使用 **rsync** 同步

内容提要

1. 熟悉 rsync 的功能及其特点
2. 掌握 rsync 语法及常用选项的功能
3. 掌握 rsync 命令的三种基本使用方法
4. 掌握如何筛选 rsync 的传输目标
5. 掌握使用 rsync 进行镜像和增量备份的方法

rsync 简介

rsync（remote synchronize）是一个远程数据同步工具，可通过 LAN/WAN 快速同步多台主机之间的文件。也可以使用 rsync 同步本地硬盘中的不同目录。

rsync 是用于替代 rcp 的一个工具，rsync 使用所谓的 **rsync**算法 **[http://www.proxyserve.net /index.php?q=aHR0cDovL3JzeW5jLnNhbWJhLm9yZy90ZWNoX3JlcG9ydC8%3D]** 进行数据同步，这种算法只传送两个文件的不同部分，而不是每次都整份传送，因此速度相当快。 您可以参考 How Rsync Works A Practical Overview [http://www.proxyserve.net/index.php?q=aHR0cDovL3JzeW5jLnNhbWJhLm9yZy9ob3ctcnN5bmMtd29ya3MuaHRtbA%3D%3D] 进一步了解 rsync 的运作机制。

rsync 的初始作者是 Andrew Tridgell 和 Paul Mackerras，目前由 http://rsync.samba.org [http://www.proxyserve.net /index.php?q=aHR0cDovL3JzeW5jLnNhbWJhLm9yZw%3D%3D] 维护。 rsync 支持大多数的类 Unix 系统，无论是 Linux、Solaris 还是 BSD上 都经过了良好的测试。 **CentOS** 系统默认就安装了 **rsync** 软件包。 此外，在 windows 平台下也有相应的版本，如 cwrsync [http://www.proxyserve.net /index.php?q=aHR0cDovL3d3dy5pdGVmaXgubm8vcGhwd3MvaW5kZXgucGhwP21vZHVsZT1wYWdlbWFzdGVyJmFtcDtQQ 和 DeltaCopy [http://www.proxyserve.net /index.php?q=aHR0cDovL3d3dy5hYm91dG15aXAuY29tL0Fib3V0TXlYQXBwL0RlbHRhQ29weS5qc3A%3D] 等。

rsync 具有如下的基本特性：

1. 可以镜像保存整个目录树和文件系统
2. 可以很容易做到保持原来文件的权限、时间、软硬链接等
3. 无须特殊权限即可安装
4. 优化的流程，文件传输效率高
5. 可以使用 rsh、ssh 方式来传输文件，当然也可以通过直接的 socket 连接
6. 支持匿名传输，以方便进行网站镜象

在使用 rsync 进行远程同步时，可以使用两种方式：远程 Shell 方式（建议使用 ssh，用户验证由 ssh 负责）和 C/S 方式（即客户连接远程 rsync 服务器，用户验证由 rsync 服务器负责）。

无论本地同步目录还是远程同步数据，首次运行时将会把全部文件拷贝一次，以后再运行时将只拷贝有变化的文件（对于新文件）或文件的变化部分（对于原有文件）。

本节重点介绍 rsync 客户命令的使用，有关 rsync 服务器的配置和使用请参见下节。

rsync 在首次复制时没有速度优势，速度不如 tar，因此当数据量很大时您可以考虑先使用 tar 进行首次复制，然后再使用 rsync 进行数据同步。

镜像、备份和归档

实施备份的两种情况：

- 需保留备份历史归档：在备份时保留历史的备份归档，是为了在系统出现错误后能恢复到从前正确的状态。这可以使用完全备份和增量备份来完成。
  - 可以使用 tar 命令保存归档文件。
  - 为了提高备份效率，也可以使用 rsync 结合 tar 来完成。

