Automated Balancing of I/O Loads across I/O Paths in an ESX Server 3.x and CLARiiON® Environment

EMC Proven™ Professional Knowledge Sharing 2008

André Rossouw
Advisory Technical Solutions Education Consultant
EMC Corporation
Rossouw_Andre@emc.com

Table of Contents

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

Introduction

Virtualization has become an increasingly important topic. VMware's ESX Server is the undisputed market leader in the server virtualization space; the combination of ESX Server 3.x and EMC CLARiiON storage systems is a very powerful and popular solution in enterprise data centers.

Provisioning storage for ESX Server, and the Virtual Machines that run on it, is simple due to the intuitive nature of EMC Navisphere® Manager. You may use Navisphere Secure CLI, for more complex environments, or where automated provisioning is desired. Due to the CLARiiON's design, and the default behavior of ESX Server, achieving optimal performance through load balancing is a more difficult process.

CLARiiON Design

CLARiiON storage systems consist of a number of modules. This article deals with the components that connect to hosts and control host access to data, the Storage Processors (Saps).

The high-end CLARiiON models that are discussed in this article, the CX-series and CX3-series, are equipped with 2 SPs, named SPA and SPB. Each has 2 or more front-end ports that allow connection to hosts, and back-end Fibre Channel ports that allow connection to disks. Though front-end ports may be either Fibre Channel or iSCSI ports, we will emphasize Fibre Channel ports.

CLARiiON is classified (for ESX Server purposes) as an active-passive storage system. Though both SPs are capable of processing data from hosts at all times, each LUN is owned by only one SP, and all access to the LUN occurs through that SP (the active or owning SP). The peer SP, or non-owning SP, is regarded as the passive SP for that specific LUN. This classification is important when dealing with ESX Server's native path failover software, discussed in the section on ESX Server Multipathing.

ESX Server 3.x

In environments that use ESX Server, end users are likely to see only Virtual Machines (VMs), virtualized hosts that have the look and feel of a physical host. Virtual Machines can use data storage hosted on external storage systems, such as CLARiiON. The actual LUN assignment, though, is to the ESX Server itself.

The ESX Server is connected to the CLARiiON storage system by Fibre Channel Host Bus Adapters (HBAs), the controllers that allow access to external Fibre Channel storage systems. Those connections use a SAN and are not direct from ESX Server to the CLARiiON. Correct zoning of the SAN is an important part of the initial setup, and plays a vital role in the load balancing process.

In ESX Server 3.x environments, best zoning practices include each installed HBA having at least one connection to each CLARiiON SP. Figure 1 shows an example of a single ESX Server with 2 HBAs connected to a CLARiiON CX3-80.

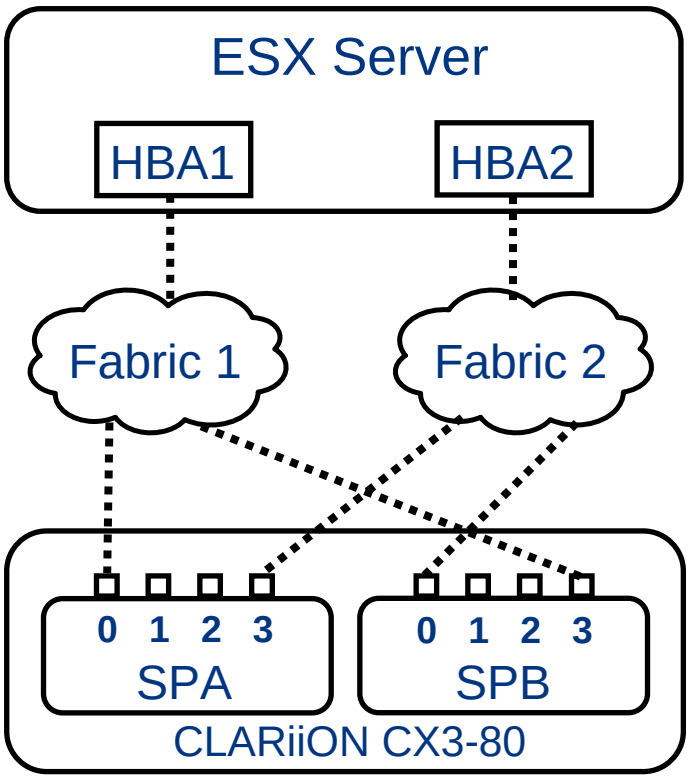

Figure 1

Note that there are a total of 4 paths from the ESX Server to the CLARiiON:

HBA1 -> SPA0
HBA1 -> SPB3
HBA2 -> SPB0
HBA2 -> SPA3

The ESX Server in this example has assigned the names vmhba1 and vmhba2 to HBA1 and HBA2 respectively. SCSI target numbers are assigned to the CLARiiON SP ports by ESX Server; the paths are reported by ESX Server in the format vmhbaX:Y. The 4 paths shown above are therefore displayed as follows:

vmhba1:0	(HBA1 -> SPB3)
vmhba1:1	(HBA1 -> SPA0)
vmhba2:1	(HBA2 -> SPB0)
vmhba2:3	(HBA2 -> SPA3)

In ESX Server 3.x, the paths to a LUN that terminate at the owning SP are described as being 'On', or capable of carrying I/O. Paths that terminate at the non-owning SP are described as being in the 'Standby' state; they will only be used if all the active paths become unavailable, for example as the result of a failure. Due to the internal design of ESX Server, only one of the available (non-standby) paths to any LUN will be active, i.e. carrying I/O.

The LUN number (the Host Logical Unit Number (HLU) assigned to the LUN by Access Logix on the CLARiiON) will be added to the path to produce a LUN address in Controller:Target:LUN (C:T:L) format.

Listing 1 is an example of how paths to a LUN – reported as a 'disk' in ESX Server - are displayed (note that some detail has been removed from this listing):

```
Disk vmhba1:0:0 (5120MB) has 4 paths
10000000c946f605<->5006016b39a01231 vmhba1:0:0 Standby
10000000c946f605<->5006016039a01231 vmhba1:1:0 On active
10000000c946eb36<->5006016839a01231 vmhba2:1:0 Standby
10000000c946eb36<->5006016339a01231 vmhba2:3:0 On
```

Listing 1

The paths reported as ‘On’ are paths that use SPA ports, while the paths reported as ‘Standby’ are paths that use SPB ports. The active path must, by definition, always be a path reported ‘On’.

The ESX Server assigns the lowest-numbered path in the ‘On’ state to a LUN as default. This path assignment occurs when ESX Server is booted, and when new LUNs are added to a running system. Listing 1, above, shows LUN 0, which is owned by SPA; the lowest-numbered path to SPA is vmhba1:1, which is the HBA1 -> SPA0 path. All other LUNs which are owned by SPA will also be assigned to the vmhba1:1 path.

The next partial listing, Listing 2, shows a LUN owned by SPB:

```
Disk vmhba1:0:16 (5120MB) has 4 paths
10000000c946f605<->5006016b39a01231 vmhba1:0:16 On active
10000000c946f605<->5006016039a01231 vmhba1:1:16 Standby
10000000c946eb36<->5006016839a01231 vmhba2:1:16 On
10000000c946eb36<->5006016339a01231 vmhba2:3:16 Standby
```

Listing 2

The lowest-numbered path to SPB is vmhba1:0, which is the HBA1 -> SPB3 path. All other LUNs which are owned by SPB will also be assigned to the vmhba1:0 path.

All LUNs assigned to the ESX Server in this example are therefore accessed through only one of the available HBAs, vmhba1 in this case, and only one of the two available ports on each SP is used to carry I/O. In an environment with a heavy I/O load, the HBA or the SP ports being used could become a performance bottleneck.

ESX Server Multipathing

At present, ESX Server Fibre Channel environments don not support EMC PowerPath®. All path failures are therefore handled by ESX Server's internal multipathing mechanism.

The internal multipathing mechanism classifies external Fibre Channel storage systems as either active-active or active-passive. Storage systems such as the EMC Symmetrix® are known as active-active since a LUN may be accessed through several 'controllers' simultaneously. The CLARiiON, because of its LUN ownership model, is classified as an active-passive storage system, as noted previously.

ESX Server Multipathing makes use of two failover policies, Fixed path, and Most Recently Used path (MRU). If you choose the Fixed policy for a LUN, then one of the paths will be marked as the Preferred path, and that path will be used if it is available. If the Preferred path fails, another path will be used; once the Preferred path becomes available again, ESX Server will fail back to the Preferred path.

If a LUN is assigned the MRU policy, the active path will be labeled as the Preferred path. All I/O will be sent to the LUN through that path. If the path fails, another path will be used; even if the Preferred path becomes available again, ESX Server will not fail back to the Preferred path. Note that the 'Preferred' label is ignored by the MRU policy when selecting a path.

