用rsync实现网站镜像和备份

简介  
对于选择Linux 作为应用平台的的中小型企业或网站来说，往往面临如何实现数据远程备份或者网站镜象的问题，虽然有商业化的备份和镜象产品可供选择，但这些产品的价格往往过于昂贵。因此如何利用自由软件高效实现远程备份和网站镜象就成为一个值得讨论的话题。  
  
通过网络进行远程数据备份或者网站镜象的最简单的方法就是使用wget，但是这种方式每次都需要将所有数据都重新在网络上传输一遍，而不考虑哪些文件是经过更新的，因此效率非常低下。尤其在需要备份的数据量很大的时候，往往需要花费数个小时来在网络上进行数据传输。  
  
因此这里就介绍一种高效的网络远程备份和镜象工具-rsync，它可以满足绝大多数要求不是特别严格的备份需求。  
  
rsync是类unix系统下的数据镜像备份工具，从软件的命名上就可以看出来了——remote sync。它的特性如下：  
  
可以镜像保存整个目录树和文件系统。  
可以很容易做到保持原来文件的权限、时间、软硬链接等等。  
无须特殊权限即可安装。  
优化的流程，文件传输效率高。  
可以使用rcp、ssh等方式来传输文件，当然也可以通过直接的socket连接。  
支持匿名传输，以方便进行网站镜象。  
软件下载  
rysnc的主页地址为：  
  
http://rsync.samba.org/  
  
目前最新版本为2.4.6。可以选择从原始网站下载：http://rsync.samba.org/ftp/rsync/。也可以选择从本站下载：rsync 2.4.6。  
  
编译安装  
rsync的编译安装非常简单，只需要以下简单的几步：  
  
[root@www rsync-2.4.6]# ./configure  
[root@www rsync-2.4.6]# make  
[root@www rsync-2.4.6]# make install  
  
但是需要注意的是必须在服务器A和B上都安装rsync，其中A服务器上是以服务器模式运行rsync，而B上则以客户端方式运行rsync。这样在web服务器A上运行rsync守护进程，在B上定时运行客户程序来备份web服务器A上需要备份的内容。  
  
rsync服务器  
1、rsync服务器的启动  
  
在web服务器A上需要以守护进程方式来启动rsync服务器，只需要运行：  
  
[root@www rsync-2.4.6]# /usr/local/bin/rsync --daemon  
  
即可启动。rsync默认服务端口为873，服务器在该端口接收客户的匿名或者认证方式的备份请求。  
  
如果要在启动时把服务起来，有几种不同的方法，比如：  
  
a、加入inetd.conf  
  
编辑/etc/services，加入rsync 873/tcp，指定rsync的服务端口是873。编加/etc/inetd.conf，加入rsync stream tcp nowait root /bin/rsync rsync --daemon  
  
注：对于xinetd，设置方法类似。  
  
b、加入rc.local  
  
编辑/etc/rc.d/rc.local，在最后添加：  
  
/usr/local/bin/rsync --daemon  
  
2、rsync的配置  
  
对于rsync服务器来说，最重要和复杂的就是它的配置了。rsync服务器的配置文件为/etc/rsyncd.conf，其控制认证、访问、日志记录等等。  
  
该文件是由一个或多个模块结构组成。一个模块定义以方括弧中的模块名开始，直到下一个模块定义开始或者文件结束，模块中包含格式为name = value的参数定义。每个模块其实就对应需要备份的一个目录树，比方说在我们的实例环境中，有三个目录树需要备份：/www/、 /home/web\_user1/和/home/web\_user2/，那么就需要在配置文件中定义三个模块，分别对应三个目录树。  
  
配置文件是行为单位的，也就是每个新行都表示一个新的注释、模块定义或者参数赋值。以#开始的行表示注释，以"\"结束的行表示下面一行是该行的继续。参数赋值中等号后可能是一个大小写不敏感的字符串、一个以trure/false表示的布尔值。  
  
