = 3
name      = server_4
acl       = 1211, owner=nasadmin, ID=201
type      = standby
slot      = 4
member_of =
standbyfor= server_2,server_3
status    :
  defined = enabled
   actual = online, ready
```

Figure 10: nas_server Information Display

The key line starts with "*standby*" for each of the primary data movers. The example in Figure 10 has a single standby data mover configured for both primary data movers. Of course, this configuration only helps for the first failure of one of the data movers. If the second data mover fails while the first is still failed over to the single standby data mover, the second data mover may not be available for your customers. (That's why they call it protection against a single point of failure).

The "*policy*" value is the other bit of important information on this line. The data movers in this example are configured to automatically fail over to the standby in the event of a failure. You could set them up so they only fail over when you tell them to, the "*manual*" policy, but personally, I would rather let the Celerra take care of that. Your environment may have different objectives making the "*manual*" or "*reboot*" policies desirable. You can also confirm the standby data mover by checking the "*standbyfor*" line listed under the standby data mover's information.

Once we have verified that the appropriate standby relationship and failover policies are set, we should ask the Celerra if it believes everything is configured properly for failover. To do this, issue the command:

```
[nasadmin]$ server_standby ALL -verify mover
server_2 : ok
server_3 : ok
server_4 : ok
```

Figure 11: Standby Configuration Database Check

This command queries the data mover's configuration database, verifying that each data mover is correctly set up for failover. A simple check for "*ok* " tells us that the Celerra believes that everything is in place for a possible failover. This is a pretty simple test, but potentially embarrassing if this check is not performed. You may discover that things are not "*ok* " during a failure!

However, we are not going to just take the Celerra's word for it, instead, we should perform regular failover tests. For the purposes of this article, the steps to actually perform a failover and failback test will be deferred until after the rest of our Celerra's redundant configuration has been checked.

One result of performing a failover test is that the uptime of each data mover will be reset. You can determine the uptime for a data mover with the **server_uptime** command:

```
[nasadmin]$ server_uptime server_2
server_2 :  up 70 days 19 hours 26 min 6 secs
```

Figure 12: server_uptime output

Combine this command with the command to list the current time on the data mover and you have documentation for how long it has been since a data mover was rebooted, as well as what time you performed the last failover test. The time on the data mover is displayed with the **server_time** command:

```
[nasadmin]$ server_time server_2
server_2 : Fri Feb 29 15:03:21 CST 2008
```

Figure 13: server_time output

Now that we have squeezed information out of the data mover hardware and software, we can move on to the data mover's connections to the network and backend storage.

Redundant Network Connectivity

Once we know that our Celerra is configured and running smoothly, we next want to check that the Celerra has redundant connections to the hosts mounting file systems and that the connections between the Ethernet network switches and Celerra are functioning properly.

This section is based upon the assumption that each Celerra data mover is using Celerra's fail safe networking feature to combine two Ethernet interfaces on a data mover into a single virtual interface. You can also achieve redundant network connections using features in the Ethernet switch itself. However, these types of connections, specifically Cisco Fast Ether Channel or IEEE 802.3ad Link Aggregation Control Protocol (LACP), are not covered in this article.

First, we identify the network configuration for a data mover and any configured fail safe network devices. Do this using the **server_ifconfig server_2 -all** command:

```
[nasadmin]$ server_ifconfig server_2 -all
server_2 :
192_168_1_5 protocol=IP device=fsn001
         inet=192.168.1.5 netmask=255.255.255.0
broadcast=192.168.1.255
        UP, ethernet, mtu=1500, vlan=0, macaddr=10:11:12:13:14:15
loop protocol=IP device=loop
         inet=127.0.0.1 netmask=255.0.0.0
broadcast=127.255.255.255
        UP, loopback, mtu=32768, vlan=0, macaddr=0:0:0:0:0:0
netname=localhost
el30 protocol=IP device=fxp0
         inet=128.221.252.2 netmask=255.255.255.0
broadcast=128.221.252.255
        UP, ethernet, mtu=1500, vlan=0, macaddr=30:31:32:33:34:35
netname=localhost
el31 protocol=IP device=fxp0
         inet=128.221.253.2 netmask=255.255.255.0
broadcast=128.221.253.255
        UP, ethernet, mtu=1500, vlan=0, macaddr=40:41:42:43:44:45
netname=localhost
```

Figure 14: Data Mover Network Configuration Output

In this example, there is a fail safe network virtual device named "*fsn001*" with an IP address configured as 192.168.1.5.

The physical ports which make up the virtual device *fsn001* can be identified with the **server_sysconfig server_2 -virtual** command:

