

ISCSI网络存储服务

iSCSI技术是一种由IBM、Cisco公司研究开发的，是一个供硬件设备使用的可以在IP协议的上层运行的SCSI指令集，这种指令集合可以实现在IP网络上运行SCSI协议，使其能够在诸如高速千兆以太网上进行路由选择。iSCSI技术是一种新储存技术，该技术是将现有SCSI接口与以太网(Ethernet)技术结合，使服务器可与使用IP网络的储存装置互相交换资料。

技术优点和成本优势主要体现以下方面：

硬件成本低：构建iSCSI存储网络，除了存储设备外，交换机、线缆、接口卡都是标准的以太网配件，价格相对来说比较低廉。

操作简单，维护方便：对iSCSI存储网络的管理，实际上就是对以太网设备的管理。

扩充性强：对于已经构建的iSCSI存储网络来说，增加iSCSI存储设备和服务器都将变得简单且无需改变网络的体系结构。

带宽和性能：iSCSI存储网络的访问带宽依赖以太网带宽。随着千兆以太网的普及和万兆以太网的应用，iSCSI存储网络会达到甚至超过FC（FiberChannel，光纤通道）存储网络的带宽和性能。

基本概念：

在SAN中，主机一般都是Initiator，存储设备则是Target。

Initiator

- SCSI会话的发起方
- 向Target请求LUN,并将数据的读写指令发送给Target。

Target（默认采用TCP 3260端口）

- 接受SCSI回话的一方
- 它接收来自Initiator的指令，为Initiator提供LUN,并实现对LUN的读写

IQN: iqn.1994-05.com.redhat:737db83a23df

全球唯一名称，用于以强制命令格式来识别启动器和目标

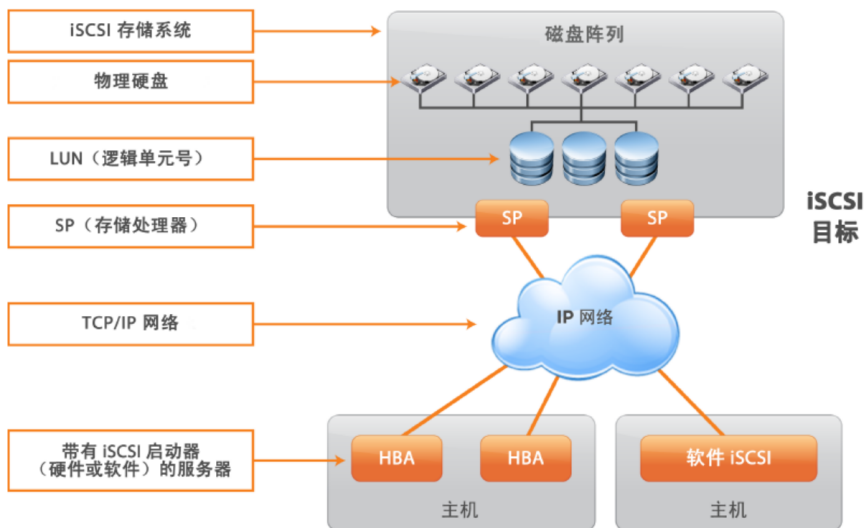
iqn 表示此名称将使用域作为其标识符

yyyy-mm 拥有域名的第一个月

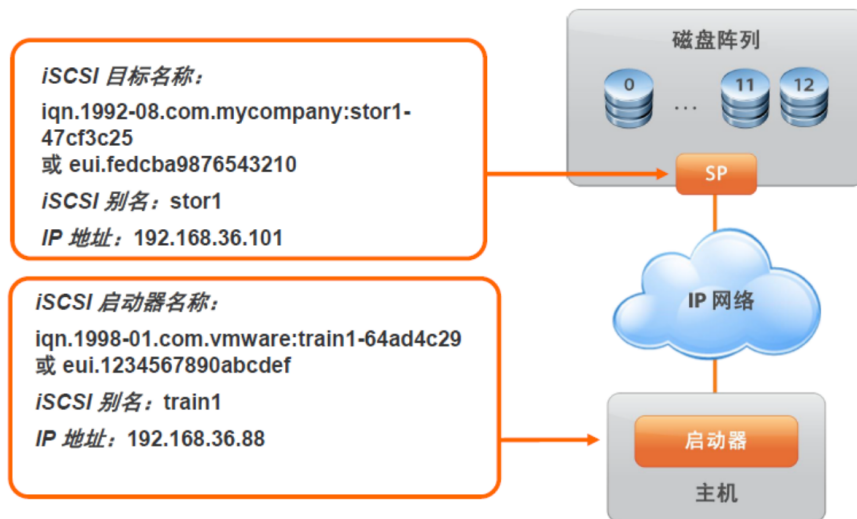
com.redhat 逆向域名

optional_string 以冒号为前缀的可选字符串，根据需要分配

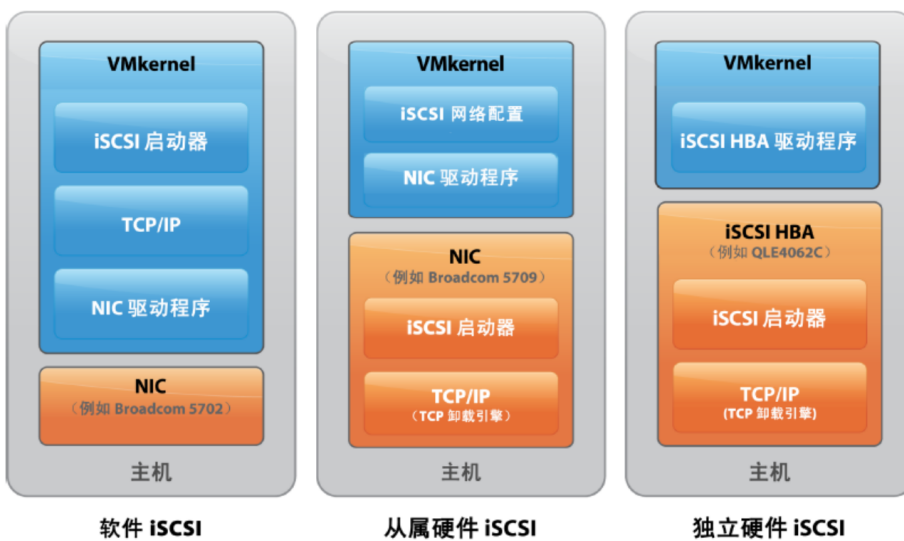
iSCSI组件图：



iSCSI寻址图:



iSCSI启动器类型:



Linux主要的开源Target项目

- Linux SCSI Target-STGT/tgt
- Linux-IO Target - LIO Linux 2.6.38开始纳入内核

- SCST-Generic SCSI Subsystem for linux

<http://scst.sourceforge.net/comparison.html>

Generic SCSI Target Subsystem for Linux

