**Morning：**

1、传统存储技术

-SCSI技术：读取远程磁盘数据到本地硬盘、光盘、磁带机等，容量小。

-DAS技术：读写存储。读写占用CPU资源，性能差。

-NAS技术：网络存储，提供文件系统共享。如NFS/HTTP/Samba

磁盘（块设备）-->分区（块设备）-->文件系统（xfs,ext4）

-SAN技术：网络存储，提供块存储。使用光纤传输，速度快。

-iSCSI技术：属于SAN技术，基于TCP/IP协议。将数据读写指令进行封装，通过网络传输到后端存储，后端数据通过网络返回主机。但速度较慢。苏单

1. iSCSI

参考：阶段1 --> 2.ENGINEER --> DAY05 --> Page1

**步骤1：安装iSCSI服务**

[root@proxy ~]# yum -y install targetcli软件

**步骤2：配置iSCSI服务**

1. 准备底层存储磁盘

[root@room9pc01 ~]# virt-manager

#打开虚拟机管理器

[root@proxy ~]# parted /dev/vdb mklabel gpt

[root@proxy ~]# parted /dev/vdb mkpart primary 1 100%

#非交互式分区

1. 使用targetcli定义后端存储

/> backstores/block create

1. 创建iqn对象（可以创建多个）

/> /iscsi create

#iqn.时间.反转域名.string(共享描述database):string(磁盘描述vdb)

4）授权客户机访问（设置ACL访问控制）

/> iscsi/iqn.2018-01.cn.tedu:server1/tpg1/acls create

1. 绑定存储

/>iscsi/iqn.2018-01.cn.tedu:server1/tpg1/luns create

6）存储绑定服务监听的地址

#portals默认为0.0.0.0，任意IP（可以不写）

/> saveconfig

/> exit

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

#targetcli实质上是修改上述配置文件

**步骤3：启动target服务**

1）启动服务

[root@proxy ~]# systemctl {start|enable} target

2）查看端口信息

[root@proxy ~]# ss -utlnp | grep :3260

3）关闭防火墙与SELinux

**步骤4：web1客户端访问**

1）客户端安装软件

[root@web1 ~]# yum -y install iscsi-initiator-utils

2）设置本机的iqn名称

[root@web1 ~]# vim /etc/iscsi/initiatorname.iscsi

1. 发现远程target存储

提示：参考man iscsiadm！

[root@web1 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.2.5 --discover

#发现共享

[root@web1 ~]# iscsiadm --mode node --targetname iqn.2018-01.cn.tedu:server1 --portal 192.168.2.5:3260 --login

#登陆iscsi

1. 重启iscsi服务

5）分区、格式化、挂载

**步骤5：拓展实验-FTP使用共享存储**

1. web1安装vsftpd，并将后端存储挂载到/var/ftp
2. 配置/etc/vsftpd/vsftpd.conf打开匿名读写
3. 给/var/ftp/psb/读写权限
4. 真机编写检测脚本：

默认使用web1共享，web1宕机后，web2自动mount挂载iscsi共享，继续提供ftp存储服务。

#iscsi不能同时挂载，因为写操作的数据不能同步

**Afternoon：**

1、Multipath多路径

**步骤1~2：proxy配置共享存储**

（Morning实验2已完成）

**步骤3：web1配置iSCSI客户端**

1）安装客户端软件iscsi-initiator-utils

2）发现存储服务器的共享磁盘（2条链路）

[root@web1 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.2.5 --discover

[root@web1 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.4.5 --discover

3）重启iscsi服务（自动登陆）

[root@web1 ~]# service iscsi restart

[root@web1 ~]# lsblk #出现两个独立的sda、sdb

4）设置开机自启动

[root@web1 ~]# systemctl enable iscsid iscsi

**步骤4：配置Multipath多路径**

1. 安装多路径软件包

[root@web1 ~]# yum list | grep multipath

[root@web1 ~]# yum install -y device-mapper-multipath

2）生成配置文件

[root@web1 ~]# cd /usr/share/doc/device-mapper-multipath-0.4.9

[root@web1 dev..4.9]# cp multipath.conf /etc/

#拷贝配置文件模板到/etc/下

3）获取wwid

[root@web1 ~]# /usr/lib/udev/scsi\_id --whitelisted --device=/dev/sdb

#通过全球唯一的wwid来判定为多路径设备

4）修改配置文件

[root@web1 ~]# vim /etc/multipath.conf

defaults { #自动发现多路径

user\_friendly\_names yes

find\_multipaths yes

}

multipaths { #声明多路径

multipath {

wwid "36001405d0d985652b9a467fb466b07b7"