```
[nasadmin]$ server_sysconfig server_2 -virtual
fsn001    active=cge0 primary=cge0 standby=cge1
fsn    failsafe nic devices : fsn001
trk    trunking devices :
```

Figure 15: Failsafe Network Virtual Device Display

This output shows that the virtual device **"***fsn001***"** is made up of the two physical interfaces `cge0` and `cge1`. Since only one of the interfaces is active at one time, we also see that cge0 is the primary or active device and cge1 is the standby device.

We can verify that data is flowing on the cge0 interface and not flowing on the cge1 interface with the **server_netstat server_2 -i** command:

```
[nasadmin]$ server_netstat server_2 -i
Name     Mtu   Ibytes        Ierror  Obytes        Oerror  PhysAddr
*************************************************************************
fge0     9000  0             0       0             0       0:60:22:29:3d:4d
fge1     9000  0             0       0             0       0:60:22:29:3d:4c
mge0     9000  0             0       0             0       0:60:22:25:74:4e
mge1     9000  0             0       0             0       0:60:22:25:74:4f
cge0     9000  1859079501    0       2279849874    0       0:60:22:29:2c:4b
cge1     9000  1613156653    0       0             0       0:60:22:29:2c:4b
cge2     9000  552686933     0       0             0       0:60:22:29:2c:5d
cge3     9000  0             0       0             0       0:60:22:29:2c:5a
cge4     9000  0             0       0             0       0:60:22:29:2c:58
cge5     9000  0             0       0             0       0:60:22:29:2c:5c
```

Figure 16: Data Mover Network Statistics Output

By repeating this command we can see that the **"*Ibytes*"**, (input bytes), and **"*Obytes*"**, (output bytes) are increasing significantly faster for the primary interface *cge0*. We should also see that these values for *cge1* are not increasing or they are increasing much more slowly than the cge0 interface. This is a good indication that the traffic is going through cge0 rather than cge1.

Stop the network port at the switch and watch the traffic move over to the standby Ethernet interface to confirm that your fail safe networking is properly configured and that the virtual network device continues to operate when the primary interface goes down. You should only do this in a maintenance window since the network failover will impact host connectivity. Should the failover not succeed, it will result in a loss of connectivity to the Celerra by the hosts.

Redundant Paths to Storage

Next, we will check the backend storage. The key point is to verify that you have at least two paths to each storage device so that should one path fail, the Celerra will still be able to access backend storage and your customers will be able to access their files.

Starting at the Celerra side, there should be two HBAs configured and on-line for each data mover.

To verify that each of your data movers has two on-line active HBAs connected to your backend storage use the command:

```
[nasadmin]$ .server_config ALL -v "fcp show"
server_2 : commands processed: 1
command(s) succeeded

FCP ONLINE     HBA 0: S_ID 030057  WWN: 10000000ab12cd34 LP9000   2 GHz
FCP scsi-16:   HBA 0: D_ID 030027 FA-10cb: 50067890abcdef29 Class 3
FCP scsi-64:   HBA 0: D_ID 030014 FA-09cb: 50067890abcdef28 Class 3
FCP ONLINE     HBA 1: S_ID 03004c  WWN: 10000000ab12cd35 LP9000   2 GHz
FCP scsi-0:    HBA 1: D_ID 03000f FA-07cb: 50067890abcdef26 Class 3
FCP scsi-48:   HBA 1: D_ID 030014 FA-08cb: 50067890abcdef27 Class 3
FCP OFFLINE    HBA 2: S_ID 000000  WWN: 10000000abcefc87 LP9020   1 GHz
FCP scsi-32:   HBA 2: CHAINS   32 -   47 OFFLINE
1234567803: ADMIN: 4: Command succeeded:  fcp show

server_3 : commands processed: 1
command(s) succeeded

FCP ONLINE     HBA 0: S_ID 030088  WWN: 10000000ab12cd3a LP9000   2 GHz
FCP scsi-0:    HBA 0: D_ID 030027 FA-10cb: 50067890abcdef29 Class 3
FCP scsi-64:   HBA 0: D_ID 030014 FA-09cb: 50067890abcdef28 Class 3
FCP ONLINE     HBA 1: S_ID 03007c  WWN: 10000000ab12cd3b LP9000   2 GHz
FCP scsi-16:   HBA 1: D_ID 03000f FA-07cb: 50067890abcdef26 Class 3
FCP scsi-48:   HBA 1: D_ID 030014 FA-08cb: 50067890abcdef27 Class 3
FCP OFFLINE    HBA 2: S_ID 000000  WWN: 10000000abcdef8a LP9020   1 GHz
FCP scsi-32:   HBA 2: CHAINS   32 -   47 OFFLINE
1234567893: ADMIN: 4: Command succeeded:  fcp show

