

GNU/Linux 磁盘管理 6



Inetnet SCSI-iSCSI

数据一般用存储服务器来存储数据。如果空间不足，可以用以下方式解决：

- (1) 外接活动硬盘 (DAS)
- (2) 网络存储服务器 (NAS)
- (3) 存储区域网络服务 (SAN)



Inetnet SCSI-iSCSI

DAS:

Direct Attached Storage— 直接连接存储

本地存储设备



Inetnet SCSI-iSCSI

NAS(Network attched storage 网附加存储)

NAS 实际上就是一部 File Server, 一般以 NFS,SAMBA,FTP,HTTP 作为服务器, 来完成存储或读取的主要方式

当前主要的 NAS 软件可以用 FreeNAS 来进行配置和完成。通过 WEB 界面直接进行控制。方便、简单、快捷不用专业 IT 人员进行完成。

Inetnet SCSI-iSCSI

SAN(Storage Area Networ, 存储区域网络)

如果拥有大量的硬盘或使用空间，但主机插槽不够，可以使用 SAN。

一般可以将 SAN 视为一个外接式的存储设备。SAN 可以通过某些特殊的方式或介质来提供网络内的主机进行资料的存储。

Inetnet SCSI-iSCSI

SAN(Storage Area Networ, 存储区域网络)

简单来说 SAN 即

通过光纤通道连接到一群计算机上。在该网络中提供了多主机连接，但并非通过标准的网络拓扑。

SAN 的结构允许任何服务器连接到任何存储阵列，这样不管数据置放在那里，服务器都可直接存取所需的数据。因为采用了光纤接口，SAN 还具有更高的带宽

Inetnet SCSI-iSCSI

iSCSI

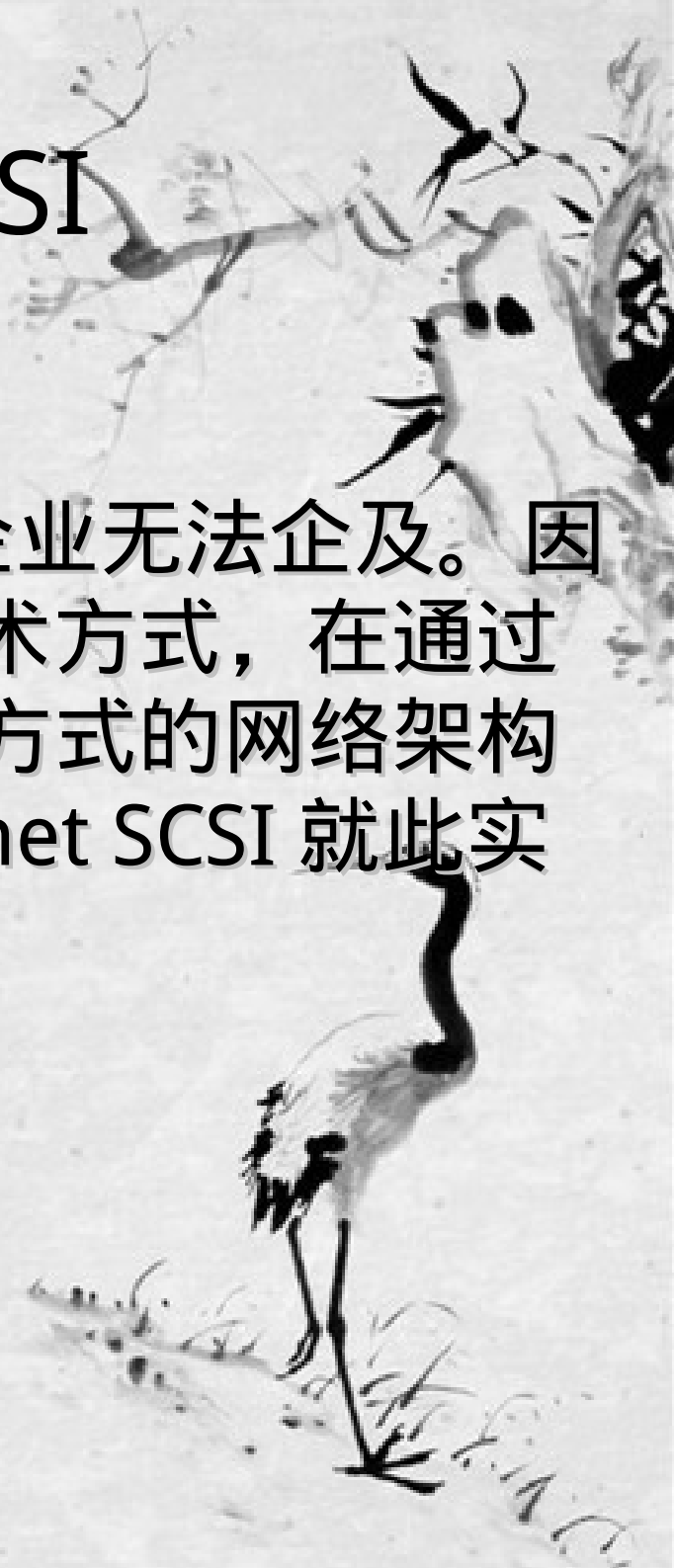
早起企业服务器基本上使用快速读写的 SCSI 硬盘。一般 SCSI 硬盘是通过 SCSI 接口来进行串联。为服务器本身提供信息存储。

