iSCSI基础入门

目录

[iSCSI基础入门 1](#_Toc344034909)

[第1章 背景知识 1](#_Toc344034910)

[1.1 SCSI介绍 1](#_Toc344034911)

[1.2 iSCSI概述 2](#_Toc344034912)

[第2章iSCSI target (lio) 3](#_Toc344034913)

[2.1 创建Storage Object 3](#_Toc344034914)

[2.1.1 Backstore和Storage Object 3](#_Toc344034915)

[2.1.2 创建Block IO Backstore 3](#_Toc344034916)

[2.2 创建iSCSI端点( Endpoint) 4](#_Toc344034917)

[2.1.1创建iSCSI端点 4](#_Toc344034918)

[2.1.2创建iSCSI网络入口 5](#_Toc344034919)

[第3章iSCSI initiator(open-iscsi) 6](#_Toc344034920)

[2.1 常用命令 6](#_Toc344034921)

[2.2 使用网络设备 6](#_Toc344034922)

# 第1章 背景知识

## 1.1 SCSI介绍

SCSI－[小型计算机系统接口](http://baike.baidu.com/view/611524.htm" \t "_blank)，是一种专门为小型[计算机系统](http://baike.baidu.com/view/1130583.htm)设计的[存储单元](http://baike.baidu.com/view/1223079.htm)接口模式。SCSI定义了一系列规则提供给I/O设备，用以请求相互之间的服务用。每个I/O设备称为“逻辑单元”（LU），每个逻辑单元都有一个唯一的地址来区分它们，这个地址称为“逻辑单元号”（LUN）。

SCSI模型采用客户端/服务器（C/S，Client/Server）模式，客户端称为initiator，服务器称为target，数据传输时，initiator向target发送request，target回应response，在iSCSI协议中也沿用了这套思路。



图1-1 SCSI设备服务和任务请求交互模型

## 1.2 iSCSI概述

ISCSI:INTERNET SCSI或TCP/IP OVER SCSI，即把SCSI协议与TCP/IP协议相结合，在IP网络上传输SCSI数据.也就是可以让SCSI命令通过网络传送到远程SCSI设备上，而SCSI协议只能访问本地的SCSI设备。

iSCSI使用客户/服务器模型。客户端称为initiator，服务器端称为target。

* Initiator：通常指用户主机系统，用户发送SCSI请求，并将SCSI命令和数据封装到TCP/IP包中发送到IP网络中。
* Target：通常位于存储设备上，用于解析TCP/IP包中的SCSI命令和数据。

在initiator端的TCP端口号随机选取，target的端口号默认是3260

其协议栈如下：



图1-2 iSCSI协议栈

# 第2章iSCSI target (lio)

Lio是一个开源的storage target的一种实现标准，他支持所有流行存储结构：[iSCSI](http://linux-iscsi.org/wiki/ISCSI), [Fibre Channel](http://linux-iscsi.org/wiki/Fibre_Channel)([QLogic](http://www.qlogic.com/)), [FCoE](http://linux-iscsi.org/wiki/Fibre_Channel_over_Ethernet" \o "Fibre Channel over Ethernet), [InfiniBand](http://linux-iscsi.org/wiki/InfiniBand" \o "InfiniBand)/[SRP](http://linux-iscsi.org/wiki/SRP) ([Mellanox](http://www.mellanox.com/)), [vHost](http://linux-iscsi.org/wiki/VHost" \o "VHost)。Lio可以把一个块设备输出到iscsi　target中，其中用到两个工具tcm\_node和lio\_node。

## 2.1 创建Storage Object

创建Storage Object 即是把要输出到target的设备(Backstore)转换为iscisi可识别的形式，这个形式叫做Storage Object.

