RSYNC and Dirvish for Disk-to-Disk Backups

LinuxFest April 17, 2004

Keith Lofstrom, KLIC
keithl@ieee.org www.keithl.com

Agenda

- Why back up, anyway?
- Alternatives
- How RSYNC works
- DIRVISH-RSYNC and alternatives
- Conclusions
- References - Ask questions

Keith's First Law of Backups

If you don't have TWO copies of your data, you will have ZERO copies

This is a restatement of Gump's law:

Keith's First Law of Backups

If you don't have TWO copies of your data, you will have ZERO copies

This is a restatement of Gump's law:

Shit Happens

Why Do Backups?

- To recover lost data!
- Backup data nightly (or more often)
- Backups should be easy
- Backups should be indexed
- Backups should be secure
- Restore should be as simple as possible

Why Is Data Lost?

- User mistakes - deletion or overwrite
- Failed programs
- Imprudent upgrades
- Hard drive failure
- Other hardware failure
- Enemy action

Backup Alternatives

- DVD-R: 8x-R $180, $1.70/4.7GB
- Tape: Exabyte VXA-2 $1300, $80 /80GB
- RAID: Controller $300, 3 + 1 parity drive
- RSYNC: 2x250GB drive, 2 hotswap cages, offline spare tray $350

Quality of Backup

	DVD-R	TAPE	RAID	**RSYNC**
Mistakes	🙂	🙂	☹	🙂
Failed Programs	🙂	😐	☹	😐
Bad Upgrades	🙂	🙂	☹	🙂
Drive Failure	😐	😑	🙂	😐
Hardware Failure	🙂	🙂	☹	😐
Enemy Action	🙂	😐	☹	😑

Quality of Backup

	DVD-R	TAPE	RAID	**RSYNC**
Mistakes	😀	😀	~~🙁~~	😀
Failed Programs	😀	🙂	~~🙁~~	🙂
Bad Upgrades	😀	😀	~~🙁~~	😀
Drive Failure	🙂	😐	~~😀~~	🙂
Hardware Failure	😀	😀	~~🙁~~	🙂
Enemy Action	😀	🙂	~~🙁~~	😐

Quality of Backup

	DVD-R	TAPE	**RSYNC**
Verifiability	🙁	☹️	😀
Indexing	🙁	☹️	😀
Single File Restore	😐	☹️	🙂
System Restore	😐	😐	😀
Backup Effort	☹️	🙁	😀
Fragility	😐	🙂	😐
Lev0 Backup Time	3 hr+	2 hr	1 hr
Drive/Contrl. Cost	180	1300	50
Media Cost $/GB	0.35	1.00	0.02

Disk to Disk Backup with RSYNC

- Disks are the cheapest, fastest backup media available

Western Digital
120GB
ULTRA DMA/100
HARD DRIVE
- Retail Boxed Hard Drive (Installation Kit Included)
- 7200 RPM
- 8.9ms Seek Time

8MB BUFFER!

Limit 1 Per Customer

$\$98^{99}$ -40 = $\$58^{99}$
In-Store Price · Mail-in Rebate · Price After Rebate

#3306472

Maxtor
250GB
ULTRA DMA/133
HARD DRIVE
- 7200 RPM
- Retail Boxed Hard Drive
- Installation Kit Included

8MB BUFFER!

Limit 1 Per Customer

$\$174^{99}$ -30 = $\$144^{99}$
In-Store Price · Mail-in Rebate · Price After Rebate

#3700356

- RSYNC copies file systems over networks

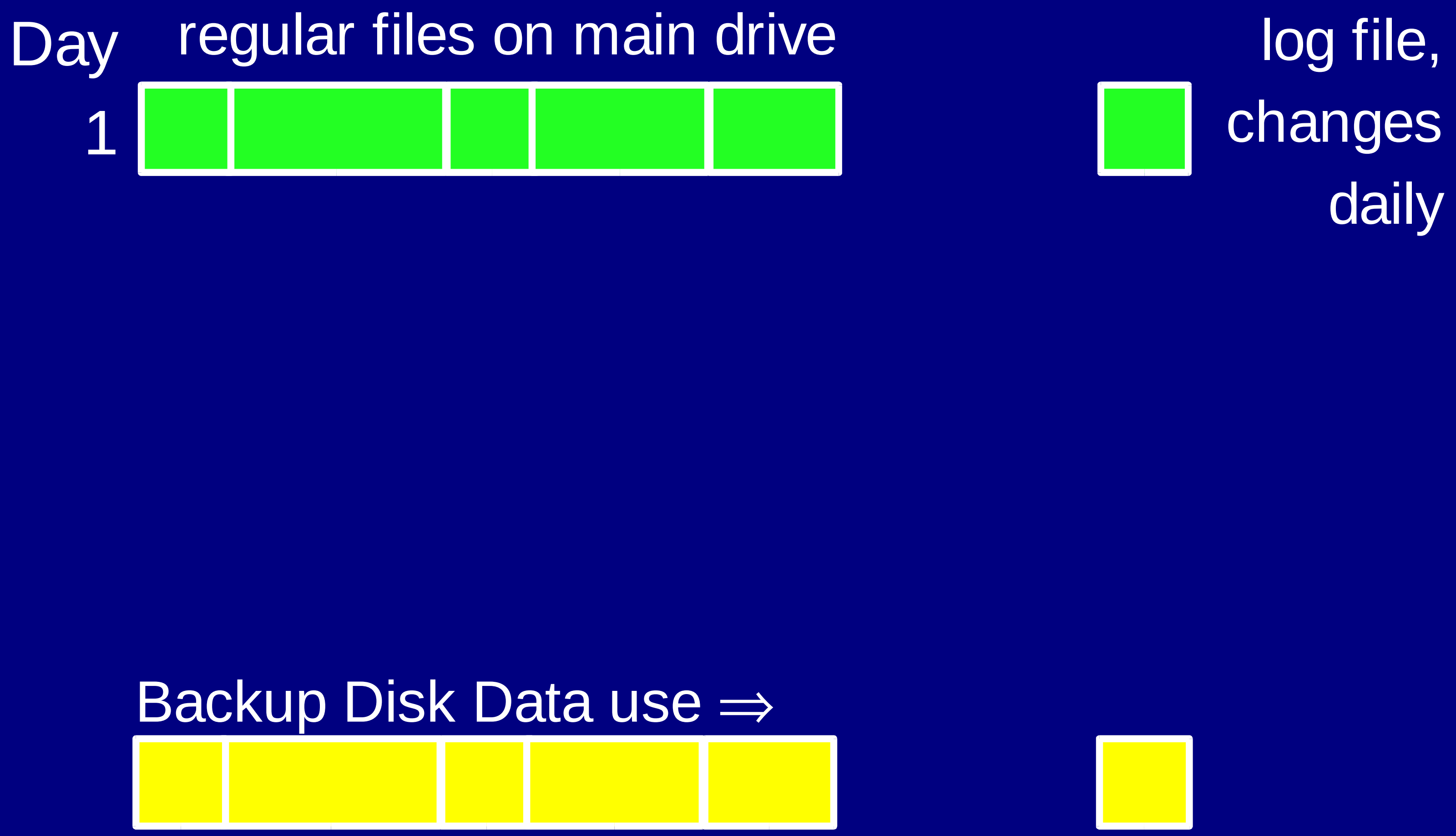
The RSYNC Protocol

- SAMBA team
- Fast - only moves changed files
- Cheap - uses hard links for unchanged files
- Builds a duplicate of client
- File system agnostic; stores data, not image
- Clients ported to many operating systems
  - Linux, Unix, Windows, MAC 9 and 10

RSYNC is *FAST*

- Compares modification times - moves only changed files
- Blocks files into segments - moves only segments with changed hashes
- Network load aware
- Moves files with ssh
- Typically 30-80 minutes to back up 80GB

