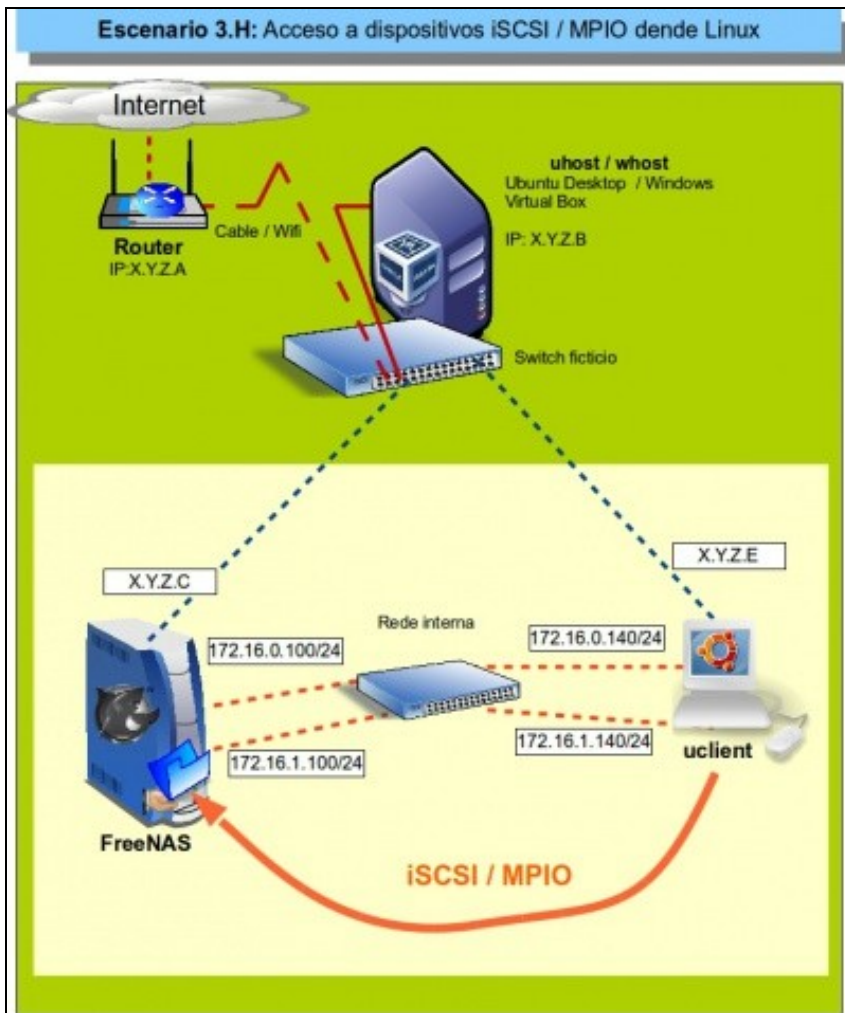


Escenario 3.H: Acceso a dispositivos iSCSI a través de MPIO desde Linux

- Neste escenario 3.H imos configurar un Linux, neste caso **uclient**, para que acceda ao dispositivo iSCSI da NAS a través de dous interfaces de rede, facendo uso de MPIO.



Sumario

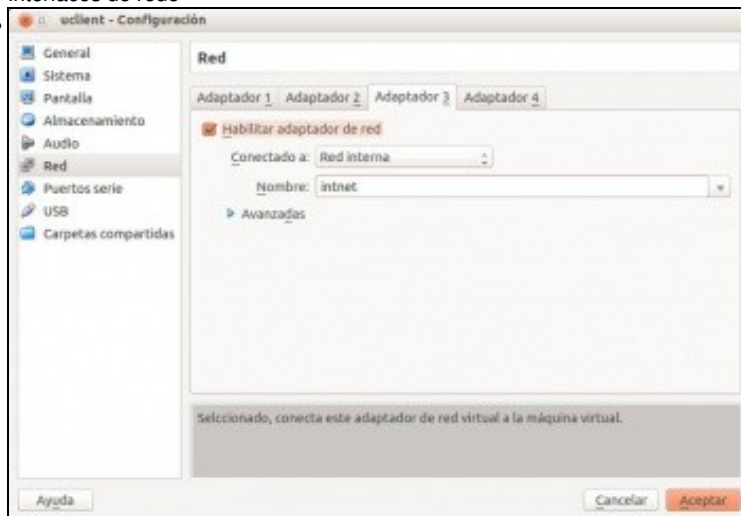
- 1 Introducción
- 2 Configuración dos interfaces de rede de uclient
- 3 Nodos dispoñibles/activos no cliente (iniciador)
- 4 Descubrir e activar o destino iSCSI
- 5 Configuración Multipath IO (MPIO)
- 6 Realización de probas
- 7 Cambiar o destino iSCSI (MPIO). Varios destinos
- 8 Documentación

Introdución

- A modo e resumo, vanse realizar os seguintes pasos:
 - ♦ Borrar os destinos que se tiñan de antes para poder comezar de cero.
 - ♦ Instalar o paquete **multipath-tools**: <http://christophe.varoqui.free.fr/>
 - ◊ Ao acceder ao un destino iSCSI por varias rutas distintas, no cliente vanse mostrar tantos discos como rutas haxa ao mesmo destino.
 - ◊ *Multipath-tools* vai facilitar un único dispositivo de bloque para un destino accesible por varias rutas (MPIO).
 - ◊ Permitirá reencamiñar o tráfico co destino cando se perda un dos camiños e recuperarse o antes posible desa perda.

