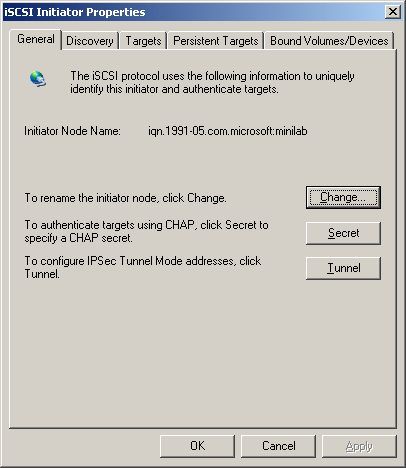
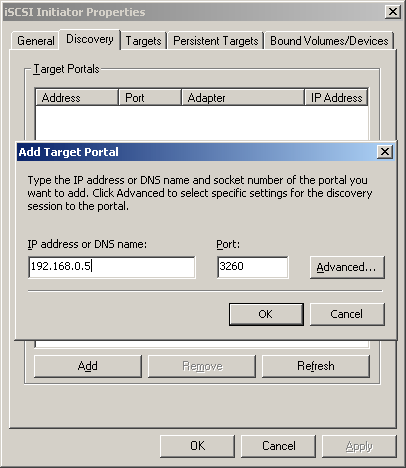
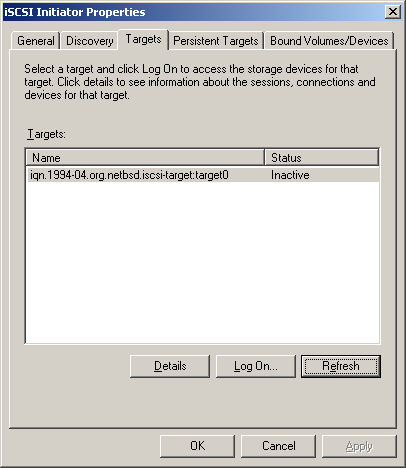
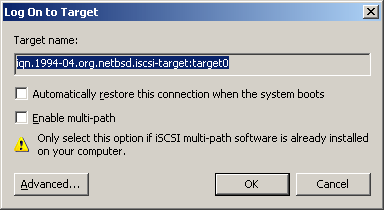
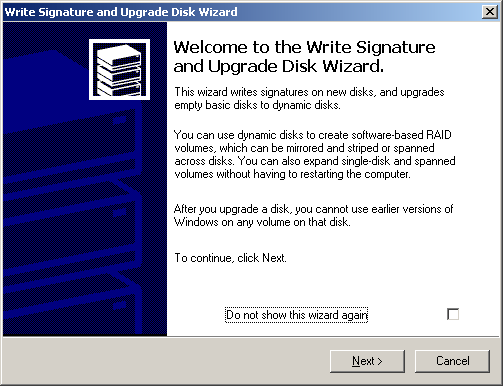
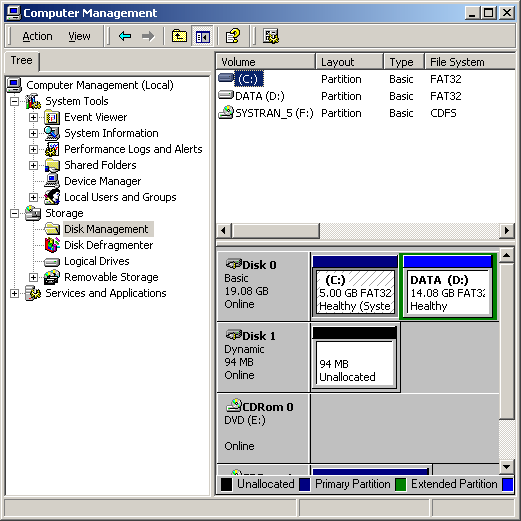
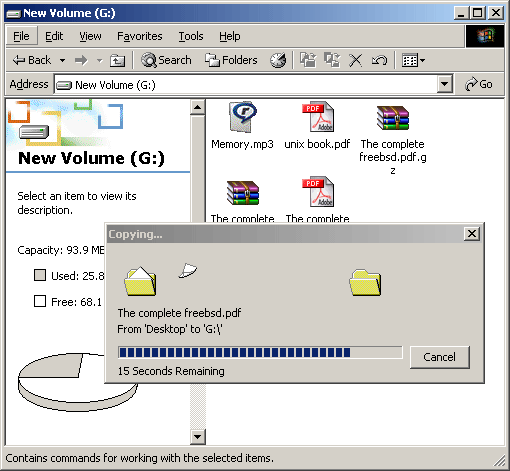
使用NetBSD构建基于iSCSI的小型SAN