全局参数  
  
在文件中[modlue]之前的所有参数都是全局参数，当然也可以在全局参数部分定义模块参数，这时候该参数的值就是所有模块的默认值。  
  
motd file  
  
"motd file"参数用来指定一个消息文件，当客户连接服务器时该文件的内容显示给客户，默认是没有motd文件的。  
  
log file  
  
"log file"指定rsync的日志文件，而不将日志发送给syslog。  
  
pid file  
  
指定rsync的pid文件。  
  
syslog facility  
  
指定rsync发送日志消息给syslog时的消息级别，常见的消息级别是：uth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, sys-log, user, uucp, local0, local1, local2, local3,local4, local5, local6和local7。默认值是daemon。  
  
模块参数  
  
在全局参数之后就需要定义一个或多个模块了，模块中可以定义以下参数：  
  
comment  
  
给模块指定一个描述，该描述连同模块名在客户连接得到模块列表时显示给客户。默认没有描述定义。  
  
path  
  
指定该模块的供备份的目录树路径，该参数是必须指定的。  
  
use chroot  
  
如果"use chroot"指定为true，那么rsync在传输文件以前首先chroot到path参数所指定的目录下。这样做的原因是实现额外的安全防护，但是缺 点是需要以roots权限，并且不能备份指向外部的符号连接所指向的目录文件。默认情况下chroot值为true。  
  
max connections  
  
指定该模块的最大并发连接数量以保护服务器，超过限制的连接请求将被告知随后再试。默认值是0，也就是没有限制。  
  
lock file  
  
指定支持max connections参数的锁文件，默认值是/var/run/rsyncd.lock。  
  
read only  
  
该选项设定是否允许客户上载文件。如果为true那么任何上载请求都会失败，如果为false并且服务器目录读写权限允许那么上载是允许的。默认值为true。  
  
list  
  
该选项设定当客户请求可以使用的模块列表时，该模块是否应该被列出。如果设置该选项为false，可以创建隐藏的模块。默认值是true。  
  
uid  
  
该选项指定当该模块传输文件时守护进程应该具有的uid，配合gid选项使用可以确定哪些可以访问怎么样的文件权限，默认值是"nobody"。  
  
gid  
  
该选项指定当该模块传输文件时守护进程应该具有的gid。默认值为"nobody"。  
  
exlude  
  
用来指定多个由空格隔开的多个模式列表，并将其添加到exclude列表中。这等同于在客户端命令中使用--exclude来指定模式，不过配置文件中指 定的exlude模式不会传递给客户端，而仅仅应用于服务器。一个模块只能指定一个exlude选项，但是可以在模式前面使用"-"和"+"来指定是 exclude还是include。  
  
但是需要注意的一点是该选项有一定的安全性问题，客户很有可能绕过exlude列表，如果希望确保特定的文件不能被访问，那就最好结合uid/gid选项一起使用。  
  
exlude from  
  
指定一个包含exclude模式的定义的文件名，服务器从该文件中读取exlude列表定义。  
  
include  
  
用来指定多个由空格隔开的多个rsync并应该exlude的模式列表。这等同于在客户端命令中使用--include来指定模式，结合 include和 exlude可以定义复杂的exlude/include规则。一个模块只能指定一个include选项，但是可以在模式前面使用"-"和"+"来指定是 exclude还是include。  
  
include from  
  
指定一个包含include模式的定义的文件名，服务器从该文件中读取include列表定义。  
  
auth users  
  
该选项指定由空格或逗号分隔的用户名列表，只有这些用户才允许连接该模块。这里的用户和系统用户没有任何关系。如果"auth users"被设置，那么客户端发出对该模块的连接请求以后会被rsync请求challenged进行验证身份这里使用的 challenge/response认证协议。用户的名和密码以明文方式存放在"secrets file"选项指定的文件中。默认情况下无需密码就可以连接模块(也就是匿名方式)。  
  
secrets file  
  
该选项指定一个包含定义用户名:密码对的文件。只有在"auth users"被定义时，该文件才有作用。文件每行包含一个username:passwd对。一般来说密码最好不要超过8个字符。没有默认的 secures file名，需要限式指定一个。(例如：/etc/rsyncd.secrets)  
  
