Why does an architecture bootstrap take 5 years?

ARM

Wookey `<wookey@linaro.org>`
SSG/Linaro

GEC

May 2016

Who am I

- Free Software developer since 1990
- Arm Linux developer since 1999
- Debian developer since 2000

Some things I had something to do with:
Survex, PsiLinux, ArmLinux book, Emdebian, bootfloppies, Therion, apt-cross, dpkg-cross, Debian cross-toolchains, OpenEmbedded, Netbook Project, LART, YAFFS, Balloonboard, xdeb, multiarch, sbuild, build profiles, arm64 port, ilp32 port

- Currently an ARM secondee to Linaro

Bootstrapping

26 Debian ports in 23 years

i386, 68000, Alpha, Sparc, PowerPC, ARM, IA64, PA-RISC, MIPS (big endian), MIPS (little endian), S/390, AMD64, FreeBSD-i386, FreeBSD-amd64, armel, armhf, sh4, s390x, PowerPC64, Hurd-i386, x32, arm64, Mips64el, PowerPCspe, OpenRisc, Nios2

Bootstrapping

26 Debian ports in 23 years

i386, 68000, Alpha, Sparc, PowerPC, ARM, IA64, PA-RISC, MIPS (big endian), MIPS (little endian), S/390, AMD64, FreeBSD-i386, FreeBSD-amd64, armel, armhf, sh4, s390x, PowerPC64, Hurd-i386, x32, arm64, Mips64el, PowerPCspe, OpenRisc, Nios2

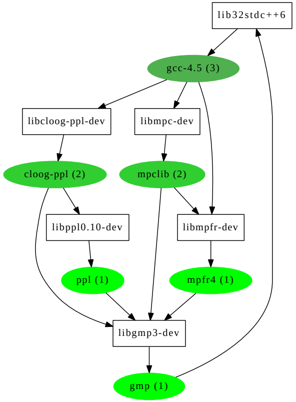
Bootstrapping is normal, not exceptional
We bootstrap more often than we release
A 'Universal OS' should be able to bootstrap itself

The Bootstrap Problem

- Build-dependency loops

The Bootstrap Problem

- Build-dependency loops



The Bootstrap Problem

- Build-dependency loops
- Natively built
- Maximally configured
- Much worse for binary distros than source-based
- Lack of flexibility in packaging, not upstream

Bootstrap solutions

Traditionally

- Cheat and use something else
- Bodgery and Hackery
- No hardware yet - models are really slow

'Universal OS' solution

- Cross Build at least initial chroot
- Linearise build order by reducing dependencies
- Switch to native building when you have 'enough'

First Bootstrap

ARM internal Bootstrap (2011)

- Existing binary cross-tooclchain
- Ubuntu Maverick
- Using xdeb, with staging support
- Equivs to fake toolchain dependencies
- Manual build order
- LAMP stack built

So that’s all done then?

So that's all done then?

ARM is an IP Company

So that’s all done then?

ARM is an IP Company

Now I can be rude about ARM legal

So that's all done then?

ARM is an IP Company

Now I can be rude about ARM legal

- Paranoid about patent grants in FLOSS licences
- No cross-fixes, bootstrapping or arm64 support upstreamed
- Engineers wait 9 months. End up frustrated

So that’s all done then?

ARM is an IP Company

Now I can be rude about ARM legal

- Paranoid about patent grants in FLOSS licences
- No cross-fixes, bootstrapping or arm64 support upstreamed
- Engineers wait 9 months. End up frustrated
- All has to be done again

Valuable IP - avert your eyes:

```
+export DEB_BUILD_GNU_TYPE ?= $(shell dpkg-architecture -qDEB_BUILD_GNU_TYPE)
+
+ifeq ($(DEB_BUILD_GNU_TYPE), $(DEB_HOST_GNU_TYPE))
+   confflags += --build $(DEB_HOST_GNU_TYPE)
+   CROSS=""
+else
+   confflags += --build $(DEB_BUILD_GNU_TYPE) --host $(DEB_HOST_GNU_TYPE)
+   CROSS=$(DEB_HOST_GNU_TYPE)-
+endif
```

On the one hand great early community engagement

On the other complete failure to give back

Illustrates community/corporate culture clash

Linaro helps mitigate

ARM

Second Bootstrap

Debian/Ubuntu Bootstrap Overview

Overview

- Initially Quantal, then Raring -based
  (and Debian Wheezy/Experimental)
- All done in public from start - upstreaming as we go along
- Multiarch building and cross-dependencies
- Profiles used but not upstreamable yet
- Standard tools: sbuild, reprepro, apt, dpkg, dpkg-cross
- Modified dpkg, apt, sbuild for build-profile support
- cross-build-essential: toolchain, libc:arm64, <triplet>-pkg-config
- No qemu available

Debian/Ubuntu Bootstrap Process

1. Prepare repository
2. Add new arch support to dpkg-architecture
3. Set up build chroot
4. Toolchain bootstrap
5. Fix support packages: dpkg-cross, cross-build-essential, autoconf
6. Build stuff…

How much ’stuff’ do we need?

