**存储技术与应用**

存储概述

存储的目标

存储是将根据不同的应用环境通过采取合理、安全、有效的方式将数据保存到某些介质上并能保证有效的访问

一方面它是数据临时或长期驻留的物理媒介

另一方面，它是保证数据完整安全存放的方式或行为

存储就是把两个方面结合起来，向用户提供一套数据存放解决方案

存储技术分类

SCSI小型计算机系统接口

DAS直连式存储，如本地磁盘

NAS网络存储技术，如NFS/SAMBA

SAN存储区域网络：分成基于IP网络的IP SAN和基于光纤的FC SAN，提供块存储，本地看见是多硬盘了

常见存储技术

scsi技术

作为输入/输出接口

主要用于硬盘、光盘、磁带机等设备

DAS技术

将存储设备通过SCSI接口或光钎通道直接连接到计算机上

不能实现数据与其他主机的共享

占用服务器资源，如CPU、IO等

数据量越大，性能越差

NAS技术

一种专用数据存储服务器，以数据为中心，将存储设备与服务器彻底分离，集中管理数据，从而释放带宽、提高性能、降低总拥有成本，保护投资，通常由独立的操作系统

用户通过TCP/IP协议访问数据

-采用标准的NFS/HTTP/CIFS等

SAN技术

通过光钎交换机、光钎路由器、光钎集线器等设备将磁盘阵列、磁带等存储设备与相关服务器连接起来，形成高速专用网络

组成部分

如路由器、光钎交换机

接口：如SCSI、FC

通信协议：如IP、SCSI

fibre channel

一种适用于千兆数据传输的、成熟而安全解决方案

与传统的scsi相比，fc提供更高的数据传输速率、更远的传输距离，更多的设备连接支持以及更稳定的性能、更简易的安装

FC主要组件

光钎

HBA（主机总线适配器）

FC交换机

FC交换机拓扑

点到点：point-to-point

简单将两个设备互联

已裁定的环路：arbitrated loop

可多达126个设备共享一段信道或环路

交换式拓扑：switched fabric

所有设备通过光纤交换机互联

iscsi技术（提供块设备，例如硬盘、U盘等）

internet scsi

IETF制定的标准，将SCSI数据库映射为以太网数据包

是一种基于IP Storage理论的新型存储技术

将存储行业广泛应用的SCSI接口技术与IP网络相结合

可以在IP网络上构建SAN

最初由cisco和ibm开发

优势

基于IP协议技术的标准

允许网络在TCP/IP协议上传输SCSI命令

相对于FC SAN，iscsi实现的ip san投资更低

解决了传输效率、存储容量、兼容性、开放性、安全性等方面的问题

没有距离限制

客户端

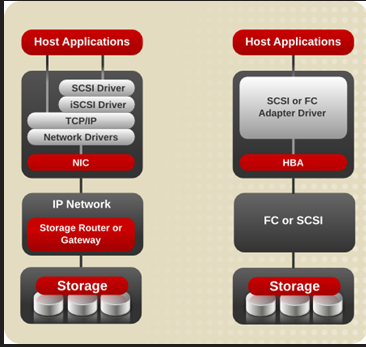
iscsi initiator：软件实现，成本低、性能较低

iscsi hba：硬件实现，性能好，成本较高

存储设备端

iscsi target

以太网交换机



**iscsi技术应用**

基础知识

iscsi操作流程

服务端为客户端提供块设备，需要起名，这个名字是IQN，IQN名称规范：iqn.时间.域名的反写.字符串:字符串，并分配大小格式化，然后客户端启用服务，发现硬盘，使用硬盘

target端（服务器端）

选择target名称

安装iscsi target

准备用于target的存储

配置target

启动服务

initiator（客户端）

安装initiator

配置initiator并启动服务

iscsi命名规范

建议采用IQN（iscsi限定名称）

名称必须全局唯一

IQN格式：

iqn.<date\_code>.<reversed\_domain>.<string>[:<substring>]