server_4 : commands processed: 1
command(s) succeeded

FCP ONLINE     HBA 0: S_ID 030019  WWN: 10000000ab12cd4f LP9000   2 GHz
FCP scsi-16:   HBA 0: D_ID 030027 FA-10cb: 50067890abcdef29 Class 3
FCP scsi-64:   HBA 0: D_ID 030014 FA-09cb: 50067890abcdef28 Class 3
FCP ONLINE     HBA 1: S_ID 03001a  WWN: 10000000ab12cd50 LP9000   2 GHz
FCP scsi-0:    HBA 1: D_ID 03000f FA-07cb: 50067890abcdef26 Class 3
FCP scsi-48:   HBA 1: D_ID 030014 FA-08cb: 50067890abcdef27 Class 3
FCP OFFLINE    HBA 2: S_ID 000000  WWN: 10000000abcdef18 LP9020   1 GHz
FCP scsi-32:   HBA 2: CHAINS   32 -   47 OFFLINE
1234567804: ADMIN: 4: Command succeeded:  fcp showserver_4 : ok
```

Figure 17: Data Mover HBA Status Output on Symmetrix

(Note the "." in front of the command is not a misprint or a flyspeck on the paper, it is part of the command.)

In this example, there are two HBAs connected to the SAN in each data mover. For each data mover, you should see the connected HBAs listed as "*ONLINE*", once for HBA 0 and once for HBA 1. If one or more of your HBAs show up as "*OFFLINE*", then you may have a problem and you should check your SAN cabling and connections as well as the data mover HBA. The offline condition could be caused by a cable problem, a switch

problem, or by a failed HBA on an individual data mover. You may have to replace the entire data mover if the data mover's HBA has failed. (This provides you with an opportunity to verify your failover configuration once the replacement has been completed.) In this configuration, we are not using HBA 2, so it is normally "*OFFLINE*".

The WWN of each HBA on your data movers is the other bit of useful information in this output. This information is handy before you discover that you need the old WWN to modify your WWN zoning during a data mover replacement.

When you run this command in the CLARiiON environment, instead of listing an FA, the output shows SP-A and SP-B connections:

```
[nasadmin]$ .server_config server_2 -v "fcp show"
server_2 : commands processed: 1
command(s) succeeded
output is complete

FCP ONLINE     HBA 0: ALPA 000001   WWN: 5006016030090a0b DX2
FCP scsi-0:    HBA 0: ALPA 0000ef SP-b3: 5006016b10090a2c Class 3
FCP ONLINE     HBA 1: ALPA 000001   WWN: 5006016130090a0b DX2
FCP scsi-16:   HBA 1: ALPA 0000ef SP-a3: 5006016310090a2c Class 3
FCP OFFLINE    HBA 2: ALPA 000001   WWN: 5006016230090a1c DX2
FCP scsi-32:   HBA 2: CHAINS   32 -  47 OFFLINE
FCP OFFLINE    HBA 3: ALPA 000001   WWN: 5006016330090a1c DX2
FCP scsi-48:   HBA 3: CHAINS   48 -  63 OFFLINE
1234981752: ADMIN: 4: Command succeeded:  fcp show
```

Figure 18: Data Mover HBA Status Output on CLARiiON

This example shows the same type of WWN information, but lists connections to SP-b3 and SP-a3 on our backend CLARiiON.

A slightly different syntax for this command will show you what is configured, (persistent), versus what is visible at the time the command is run, (dynamic).

This command is:

```
[nasadmin]$ .server_config ALL -v "fcp bind show"
server_2 : commands processed: 1
command(s) succeeded
output is complete