引言   
  
随着信息系统应用的复杂和多样化，存储越来越受到[企业](http://www.chkh.com/fwzs.asp)关注，存储的需求也越来越多样化。NetBSD在近期的HEAD（CURRENT）代码中加入了对iSCSI服务的支持，下面就让我们来看看如何利用NetBSD来构建一个基本的SAN（Storage Area Network）环境。   
  
开启iSCSI服务   
  
首先要使用iSCSI服务，你得更新你的系统到HEAD版本，本文使用的NetBSD版本信息如下：   
# uname -a   
NetBSD 3.99.16 NetBSD 3.99.16 (GENERIC) #0: Sun Mar 5 09:59:31 UTC 2006 [email]builds@works[.net](http://www.chkh.com/Article/List_188.html)bsd.org[/email]:/home/builds/ab/HEAD/i386/200603050000Z-obj/home/builds/ab/HEAD/src/sys/arch/i386/compile/GENERIC i386   
如果你的NetBSD版本过低，你可以到ftp://ftp[.net](http://www.chkh.com/Article/List_188.html)bsd.org/pub/NetBSD-daily/HEAD下载安装包，或者使用cvs同步源代码，并升级你的NetBSD，这里就不详细介绍，具体升级方法请参见NetBSD Guide.    
目前NetBSD对iSCSI的支持是用一个用户状态的守护进程iscsi-target来实现的，它位于/usr/sbin/iscsi-target。这里有两个概念需要简单的介绍一下，在服务端提供iSCSI存储服务的存储对像称为target，一个服务端可以有多个target，使用iSCSI服务的客户端称为initiator。   
NetBSD上iSCSI的默认配置[文件](http://www.chkh.com/Soft/List_108.html)为/etc/iscsi/targets，它包含了每个target的描述，它的内容如下：   
# $NetBSD: targets,v 1.2 2006/02/16 19:30:57 agc Exp $   
#   
# Structure of this file:   
#   
# + an extent is a straight (offset, length) pair of a file or device   
# it's the lowest common storage denominator   
# at least one is needed   
# + a device is made up of one or more extents or other devices   
# devices can be added in a hierachical manner, to enhance resilience   
# + in this example, no device definitions are necessary, as the target   
# will just use a simple extent for persistent storage   
# + a target is made up of 1 or more devices   
# The code does not support RAID1 recovery at present    
# Simple file showing 1 extent, mapped straight into 1 target   
# extent file or device start length   
extent0 /tmp/iscsi-target0 0 100MB   
# target flags storage netmask   
target0 rw extent0 0.0.0.0/0   
其中，extent指定了iSCSI数据的存储位置，而target则是iSCSI提供给客户端（initiator）的存储点，客户端将通过它来完成具体的存储请求。    
上面配置文件中的具体含义是：定义了一个extent存储介质extent0，它将使用文件/tmp/iscsi-target0来存储数据，数据存储从[文件](http://www.chkh.com/Soft/List_108.html)头部开始，容量为100MB；定义了一个target存储点target0，权限为rw（读和写），它使用先前定义的存储介质extent0来进行存储，该存储点允许IP段为 0.0.0.0/0 的机器访问。   
接下来是启用iSCSI服务，在/etc/rc.conf中增加如下行：   
iscsi\_target=YES   
接下来便可以启动iSCSI服务了：   
# /etc/rc.d/iscsi\_target start   
如果你的配置正确，你将会看到类似以下的启动信息：   
Starting iscsi\_target.   
Reading configuration from `/etc/iscsi/targets'   
target0:rw:0.0.0.0/0   
extent0:/tmp/iscsi-target0:0:104857600    
DISK: 1 logical units (204800 blocks, 512 bytes/block), type iscsi fs   
DISK: LU 0: 100 MB disk storage for "target0"   
TARGET: TargetName is iqn.1994-04.org[.net](http://www.chkh.com/Article/List_188.html)bsd.iscsi-target   
这表示iSCSI服务已经启动成功，是不是非常简便呢？    
  
客户端连接iSCSI服务   
  
现在iSCSI服务已经开始工作了，下面就是用客户端连接上iSCSI，以检测我们之前的工作是否正确。   
这里我使用的客户端是Microsoft iSCSI Initiator 2.01 x86，你可以从http://download.microsoft.com/download/a/e/9/ae91dea1-66d9-417c-ade4-92d824b871af/Initiator-2.01-QFE-908935-x86fre.exe 得到它，需要注意的是，它只支持Windows 2000 SP3及以上的[windows](http://www.chkh.com/Soft/List_267.html)版本。下载完并安装好后，你会在桌面上看到一个名为Microsoft iSCSI Initiator的图标，双击打开后，可以看到Microsoft iSCSI Initiator窗口，如图1。   
  