iqn.时间.域名的反写.字符串:字符串



部署iscsi服务

添加额外硬盘

安装target软件

查询yum仓库

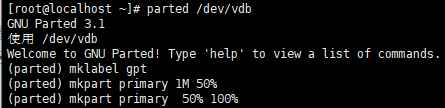
yum list | grep target

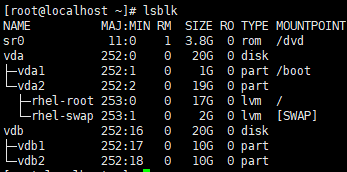
安装

yum -y install targetcli



为vdb分区



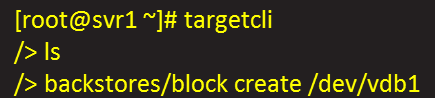


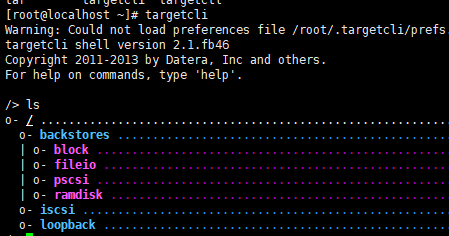
查看iscsi target信息

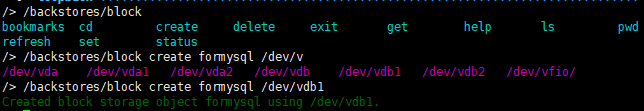
yum info targetcli

配置iscsi target

定义后端存储（给vdb1起个名字formysql,加入到iscsi管理中）

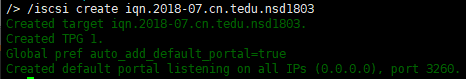




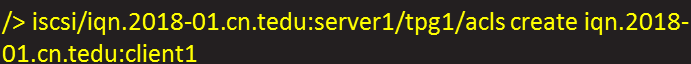


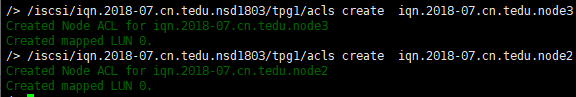
创建iqn对象（创建供客户端访问的iqn名字）





授权客户机访问（配置acl，限定允许访问的客户端）





绑定存储（把存储绑定到iqn名字中）



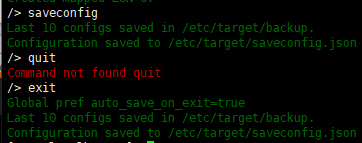


绑定监听地址



保存配置





服务管理

控制服务

systemctl start target

设置开机启动

systemctl enable target

查看端口

netstat -tunlp | grep 3260

安装客户端软件（客户端node2访问并验证）

查询yum仓库

yum list | grep initiator

安装

yum -y install iscsi-initator-utils



查看iscsi target信息

yum info iscsi-initator-utils

启动服务

service iscsi start

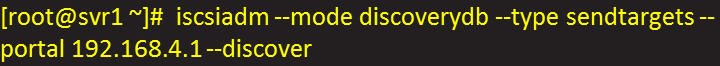
设置本机的iqn名称



+

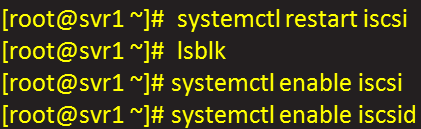


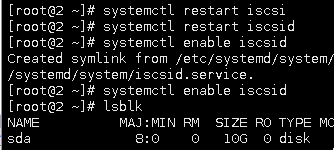
发现远程target存储，此时可以发现名字，但是不会多出存储端





登录iscsi设备，出现新硬盘

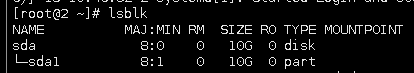




重启后，本地会有sda，第一个是重启服务，第二个是重启守护进程

分区，格式化，进行使用