SAN 出现之后，以光纤为介质进行传输其速度远远超过本地 SCSI，大型企业逐步采用了 SAN

Internet SCSI-iSCSI

iSCSI

因光纤成本高，使得许多中小型企业无法企及。因此有厂商将 SAN 连接方式改为 IP 技术方式，在通过一些标准的定义，可以实现通过任何方式的网络架构，来获取硬盘空间资源。至此，Internet SCSI 就此实现。



Inetnet SCSI-iSCSI

iSCSI

iSCSI 技术是一种由 IBM 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 接口与以太网网络 (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。

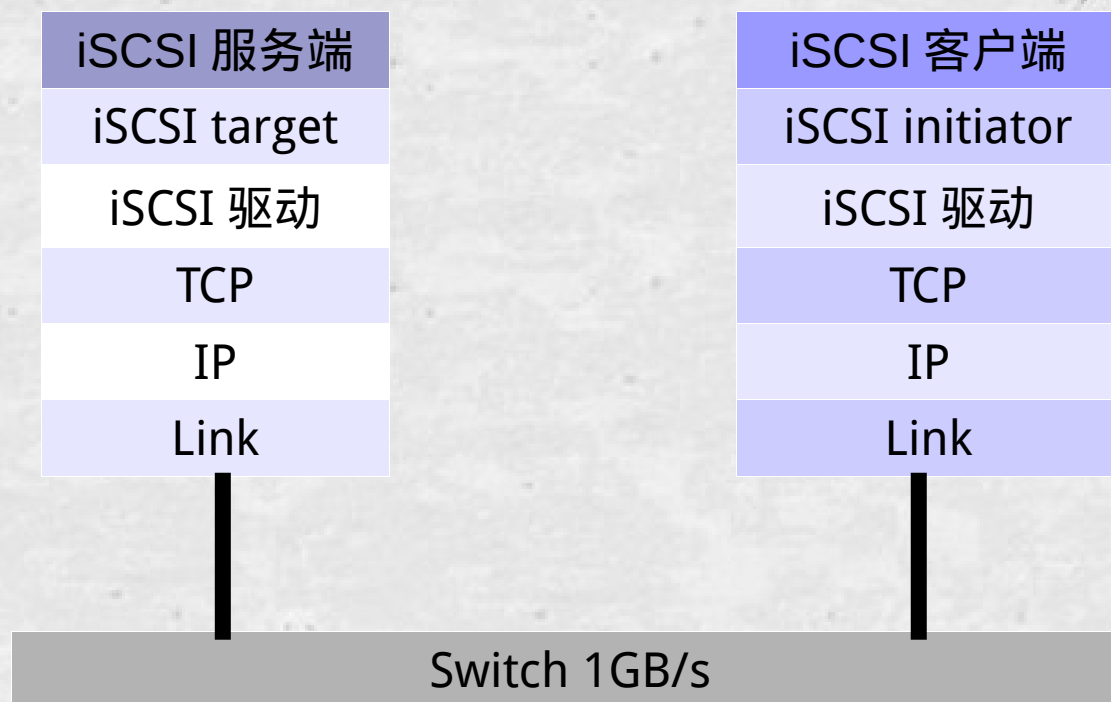
Inetnet SCSI-iSCSI

iSCSI 架构部分

- 1) 存储设备 (iSCSI target): 存放硬盘和 RAID 设备
- 2) 主机部分 (iSCSI initiator): 使用存储设备的客户端 (即连接到 iscsi 的服务器)

