PXE/Thin-Client Deployments

Dustin King
Student Computing, Stanford University

How We Use Them

- Imaging environment
- Print release stations
- Library kiosks (in development)

Motivations

- Robustness
- Ease of management and troubleshooting
  - Server-side configuration
  - Current image at every boot
- Out-of-warranty hardware
  - No dying hard drives

The Basics

- Linux
- Diskless workstations
  - Applications run locally
  - Avoid traditional thin-client A/V issues
  - Some tolerance for server downtime
- No reliance on local drives
- Configuration controlled on server
- Boot using network resources

Preparing the Machines

- Enable PXE
- Set network booting to highest priority
- Lock boot options
- Disable SATA controllers (optional)
- Remove hard drives (optional)

PXE

- Clients find server using DHCP options
- Configuration files on server specify kernel, root file system, other boot parameters
- Configuration files can be applied to hardware addresses, ip ranges
- Kernel, initramfs downloaded via TFTP

Early Userspace

- Custom initramfs
- Run rebuilt ipconfig
- Mount NFS directory containing disk image
- Loopback mount disk image and chroot

Early Userspace init

```
/bin/ipconfig eth0
/bin/mkdir /newroot
while [ !-e /newroot/image ];do
    if [ "${nfsroot%%,*}" = "${nfsroot}" ];then
        /bin/nfsmount ${nfsroot} /newroot
    else
        /bin/nfsmount -o ${nfsroot#*,} ${nfsroot%%,*} /newroot
    fi
done
/bin/losetup /dev/loop0 /newroot/image
/bin/mount -t ext2 /dev/loop0 /newroot/mnt
/bin/mount -t tmpfs -o size="2M" none /newroot/mnt/dev/
/bin/mknod /newroot/mnt/dev/console c 5 1
/bin/mknod /newroot/mnt/dev/null c 1 3

exec /bin/chroot /newroot/mnt /sbin/init </dev/console >/dev/console 2>&1
```

Pretend /is Writable

- Create appropriate tmpfs,unionfs mounts
  - /etc, /home, /tmp, /var
- Remove rw /remount,fsck,file cleanup from init scripts

/etc/rc.sysinit

...

```
/bin/mount -n -t tmpfs tmpfs /unionfs/etc -o size=2M
/bin/mount -n -t unionfs unionfs /etc -o dirs=/unionfs/etc=rw:/etc=ro
echo "tmpfs /unionfs/etc tmpfs rw,size=2M 0 0" >> /etc/mtab
echo "unionfs /etc unionfs rw,dirs=/unionfs/etc=rw:/etc=ro 0 0" >> /etc/mtab
/bin/mount -n -t tmpfs tmpfs /unionfs/var -o size=8M
/bin/mount -n -t unionfs unionfs /var -o dirs=/unionfs/var=rw:/var=ro
echo "tmpfs /unionfs/var tmpfs rw,size=8M 0 0" >> /etc/mtab
echo "unionfs /var unionfs rw,dirs=/unionfs/var=rw:/var=ro 0 0" >> /etc/mtab
/bin/mount -n -t tmpfs tmpfs /tmp -o size=8M
echo "tmpfs /tmp tmpfs rw,size=8M 0 0" >> /etc/mtab
```

Implementations

- Imaging environment
- Print release stations
- Library kiosks (in development)

Imaging (Genie)

- Bash scripts manage process
- Imaging wrapper started by init scripts
- List of tasks and image locations provided by server-side script
- Automated image installation leads to reboot before login available
- Manual usage possible for image creation, troubleshooting

imaging_wrapper

```
#!/bin/bash
mkdir /tmp/imaging
cd /tmp/imaging
wget http://our.imaging.server/genie/tasks
export local_dev=/dev/sda
export image_target=/dev/sda1
export img_stor_root=/mnt/img_stor
export image_dir=/mnt/img_stor/imagecache
eval $(sed 's/^/export /;s/$/;/' tasks)

...
```

imaging_wrapper (cont)

```
if [ "$partition" = "true" ];then
    /root/partition_drive
fi
mount -t vfat /dev/sda2 "$img_stor_root"
if [ "$download" = "true" ];then
    /root/download_image
fi
if [ "$install" = "true" ];then
    echo Installing image.This will take a while...
    /root/install_image
fi
/root/check_in
if [ "$reboot" = "true" ];then
    /sbin/reboot
fi
```

install_image

```
#!/bin/bash
cat $image_dir/*.gz.* | gzip -dc | dd of=$image_target bs=4k &
while [ "$(ps aux | grep sda[1])" ];do ps aux | grep sda[1] | awk '{print "kill
    -s SIGUSR1 "$2}' | bash;sleep 20;done
start=$(printf "%08x" $(fdisk -ul $local_dev | grep "$image_target" | awk
    '{print $3}'))
printf "%b" "\x${start:6:2}\x${start:4:2}\x${start:2:2}\x${start:0:2}" | dd of=
    $image_target bs=1 count=4 seek=28
dd if=/root/mbr of=$local_dev bs=446 count=1
```

Print Release Stations

- Custom inittab
  - Only one virtual console
  - Display manager is a shell script that cleans home directory, starts X
- Locked-down X
  - Restarts when Firefox exits
  - R-kiosk Firefox plugin
  - Ratpoison window manager
  - Web application for print job release

/usr/sbin/kioskdm

```
#!/bin/sh
/bin/umount /home/kiosk
/bin/umount /unionfs/home
/bin/mount -t tmpfs tmpfs /unionfs/home -o size=32M
/bin/mount -t unionfs unionfs /home/kiosk -o
      dirs=/unionfs/home=rw:/home/kiosk=ro
/bin/su --command="/usr/bin/xinit" kiosk
```

/home/kiosk/.xinitrc

```
#!/bin/sh
find /home/kiosk -name sessionstore.js -exec rm {} \;
/usr/bin/ratpoison &
exec /usr/bin/firefox -fullscreen
```

Print Release Stations

- Custom inittab
- Locked-down X
- ACPI configured to kill Firefox (restart X) when power button pressed

/etc/acpi

./events/power:

```
event=button/power.*
action=/etc/acpi/resetx.sh
```

./resetx.sh:

```
#!/bin/sh
killall firefox
```

Library Kiosk Features

- Kerberos authentication
- Printing
- Saving files to USB devices
- Java applets
- Firefox plugins
- Audio
- Viewing PDFs, videos, Office documents...

Particularly Helpful Sites

- http://www.gentoo-wiki.info/HOWTO_Read-only_root_filesystem
- http://www.eng.uwaterloo.ca/twiki/bin/view/Linux/LinuxKiosk

Acknowlegments

- Surajit Bose
- Paul Nuyujukian
- Ian Comfort

Thanks. Questions?

For further information, contact

daking@ stanford.edu

Fill out evaluations at:

http://www.resnetsymposium.org/rspm/evaluation/

STANFORD stucomp