ELCE-E/T-DOSE 2011

Booting userspace in less than 1 second

Defining boot time

- from power up to usable state?

Angstrom running on beagleboard

- Reasonably fast platform - 1GHz Cortex A8
- Distro optimized for the platform
- I work at TI
- I'm part of the beagleboard.org team

Cheating

- switching to pandaboard (dual 1GHz Cortex A9) after noticing boot is CPU bound after doing all the work explained later on

Boot up sections

- ROM code based loader
- First stage bootloader (MLO)
- Second stage bootloader (u-boot.bin)
- Kernel
- Base userspace (system session)
- Applications (user session)

The current situation

- ROM code based loader - instantanious
- First stage bootloader (MLO) - 0.1 - 0.5s
- Second stage bootloader (u-boot.bin) - 12s
- Kernel - 7s
- Base userspace (system session) - 120s
- Applications (user session) - 20s

Solutions

- bootloader/kernel startup optimizations are known and presented to death at each ELC(-E)
- User session optimization has people working on it as well
- But those require significant code changes

I lack coding skills!

Do what you do best

- Attack the part that is the slowest
- Attack the part that only involves shell
- Try the solution written by rockstars

systemd!

- not systemD
- not system d
- certainly not système d

systemd

- replacement for sysvinit
- from the ground up new design
- emphasis on parallellism

why systemd?

- website listed rockstar developers I’ve met in person (Lennart, Gustavo)
- website listed an embedded consulting firm for optional support
- At the time scheduled to be standard in Fedora, F15 and F16 have it standard

why systemd? - 2

- No shell scripts!
- unit structure design resembles bitbake recipes
- cross compiles cleanly
- autotools build system
- source managed in git at freedesktop.org

First steps

- watch all videos of systemd presentations
- read nearly all blog posts about systemd
- join #systemd on freenode

First steps - 2

- Try existing recipe
- Get serial console working
- Boot a small rootfs
- Get timing info

As slow as sysVinit :(

Digging deeper

- use builtin units for udev, dbus, avahi, etc
- steal units for e.g. dropbear from Arch linux
- mask sysV duplicates with different names (dbus vs dbus-1)
- complain a lot in the #systemd channel

from ±60 seconds to ±18 seconds

- show http://dominion.thruhere.net/koen/angstrom/systemd/before-udev-cleanup.svg

Still not good enough

- time (udevadm trigger ; udevadm settle)
- always 180 seconds
- same amount of time as time-out after accept4() call

Toolchain problem

- sys_accept4() is added for ARM in 2.6.36
- (e)glibc built against 2.6.32 headers
- beagle running 2.6.32, panda 2.6.35
- backport, cherry-pick, update, rebuild

Toolchain solution

- (e)glibc rebuilt against 2.6.37 headers
- beagle running 2.6.32 + accept() patch, panda 2.6.35 + accept4() patch

down to ±3 seconds

- show http://dominion.thruhere.net/koen/angstrom/systemd/after-udev-cleanup.svg

Making it even faster

- remove unused udev rules
- remove unused units (vconsole, mounts)
- switch to a faster machine

Down to 1.1 seconds

- Show http://dominion.thruhere.net/koen/angstrom/systemd/omap4430-panda-201110190919.svg

Trying to make it < 1s

- Since we're at 1.1s, why not try going subsecond?
- CPUfreq governor -> performance
- Faster SD card
- more complaining in #systemd

1.032 seconds, drat!

```
Startup finished in 3702ms (kernel) + 1032ms (userspace) =4734ms
```

Consulting the labjournal

- hmmm, I patched uboot for the ttyS2 -> ttyO2 change
- console=tty1 quiet

0.825 seconds!!!!!

```
Startup finished in 1607ms (kernel) + 825ms (userspace) = 2432ms
```

- show http://dominion.thruhere.net/koen/angstrom/systemd/omap4430-panda-201110221738.svg

Saving 3 more seconds

- disable bootdelay in uboot

The current situation

- ROM code based loader - instantanious
- First stage bootloader (MLO) - 0.1 - 0.5s
- Second stage bootloader (u-boot.bin) - 9s
- Kernel - 1.5s
- Base userspace (system session) - 0.825s
- Applications (user session) - 20s

- show http://dominion.thruhere.net/koen/angstrom/systemd/omap4430-panda-201110221738.svg

Further steps

- investigate LZO or even uncompressed kernels
- make u-boot d-cache safe
- ditch GDM and use slim/simpleDM/nodm
- ditch GNOME 2.x and use EFL

Resources

- http://www.angstrom-distribution.org/
- http://freedesktop.org/wiki/Software/systemd
- http://www.google.com
- http://dominion.thruhere.net/koen/angstrom/systemd/