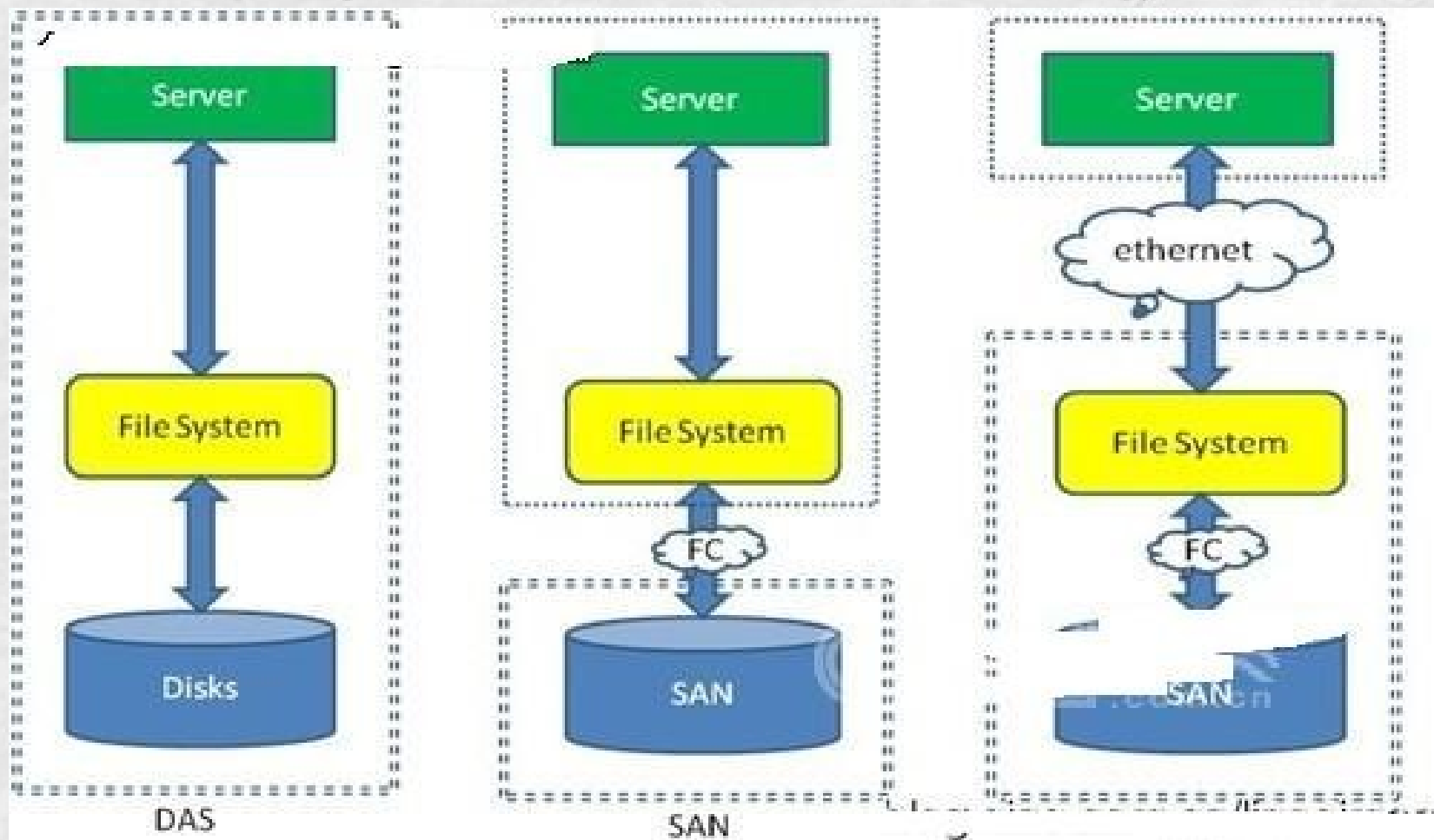
Inetnet SCSI-iSCSI

iSCSI 架构部分



Inetnet SCSI-iSCSI

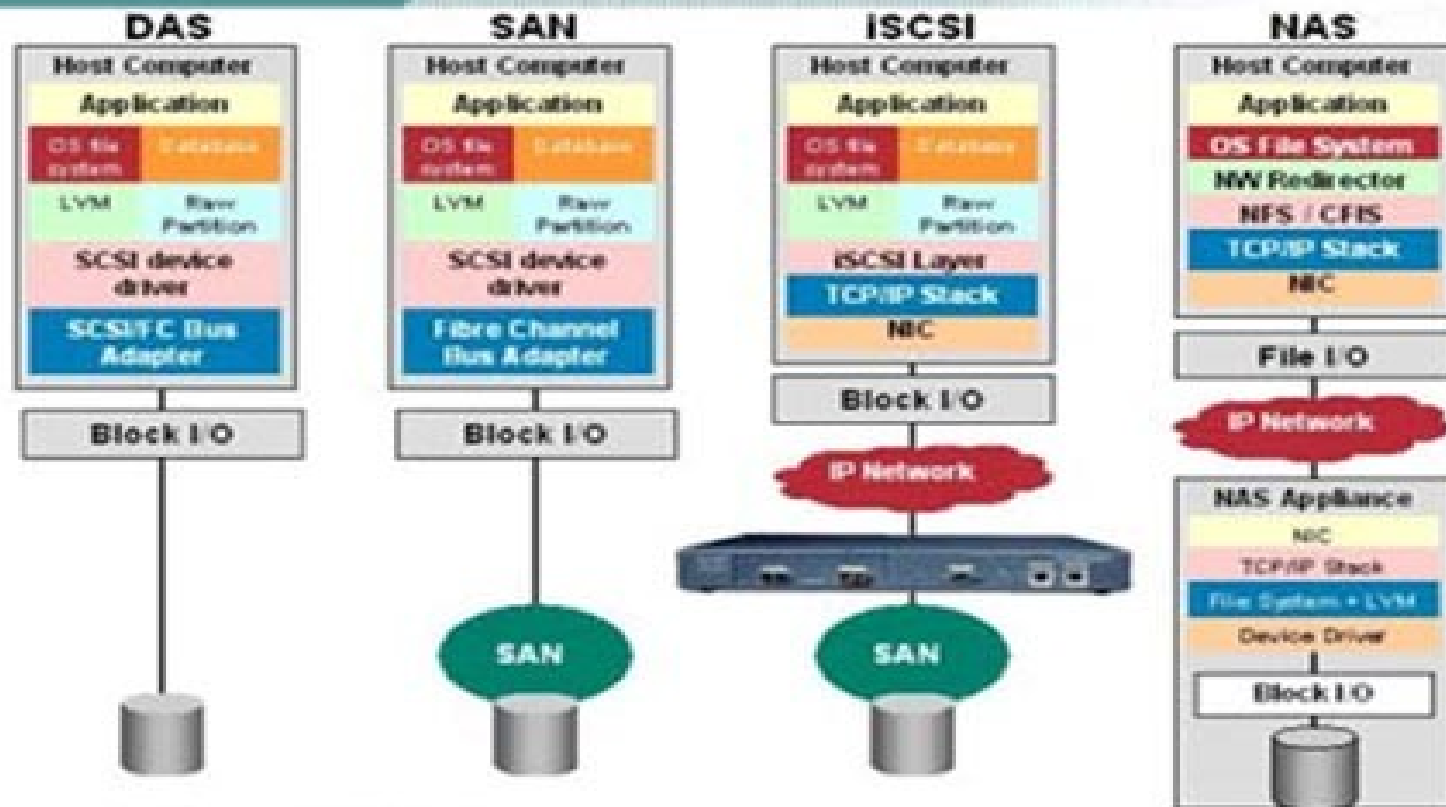
DAS、NAS、SAN 对比



Inetnet SCSI-iSCSI

DAS、NAS、SAN、iSCSI 对比

DAS/SAN/iSCSI/NAS Comparison



Inetnet SCSI-iSCSI

iSCSI 架构部分

1) RedHat 使用 targetcli 作为 iSCSI 服务的生成工具

2) 所需要的软件

软件名	软件功能	使用方向
targetcli	将 Linux 系统模拟成为 iSCSI target 功能	iSCSI 服务器
iscsi-initiator-utils	挂载来自 target 的硬盘到 Linux 本机上	iSCSI 客户端

Inetnet SCSI-iSCSI

iSCSI 服务端配置

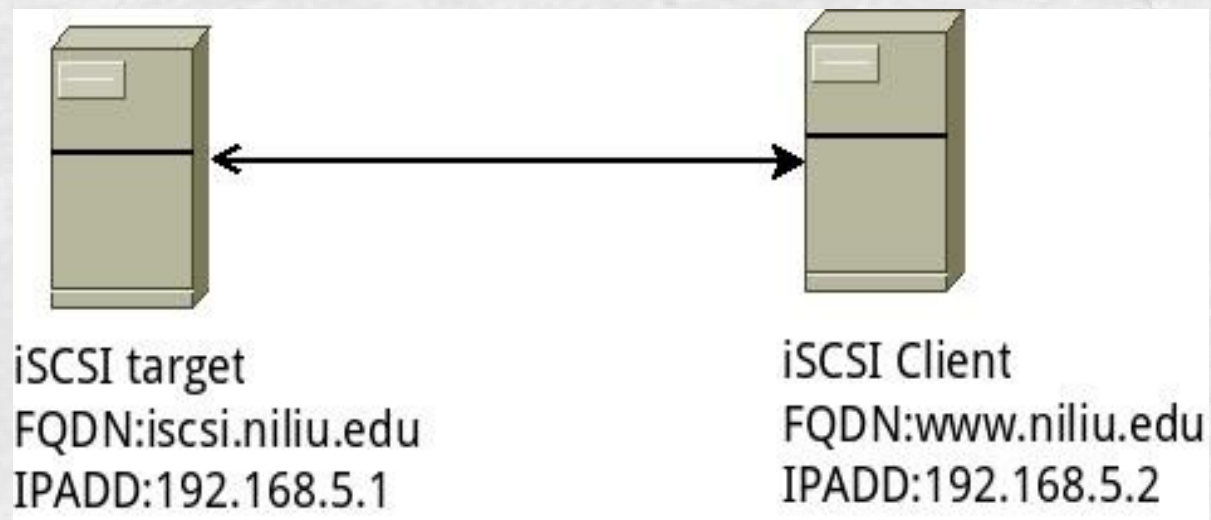
1) 可共享的硬盘类型

- (1) dd 指令所建立的 img 文件 /targetcli 中生成
- (2) 单独分区
- (3) 完整磁盘
- (4) 磁盘阵列
- (5) 软磁盘阵列为单一磁盘
- (6) LVM 的 LV 为磁盘



Inetnet SCSI-iSCSI

iSCSI 服务端实现 实验拓扑图



Inetnet SCSI-iSCSI

iSCSI 服务端实现

1. 准备磁盘

1) 准备 img 类型的硬盘的存放目录

```
#mkdir -v /iscsihd
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

1. 准备磁盘

2) 建立 iSCSI 所需的四块磁盘

(1) 镜像文件 :disk1.img

(2) 独立分区 :/dev/sdb1

(3) LVM: /iscsi/iscsi01

(4) disk2.img 在 targetcli 中实现



Inetnet SCSI-iSCSI

iSCSI 服务端实现

1. 准备磁盘

(1) 镜像文件 :disk1.img