strict modes  
  
该选项指定是否监测密码文件的权限，如果该选项值为true那么密码文件只能被rsync服务器运行身份的用户访问，其他任何用户不可以访问该文件。默认值为true。  
  
hosts allow  
  
该选项指定哪些IP的客户允许连接该模块。客户模式定义可以是以下形式：  
  
o xxx.xxx.xxx.xxx，客户主机只有完全匹配该IP才允许访问。例如：192.167.0.1  
  
o a.b.c.d/n，属于该网络的客户都允许连接该模块。例如：192.168.0.0/24  
  
o a.b.c.d/e.f.g.h，属于该网络的客户都允许连接该模块。例如：192.168.0.0/255.255.255.0  
  
o 一个主机名，客户主机只有拥有该主机名才允许访问，例如：backup.linuxaid.com.cn。  
  
  
o \*.linuxaid.com.cn，所有属于该域的主机都允许。  
  
默认是允许所有主机连接。  
  
hosts deny  
  
指定不允许连接rsync服务器的机器，可以使用hosts allow的定义方式来进行定义。默认是没有hosts deny定义。  
  
ignore errors  
  
指定rsyncd在判断是否运行传输时的删除操作时忽略server上的IP错误，一般来说rsync在出现IO错误时将将跳过--delete操作，以防止因为暂时的资源不足或其它IO错误导致的严重问题。  
  
ignore nonreadable  
  
指定rysnc服务器完全忽略那些用户没有访问权限的文件。这对于在需要备份的目录中有些文件是不应该被备份者得到的情况是有意义的。  
  
transfer logging  
  
使rsync服务器使用ftp格式的文件来记录下载和上载操作在自己单独的日志中。  
  
log format  
  
通过该选项用户在使用transfer logging可以自己定制日志文件的字段。其格式是一个包含格式定义符的字符串，可以使用的格式定义符如下所示：  
  
o %h 远程主机名  
  
o %a 远程IP地址  
  
o %l 文件长度字符数  
  
o %p 该次rsync会话的进程id  
  
o %o 操作类型："send"或"recv"  
  
o %f 文件名  
  
o %P 模块路径  
  
o %m 模块名  
  
o %t 当前时间  
  
o %u 认证的用户名(匿名时是null)  
  
o %b 实际传输的字节数  
  
o %c 当发送文件时，该字段记录该文件的校验码  
  
默认log格式为："%o %h [%a] %m (%u) %f %l"，一般来说,在每行的头上会添加"%t [%p] "。在源代码中同时发布有一个叫rsyncstats的perl脚本程序来统计这种格式的日志文件。  
  
timeout  
  
通过该选项可以覆盖客户指定的IP超时时间。通过该选项可以确保rsync服务器不会永远等待一个崩溃的客户。超时单位为秒钟，0表示没有超时定义，这也是默认值。对于匿名rsync服务器来说，一个理想的数字是600。  
  
refuse options  
  
通过该选项可以定义一些不允许客户对该模块使用的命令参数列表。这里必须使用命令全名，而不能是简称。但发生拒绝某个命令的情况时服务器将报告错误信息然后退出。如果要防止使用压缩，应该是："dont compress = \*"。  
  
dont compress  
  
用来指定那些不进行压缩处理再传输的文件，默认值是  
  
\*.gz \*.tgz \*.zip \*.z \*.rpm \*.deb \*.iso \*.bz2 \*.tbz  
  
rsync客户命令  
在对rsync服务器配置结束以后，下一步就需要在客户端发出rsync命令来实现将服务器端的文件备份到客户端来。rsync是一个功能非常强大的工具，其命令也有很多功能特色选项，我们下面就对它的选项一一进行分析说明。  
  