Configuración dos interfaces de rede de uclient

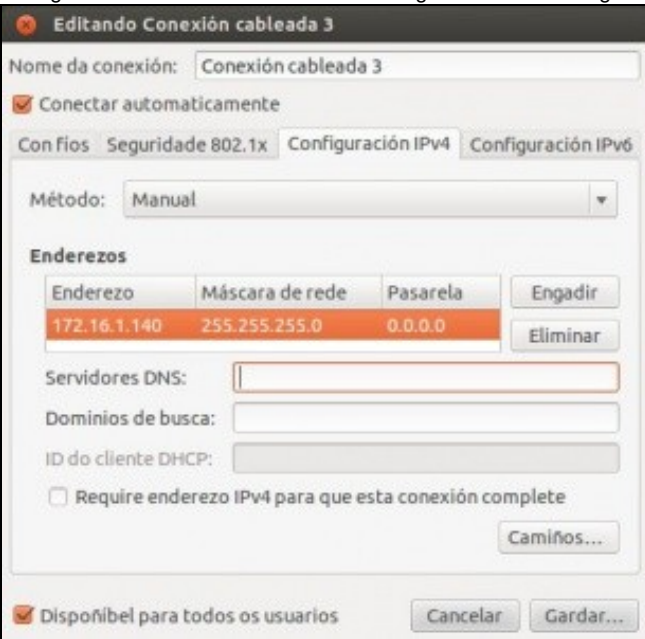
- Interfaces de rede



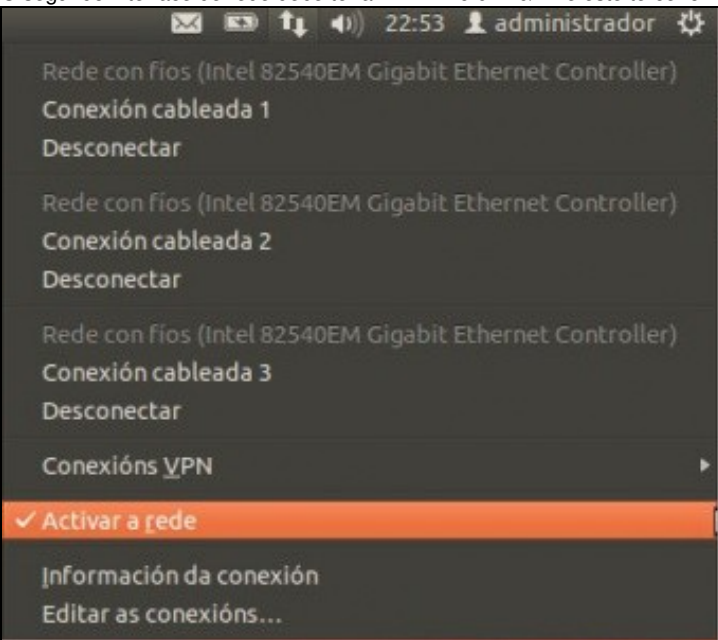
Engadir á MV uclient os adaptadores 2 e 3 en modo Rede Interna.



Configurar o terceiro interface de rede. A segunda xa viña configurada do escenario anterior, en calquera caso ...

- 

O segundo interface de rede debe ter a IP 172.16.0.140/24 e este terceiro 172.16.1.140/24.

- 

Desactivamos a rede e volvémosla a activar para que teña en conta a nova configuración.

```
administrador@uclient: ~
administrador@uclient:~$ ping 172.16.0.100
PING 172.16.0.100 (172.16.0.100) 56(84) bytes of data.
64 bytes from 172.16.0.100: icmp_seq=1 ttl=64 time=0.911 ms
64 bytes from 172.16.0.100: icmp_seq=2 ttl=64 time=0.724 ms
64 bytes from 172.16.0.100: icmp_seq=3 ttl=64 time=0.581 ms
^C
--- 172.16.0.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.581/0.738/0.911/0.138 ms
administrador@uclient:~$
administrador@uclient:~$ ping 172.16.1.100
PING 172.16.1.100 (172.16.1.100) 56(84) bytes of data.
64 bytes from 172.16.1.100: icmp_seq=1 ttl=64 time=0.975 ms
64 bytes from 172.16.1.100: icmp_seq=2 ttl=64 time=0.594 ms
64 bytes from 172.16.1.100: icmp_seq=3 ttl=64 time=0.524 ms
^C
--- 172.16.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.524/0.697/0.975/0.200 ms
administrador@uclient:~$
```

Probamos a conectividade contra as interfaces 172.16.X.100 da NAS.

Nodos dispoñibles/activos no cliente (iniciador)

- Revisar en que situación se atopan os nodos/destinos que se viñan usando en uclient.

- Estado dos nodos

```
root@uclient: ~
root@uclient:~# iscsiadm -m node
10.0.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:disco2
10.0.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:disco1
10.0.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:disco3
root@uclient:~#
root@uclient:~# iscsiadm -m session
iscsiadm: No active sessions.
root@uclient:~#
root@uclient:~#
```

Miramos que nodos hai dispoñibles de modo permanente (os 3 destinos de antes) e cales están activos. Ningún está activo porque non se admiten iniciadores de rede 10.0.0.0/8 que é por onde accedían estes nodos.

```
root@uclient: ~
root@uclient:~# rm -r /etc/iscsi/nodes/
root@uclient:~#
root@uclient:~# rm -r /etc/iscsi/send_targets/
root@uclient:~#
root@uclient:~# ls -l /etc/iscsi/
total 16
-rw-r--r-- 1 root root 348 Jan 28 23:00 initiatorname.iscsi
-rw-r--r-- 1 root root 11768 Jan 28 23:05 iscsid.conf
root@uclient:~#
```