图1   
点击Discovery标签，再点击Add按钮，出现增加Target Portal的对话杠，输入你的NetBSD服务器的IP地址，然后点确定，如图2。   
  
图2   
这里就可以看到你的NetBSD服务器的IP已经添加到Target Portals中，点击Targets标签，你将看到刚才我们设置的存储点target0，如果没有出现，请点击Refresh按钮刷新，如图3，点击Log On铵钮，出现登录框，直接点击OK即可，如图4。   
  
图3   
  
图4   
登录成功后，你可以看到Targts中的状态已经变为Connected，同时NetBSD上的/var/log/messages中你也可以看到一条类似以下的initiator登录提示信息：   
Mar 30 09:07:59 iscsi-target: > Normal login from iqn.1991-05.com.microsoft:minilab on 192.168.0.9   
接下来就是初始化iSCSI硬盘了。进入控制面板->管理[工具](http://www.chkh.com/soft/)->计算机管理，点击磁盘管理，这里会出现磁盘签署升级向导，如图5。    
  
图5   
点击下一步按钮，这里列出了要签署的硬盘，选中它，继续点击下一步，出现硬盘升级列表，同样选中它，点击下一步完成升级。这里，计算机管理的磁盘管理器列表中会多出一个硬盘，本例中为Disk1，如图6。   
  
图6   
右击它，创建新的磁盘卷。本例中创建了一个新的卷G，打开我的电脑，你便可以看到它，现在你就可以像使用本地硬盘一样使用它。如图7为向该硬盘（G:）中复制[文件](http://www.chkh.com/Soft/List_108.html)。   
  