安装mysql进行检测







客户端node2如果宕掉，可以使用node3进行替换

安装软件包

yum -y install iscsi-initator-utils

systemctl start iscsi





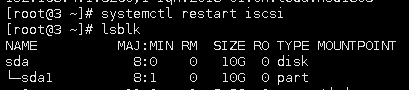
设置本机iqn名称



发现远程target存储，此时可以发现名字，但是不会多出存储端

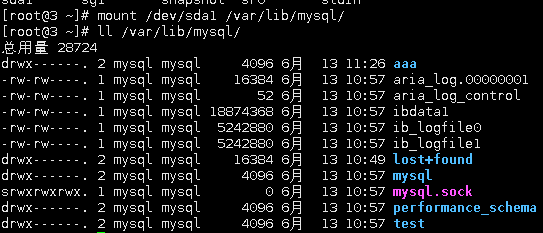


登录iscsi设备，出现新硬盘



安装数据库，进行检验







注意：千万不要使用两个节点同时挂载相同的文件系统（分区），如果多个节点同时挂在这个文件系统，很可能会导致文件系统损坏，数据丢失！因为XFS/EXT3/EXT4这些文件系统都是单节点文件系统。红帽的GFS才能支持多节点同时挂载。

**udev配置**

基础知识

设备文件管理办法

devfs

linux早期采用的静态管理办法

/dev目录下有大量静态文件

内核版本2.6.13开始完全被取代

udev

只有连到系统上来的设备才在/dev下创建设备文件

与主、次设备编号无关

为设备提供持久、一致的名字

如，把U盘接入到主机，主机会多一个硬盘文件，把U盘移除的时候，会少一个硬盘文件

接入设备事件链

内核发现设备并导入设备状态到sysfs

udev接到事件通知

udev创建设备节点或是运行指定程序

udev通知hald守护进程

HAL探测设备信息

HAL创建设备对象结构

HAL通过系统消息总线广播该事件

用户程序也可以监控该事件

udev的作用

从内核收到添加/移除硬件事件时，udev将会分析：

/sys目录下信息

/etc/udev/rules.d目录中的规则

基于分析结果，udev会：

处理设备命名

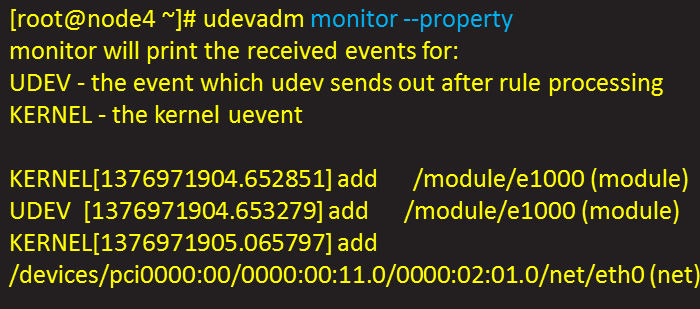
决定要创建哪些设备文件或链接

决定如何设置属性

决定触发哪些事件

udev应用

udev事件监控



配置udev

主配置文件/etc/udev/udev.conf

udev\_root：创建设备文件位置，默认为/dev

udev\_rules：udev规则文件位置，默认为/etc/udev/rules.d

udev\_log：syslog优先级，默认为err

文件格式及位置

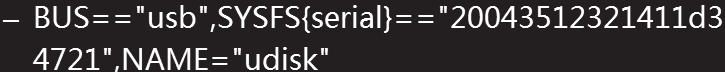
/etc/udev/rules.d/数字-<rule\_name>.rules



数字大小顺序是规则文件执行的顺序

规则格式

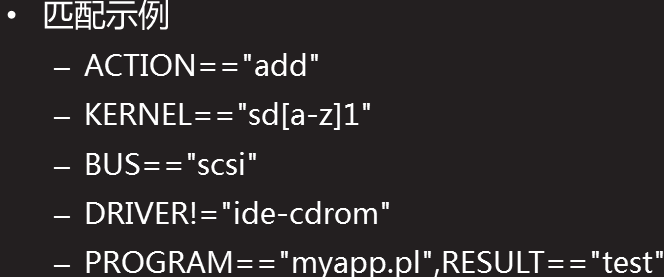
<match-key><op><value>[,...]<assignment=key><op>value[,...]