All active-passive storage systems must use the MRU policy for their LUNs to avoid situations where LUNs trespass back and forth, the so-called 'trespass storm'. As a result, no automated failback occurs once a previously used path, which has suffered a failure, becomes available again.

In the event of an SP failure or reboot, LUNs will trespass to the peer SP, and host access will continue. Because of the MRU policy, those LUNs will not trespass back to their default owners when the failed SP has been replaced, and/or the SP reboot completes.

We can upgrade CLARiiON storage systems to newer revisions of software using a Non-Disruptive Upgrade (NDU). To ensure that the process does not disrupt host access to LUNs, one SP is upgraded at a time, and then reboots. It is apparent that the NDU process in an ESX Server environment will leave all LUNs owned by one SP. This could result in performance issues for the CLARiiON and the ESX Server.

Restoring Paths

There are no tools to automate the failback process by restoring LUNs with the MRU policy to their previous paths. The failback process is manual and consists of two separate stages: LUNs must first be moved back to their default owners by means of a trespass, and must then be moved to the correct HBA if they are not already there.

You can trespass LUNs to their default owners using Navisphere Manager or Navisphere Secure CLI. You perform the operation in Navisphere Manager by right-clicking a LUN, then choosing the Trespass… option. Note that this trespasses the LUN from its current owner to the peer, with no regard for which SP is the default owner. In addition, there is no indication which SP is the default owner, that information must be obtained by right-clicking the LUN and viewing its properties. This is a slow process in large environments.

Navisphere Secure CLI also allows LUNs to be trespassed one at a time by specifying the LUN ID in the trespass command. It also has two very powerful switches associated with trespassing, the ‘trespass all’ and ‘trespass mine’ commands. The ‘trespass all’ command trespasses all LUNs to the SP named in the command, while the ‘trespass mine’ command trespasses LUNs owned by the SP named in the command to the SP named in the command. The latter form of the command, run against each SP, will trespass all LUNs to their default owner.

The next step in the process of restoring the LUN to the desired path is to move it to the alternate HBA if required. This step may be performed either from the Virtual Infrastructure Client (VI Client) or from the ESX Server Service Console command line.

From the VI Client, the only way to force a path switch is to disable all non-desired paths to prevent their use. ESX Server will then have to use the only remaining enabled path. This process is slow and potentially dangerous; it can only be performed on one LUN at a time. If the owning SP fails, and all paths to the peer SP are disabled, ESX Server loses access to the LUN. Loss of access to a LUN while the VM using it is running will bring down the VM, and may cause issues with ESX Server itself.