```
#mkdir -v /srv/iscsi
```

```
#dd if=/dev/zero of=/iscsihd/disk01.img bs=1M count=500
```

```
#du -sh disk01.ming
```

如开启 **Selinux** 可加入 :

```
#chcon -Rv t tgtd_var_lib_t /srv/iscsi
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

1. 准备磁盘

(2) 独立分区 `:/dev/sdb1`

```
#fdisk /dev/sdb
```

```
#fdisk -l
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

1. 准备磁盘

(3) LVM: /iscsi/iscsi01

```
#fdisk /dev/sdc
```

System ID 改为 8e(LVM)

```
#fdisk -l
```

```
#pvcreate /dev/sdb2
```

```
#vgcreate -s 4M iscsi /dev/sdb2
```

```
#lvcreate -L 2G -n iscsi01 iscsi
```

```
#lvscan
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

2. targetcli 命令说明

cd 进入某个目录

ls 列示某个目录

create 创建指定目标

delete 删除指定目标

exit 退出

pwd 显示当前工作目录

set 对指定的项目进行相关设定

status 查看当前状态



Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

```
#targetcli
```

(1) 进入”后台存储文件”目录下的“fileio”目录

```
>cd backstores/fileio
```

(2) 创建 LUNS(将准备的硬盘加入至 iSCSI 服务中)

```
/backstores/fileio>create disk01 /iscsihd/disk01.img
```

Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(2) 创建 LUNS(将准备的硬盘加入至 iSCSI 服务中)

```
/backstores/fileio>create disk02 /dev/sdb1
```

```
/backstores/fileio>create disk03 /dev/iscsi/iscsi01
```

* 创建第四块硬盘 , 空间为 2G

```
/backstores/fileio>create disk04 /iscsihd/disk04.img 2G
```

```
/backstores/fileio>ls
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(3) 进入 /iscsi 目录

```
/backstores/fileio>cd /iscsi
```

(4) 设定 iqn.

*iqn 的格式为 iqn. 年 - 月 .FQDN 倒写 :iscsi 存储

名

```
/iscsi>create iqn.2014-10.edu.niliu.www:niliustores1
```

Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(5) 开启 iscsi 服务及监听端口 (TCP:3260)

```
/iscsi>cd "you_iqn_name" /tpg1/portals
```

```
/iscsi/...portals>create
```

删除 IP 及端口：

```
/iscsi/...portals>delete ip_address=0.0.0.0 ip_port=3260
```

Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

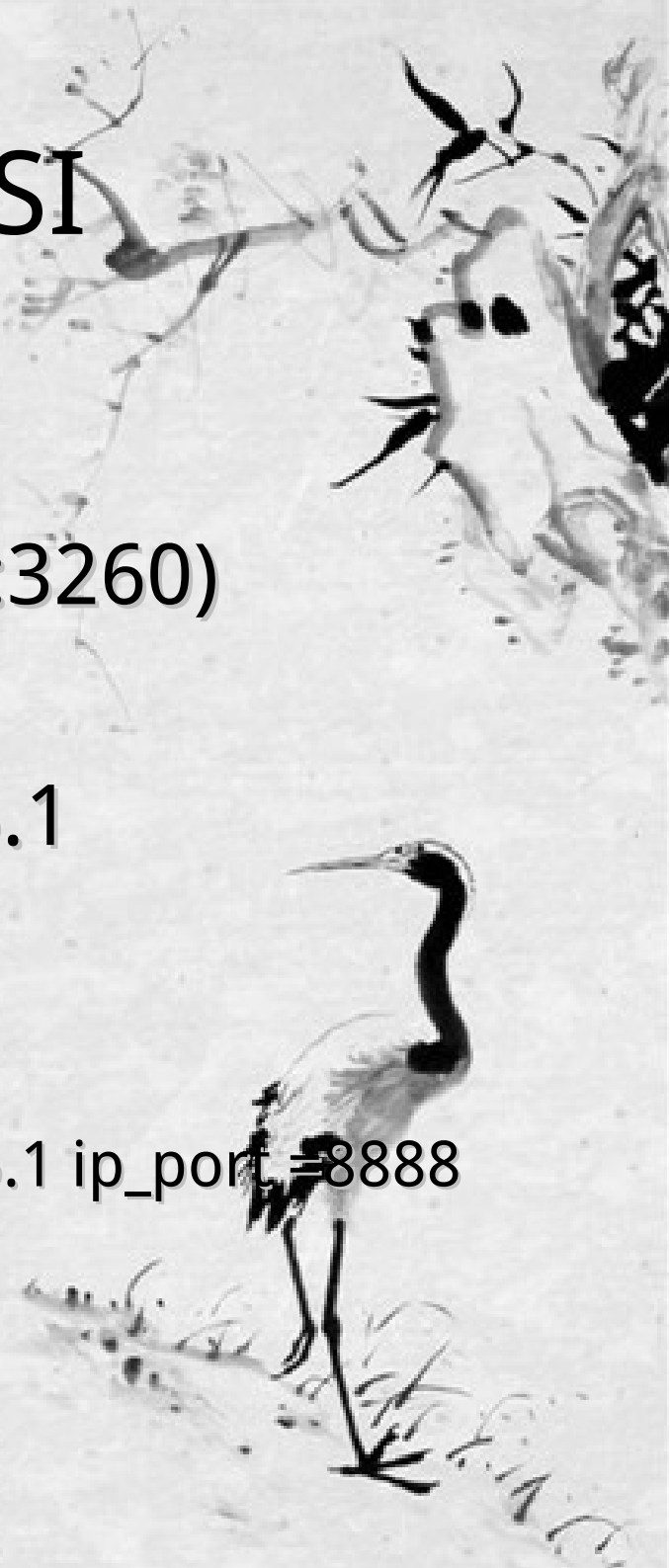
(5) 开启 iscsi 服务及监听端口 (TCP:3260)

指定某个 IP