图7   
到此，我们便完成了iSCSI服务的建立和客户端的连接，接下来我们继续看看如何使用NetBSD的RAIDframe提高存储性能和保护数据的安全性。   
  
使用RAID5保护iSCSI数据安全   
  
首先要确认你的Kernel是否已经包含对RAID的支持（GENERIC内核默认情况下支持RAID），方法如下：   
# grep RAIDframe /var/run/dmesg.boot   
如果看到：   
Kernelized RAIDframe activated   
说明你的内核已经开启了RAID支持，如果没有看到，你就需要在你的内核配置[文件](http://www.chkh.com/Soft/List_108.html)中加入：   
options RAID\_AUTOCONFIG   
pseudo-device raid   
并重新编译内核。    
RAIDframe使用/sbin/raidctl来对内核RAID进行管理，raidctl将读取一个配置文件来实现对RAID设备的配置，raid配置文件分为4个部分和2个可选部分，每个部分使用START标示该部分的开始。以下是本例中的RAID配置[文件](http://www.chkh.com/Soft/List_108.html)/etc/raid5.conf：   
START array   
1 3 1   
START disks   
/dev/sd0a   
/dev/sd1a   
/dev/sd2a   
START spare   
/dev/sd3a   
START layout   
# sectPerSU SUsPerParityUnit SUsPerReconUnit RAID\_level   
32 1 1 5   
START queue   
fifo 100   
配置[文件](http://www.chkh.com/Soft/List_108.html)的具体意义如下：   
START array   
1 3 1   
这里是一个array的配置，它分别表示该配置为1个RAID，3个磁盘设备，1个备用盘。   
START disks   
/dev/sd0a   
/dev/sd1a   
/dev/sd2a   
这里是一个disks的配置，它列出了要加入到RAID中的硬盘设备为sd0a、sd1a、sd2a。   
START spare   
/dev/sd3a   
这里是一个备用盘的配置，它说明sd3a将作为备用盘加入到RAID中。   
START layout   
# sectPerSU SUsPerParityUnit SUsPerReconUnit RAID\_level   
32 1 1 5   
这里配置了RAID的具体细节，分别表示每个段占用32个扇区，每个校验单位占用1个段，每个重建单位点用1个段，该RAID的级别为RAID5。   
START queue   
fifo 100   
这里主要配置了队列参数，使用fifo队列，每个部分的队列限制为100个请求。    
接下来初始化RAID设备，使用/dev/raid5作为RAID设备：   
# raidctl -C /etc/raid5.conf raid5   
设置启动时自动配置RAID设备：   
# raidctl -A yes raid5   
成功时将返回：   
raid5: Autoconfigure: Yes   
初始化RAID序列号，这里的序列号为22334455，可根据需要改变：   
# raidctl -I 22334455 raid5   
检测RAID是否配置成功：   
# raidctl -s raid5   
Components:   
/dev/sd0a: optimal   
/dev/sd1a: optimal   
/dev/sd2a: optimal   
Spares:   
/dev/sd3a: spare   
Component label for /dev/sd0a:   
Row: 0, Column: 0, Num Rows: 1, Num Columns: 3   
Version: 2, Serial Number: 22334455, Mod Counter: 15   
Clean: No, Status: 0   
sectPerSU: 32, SUsPerPU: 1, SUsPerRU: 1   
Queue size: 100, blocksize: 512, numBlocks: 2097088   
RAID Level: 5   
Autoconfig: Yes   
Root partition: No   
Last configured as: raid5   
Component label for /dev/sd1a:   
Row: 0, Column: 1, Num Rows: 1, Num Columns: 3   
Version: 2, Serial Number: 22334455, Mod Counter: 15   
Clean: No, Status: 0   
sectPerSU: 32, SUsPerPU: 1, SUsPerRU: 1   
Queue size: 100, blocksize: 512, numBlocks: 2097088   
RAID Level: 5   
Autoconfig: Yes   
Root partition: No   
Last configured as: raid5   
Component label for /dev/sd2a:   
Row: 0, Column: 2, Num Rows: 1, Num Columns: 3   
Version: 2, Serial Number: 22334455, Mod Counter: 15   
Clean: No, Status: 0   
sectPerSU: 32, SUsPerPU: 1, SUsPerRU: 1   
Queue size: 100, blocksize: 512, numBlocks: 2097088   
RAID Level: 5   
Autoconfig: Yes   
Root partition: No   
Last configured as: raid5   
/dev/sd3a status is: spare. Skipping label.   
Parity status: DIRTY   
Reconstruction is 100% complete.   
Parity Re-write is 100% complete.   
Copyback is 100% complete.   
上面的信息说明我们的RAID5已经配置成功，接下来是建立disklabel和创建[文件](http://www.chkh.com/Soft/List_108.html)系统：   
# disklabel raid5   
输出：   
# /dev/rraid5d:   
type: RAID   
disk: raid   
label: fictitious   
flags:   
bytes/sector: 512   
sectors/track: 64   
tracks/cylinder: 12   
sectors/cylinder: 768   
cylinders: 5461   
total sectors: 4194176   
rpm: 3600   
interleave: 1   
trackskew: 0   
cylinderskew: 0   
headswitch: 0 # microseconds   
track-to-track seek: 0 # microseconds   
drivedata: 0    
4 partitions:   
# size offset fstype [fsize bsize cpg/sgs]   
a: 4194176 0 4.2BSD 0 0 0 # (Cyl. 0 - 5461\*)   
d: 4194176 0 unused 0 0 # (Cyl. 0 - 5461\*)   
disklabel: boot block size 0   
disklabel: super block size 0   
创建[文件](http://www.chkh.com/Soft/List_108.html)系统   
# newfs -O 2 /dev/raid5a   
这里我们创建的是UFS2的[文件](http://www.chkh.com/Soft/List_108.html)系统。   
接下来加载新的[文件](http://www.chkh.com/Soft/List_108.html)系统：   
# mkdir /r5   
# mount /dev/raid5a /r5   
查看加载情况（这里sd0,sd1,sd2,sd3都为1G的SCSI硬盘）   
# df -h   
Filesystem Size Used Avail Capacity Mounted on   
/dev/wd0a 497M 23M 449M 4% /   
/dev/wd0e 2.2G 189M 1.9G 8% /usr   
kernfs 1.0K 1.0K 0B 100% /kern   
/dev/raid5a 1.9G 2.0K 1.8G 0% /r5   
现在，将/etc/iscsi/targets中的配置[文件](http://www.chkh.com/Soft/List_108.html)改成：   
# $NetBSD: targets,v 1.2 2006/02/16 19:30:57 agc Exp $   
#   
# Structure of this file:   
#   
# + an extent is a straight (offset, length) pair of a file or device   
# it's the lowest common storage denominator   
# at least one is needed   
# + a device is made up of one or more extents or other devices   
# devices can be added in a hierachical manner, to enhance resilience   
# + in this example, no device definitions are necessary, as the target   
# will just use a simple extent for persistent storage   
# + a target is made up of 1 or more devices   
# The code does not support RAID1 recovery at present    
# Simple file showing 1 extent, mapped straight into 1 target   
# extent file or device start length   
extent0 /r5/iscsi-target0 0 1800MB   
# target flags storage netmask   
target0 rw extent0 0.0.0.0/0   
重新启动iSCSI服务：   
# /etc/rc.d/iscsi\_target restart   
现在你的iSCSI数据就已经保存在我们的RAID5设备上了，一个基本的SAN就已经完成，快快体验你的成果吧。