- 无需保留备份历史归档：若无需从历史备份恢复到正确状态，则只备份系统最"新鲜"的状态即可。这可以简单地使用 rsync 同步来完成。此时通常称为镜像。镜像可以分为两种：
  - 被镜像的目录在各个主机上保持相同的位置。此时一般是为了实施负载均衡而对多个主机进行同步镜像。例如：将主机 A 的 /srv/www 目录同步到主机 B 的 /srv/www 目录等。
  - 被镜像的目录在各个主机上不保持相同的位置。例如：主机 A 和主机 B 都运行着各自的业务，同时又互为镜像备份。此时主机 A 的 /srv/www 目录同步到主机 B 的 /backups/hosta/www 目录；主机 B 的 /srv/www 目录同步到主机 A 的 /backups/hostb/www 目录等。

rsync 命令

rsync 是一个功能非常强大的工具，其命令也有很多功能选项。rsync 的命令格式为：

```
1）本地使用：

rsync [OPTION...] SRC... [DEST]

2）通过远程 Shell 使用：

拉：rsync [OPTION...] [USER@]HOST:SRC... [DEST]
推：rsync [OPTION...] SRC... [USER@]HOST:DEST

3）访问 rsync 服务器：

拉：rsync [OPTION...] [USER@]HOST::SRC... [DEST]
推：rsync [OPTION...] SRC... [USER@]HOST::DEST

拉：rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC... [DEST]
推：rsync [OPTION...] SRC... rsync://[USER@]HOST[:PORT]/DEST
```

其中：

- **SRC**：是要复制的源位置
- **DEST**：是复制目标位置
- 若本地登录用户与远程主机上的用户一致，可以省略 USER@
- 使用远程 shell 同步时，主机名与资源之间使用单个冒号":"作为分隔符
- 使用 rsync 服务器同步时，主机名与资源之间使用两个冒号"::"作为分隔符
- 当访问 rsync 服务器时也可以使用 rsync:// URL
- "拉"复制是指从远程主机复制文件到本地主机
- "推"复制是指从本地主机复制文件到远程主机
- 当进行"拉"复制时，若指定一个 SRC 且省略 DEST，则只列出资源而不进行复制

下面列出常用选项：

选项	说明
-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，等价于 -rlptgoD (注意不包括 -H)
-r, --recursive	对子目录以递归模式处理
-l, --links	保持符号链接文件
-H, --hard-links	保持硬链接文件
-p, --perms	保持文件权限
-t, --times	保持文件时间信息
-g, --group	保持文件属组信息
-o, --owner	保持文件属主信息 (super-user only)
-D	保持设备文件和特殊文件 (super-user only)
-z, --compress	在传输文件时进行压缩处理
--exclude=PATTERN	指定排除一个不需要传输的文件匹配模式
--exclude-from=FILE	从 FILE 中读取排除规则
--include=PATTERN	指定需要传输的文件匹配模式
--include-from=FILE	从 FILE 中读取包含规则
--copy-unsafe-links	拷贝指向SRC路径目录树以外的链接文件
--safe-links	忽略指向SRC路径目录树以外的链接文件（默认）

--existing	仅仅更新那些已经存在于接收端的文件，而不备份那些新创建的文件
--ignore-existing	忽略那些已经存在于接收端的文件，仅备份那些新创建的文件
-b, --backup	当有变化时，对目标目录中的旧版文件进行备份
--backup-dir=DIR	与 -b 结合使用，将备份的文件存到 DIR 目录中
--link-dest=DIR	当文件未改变时基于 DIR 创建硬链接文件
--delete	删除那些接收端还有而发送端已经不存在的文件
--delete-before	接收者在传输之前进行删除操作 (默认)
--delete-during	接收者在传输过程中进行删除操作
--delete-after	接收者在传输之后进行删除操作
--delete-excluded	在接收方同时删除被排除的文件
-e, --rsh=COMMAND	指定替代 rsh 的 shell 程序
--ignore-errors	即使出现 I/O 错误也进行删除
--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
--progress	在传输时显示传输过程
-P	等价于 --partial --progress
--delay-updates	将正在更新的文件先保存到一个临时目录（默认为 ".~tmp~"），待传输完毕再更新目标文件
-v, --verbose	详细输出模式
-q, --quiet	精简输出模式
-h, --human-readable	输出文件大小使用易读的单位（如，K，M等）
-n, --dry-run	显示哪些文件将被传输
--list-only	仅仅列出文件而不进行复制
--rsync-path=PROGRAM	指定远程服务器上的 rsync 命令所在路径
--password-file=FILE	从 FILE 中读取口令，以避免在终端上输入口令，通常在 cron 中连接 rsync 服务器时使用
-4, --ipv4	使用 IPv4
-6, --ipv6	使用 IPv6
--version	打印版本信息
--help	显示帮助信息