Home | Main | News | Drivers | Downloads | Contributing | **Comparison** | Users

> **COMPARISON**

- Features comparison
- SCST vs LIO/TCM
- SCST vs STGT
- MC/S vs MPIO

FEATURES COMPARISON BETWEEN LINUX SCSI TARGETS

This features comparison is intended to be a complete and fair feature-by-feature comparison between the listed targets without any bias to SCST. If you see anything wrong somewhere or anything missed, you are welcome to report it in [scst-devel mailing list](#) and it will be corrected.

Also Sebastian Riemer wrote a good summary in [his e-mail](#) (April 2013)

As on June 2011, briefly reviewed April 2013.

	SCST	STGT	IET	LIO/TCM
General				
Upstream kernel	-	-	-	Since 2.6.38
Generic Target Engine	+	+	ISCSI only	+
Architecture	Kernel only	User space only	Split 1	Kernel only
Stability	+	+	+	Probably
Performance 2	**** 3	***	****	****
Zero-copy passing data between	+	+	-	-

Linux-IO Target概述

- http://linux-iscsi.org/wiki/Main_Page
- Linux-IO Target在Linux内核中，用软件实现各种SCSI Target
- 前端：FC、FCoE、iSCSI、1394、USB、vHost等
- 后端：SATA, SAS, SCSI, SSD, FLASH, DVD, USB, ramdisk等
- 架构：
 - 支持SSE4.2高性能、多线程
 - 支持x86, ia64, Alpha, Cell, PPC, ARM, MIPS等多种CPU
 - 支持高可用、负载均衡群集
 -