操作符

= =：表示匹配

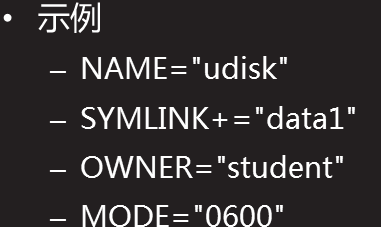
!=：表示不匹配



=：指定赋予的值

+=：添加新值

:=：指定值，且不允许被替换



udev变量

可以简化或缩写规则

KERNEL= =”sda\*”,SYMLINK+=”iscsi%n”

常量替代变量

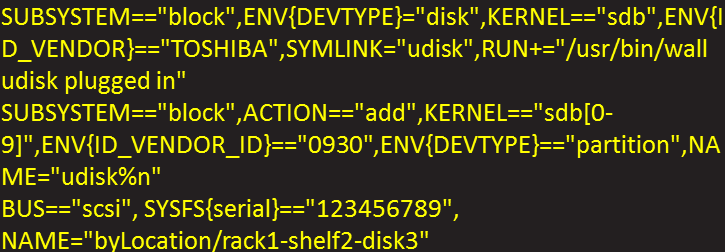
%k：内核所识别出来的设备名，如sdb1

%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

udev案例分析



应用udev，给iscsi创建一个软链接叫idisk

1. 查看iscsi磁盘（sda）在sys目录中的位置

udevadm info --query=path --name=/dev/sda1



1. 通过上一步的路径，查看Iscsi(sda)磁盘的信息（以和其他信息进行区别）

udevadm info --query=all --attribute-walk --path=/devices/platform/host4/session3/target4:0:0/4:0:0:0/block/sda/sda1