Borramos as carpetas **nodes** e **send_targets**, para así comezar dende cero.

```
root@uclient: ~
GNU nano 2.2.6 Ficheiro: /etc/fstab

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=dfeaf08-143d-acf1-8dcs-258df34ae0ba / ext4 errors=remount-ro 0 1
# swap was on /dev/sda5 during installation
UUID=79b01182-e916-4058-ade7-fcf640cc66c9 none swap sw 0 0

# /dev/sdb1 /media/disco1 ntfs _netdev 0 0
# /dev/sdc1 /media/disco2 ntfs _netdev 0 0
# /dev/sdd1 /media/disco3 ntfs _netdev 0 0
```

Comentamos os puntos de montaxe aos dispositivos iSCSI en /etc/fstab.

Descubrir e activar o destino iSCSI

- Descubrir e activar os nodos

```
root@uc1ient: ~
root@uc1ient:~# iscsiadm -n discovery -t sendtargets -p 172.16.0.100
172.16.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
172.16.1.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
root@uc1ient:~#
root@uc1ient:~# iscsiadm -n discovery -t sendtargets -p 172.16.1.100
172.16.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
172.16.1.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
root@uc1ient:~#
root@uc1ient:~#
```

Descubrimos os destinos que ofrece a NAS, tanto por unha IP como pola outra ofrecen os mesmos destinos: **2 rutas para o mesmo destino**. Dá igual por cal se busque.

```
root@uc1ient: ~
root@uc1ient:~# ls /etc/iscsi/nodes/iqn.2014-01.ga.cursosv.istg:discompio/
172.16.0.100,3260,1 172.16.1.100,3260,1
root@uc1ient:~#
root@uc1ient:~# ls /etc/iscsi/send_targets/
172.16.0.100,3260 172.16.1.100,3260
root@uc1ient:~#
root@uc1ient:~# ls /etc/iscsi/send_targets/172.16.0.100,3260/
st_config
root@uc1ient:~# cat /etc/iscsi/send_targets/172.16.0.100,3260/st_config
# BEGIN RECORD 2.0-873
discovery.startup = manual
discovery.type = sendtargets
discovery.sendtargets.address = 172.16.0.100
discovery.sendtargets.port = 3260
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = cursosv
discovery.sendtargets.auth.password = abc123456789.
discovery.sendtargets.timeo.login_timeout = 15
discovery.sendtargets.use_discoveryd = No
discovery.sendtargets.discoveryd_poll_inval = 30
discovery.sendtargets.reopen_max = 5
discovery.sendtargets.timeo.auth_timeout = 45
discovery.sendtargets.timeo.active_timeout = 30
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 32768
# END RECORD
root@uc1ient:~#
```

Na base de datos dos destinos permanentes (**nodos**) vese que para o mesmo destino hai 2 rutas, isto é, 2 cartafoles. Tamén se pode observar o contido do ficheiro **st_config**.

```
root@uc1ient: ~
root@uc1ient:~# cat /etc/iscsi/nodes/iqn.2014-01.ga.cursosv.istg:discompio/172.16.1.100,3260,1/default
# BEGIN RECORD 2.0-873
node.name = iqn.2014-01.ga.cursosv.istg:discompio
node.tgt = 1
node.startup = automatic
node.leading_login = No
iface.iscsi.ifacename = default
iface.transport_name = tcp
iface.vlan_id = 0
iface.vlan_priority = 0
iface.iface_num = 0
iface.mtu = 0
iface.port = 0
node.discovery_address = 172.16.1.100
node.discovery_port = 3260
node.discovery_type = send_targets
node.session.initial_cmds = 0
node.session.initial_login_retry_max = 8
node.session.xmlt_thread_priority = -20
node.session.cnds_max = 128
node.session.queue_depth = 32
node.session.nr_sessions = 1
node.session.auth.authmethod = CHAP
node.session.auth.username = cursosv
node.session.auth.password = abc123456789.
node.session.timeo.replacement_timeout = 120
node.session.err_timeo.abort_timeout = 15
node.session.err_timeo.lu_reset_timeout = 30
node.session.err_timeo.tgt_reset_timeout = 30
node.session.err_timeo.host_reset_timeout = 60
node.session.iscsi.PastAbort = Yes
```


E o ficheiro **default** dun dos camiños ao destino. Observar que o parámetro **node.startup** está posto a automático, para que cando se acenda o equipo xa estableza a conexión co destino.

```
root@uc1lent:~#
node.session.auth.username = curso5v
node.session.auth.password = abc123456789
node.session.tlneo.replacement_timeout = 120
node.session.err_tlneo.abort_timeout = 15
node.session.err_tlneo.lu_reset_timeout = 30
node.session.err_tlneo.lgt_reset_timeout = 30
node.session.err_tlneo.host_reset_timeout = 60
node.session.iscsi.FastAbort = Yes
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.session.iscsi.DefaultTime2Retain = 0
node.session.iscsi.DefaultTime2Wait = 2
node.session.iscsi.MaxConnections = 1
node.session.iscsi.MaxOutstandingR2T = 1
node.session.iscsi.ERL = 0
node.conn[0].address = 172.16.1.100
node.conn[0].port = 3260
node.conn[0].startup = manual
node.conn[0].tcp.window_size = 524288
node.conn[0].tcp.type_of_service = 0
node.conn[0].tlneo.logout_timeout = 15
node.conn[0].tlneo.login_timeout = 15
node.conn[0].tlneo.auth_timeout = 45
node.conn[0].tlneo.noop_out_interval = 5
node.conn[0].tlneo.noop_out_timeout = 5
node.conn[0].iscsi.MaxXmitDataSegmentLength = 0
node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144
node.conn[0].iscsi.HeaderDigest = None
node.conn[0].iscsi.DataDigest = None
node.conn[0].iscsi.IFMarker = No
node.conn[0].iscsi.OFMarker = No
# END RECORD
root@uc1lent:~#
```