### 2.1.1 Backstore和Storage Object

Backstore指的是iSCSI Targets所能输出的存储设备类型，包括

• File IO

• Block IO

• Raw SCSI disk

• RAMDISK

上面这些存储类型要想被iSCSI识别，必须由tcm\_node转换为iSCSI可以识别的iSCSI Storage Object

### 2.1.2 创建Block IO Backstore

由Block IO Backstore生成iSCSI Storage Object形式如下*：*

tcm\_node --block <HBA>/<StorageObjectName> <PathName>/<BlockDeviceName>

其中 udev path 可以是

• /dev

• /dev/mapper

• /dev/*VolumeGroup*/*LogicalVolume*

• /dev/disk

* 创建*Storage Object*

# tcm\_node --block iblock\_0/test\_lv0 /dev/vgszx2/lv0

Status: DEACTIVATED Execute/Left/Max Queue Depth: 0/128/128 SectorSize: 512 MaxSectors: 512

iBlock device: dm-3 UDEV PATH: /dev/vgszx2/lv0

Major: 253 Minor: 3 CLAIMED: IBLOCK

* 创建Storage Object的同时生成T10 WWN Unit Serial序列号。这个序列号在这个文件中：

/sys/kernel/config/target/core/iblock\_0/test\_lv0/wwn/vpd\_unit\_serial。

可以 用命令查看

# tcm\_node --wwn iblock\_0/test\_lv0

T10 VPD Unit Serial Number: 9b48a694-35ec-409a-ac1f-804f26da9908

而且<HBA>/<StorageObjectName>可以唯一标示这个设备。

* 显示设备的属性

# tcm\_node --listdevattr iblock\_0/test\_lv0

\TCM Storage Object Attributes for /sys/kernel/config/target/core/iblock\_0/test\_lv0

unmap\_granularity\_alignment: 0

……

* 查看Storage Object所对应的设备路径udev Path

# tcm\_node --udevpath iblock\_0/test\_lv0

/dev/vgszx2/lv0

* 释放*Storage Object*

tcm\_node --freedev iblock\_0/test\_lv0

* 其他的命令行选项可以通过 # tcm\_node -h来查看

## 2.2 创建iSCSI端点( Endpoint)

之后，就可以利用生成好的iSCSI Storage Object和lio\_node命令创建iSCSI端点了。

一个iSCSI 端点由两部分构成 : IQN和 TPG.

* TPG (iSCSI Target Portal Group)通过关联一个Storage Object构成，其中Storage Object可以包含一或多个LUN，TPG具体是一个序数。
* IQN (iSCSI Qualified Name)是用户自定义的一个字符串，推荐使用Target’s T10 WWN，可以使得这个iSCSI端点全球唯一。

### 2.1.1创建iSCSI端点

* 从逻辑卷Backstore创建iSCSI端点

lio\_node --addlun <TargetIQN> <TPG#> <LUN#> <LUNName> <HBA>/<StorageObjectName>

# lio\_node --addlun com.netease 1 0 lun\_my\_lv iblock\_0/test\_lv0

Successfully created iSCSI Target Logical Unit

其中 iblock\_0/test\_lv0 即为之前生成的Storage Object

可以在这个iSCSI端点上创建多个LUN。

注意查看目录/sys/kernel/config/target/iscsi/com.netease生成了，里面存放所有LUN的信息。

* 列出所有iSCSI 端点

# lio\_node --listendpoints

* 从iSCSI端点上删除一个iSCSI LUN

lio\_node --dellun <TargetIQN> <TPG#> <LUN#>

# lio\_node --dellun com.netease 1 0

Successfully deleted iSCSI Target Logical Unit

* 更多的命令查看lio\_node –h

### 2.1.2创建iSCSI网络入口

上面生成的iSCSI端点并不能通过网络来识别，用命令查看

# iscsiadm -m discovery -t sendtargets -p 114.113.199.19:3260

……

114.113.199.19:3260,-1 com.netease

…..