首先，rsync的命令格式可以为：  
  
rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST  
  
rsync [OPTION]... [USER@]HOST:SRC DEST  
  
rsync [OPTION]... SRC [SRC]... DEST  
  
rsync [OPTION]... [USER@]HOST::SRC [DEST]  
  
rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST  
  
rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]  
rsync有六种不同的工作模式：  
  
拷贝本地文件；当SRC和DES路径信息都不包含有单个冒号":"分隔符时就启动这种工作模式。  
  
使用一个远程shell程序（如rsh、ssh）来实现将本地机器的内容拷贝到远程机器。当DST路径地址包含单个冒号":"分隔符时启动该模式。  
  
使用一个远程shell程序（如rsh、ssh）来实现将远程机器的内容拷贝到本地机器。当SRC地址路径包含单个冒号":"分隔符时启动该模式。  
  
从远程rsync服务器中拷贝文件到本地机。当SRC路径信息包含"::"分隔符时启动该模式。  
  
从本地机器拷贝文件到远程rsync服务器中。当DST路径信息包含"::"分隔符时启动该模式。  
  
列远程机的文件列表。这类似于rsync传输，不过只要在命令中省略掉本地机信息即可。  
1、用法  
  
在使用rsync传输文件时，需要指定一个源和一个目的，其中一个可能是远程机器的资源信息。例如：  
  
rsync \*.c foo:src/  
  
表示将传输当前目录下所有以.c结尾的文件到机器foo的src目录下。如果任何文件已经存在于远程系统，则会调用远程更新协议来实现仅仅传输那些更新过的文件。  
  
rsync -avz foo:src/bar /data/tmp  
  
该命令则递归地传输机器foo上的src/bar目录下的所有内容到本地/data/tmp/bar目录中。文件以归档模式进行传输，以确保符号链结、属性、权限、属主等信息在传输中都被保存。此外，可以使用压缩技术来加快数据传输：  
  
rsync -avz foo:src/bar/ /data/tmp  
  
路径信息以"/"结尾时表示拷贝该目录，而不以"/"结尾表示拷贝该目录。当配合使用--delete选项时这两种情况的区别将会表现出来。  
  
也可以以本地模式来使用rsync，如果SRC和DST路径中都没有任何":"符号则表示该命令运行在本地模式，等同于cp命令。  
  
rsync somehost.mydomain.com::  
  
这种模式则将会列出somehost.mydomain.com.可以访问的所有模块信息。  
  
选项说明  
  
-v, --verbose 详细模式输出  
-q, --quiet 精简输出模式  
-c, --checksum 打开校验开关，强制对文件传输进行校验  
-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-rlptgoD  
-r, --recursive 对子目录以递归模式处理  
-R, --relative 使用相对路径信息  
  
