

linux backup strategies and techniques

Presented by Jason Tower and Jeremy Portzer

Triangle Linux Users Group

Thursday, May 13th, 2004, 7:00 pm

why backup?

- Because hardware (especially hard drives) fail
- “To err is human”
- Acts of \$DIETY
- But mostly, because shit happens



why use linux?

- Multilingual (speaks lots of protocols)
- Lends itself to scripting and other automated tasks
- Excellent network transparency
- Cheap, reliable, yada yada



two types to tame your troubles

1. System backup (imaging)

- Used for backing up an entire system in one shot
- Ideal for bare metal restores or cloning

2. File backup

- Typically run daily
 - Usually only backs up data and user files, not the entire OS and apps
-
-

attack of the clones

- Most basic form – clone one disk to another
dd if=/dev/hdc of=/dev/hdd (to clone)
dd if=/dev/hdc of=/path/to/file.img (backup to file)
dd if=/path/to/file.img of=/dev/hdc (restore from file)
 - Assumes that hdc is smaller or equal in size to hdd
-
-

*SSH*How me the network

- Use stdin/stdout to run dd thru ssh

```
dd if=/dev/hda | ssh 10.1.1.2 dd of=/dev/hda
```

```
dd if=/dev/hda | ssh 10.1.1.2 dd
```

```
of=/path/to/file.img
```

```
ssh 10.1.1.2 dd if=/path/to/file.img | dd
```

```
of=/dev/hda
```

- You will be prompted for a password unless a key exchange has been set up
- Great for backing up a system before making

g4u is good 4 me 2

- G4U (ghost for unix) is a very easy way to backup a disk or partition across a network
 - Consists of a floppy or bootable CD image with a NetBSD kernel and a few shell utilities
 - Requires a FTP server to store the resulting images
 - Image is gzip'd on the fly to save space
-
-

tips and tricks

- Boot the system being cloned using Knoppix, Tom's root boot, rescue image, etc. Don't try to dd a live, active, mounted filesystem
- Make disk images smaller by zeroing out unused space (on each filesystem) and compressing

```
dd if=/dev/zero of=/home/tmpfile bs=1k
```

- See www.cpqlinux.com/sshcopy.html for lots of examples

file backup

- One flexible and powerful method is to mount directories from other hosts on the network
 - Once a directory or filesystem is mounted it looks and acts just like it were local
 - This “pull” method is generally very safe since the client shares can be set read-only
 - All scripting is run from the backup server, not the clients
-
-

different strokes for different hosts

- For windows hosts, “share” the directory and use smbmount to mount it
smbmount //winhost/c /mnt/smb -o guest
 - For *nix hosts, add directories in /etc/exports and make them available using NFS
mount -t nfs 10.1.1.2:/home /mnt/nfs
 - OS X can use either method
-
-

time to do the heavy lifting

- Once a remote directory is mounted, there are several methods available to actually copy data:

```
cp -av /mnt/smb/* /backup/winhost/
```

```
tar -cvzf /backup/tarball /mnt/smb/
```

```
rsync -av /mnt/smb//backup/winhost/ (note trailing  
/)
```

- Make sure backup device is faster than the

the hype about media type

- Cheap, portable, high capacity – choose any two
 - Tapes are easily stored and hold a lot of data, but they're expensive
 - Optical media (CD-R/DVD-R) are cost effective and small, but capacity is limited
 - Hard drives get cheaper every day and hold a ton, but are essentially fixed onsite
-
-

the array is not the way

*****RAID IS NOT A BACKUP SOLUTION!!!*****

- Helps protect against hardware failure only
- Vulnerable to same human errors and physical disaster as a single disk

*****RAID IS NOT A BACKUP SOLUTION!!!*****

(just in case you weren't listening)

tapes and tarballs

- Only need a few basic commands
 - Most tapes devices are recognized as /dev/st0
 - tar -cvf /dev/st0 /home /etc /var (to backup)
 - tar -xvf /dev/st0 /tmp (to restore)
 - Other basic commands
 - mt -f /dev/st0 rewind (to rewind tape)
 - mt -f /dev/st0 offline (to eject)
-
-

rsync/snapshot overview

- rsync basics
- simple rsync backup script
- snapshot/hardlink basics
- rsback demonstration
- rsync servers
- ssh keys



rsync basics

- rsync at its lowest level is just like 'cp'
 - rsync [-av] <source> <destination>
 - trailing slash difference
- Only copies differences
 - use --delete flag to propagate deletes
- Built-in understanding of networking
 - RSYNC_RSH=ssh or “-e ssh”
 - rsync protocol with ::
 - rsync must be installed on both sides