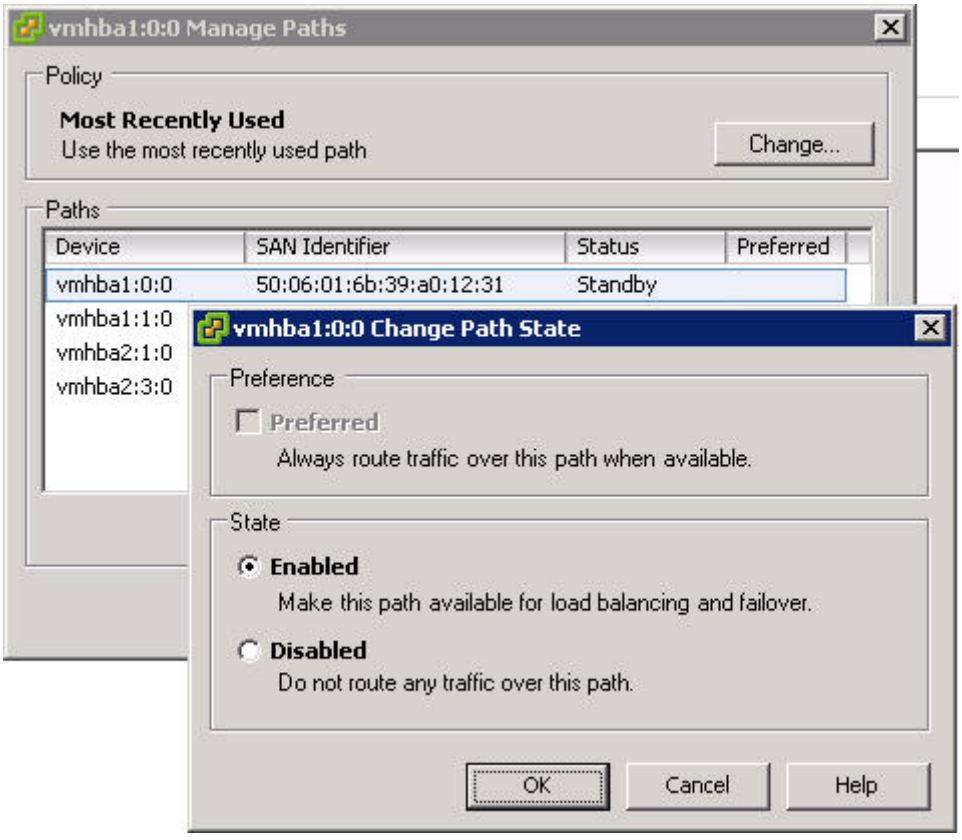

Figure 2

Figure 2 shows the 'Manage Paths' dialog from the VI Client. Note that, as described earlier, only 1 path belonging to 1 LUN may be processed at a time, and the only options are to enable or disable a path. Disabling makes the path unavailable for I/O; enabling it will cause it to become available in either the 'On' or 'Standby' state.

Once the LUN has been failed to the correct HBA, it is important to enable all the paths that were disabled during the failover. This ensures that the LUN will again be protected against path failure. The process for a single LUN takes a number of steps – 3 paths must be disabled, and then re-enabled after the failover. When the number of LUNs becomes large, the time and danger involved dictate the use of the command line at minimum, and preferably a script.

To manipulate LUN paths from the Service Console command line, use the command esxcfg-mpath. This command operates on one path (and one LUN) at a time when used to enable or disable paths. The advantage of using the command line is that it can be scripted; the commands can then be run in succession very quickly, minimizing the probability of loss of LUN access in the event of a failure like the one described above.

The process is identical to that used from the VI Client – first disable all the non-desired paths to force a failover, and then re-enable the paths. Even when the Service Console's ability to recall previous command lines is used, this process is lengthy, and still somewhat dangerous.

Automating the Process

It is apparent, then, that a script should be used once the total number of LUNs exceeds 10 or so. At minimum, the script must perform the following operations:

1. Ensure that the CLARiiON is operating normally
2. Trespass all LUNs to their default owner
3. Balance LUNs across SPA and SPB
4. Balance LUNs across HBAs

Step 1 is required to ensure that the LUN trespasses were not caused by a hardware failure. Once the CLARiiON has a clean bill of health, the CLARiiON will, as Step 2, perform a 'trespass mine' from each SP in turn. The ESX Server must be given time to update the paths; the process is facilitated by forcing a bus rescan.

LUNs will then be balanced across SPs by counting the number of LUNs allocated to this ESX Server that are owned by each SP. If the difference is 2 or greater, then a number of LUNs equal to half of the difference will be trespassed to the SP with the fewest LUNs. The ESX Server will again be forced to rescan the bus to update the path information.

Once LUNs are balanced across SPs, Step 4 will balance the LUNs across HBAs. This operation is performed as described above; disable all paths except the desired one, allow the failover to occur, and then re-enable all paths.

The process makes a number of assumptions, including that Navisphere Secure CLI is installed and in the system path, the SPs are reachable over IP, and that all hardware is operating correctly. The process performs only static balancing; LUNs are balanced across paths with no regard for the amount of I/O sent to each LUN.

Testing the Process

The process and the script were tested on an ESX Server with 2 Emulex LP10000 HBAs connected through a fabric to 4 ports (SPA0, SPA3, SPB0, SPB3) of a CX3-80.
Two 5-disk RAID 5 Groups were created, and assigned RAID Group IDs of 5 and 6.

LUNs 500 through 515 were bound on RAID Group 5, and LUNs 600 through 615 were bound on RAID Group 6. CLARiiON defaults for ownership were left in place; as a result, LUNs 500 through 515 were assigned to SPA, and LUNs 600 through 615 were assigned to SPB. (LUNs from an odd-numbered RAID Group are owned by SPA, and LUNs from an even-numbered RAID Group are owned by SPB if you use the default setting).

We added the LUNs to an ESX Server Storage Group, and Access Logix™ was permitted to assign the host LUN IDs automatically. LUNs 500 through 515 were assigned HLUs of 0 through 15 respectively, and LUNs 600 through 615 were assigned HLUs of 16 through 31 respectively.