RSYNC uses Linux Hardlinks



Day

1

regular files on main drive

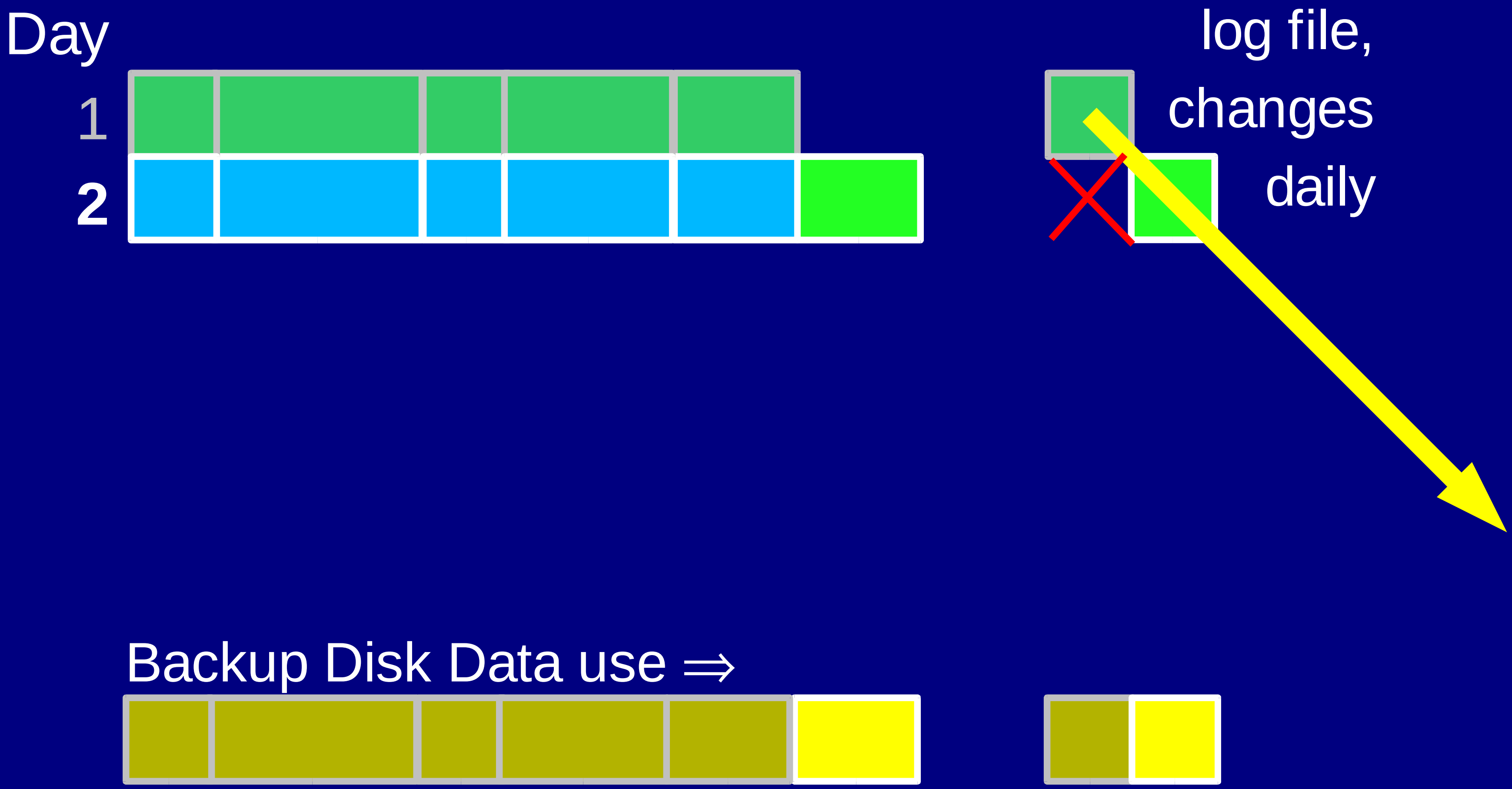
log file,

changes

daily

Backup Disk Data use ⇒

Backups - Day 2



Day

1

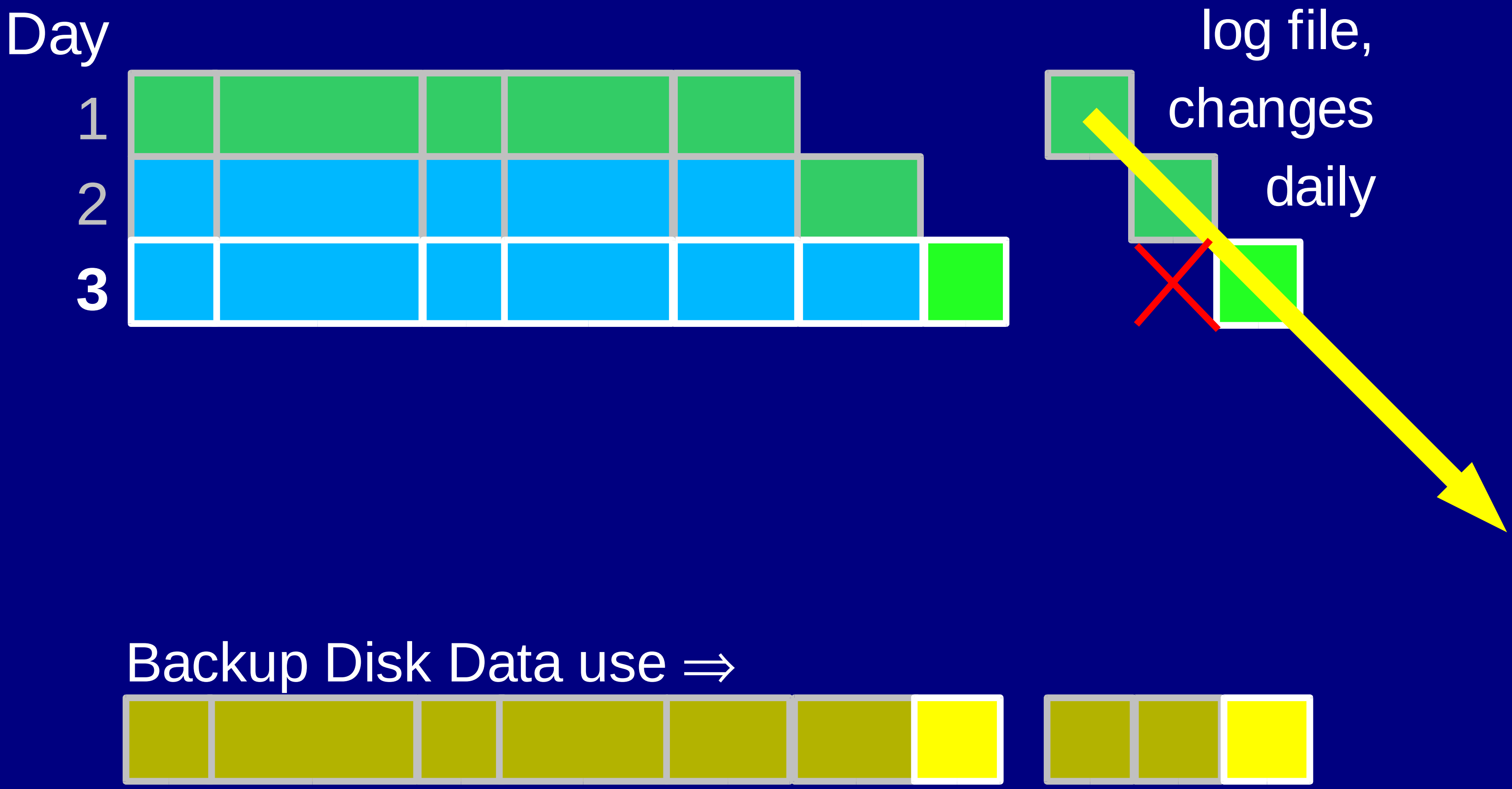
2

log file,
changes
daily

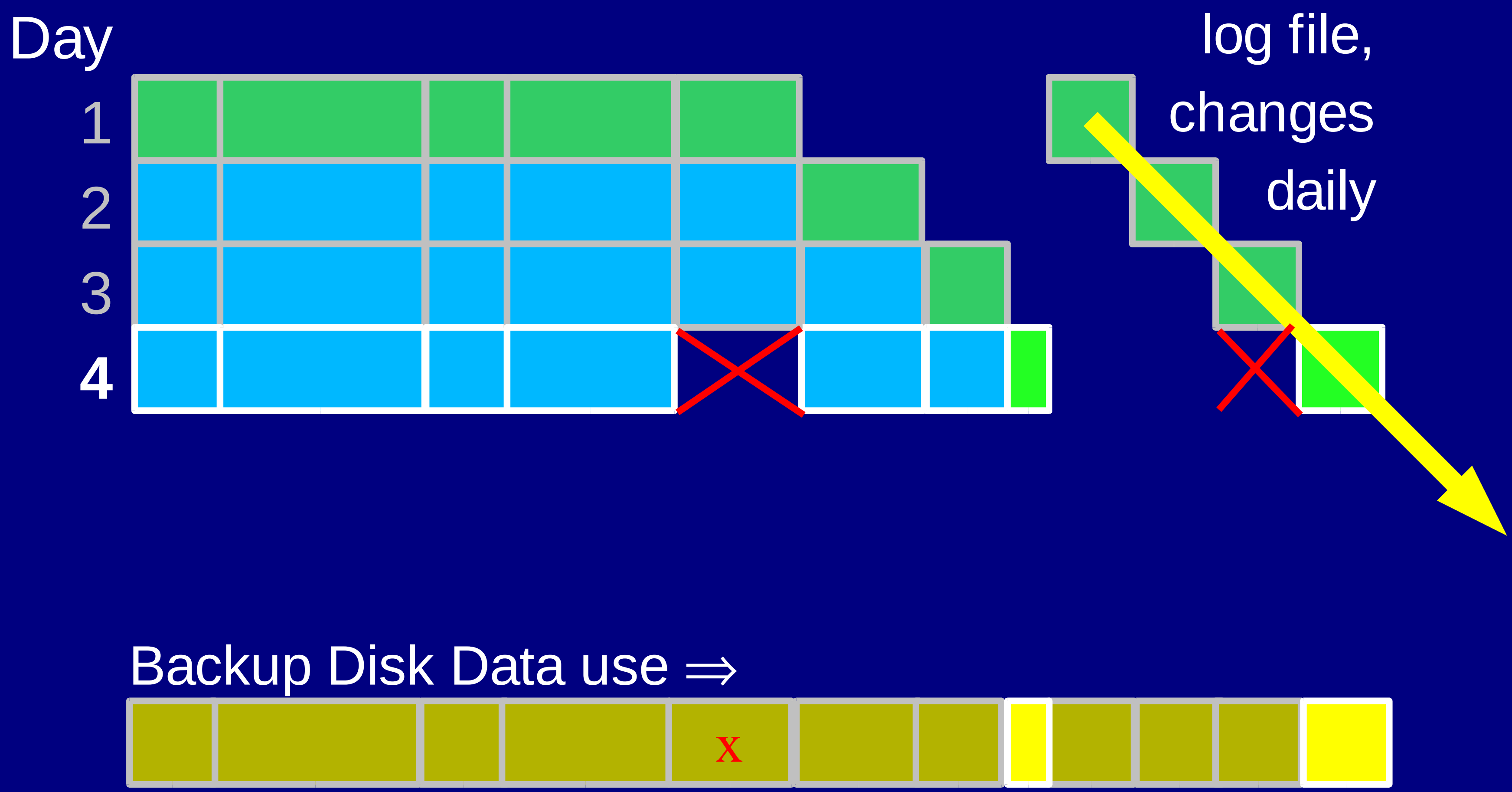
Backup Disk Data use ⇒

Adds another directory tree, but only new or changed data!