- 若使用普通用户身份运行 rsync 命令，同步后的文件的属主将改变为这个普通用户身份。
- 若使用超级用户身份运行 rsync 命令，同步后的文件的属主将保持原来的用户身份。

rsync 的基本使用

在本地磁盘同步数据

```
# rsync -a --delete /home /backups
# rsync -a --delete /home/ /backups/home.0
```

在指定复制源时，路径是否有最后的 "/" 有不同的含义，例如：

- /home ： 表示将整个 /home 目录复制到目标目录
- /home/ ： 表示将 /home 目录中的所有内容复制到目标目录

使用基于 ssh 的 rsync 远程同步数据

1、同步静态主机表文件

```
# 执行“推”复制同步（centos5 是可解析的远程主机名）
[root@soho ~]# rsync /etc/hosts centos5:/etc/hosts
# 执行“拉”复制同步（soho 是可解析的远程主机名）
[root@centos5 ~]# rsync soho:/etc/hosts /etc/hosts
```

2、同步用户的环境文件

```
# 执行“推”复制同步
[osmond@soho ~]$ rsync ~/.bash* centos5:
# 执行“拉”复制同步
```

```
[osmond@cnetos5 ~]$ rsync soho:~/.bash* .
```

3、同步站点根目录

```
# 执行“推”复制同步
[osmond@soho ~]$ rsync -avz --delete /var/www root@192.168.0.101:/var/www
# 执行“拉”复制同步
[osmond@cnetos5 ~]$ rsync -avz --delete root@192.168.0.55:/var/www /var/www
```

- 使用基于 ssh 的 rsync 同步数据可以使用 -e ssh 参数，当前的 CentOS 默认指定使用 ssh 作为远程 Shell。若您在其他系统上执行 rsync 命令，为确保使用 ssh 作为远程 Shell，请添加 -e ssh 参数。
- 通常 rsync 命令在后台以 cron 任务形式执行，为了避免从终端上输入口令需要设置 ssh。ssh 的设置方法请参考 安全登录守护进程。

使用 rsync 从远程 rsync 服务器同步数据

下面以镜像 CentOS 和 Ubuntu 的软件库为例来说明。

您可以到如下站点查找离自己最近的提供 rsync 服务的镜像站点

- CentOS — http://www.centos.org/modules/tinycontent/index.php?id=13 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jZW50b3Mub3JnL21vZHVsZXMvdGlueWNvbnRlbnQvaW5kZXgucGhwP2lkPTEz]
- Ubuntu — https://launchpad.net/ubuntu/+archivemirrors [http://www.proxyserve.net/index.php?q=aHR0cHM6Ly9sYXVuY2hwYWQubmV0L3VidW50dS8rYXJjaGl2ZW1pcnJvcnM%3D]

然后执行类似如下命令：

```
rsync -aqzH --delete --delay-updates \
  rsync://mirror.centos.net.cn/centos /var/www/mirror/centos
```

```
rsync -azH --progress --delete --delay-updates \
  rsync://ubuntu.org.cn/ubuntu /var/www/mirror/ubuntu/
rsync -azH --progress --delete --delay-updates \
  rsync://ubuntu.org.cn/ubuntu-cn /var/www/mirror/ubuntu-cn/
```

为了每天不断更新，可以安排一个 cron 任务：

```
# crontab -e
```

```
# mirror centos at 0:10AM everyday
10 0 * * * rsync -aqzH --delete --delay-updates rsync://mirror.centos.net.cn/centos /var/www/mirror/centos
# mirror ubuntu at 2:10AM everyday
10 2 * * * rsync -azH --progress --delete --delay-updates rsync://ubuntu.org.cn/ubuntu /var/www/mirror/ubuntu/
# mirror ubuntu-cn at 4:10AM everyday
10 4 * * * rsync -azH --progress --delete --delay-updates  rsync://ubuntu.org.cn/ubuntu-cn /var/www/mirror/ubuntu-cn/
```

