rageek

A place for Unix Thoughts and Ideas

San booting Sparc Systems with Emulex HBAs

1 Comment Posted by jflaster on May 24, 2012

While booting from the SAN is more common on domain'd systems, it is less used on the smaller systems.

I have been using it quite a bit over that last year as part of converting my existing servers from UFS->ZFS and upgrading to Solaris 10u9 to support Oracle 11gR2.

Here is the process for enabling SAN Boot on Sparc Systems with Emulex HBAs.

This will change your fibre device paths and you will want to do this before creating any live upgrade environments on the SAN

Here are the Steps:

1. Enable boot_code setting in HBAnywhere or emlxadm on all your adapters. This crucial for enabling the sfs boot support. In practice, it may be advisable to powercycle the server/domain after enabling this.
This can be found, checked, and set through emlxadm:

```
root@testserver # /opt/EMLXemlxu/bin/emlxadm -iall -y boot_code

Found 2 HBA ports.

HBA 1: /devices/pci@1,700000/emlx@0/fp@0,0

Boot code: Disabled

HBA 2: /devices/pci@3,700000/emlx@0/fp@0,0

Boot code: Disabled
```

```
root@testserver # /opt/EMLXemlxu/bin/emlxadm -iall -y boot_code enable

Found 2 HBA ports.

HBA 1: /devices/pci@1,700000/emlx@0/fp@0,0

Boot code: Enabled

HBA 2: /devices/pci@3,700000/emlx@0/fp@0,0

Boot code: Enabled
```

2. Set set-sfs-boot- setting in openboot. This property must be set on each fibre adapter and will not be available unless boot code has already been enabled. Cycling the server may be necessary for this option to show up. Here is a example of enabling it on a Netra T2000 and a M4000. I have also read that this can be set through HBA Anywhere.

Enabling this setting will change the device paths of the fibre controllers on reboot and you may want to take caution and be prepared to resolve any device path issues.

The device paths will change from being reported as lpfc to emlx, again you'll want to do this prior to creating live upgrade environments, otherwise those environment may not boot properly.

Device Paths Before:

```
root@testserver # prtdiag -v | egrep 'fibre|lpfc|emlx'

01  PCIe  17      2, fe00, 10df       2,  0,  0  okay     4,    4  lpfc-pciex10df,fe00
LPe11000-M4

    /pci@11,700000/lpfc@0

01  PCIe  19      2, fe00, 10df       2,  0,  0  okay     4,    4  lpfc-pciex10df,fe00
LPe11000-M4

    /pci@13,700000/lpfc@0
```

```
root@testserver $ prtconf | egrep -i 'fibre|lpfc|emlx'

                        lpfc, instance #0

                        lpfc, instance #1
```

On some systems it will be listed as "fibre-channel" instead of lpfc

```
root@testserver # prtdiag  | egrep 'fibre|lpfc|emlx'

IOBD/PCIX    PCIX IOBD    /pci@7c0/pci@0/pci@9/pci@0/fibre-channel@1 fibre-channel-
pci10df,fd+LP11000-M4

root@testserver # prtconf | egrep -i 'fibre|lpfc|emlx'

                    fibre-channel, instance #0

                    fibre-channel, instance #1
```

Device Paths After:

```
root@testserver # prtdiag -v | egrep 'fibre|lpfc|emlx'

01  PCIe  17      2, fe00, 10df       2,  0,  0  okay     4,    4  emlx-pciex10df,fe00
LPe11000-M4

    /pci@11,700000/emlx@0

01  PCIe  19      2, fe00, 10df       2,  0,  0  okay     4,    4  emlx-pciex10df,fe00
LPe11000-M4

    /pci@13,700000/emlx@0
```

```
root@testserver # prtconf | egrep -i 'fibre|lpfc|emlx'

                    emlx, instance #2

                    emlx, instance #3
```

In the open boot prom, you'll be using the show-disks command to get the paths of the fibre cards. If the cards do not show up, double check and make sure that the boot code setting is enabled.