*** Persistent Binding Table ***
Chain 0000: WWN 50067890abcdef26 HBA 1 FA-07cb Bound
Chain 0016: WWN 50067890abcdef29 HBA 0 FA-10cb Bound
Chain 0032: WWN 5006048000000000 HBA 2  N_PORT Bind Pending
Chain 0048: WWN 50067890abcdef27 HBA 1 FA-08cb Bound
Chain 0064: WWN 50067890abcdef28 HBA 0 FA-09cb Bound
Existing CRC: ea636cac, Actual: ea636cac, CRC Matchs
***   Dynamic  Binding Table ***
Chain 0000: WWN 50067890abcdef26 HBA 1 ID 1 … Pid 0000 S_ID 03000f Sys
Chain 0016: WWN 50067890abcdef29 HBA 0 ID 0 … Pid 0016 S_ID 030027 Sys
Chain 0032: WWN 0000000000000000 HBA 2 ID 2 … Pid 0032 S_ID 000000 Non
Chain 0048: WWN 50067890abcdef27 HBA 1 ID 1 … Pid 0048 S_ID 030014 Non
Chain 0064: WWN 50067890abcdef28 HBA 0 ID 0 … Pid 0064 S_ID 030014 Non
FCP Base Chain: 0  Dump Slot: 2  Dump Chain: 0 16
Adapter Chain Offset 0:0 1:0 2:32  dumpInit 1
1234587990: ADMIN: 4: Command succeeded:  fcp bind show
```

Figure 19: Data Mover HBA Bind Status Output

Ensure that each HBA listed in the "**** Persistent Binding Table ****" has a state of "*Bound* " and that there is a valid, non-zero WWN in the "**** Dynamic Binding Table ****" section of the output. If you see "*Bind Pending* " for HBA 0 or HBA 1 or if the WWN is all zeros, then you should begin to troubleshoot your connection to your backend storage. It is possible to have an HBA listed as "*Bind Pending* " in the Persistent Binding table section, yet still have the correct WWN listed in the Dynamic Binding Table section. Usually this means that both paths to backend storage are active. However, you should investigate further to confirm or correct this situation.

You will also need to understand how your backend storage is connected to determine which HBAs should be Bound. In these examples, HBA 2 is not connected to any backend storage, so its state will normally and correctly be "*Bind Pending"*.

In environments with a CLARiiON backend, the FA information will again be replaced by the CLARiiON SP interface. The same persistent and dynamic binding tables are shown and the same status of Bound or Bind pending will be listed for each HBA.

Now that we have verified that each of our HBAs is on-line and bound, we should verify that each data mover can see at least two paths to each configured storage device. This information is available from the **server_devconfig** command.

```
[nasadmin]$ server_devconfig ALL -list -scsi -all
server_2 :
               Scsi Disk Table
                                    Director   Port
name           addr       num  type  num  sts stor_id         stor_dev
root_disk      c0t0l0     07C  FA    1    On  000387720160    0043
root_disk      c16t0l0    10C  FA    1    On  000387720160    0043
root_ldisk     c0t0l1     07C  FA    1    On  000387720160    0044
root_ldisk     c16t0l1    10C  FA    1    On  000387720160    0044
d3             c0t1l0     07C  FA    1    On  000387720160    004B
d3             c16t1l0    10C  FA    1    On  000387720160    004B
d4             c0t1l1     07C  FA    1    On  000387720160    004C
d4             c16t1l1    10C  FA    1    On  000387720160    004C
d5             c0t1l2     07C  FA    1    On  000387720160    004D
d5             c16t1l2    10C  FA    1    On  000387720160    004D
d6             c0t1l3     07C  FA    1    On  000387720160    004E
d6             c16t1l3    10C  FA    1    On  000387720160    004E
d7             c0t1l4     07C  FA    1    On  000387720160    004F
d7             c16t1l4    10C  FA    1    On  000387720160    004F
d8             c0t1l5     07C  FA    1    On  000387720160    0050
d8             c16t1l5    10C  FA    1    On  000387720160    0050
d9             c0t1l6     07C  FA    1    On  000387720160    0051
d9             c16t1l6    10C  FA    1    On  000387720160    0051
d10            c0t1l7     07C  FA    1    On  000387720160    0052
d10            c16t1l7    10C  FA    1    On  000387720160    0052
```

Figure 20: Listing Celerra Paths to Backend Storage

The output of this command reveals that each named disk is listed twice, once for the first FA connected to this Celerra and again for the second FA. (For CLARiiON backends, each disk is listed twice, but of course, the FA columns are blank.)

Perform one simple comparison on this output; verify that each disk name is listed twice for each data mover. It is also important to verify that your primary and standby data movers see all of the same devices. If you have configured your standby data mover to standby for more than one primary, make sure that the standby can see all of the disks and paths visible to all of the primary data movers. Conversely, you should also make sure that the primary data movers can see all of the disks and paths visible to the standby data mover. This may seem obvious, however, it is possible to have two primary data movers that each see different disks, yet still share the same standby data mover. If the standby data mover sees all of the disks visible to both primary data movers, then failover from primary to secondary will be successful. However, you may not be able to fail back to your primary data mover, since the primary does not see all of the disks seen

by the standby data mover. In earlier versions of DART code before DART v5.5, the **server_standby ALL -verify mover** command would report **"*ok*"** indicating that the initial fail over would be successful, but the Celerra would not fail back due to a device mismatch between the standby and the original primary data mover. This situation may not be possible in newer versions of DART, however, it probably is not something you want to test in a production environment.

What would cause a disk to be listed only once? This could be caused by a number of events. First, not running the **server_devconfig server_X -probe -scsi -all** and **server_devconfig server_X -create -scsi -all** commands for each data mover after provisioning new devices to the Celerra can create a device mismatch. If you provisioned additional devices to the Celerra in our example above where server_2 and server_3 were both using server_4 as a standby and then only ran the **server_devconfig** command with the **-probe** and **-create** option for server_2 and server_4, server_3 would not be able to see the new disks. In this case, server_2 would be able to fail over and fail back with server_4. Also, server_3 would be able to fail over to server_4. However, fail back from server_4 to server_3 would not be possible.

A similar situation results if you zoned an additional backend storage array to HBAs on server_2 and server_4 but did not zone this backend to the HBAs on server_3. Even running the correct **server_devconfig** commands on all data movers would not resolve this situation.

Disk mismatch on the Celerra could also be the result of failure to map devices on your backend Symmetrix to each of the three data movers in our example.

Your procedure to provision additional storage to your Celerra should always include a **server_devconfig** verification step for each data mover in your Celerra, checking that all disks are visible to each data mover and that there are two paths to each disk in each data mover.

Redundant Control Stations

It is not absolutely necessary to have redundant control stations in your Celerra environment. Just be aware that without a functioning control station, a data mover will not automatically failover.

Only one control station at a time can be used to monitor and manage your Celerra if there are redundant control stations in your environment. Please be aware that if you set up jobs to run in cron on a control station, the cron configuration will not move with the active control station. Set up any automation scripts on both primary and secondary control stations, (if a secondary is present), and the script should verify that it is running on the primary control station before trying to retrieve the status of the Celerra.

You can see which slot is the primary control station using the **`/nas/sbin/getreason`** command:

```
[nasadmin]$ /nas/sbin/getreason
10 - slot_0 primary control station
11 - slot_1 secondary control station
 5 - slot_2 contacted
 5 - slot_3 contacted
 5 - slot_4 contacted
```

Figure 21: Displaying the Primary and Secondary Control Station Hardware

When logged into the secondary control station, the /nas file system is not present, making it impossible to run the command from the /nas directory. So, checking that the /nas file system is mounted will confirm that you are working on the primary control station. One way to check if the /nas file system is mounted using the **`mount`** command and searching for the string "**`/nas `**", with a space after the file system name:

```
[nasadmin]$ mount | grep "/nas "
/dev/hda5 on /nas type ext3 (rw,sync)
```

Figure 22: Identifying Primary Data Mover

The return code from grep when the file system is mounted will be 0. If the file system is not mounted, the return code will be 1.

Failover and Failback Procedure

Now that we have checked the Celerra's configuration, we are ready to perform a failover and failback test.

Performing a failover and failback during a planned maintenance window provides much better confidence that everything is properly configured and ready to maintain services in the event of an actual incident. Perform failover and failback checks at regular intervals, as well as after any significant configuration changes have been made in the Celerra environment. Examples of what I would consider a significant change are things like: upgrades or replacement of data movers, rezoning or replacement of SAN switches, backend storage changes including connecting additional backend storage frames or zoning additional FAs to the Celerra and, for the extra cautious, provisioning additional disk devices to the Celerra.

Failover and failback tests have impacts. In the NFS environment, your connected hosts will see a short delay as they process read and write requests. Depending on the configuration of the connected hosts, they may perform automatic recovery tasks such as cluster failover and the like. In the CIFS environment read and write requests interrupted by the failover test may have to be restarted. This depends on the application accessing the Celerra file system. For example, if your customer is saving a large Excel spreadsheet when you start your failover, they will likely receive an error message along the lines of "File Write Failure." If the customer reissues the save command it will most likely be successful since the failover completes fairly quickly. This is just one reason why you want to perform failover testing during a planned maintenance window.

We will cover the basic steps for performing a failover and failback test next. These are just the basic steps; please make sure that you understand the entire process, as well as your environment, before performing this activity.

The first step is to verify that all data movers are in their normal, primary configuration with the **nas_server -list** and the **/nas/sbin/getreason -list** commands.