```
/iscsi/...portals>create 192.168.5.1
```

指定某个 IP 的某个端口

```
/iscsi/...portals>cretea ip_address=192.168.5.1 ip_port=8888
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(6) 配置 LUNS(增加所需要的硬盘)

```
/iscsi/...portals>cd ../luns
```

```
/iscsi/...luns>create /backstores/fileio/disk01
```

```
/iscsi/...luns>create /backstores/fileio/disk02
```

```
/iscsi/...luns>create /backstores/fileio/disk03
```

```
/iscsi/...luns>create /backstores/fileio/disk04
```

Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(7) 设置访问权限

关闭全局认证

```
/iscsi/...luns>cd ../
```

```
/iscsi/...tpg1/>set attribute authentication = 0
```

生成节点 acl

```
/iscsi/...tpg1>set attribute generate_node_acls = 1
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(7) 设置访问权限

```
/iscsi/...portals>cd acs
```

* 创建 iqn 的 FQDN

```
/iscsi/...acs>create “you_iqn_name:FQDN 整写”
```

如

```
/iscsi/...acs>create iqn.2014-10.edu.niliu.www:www.niliu.edu
```

```
/iscsi/...acs>ls
```

```
/iscsi/...acs>cd iqn.2014-10.edu.niliu.www:www.niliu.edu
```



Inetnet SCSI-iSCSI

iSCSI 服务端实现

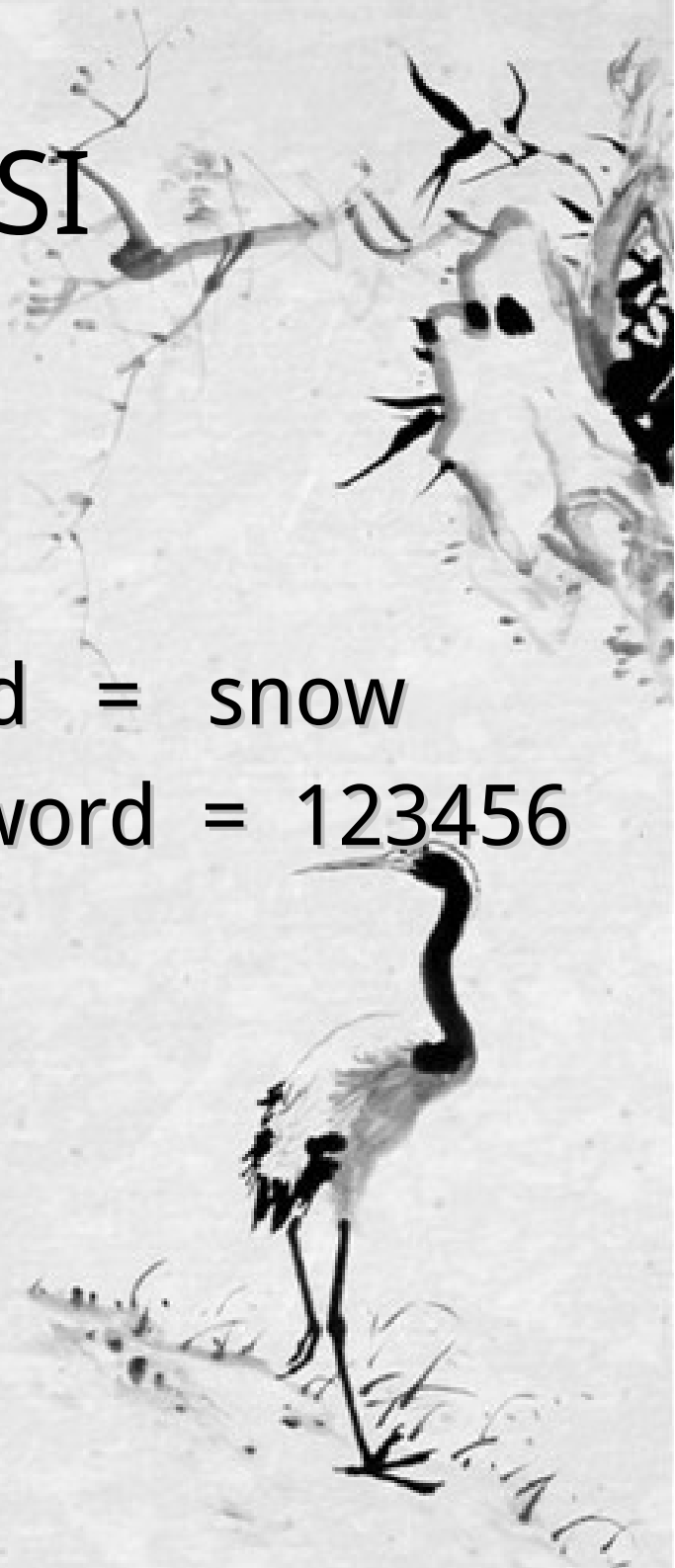
3. 配置 iSCSI 服务的生成

(7) 设置访问权限