Binary(source) packages needed

	Debian Sid src/binary	Ubuntu Saucy src/binary
Base system	65/116	75/128
+ `build-essential`	69/128	79/140
Sources including build-deps	119/503	
Main SCC	383/2500	

Ubuntu Bootstrap Timeline

Overview

- Start Oct 2012 (Quantal)
- December: moved to raring, dropped Debian
- Linaro doing upstream work in parallel, testing in OE with models
- Debootstrapable Feb 2013.
- Canonical continued from ~June with secret hardware
- New box arrived 3 weeks before Saucy. 2/3rds built
- ‘Soft’ Saucy release
- Easier in Ubuntu due to ’directives from on high’ and looser package ownership
- First ever bootstrap of Debian using itself

Lessons

Overview

- only 10% of work is Aarch64 porting
- 25% cross-building fixes
- 25% configure fixes
- 25% multiarch fixes
- 15% dependency-loop untangling

Third Bootstrap

Debian Bootstrap

Method I had to do other stuff Feb to Oct...

- Native build Debian sources in Saucy chroot
- Nobble dpkg origin and lsb_release info
- Clean Saucy tarball chroot + debianise script
- All deps available - take care to only use debian libs
- Pin debian bootstrap repo as preferred
- debootstrap unstable once build-essential is done
- clean rebuild once SCC done and hardware available

No hardware

- 80-core, 128G Xeon box in Huawei lab - no root access
- Model very slow and annoying (X, network tap)
- qemu-arm64 released Nov 2013 - Way better!
- Linaro has hardware I can’t use due to incompatible lawyers

Debian Bootstrap Issues

- Cyclic dependencies
  - pulseaudio⟶bluez⟶gst-base-plugins⟶libtheora⟶libsdl1.2⟶pulseaudio
  - cups⟶cups-filters⟶ cups
  - dbus⟶systemd⟶audit⟶dbus
- perl!
- debian/ubuntu verison skew
- config.(sub/guess)
- random build failures in unstable
- QEMU limitations

ARM

Debian Bootstrap Summary

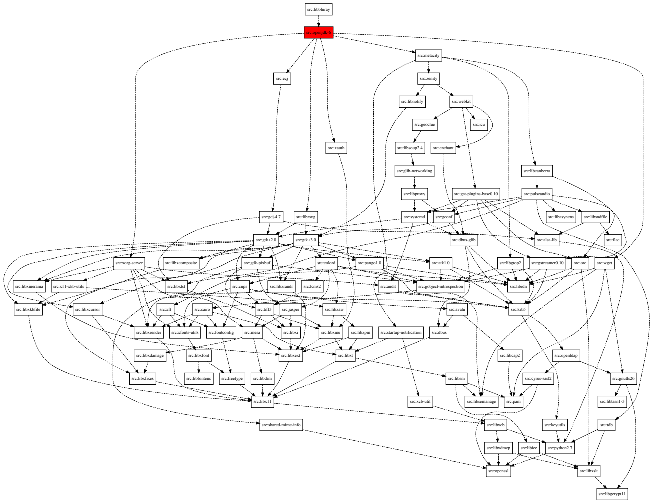
- 348 source Packages, 2109 binaries built
- 523 bugs filed
- Waiting for a buildd…

Useful outcomes

- Google Summer Of Code 2012,2013
- bootstrap.debian.net spun off: Johannes Schauer
- cross-buildable base
- Multiarch proved
- Build profiles demonstrated

Better tools

- Botch



Fourth Bootstrap

Debian Ports

- Space found on Debian-ports
- Mysterious Chinese buildd (2 APMs)
- Rebuild everything natively and upload into debian-ports
- Much faster, much simpler
- Signed with bootstrap key
- 75% of debian built

Fifth Bootstrap

Debian Archive

- Rebuild for proper archive
- 2 Junos from ARM (Aug 2014)
- 3 APM from Linaro (Oct 2014)
- Freeze on 4th November!
- 85% built to qualify for Debian stable release
- Excellent co-operation

Status

- Long tail of things getting fixed
- 85 arm64 bugs still pending
- Now looking at optimisations

Observations

- For servers and desktops distros matter
- Binaries were built 2-10 years ago from software released 2 years before that with toolchains of the same age
- Major implications for errata - can’t just fix it in the linker
- Code will not be optimised

ARM