如果您安装了自己的匿名 rsync 服务器请相应地更改 rsync URL。有关如何配置匿名 rsync 服务器的内容请参见下节。

筛选 rsync 的传输目标

使用 --exclude/--include 选项

可以使用 - -exclude 选项排除源目录中要传输的文件；同样地，也可以使用 - -include 选项指定要传输的文件。

例如：下面的 rsync 命令将 192.168.0.101 主机上的 /www 目录（不包含 /www/logs 和 /www/conf 子目录）复制到本地的 /backup/www/ 。

```
# rsync -vzrtopg --delete --exclude "logs/" --exclude "conf/" --progress \
backup@192.168.0.101:/www/ /backup/www/
```

又如：下面的 rsync 命令仅复制目录结构而忽略掉目录中的文件。

```
# rsync -av --include '*/' --exclude '*' \
```

```
backup@192.168.0.101:/www/ /backup/www-tree/
```

选项 - -include 和 - -exclude 都不能使用间隔符。例如：

```
  --exclude "logs/" --exclude "conf/"
```

不能写成

```
  --exclude "logs/ conf/"
```

使用 --exclude-from/--include-from 选项

当 include/exclude 的规则较复杂时，可以将规则写入规则文件。使用规则文件可以灵活地选择传输哪些文件（include）以及忽略哪些文件（exclude）。

- 若文件/目录在剔除列表中，则忽略传输
- 若文件/目录在包含列表中，则传输之
- 若文件/目录未被提及，也传输之

在 rsync 的命令行中使用 - -exclude-from=FILE 或 - -include-from=FILE 读取规则文件。

规则文件 FILE 的书写约定：

- 每行书写一条规则 RULE
- 以 # 或 ; 开始的行为注释行

包含（include）和排除（exclude）规则的语法如下：

```
include PATTERN 或简写为 + PATTERN
exclude PATTERN 或简写为 - PATTERN
```

PATTERN 的书写规则如下：

- 以 / 开头：匹配被传输的跟路径上的文件或目录
- 以 / 结尾：匹配目录而非普通文件、链接文件或设备文件
- 使用通配符
  - *：匹配非空目录或文件（遇到 / 截止）
  - **：匹配任何路径（包含 / ）
  - ?：匹配除了 / 的任意单个字符
  - [：匹配字符集中的任意一个字符，如 [a-z] 或 [[:alpha:]]
  - 可以使用转义字符 \ 将上述通配符还原为字符本身含义

下面给出几个使用规则的例子：

例1：

```
# 不传输所有后缀为 .o 的文件
- *.o
# 不传输传输根目录下名为 foo 的文件或目录
- /foo
# 不传输名为 foo 的目录
- foo/
# 不传输 /foo 目录下的名为 bar 的文件或目录
- /foo/bar
```

例2：

```
# 传输所有目录和C语言源文件并禁止传输其他文件
+ */
+ *.c
- *
```

例3：

```
# 仅传输 foo 目录和其下的 bar.c 文件
```

```
+ foo/
+ foo/bar.c
- *
```

将规则写入规则文件之后，如何在命令行上使用它呢？下面给出一个例子：

首先将下面的规则存入名为 www-rsync-rules 的文件

```
# 不传输 logs 目录
- logs/
# 不传输后缀为 .tmp 的文件
- *.tmp

# 传输 Apache 虚拟主机文档目录（/*/ 匹配域名）
+ /srv/www/
+ /srv/www/*/
+ /srv/www/*/htdocs/
+ /srv/www/*/htdocs/**

# 传输每个用户的 public_html 目录（/*/ 匹配用户名）
+ /home/
+ /home/*/
+ /home/*/public_html/
+ /home/*/public_html/**

# 禁止传输其他
- *
```

然后即可使用类似如下的 rsync 命令：

```
rsync -av --delete --exclude-from=www-rsync-rules / remotehost:/dest/dir
```

rsync 应用示例

使用 rsync 镜像

使用 rsync 对目录做镜像实际上就是做无历史归档的完全备份。下面给出一个镜像远程 Web 站点例子。

笔者在 dreamhost 上维护了3个 Dokuwiki [http://www.proxyserve.net/index.php?q=aHR0cDovL3dpa2kuc3BsaXRicmFpbi5vcmcvd2lraTpkb2t1d2lraQ%3D%3D] 站点。为了备份这3个站点笔者使用 rsync 进行镜像。远程站点的目录结构如下：

```
~
|-- sinosmond.com
|   `-- dokuwiki
|-- smartraining.cn
|   `-- dokuwiki
`-- symfony-project.cn
    `-- dokuwiki
```