Máis abaixo no ficheiro, o campo **node.conn[0].address** indica á IP do destino á que se conecta unha das rutas e no ficheiro **default** da outra ruta ...

```
root@uc1lent:~#
node.session.auth.username = curso5v
node.session.auth.password = abc123456789
node.session.tlneo.replacement_timeout = 120
node.session.err_tlneo.abort_timeout = 15
node.session.err_tlneo.lu_reset_timeout = 30
node.session.err_tlneo.lgt_reset_timeout = 30
node.session.err_tlneo.host_reset_timeout = 60
node.session.iscsi.FastAbort = Yes
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.session.iscsi.DefaultTime2Retain = 0
node.session.iscsi.DefaultTime2Wait = 2
node.session.iscsi.MaxConnections = 1
node.session.iscsi.MaxOutstandingR2T = 1
node.session.iscsi.ERL = 0
node.conn[0].address = 172.16.0.100
node.conn[0].port = 3260
node.conn[0].startup = manual
node.conn[0].tcp.window_size = 524288
node.conn[0].tcp.type_of_service = 0
node.conn[0].tlneo.logout_timeout = 15
node.conn[0].tlneo.login_timeout = 15
node.conn[0].tlneo.auth_timeout = 45
node.conn[0].tlneo.noop_out_interval = 5
node.conn[0].tlneo.noop_out_timeout = 5
node.conn[0].iscsi.MaxXmitDataSegmentLength = 0
node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144
node.conn[0].iscsi.HeaderDigest = None
node.conn[0].iscsi.DataDigest = None
node.conn[0].iscsi.IFMarker = No
node.conn[0].iscsi.OFMarker = No
# END RECORD
root@uc1lent:~#
```

... amosa a outra IP pola cal está accesible o destino na NAS.

```
root@ucient: ~
root@ucient:~# iscsiadm -m node
172.16.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
172.16.1.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
root@ucient:~#
root@ucient:~# iscsiadm -m session
iscsiadm: No active sessions.
root@ucient:~#
root@ucient:~# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5
root@ucient:~#
```

Con **iscsiadm -m node** comprobamos que nodos hai dispoñibles permanentemente. Por agora non hai ningunha sesión activa (**iscsiadm -m session**), e os discos que se teñen son os locais.

```
root@ucient: ~
root@ucient:~# iscsiadm -m node --loginall=all
Logging in to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.0.100,3260] (multiple)
Logging in to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260] (multiple)
Login to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.0.100,3260] successful.
Login to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260] successful.
root@ucient:~#
root@ucient:~# iscsiadm -m session
tcp: [1] 172.16.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
tcp: [2] 172.16.1.100:3260,1 iqn.2014-01.ga.cursosv.istg:discompio
root@ucient:~#
root@ucient:~# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sda5 /dev/sdb /dev/sdb1 /dev/sdc /dev/sdc1
root@ucient:~#
```

Con **iscsiadm -m node --loginall=all** activamos os nodos dispoñibles en nodes. Comprobamos que agora as sesións están activas: para o mesmo dispositivo iSCSI temos 2 rutas. En realidade é como se se tiveran 2 destinos; ao facer **ls /dev/sd*** pódese ver que se teñen 2 discos novos (sdb e sdc) cada un coa súa partición.

Configuración Multipath IO (MPIO)

- A última imaxe anterior non proporciona redundancia pois, non se ten un dispositivo de bloque que nos permita escribir nel e que este se encargue de escribir en sdb ou sdc indistintamente. Se nós escribimos en sdb sempre non estamos conseguindo usar a outra ruta e por tanto non temos balanceo nin redundancia.
- Para iso vén socorrernos o paquete **multipath-tools**.

- Configurar MPIO

```
root@ucient: ~
root@ucient:~# apt-get install multipath-tools
Lendo as listas de paquetes... Feito
Construíndo a árbore de dependencias
Lendo a información do estado... Feito
Instalaranse os seguintes paquetes extra:
  kpartx libaio1
Paquetes suxeridos:
  multipath-tools-boot
Os seguintes paquetes NOVOS hanse instalar:
  kpartx libaio1 multipath-tools
0 anovados, 3 instalados, Vanse retirar 0 e delxar 189 sen anovar.
Ten que recibir 199 kB de arquivos.
Despois desta operación ocuparanse 811 kB de disco adicionais.
Quere continuar [S/n]? S
```

Instalar o paquete **multipath-tools**: `apt-get install multipath-tools`.

```
root@uclient: ~  
root@uclient:~# multipath -ll  
3300000000cia91f93 dn-0 FreeBSD,iSCSI Disk  
size=5.0G features='0' hwhandler='0' wp=rw  
|-+- policy='round-robin 0' prio=1 status=active  
|  - 3:0:0:0 sdb 8:16 active ready running  
|-+- policy='round-robin 0' prio=1 status=enabled  
|  - 4:0:0:0 sdc 8:32 active ready running  
root@uclient:~#
```