1. 创建规则文件





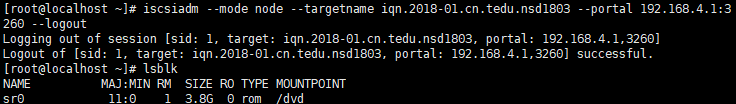
注：KENNEL==”sd[a-z]\*”表示内核识别出来的设备名

ACTION==”add”表示新接入设备

SUBSYSTEMS==”scsi”和ATTRS是上一步查到的属性

SYMLINK表示符号链接，+=表示额外创建，%n是分区号

1. 只有把磁盘移除再接入才能生效或是重启系统



退出scsi



再次查找



重启服务



查看是否成功

**NFS网络文件系统**

**不管是NFS，还是SAMBA，还是FTP，只要是读写共享，必须注意本地权限和配置文件内的授权**

NFS服务基础

文件系统类型

本地文件系统

EXT3/4、SWAP、NTFS、......（本地磁盘）

伪文件系统

/proc、/sys、.....（内存空间）

网络文件系统

NFS（网络存储空间）

NFS共享协议

unix/linux最基本的文件共享机制

1980年由sun公司开发

依赖于RPC（远程过程调用，端口111）映射机制，端口2049，NFS只提供共享功能，底层数据传输交给RPC服务

存取位于远程磁盘中的文档数据，对应用程序是透明的，就好像访问本地的文件一样

配置并访问NFS共享（只读，读写共享）

主要软件包

nfs-utils-1.3.0-0.48.el7.x86\_64

rpcbind-0.2.0-42.el7.x86\_64

系统服务脚本

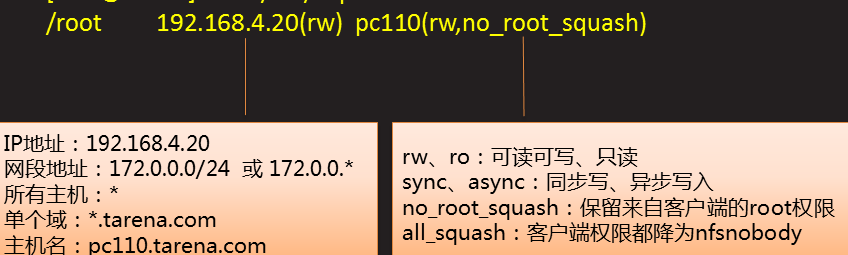
nfs、rpcbind

主配置文件

/etc/exports

/etc/exports配置解析

共享目录 客户机地址(参数,参数,.. ...)...



NFS选项

no\_root\_squash表示保留文件的属主属组还是root，默认写入到共享目录中的文件属主属组是nfsnobody（只对root有用）

all\_squash作用是客户端任何用户写入的文件属主属组都是nfsnody

注意事项：1.从未授权的客户机将无法挂载此共享

2.若未启用no\_root\_squash，挂载后会无权限浏览（750）

只读共享

服务端

1. 安装软件包

yum -y install nfs-utils



1. 创建共享目录并拷贝文件

mkdir -p /nfsroot/nfsro





1. 修改配置文件





1. 启动服务并验证





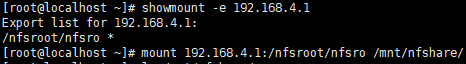
客户端

1. 创建挂载点

mkdir /mnt/nfshare



1. 查看共享并挂载



1. 检查



读写共享

服务端

1. 服务器上创建用于读写的共享目录

mkdir -m 777 /nfsroot/nfsrw





1. 编辑配置文件

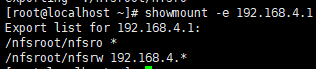




1. 重新输出共享



1. 检查

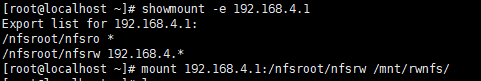


客户端

1. 创建挂载点



1. 挂载测试





**Mulitipath多路径**

基础知识

多路径描述

当服务器到某一存储设备有多条路径时，每条路径都会识别为一个单独的设备

多路径允许您将服务器节点和储存阵列间的多个I/O路径配置为一个单一设备

这些I/O路径是可包含独立电缆、交换机和控制器的实体SAN链接

多路径集合了I/O路径，并生成由这些集合路径组成的欣设备

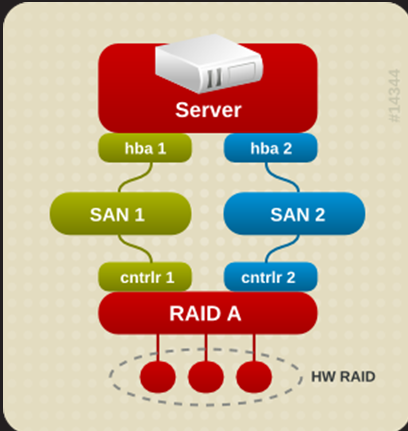
多路径主要功能

冗余

主备模式，高可用

改进性能

主主模式，负载均衡



配置案例

拓扑规划

利用iscsi实现多路径

|  |  |
| --- | --- |
| 应用服务器 | IP地址 |
| eth0 | 192.168.1.10/24 |
| eth1 | 192.168.2.10/24 |

|  |  |
| --- | --- |
| 存储节点 | IP地址 |
| eth0 | 192.168.1.20/24 |
| eth1 | 192.168.2.20/24 |