Backups - Day 3

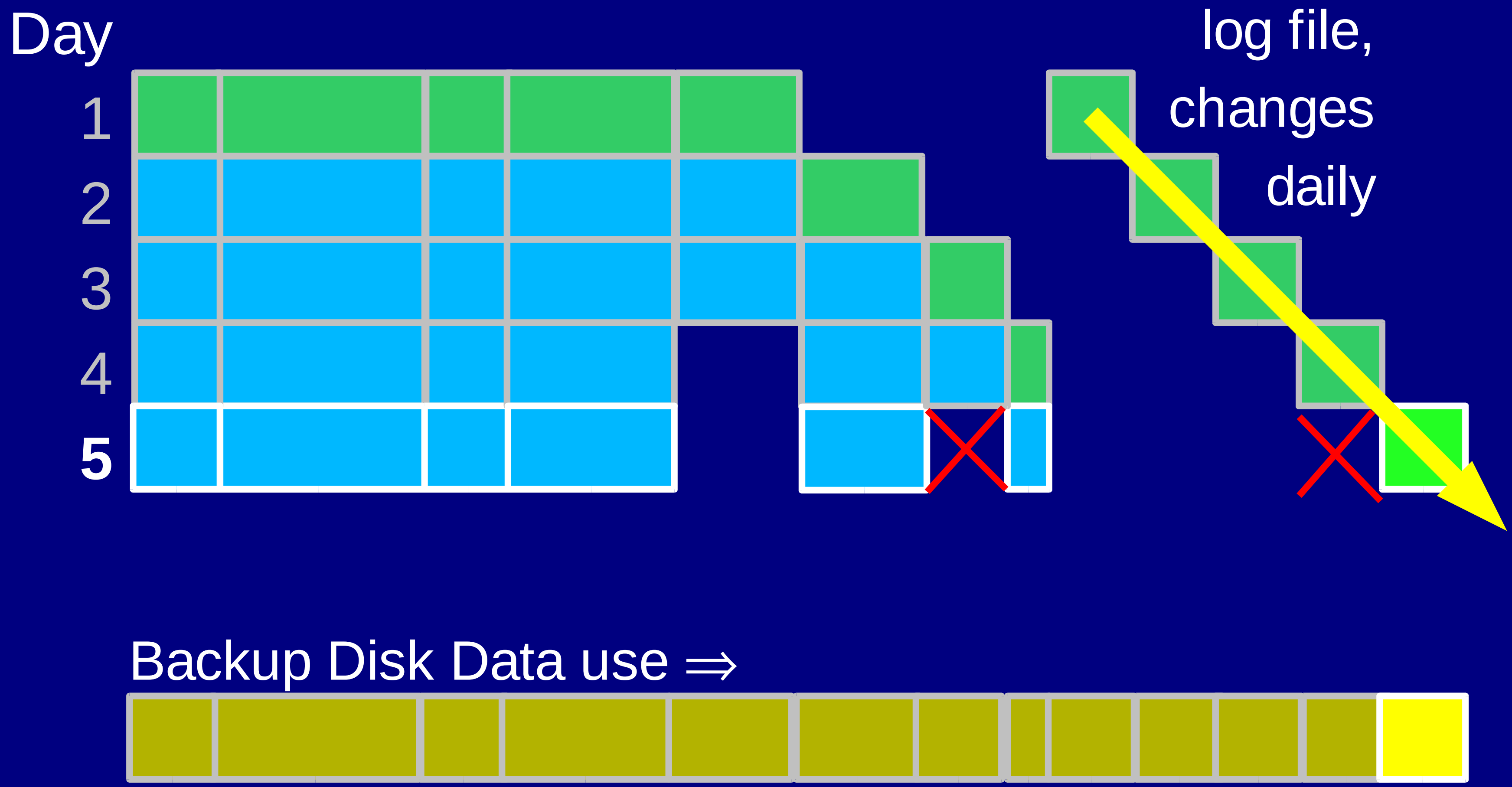


Backups - Day 4

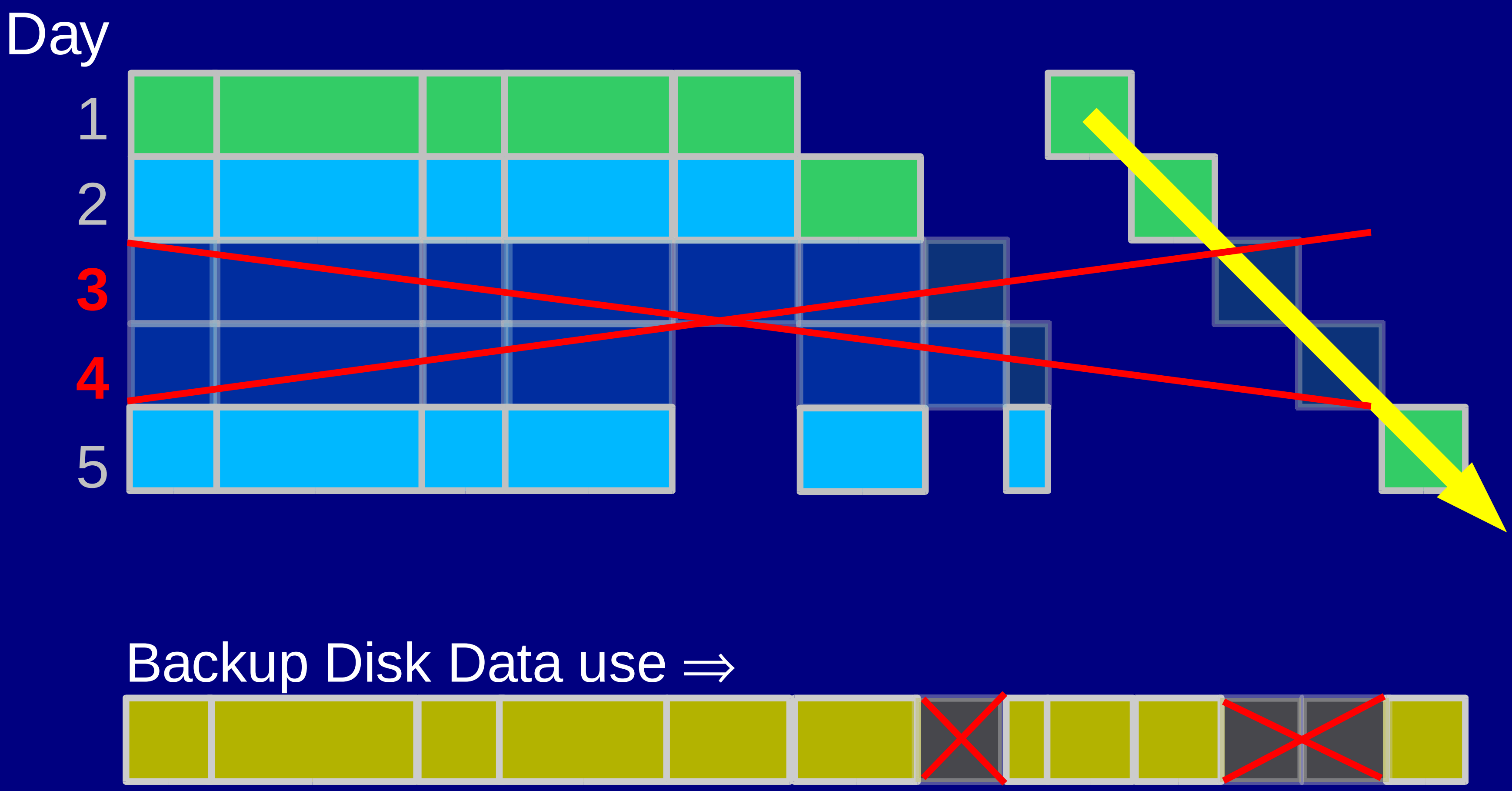


Deleted file remains in image

Backups - Day 5



Expire - Day 3 & 4



Two directory trees and some file space recovered

RSYNC Backup Disk Usage

- Backup disk usage accumulates daily
  - New files and directories
  - Changed files and directories
  - Growing log files
  - Mail spools
- Backup disk space can be reduced by expiring old images

Example System - KLIC Network

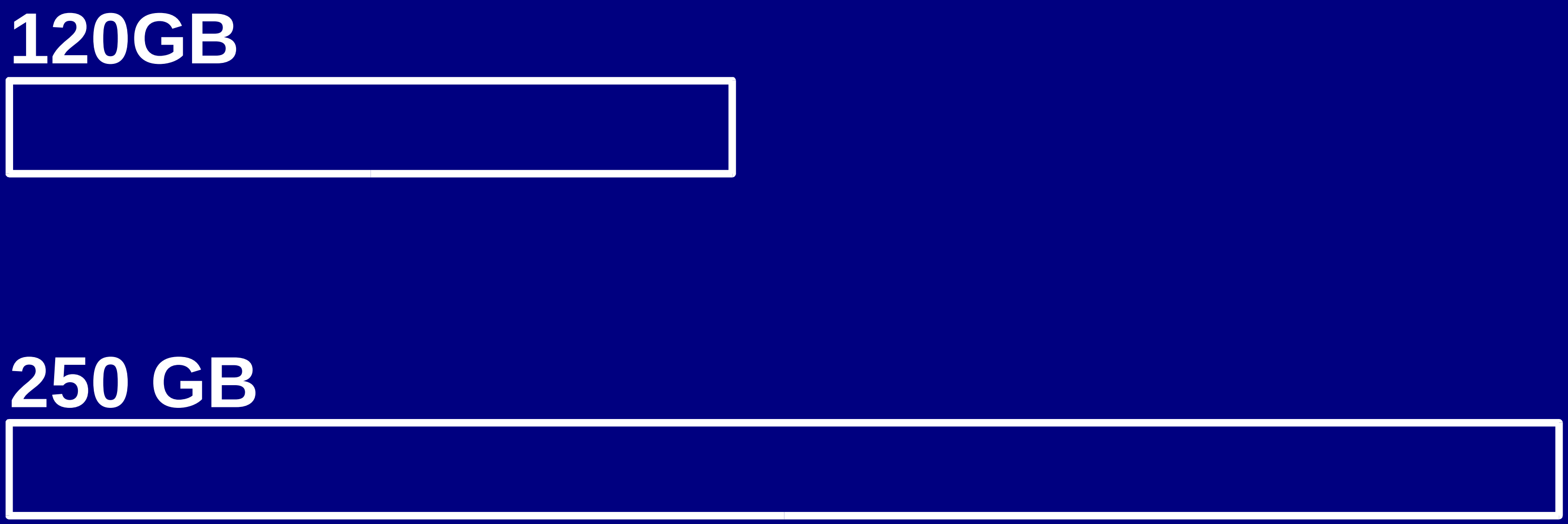
- Four networked machines
  - Linux Server
  - Linux Firewall
  - 2 Linux Laptops
- Data changes slowly
- 80 GB total

KLIC - RSYNC Backup Disk Usage

- 80 GB initial image + 4GB extra
- KLIC averages 600MB/day new files
- Excluded ISO image directory