rsync foo/bar/foo.c remote:/tmp/  
  
则在/tmp目录下创建foo.c文件，而如果使用-R参数：  
  
rsync -R foo/bar/foo.c remote:/tmp/  
  
则会创建文件/tmp/foo/bar/foo.c，也就是会保持完全路径信息。  
  
-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件重新命名为~filename。可以使用--suffix选项来指定不同的备份文件前缀。  
--backup-dir 将备份文件(如~filename)存放在在目录下。  
-suffix=SUFFIX 定义备份文件前缀  
-u, --update 仅仅进行更新，也就是跳过所有已经存在于DST，并且文件时间晚于要备份的文件。(不覆盖更新的文件)  
-l, --links 保留软链结  
-L, --copy-links 想对待常规文件一样处理软链结  
--copy-unsafe-links 仅仅拷贝指向SRC路径目录树以外的链结  
--safe-links 忽略指向SRC路径目录树以外的链结  
-H, --hard-links 保留硬链结  
-p, --perms 保持文件权限  
-o, --owner 保持文件属主信息  
-g, --group 保持文件属组信息  
-D, --devices 保持设备文件信息  
-t, --times 保持文件时间信息  
-S, --sparse 对稀疏文件进行特殊处理以节省DST的空间  
-n, --dry-run现实哪些文件将被传输  
-W, --whole-file 拷贝文件，不进行增量检测  
-x, --one-file-system 不要跨越文件系统边界  
-B, --block-size=SIZE 检验算法使用的块尺寸，默认是700字节  
-e, --rsh=COMMAND 指定替代rsh的shell程序  
--rsync-path=PATH 指定远程服务器上的rsync命令所在路径信息  
-C, --cvs-exclude 使用和CVS一样的方法自动忽略文件，用来排除那些不希望传输的文件  
--existing 仅仅更新那些已经存在于DST的文件，而不备份那些新创建的文件  
--delete 删除那些DST中SRC没有的文件  
--delete-excluded 同样删除接收端那些被该选项指定排除的文件  
--delete-after 传输结束以后再删除  
--ignore-errors 及时出现IO错误也进行删除  
--max-delete=NUM 最多删除NUM个文件  
--partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输  
--force 强制删除目录，即使不为空  
--numeric-ids 不将数字的用户和组ID匹配为用户名和组名  
--timeout=TIME IP超时时间，单位为秒  
-I, --ignore-times 不跳过那些有同样的时间和长度的文件  
--size-only 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间  
--modify-window=NUM 决定文件是否时间相同时使用的时间戳窗口，默认为0  
-T --temp-dir=DIR 在DIR中创建临时文件  
--compare-dest=DIR 同样比较DIR中的文件来决定是否需要备份  
-P 等同于 --partial  
--progress 显示备份过程  
-z, --compress 对备份的文件在传输时进行压缩处理  
--exclude=PATTERN 指定排除不需要传输的文件模式  
--include=PATTERN 指定不排除而需要传输的文件模式  
--exclude-from=FILE 排除FILE中指定模式的文件  
--include-from=FILE 不排除FILE指定模式匹配的文件  
--version 打印版本信息  
--address 绑定到特定的地址  
--config=FILE 指定其他的配置文件，不使用默认的rsyncd.conf文件  
--port=PORT 指定其他的rsync服务端口  
--blocking-io 对远程shell使用阻塞IO  
-stats 给出某些文件的传输状态  
--progress 在传输时现实传输过程  
--log-format=FORMAT 指定日志文件格式  
--password-file=FILE 从FILE中得到密码  
--bwlimit=KBPS 限制I/O带宽，KBytes per second  
-h, --help 显示帮助信息  
实例分析  
这里假设有两台服务器：A和B。其中A是主web服务器，具有域名www.linuxaid.com.cn(202.99.11.120)，B服务器是备 份机，其域名为backup.linuxaid.com.cn(202.99.11.121)。其中A的web内容存放在以下几个地方： /www/和 /home/web\_user1/和/home/web\_user2/。我们需要在备份机B上建立对这几个目录内容的备份。  
  
服务器配置实例  
  
那么在www.linuxaid.com.cn上创建rsyncd的配置文件/etc/rsyncd.conf，内容如下：  
  
uid = nobody  
gid = nobody  
use chroot = no  
max connections = 4  
pid file = /var/run/rsyncd.pid  
lock file = /var/run/rsync.lock  
log file = /var/log/rsyncd.log  
  
[www]  
path = /www/  
ignore errors  
read only = true  
list = false  
hosts allow = 202.99.11.121  
hosts deny = 0.0.0.0/32  
auth users = backup  
secrets file = /etc/backserver.pas  
  
[web\_user1]  
path = /home/web\_user1/  
ignore errors  
read only = true  
list = false  
hosts allow = 202.99.11.121  
hosts deny = 0.0.0.0/32  
uid = web\_user1  
gid = web\_user1  
auth users = backup  
secrets file = /etc/backserver.pas  
  
[web\_user2]  
path = /home/web\_user2/  
ignore errors  
read only = true  
list = false  
hosts allow = 202.99.11.121  
hosts deny = 0.0.0.0/32  
uid = web\_user2  
gid = web\_user2  
auth users = backup  
secrets file = /etc/backserver.pas  
  
这里定义有四个三个模块，分别对应于三个需要备份的目录树。这里只允许202.99.11.121备份本机的数据，并且需要认证。三个模块授权的备份用户都为backup，并且用户信息保存在文件/etc/backserver.pas中，其内容如下：  
  
backup:bk\_passwd  
  
并且该文件只能是root用户可读写的，否则rsyncd启动时会出错。这些文件配置完毕以后，就需要在A服务器上启动rsyncd服务器：  
  
rsync --daemon  
  
客户命令示例  
  
/usr/local/bin/rsync -vzrtopg --delete --exclude "logs/" --exclude "conf/ssl.\*/" --progress backup@202.99.11.120::www /backup/www/ --password-file=/etc/rsync.pass  
  
