iSCSI : Storage barato y fácil de usar

Comúnmente compartimos directorios con el fin de que uno o varios clientes puedan escribir y leer de forma paralela, y es a mi parecer una solución práctica. El problema radica en que para realizar dicha tarea necesitamos un servicio adicional (ej, NFS, Samba, entre otros) el cual genera una capa de abstracción que se encarga de traducir las peticiones y orquestar los accesos a disco, perdiendo muchísimo performance y desperdiciando recursos a destajo. Hagan la prueba: escriban sobre un recurso NFS compartido montado y se darán cuenta que la velocidad es muchísimo mas lenta si la comparamos a estar escribiendo directamente en una partición montada. Peor aún si tienen hardware viejo, no van a llegar a unos cuantos megas por segundo de escritura.

Arquitectura básica

Como supongo que ya saben, SCSI (*Small Computer System Interface*, *http://www.ietf.org/rfc/rfc3720.txt*) es la interfaz estándar para transferencia de datos entre periféricos usando el bus del computador, osea es por ejemplo la forma de transmitir datos entre el disco duro y la placa madre. Ahora, imagínense transmitir dicho protocolo por internet, podríamos compartir directamente una partición (o logical volume) que se encuentre en nuestra máquina y permitir que otra máquina remota pueda verlo como un simple dispositivo de bloques, utilizando la red para transferir la data y eliminando la capa de abstracción que generaría un servicio de por medio. La escalabilidad de acceso al dispositivo es tan amplia que incluso podríamos darle acceso a la máquina remota para que pueda formatear la partición en cuestión. A nivel de arquitectura, el driver iSCSI provee las mismas funciones que un canal de fibra, pero con la diferencia de que en vez de utilizar una HBA, utilizamos una simple tarjeta Ethernet. Claramente si necesitamos una solución de alta disponibildad y queremos eliminar todos los puntos de fallo sólo necesitaremos realizar un bonding de las interfaces de red y listo.

El mercado

El driver iSCSI apareció hace algunos años como alternativa a las soluciones de storage existentes en el mercado, que tienen como principal diferencia utilizar fibra óptica y además ser excesivamente caras, incluso inaccesibles para Pymes o personas comunes. De esta forma iSCSI es la solución perfecta para quienes necesiten por ejemplo adosar un storage compartido a un cluster, recurso que si además lo formateamos en GFS podríamos permitir que los nodos participantes puedan realizar lecto-escritura simultánea sin tener un servicio de por medio y con una excelente performance.

Tutorial iSCSI

A continuación les explicaré como configurar un servidor iSCSI, como configurar un cliente que use un recurso iSCSI compartido y algunos tips de por medio.

El famoso target

Lo primero que haremos es armar el target, que es el server (**1.1.1.1**) que va a compartir el recurso, que en este caso es un logical volume.

1. Creamos un logical volume llamado *iscsi*, de 10GB, el cual posteriormente será exportado.

```
[ root@server ]# lvcreate -L 10G -n iscsi vg_local
```

2. Instalamos scsi-target-utils

```
[ root@server ]# yum install -y scsi-target-utils
```

3. Ahora tenemos que modificar /etc/tgt/targets.conf para crear la LUN a exportar.
   Dentro de la configuración agregaremos el LV que acabamos de crear (*/dev/vg_local/iscsi*), el IQN (*iSCSI Qualified Name*) el cual le llamaremos '*iqn.2011-04.com.example.storage:iscsi*' y agregarmos que sólo el cliente **1.1.1.2** podrá utilizarlo.

```
[ root@server ]# cat <<FIN >> /etc/tgt/targets.conf
<target iqn.2011-04.com.example.storage:iscsi>
backing-store /dev/vg_local/iscsi
initiator-address 1.1.1.2
</target>
FIN
```

4. Iniciamos el servicio llamado tgt, el cual que se encargará de entregar el recurso. Obviamente es de suma importancia dejarlo activo al inicio para evitar que los clientes se cuelguen y/o arrojen un error de acceso al dispositivo.

```
[ root@server ]# service tgtd start
[ root@server ]# chkconfig tgtd on
```

