Building and Maintaining a FooBSD

May 14, 2010

John Baldwin
jhb@FreeBSD.org

What is a FooBSD

- Customized version of FreeBSD*
- Targeted towards clusters of x86 servers (PXE, etc.)
- Typically used for private extensions, hacks, early MFCs, etc.

Main Topics

- Managing the Source Tree
- Automated Installs
- Handling Upgrades

Managing the Source Tree

- Push changes upstream when possible!
- Use source code control with FooBSD as a branch of FreeBSD
  - CVS + patches
  - CVS → p4
  - SVN mirror via SVK
  - Other

Staying Current

- Consider tracking stable branches rather than releases
- Merge early and often!
  - Merge conflicts easier to handle in small batches
  - 7.3-FOO-20100514
- Change log

Building Releases

- Use FreeBSD's existing `make release` build process
  - Easy to get up and running quickly
  - Is not tied to sysinstall
- Devise a custom build process
  - More work from scratch

Automated Installs

- Use an `install.cfg` script to automate sysinstall
  - Not very flexible out of the box
  - Can leverage sysinstall's ability to manage different media, etc.
- Build your own install environment
  - Can be very flexible
  - Requires a bit more work to setup

Making `sysinstall` More Flexible

- Generate install config files at runtime during install
- Use shell scripts to generate config files that are subsequently executed
- Add more tools to sysinstall's MFS root (dialog, kenv)
- Pass variables via loader.conf settings (media or per-colo settings)

`sysinstall` Example I

install.cfg:

```
# Figure out the disk configuration
command=/bin/sh /stand/dodisk.sh
system
configFile=/stand/disk.cfg
loadConfig
```

sysinstall Example II

dodisk.sh:

```
disk=`kenv -q install.disk`

# Generate the config for the disk
cat > /stand/disk.cfg <<EOF
disk=${disk}
partition=all
bootManager=standard
diskPartitionEditor
…
EOF
```

Upgrading Existing Installs

- Reinstall the machine
- Install a new world over NFS
  - Slow, lots of NFS I/O to run make
- Would be nice to replace with a simple tarball extraction
  - Would have to handle edge cases like ld-elf.so.1.old
- freebsd-upgrade

Updating `/etc` I

- `mergemaster`
  - No 3-way merge
  - Updates in place
  - Hard to automate
- `etcmerge`
  - 3-way merge
  - Updates in a separate tree

Updating `/etc` II

- `etcupdate`
  - 3-way merge
  - Updates in place
  - Best effort first pass

Conclusion

- Many other ways and tools as well
- Other related issues (packages)

Q&A

- http://www.FreeBSD.org/~jhb/stand
  - `domedia.sh` picks install media and network interface
  - `dodisk.sh` picks disk
- http://www.FreeBSD.org/~jhb/etcupdate
- Questions?