• -a (all) -v (verbose) -z (compression, use with

basic rsync backup

- `rsync -avz -e ssh --progress`
`/home/trilug/jeremy/`
`portzer.com:/home/jeremy/trilug-backup`
- run this from cron every night, 4:20 am:
`20 4 * * * rsync ...`
(see “man 5 crontab”)
- requires ssh to work without passwords, ssh
keys with keychain is one option

snapshot backups

- Incremental backups (last hour, last day, last week, last month, etc.)
- Only backs up changes each time, yet allows you to see a “snapshot” at each previous interval
- Requires about 2x disk space with rsync/hard-link method
- True snapshots require fancy filesystems
(LVM, or commercial products)

hard link review

- All files in *nix reside in inodes
 - File names are “links” to the inodes
 - You can have multiple links (filenames) referring to the same file, called a “hard link”
 - Note effect of changes in contents, permissions, ownership
 - A file is not deleted until all of its links are removed (rm removes a link, not a file).
-
- Multiple hard links to the same file only works

basic snapshots

- “cp -al” creates links instead of copying
- rsync unlinks before overwriting
 - note that cp needs --remove-destination flag to do this
- Basic Snapshot script:
 - rm -rf backup.3
 - mv backup.2 backup.3
 - mv backup.1 backup.2
 - cp -al backup.0 backup.1
 - rsync -a --delete source/ backup.0/
- Even though these backups are incremental

rsback script

- Manages rsync snapshot backups for you
- Set up sections for hourly, daily, weekly
 - Hourly is every 4 hours on TriLUG setup
- chiana - backup server with large disk, runs rsback script
- moya - home directory and mail server
- Source of backups:
 - Hourly: NFS-mounted partition from moya
 - Daily: last hourly backup (hourly.0)

rsback drawbacks

- Requires additional script to kick it off and run backups in correct order (see <http://www.trilug.org/~jeremy/run-rsback.sh>)
 - rsync sometimes gets confused when files move around during the backups
 - not a true snapshot due to time to perform backup
 - other options; rdiff-backup, rsnapshot
-
- (may require special utility for restoration)

restoration

- Backup directory is exported (read-only) to clients in cluster
- Users can restore files simply by copying them from /backup
- Notes on preserving permissions:
 - rsback is run as root on backup server
 - source directories exported (ro,no_root_squash)
 - backup server has access to LDAP for UID to username mapping (rsync copies by username)

rsync servers

- use of rsync server triggered by double-colon in command line:

```
rsync -av --progress ftp.ibiblio.org::
```

- single-colon uses rsh, or value of RSYNC_RSH, or “-e ssh”
- configuration in /etc/rsyncd.conf
- runs from [x]inetd, default port 873, as “rsync --daemon”

-
- can be configured with host-based and user-

ssh keys

- Allows use of ssh transport for rsync without typing a password
- operates in 'pull' mode so that private keys are on backup server
- 'keychain' application keeps private keys cached (with ssh-agent) to allow connections from cron, or reattaching to existing agents
- very good howto with details:

<http://www-106.ibm.com/developerworks/linux/library/l-keyc2/>

references

- Network File Copy with SSH:
www.cpqlinux.com/sshcopy.html
- Easy Snapshot-Style Backups with rsync :
http://www.mikerubel.org/computers/rsync_snapshots/
- rsback:
<http://www.pollux.fr/franken.de/hjb/rsback/>
- rsback startup script:
<http://www.trilug.org/~jeremy/run-rsback.sh>

thanks!

copyright 2004

Jeremy Portzer and Jason Tower
released under the Creative Commons
Attribution-Sharealike 1.0 license:
<http://creativecommons.org/licenses/by-sa/1.0/>