上面这个命令行中-vzrtopg里的v是verbose，z是压缩，r是recursive，topg都是保持文件原有属性如属主、时间的参数。-- progress是指显示出详细的进度情况，--delete是指如果服务器端删除了这一文件，那么客户端也相应把文件删除，保持真正的一致。-- exclude "logs/" 表示不对/www/logs目录下的文件进行备份。--exclude "conf/ssl.\*/"表示不对/www/conf/ssl.\*/目录下的文件进行备份。  
  
backup@202.99.11.120::www 表示对该命令是对服务器202.99.11.120中的www模块进行备份，backup表示使用backup来对该模块进行备份。  
  
--password-file=/etc/rsync.pass来指定密码文件，这样就可以在脚本中使用而无需交互式地输入验证密码了，这里需要注意的是这份密码文件权限属性要设得只有root可读。  
  
这里将备份的内容存放在备份机的/backup/www/目录下。  
  
[root@linuxaid /]# /usr/local/bin/rsync -vzrtopg --delete --exclude "logs/" --exclude "conf/ssl.\*/" --progress backup@202.99.11.120::www /backup/www/ --password-file=/etc/rsync.pass  
receiving file list ... done  
./  
1  
785 (100%)  
1.py  
4086 (100%)  
2.py  
10680 (100%)  
a  
0 (100%)  
ip  
3956 (100%)  
./  
wrote 2900 bytes read 145499 bytes 576.34 bytes/sec  
total size is 2374927 speedup is 45.34  
  
对其它两个模块操作的命令分别为：  
  
/usr/local/bin/rsync -vzrtopg --delete --progress backup@202.99.11.120::web\_user1 /backup/web\_user1/ --password-file=/etc/rsync.pass  
  
/usr/local/bin/rsync -vzrtopg --delete --progress backup@202.99.11.120::web\_user2 /backup/web\_user2/ --password-file=/etc/rsync.pass  
  
可以将客户命令通过crontab -e命令来实现自动备份，如crontab -e：  
  
  
  
一些示例脚本  
这里这些脚本都是rsync网站上的例子：  
  
1、每隔七天将数据往中心服务器做增量备份  
  
#!/bin/sh  
  
# This script does personal backups to a rsync backup server. You will end up  
# with a 7 day rotating incremental backup. The incrementals will go  
# into subdirectories named after the day of the week, and the current  
# full backup goes into a directory called "current"  
# tridge@linuxcare.com  
  
# directory to backup  
BDIR=/home/$USER  
  
# excludes file - this contains a wildcard pattern per line of files to exclude  
EXCLUDES=$HOME/cron/excludes  
  
# the name of the backup machine  
BSERVER=owl  
  
# your password on the backup server  
export RSYNC\_PASSWORD=XXXXXX  
  
  
########################################################################  
  
BACKUPDIR=`date +%A`  
OPTS="--force --ignore-errors --delete-excluded --exclude-from=$EXCLUDES  
--delete --backup --backup-dir=/$BACKUPDIR -a"  
  
export PATH=$PATH:/bin:/usr/bin:/usr/local/bin  
  
# the following line clears the last weeks incremental directory  
[ -d $HOME/emptydir ] || mkdir $HOME/emptydir  
rsync --delete -a $HOME/emptydir/ $BSERVER::$USER/$BACKUPDIR/  
rmdir $HOME/emptydir  
  
# now the actual transfer  
rsync $OPTS $BDIR $BSERVER::$USER/current  
  
2、备份至一个空闲的硬盘  
  
#!/bin/sh  
  
export PATH=/usr/local/bin:/usr/bin:/bin  
  
LIST="rootfs usr data data2"  
  
for d in $LIST; do  
mount /backup/$d  
rsync -ax --exclude fstab --delete /$d/ /backup/$d/  
umount /backup/$d  
done  
  
DAY=`date "+%A"`  
  
rsync -a --delete /usr/local/apache /data2/backups/$DAY  
rsync -a --delete /data/solid /data2/backups/$DAY  
  