Executar **multipath -ll**, e veremos entre outras cousas: O **número de serie** que ten o disco (hai que copialo para usar a continuación), información sobre a NAS, o tamaño do disco, o algoritmo de balanceo no uso das rutas que vai usar (**round-robin**), etc. (Se non aparece nada, reiniciése a máquina *uclient*)

Vese que un disco está activo pero o outro non.

```
GNU nano 2.2.6 Fichero: /etc/multipath.conf Modificado  
defaults {  
    udev_dir                /dev  
    polling_interval        10  
    selector                "round-robin 0"  
    path_grouping_policy    multibus  
    getuid_callout          "/lib/udev/scsi_id --whitelisted --device=/dev/%n"  
    prio_callout            /bin/true  
    path_checker            readsector0  
    prio                    const  
    rr_min_io               100  
    rr_weight               priorities  
    fallback                immediate  
    no_path_retry           fail  
    user_friendly_name      yes  
}  
blacklist {  
    devnode "sda"  
    devnode "^([a-z]|loop|fd|nd|dn|sr|scd|st)[0-9]*"  
    devnode "^hd[a-z]([0-9])*"  
    devnode "^vd[a-z]([0-9])*"  
    devnode "^cciss:[0-9]d[0-9]*[p[0-9]]*"   
}  
multipaths {  
    multipath {  
        wwid 3300000000cia91f93  
        alias open-e-test  
    }  
}  
Gardar o búfer modificado (SE RESPONDA «Non» PERDERANSE OS CAMBIOS)?  
S Si  
N Non  C Cancelar
```

Creamos/editamos o ficheiro **/etc/multipath.conf** e pegamos o contido que se pon ao final deste punto, modificando o **wwid** (World Wide ID) co número de serie obtido na execución da instrución *multipath -ll*. O **alias** é o nome co que se vai identificar o novo **disco virtual** asentado sobre os discos reais (sdb e sdc).

```
root@uclient: ~  
root@uclient:~# service multipath-tools restart  
* Stopping multipath daemon multipathd [ OK ]  
* Starting multipath daemon multipathd [ OK ]  
root@uclient:~#
```

Reiniciar o servizo **multipath-tools**.

```
root@uclient:~# multipath -ll  
open-e-test (3300000000cia91f93) dn-0 FreeBSD,iSCSI Disk  
size=5.0G features='0' hwhandler='0' wp=rw  
|-+- policy='round-robin 0' prio=1 status=active  
|  - 3:0:0:0 sdb 8:16 active ready running  
|  - 4:0:0:0 sdc 8:32 active ready running  
root@uclient:~#
```


Se agora se executa **multipath -ll** vese que os dous discos están activos, as 2 rutas están operativas e balanceadas co algoritmo round-robin.

```
• Disk /dev/mapper/open-e-test: 5368 MB, 5368709120 bytes
255 heads, 63 sectors/track, 652 cylinders, total 10485760 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disk identifier: 0x952aca82

    Device Boot      Start         End      Blocks   Id  System
/dev/mapper/open-e-test1  2048     10481663     5239808    7   HPFS/NTFS/exFAT

Disk /dev/mapper/open-e-test-part1: 5365 MB, 5365563392 bytes
255 heads, 63 sectors/track, 652 cylinders, total 10479616 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 1048576 bytes
Disk identifier: 0x7270d0f
```

Executamos **fdisk -l** e vemos que aparece un novo dispositivo **/dev/mapper/open-e-test** cunha partición chamada **/dev/mapper/open-e-test-part1** formatada en NTFS (Que era como estaba o disco2). Este novo dispositivo é o que está asentado sobre sdb e sdc ao mesmo tempo, o mesmo pasa coa partición. Recibe este nome **open-e-test...** porque é como se identificou no campo **alias** do ficheiro **multipath.conf**.


```
• root@uclient:~# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root 10, 236 Feb  2 14:09 control
brw-rw---- 1 root disk 252,  0 Feb  2 15:21 open-e-test
brw-rw---- 1 root disk 252,  1 Feb  2 15:21 open-e-test-part1
root@uclient:~#
root@uclient:~#
```

O contido de mapper ... A carpeta **/dev/mapper** é unha especie de portal para mapear dispositivos de bloque. Observar como se amosa o nome do alias no canto do wwid, (isto verase máis adiante).

```
• root@uclient:~# mkdir /media/discompio
root@uclient:~#
root@uclient:~# mount /dev/mapper/open-e-test-part1 /media/discompio/
root@uclient:~#
root@uclient:~#
```

Creamos unha carpeta para montar o novo disco MPIO: **mkdir /media/discompio**. Montamos a partición do mapper: **mount /dev/mapper/open-e-test-part1 /media/discompio**.

```
• root@uclient: ~
root@uclient:~# ls /media/discompio/
disco2 no disco2  disco2 NOVA no disco2
root@uclient:~#
```



Vemos o contido que xa tiña o disco e ademais podemos crear unha carpeta nova.

```
defaults {
    udev_dir                /dev
    polling_interval        10
    selector                "round-robin 0"
    path_grouping_policy    multibus
    getuid_callout          "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
    prio_callout            /bin/true
    path_checker            readsector0
    prio                    const
    rr_min_io               100
    rr_weight               priorities
    failback                immediate
    no_path_retry           fail
    user_friendly_name      yes
}
blacklist {
    devnode "sda"
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^hd[a-z] [[0-9]*]"
    devnode "^vd[a-z]"
    devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
}
multipaths {
    multipath {
        wwid 33000000031c05120 (Cambiar este número polo obtido con multipath -ll e
                                borrar este comentario)
        alias open-e-test (quen o desexe pode cambiar o nome do alias que se lle vai
                           dar ao wwid, eliminar tamén este comentario entre paréntese)
    }
}
```

Realización de probas

- A continuación vanse realizar unas probas de caídas de enlaces ou rutas.
- Para iso vanse desconectar o cable nos adaptadores de rede ou ben na NAS ou ben co cliente (*uclient*).
- **IMPORTANTE:** se por calquera motivo se tivera que volver a descubrir ou activar os dispositivos iSCSI, asegurarse de que se teñen conectados todos os cables de rede nos 2 equipos: NAS e uclient.