The esxcfg-mpath –l command was run on the ESX Server to show the paths. I have edited the output to remove listings for internal storage and the LUNZ entries. Only the first and last entries for each SP are shown; other LUNs owned by the SP have the same path assignment:

Disk vmhba1:0:0 /dev/sdb (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:0 Standby preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:0 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:0 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:0 On

…

Disk vmhba1:0:15 /dev/sdi (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:15 Standby preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:15 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:15 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:15 On

Disk vmhba1:0:16 /dev/sdj (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:16 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:16 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:16 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:16 Standby

…

Disk vmhba1:0:31 /dev/sdaa (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:31 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:31 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:31 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:31 Standby

Listing 3

The paths of the last 8 LUNs were than changed manually by disabling the active path, and allowing the LUN to fail over to the alternate path – a path to the same SP, but via the other HBA. The output has again been edited to remove listings for the internal storage and the LUNZ entries. Only the first and last entries are shown for the 8 LUNs that were failed over; the remaining 6 have the same path assignments:

Disk vmhba1:0:24 /dev/sds (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:24 Off preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:24 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:24 On active
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:24 Standby

…

Disk vmhba1:0:31 /dev/sdaa (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:31 Off preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:31 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:31 On active
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:31 Standby

Listing 4

Note that the LUNs with HLUs 24 through 31 now show the previously active path, HBA1 to port SPB3, as disabled (off), and have failed to the path from HBA2 to port SPB0. The listing was captured before the paths were re-enabled.

All paths were then enabled, and the ESX Server was rebooted. The esxcfg-mpath –l command was run again. The output has again been edited to remove listings for the internal storage and the LUNZ entries. Once again, only the first and last LUNs are shown for SPA and SPB:

```
Disk vmhba1:0:0 /dev/sdb (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:0 Standby  preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:0 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:0 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:0 On

…

Disk vmhba1:0:15 /dev/sdq (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:15 Standby  preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:15 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:15 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:15 On

Disk vmhba1:0:16 /dev/sdr (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:16 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:16 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:16 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:16 Standby

…

Disk vmhba1:0:31 /dev/sdag (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:31 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:31 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:31 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:31 Standby
```

Listing 5

Note that the LUN assignment has returned to what it was when the LUNs were first assigned to the ESX Server; all LUNs are on the lowest numbered available path. LUNs owned by SPA use path vmhba1:1, the path from HBA1 to port SPA0, while LUNs owned by SPB use path vmhba1:0, the path from HBA1 to port SPB3.

It is obvious, therefore, that LUNs are not evenly distributed across the available paths – all LUNs are being accessed through one HBA, and only one of the two available ports per SP is being used. Note that if there were more HBAs and more SP ports in use, the situation would be no different, only one HBA and one port per SP would be active.

In environments where the workload is slight, it is possible that this imbalance would not cause any performance problems and the customer might be unaware that the load is not evenly distributed. Once the workload becomes significant, however, the HBA or the SP ports being used for the active path could become a bottleneck. An additional concern in very large environments is that too many hosts sending I/O to one SP port could cause buffer overflows on the SP port – the 'queue full' condition.

The script was then run against the specific CLARiiON on the ESX Server Service Console. Listing 6 shows the output, edited to remove LUNZ and internal disk entries. The 32 LUNs are now divided into 4 groups of 8 LUNs, with each group assigned to a different path. No LUNs were trespassed to the non-owning SP.

```
Disk vmhba1:0:0 /dev/sdb (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:0 Standby  preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:0 On
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:0 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:0 On active

…

Disk vmhba1:0:7 /dev/sdae (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:7 Standby  preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:7 On
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:7 Standby
```

FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:7 On active

Disk vmhba1:0:8 /dev/sdaf (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:8 Standby preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:8 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:8 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:8 On

…

Disk vmhba1:0:15 /dev/sdi (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:15 Standby preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:15 On active
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:15 Standby
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:15 On

Disk vmhba1:0:16 /dev/sdj (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:16 On preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:16 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:16 On active
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:16 Standby

…

Disk vmhba1:0:23 /dev/sdr (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:23 On preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:23 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:23 On active
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:23 Standby

Disk vmhba1:0:24 /dev/sds (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:24 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:24 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:24 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:24 Standby

…

```
Disk vmhba1:0:31 /dev/sdaa (5120MB) has 4 paths and policy of Most Recently Used
FC 2:12.0 10000000c946f605<->5006016b39a01231 vmhba1:0:31 On active preferred
FC 2:12.0 10000000c946f605<->5006016039a01231 vmhba1:1:31 Standby
FC 3:11.0 10000000c946eb36<->5006016839a01231 vmhba2:1:31 On
FC 3:11.0 10000000c946eb36<->5006016339a01231 vmhba2:3:31 Standby
```

Listing 6

Only the first and last LUNs in each group of 8 are shown. The listing illustrates that static balancing has been successful.