每个 Dokuwiki 的目录结构如下：

```
dokuwiki
|-- bin
|-- inc
|-- conf                        ---  存放配置文件的目录
|   |-- acl.auth.php              --- 访问控制配置文件 ★
|   |-- local.php                 --- 本地配置文件 ★
|   |-- users.auth.php            --- 用户口令文件 ★
|   `-- ...................
|-- data                        ---  存放数据的目录
|   |-- attic                     ---  存放WIKI版本信息 ★
|   |-- cache                     ---  存放数据缓存
|   |-- index                     ---  存放站内索引
|   |-- locks                     ---  存放编辑页面时的锁定文件
|   |-- media                     ---  存放图片等 ★
|   |-- meta                      ---  存放 meta 以便系统读取这些信息生成页面 ★
|   `-- pages                     ---  存放 wiki 页面 ★
`-- lib
    |-- plugins                   ---  存放插件的目录 ☆
    |-- tpl                       ---  存放模版的目录 ☆
    `-- ...................
```

为了减少网络流量，只同步标有 ★ 的目录或文件。若在站点运行过程中新安装了插件或更换了模板，也应该同步标有 ☆ 的目录。为此编写如下的规则文件 /root/bin/backup/dw-exclude.txt：

```
- dokuwiki/bin/
- dokuwiki/inc/
- dokuwiki/data/cache/
- dokuwiki/data/locks/
- dokuwiki/data/index/

+ dokuwiki/conf/acl.auth.php
+ dokuwiki/conf/local.php
+ dokuwiki/conf/users.auth.php
- dokuwiki/conf/*

+ dokuwiki/lib/plugins/
# 不同步系统默认安装的插件
- dokuwiki/lib/plugins/acl/
- dokuwiki/lib/plugins/config/
- dokuwiki/lib/plugins/importoldchangelog/
- dokuwiki/lib/plugins/importoldindex/
- dokuwiki/lib/plugins/info/
- dokuwiki/lib/plugins/plugin/
- dokuwiki/lib/plugins/revert/
- dokuwiki/lib/plugins/usermanager/
- dokuwiki/lib/plugins/action.php
- dokuwiki/lib/plugins/admin.php
- dokuwiki/lib/plugins/syntax.php
+ dokuwiki/lib/tpl
# 不同步系统默认安装的模板
- dokuwiki/lib/tpl/default/
- dokuwiki/lib/*

- dokuwiki/COPYING
- dokuwiki/doku.php
- dokuwiki/feed.php
- dokuwiki/index.php
- dokuwiki/install*
- dokuwiki/README
- dokuwiki/VERSION
```

下面是同步脚本 /root/bin/backup/rsync-dw.sh

```
#!/bin/bash
#######################################
# mirror dokuwiki website
# $1 --- domain (ex: smartraining.cn)
# $2 --- full or update
#######################################
# declare some variable
RmtUser=osmond
RmtIP=208.113.163.110
RmtPath=$1/dokuwiki
BackupRoot=/backups/$1
Excludes="--exclude-from=/root/bin/backup/dw-exclude.txt"

# use rsync for mirror
if [ "$2" == "full" ]
then

[ -d /backups/$1 ]  || mkdir -p /backups/$1
excludesfile="/tmp/first-excludes"
cat > ${excludesfile} << EOF
+ dokuwiki/data/cache/_dummy
- dokuwiki/data/cache/*
+ dokuwiki/data/locks/_dummy
- dokuwiki/data/locks/*
+ dokuwiki/data/index/_dummy
- dokuwiki/data/index/*
EOF
/usr/bin/rsync -avzP --exclude-from=${excludesfile} \
$RmtUser@$RmtIP:$RmtPath  $BackupRoot

else

/usr/bin/rsync -avzP --delete $Excludes \
$RmtUser@$RmtIP:$RmtPath      $BackupRoot

fi
```