- Configurar MPIO

```
administrador@uclient: ~
administrador@uclient:~$ dd if=/dev/zero of=file1G bs=1G count=1
1+0 records in
1+0 records out
1073741824 bytes (1,1 GB) copiados, 23,2879 s, 46,1 MB/s
administrador@uclient:~$
```

Creamos no directorio actual un ficheiro de 1 GB no equipo local (uclient) para ser transferido á nas a usando MPIO:

dd if=/dev/zero of=file1G bs=1G count=1

Para mirar o funcionamento do comando **dd**: http://es.wikipedia.org/wiki/Dd_%28Unix%29

```
eth1      Link encap:Ethernet  HWaddr 08:00:27:ff:b3:a2
          inet addr:172.16.0.140  Bcast:172.16.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feff:b3a2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4709 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2659 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4396260 (4.3 MB)  TX bytes:263104 (263.1 KB)

eth2      Link encap:Ethernet  HWaddr 08:00:27:e6:1b:fe
          inet addr:172.16.1.140  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee6:1bfe/64 Scope:link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2595 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1700 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2032022 (2.0 MB)  TX bytes:200799 (200.7 KB)
```

Revisamos a configuración IP das dúas interfaces polos cales se vai encamiñar o tráfico. Neste caso son eth1 e eth2, que cada quen revise cales son na súa situación.

```
administrador@ucient: ~$ ifconfig eth1 | grep bytes
administrador@ucient: ~$ ifconfig eth2 | grep bytes
```

grep bytes"

filtrando con **grep** a saída do comando **ifconfig**.

O usuario pode estar usando interfaces de rede distintas a eth1 e eth2.

```
administrador@ucient: ~$ watch -n1 'echo eth1 && ifconfig eth1 | grep bytes && echo eth2 | grep bytes'
Every 1.0s: echo eth1 && ifconfig eth1 | grep bytes &&... Sun Feb  2 15:50:28 2014
eth1
eth2
RX bytes:4410284 (4.4 MB) TX bytes:273544 (273.5 KB)
RX bytes:2046046 (2.0 MB) TX bytes:211239 (211.2 KB)
```

A imaxe amosa que apenas se transmitiron e recibiron uns poucos MB.

```
administrador@ucient: ~$ cp file1G /media/discompio/
administrador@ucient: ~$ watch -n1 'echo eth1 && ifconfig eth1 | grep bytes && echo eth2 | grep bytes'
Every 1.0s: echo eth1 && ifconfig eth1 | grep bytes &&... Sun Feb  2 15:55:31 2014
eth1
eth2
RX bytes:9655847 (9.6 MB) TX bytes:112422740 (112.4 MB)
RX bytes:6689698 (6.6 MB) TX bytes:96236498 (96.2 MB)
```

Na primeira consola comezamos a copia do ficheiro: **cp file1G /media/discompio**.

Na segunda consola vese como a transmisión vai máis ou menos parella por cada interface.



Pero ... desconectamos o cable de rede do adaptador 2 (ou 3) de uclint.

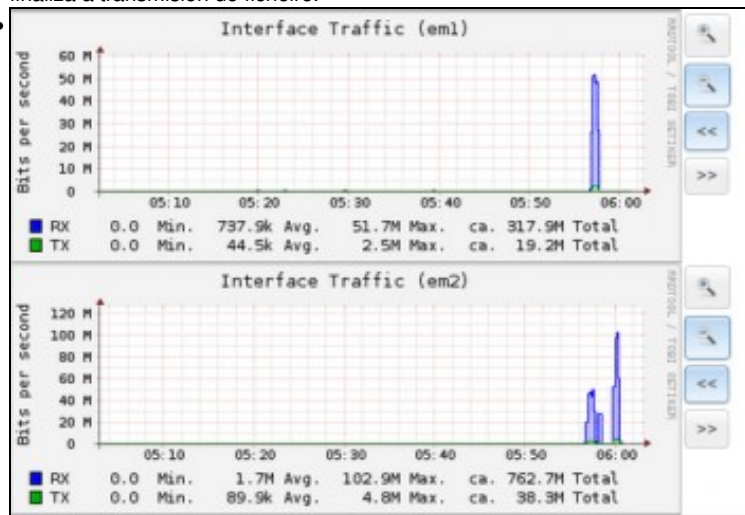
```
administrador@ucient: ~$ cp file1G /media/discompio/
administrador@ucient: ~$ watch -n1 'echo eth1 && ifconfig eth1 | grep bytes && echo eth2 | grep bytes'
Every 1.0s: echo eth1 && ifconfig eth1 | grep bytes &&... Sun Feb  2 15:58:37 2014
eth1
eth2
RX bytes:20160983 (20.1 MB) TX bytes:333429332 (333.4 MB)
RX bytes:34080208 (34.0 MB) TX bytes:675994720 (675.9 MB)
```

Pasa o tempo, até que pasan 120 segundos e se restablece a transmisión neste caso só pola interface eth2, e...

```
administrador@ucient:~$ cp file1G /media/discompio/
administrador@ucient:~$
administrador@ucient:~$
Every 1.0s: echo eth1 && ifconfig eth1 | grep bytes &&... Sun Feb 2 15:59:23 2014

eth1
RX bytes:20160983 (20.1 MB) TX bytes:333429332 (333.4 MB)
eth2
RX bytes:40210188 (40.2 MB) TX bytes:790898168 (799.8 MB)
```

finaliza a transmisión do ficheiro.