```
[nasadmin]$ nas_server -list
id      type  acl  slot groupID  state  name
1        1    1211  2              0    server_2
2        1    1211  3              0    server_3
3        4    1211  4              0    server_4
```

Figure 23: Listing Status of Data Movers

This output shows us that all servers are functioning normally and that there are not any faulted data movers in the Celerra. Next, we verify that all data movers are in their proper "contacted" state indicating that the standby data movers are ready to take over for a primary data mover.

```
[nasadmin]$ /nas/sbin/getreason -list
10 - slot_0 primary control station
 5 - slot_2 contacted
 5 - slot_3 contacted
 5 - slot_4 contacted
```

Figure 24: Listing Current Run Status of Data Movers

Before proceeding, you may also want to capture other information such as a list of all mounted file systems and the current file system export configuration. Make sure you capture all the appropriate information from your environment before performing any failover activities. My preference is to capture this information in a temporary file to facilitate comparing the before and after configuration once the failover activities are complete and everything is back to normal.

```
[nasadmin]$ server_mount ALL > server_mount.pre
[nasadmin]$ server_export ALL -list > server_export.pre
```

Figure 25: Capture File System Mount and Export Information to a File

You should be ready to perform the first failover once all of the current configuration information is gathered. In this example, we will failover the primary server_2 to the standby server_4. WARNING: Executing these commands will impact file access activities on your Celerra. Do not proceed unless you have appropriate authorization and have informed your customers!

To initiate the failover, enter the following command and wait while it completes:

```
[nasadmin]$ server_standby server_2 -activate mover
server_2 :
 server_2 : going offline
 server_4 : going active
 replace in progress ...done
 failover activity complete
 commit  in progress (not interruptible)...done

server_2 : renamed as server_2.faulted.server_4
server_4 : renamed as server_2
```

Figure 26: Failing Data Mover over to Standby

Once the failover of server_2 is complete, repeating the status commands shows us the failed over configuration:

```
[nasadmin]$ nas_server -list
id      type  acl  slot groupID  state  name
1        1    1211  2             0    server_2.faulted.server_4
2        1    1211  3             0    server_3
3        4    1211  4             0    server_2
```

Figure 27: Data Mover Status when Failed Over

```
[nasadmin]$ /nas/sbin/getreason -list
10 - slot_0 primary control station
 5 - slot_2 reset
 5 - slot_3 contacted
 5 - slot_4 contacted
```

Figure 28: Data Mover Hardware Status after Failover

This shows us that the failover has been successful and that the standby hardware in slot_4 which was named `server_4` is now named `server_2.` Also, notice that the former primary data mover in slot_2 which had been named `server_2` is now named `server_2.faulted.server_4`. Additionally, we see that the hardware in slot_2 is now in the `reset` state. In this example, the slot_2 hardware did not reboot after the failover and needs to be rebooted before we can fail back to that hardware.

As long as we are failed over to the standby slot_4 hardware, we can repeat the server_mount and server_export commands and verify that all file systems are mounted and exported as they were before.

```
[nasadmin]$ server_mount ALL > server_mount.failover
[nasadmin]$ server_export ALL -list > server_export.failover
```

Figure 29: Capturing File System Mount and Export Information in the Failed Over State

We can verify that the exports and mounts are the same by comparing the **.pre** and **.failover** files:

```
[nasadmin]$ diff server_mount.pre server_mount.failover
[nasadmin]$ diff server_export.pre server_export.failover
```

Figure 30: Comparing Pre and Failed Over Mount and Export Listings

There is no output from the diff command since both pairs of files match. If you do see differences, first check that they are not due to a difference in the order of the lines in the file. If there are other differences, then you should identify and resolve those differences as they may be an indication of an incomplete failover.

Next we need to prepare the hardware in slot_2 to again resume its primary role. Verify that it still is in the `reset` status with the **/nas/sbin/getreason -list** command and if necessary, reboot the slot_2 hardware with the command:

```
[nasadmin]$ server_cpu server_2.faulted.server_4 -r -monitor now
server_2.faulted.server_3 : reboot in progress
0.0.0.0.0.0.0.0.0.0.3.3.3.3.4.
done
```

Figure 31: Rebooting Data Mover after Failover

Since we used the `-monitor` option on the reboot command, the output displayed a string of numbers indicating the hardware run-level as the reboot progressed. You can also watch the reboot status with the **/nas/sbin/getreason -list** command and see it go through the status of `powered off`, then `reset`, (both run-level 0 states), then `loaded`, (run-level 3), then `configured` and finally `contacted`, (run-level 4). Don't be surprised if you miss one or more of these states, some are only visible for a few seconds.

Once the reboot is complete, we can verify that the hardware in slot_2 is ready to resume its primary role:

```
[nasadmin]$ nas_server -list
id      type  acl  slot groupID  state  name
1        1    1211  2             0    server_2.faulted.server_4
2        1    1211  3             0    server_3
3        4    1211  4             0    server_2


[nasadmin]$ /nas/sbin/getreason -list
10 - slot_0 primary control station
 5 - slot_2 contacted
 5 - slot_3 contacted
 5 - slot_4 contacted

[nasadmin]$ server_standby ALL -v mover
server_2.faulted.server_4 : ok
server_3 : ok
server_2 : ok
```

Figure 32: Data Mover Status Check before Failback

Since all slots are in the contacted state, and the standby configuration data base is consistent, we can proceed with the failback process.

```
[nasadmin]$ server_standby server_2 -restore mover
server_2 :
 server_2 : going standby
 server_2.faulted.server_4 : going active
 replace in progress ...done
 failover activity complete
 commit  in progress (not interruptible)...done

server_2 : renamed as server_4
server_2.faulted.server_4 : renamed as server_2
```

Figure 33: Failing Back to Primary Data Mover

Again we should verify that all of the slots are in the contacted state. The hardware in slot_4 may take a minute or so before it reboots and returns to the contacted state, so make sure to give it enough time to reset.

Finally, verify that all of the data movers are in the expected state and that the file systems are mounted and exported as they were before the failover test was started.

```
[nasadmin]$ nas_server -list
id      type  acl  slot groupID  state  name
1        1    1211  2              0    server_2
2        1    1211  3              0    server_3
3        4    1211  4              0    server_4

[nasadmin]$ /nas/sbin/getreason -list
10 - slot_0 primary control station
 5 - slot_2 contacted
 5 - slot_3 contacted
 5 - slot_4 contacted

[nasadmin]$ server_mount ALL > server_mount.post
[nasadmin]$ server_export ALL -list > server_export.post

[nasadmin]$ diff server_mount.pre server_mount.post
[nasadmin]$ diff server_export.pre server_export.post
```

Figure 34: Final Status Check after Successful Failback

With this final verification, our failover test is complete. All servers are named properly, all slots are in the contacted state, all file systems are mounted as they were before the start of the failover activities, and they still have the same exports.

.

Pulling it all Together

Now we have a bunch of commands providing us with all sorts of useful information, and we have verified that our Celerra properly fails over and fails back. What do we do with this information?

For quick notification or status scripts, you can pipe the output of a command to the mail program and direct the results to an Email address or mailbox of your choice. An example of this would be:

```
[nasadmin]$ /nas/sbin/getreason | mail -s "Celerra Status" \
celerra.admin@example.com
```

Figure 35: Directing Command Output to Email

This command will generate no output on the screen, but instead will direct all of the output of the **/nas/sbin/getreason** command to a mail message to be sent to the Email box of **celerra.admin@example.com** with the subject of **Celerra Status**. (The command was split into two separate lines by escaping the line break with a backslash at the end of the first line.) Since the subject is multiple words, it was enclosed in quotes to keep the mail program from interpreting the second word of the subject: **Status** as an Email address.

The Celerra control station also provides many of the resources of a Linux operating system, including shell scripts, Perl scripting, cron and at jobs. All of the commands that are referenced in this article run under the various scripting languages. All of the resulting scripts can be run as a cron or at jobs on the control station. Remember to set up cron jobs on both your primary and secondary control stations if they exist in your environment.

The previous Email example ran immediately, which is helpful if you want to provide Celerra status information to someone else. What if you want to know the status of the Celerra is at 6 am Monday morning, before your Customers start to log in and use their files at 8 am Monday? Combine the "**at**" command with the mail command results in a status check that is Emailed to you a 6 am Monday morning:

```
[nasadmin]$ echo '/nas/sbin/getreason| mail -s "Celerra Status" \
Admin.pager@example.com' | at 0600 monday
```

Figure 36: Scheduling Output to be Emailed Monday Morning

The **/nas/sbin/getreason** command and **mail** commands we used before are enclosed in single quotes, (since we used double quotes for the subject), and that command string is sent to the **at** scheduler. The at scheduler will run the commands once on the next Monday at 6 am, and the results will be sent to the **Admin.pager** mailbox. So, you will receive your status report even if you as stuck in traffic on your way to work.

Now you need to decide how frequently you want to run your configuration or health verification script. Obviously you do not want to run this all the time, since that would just waste control station CPU cycles as well as add another potential for problems into your environment. Additionally, you do not want to have a script sending out a message every

day, since it is likely to get too familiar and be automatically overlooked. But, shortly after a problem occurs, you can count on someone asking when the last time the configuration and health of the Celerra was checked. Keep documentation of your regular checks as well as the output of those checks. That could be a life saver!

The next question is: Where and how should this documentation be stored? Having a file on the Celerra control station may be handy, but will not be much help if your control station is down or inaccessible. Many scripting languages have library routines which allow a script to send out SNMP traps or Emails, as well as creating output files. Clearly define your messages so that "configuration ok" messages can easily be distinguished from warnings or errors requiring more immediate attention.

So where do you start when creating a configuration or health check script? One place is the ConnectHome log. It is possible to parse this log, extract the last ConnectHome event for each transport you have configured. Use the various time libraries to translate the date in the log into something that can be manipulated by your scripting language of choice. This opens up the possibility of customizing an alert for any transport that has not been exercised in the last 30 days. I recommend against adding an automatic ConnectHome test, as tempting as that would be, because ConnectHome testing requires root authority, and having root cron or root at jobs running unnecessarily can create a security vulnerability.

Testing for consistent data mover standby configuration is simple if all you need to do is check for the "ok" output from each data mover standby verification command. As long as you are checking each data mover, you might as well extract the primary and standby relationship for each data mover as well and perhaps even the uptime and current time stamp for each data mover. This resulting report could be stored both on the control station as well as in an Email box or a master configuration repository.

Verifying that all of the HBAs in each of your data movers are online would be another good check to add to your regular configuration check script. There have been instances in my environment where an HBA has gone offline without triggering a ConnectHome event. Without a manual HBA check, an outage could have easily occurred when a switch port failed or cable was inadvertently disconnected. Again, you want to find out about these potential problems before you have an interruption of service!

Checking that each device has two paths to backend storage might be more difficult to automate and harder to include in a simple health check script. However, it should certainly be done each time storage is added, removed or reconfigured.

Regardless of what is included in a regular health check script, whether it is running automatically or manually, consider security if you are writing a script to run in a production environment. Make sure that your script does not run as the root user to reduce the vulnerabilities created by running a regular script and that each portion of your script runs without changing the Celerra's configuration. Testing in a pre-production environment is well worth your time.

Conclusions

What lessons have we learned from all these notifications, commands, and the effort involved in writing a health check script?

We need to understand the NAS environment if we are to document our processes and continuously improve them. The outputs you gather become proof that you are taking reasonable measures to protect your Celerra against failures and accidental configuration problems. This evidence is not useful only after your environment has problems, it is a valuable resource for ongoing support and improving the speed with which you can address current or potential problems. If your process says that you will verify redundant paths to backend storage after each configuration change, then you should gather the evidence of that test and prove it. You should produce a record that you have been following your processes and improving them. Your processes should show that you are taking reasonable precautions and that you are performing regular checks. The evidence gathered in the past becomes input to your future process reviews and improvements.

After the inevitable problem or failure is resolved, you should loop back through your processes and documentation and determine if you can modify them to preemptively identify potential problem areas and resolve issues before they impact your environment.

It is critical to focus on facts and evidence when analyzing a problem. After a failure has occurred or a problem has been encountered, how will you make certain that the environment will not have the same problem in the future? How will you make certain that you have minimized the impact of future failures if it is not possible to eliminate them entirely?

Ultimately the goal is to have your Celerra provide services regardless of what may fail or what else is occurring in your environment. You want your Celerra to be invisible except when the 100% uptime metric is reported at the end of each month, each quarter and each year.

Acknowledgements

I would like to thank all of the EMC NAS support staff, many of whom have worked with me and my team to configure, resolve problems and clarify the configuration requirements of the various Celerras which we support. I have always found them helpful and willing to explain the commands they are using as well as persistent in resolving the problems we have encountered.

I would also like to thank my team members who have worked long hours to install, configure and support all of our EMC equipment.

Biography

Robert Wittig is an Infrastructure Specialist for EDS, an HP company, with over 20 years of System Administration experience working on high availability servers and redundant disk storage. He has 10 years of experience in UNIX System Administration with responsibilities managing Sun, HP, Linux and AIX UNIX systems. He is currently working as a Storage Administrator supporting EMC Celerra, Centera®, CLARiiON and Symmetrix storage frames. Certifications include: EMC Certified Storage Administrator, NAS Specialist, Sun Certified System Administrator, HP UX Certified System Administrator as well as other non-technical certifications. He holds a Bachelor of Science degree in Electrical Engineering from Michigan Technological University.