The script

The script starts by getting a list of all the HBAs in the ESX Server. The command used lists the HBA WWN as well as its vmhba designation. This information is used to populate a hash.

```
my @hba_info = `esxcfg-mpath -a`;   # command to get HBA information
for (@hba_info)
{
   if (/^(vmhba\d+)\s+(\w{16})\s+.+/)
   {
      $vmhba{$1} = $2;
   }
}
```

Code Fragment 1

The next step is to trespass all LUNs on the specified CLARiiON to their default owner. We made the assumption that LUNs were previously balanced across SPs by the user (or by a script such as this one), and that any trespasses that have occurred are due to a prior failure or administrative action.

To ensure that both SPs have a true picture of current LUN ownership, a ‘getlun’ command is sent to each to refresh the LUN information.

```
# start by trespassing LUNs to their rightful owner
print "Trespassing LUNs to their default owners ...\n";
system("naviseccli -h $spa trespass mine > /dev/null");
system("naviseccli -h $spb trespass mine > /dev/null");

# force a poll
system("naviseccli -h $spa getlun -uid > /dev/null 2>&1");  system("naviseccli -h $spb getlun -uid
> /dev/null 2>&1");
```

Code Fragment 2

Once the CLARiiON has completed the trespass process, the ESX Server must perform a rescan to refresh its view of the LUN ownership. The script waits for 60 seconds to allow any ESX Server failover processes to complete, then rescans all HBAs.

```
print "Waiting 60 seconds for ESX Server to fail over ...\n";
for my $i (1 .. 60)
{
   print ".";
   sleep 1;
}
print "\n";

# make sure ESX Server gets the picture
rescan("all");

sleep 10;
```

Code Fragment 3

CLARiiON utilities such as Navisphere Secure CLI usually use Array Logical Unit numbers (ALUs) in their commands, while ESX Server utilities are more likely to use the LUN's globally Unique ID (UID), also known as the LUN WWN. The script therefore performs a getlun operation to get a list of ALUs and their respective UIDs, and then populates a hash with them to help with the translation of ALUs to UIDs. Because both the ALUs and UIDs are unique for any given CLARiiON, reversing the hash is a safe procedure; a reversed hash is saved to aid translation from UID to ALU when required.

```
my @getlun_output = `naviseccli -h $spa getlun -uid`;
for (@getlun_output)
{
   if (/^LOGICAL UNIT NUMBER\s+(\d+)/)
   {
      $lun_id = $1;
   }

   if (/^UID:\s+((?:[\da-f]{2}:){15}[\da-f]{2})/i)
   {
      ($lun_uid = $1) =~ s/://g;
      $lun_id2uid{$lun_id} = $lun_uid;
   }
}

%lun_uid2id = reverse %lun_id2uid;

# get this CLARiiON's WWN
my @output = `naviseccli -h $spa getarrayuid`;
for (@output)
{
   if (/.+\s+(?:[\da-f]{2}:){4}[\da-f](.+)/i)
   {
      ($sp_uid = $1) =~ s/://g;
   }
}
```