Welcome to Linux-IO,
the Linux SCSI Target wiki.
100 articles, 35,315,554 pageviews

- Targetcli
- Downloads
- Support
- LIO
- Features
- Ecosystem
- Datera
- Contributing

Summary

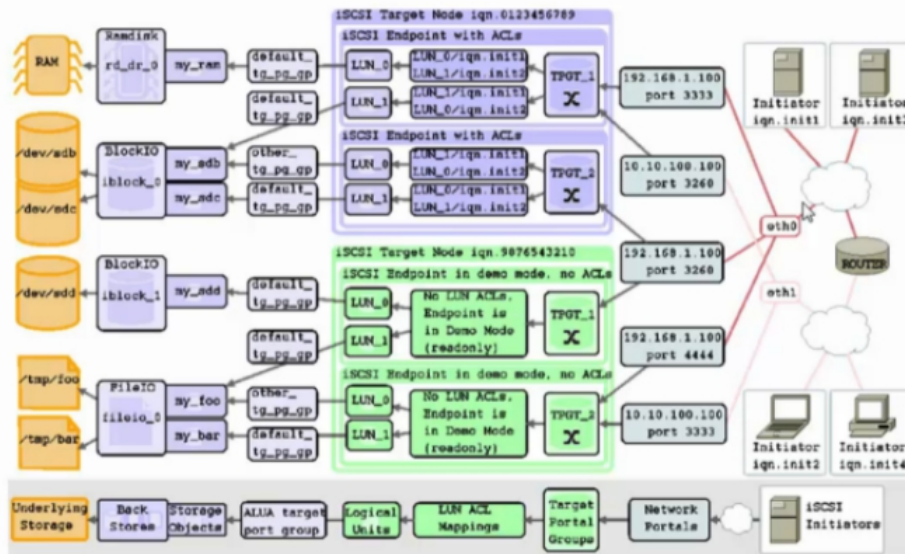
LinuxIO (LIO™) is the standard open-source SCSI target in Linux. It supports all prevalent storage fabrics, including Fibre Channel (QLogic®, Emulex), FCoE, IEEE 1394, iSCSI (incl. Chelsio offload support), NVMe-OF, iSER (Mellanox® InfiniBand), SRP (Mellanox InfiniBand), USB, vHost, etc.

The advanced feature set of LinuxIO has made it the SCSI target of choice for many storage array vendors, for instance allowing them to achieve VMware® Ready certifications. Native support for LIO in QEMU/KVM, libvirt, and OpenStack™ (setup 🛠️, code 📄) makes it an attractive storage option for cloud deployments.

LIO includes targetcli, a management shell and API with a single namespace for all storage objects.

LIO and targetcli are developed by Datera, Inc., a data storage systems and software company located in Mountain View in the Silicon Valley.

Linux-IO的iSCSI Target架构（摘自LIO官网）



iscsi实现网络存储，提供存储端叫 target，使用存储端叫 initiator。target 上可以提供存储空间，initiator负责连接ISCSI设备，在ISCSI设备中创建文件系统，以及存取数据，在initiator 看上去是多了一块硬盘。

服务器端配置target，准备做为LUN发布的ISCSI的存储设备（可以是一整块磁盘、一个分区、逻辑卷或RAID阵列）。

事先准备好两个逻辑卷作为ISCSI的存储设备。

```
ACTIVE      '/dev/mypool/lv01' [800.00 MiB] inherit
ACTIVE      '/dev/vg00/lv00' [300.00 MiB] inherit
```

首先安装target

```
# yum -y install targetd targetcli
```

启动服务

```
[root@server1 ~]# systemctl enable target
ln -s '/usr/lib/systemd/system/target.service' '/etc/systemd/system/multi-user.target.wants/target.service'
[root@server1 ~]# systemctl start target
```

设置防火墙

```
[root@server1 ~]# firewall-cmd --permanent --add-port=3260/tcp
success
[root@server1 ~]# firewall-cmd --reload
success
```

执行targetcli工具

```
[root@server1 ~]# targetcli
targetcli shell version 2.1.fb34
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/ >
```