发现第二项是-1

* 创建iSCSI网络入口

lio\_node --addnp <TargetIQN> <TPG#> <IPaddr>:<Port#>

# lio\_node --addnp com.netease 1 0.0.0.0:3260

Successfully created network portal: 0.0.0.0:3260 created com.netease TPGT: 1

再次查看，# iscsiadm -m discovery -t sendtargets -p 114.113.199.19:3260

仍然是-1

* 此时的iSCSI端点仍然不能从Initiator登陆，要想登陆必须打开选项：

lio\_node --enabletpg <TargetIQN> <TPG#>

# lio\_node --enabletpg com.netease 1

Successfully enabled iSCSI Target Portal Group: com.netease 1

* 打开登陆选项之后，发现仍然不能登陆。原因是登陆时必须经过验证，

此时可以通过关闭验证(Disabling Authentication)的方式来解决。

需两个步骤

* 打开demo模式

打开demo模式意味着所有的Initiator可以访问iSCSI 端点上所有的LUN

lio\_node --demomode <TargetName> <TPG#>

# lio\_node --demomode com.netease 1

Successfully disabled Initiator ACL mode (Enabled DemoMode) on iSCSI Target Portal Group: com.netease 1

* 关闭CHAP验证

lio\_node --disableauth <TargetName> <TPG#>

# lio\_node --disableauth com.netease 1

Successfully disabled iSCSI Authentication on iSCSI Target Portal Group: com.netease 1

经过以上步骤，iSCSI Initiator就可以登录iSCSI 端点来访问上面的LUN了。

# iscsiadm -m node -T com.netease -p 114.113.199.19:3260 --login

Logging in to [iface: default, target: com.netease, portal: 114.113.199.19,3260]

Login to [iface: default, target: com.netease, portal: 114.113.199.19,3260] successful.

* 去掉iSCSI端点的写保护属性

此时登录上去的设备不能被正常挂载(如果挂载，参考第三章)，必须去掉写保护属性，然后重新挂载。

# lio\_node --settpgattr=com.netease 1 demo\_mode\_write\_protect 0

# 第3章iSCSI initiator(open-iscsi)

open-iscsi 是iSCSI initiator的一种实现标准，其命令行工具为iscsiadm.

## 2.1 常用命令

* 发现target：

# iscsiadm -m discovery -t sendtargets -p 114.113.199.19:3260

…..

114.113.199.19:3260,1 com.netease

* 登陆已发现target：

# iscsiadm -m node -T com.netease -p 114.113.199.19:3260 --login

Logging in to [iface: default, target: com.netease, portal: 114.113.199.19,3260]

Login to [iface: default, target: com.netease, portal: 114.113.199.19,3260] successful.

或者iscsiadm -m node -T com.netease -p 114.113.199.19:3260 -l

* 登出target

iscsiadm -m node -T com.netease -p 114.113.199.19:3260 --logout

Logging out of session [sid: 1676, target: com.netease, portal: 114.113.199.19,3260]

Logout of [sid: 1676, target: com.netease, portal: 114.113.199.19,3260] successful.

或者iscsiadm -m node -T com.netease -p 114.113.199.19:3260 -u

iscsiadm -m node -T iqn.2005-03.com.max -p 192.168.120.198 –u

* 更多的用法查看 man iscsi

## 2.2 使用网络设备

通过iscsiadm登录进去之后，就可以使用该设备了。

* 登录

iscsiadm -m node -T com.netease -p 114.113.199.19:3260 –login

* 查看设备

# iscsiadm -m session -P3

Target: com.netease

Current Portal: 114.113.199.19:3260,1

……..

scsi1682 Channel 00 Id 0 Lun: 0

Attached scsi disk sdat State: running

可以看出，检测到的网络设备为sdat

* 挂载设备

现在可已挂载该设备并使用了。

* 正如所述，如果不去掉iSCSI端点的写保护属性会报如下错误

# mount /dev/sdal /home/ebs/dir/sdat

mount: block device /dev/sdal is write-protected, mounting read-only

mount: you must specify the filesystem type

* 格式化并挂载成功之后用df查看

/dev/sdat 194M 5.6M 179M 4% /home/ebs/dir/sdat

* 使用设备

现在使用设备都是在操作网络设备。