Big Backup Drives are Better

- 80GB + 4GB + 0.6GB × days

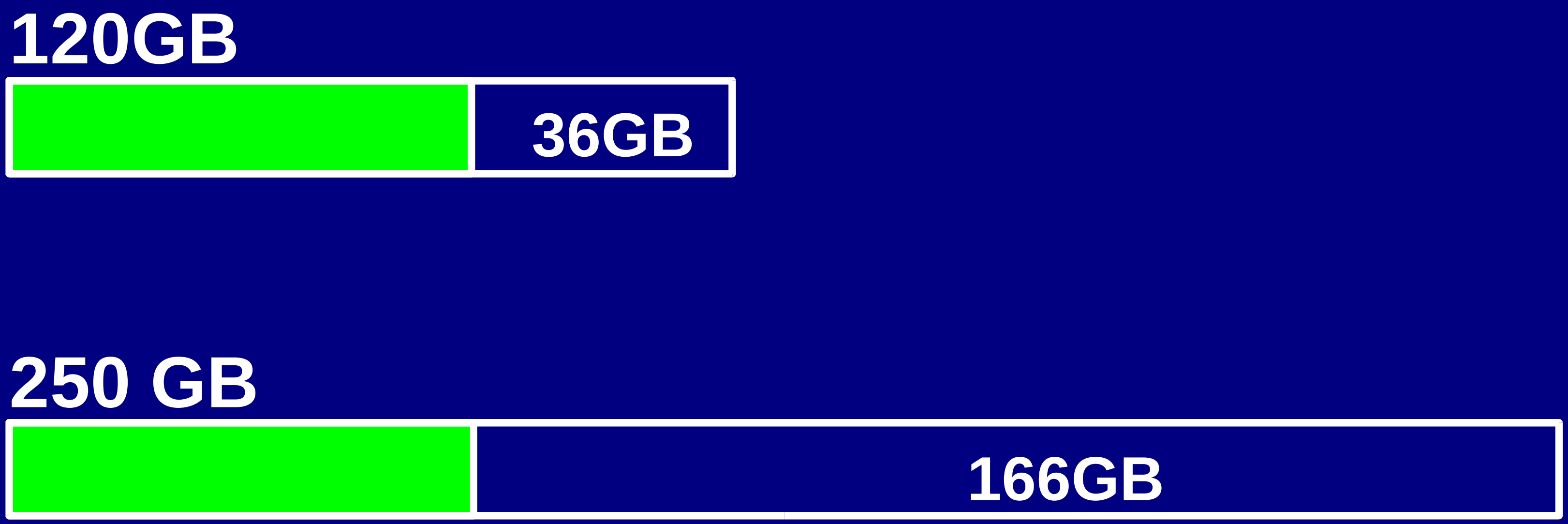


120GB

250 GB

Big Backup Drives are Better

- 80GB + 4GB + 0.6GB ×days



120GB

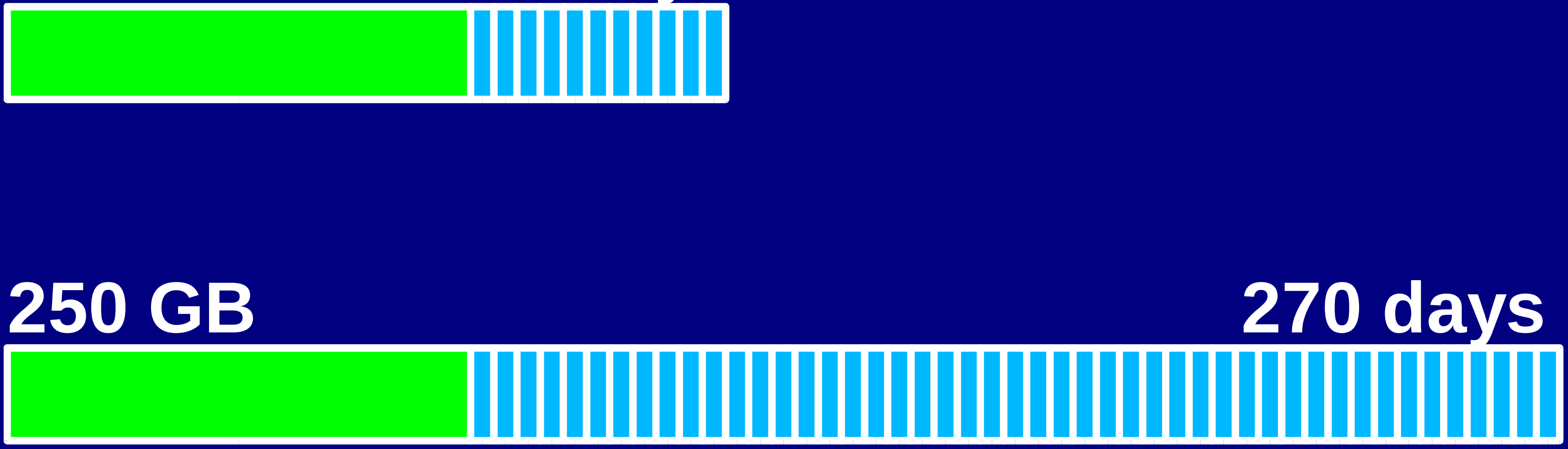
36GB

250 GB

166GB

Big Backup Drives are Better

- 80GB + 4GB + 0.6GB ×days



120GB 60 days

250 GB 270 days

Big Drives - LBA48

- Older Controllers are LBA28
  - Maximum Drive Size 137GigaBytes (137E12)
- Big Drives are LBA48
  - Maximum Drive Size 144PetaBytes (144E18)
    - 10Kyears of video / genomes of all species on Earth
- LBA48, ATA-133, ATA-6 controllers