注：输入help可以查看targetcli的帮助

```
> ls
0- / ..... [..]
  o- backstores ..... [..]
    | o- block ..... [Storage Objects: 0]
    | o- fileio ..... [Storage Objects: 0]
    | o- pscsi ..... [Storage Objects: 0]
    | o- ramdisk ..... [Storage Objects: 0]
    o- iscsi ..... [Targets: 0]
    o- loopback ..... [Targets: 0]
```

基本思路：

先把准备共享的块做出来，创建一个target，在target上创建LUN，一个LUN连接一个块

1、创建块即给要发布的逻辑卷起个名字

```
> /backstores/block create server0.disk1 /dev/vg00/lv00
Created block storage object server0.disk1 using /dev/vg00/lv00.
> /backstores/block create server0.disk2 /dev/mypool/lv01
Created block storage object server0.disk2 using /dev/mypool/lv01.
>
```

注：给/dev/vg00/lv00逻辑卷起个名字叫server0.disk1;给/dev/mypool/lv01逻辑卷起个名字叫server0.disk2

查看块如下图所示：

```
> ls /
0- / ..... [..]
  o- backstores ..... [..]
    | o- block ..... [Storage Objects: 2]
    | | o- server0.disk1 ..... [/dev/vg00/lv00 (300.0MiB) write-thru deactivated]
    | | o- server0.disk2 ..... [/dev/mypool/lv01 (800.0MiB) write-thru deactivated]
    | o- fileio ..... [Storage Objects: 0]
    | o- pscsi ..... [Storage Objects: 0]
    | o- ramdisk ..... [Storage Objects: 0]
    o- iscsi ..... [Targets: 0]
    o- loopback ..... [Targets: 0]
```

2、创建 iqn 名字即创建ISCSI对象

```
> /iscsi create iqn.2015-06.com.benet:disk1
Created target iqn.2015-06.com.benet:disk1.
Created TPG 1.
> /iscsi create iqn.2015-06.com.benet:disk2
Created target iqn.2015-06.com.benet:disk2.
Created TPG 1.
```

查看ISCSI对象如下图所示：


```

/> ls
0- / ..... [..]
  o- backstores ..... [..]
    | o- block ..... [Storage Objects: 2]
    | | o- server0.disk1 ..... [/dev/vg00/lv00 (300.0MiB) write-thru deactivated]
    | | o- server0.disk2 ..... [/dev/mypool/lv01 (800.0MiB) write-thru deactivated]
    | o- fileio ..... [Storage Objects: 0]
    | o- pscsi ..... [Storage Objects: 0]
    | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 2]
    | o- iqn.2015-06.com.benet:disk1 ..... [TPGs: 1]
    | | o- tpg1 ..... [no-gen-acls, no-auth]
    | | | o- acls ..... [ACLs: 0]
    | | | o- luns ..... [LUNs: 0]
    | | | o- portals ..... [Portals: 0]
    | o- iqn.2015-06.com.benet:disk2 ..... [TPGs: 1]
    | | o- tpg1 ..... [no-gen-acls, no-auth]
    | | | o- acls ..... [ACLs: 0]
    | | | o- luns ..... [LUNs: 0]
    | | | o- portals ..... [Portals: 0]
  o- loopback ..... [Targets: 0]

```

3、设置ACL即将ISCSI对象与客户端IP或主机名绑定

```

/> /iscsi/iqn.2015-06.com.benet:disk1/tpg1/acls create iqn.2014-06.com.benet:client1
Created Node ACL for iqn.2014-06.com.benet:client1
/> /iscsi/iqn.2015-06.com.benet:disk2/tpg1/acls create iqn.2014-06.com.benet:client2
Created Node ACL for iqn.2014-06.com.benet:client2

```

注意：iqn.2015-06.com.benet:client1是 initiator 的名字，需要在客户端中设置的。

4、创建LUN并绑定块

```

/> /iscsi/iqn.2015-06.com.benet:disk1/tpg1/luns create /backstores/block/server0.disk1
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2014-06.com.benet:client1
/> /iscsi/iqn.2015-06.com.benet:disk2/tpg1/luns create /backstores/block/server0.disk2
Created LUN 0.
Created LUN 0->0 mapping in node ACL iqn.2014-06.com.benet:client2