首次备份可以使用类似如下的命令（为了在本地保留一个完整复本）：

```
# /root/bin/backup/rsync-dw.sh smartraining.cn full
# /root/bin/backup/rsync-dw.sh sinosmond.com full
# /root/bin/backup/rsync-dw.sh symfony-project.cn full
```

可以安排 cron 任务以便日后更新：

```
# crontab -e
```

```
05 1 * * * /root/bin/backup/rsync-dw.sh smartraining.cn
25 1 * * * /root/bin/backup/rsync-dw.sh sinosmond.com
45 1 * * * /root/bin/backup/rsync-dw.sh symfony-project.cn
```

普通型增量备份

使用 rsync 可以做增量备份。rsync 提供了 -b --backup-dir 选项，使用这个选项可以将有变化的文件进行更新同时将其旧版本保存在指定的目录中，从而实现增量备份。 下面是对 /home 进行增量备份的步骤说明：

```
# 第0次备份
# 首先复制 /home 目录的内容到备份目录 /backups/daily/home.0，
# rsync -a /home/ /backups/daily/home.0
# /backups/daily/home.0 总是同步到最新的状态，可以每隔一段时间（如一周）
# 对其内容进行打包压缩生成归档文件（完全备份）存在 /backups/archive/。

# 第1次备份（此为核心操作）
# 将 /home 目录的内容同步到目录 /backups/daily/home.0，
# 并将有变化的文件的旧版本保存到 /backups/daily/home.1，
# 若每天执行一次，则目录 /backups/daily/home.1 保存了有变化文件一天前的状态。
# rsync -a --delete -b --backup-dir=/backups/daily/home.1 /home/ /backups/daily/home.0

# 第2次备份
# 将备份目录 /backups/daily/home.1 更名为 /backups/daily/home.2
# mv /backups/daily/home.1 /backups/daily/home.2
# 执行第1次备份的核心操作

# 第n次备份
# 将早先的备份目录 /backups/daily/home.n 到 /backups/daily/home.1
# 依次更名为  /backups/daily/home.(n+1) 到 /backups/daily/home.2
# 执行第1次备份的核心操作
```

下面给出一个增量备份示例脚本。

```
#!/bin/bash
#=======================
# 您可以安排 cron 任务执行本脚本
# > crontab -e
#
# daily : 1 1 * * * /path/to/script/rsync-backup.sh
#=======================
mydate="`date '+%Y%m%d.%H%M'`"

# Define rmt location
RmtUser=root
RmtHost=192.168.0.55
RmtPath=/home/
BackupSource="${RmtUser}@${RmtHost}:${RmtPath}"
#BackupSource="/home/"              # 若进行本地备份则用本地路径替换上面的行

# Define location of backup
BackupRoot="/backups/$RmtHost/"
# BackupRoot="/backups/localhost/" # 若进行本地备份则用本地路径替换上面的行
LogFile="${BackupRoot}/backup.log"
ExcludeList="/root/backup/backup-exclude-list.txt"

BackupName='home'
BackupNum="7"            # 指定保留多少个增量备份（适用于每周生成归档文件）
#BackupNum="31"          # 指定保留多少个增量备份（适用于每月生成归档文件）

# 定义函数检查目录 $1 是否存在，若不存在创建之
checkDir() {
    if [ ! -d "${BackupRoot}/$1" ] ; then
        mkdir -p "${BackupRoot}/$1"
    fi
}

# 定义函数实现目录滚动
```

```
# $1 -> 备份路径
# $2 -> 备份名称
# $3 -> 增量备份的数量
rotateDir() {
    for i in `seq $(($3 - 1)) -1 1`
      do
      if [ -d "$1/$2.$i" ] ; then
          /bin/rm -rf "$1/$2.$((i + 1))"
          mv "$1/$2.$i" "$1/$2.$((i + 1))"
      fi
    done
}

# 调用函数 checkDir , 确保目录存在
checkDir "archive"
checkDir "daily"

#======= Backup Begin =================
# S1: Rotate daily.
rotateDir "${BackupRoot}/daily" "$BackupName" "$BackupNum"

checkDir "daily/${BackupName}.0/"
checkDir "daily/${BackupName}.1/"

mv ${LogFile} ${BackupRoot}/daily/${BackupName}.1/

cat >> ${LogFile} <<_EOF
==========================================
  Backup done on: $mydate
==========================================
_EOF

# S2: Do the backup and save difference in ${BackupName}.1
rsync -av --delete \
    -b --backup-dir=${BackupRoot}/daily/${BackupName}.1 \
    --exclude-from=${ExcludeList} \
    $BackupSource ${BackupRoot}/daily/${BackupName}.0 \
    1>> ${LogFile} 2>&1

# S3: Create an archive backup every week
if [ `date +%w` == "0" ]          # 每周日做归档
# if [ `date +%d` == "01" ]       # 每月1日做归档
then
  tar -cjf ${BackupRoot}/archive/${BackupName}-${mydate}.tar.bz2 \
    -C ${BackupRoot}/daily/${BackupName}.0 .
fi
```