  - PROMISE Ultra133 TX2 (PDC 20269)

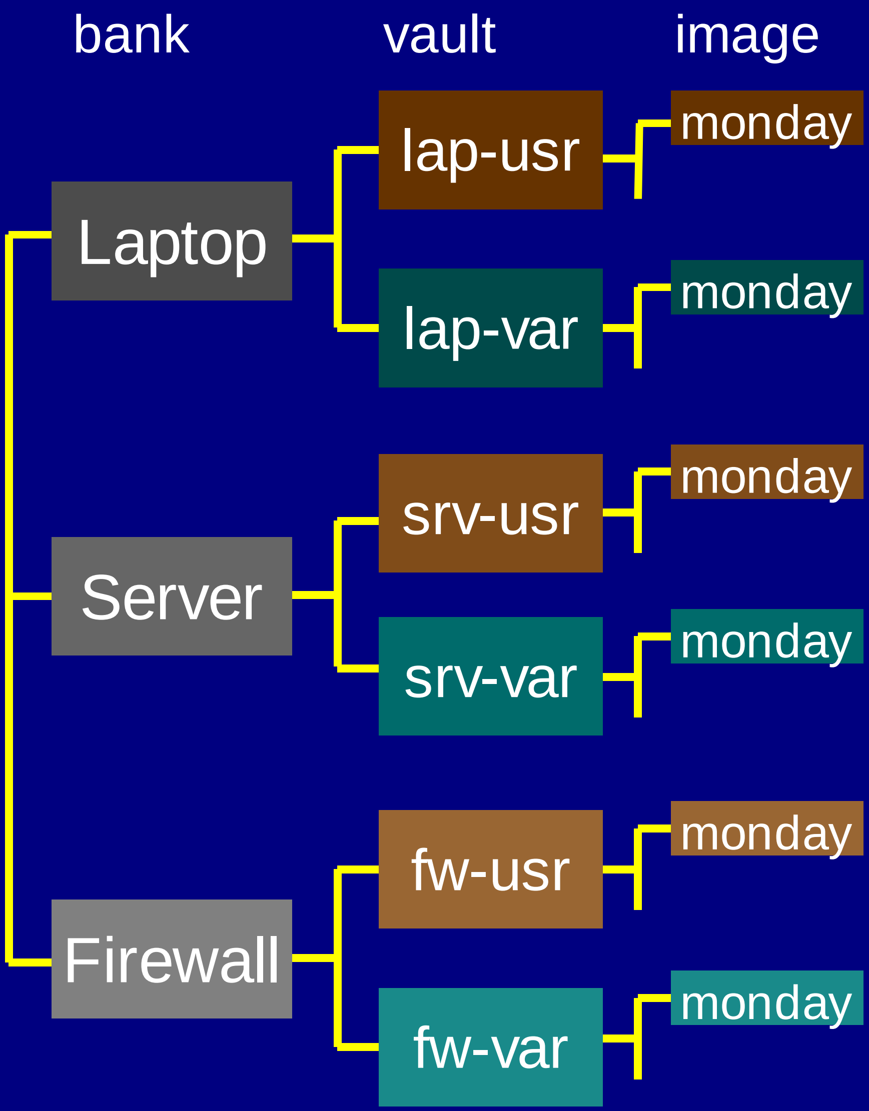
RSYNC - Gotchas

- Initialization takes hours
- RSYNC really thrashes hardware
  - read/writes a lot of data rapidly
  - fills RAM - other apps swap in slowly afterwards
  - finds media and driver bugs

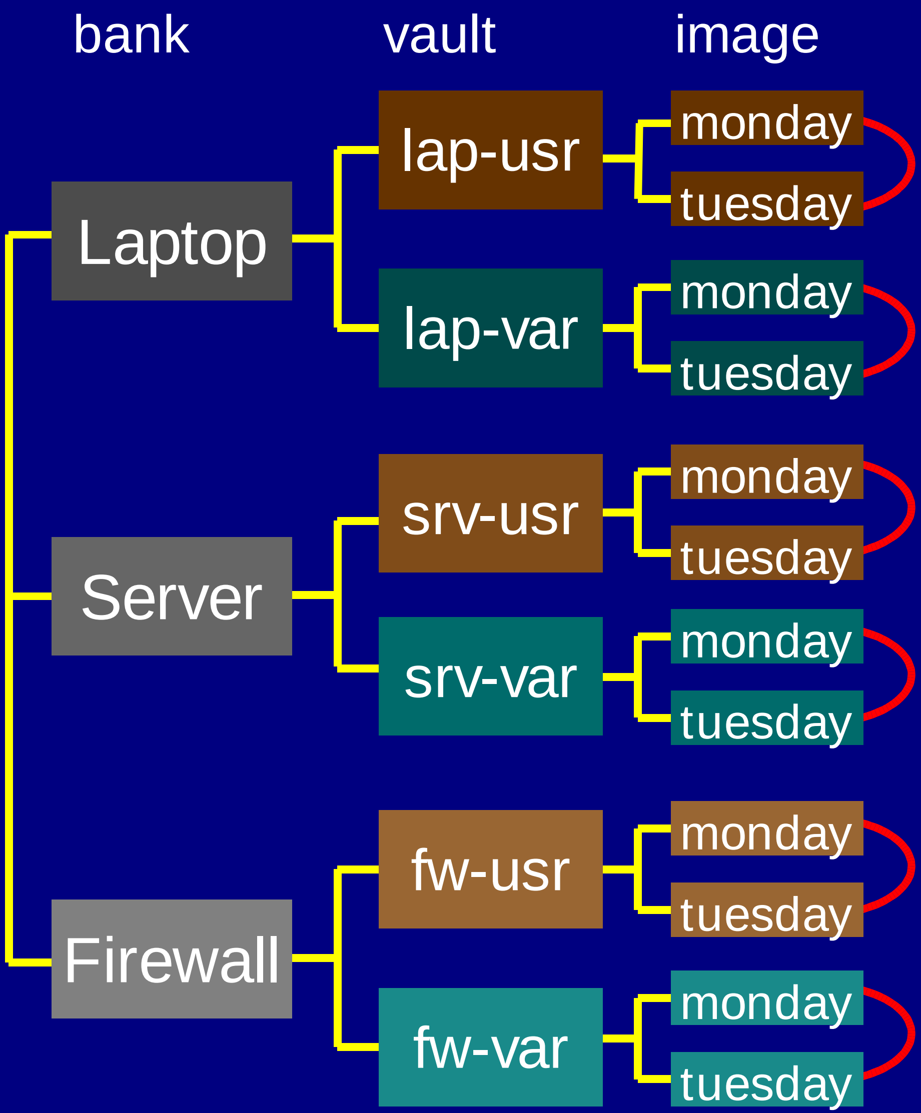
RSYNC - Dirvish

- PERL wrapper by J.W. Schultz
- Automates RSYNC from config files
- Driven from server, pulls files
- Model and documentation "challenging"
- Flexible behavior and configuration
- Design for the structure of your file systems
- Adapt your file systems for best backup