准备共享存储

配置iscsi服务端

准备共享介质（分区、LV或磁盘镜像）

安装scsi-target-utils

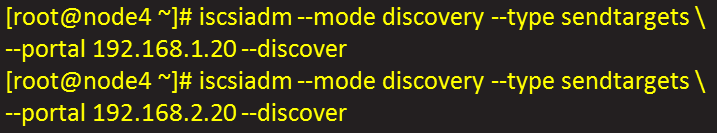
准备规划iqn名称

修改target.conf配置文件，以提供存储

启动tgtd服务

访问共享存储

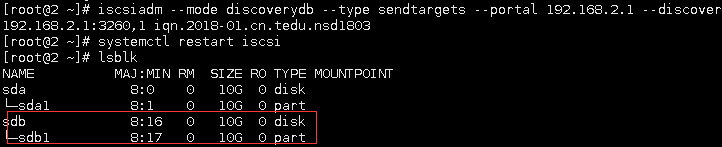
因为达到共享存储有两条路径，所以需要在两条路径上都执行发现命令



设置开机自启

systemctl enable iscsi

systemctl enable iscsid



多路径设备

若没有DM Multipath，从服务器节点到储存控制器的每一条路径都会被系统视为独立的设备，即使I/O路径连接的是相同的服务器节点到相同的储存控制器也是如此

DM Multipath提供了有逻辑的管理I/O路径的方法，即在基础设备顶端生成单一多路径设备

多路径设备识别符

每个多路径设备都有一个WWID（全球识别符），它是全球唯一的，无法更改的号码

默认情况下回将多路径设备的名称设定为它的WWID

可以在多路径配置文件中设置user\_friendly\_names选项，该选项可将别名设为格式为mpathn的节点唯一名称

也可以自定义存储设备名称（合并出来的多路径设备，一般起名为mpath[a-z]）

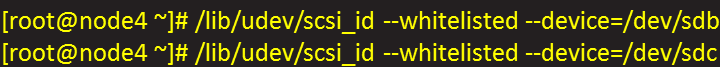
安装软件包

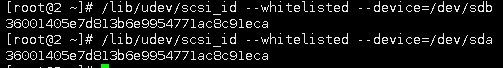
yum -y install device-mapper-multipath



获取WWID

假如共享存储在本地被识别为/dev/sdb和/dev/sdc，那么获取它WWID的方法是





因为两个设备虽然名称不一样，但是实际上是一个设备，所以他们的WWID是相同的

使用mpathconf命令创建配置文件并启用多路径

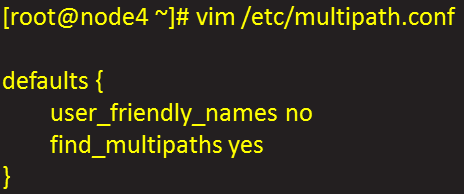
mpathconf --user\_friendly\_names n



若无需编辑该配置文件，可使用此命令启动多路径守护程序

修改配置文件

在配置文件中声明获取WWID的方法

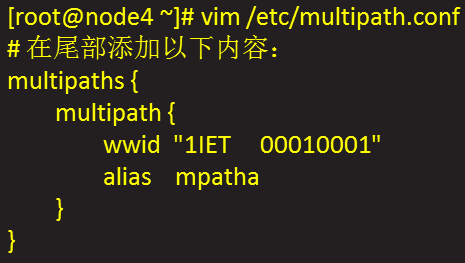


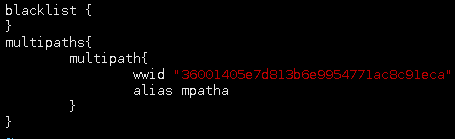




为多路径设备配置别名

根据得到的WWID，为多路径设置配置别名





启动服务并验证

启动服务

systemctl start multipathd

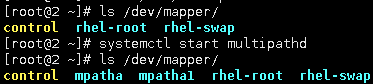
systemctl enable multipathd

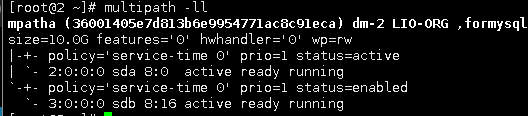
验证

ls /dev/mapper #mpatha即为多路径设备

multipath -rr #重新加载多路径信息

multipath -ll #查看多路径信息





分区

为/dev/mapper/mpatha分区，得到的第一个分区名为/dev/mapper/mpatha1