```
/iscsi/...niliu.edu>set auth   userid = snow
```

```
/iscsi/...niliu.edu>set auth   password = 123456
```

* 用户可以非系统有效用户



Inetnet SCSI-iSCSI

iSCSI 服务端实现

3. 配置 iSCSI 服务的生成

(8) 保存退出

* 注意配置的保存路径

(9) 查看服务的启动

```
#netstat -lnat | grep 3260
```

(10) 查看进行

```
#ps aux | grep targetcli
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

1. 安装 iSCSI 客户端工具

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

2. 配置验证区域名称及密码

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

将所需要验证的区域写入

如：

```
InitiatorName=iqn.2014-10.edu.niliu.www:www.niliu.edu
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

2. 配置验证区域名称及密码

2) 添加账户和密码

```
#vi /etc/iscsi/iscsid.conf
```

将 54 行开启：启动 CHAP 认证方式

```
#node.session.auth.authmethod = CHAP
```

将 58 行开启：写入合法的验证用户账号

```
#node.session.auth.username = username
```

将 59 行开启：写入合法的密码

```
#node.session.auth.password = password
```

Inetnet SCSI-iSCSI

iSCSI 客户端实现

2. 配置验证区域名称及密码

*)CHAP

询问握手认证协议（ CHAP ）通过三次握手周期性的校验对端的身份，在初始链路建立时完成，可以在链路建立之后的任何时候重复进行。



Inetnet SCSI-iSCSI

iSCSI 客户端实现

2. 配置验证区域名称及密码

*)CHAP 简述

- 1) 链路建立阶段结束之后，认证者向对端点发送“challenge”消息。
- 2) 对端点用经过单向哈希函数计算出来的值做应答。

Inetnet SCSI-iSCSI

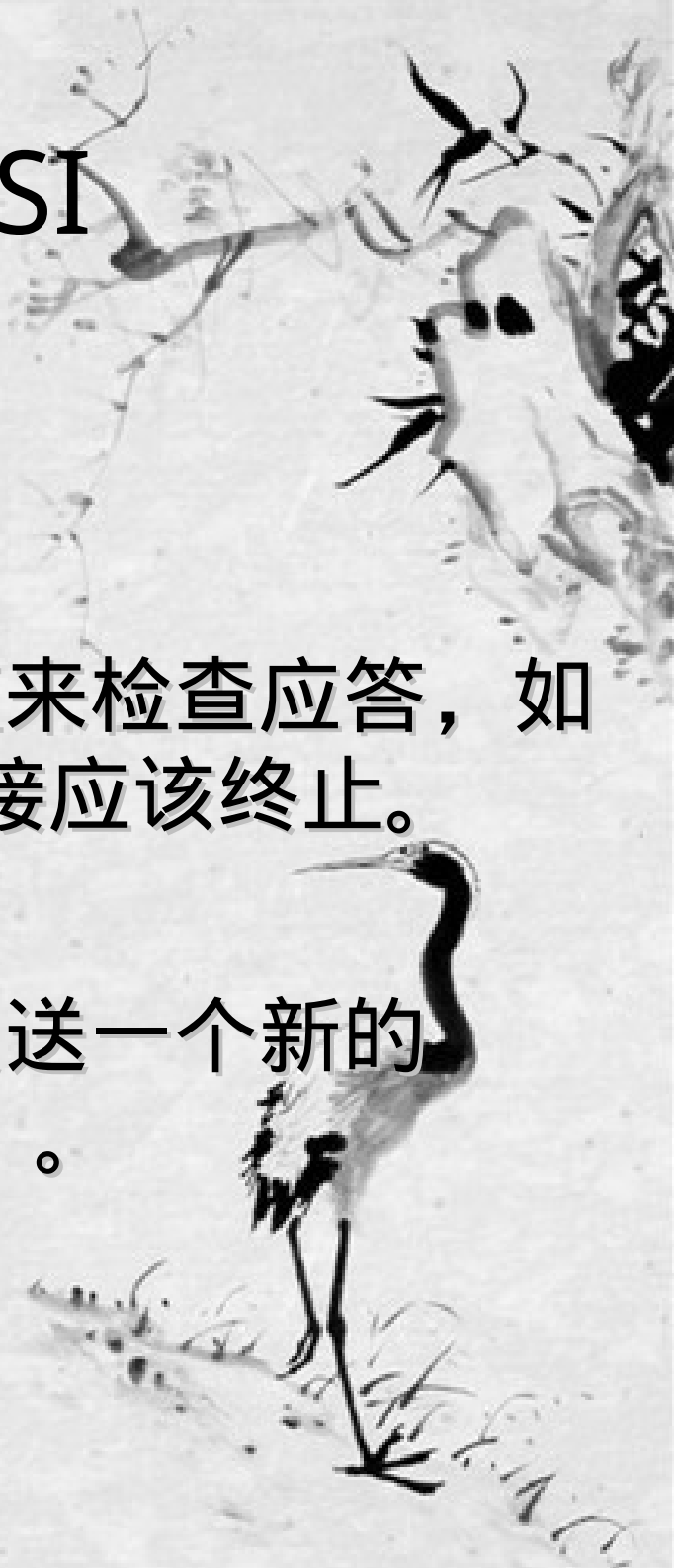
iSCSI 客户端实现

2. 配置验证区域名称及密码

*)CHAP 简述

3) 认证者根据它自己计算的哈希值来检查应答，如果值匹配，认证得到承认；否则，连接应该终止。

4) 经过一定的随机间隔，认证者发送一个新的 challenge 给端点，重复步骤 1 到 3 。



Inetnet SCSI-iSCSI

iSCSI 客户端实现

2. 配置验证区域名称及密码

*)CHAP 简述

5) 通过递增改变的标识符和可变的询问值，CHAP防止了来自端点的重放攻击，使用重复校验可以限制暴露于单个攻击的时间。认证者控制验证频度和时间。

Inetnet SCSI-iSCSI

iSCSI 客户端实现

2. 配置验证区域名称及密码

3) 启动 iscsid 服务