Dirvish File Structure



Dirvish File Structure



Dirvish - File Structure

/backup/laptop/	machine - "bank"
/backup/laptop/lap-usr/	filesystem - "vault"
/backup/laptop/lap-usr/2003-1215/	first image
/backup/laptop/lap-usr/2003-1215/tree/	directory of files
/backup/laptop/lap-usr/2003-1215/logs/	
/backup/laptop/lap-usr/2003-1216/	second image
/backup/laptop/lap-usr/2003-1216/tree/	directory of files hardlinked to 1216
/backup/laptop/lap-usr/2003-1216/logs/	

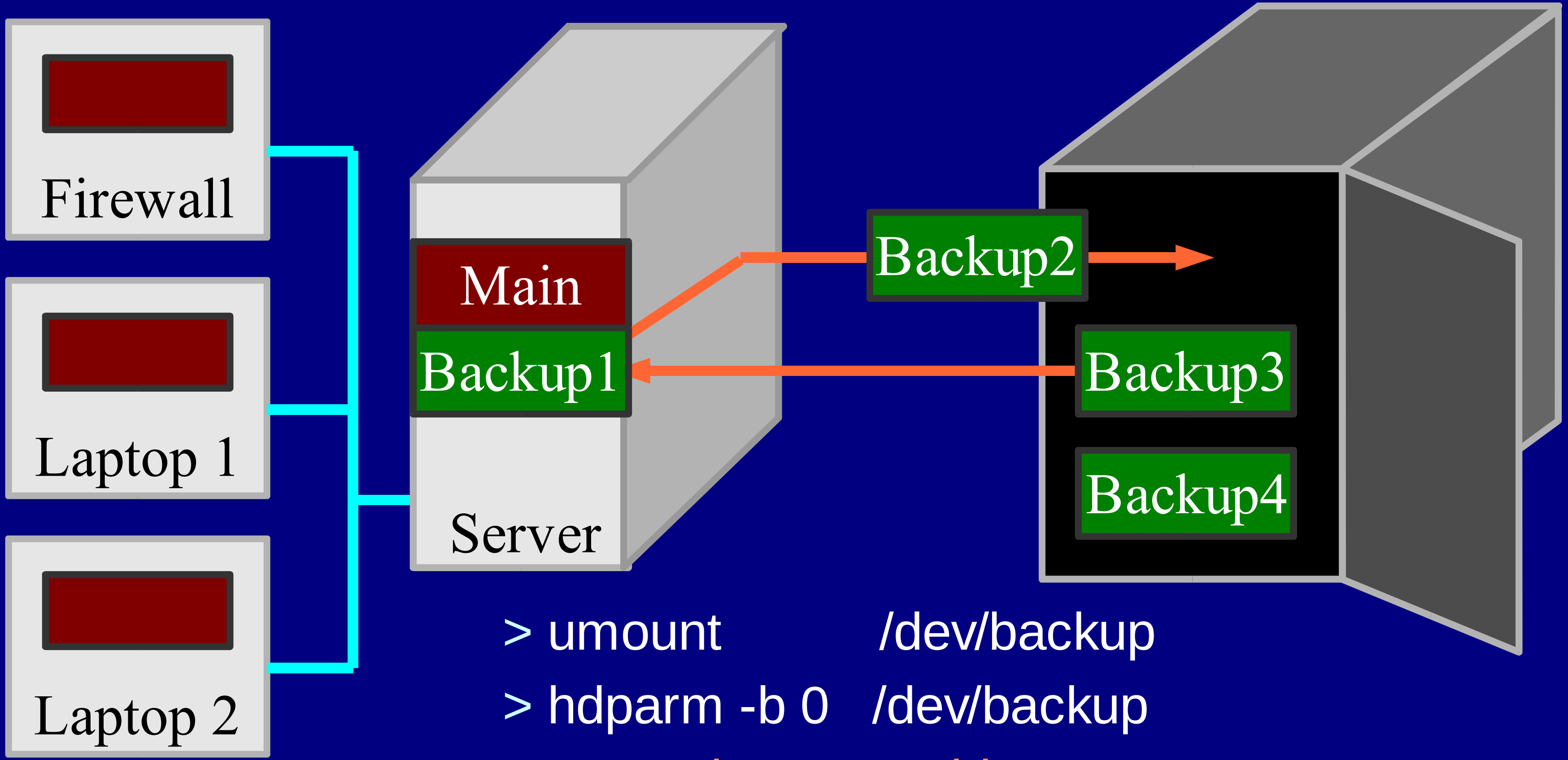
Alternatives to Dirvish

- rsnapshot (Mike Rubel's rsync_snapshot)
  - Driven from clients
- rdiff-backup
  - Stores diffs of the data, good for log files
- Simple drive mirroring with RSYNC
  - Overwrites old data every day, no history
- Compressed TAR or DUMP to hard disk
  - Slow, cannot use hard link layering

Another Alternative: BackupPC

- Web GUI
- Easy, user driven single file restore
- Efficient disk space use
  - finds same data under different names
- Good for large systems of similar computers
- Not good for drive swapping

Swapping Backup Drives



```
> umount          /dev/backup
> hdparm -b 0   /dev/backup
        - - hot swap drives - -
> hdparm -zb 1 /dev/backup
> mount           /dev/backup  /backup
```

IDE works with 2.4.22 series kernels, not 2.6.x yet - Alan Cox says 8/04?

ViPower Swap Cages & Trays

- Slide switch, not key
- $16 mail order
- Extra tray $10?
- USB2 version available
  - works with 2.6.x

- Alternative: InClose @ Fry's
  - USB2 has errors
  - may be kernel error

Alternative USB2 external drive

- + Separate power supply - more robust
- + USB2 hotswap is fast and easy
- - Slower
- - More expensive
- - USB2 + LBA48 (>137GB) hard to find
  - Most external cases are still LBA28

Padded Bag for Drive Transport

RSYNC - Tricks

- Build backup drives with boot & swap
  - bootable system + swap on 4GB or so
- Use few, big partitions for your systems
  - No need to accomodate small media anymore!
- Backup drive unmounted, bus turned off
  - safe from program fails
  - won't stop savvy invader

RSYNC - More Tricks

- ln -s   /dev/hdXN   /dev/backup
- Build with backup partition with *mkfs -m 1*
- *slocate:* Exclude /backup partition
  - otherwise, all backup images in slocatedb
- Save *sfdisk* partition info with backup data
- Save *df* output with backup data
  - helps with rebuild/restore decisions

Eliminate big, slow changing files

- Use MAILDIR mail directory format
  - small separate, non-changing files rather than one big file per folder
- Use SUSE-style *logrotate* & *dateext*
  - /var/log: dated extensions, *not* numbered. Whole set does not change daily. somelog.20040219, somelog.20040218, etc., *not* somelog.1, somelog.2

