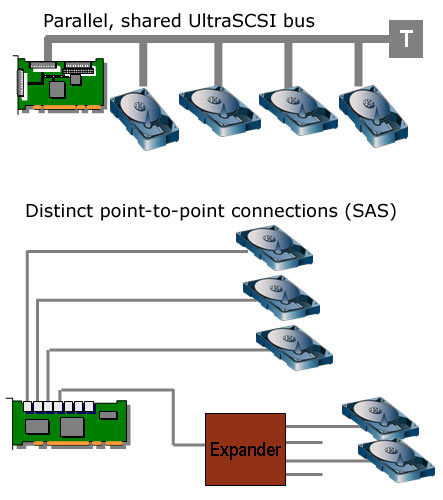
# 1 SCSI

small computer system interface，是一种I/O技术，规范了一种并行的I/O总线和相关的协议，以块的方式传输。



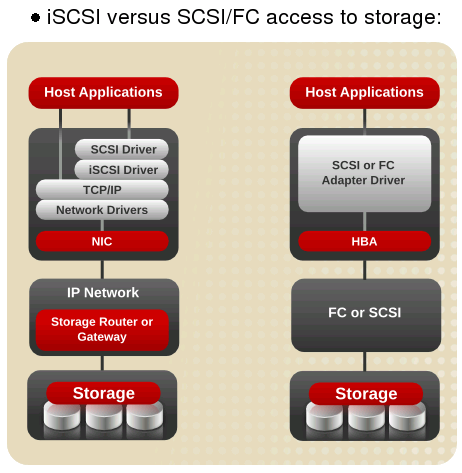
在总线末端存在终结器，一般宽的总线有16个接口或者说target，而窄的总线存在8个接

口，但因为终结器占用了一个接口，导致实际可用的接口为15或者7个。而每个接口下可以连接最多32个LU（logic unit），每个lu可以是一个分区，一块磁盘，或者说一个逻辑卷，每个lu的存在唯一的lun（number）。

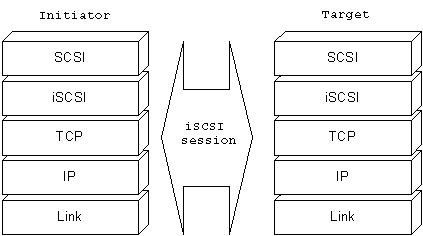
而SAS有取代SCSI的趋势。

# 2 iSCSI

因为scsi的传输距离有限（仅仅只有十几米），为了能弥补这个缺点，研发了fc和iscsi，那么这两者有什么区别？



在一个iSCSI的domain中，存在initiator和target，前者是服务请求放，或者说客户端，后者是服务提供方（服务器端），



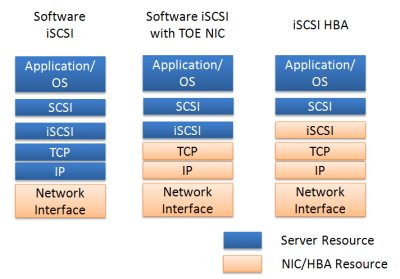
## iSCS Iinitiator类型

1 采用内建SCSI指令及TOE引擎的ASIC芯片的适配卡，在三种iSCSI Initiator中，价格最贵，但性能最佳

2 内建TOE引擎的ASIC芯片适配卡，由于SCSI指令仍以软件方式运作，所以仍会吃掉些许的CPU资源

3 目前不论是Microsoft Windows、IBM AIX、HP-UX、Linux、Novell Netware等各家操作系统，皆已陆续提供这方面的服务，其中以微软最为积极，也最全面。在价格上，比起前两种方案，远为低廉，甚至完全免费

但由于Initiator驱动程序工作时会耗费大量的CPU使用率及系统资源，所以性能最差



## iSCSI Target:

工作在 tcp的3260端口。

targetname:

iqn.yyyy-mm.<reversed domain name>[:identifier]

iqn.2013-05.com.magedu:tstore.disk1

客户端认正方式：

1、基于IP

2、基于用户，CHAP

CHAP是双向认证，c要向s端提出连接时，出示用户密码证明自己，而s端也要 向c端出示用户密码证明自己。

## 如何安装？

准备存储设备，可以是分区、RAID、LVM

安装target scsi-target-utils

将准备好存储设备输出成为IP SAN设备

定义允许哪些客户IP可以登录此IP SAN

安装initiator组件 iscsi-initiator-utils

发现某主机上的IP SAN存储

登录某IP SAN

分区，格式化，挂载

## 命令的使用

**tgtadm模式化的命令**

--mode

常用模式：target、logicalunit、account

target --op

new、delete、show、update、bind、unbind

logicalunit --op

new、delete

account --op

new、delete、bind、unbind

--lld, -L

--tid, -t

--lun, -l

--backing-store <path>, -b

--initiator-address <address>, -I

-T, --targetname <targetname>

**iscsiadm模式化的命令**

-m {discovery|node|session|iface}

discovery: 发现某服务器是否有target输出，以及输出了哪些target；

node: 管理跟某target的关联关系；

session: 会话管理

iface: 接口管理

iscsiadm -m discovery [ -d debug\_level ] [ -P printlevel ] [ -I iface -t type -p ip:port [ -l ] ]

-d: 0-8

-I:

-t type: SendTargets(st), SLP, and iSNS

-p: IP:port

iscsiadm -m discovery -d 2 -t st -p 172.16.100.100

iscsiadm -m node [ -d debug\_level ] [ -L all,manual,automatic ] | [ -U all,manual,automatic ]

iscsiadm -m node [ -d debug\_level ] [ [ -T targetname -p ip:port -I ifaceN ] [ -l | -u ] ] [ [ -o operation ] [ -n name ] [ -v value ] ]