您可以适当修该上述脚本中变量：

```
RmtPath="$1/"
#BackupSource="$1/"
BackupName="$1"
```

然后传递脚本参数备份其他目录，例如要备份 /www 可以使用如下命令：

```
./rsync-backup.sh /www
```

快照型增量备份

使用 rsync 可以做快照（Snapshot）型增量备份。每一个快照都相当于一个完全备份。其核心思想是：对有变化的文件进行复制；对无变化的文件创建硬链接以减少磁盘占用。

下面是对 /home 进行快照型增量备份的步骤说明：

```
# 第0次备份
# 首先复制 /home 目录的内容到备份目录 /backups/home.0
# rsync -a /home/ /backups/home.0

# 第1次备份（此为核心操作）
# 以硬链接形式复制 /backups/home.0 到 /backups/home.1
# cp -al /backups/home.0 /backups/home.1
# 将 /home 目录的内容同步到目录 /backups/home.0
# （rsync 在发现变化的文件时，先删除之，然后在创建该文件）
# rsync -a --delete /home/ /backups/home.0

# 第2次备份
# 将备份目录 /backups/home.1 更名为 /backups/home.2
```

```
# mv /backups/home.1 /backups/home.2
# 执行第1次备份的核心操作

# 第n次备份
# 将早先的备份目录 /backups/home.n 到 /backups/home.1
# 依次更名为 /backups/home.(n+1) 到 /backups/home.2
# 执行第1次备份的核心操作
```

rsync 2.5.6 版本之后提供了 - -link-dest 选项，如下两条核心操作命令：

```
cp -al /backups/home.0 /backups/home.1
rsync -a --delete /home/ /backups/home.0
```

可以简化为如下的一条命令：

```
rsync -a --delete --link-dest=/backups/home.1 /home/ /backups/home.0
```