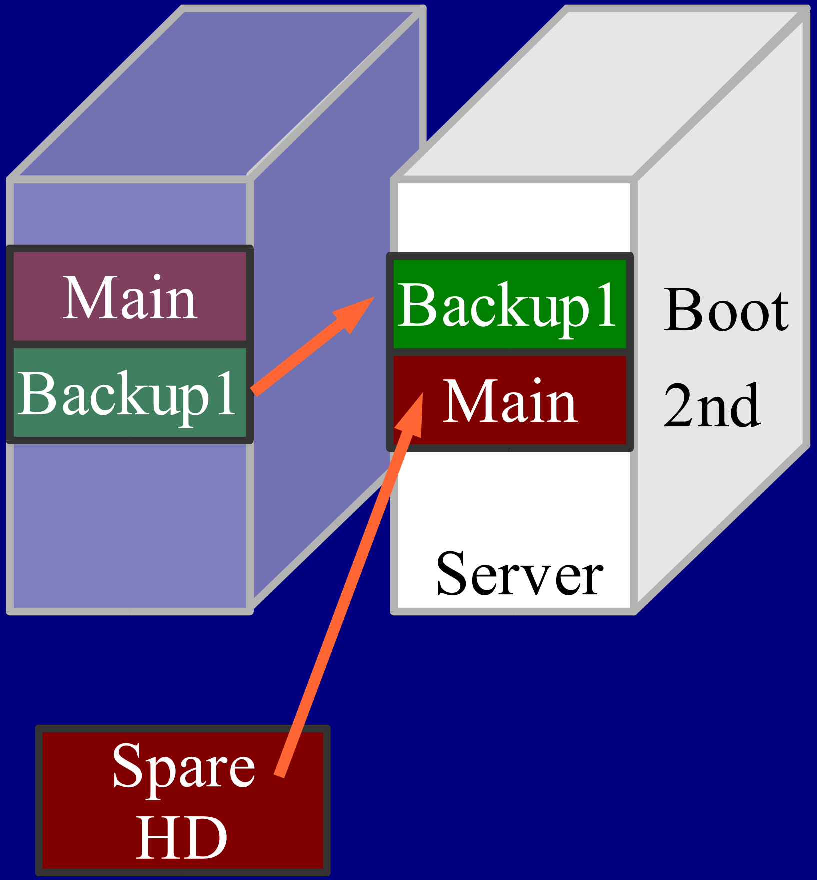
Bare Metal Restore

- Build shell script to do bare metal restore
  - **and TEST IT!**
- saved *sfdisk* output can partition new drive
  - text format can be edited for changing drive size
- Automatic process is lifesaver during a very stressful time. Do your future self a favor!

Bare Metal Restore

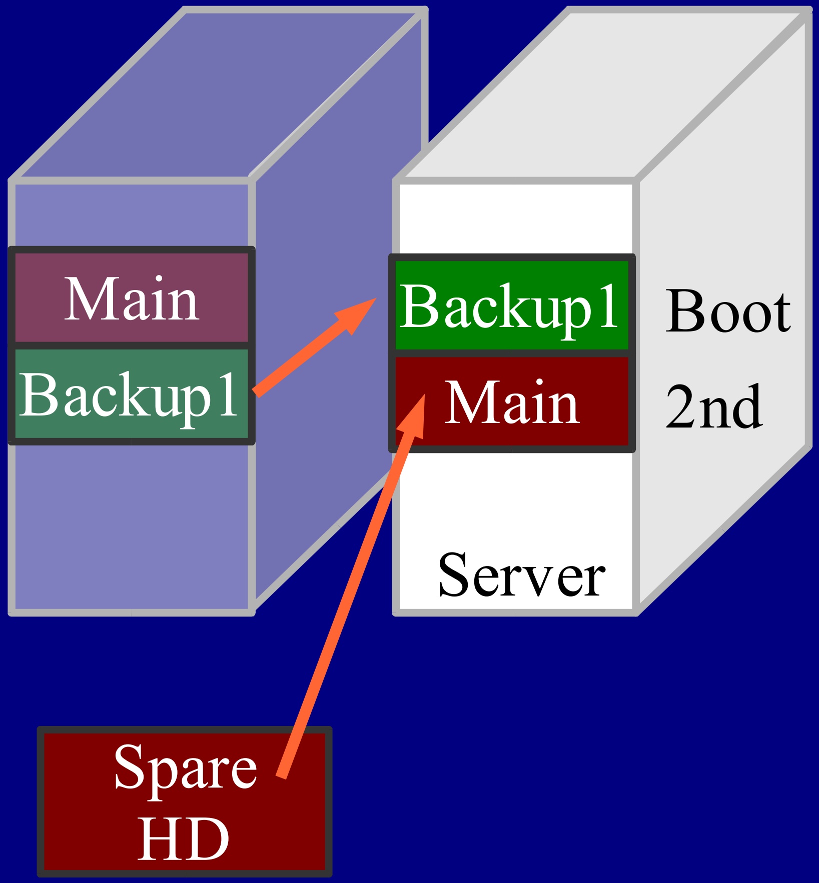
- Keep main server drive in swap cage, too
- Have replacement drives available
- Have a spare tray to mount laptop drive
  - with IDE mini adapter
- Restoring 50GB to a bare drive
  - Takes about 5 minutes of setup
  - Takes about 2 hours of runtime

Restoring Server



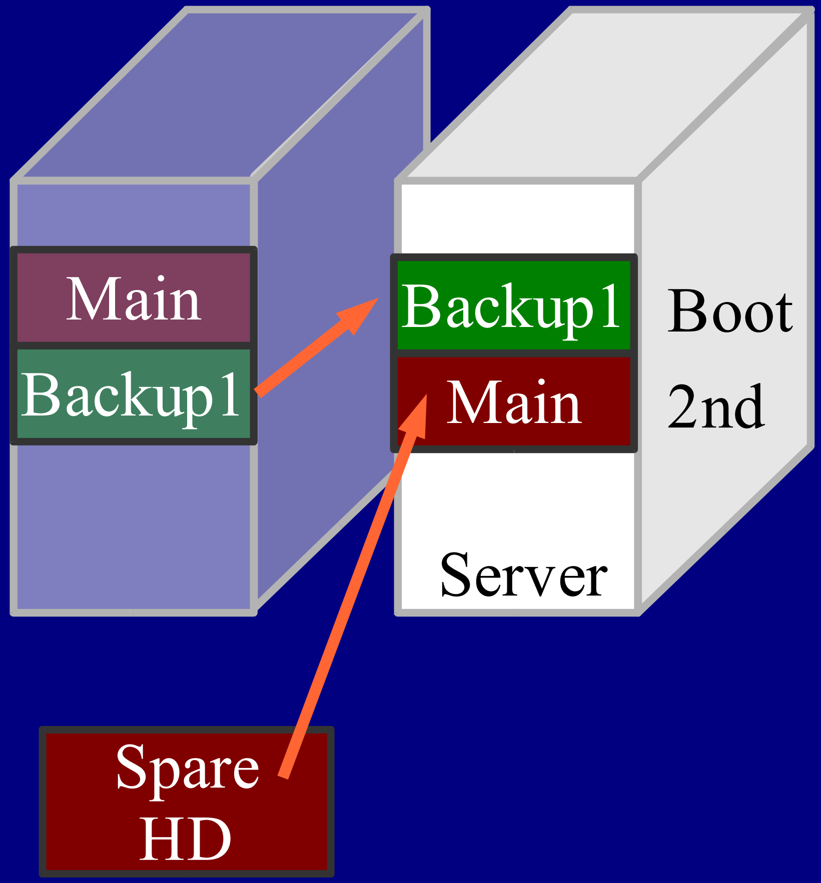
- Power down
- Swap drives
- Reboot from backup drive
- Modify and run restore script
  - (Select image to use)

Restoring Server



- Power down
- Swap drives
- Reboot from backup drive
- Modify and run restore script
- Go to movie, escape angry users
  - (this may take 2 hours)

Restoring Server



- Power down
- Swap drives
- Reboot from backup drive
- Modify and run restore script
- Go to movie, escape angry users
- Restore portions of other images
- Power down, swap drives
- Reboot from new main drive

Cost of RSYNC

- 2x250 GB drives cost $300
- 2 drive trays cost $16
- 2 drives fill in 9 months (no expire)
- ⇒ $1.20/day
- ⇒ $0.015 /GB
- Expires and excludes can reduce cost
- Retire the drives after they fill (archival)

Contractors and Consultants

- Your contract may specify the *removal* of client data at the end of the job.
- Impossible to remove from tape or DVD-R!
- Using RSYNC imaged disk backups, a script can *find* and *remove* client directories and email from the backup hard drives, *leaving the rest intact*.

Life after RSYNC

- Backups and restores are easy,
  - **so you can take more risks!**
- New distro?   Why not? Easy to go back
- Enemy Action? Rebuild fast!
  - yesterday's image or a *combination* of images
- Watch newspaper ads for hard drive sales
  - Fry's "new"drives are often manufacturer *refurbs* - "no defect found" - check SMART data

What's next?