3、对vger.rutgers.edu的cvs树进行镜像  
  
#!/bin/bash  
  
cd /var/www/cvs/vger/  
PATH=/usr/local/bin:/usr/freeware/bin:/usr/bin:/bin  
  
RUN=`lps x | grep rsync | grep -v grep | wc -l`  
if [ "$RUN" -gt 0 ]; then  
echo already running  
exit 1  
fi  
  
rsync -az vger.rutgers.edu::cvs/CVSROOT/ChangeLog $HOME/ChangeLog  
  
sum1=`sum $HOME/ChangeLog`  
sum2=`sum /var/www/cvs/vger/CVSROOT/ChangeLog`  
  
if [ "$sum1" = "$sum2" ]; then  
echo nothing to do  
exit 0  
fi  
  
rsync -az --delete --force vger.rutgers.edu::cvs/ /var/www/cvs/vger/  
exit 0  
  
FAQ  
Q：如何通过ssh进行rsync，而且无须输入密码？  
A：可以通过以下几个步骤  
  
1. 通过ssh-keygen在server A上建立SSH keys，不要指定密码，你会在~/.ssh下看到identity和identity.pub文件  
2. 在server B上的home目录建立子目录.ssh  
3. 将A的identity.pub拷贝到server B上  
4. 将identity.pub加到~[user b]/.ssh/authorized\_keys  
5. 于是server A上的A用户，可通过下面命令以用户B ssh到server B上了  
e.g. ssh -l userB serverB  
这样就使server A上的用户A就可以ssh以用户B的身份无需密码登陆到server B上了。  
  
Q：如何通过在不危害安全的情况下通过防火墙使用rsync?  
A：解答如下：  
  
这通常有两种情况，一种是服务器在防火墙内，一种是服务器在防火墙外。无论哪种情况，通常还是使用ssh，这时最好新建一个备份用户，并且配置 sshd仅允许这个用户通过RSA认证方式进入。如果服务器在防火墙内，则最好限定客户端的IP地址，拒绝其它所有连接。如果客户机在防火墙内，则可以简 单允许防火墙打开TCP端口22的ssh外发连接就ok了。  
  
Q：我能将更改过或者删除的文件也备份上来吗？  
A：当然可以：  
  
你可以使用如：rsync -other -options -backupdir = ./backup-2000-2-13 ...这样的命令来实现。  
这样如果源文件:/path/to/some/file.c改变了，那么旧的文件就会被移到./backup-2000-2-13/path/to/some/file.c，  
这里这个目录需要自己手工建立起来  
  
Q：我需要在防火墙上开放哪些端口以适应rsync？  
A：视情况而定  
  
rsync可以直接通过873端口的tcp连接传文件，也可以通过22端口的ssh来进行文件传递，但你也可以通过下列命令改变它的端口：  
  
rsync --port 8730 otherhost::  
或者  
rsync -e 'ssh -p 2002' otherhost:  
  
Q：我如何通过rsync只复制目录结构，忽略掉文件呢？  
A：rsync -av --include '\*/' --exclude '\*' source-dir dest-dir  
  
Q：为什么我总会出现"Read-only file system"的错误呢？  
A：看看是否忘了设"read only = no"了  
  
Q：为什么我会出现'@ERROR: invalid gid'的错误呢？  
A：rsync使用时默认是用uid=nobody;gid=nobody来运行的，如果你的系统不存在nobody组的话，就会出现这样的错误，可以试试gid = nogroup或者其它  
  
Q：绑定端口873失败是怎么回事？  
A：如果你不是以root权限运行这一守护进程的话，因为1024端口以下是特权端口，会出现这样的错误。你可以用--port参数来改变。  
  
Q：为什么我认证失败？  
A：从你的命令行看来：  
  
你用的是：  
>; bash$ rsync -a 144.16.251.213::test test  
>; Password:  
>; @ERROR: auth failed on module test  
>;  
>; I dont understand this. Can somebody explain as to how to acomplish this.  
>; All suggestions are welcome.  
  
应该是没有以你的用户名登陆导致的问题，试试rsync -a max@144.16.251.213::test test