# SCSI接口

SCSI是小型计算机系统接口（small Computer system interface）的简称，SCSI作为输入输出接口，主要用于磁盘，光盘，磁带机，扫描仪等设备。

## SCSI控制器

普通台式机主板一般不集成SCSI控制器，则必须增加SCSI卡。SCSI卡一段接入主机的PCI总线，另一端用一个SCSI控制器接入SCSI总线。卡上有自己的CPU（频率很低，一般是RISC架构），通过执行ROM中的代码来控制整个SCSI卡的工作。如果这个卡不只有一个SCSI控制器，则每个控制器都可以单独掌管一条SCSI总线，这就是多通道SCSI卡。通道越多，一张卡可接入的SCSI设备就越多。

## SCSI ID

SCSI协议利用了SCSI ID来区分每个节点，在Ultra 320SCSI协议中，一条SCSI总线上可以存在16个节点，其中SCSI控制器占用一个节点，SCSI ID被恒定设置为7，其他15个节点的SCSI ID可以随便设置但是不能重复。

这16个ID中，7具有最高的优先级，也就是说，如果ID 7要发起传输，其他15个ID都必须把总线的使用权交给他。由于总线是共享的，所以同一时刻只能有一个节点向这条总线上放数据。这就涉及到仲裁机制，SCSI接口并不只有8或者16条数据线，还有很多控制信号线。

## LUN ID

SAN是一种基于块的存储，利用高速架构将服务器与其逻辑磁盘单元 (Logical Disk Unit, LUN) 相连。LUN是一系列通过共享存储池配置的块，以逻辑磁盘的形式呈现给服务器。服务器会对这些块进行分区和格式化，通常使用文件系统，以便可以像在本地磁盘存储上一样在LUN上存储数据。

SCSI总线的寻址方式，按照控制器-通道-SCSI ID-LUN ID来寻址。LUN（Logic Unit Number）控制器指的是SCSI控制器，集成在南桥或者单独一个PCI插卡。

在SCSI ID下逻辑划分多个LUN ID，控制器初始化的时候，会对每个SCSI ID上的设备打出一条Report LUN指令，用来收集每个SCSI ID设备的LUN信息。这样，一条SCSI总线上可以接入的最终逻辑存储单元数量就大大增加了。如果一个屋里设备上没有再次划分的逻辑单元，那么这个物理设备必须向控制器报告一个LUN0，代表物理设备本身。对于传统的SCSI总线LUN意义不大，但是对于带RAID功能的SCSI接口磁盘阵列设备来说，由于会产生很多的虚拟磁盘，所以只靠SCSI ID是不够的，就需要LUN来扩充可寻址的范围，所以习惯上称磁盘阵列生成的虚拟磁盘为LUN，由软件生成的虚拟磁盘统一称之为“卷”。

# iSCSI协议

## 简介

iSCSI技术最初由CISCO和IBM两家开发，iSCSI即internet SCSI，是IETF制定的一项标准，用于将SCSI数据库映射成以太网数据包。从根本上来说，它是一种基于IP storage理论的新型存储技术。该技术将存储行业广泛应用的SCSI接口技术与IP技术结合，可以在IP网络上构建SAN。简单的说，就是在TCP/IP协议上传输SCSI命令，实现SCSI和TCP/IP的连接。

iSCSI优势主要由以下几点：

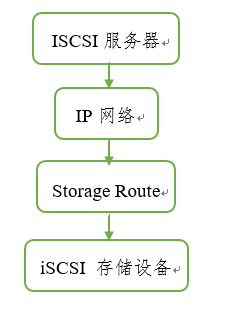
1. 沿用TCP/IP协议，且IP基础建设非常完善，可以长距离传输；
2. SCSI技术是被磁盘和磁带设备最广泛采用的存储标准；
3. 建设和维护成本低廉；
4. iSCSI支持一般的以太网交换机而不是特殊的光纤通道交换机。

## 组成

### iSCSI系统

一个简单的iSCSI系统大致由以下部分构成：iSCSI Initiator或者iSCSI HBA，iSCSI Target，以太网交换机，一台或多台服务器。

结构图如下：



说明：

iSCSI服务器（即连接磁阵的主机）用来安装iSCSI驱动程序，即安装iSCSI Initiator；IP就是以太网；Storage Route可以是以太网交换机或者路由器；iSCSI存储设备可以是iSCSI磁盘阵列，可以是具有存储功能的PC或者服务器。