- Better documentation for DIRVISH
- MD5 checksum for rsync files
  - protect backup drive from enemy action
- Automated restore script generation
- Debug USB2 kernel error
- Backup in a box

Conclusions

- RSYNC and Dirvish make drive-to-drive backups automatic and easy
- Inexpensive, fast, robust
- Backs up from server over network
- Opportunities for simple improvements

REFERENCES

RSYNC	www.rsync.org
Dirvish	www.pegasys.ws/dirvish
ViPower cages	www.vipower.com
InClose cages, bag	www.sanmax.com
this talk, more info	www.keithl.com/linuxbackup.html
www.aracnet.com/~seniorr/plug-advanced-topics-2003-12-17.pdf	
BackupPC	backuppc.sourceforge.net
rsnapshot	rsnapshot.org

/usr/local/sbin/dirvish-daily page 1

```
#!/bin/bash
#/usr/local/sbin/dirvish-daily
# mount disks and run dirvish
# KHL  October 30, 2003
#
# this is called by /etc/cron.daily/backup
#--------------------------------------------------------------
# Variables used.   Note that if BACKUPTOUCH is changed, you
# will also need to change it in /usr/local/sbin/dirvish-post

PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/sbin
BACKUPDRIVELOG=/var/log/backup_drivelog
BACKUPTOUCH=/tmp/backuptouch
DISKLABEL=/backup/DISKLABEL
DIRVISHRUNALL=/usr/local/sbin/dirvish-runall

#----------------
# Mount the backup drive.  /dev/backup is a symbolic link made
# by the sysadmin to the actual drive used for backups

/bin/mount /dev/backup /backup
```

D250 WD2500 |

/usr/local/sbin/dirvish-daily   page 2

```
#-----------------
# Make a "touched system" directory for backups used.
# Each dirvish pass (in dirvish-post) will touch a filename
# corresponding to the machine that was successfully backed up.
# This allows us to keep track of which machines were online
# when a particular backup was made.
/bin/mkdir  $BACKUPTOUCH

# Do the backup itself.  dirvish-runall is jw's perl script,
# which uses the config file in /etc/dirvish
$DIRVISHRUNALL

# Make string with machines actually backed up indicated
TOUCH=`/bin/ls -w 1000 -C $BACKUPTOUCH `

# Make string with percentage full

FULL=`/bin/df /dev/backup | /usr/bin/tail -1 | \
      /bin/awk "{ printf \"%3s%14d\",\\$5,\\$4 }"`
```

/usr/local/sbin/dirvish-daily   page 3

```
# Log the backup drive, add a single line with backup drive used,
# the date, and which systems got backed up this time.  This will
# be useful in locating backup drives for restore.
/bin/echo `/bin/cat $DISKLABEL` `/bin/date +"%a %b %d %T %Z %Y"` "|" \
          $TOUCH "|" $FULL >> $BACKUPDRIVELOG

# Remove touched files from /tmp/backuptouch
/bin/rm -rf $BACKUPTOUCH

# Unmount backups for security.  Prevent exposing backup partition
# to a rogue program.
/bin/umount /dev/backup

# All done!
exit 0
```

/var/log/backup_drivelog

```
C250 6Y250P | Sun Feb 22 05:10:02 PST 2004 | fw gate life t30 | 42% 135831060
B200 6Y200P | Mon Feb 23 05:29:26 PST 2004 | fw gate life t30 | 88% 24167540
D250 WD2500 | Tue Feb 24 05:15:09 PST 2004 | | 33% 155679048
...
D250 WD2500 | Fri Feb 27 05:10:36 PST 2004 | fw gate life t30 | 34% 153715288
D250 WD2500 | Sat Feb 28 05:10:53 PST 2004 | fw gate life t30 | 34% 153102004
```

/usr/local/sbin/dirvish-post   page 1

```
#!/bin/bash
# /usr/local/sbin/dirvish-post
#
# KHL  02-23-2004added "success" test
# KHL  11-13-2003original
#
# This is run after dirvish completes.   It assumes Linux clients at
# the far end of the pipe, and will need to be modified for other OS
# types, specifically so it can save disk partition and disk full
#  information.  It may be easier to find "df" and "sfdisk" for those
# OS types and keep them in same the executables directories.
#---------------------------------------------------------------------
# Client commands.  Full paths given for security.

SFDISK='/sbin/sfdisk -d /dev/hdmain '
DF='/bin/df '

# Server commands.  Full paths given for security.
SSH='/usr/bin/ssh'
```

/usr/local/sbin/dirvish-post   page 2

```
# variables
# DIRVISH_CLIENT, _SERVER, _DEST, _STATUS provided from dirvish

BACKUPTOUCH=/tmp/backuptouch

# Write df files to backup directory (image level) to keep track
# of disk usage,  in case we need to rebuild a disk.
# This writes the df info into each vault image, which is redundant
# but necessary given that dirvish is configured for multiple
# vaults per client.

$SSH $DIRVISH_CLIENT $DF > $DIRVISH_DEST/../df.$i

# Touch a marker file that indicates that the client has been visited

if [ "$DIRVISH_STATUS" = "success" ]; then
   /bin/touch  $BACKUPTOUCH/$DIRVISH_CLIENT
fi

# All done!
exit 0
```

/backup/dirvish/server/spare/dirvish/default.conf

```
client: server
tree: /spare
xdev: true
index: gzip
image-default: %Y-%m%d-%H%M
exclude:
    /proc
    /spare/iso
    /iso
```

/etc/dirvish/master.conf

```
bank:
    /backup/dirvish/server
    /backup/dirvish/laptop
    /backup/dirvish/fw
exclude:
    lost+found/
         proc/
    core
Runall:
    srvhome            03:00
    srvspare           03:00
    srvopt             03:00
    srvroot            03:00
    srvusr             03:00
    srvvar             03:00
    srvvarlog          03:00
    srvvarspool        03:00
    srvtmp             03:00
    laproot            03:00
    lapboot            03:00
    lapusr             03:00
    lapvar             03:00
    laphome            03:00
    lapopt             03:00
    lapspare           03:00
    fwroot             03:00
    fwtmp              03:00
    fwusr              03:00
    fwvar              03:00

expire-default:   never

# keep the sundays forever, the
# dailies for 3 months

expire-rule:
#   MIN HR  DOM MON DOW STRFTIME_FMT
    *   *   *   *   *   +3 months
    *   *   *   *   1   never

pre-server: /usr/local/sbin/dirvish-pre
post-server: /usr/local/sbin/dirvish-post
```

recover.hdg page 1

```
#! /bin/bash

BDIR=2003-1110-0300
S=/backup/dirvish/server
T=/new
DISK=/dev/hdg
SFD=$S/sfdisk.

MKFS='/sbin/mkfs.ext3 '
MOUNT=/bin/mount
ECHO=/bin/echo

COPY='/usr/bin/rsync -a'
E='/'

# time it

/bin/date
/bin/sleep 10
/bin/date > tmp/recoverlog
```

```
# first, build disk partitions
# from sfdisk file

/bin/cat $SFD | /
   /bin/sfdisk --force $DISK

# second, build partitions

$MKFS  ${DISK}1
$MKFS  ${DISK}5
$MKFS  ${DISK}6
$MKFS  ${DISK}7
$MKFS  ${DISK}8
$MKFS  ${DISK}9
$MKFS  ${DISK}10
$MKFS  ${DISK}11
$MKFS  ${DISK}12
$MKFS  ${DISK}13
$MKFS  ${DISK}14

#  make the swap partition

/sbin/mkswap     ${DISK}15
```

recover.hdg   page 2

```
# THIS IS VERY PARTITION DEPENDENT

$MOUNT       ${DISK}1                  $T
$COPY        $S/root/$BDIR/tree$E      $T

$ECHO "now copying usr"
$MOUNT       ${DISK}5                  $T/usr
$COPY        $S/usr/$BDIR/tree$E       $T/usr

$ECHO "now copying var"
$MOUNT       ${DISK}6                  $T/var
$COPY        $S/var/$BDIR/tree$E       $T/var

$ECHO "now copying var/log"
$MOUNT       ${DISK}7                  $T/var/log
$COPY        $S/varlog/$BDIR/tree$E    $T/var/log


#             ... more partitions, not shown
```

recover.hdg page 3

```
# fourth, make /proc

/bin/mkdir  $T/proc

# fifth, install grub boot loader

/sbin/grub --batch --device-map=/dev/null << EOF
device (hd1) ${DISK}
root (hd1,0)
setup (hd1)
quit
EOF

# time it again

/bin/date
/bin/date  >> /tmp/recoverlog

# finish up

exit 0
```