```

一个ISCSI对象可以创建多个LUN (LUN0、LUN1.....) 。

执行ls查看

```

/> ls
0- / ..... [..]
  o- backstores ..... [..]
    | o- block ..... [Storage Objects: 2]
    | | o- server0.disk1 ..... [/dev/vg00/lv00 (300.0MiB) write-thru activated]
    | | o- server0.disk2 ..... [/dev/mypool/lv01 (800.0MiB) write-thru activated]
    | o- fileio ..... [Storage Objects: 0]
    | o- pscsi ..... [Storage Objects: 0]
    | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 2]
    | o- iqn.2015-06.com.benet:disk1 ..... [TPGs: 1]
    | | o- tpg1 ..... [no-gen-acls, no-auth]
    | | | o- acls ..... [ACLs: 1]
    | | | | o- iqn.2014-06.com.benet:client1 ..... [Mapped LUNs: 1]
    | | | | | o- mapped_lun0 ..... [lun0 block/server0.disk1 (rw)]
    | | | o- luns ..... [LUNs: 1]
    | | | | o- lun0 ..... [block/server0.disk1 (/dev/vg00/lv00)]
    | | | o- portals ..... [Portals: 0]
    | o- iqn.2015-06.com.benet:disk2 ..... [TPGs: 1]
    | | o- tpg1 ..... [no-gen-acls, no-auth]
    | | | o- acls ..... [ACLs: 1]
    | | | | o- iqn.2014-06.com.benet:client2 ..... [Mapped LUNs: 1]
    | | | | | o- mapped_lun0 ..... [lun0 block/server0.disk2 (rw)]
    | | | o- luns ..... [LUNs: 1]
    | | | | o- lun0 ..... [block/server0.disk2 (/dev/mypool/lv01)]
    | | | o- portals ..... [Portals: 0]
  o- loopback ..... [Targets: 0]

```

启动监听程序

```
/> /iscsi/ign.2015-06.com.benet:disk1/tpg1/portals create 172.24.3.5
Using default IP port 3260
Created network portal 172.24.3.5:3260.
/> /iscsi/ign.2015-06.com.benet:disk2/tpg1/portals create 172.24.3.5
Using default IP port 3260
Created network portal 172.24.3.5:3260.
```

注：172.24.3.5是ISCSI服务端网卡IP

```
/> ls
o- /
o- backstores ..... [..]
| o- block ..... [Storage Objects: 2]
| | o- server0.disk1 ..... [/dev/vg00/lv00 (300.0MiB) write-thru activated]
| | o- server0.disk2 ..... [/dev/mypool/lv01 (800.0MiB) write-thru activated]
| o- fileio ..... [Storage Objects: 0]
| o- pscsi ..... [Storage Objects: 0]
| o- ramdisk ..... [Storage Objects: 0]
o- iscsi ..... [Targets: 2]
| o- ign.2015-06.com.benet:disk1 ..... [TPGs: 1]
| | o- tpg1 ..... [no-gen-acls, no-auth]
| | | o- acls ..... [ACLs: 1]
| | | | o- ign.2014-06.com.benet:client1 ..... [Mapped LUNs: 1]
| | | | o- mapped_lun0 ..... [lun0 block/server0.disk1 (rw)]
| | | o- luns ..... [LUNs: 1]
| | | | o- lun0 ..... [block/server0.disk1 (/dev/vg00/lv00)]
| | | o- portals ..... [Portals: 1]
| | | | o- 172.24.3.5:3260 ..... [OK]
| o- ign.2015-06.com.benet:disk2 ..... [TPGs: 1]
| | o- tpg1 ..... [no-gen-acls, no-auth]
| | | o- acls ..... [ACLs: 1]
| | | | o- ign.2014-06.com.benet:client2 ..... [Mapped LUNs: 1]
| | | | o- mapped_lun0 ..... [lun0 block/server0.disk2 (rw)]
| | | o- luns ..... [LUNs: 1]
| | | | o- lun0 ..... [block/server0.disk2 (/dev/mypool/lv01)]
| | | o- portals ..... [Portals: 1]
| | | | o- 172.24.3.5:3260 ..... [OK]
o- loopback ..... [Targets: 0]
```

可以查看/etc/target/saveconfig.json配置文件，该配置文件保存着ISCSI的配置。

```
[root@server1 ~]# cat /etc/target/saveconfig.json
```