Here is the process on a T2000, it should be the same on other T-series.

```
{0} ok show-disks

a) /pci@7c0/pci@0/pci@9/pci@0,2/lpfc@1/sd

b) /pci@7c0/pci@0/pci@9/pci@0/lpfc@1/sd

c) /pci@7c0/pci@0/pci@1/pci@0/ide@8/cdrom

d) /pci@7c0/pci@0/pci@1/pci@0/ide@8/disk

e) /pci@780/pci@0/pci@9/scsi@0/disk

q) NO SELECTION

Enter Selection, q to quit:

{0} ok select /pci@7c0/pci@0/pci@9/pci@0,2/lpfc@1

{0} ok set-sfs-boot

Flash data structure updated.
Signature         4e45504f
Valid_flag              4a
Host_did                 0
Enable_flag              5
SFS_Support              1
Topology_flag            0
Link_Speed_flag          0
Diag_Switch              0
Boot_id                  0
Lnk_timer                f
Plogi-timer              0
LUN (1 byte)             0
DID                      0
WWPN              0000.0000.0000.0000
LUN (8 bytes)     0000.0000.0000.0000
*** Type reset-all to update. ***

{0} ok select /pci@7c0/pci@0/pci@9/pci@0/lpfc@1
```

```
{0} ok set-sfs-boot

Flash data structure updated.

Signature         4e45504f
Valid_flag              4a
Host_did                 0
Enable_flag              5
SFS_Support              1
Topology_flag            0
Link_Speed_flag          0
Diag_Switch              0
Boot_id                  0
Lnk_timer                f
Plogi-timer              0
LUN (1 byte)             0
DID                      0
WWPN              0000.0000.0000.0000
LUN (8 bytes)     0000.0000.0000.0000

*** Type reset-all to update. ***

{0} ok reset-all
```

Here is the process on a M4000. For the select-dev command, note that there is a space entered before the device path, after the quote.

```
{0} ok show-disks
a) /pci@2,600000/lpfc@0,1/sd
b) /pci@2,600000/lpfc@0/sd
c) /pci@0,600000/pci@0/pci@9/lpfc@0,1/sd
d) /pci@0,600000/pci@0/pci@9/lpfc@0/sd
e) /pci@0,600000/pci@0/pci@8/pci@0/scsi@1/disk
q) NO SELECTION
Enter Selection, q to quit: b
/pci@2,600000/lpfc@0/sd has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
         for creating devalias mydev for /pci@2,600000/lpfc@0/sd
{0} ok " /pci@2,600000/lpfc@0" select-dev
{0} ok set-sfs-boot
Flash data structure updated.

Signature         4e45504f
Valid_flag              4a
Host_did                 0
Enable_flag              5
SFS_Support              1
Topology_flag            0
Link_Speed_flag          0
Diag_Switch              0
Boot_id                  0
Lnk_timer                f
Plogi-timer              0
LUN (1 byte)             0
DID                      0
WWPN              0000.0000.0000.0000
LUN (8 bytes)     0000.0000.0000.0000
```

```
*** Type reset-all to update. ***

{0} ok " /pci@0,600000/pci@0/pci@9/lpfc@0" select-dev
{0} ok set-sfs-boot
Flash data structure updated.
Signature          4e45504f
Valid_flag               4a
Host_did                  0
Enable_flag               5
SFS_Support               1
Topology_flag             0
Link_Speed_flag           0
Diag_Switch               0
Boot_id                   0
Lnk_timer                 f
Plogi-timer               0
LUN (1 byte)              0
DID                       0
WWPN               0000.0000.0000.0000
LUN (8 bytes)      0000.0000.0000.0000

*** Type reset-all to update. ***

{0} ok  reset-all
```

3. On reboot you will want to do a devfsadm -C to clear up the device tree and create the new device paths and you may need to reboot a second time to clear up anything in veritas.

Once this is done, you can now create and boot from Live Upgrade environments on the SAN.

Like this:

Like
Be the first to like this.
Solaris Solaris
← Downloading Solaris images from Oracle using wget Customizing Your Solaris 11 Auto Installer builds using a first boot script →

One Response to *San booting Sparc Systems with Emulex HBAs*

1. **emulexalex** June 22, 2012 at 12:11 pm

Great Post! We are always looking for how customers use Emulex Technology and this is a great example. Keep it up.

–Alex
http://www.implementerslab.com

Reply