Esta é a gráfica e o informe da NAS de uso de em1 e em2 (os interfaces de rede), que podemos ver picando no botón **Informe** da barra superior. Obsérvese como despois dun intervalo no que as dúas interfaces recibiron datos de forma simultánea, logo pasou a recibir unicamente em2.

```
administrador@ucient:~$
GNU nano 2.2.6 Ficheiro: /etc/iscsi/lscsid.conf Modificado
# *****
# See the iSCSI REAME's Advanced Configuration section for tips
# on setting timeouts when using multipath or doing root over iSCSI.
#
# To specify the length of time to wait for session re-establishment
# before failing SCSI commands back to the application when running
# the Linux SCSI Layer error handler, edit the line.
# The value is in seconds and the default is 120 seconds.
# Special values:
# - If the value is 0, IO will be failed immediately.
# - If the value is less than 0, IO will remain queued until the session
# is logged back in, or until the user runs the logout command.
node.session.timeout.replacement_timeout = 30
#
# To specify the time to wait for login to complete, edit the line.
# The value is in seconds and the default is 15 seconds.
node.conn[0].timeout.login_timeout = 15
#
# To specify the time to wait for logout to complete, edit the line.
# The value is in seconds and the default is 15 seconds.
node.conn[0].timeout.logout_timeout = 15
#
# Time interval to wait for on connection before sending a ping.
node.conn[0].timeout.noop_out_interval = 5
#
# To specify the time to wait for a Nop-out response before failing
# the connection, edit this line. Failing the connection will
# cause IO to be failed back to the SCSI layer. If using dm-multipath
# this will cause the IO to be failed to the multipath layer.
#
# Obter axud Grawar Ler Fich Páxina ant CortarText PosicAct
# Seir Xustificir U-lo? Páxina seg RepórTexto Ortografía
```

Pero 120 segundos é moito esperar!!!. O parámetro que hai que modificar é **node.session.timeo.replacement_timeout = 120**, pero xa sabemos que se o cambiamos en **/etc/iscsi/iscsid.conf** quedará para posibles descubrimentos futuros pero non para os nodos presentes.

```
root@uclient: ~  
root@uclient:~# umount /media/discompio  
root@uclient:~#
```

Antes de cambiar nada nos nodos desmontamos o disco MPIO.

```
root@uclient: ~  
root@uclient:~# nano /etc/iscsi/nodes/iqn.2014-01.ga.cursosv.istg:discompio  
/172.16.1.100\,3260\,1/default
```

Co cal cambiamos ese parámetro no ficheiro **default** de cada un dos nodos polos que chegamos ao mesmo dispositivo iSCSI.

```
GNU nano 2.2.6 Ficheiro: ...ompio/172.16.1.100,3260,1/default Modificado  
  
# BEGIN RECORD 2.0-873  
node.name = iqn.2014-01.ga.cursosv.istg:discompio  
node.tpgt = 1  
node.startup = automatic  
node.leading_login = No  
iface.iscsi_ifacename = default  
iface.transport_name = tcp  
iface.vlan_id = 0  
iface.vlan_priority = 0  
iface.iface_num = 0  
iface.mtu = 0  
iface.port = 0  
node.discovery_address = 172.16.1.100  
node.discovery_port = 3260  
node.discovery_type = send_targets  
node.session.initial_cmds_n = 0  
node.session.initial_login_retry_max = 8  
node.session.xmit_thread_priority = -20  
node.session.cmds_max = 128  
node.session.queue_depth = 32  
node.session.nr_sessions = 1  
node.session.auth.authmethod = CHAP  
node.session.auth.username = cursosv  
node.session.auth.password = abc123456789.  
node.session.timeo.replacement_timeout = 30  
node.session.err_timeo.abort_timeout = 15
```

Poñemos por exemplo 30 segundos. O usuario pode probar cal pode ser o máis baixo que pode poñer e que se recupere a transmisión tanto no corte como na recuperación dun enlace.


```
• root@uc1lent:~  
root@uc1lent:~# service open-iscsi restart  
* Unmounting iscsi-backed filesystems [ OK ]  
* Disconnecting iSCSI targets  
Logging out of session [sid: 3, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260]  
Logout of [sid: 3, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260] successful.  
* Stopping iSCSI initiator service [ OK ]  
* Starting iSCSI initiator service iscsid [ OK ]  
* Setting up iSCSI targets  
Logging in to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.0.100,3260] (multiple)  
Logging in to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260] (multiple)  
Login to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.0.100,3260] successful.  
Login to [iface: default, target: iqn.2014-01.ga.cursosv.istg:discompio, portal: 172.16.1.100,3260] successful.  
* Mounting network filesystems [ OK ]  
root@uc1lent:~#  
root@uc1lent:~# service multipath-tools restart  
* Stopping multipath daemon multipathd [ OK ]  
* Starting multipath daemon multipathd [ OK ]  
root@uc1lent:~#  
root@uc1lent:~# mount /dev/napper/open-e-test-part1 /media/discompio/  
root@uc1lent:~#  
root@uc1lent:~# █
```

Reiniciar iSCSI: **service open-iscsi restart**

Reiniciar Multipath: **service multipath-tools restart**

Volver a montar o disco MPIO.