### iSCSI Initiator

这是安装在计算机上的的软件或者硬件设备，它负责与iSCSI存储设备进行通信，有两种通信方式。

#### 软件形式

即iSCSI Initiator软件，在iSCSI服务器上安装initiator后，initiator软件可以将以太网虚拟成iSCSI卡，进而接收和发送iSCSI服务器上，从而实现主机和iSCSI存储设备之间的iSCSI协议和TCP/IP协议传输功能。这种方式只需要以太网卡和以太网交换机，无需其他的设备，因此成本最低。但是iSCSI报文和TCP/IP报文转换需要消耗服务器的CPU资源，只有在低I/O和低带宽性能要求的应用环境中才能使用这种方式。

#### 硬件形式

使用iSCSI HBA（Host Bus Adapter），即iSCSI Initiator硬件。这种方式需要先购买iSCSI HBA卡，然后将其安装在iSCSI服务器上，从而实现iSCSI服务器与交换机之间、iSCSI服务器与存储设备之间的高效数据传输。与第一种方式比较，硬件iSCSI HBA卡方式不需要消耗iSCSI服务器的CPU资源，同时硬件设备专用的。

所以基于硬件的iSCSI Initiator可以提供更好的数据传输速度和存储性能，但是iSCSI HBA卡价格昂贵，需要使用者在性能和成本之间做权衡。

目前，iSCSI Initiator软件一般都是免费的，CentOS和rhel都能够很好的支持，现在发行的Linux系统都默认自带了iscsi initiator。

#### iSCSI Target

一个可以用于存储数据的iSCSI磁盘阵列或者具有iSCSI功能的设备都被称为“iSCSI Target”，因为大多数操作系统都可以采用一些软件将系统转为一个iSCSI Target。Linux下的开源iSCSI Target软件——iSCSI Enterprise Target。

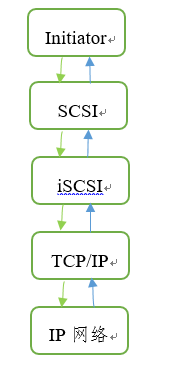
利用iSCSI Target软件，可以将服务器存储空间分配给客户机使用，客户机可以像使用本地磁盘一样使用iSCSI磁盘。包括对齐分区，格式化以及读写等。而且每个客户端可以向iSCSI磁盘写数据，互不干扰，并且不会破坏存储到服务器中的数据。同时，iSCSI Target软件对用户的权限控制非常灵活，支持配置文件。

我们知道，iSCSI是使用TCP/IP协议进行通信的，因此iSCSI两端连接起来，仅仅需要一个以太网络就行了，所以，以太网的速度很大程度上决定了iSCSI的性能，所以我们需要配置一个最好的万兆交换机，再差劲也不能低于千兆交换机，且要做好交换机冗余，避免交换机宕机造成存储无法使用。

## 原理

### iSCSI层次结构

根据OSI模型，iSCSI协议从上往下一共分为三层，如下所示：



SCSI层：根据客户端发来的请求建立SCSI CDB（命令描述块），并且传给iSCSI层，同时接收来自iSCSI层的CDB，并向应用返回数据。

iSCSI层：iSCSI对SCSI CDB进行封装，以便能够在基于TCP/IP协议的网络上进行传输，完成SCSI到TCP/IP的协议映射，这一层是iSCSI协议的核心层。

TCP/IP层：对IP报文进行路由和转发，并且提供端到端的透明可靠传输。

### 传输过程

iSCSI协议定义了在TCP/IP网络发送，接收数据块存储数据的规则和方式，发送端将SCSI命令和数据封装到TCP/IP包中，然后通过IP网络转发，接收端收到TCP/IP包之后，将其还原为SCSI命令和数据并执行，执行完成之后，将返回SCSI命令和数据再封装到TCP/IP包中，之后再传回发送端，这样完成这个传输过程。

iSCSI的整个数据传输过程在用户看来都是透明的，用户使用远端的存储设备就像使用本地的硬盘设备一样，实际上，iSCSI的数据传输速率并不能完全达到本地磁盘的数据传输率，但差别并不明显，而这种网络存储模式还有一点安全性高，这对于数据集中存储的iSCSI来说非常重要。