```
#systemctl restart iscsid
```

4) 永久性启动 iscsid

```
#systemctl enable iscsid
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

3. 查找 iSCSI 服务

```
#iscsiadm -m discovery -t sendtargets -p 192.168.5.1
```

参数	说明
-m discovery	通过检测方式使用 iscsiadmin 命令
-t sendtargets	通过 iscsi 协议，检测设备所拥有的信息
-p ip_addr:prot	指定 Server 的 IP 地址及端口号。默认不写则为 3260

4. 查询本地是否收到 Server 端的 target

```
#ll -R /var/lib/iscsi/nodes
```


Inetnet SCSI-iSCSI

iSCSI 客户端实现

5. 找出目前本机上所有检测到的 target 信息

```
#iscsiadm -m node -o show
```

6. 登陆到 iscsi 空间

```
#iscsiadm -m node -T iqn.2014-10.edu.niliu.www:niliustorage1 --login
```

7. 查看本地磁盘情况

```
#lsblk
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

8. 登出 iscsi 空间

```
#iscsiadm -m node -T iqn.2014-10.edu.niliu.www:niliustorage1 --logout
```

9. 查看本地磁盘情况

```
#lsblk
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

10. 添加 / 删除 / 更新 target

```
#iscsiadmin -m node -T iqn.2014-10.edu.niliu.www:niliustorage1 --logout  
#iscsiadm -m node -o delete -T iqn.2014-10.edu.niliu.www:niliustorage1  
#iscsiadm -m node
```

参数	说明
-o delete	删除
-o update	更新
-o new	增加

Inetnet SCSI-iSCSI

iSCSI 客户端实现

11. 使用 target 空间生成 LVM

```
#fdisk /dev/sdb
```

```
#fdisk /dev/sdc
```

```
#fdisk /dev/sdd
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

11. 使用 target 空间生成 LVM

```
#pvcreate /dev/sd{b,c,d,e}1
```

```
#vgcreate iscsi /dev/sd{b,c,d,e}1
```

```
#vgdiskplay
```

```
#lvcreate -L 空间大小 -n iscsi01 iscsi
```

```
#mkfs.ext4 /dev/iscsi/iscsi01
```

```
#mkdir -v /mnt/iscsi
```

```
#mount /dev/iscsi/iscsi01 /mnt/iscsi
```



Inetnet SCSI-iSCSI

iSCSI 客户端实现

12. 测试

13. 自动挂载

1) 本地确定 IP 无问题, 且自启动后可以直接联系到 iSCSI 服务

2) 将指定的 target 作为自动链接

```
#iscsiadm -m node -T targetiqn.2014-10.edu.niliu.www:niliustorage1 -p 192.168.5.1 -o update -n node.startup -v automatic
```

Inetnet SCSI-iSCSI

iSCSI 客户端实现

13. 自动挂载

3) 修改 fstab

```
#tail -n 1 fstab
```

```
/dev/iscsi/iscsi01 /mnt/iscsi ext4 defaults,_netdev 0 0
```

[注 : _netdev 表示网络启动后在挂载]

4) 服务器端将 target 配置为自启动 (IP 需完成)

```
#systemctl enable target
```