在initiator端的配置：

1、安装软件

```
# yum install -y iscsi-initiator-utils
```

2、给 initiator 起个名字

```
# vim /etc/iscsi/initiatorname.iscsi
```

内容如下：

```
[root@localhost ~]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=ign.2014-06.com.benet:client1
```

3、启动服务

```
# systemctl enable iscsi; systemctl start iscsi
```

4、发现存储

```
# iscsiadm -m discovery -t st -p 172.24.3.5
```

```
[root@localhost ~]# iscsiadm -m discovery -t st -p 172.24.3.5
172.24.3.5:3260,1 iqn.2015-06.com.benet:disk2
172.24.3.5:3260,1 iqn.2015-06.com.benet:disk1
```

5、登录存储

```
# iscsiadm -m node -T iqn.2015-06.com.benet:disk1 -p 172.24.3.5 -l
```

```
[root@localhost ~]# iscsiadm -m node -T iqn.2015-06.com.benet:disk1 -p 172.24.3.5 -l
Logging in to [iface: default, target: iqn.2015-06.com.benet:disk1, portal: 172.24.3.5,3260] (multiple)
Login to [iface: default, target: iqn.2015-06.com.benet:disk1, portal: 172.24.3.5,3260] successful.
```

注：-l表示连接ISCSI目标；-u表示断开和ISCSI目标的连接

验证客户端ISCSI连接

```
[root@localhost ~]# ls SCSI
[2:0:0:0] disk VMware, VMware Virtual S 1.0 /dev/sda
[4:0:0:0] cd/dvd NECUMWar VMware SATA CD01 1.00 /dev/sr0
[33:0:0:0] disk LIO-ORG server0.disk1 4.0 /dev/sdb
```

其它命令：

```
[root@cloud001 ~]# iscsiadm -m discovery [-P 1|0]显示有关已发现目标的信息
```

```
[root@cloud001 ~]# iscsiadm -m node [-P 1|0]显示有关已知目标的信息
```

```
[root@cloud001 ~]# iscsiadm -m session [-P 1|0]显示有关活动回话的信息
```

连接后，剩下的操作就和管理本地磁盘一样了。

报认证错误如下：

```
[root@cloud002 ~]# iscsiadm --mode node --targetname iqn.2018-06.com.test:disk1 --portal 192.168.5.190:3260 --login
Logging in to [iface: default, target: iqn.2018-06.com.test:disk1, portal: 192.168.5.190,3260] (multiple)
iscsiadm: Could not login to [iface: default, target: iqn.2018-06.com.test:disk1, portal: 192.168.5.190,3260].
iscsiadm: initiator reported error (24 - iSCSI login failed due to authorization failure)
iscsiadm: Could not log into all portals
```

可以尝试重启iscsid服务

```
[root@cloud002 ~]# systemctl restart iscsid.service
```

注意：如果实现iscsi硬盘自动开机挂载，需要在挂载选项中添加_netdev选项。


```
[root@cloud002 ~]# cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Mon Apr 23 15:11:14 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/cl-root    /          xfs      defaults        0 0
UUID=274be7e4-6dc6-485c-badf-27051cfb1a7b /boot      xfs      defaults        0 0
/dev/mapper/cl-swap    swap       swap     defaults        0 0
/iso/centos7.iso       /media/cdrom iso9660   defaults,loop 0 0
/swapfile              swap       swap     defaults        0 0
UUID="3a14e277-8ed1-497e-86ac-1d2a3f20b799" /sdc1      ext4     defaults 0 0
UUID="fd11e253-3057-49ce-a2af-13741695d288" /sdc2      xfs      defaults 0 0
UUID=90e49e51-4988-4424-bc14-d587a1ec8ba4 swap       swap     defaults 0 0
UUID="4bbdfb42-0d55-49ac-92b8-d3c32a8f8a14" /test/jishu ext4     defaults,usrquota,grpquota 0 0
UUID="4c85ffcf-a9ef-4fe1-9950-3b2a9a6982b1" /test/soft ext4     defaults,_netdev 0 0
UUID="9cd8bc7a-711d-491a-9459-595921b6c00a" /netstorage ext4     defaults,_netdev 0 0
```