alias mpatha

}

}

**步骤5：启动Multipath多路径，并测试**

1）启动Multipath，开机自启

[root@web1 ~]# systemctl {start|enable} multipathd

2）检查多路径设备文件

[root@web1 ~]# ls /dev/mapper/

#查看是否生成mpatha

3）对多路径设备分区、格式化、挂载

[root@web1 ~]# fdisk /dev/mapper/mpatha #分区

[root@web1 ~]# ls /dev/mapper/

#再次查看，出现新的分区mpathap1

[root@web1 ~]# mkfs.xfs /dev/mapper/mpathap1 #格式化

[root@web1 ~]# mkdir /data

[root@web1 ~]# mount /dev/mapper/mpathap1 /data/ #挂载

1. 验证多路径

[root@web1 ~]# multipath -rr

#查看多路径，显示sda、sdb都是running状态

[root@web1 ~]# nmcli connection down eth1

[root@web1 ~]# multipath -rr

#关闭网卡0，sdb状态丢失

2、udev

**-设备管理与udev概述**

1）设备管理方法

devfs：静态管理方法，全部设备文件放在/dev下（无论是否添加设备）

udev：动态管理方法，/sys存放实时动态设备，连接后放到/dev下

2）接入设备时间链

-内核发现设备并导入设备状态到sysfs（厂商、型号、uuid等）

-udev接到事件通知

-udev写入设备节点到/dev，或者运行指定程序（mount、cp等）

...

-用户程序可以监控该事件（如lsblk）

3）udev的作用

-从内核收到添加/移除硬件事件时，udev会分析/sys目录下信息，以及/etc/udev/rules.d目录中的规则

-基于分析结果，用什么设备，做什么事情。

**-编写udev规则**

1. 查看设备属性

动态查看设备属性（插拔动作）：

[root@room9pc01 ~]# udevadm monitor --property

静态查看设备在/sys下的位置：

[root@room9pc01 ~]# udevadm info --query=path --name=/dev/sdb

静态查看设备属性（--path只写/sys下的后两级路径）：

[root@room9pc01 ~]# udevadm info --query=property --path=/block/sdb

静态查看设备所有属性

[root@room9pc01 ~]# udevadm info --query=all --attribute-walk --path=/block/sda

2）编写udev规则文件（实现广播信息）

[root@proxy ~]# vim /etc/udev/rules.d/70-usb.rules

SUBSYSTEMS=="usb",ATTRS{serial}=="00241D8CE480B1A0493B3B4C",RUN+="/usr/bin/wall haha"

#RUN+=在原来执行动作基础下追加，写绝对路径命令

#/usr/bin/wall在本机终端广播消息

排错方法：通过查看/var/log/messages日志文件排错。

3）继续修改规则文件（实现给分区命名）

[root@room9pc01 ~]# udevadm info --query=property --path=/block/sdb/sdb1

#使用property查看属性，在编写规则时加ENV{}；用all不用

[root@room9pc01 ~]# vim /etc/udev/rules.d/70-usb.rules

ACTION=="add",ENV{ID\_SERIAL\_SHORT}=="00241D8CE480B1A0493B3B4C",SYMLINK="GJQ%n"

[root@room9pc01 ~]# ls /dev/GJQ\*

4）继续修改规则文件（修改设备所有者和权限）

[root@room9pc01 ~]# vim /etc/udev/rules.d/70-usb.rules

ACTION=="ad...OWNER="root",GROUP="root",MODE="0644"

[root@room9pc01 ~]# ls -ld /dev/sdb1

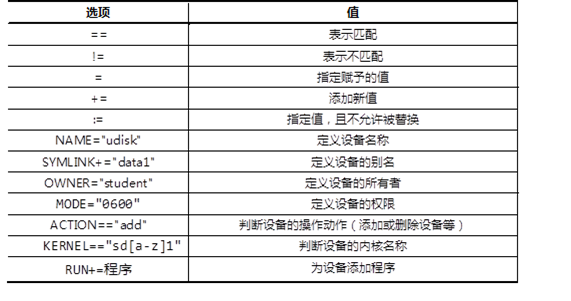
5）继续修改规则文件（插拔U盘等于启停服务）

[root@room9pc01 ~]# vim /etc/udev/rules.d/70-usb.rules

ACTION=="add"...RUN+="/usr/bin/systemctl start httpd"

ACTION=="remove"...RUN+="/usr/bin/systemctl stop httpd"

**-udev常见指令**



udev常用替代变量：

%k：内核所识别出来的设备名，如sdb1

%n：设备的内核编号，如sda3中的3

%p：设备路径，如/sys/block/sdb/sdb1

%%：%符号本身

3、NFS共享

**-实验要求**

1）将目录/root共享给192.168.2.100，客户机的root用户有权限写入

2）将/usr/src目录共享给192.168.2.0/24网段，只开放读取权限

**步骤1：配置NFS服务端**

[root@proxy ~]# rpm -q nfs-utils rpcbind #支持RPC协议

[root@proxy ~]# vim /etc/exports

/root 192.168.2.100(rw,no\_root\_squash)

/usr/src 192.168.2.0/24(ro)

#远程用root访问时，自动降级为nfsnobody

#参数no\_root\_squash，不降级

[root@proxy ~]# systemctl {restart|enable} rpcbind

[root@proxy ~]# systemctl {restart|enable} nfs

**步骤2：客户机访问NFS共享**

[root@web1 ~]# systemctl {start|enable} rpcbind #支持RPC协议

[root@web1 ~]# showmount -e 192.168.2.5

[root@web1 ~]# mkdir /root5；mount 192.168.2.5:/root /root5

#建立挂载点，挂载NFS共享

[root@web1 ~]# vim /etc/fstab

.. ..

192.168.2.5:/usr/src nfsdir nfs default,ro 0 0

192.168.2.5:/root root5 nfs default 0 0

#设置永久挂载