下面给出一个快照型增量备份示例脚本，该脚本来自 http://www.mikerubel.org/computers/rsync_snapshots/contributed/peter_schneider-kamp [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5taWtlcnVibWwub3JnL2NvbXB1dGVycy9yc3luY19zbmFwc2hvdHMvY29udHJpYnV0ZWQv

```
#!/bin/bash
# ------------------------------------------------------------------
# mikes handy rotating-filesystem-snapshot utility
# ------------------------------------------------------------------
# RCS info: $Id: make_snapshot.sh,v 1.6 2002/04/06 04:20:00 mrubel Exp $
# ------------------------------------------------------------------
# this needs to be a lot more general, but the basic idea is it makes
# rotating backup-snapshots of /home whenever called
# ------------------------------------------------------------------

# ------------- system commands used by this script --------------------
ID='/usr/bin/id';
ECHO='/bin/echo';

MOUNT='/bin/mount';
RM='/bin/rm';
MV='/bin/mv';
CP='/bin/cp';
TOUCH='/usr/bin/touch';

RSYNC='/usr/bin/rsync';


# ------------- file locations -----------------------------------------

MOUNT_DEVICE=/dev/hdb1;
SNAPSHOT_RW=/root/snapshots;
EXCLUDES=/etc/snapshot_exclude;


# ------------- backup configuration

BACKUP_DIRS="/etc /home"
NUM_OF_SNAPSHOTS=3
BACKUP_INTERVAL=hourly


# ------------- the script itself --------------------------------------

# make sure we're running as root
if (( `$ID -u` != 0 )); then { $ECHO "Sorry, must be root.  Exiting..."; exit; } fi

echo "Starting snapshot on "`date`

# attempt to remount the RW mount point as RW; else abort
$MOUNT -o remount,rw $MOUNT_DEVICE $SNAPSHOT_RW ;
if (( $? )); then
{
       $ECHO "snapshot: could not remount $SNAPSHOT_RW readwrite";
       exit;
}
fi;

# rotating snapshots
for BACKUP_DIR in $BACKUP_DIRS
do
```

```
  NUM=$NUM_OF_SNAPSHOTS

  # step 1: delete the oldest snapshot, if it exists:
  if [ -d ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.$NUM ] ; then                \
    $RM -rf ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.$NUM ;                          \
  fi ;

  NUM=$(($NUM-1))

  # step 2: shift the middle snapshots(s) back by one, if they exist
  while [[ $NUM -ge 1 ]]
  do
    if [ -d ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.$NUM ] ; then                    \
      $MV ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.$NUM ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.$(($NUM+1)) ;     \
    fi;
    NUM=$(($NUM-1))
  done

  # step 3: make a hard-link-only (except for dirs) copy of the latest snapshot,
  # if that exists
  if [ -d ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.0 ] ; then                   \
  $CP -al ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.0 ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.1 ;  \
  fi;

  # step 4: rsync from the system into the latest snapshot (notice that
  # rsync behaves like cp --remove-destination by default, so the destination
  # is unlinked first.  If it were not so, this would copy over the other
  # snapshot(s) too!
  $RSYNC                                                          \
          -va --delete --delete-excluded                          \
          --exclude-from="$EXCLUDES"                          \
          ${BACKUP_DIR}/ ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.0 ;

  # step 5: update the mtime of ${BACKUP_INTERVAL}.0 to reflect the snapshot time
  $TOUCH ${SNAPSHOT_RW}${BACKUP_DIR}/${BACKUP_INTERVAL}.0 ;

done

# now remount the RW snapshot mountpoint as readonly

$MOUNT -o remount,ro $MOUNT_DEVICE $SNAPSHOT_RW ;
if (( $? )); then
{
       $ECHO "snapshot: could not remount $SNAPSHOT_RW readonly";
       exit;
} fi;
```

参考

- http://rsync.samba.org/examples.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3JzeW5jLnNhbWJhLm9yZy9leGFtcGxlcy5odG1s]
- http://sial.org/howto/rsync/ [http://www.proxyserve.net/index.php?q=aHR0cDovL3NpYWwub3JnL2hvd3RvL3JzeW5jLw%3D%3D]
- http://www.linuxsir.org/main/?q=node/256 [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5saW51eHNpci5vcmcvbWFpbi8%2FcT1ub2RlLzI1Ng%3D%3D]
- http://www.dbanotes.net/techmemo/rsync_openssh.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5kYmFub3Rlcy5uZXQvdGVjaG1lbW8vcnN5bmNfb3BlbnNzaC5odG1s]
- http://www.mikerubel.org/computers/rsync_snapshots/index.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5taWtlcnVlZWwub3JnL2NvbXB1dGVycy9yc3luY19zbmFwc2hvdHMvaW5kZXguaHRtbA%3D%3D]
- http://howtoforge.com/rsync_incremental_snapshot_backups [http://www.proxyserve.net/index.php?q=aHR0cDovL2hvd3RvZm9yZ2UuY29tL3JzeW5jX2luY3JlbWVudGFsX3NuYXBzaG90X2JhY2t1cHM%3D]
- http://mirror.actusa.net/pub/sample-files/mirror.dist [http://www.proxyserve.net/index.php?q=aHR0cDovL21pcnJvci5hY3R1c2EubmV0L3B1Yi9zYW1wbGUtZmlsZXMvbWlycm9yLmRpc3Q%3D]
- http://wiki.splitbrain.org/wiki:tips:backup_script [http://www.proxyserve.net/index.php?q=aHR0cDovL3dpa2kuc3BsaXRicmFpbi5vcmcvd2lraTp0aXBzOmJhY2t1cF9zY3JpcHQ%3D]
- 显示源文件
- 登录