5. Para comprobar que está todo funcionando bien podríamos utilizar tgt-admin. Cabe recalcar que por cada LUN que exportemos el sistema creará otra LUN adicional, de 0MB que será utilizada por el servicio para fines de control.

```
[ root@server ]# tgt-admin -s
       Target 1:  iqn.2011-04.com.example.storage:iscsi
              System information:
                     Driver: iscsi
                     State: ready
              I_T nexus information:
              LUN information:
                     LUN: 0
                            Type: controller
                            SCSI ID: deadbeaf1:0
                            SCSI SN: beaf10
                            Size: 0 MB
                            Online: Yes
                            Removable media: No
                            Backing store: No backing store
                     LUN: 1
                            Type: disk
                            SCSI ID: deadbeaf1:1
                            SCSI SN: beaf11
                            Size: 10240 MB
                            Online: Yes
                            Removable media: No
                            Backing store: /dev/vg_local/iscsi
                     Account information:
                            ACL information:
                                   1.1.1.2
```

El cliente

1. En el cliente debemos primero instalar el paquete '*iscsi-initiator-utils*', que nos entregará las herramientas para acceder al target.

```
[ root@client ]# yum install -y iscsi-initiator-utils
```

2. Luego debemos configurar un alias en el archivo */etc/iscsi/initiatorname.iscsi* con el fin de que

el servicio tgt pueda conversar de forma correcta.

```
[ root@client ]# echo "InitiatorAlias=cliente1" >> /etc/iscsi/initiatorname.iscsi
```

3. Iniciamos el servicio iSCSI y lo dejamos activo.

```
[ root@client ]# chkconfig iscsi on; service iscsi start
```

4. Si no encontramos error alguno podemos empezar a buscar los targets que acabamos de configurar.

```
[ root@client ]# iscsiadm -m discovery -t sendtargets -p 1.1.1.1
1.1.1.1.:3260,1 iqn.2011-04.com.example.storage:iscsi
```

La salida de del comando debería mostrar los dispositivos de bloques disponibles en la siguiente estructura: <target_IP:port> <target_iqn_name>. En este caso ve sin problemas el target configurado y sólo tenemos que adosarlo como un disco duro más.

5. Para realizar la conección al recurso desde el cliente ejecutamos el comando *iscsiadm* agregando el parámetro de login al final (-l) activando la opción de realizar un adosamiento del dispositivo de bloques compartido.

```
[ root@client ]# iscsiadm -m node -T iqn.2011-04.com.example.storage:iscsi  -p
1.1.1.1 -l
```

6. Listo. Ahora tenemos el dispositivo adosado y ya podemos empezar a utilizarlo. Para ello podemos analizar si existe con el comando '*fdisk -l'* y con el mismo comando empezar a crear las particiones, para luego formatearlo con *mkfs*.

```
[ root@client ]# fdisk -l | grep GB
...
Disk /dev/sdb: 10 GB, 10001569 bytes
...
[ root@client ]# fdisk /dev/sdb
...
[ root@client ]# partprobe
[ root@client ]# mkfs.ext3 /dev/sdb1
...
```

7. Finalmente si queremos agregarlo a fstab para que el dispositivo se monte durante el reinicio del sistema tenemos que tener en cuenta que hay que agregarle el parámetro '*_netdev*'.

```
[ root@client ]# cat <<FIN >> /etc/fstab
/dev/sdb1      /punto/de/montaje      ext3    _netdev    0 0
FIN
```

8. Una vez realizado el adosamiento del dispositivo, éste queda activo incluso despues de un reinicio. Si no lo queremos usar más entonces podemos desactivarlo agregando la opción '-u' al comando *iscsiadm* de la siguiente forma:

```
[ root@client ]# iscsiadm -m node -T iqn.2011-04.com.example.storage:iscsi  -p
1.1.1.1 -u
```

Síntesis

El driver iSCSI es muy util si queremos montar dispositivos y acceder a ellos a gran velocidad utilizando de por medio la tarjeta de red. Obviamente si tenemos una NIC lenta tendremos limitaciones, pero en caso de utilizar tarjetas de 1GB o superior podríamos tener tasas de transferencias altísimas. El escenario se complica si tenemos muchos nodos que quieren escribir de forma simultánea, ya que no podremos utilizar ext3 o ext4, sino que tendremos que formatear el dispositivo en GFS y para ello necesitamos configurar un ambiente clusterizado (escenario que tampoco es dificil de armar). De todas formas es una solución fácil de implementar y que da muy buenos resultados a la hora de armar soluciones baratas y sin tener la necesidad de comprar hardware caro.

Saludos.
--
Antonio Sebastián Sallés M.
Fedora Ambassador, Chile
Technical Account Manager para Red Hat Latam
asalles@redhat.com