E a probar ...

- Se o usuario o desexa pode xogar cos seguintes valores, antes de implantar un sistema en produción.
- Sempre desmontar o disco antes dos cambios e logo reiniciar os servizos e volver a montar o disco.

```
# Time interval to wait for on connection before sending a ping.  
node.conn[0].timeo.noop_out_interval = 5
```

```
# To specify the time to wait for a Nop-out response before failing  
# the connection, edit this line. Failing the connection will  
# cause IO to be failed back to the SCSI layer. If using dm-multipath  
# this will cause the IO to be failed to the multipath layer.  
node.conn[0].timeo.noop_out_timeout = 5
```

```
# To specify the time to wait for abort response before  
# failing the operation and trying a logical unit reset edit the line.  
# The value is in seconds and the default is 15 seconds.  
node.session.err_timeo.abort_timeout = 15
```

```
# To specify the time to wait for a logical unit response  
# before failing the operation and trying session re-establishment  
# edit the line.  
# The value is in seconds and the default is 30 seconds.  
node.session.err_timeo.lu_reset_timeout = 20
```

Cambiar o destino iSCSI (MPIO). Varios destinos

- Unha das cousas que se pode desexar é conectarse a outro destino distinto ao que se ten.
- Este novo destino terá o seu **wwid** distinto do do destino actual.
- Na seguinte exemplo creamos un novo dispositivo iSCSI na NAS: **iscsi-sr**, descubrímolos (discovery) e conectámolos (loginall) como nos exemplos anteriores.
- Cambiar Descubrir e activar os nodos

```

root@ucilent:~# iscsiadm -m node sessions
172.16.0.100:3260,1 iqn.2014-01.ga.cursosv.istg:iscsi-sr
172.16.1.100:3260,1 iqn.2014-01.ga.cursosv.istg:iscsi-sr
root@ucilent:~#
root@ucilent:~# multipath -ll
330000000399d053a dn-2 FreeBSD,iSCSI Disk
size=200G features='0' hwhandler='0' wp=rw
'-+ policy='round-robin 0' prio=1 status=active
|- 0:0:0:0 sdc 8:32 active ready running
|- 8:0:0:0 sdb 8:16 active ready running
root@ucilent:~#
root@ucilent:~# ls -l /dev/mapper/
total 0
lrwxrwxrwx 1 root root    7 Feb  2 16:37 330000000399d053a -> ../dn-2
crw----- 1 root root 10, 236 Feb  2 14:09 control
root@ucilent:~#
root@ucilent:~# cat /etc/multipath.conf | grep wwid
wwid 330000000c1a91f93
root@ucilent:~#

```

iscsiadm -m node sessions amosa as sesións activas; vemos que hai dúas rutas ao mesmo destino.

multipath -ll amosa, entre outras cousas, o **wwid** asociado ao novo dispositivo de almacenamento iSCSI.

Con **ls -l /dev/mapper** vemos os enlaces aos dispositivos de bloque multicamiño.

Observar que aparece o **wwid** e non o alias; iso é porque o **wwid** asociado ao novo dispositivo de almacenamento non coincide co que está no arquivo de configuración **/etc/multipath.conf**.

```

root@ucilent:~#
root@ucilent:~# tail -n 6 /etc/multipath.conf
multipaths {
    multipath {
        wwid 330000000399d053a
        alias disco_mpio
    }
}
root@ucilent:~#
root@ucilent:~# service multipath-tools restart
 * Stopping multipath daemon multipathd [ OK ]
 * Starting multipath daemon multipathd [ OK ]
root@ucilent:~#
root@ucilent:~# multipath -ll
disco_mpio (330000000399d053a) dn-2 FreeBSD,iSCSI Disk
size=200G features='0' hwhandler='0' wp=rw
'-+ policy='round-robin 0' prio=1 status=active
|- 0:0:0:0 sdc 8:32 active ready running
|- 8:0:0:0 sdb 8:16 active ready running
root@ucilent:~#
root@ucilent:~# ls -l /dev/mapper/
total 0
crw----- 1 root root 10, 236 Feb  2 14:09 control
lrwxrwxrwx 1 root root    7 Feb  2 16:46 disco_mpio -> ../dn-2
root@ucilent:~#

```

Cambiamos o **wwid** no ficheiro de configuración **/etc/multipath.conf** co **wwid** do novo dispositivo iSCSI. Incluso cambiamos o **alias** por un novo, ... simplemente para adaptar o alias aos nosos intereses.

Reiniciamos o servizo multipath: **service multipath-tools restart**. Vemos que o novo alias xa está como apuntando ao novo dispositivo de bloque multicamiño.

- No caso de querer ter varios destinos distintos iSCSI, todos eles accesibles por MPIO entón hai que achar o **wwid** de cada un deles e logo configurar o ficheiro **/etc/multipath.conf** creando unha entrada **multipath{}** co **wwid** asociado ao dispositivo iSCSI e o alias que desexemos.

Documentación

- A meirande parte da documentación usada para elaborar este apartado é do seguinte enlace e adaptada ás nosas circunstancias:
http://kb.open-e.com/Setup-a-Ubuntu-Server-with-iSCSI-and-MPIO-to-connect-to-an-iSCSI-Tartget-in-DSS-V7_1653.html