## 使用

### 环境搭建

准备两台CentOS7服务器做实验：

|  |  |
| --- | --- |
| IP | 安装软件 |
| host1 | iSCSI Target端：scisi-target-utils.x86\_64 |
| host2 | iSCSI客户端：scsi-initiator-uitils |

1、在host1上安装iSCSI Target，采用yum安装：

yum –y install scisi-target-utils

rpm –ql scsi-target-utils

2、编辑配置文件/etc/tgt/targets.conf：

<target iqn.2016-11.com.ljf:server.target1>

backing-store /dev/sdb #/dev/sdb新硬盘，没有格式化和创建文件系统的

</target>

说明：

iqn.2016-11.com.ljf:\server.target1

target后面跟着的名称在同一个子网内部应该是唯一的，标准命名方式是：

iqn,yyyy-mm

iqn=iSCSI Qualified name

yyyy-mm 表示“年份-月份”

reserved domain name：表示倒过来的域名

identified：表示识别代码

backing-store：

backing-store：任何Initiator可以访问

direct-store：可以设定acl进行限制哪些Initiator访问

这里/dev/sdb表示可以把/dev/sdb作为iSCSI盘共享出去。

### 存储端

#### 启动服务

确定配置好后，启动：

service tgtd start

service tgtd status

service –lnpt | grep tgtd

在host2上安装iscsi-initiator-utils：

yum –y install iscsi-initiator-utils

rpm –ql iscsi-initiator-utils

#### 创建target

启动服务之后，需要在存储端创建targer，方便后面映射给客户端使用，具体指令为：

tgtadm --lld iscsi --op new --mode target --tid 1 -T iqn.2020-09.cn.share.dbdisk

说明：

--lld，指明驱动是iscsi

--op(option)，创建新的

--mode，模式是target

--tid，target ID，这个可以任意设置，但是一般遵循规范，iqn-\*\*\*

查看target：

tgtadm --lld iscsi --op show --mode target

创建的target中存在一个LUN（此时显示大小0），不占空间大小，是为了管理存在的。后面创建逻辑单元的时候这里对应显示多个LUN。

#### 创建LUN

创建逻辑单元：

tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 1 -b /dev/sdb1

tgtadm --lld iscsi --op new --mode logicalunit --tid 1 --lun 2 -b /dev/sdb1

这里将磁盘的/dev/sdb1分区划分两个LUN，添加到target上，此时查看target，就显示里面有两个LUN，并且都有对应的空间大小。

#### 授权IP

授权一段IP：tgtadm --lld iscsi --op bind --mode target --tid 1 -l 192.168.50.0/24

#### 新建账号

新建账号：tgtadm --lld iscsi --op new --mode account --user \* --password \*

显示账号信息：tgtadm --lld iscsi --op show --mode account

账号绑定target：tgtadm --lld iscsi --op bind --mode account --tid 2 --user \*

### 客户端

Linux连接网络存储服务的基本操作：

#### 启动服务

下载服务：iscsi-initiator-utils

启动：service iscsi start + chkconfig iscsi on

#### 发现target

发现target：iscsiadm -m discovery -t sendtargets -p \*\*\*\*

注：这里发现target，实际就是扫描/var/lib/iscsi/send\_target。

#### 查看target

查看发现的target：

iscsiadm -m node

#### 登录

iscsi -m node -T iqn.\*\*\*\* -p \*\*\*\*:3260 -l

注：登录到对应的iscsi设备后，然后fdisk -l会扫描多出几个盘，这个就是远端存储的磁盘，我们可以当作普通本地盘一样分区格式化。与此同时，登录服务端，通过查看target信息，也会显示客户端的连接信息。

#### 退出登录

iscsiadm -m node -T iqn.\*\*\*\* -p \*\*\*\*:3260 -u

#### 格式化分区

## 安全配置

## ISCSI命令

### 发现

### 查看

### 删除

### 登入

### 登出