AFS as root filesystem for Clusters
(or maybe anything else)

Troy Benjegerdes
Scalable Computing Lab
DOE Ames Laboratory
<troy@scl.ameslab.gov>
What's my real job..

• High performance computing research
  – middleware research (schedulers, libraries, etc)
  – Network performance evaluation (NetPIPE)
• Often means changing 1 line on N nodes
• single system image? doesn't scale.
• Replicating node images? takes too long
  – whoops, node 25 was down last time, it's out of sync
a couple of years ago..

• Bring up 64 nodes to see if the hardware works
  – I wonder if an NFS server would survive
  – Modify Debian to deal with read-only filesystem
  – Used to use NFS-Root for embedded development
• Wow.. it actually works
  – booting all 64 nodes hit the ldap server more than NFS (yes, this was a crufty openldap version)
    • FYI, 64 nodes * 16 ldap connections per node == dead ldap server due to 1024 file descriptors
Today...

- running in production.. until the NFS server hiccups.
- images are annoying to manage (1 node is allowed to mount it r/w)
- tools like lessdisks and oscar make this trivial to set up
- upgrading libc... um, might as well just reboot everything.
AFS > NFS

• Good:
  – Nice volume management
  – replicated volumes
  – failover

• Not-so-good
  – only afs admins can change UID's, or set suid bit
  – ACL's
  – debian libc package uses hardlinks
  – getting kernel module in the initrd
Hack the fileserver

Troy Benjegerdes <troy@scl.ameslab.gov>, <hozer@hozed.org>
suid and hardlinks: you lose

- SUID bits: It was a hack then, it's a hack now
  - Just seems like a bad idea in afs.
  - there is no good answer.. maybe allow it with auditing?
- Hardlinks
  - libc wants to save space with timezone files
  - Maybe we can just support posix ACL's and allow hardlinks again?
Debian initramfs-tools package

- http://source.scl.ameslab.gov/hg/mkinitramfs-openafs
- /etc/mkinitramfs/hooks/openafs
  - copy afsd, config files, etc to initrd
- /etc/mkinitramfs/scripts/openafs
  - start afsd and mount /afs in the right places
- /etc/mkinitramfs/scripts/openafs-premount/openafs
  - Optionally mount a disk cache partition (otherwise use memcache)
Things to try..

- Use the linux kernel kafs client for read-only filesystem, openafs for read-write
- cache pinning?
  - .... hrrm, where's that disconnected code...
Stupid afs tricks 101

• I wonder if I can make my laptop afs-root
  – (shameless plug) http://kurobox.com/
    • embedded PPC with IDE disk.. makes a real nice portable Debian machine..
    • apt-get install openafs-fileserver
    • mknitramfs -o /boot/initrd.img-2.6.16-1-powerpc-afs
    • yaboot.conf:
      image=/boot/vmlinux-2.6.16-1-powerpc
      label=afstest
      root=/afs/kbox.hozed.org/nodeimg/hozer.ppc
      initrd=/boot/initrd.img-2.6.16-1-powerpc-afs
      append="cachesize=64000 boot=openafs"
It um, mostly works

- This works better for a cluster node
  - running 10 IBM Power5 systems this way
  - Handy for updating software.. chroot /afs/....../nodeimg/ppc64.test from a desktop G5, build software.

- X works, seems to boot in about the same time.
- Openoffice blows up in a strange way
- laptop sleep daemons don't work to well when the network and fs go away.

Troy Benjegerdes <troy@scl.ameslab.gov>, <hozer@hozed.org>
Thanks

- DOE MICS ($$)
- OpenAFS
  - (Derrick & Russ for debian package)
- Debian & Ubuntu initramfs maintainers

Troy Benjegerdes <troy@scl.ameslab.gov>, <hozer@hozed.org>
Questions?