Code Fragment 4

The script then calls a subroutine, imaginatively named get_esx_info, to get the LUN information from an ESX Server perspective. This subroutine gets a full listing of all the storage available to the ESX Server, removes the internal storage from the list, and then translates the WWPNs into SP port numbers. The subroutine tracks each LUN by LUN ID, canonical name, and active path.

```
sub get_esx_info                # get ESX Server LUN information
{
   my @mpath_verbose = `esxcfg-mpath -v -l`;

   for (@mpath_verbose)
   {
      my $string;

      if (/^Disk\s+(vmhba\d+:\d+:\d+)\s+vml\.[\da-f]{10}(6006016[\da-f]{25})[\da-
f]{12}\s+(\/dev\/sd\w+)\s+.+\d+\w+.+(\d+)\s+paths.+policy of\s+(.+)/i)
      {
         if (exists $lun_uid2id{uc $2})
         {
            $lun_number = $lun_uid2id{uc $2};
            if ($5 eq "Most Recently Used") {$policy = "MRU"}
            $lun2canonical{$lun_number} = $1;
            print "\n$1 [LUN $lun_number]: $4 paths policy  $policy\n" if $verbose;
         }
      }

      if (/FC.+[\da-f]{16}:[\da-f]{16}<->5006016([\da-f])[\da-f]$sp_uid:[\da-
f]{16}\s+(vmhba\d+):\d+:\d+\s+(.+)/i)
      {
         if ($1 le "7")
         {
            $port = "SPA$1"
         }
         elsif (uc $1 ge "A")
         {
            $port = "SPB" . chr((ord(uc($1)) - 15))
```

```
            }
            elsif ($1 le "9") {$port = "SPB" . ($1 - "8")}

            $hba = $2;
            $path_state = $3;
            $path = "$hba -> $port";
            push @paths, $path if not $seen{$path};
            $seen{$path}++;

            print "$path : $path_state\n" if $verbose;
# increment the active counter for this path
            if ($path_state =~ /Active/i)
            {
               $lun_path{$path}++;
               $active_path = $path;
               $string = "LUN $lun_number UID $lun_id2uid{$lun_number} $hba $port $active_path";
               $lun_hba{$hba}++;
               $lun_port{$port}++;
               $port =~ /SPA/ ? push @spa_luns, $lun_number : push @spb_luns, $lun_number;
            }

            if ($path_state =~ /Preferred/i)
            {
               $preferred_path = $path;
            }

         }
         $string .= " $preferred_path";
         push @active, $string;
      }

      @sort_active = sort @active;
}
```

Code Fragment 5

Once the LUN information is available, the script checks the difference between the number of LUNs owned by SPA and the number owned by SPB. If the difference is 2 or greater, LUNs will be trespassed in order to balance the assignment. Code Fragment 3 shows the calculation of the number of LUNs to trespass to SPA if SPB currently owns more LUNs. Similar code trespasses LUNs to SPB as required.

```
{
   $luns2trespass = int(($num_spb_luns - $num_spa_luns) / 2);
   $dest_sp = $spa;
   @luns2move = @spb_luns;
   $default_owner = 0;     # SPA
}
```

Code Fragment 6

Once the LUNs are trespassed, the script balances LUNs across paths on a per SP basis. Code Fragment 7 shows part of the process. In this code fragment, paths other than the desired path are disabled.

```
for my $lun (@luns2move)
   {
      system("esxcfg-mpath -q --lun=$lun2canonical{$lun}");
      my @paths = ` esxcfg-mpath -q --lun=$lun2canonical{$lun}`;
      for (@paths)
      {
         if (/FC\s+.+[\da-f]{16}<->[\da-f]{16}\s+(vmhba\d+:\d+:\d+)\s+.+active.+/)
         {
            system("esxcfg-mpath --path=$1 --state=off --lun=$lun2canonical{$lun}");
         }
      }
         system("esxcfg-mpath -q --lun=$lun2canonical{$lun}");
   }
```

Code Fragment 7

Once the paths are balanced, the script prints the final output. An example was shown above in Listing 6.

Proposed enhancements

While static load balancing is better than no load balancing at all, it does not take into account the I/O load related to the individual LUNs. A future enhancement will measure the I/O load to the LUNs for a user-defined period, then balance LUNs across SPs and paths according to the measured workload.

Another enhancement would balance workloads across HBAs when 2 or more CLARiiONs are connected to the same ESX Server. This type of environment is less common than a single CLARiiON environment; as a result, support was not included in the first version of the script.

Summary

CLARiiON storage systems and ESX Server are becoming a popular combination in enterprise data centers. ESX Server does not balance I/O workloads across all paths by default; the behavior of ESX Server Multipathing and the CLARiiON NDU behavior make the problem worse.

Manual methods can be used to balance LUNs across paths, but are time-consuming and potentially dangerous. This article introduces a script that will automatically balance LUNs across all available paths.

Author's Biography

André has worked in the IT industry since CP/M was a state of the art operating system for small computers. His roles have included Technical Support Engineer for a repair center environment, Customer Service Engineer, course developer, and instructor. André is an EMC Proven Professional Expert in the TA and IE CLARiiON tracks. He lives in North Carolina with his wife, daughter, and a garden full of squirrels.