Automating Architecture Bootstrap

Helmut Grohne et al.

August 17, 2015
Why care about Bootstrap?

- Ca. 1 bootstrap per year
Why care about Bootstrap?

- Ca. 1 bootstrap per year
- Building Debian from source
Why care about Bootstrap?

- Ca. 1 bootstrap per year
- Building Debian from source
- Bootstrap subarchitectures on demand
Why care about Bootstrap?

- Ca. 1 bootstrap per year
- Building Debian from source
- Bootstrap subarchitectures on demand
- Debian as an embedded distribution
rebootstrap

- A QA tool to bootstrap an architecture from nothing

---

1 Thanks to Holger Levsen and ProfitBricks (hardware)
2 Thanks to Bernhard R. Link, Daniel Schepler, Guillem Jover, Johannes Schauer, Peter Pentchev, Wookey, and many package maintainers
rebootstrap

- A QA tool to bootstrap an architecture from nothing
- jenkins.debian.net¹

¹Thanks to Holger Levsen and ProfitBricks (hardware)
²Thanks to Bernhard R. Link, Daniel Schepler, Guillem Jover, Johannes Schauer, Peter Pentchev, Wookey, and many package maintainers
rebootstrap

- A QA tool to bootstrap an architecture from nothing
- jenkins.debian.net\(^1\)
- Tests 20 architectures

\(^1\)Thanks to Holger Levsen and ProfitBricks (hardware)
\(^2\)Thanks to Bernhard R. Link, Daniel Schepler, Guillem Jover, Johannes Schauer, Peter Pentchev, Wookey, and many package maintainers
rebootstrap

- A QA tool to bootstrap an architecture from nothing
- jenkins.debian.net\(^1\)
- Tests 20 architectures
- Cross builds 100 packages

\(^1\)Thanks to Holger Levsen and ProfitBricks (hardware)
\(^2\)Thanks to Bernhard R. Link, Daniel Schepler, Guillem Jover, Johannes Schauer, Peter Pentchev, Wookey, and many package maintainers
rebootstrap

- A QA tool to bootstrap an architecture from nothing
- jenkins.debian.net\textsuperscript{1}
- Tests 20 architectures
- Cross builds 100 packages
- 190 bugs filed, 120 fixed\textsuperscript{2}

\textsuperscript{1}Thanks to Holger Levsen and ProfitBricks (hardware)
\textsuperscript{2}Thanks to Bernhard R. Link, Daniel Schepler, Guillem Jover, Johannes Schauer, Peter Pentchev, Wookey, and many package maintainers
Outline

1. Cross Toolchains
2. Cross Building
3. Architecture Bootstrap
Terminology

GNU Terminology for architectures

- build: what you are building on
- host: where your binary runs
- target: what the resulting compiler will produce

Most of the time: `$DEB_HOST_*`
Cross Toolchains

- Mostly a solved problem\(^3\)

---

\(^3\)Thanks to: Adam Conrad, Aurelien Jarno, Dima Kogan, Dimitri John Ledkov, Matthias Klose, Wookey, Yunqiang Su, and porters
Cross Toolchains

- Mostly a solved problem\(^3\)
- Bootstrap: gcc/glibc “dance”

\(^3\)Thanks to: Adam Conrad, Aurelien Jarno, Dima Kogan, Dimitri John Ledkov, Matthias Klose, Wookey, Yunqiang Su, and porters
Cross Toolchains

- Mostly a solved problem\(^3\)
- Bootstrap: gcc/glibc “dance”
- Two approaches (use cross-arch dependencies?)

---

\(^3\) Thanks to: Adam Conrad, Aurelien Jarno, Dima Kogan, Dimitri John Ledkov, Matthias Klose, Wookey, Yunqiang Su, and porters
Cross Toolchains

- Mostly a solved problem\(^3\)
- Bootstrap: gcc/glibc “dance”
- Two approaches (use cross-arch dependencies?)
- src:glibc still needs patches

\(^3\)Thanks to: Adam Conrad, Aurelien Jarno, Dima Kogan, Dimitri John Ledkov, Matthias Klose, Wookey, Yunqiang Su, and porters
## Problematic Architectures

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>libc packages conflict</td>
</tr>
<tr>
<td>hppa</td>
<td>file conflicts for libc packages</td>
</tr>
<tr>
<td>or1k, riscv</td>
<td>gcc patches not upstreamed</td>
</tr>
<tr>
<td>powerpcel</td>
<td>_old_getrlimit defined multiple times</td>
</tr>
<tr>
<td>sh4</td>
<td>wrong paths in cross compiler</td>
</tr>
<tr>
<td>armel, armhf, sh4</td>
<td>sid glibc cannot be built with default gcc</td>
</tr>
</tbody>
</table>
Status of Cross Building

- Works in principle for years (Emdebian)
Status of Cross Building

- Works in principle for years (Emdebian)
- Mostly works in unstable with sbuild now
Status of Cross Building

- Works in principle for years (Emdebian)
- Mostly works in unstable with sbuild now
- `dpkg-buildpackage -a$HOST`
Satisfiability of Build-Depends

- 3000 packages can satisfy cross Build-Depends

\(^4\) Thanks to Johannes Schauer
Satisfiability of Build-Depends

- 3000 packages can satisfy cross Build-Depends
- http://bootstrap.debian.net/cross{,,_all}.html

\[\text{\textsuperscript{4}}\text{Thanks to Johannes Schauer}\]
Satisfiability of Build-Depends

- 3000 packages can satisfy cross Build-Depends
- http://bootstrap.debian.net/cross{,_all}.html
  - Toolchain dependencies need translation

---

4 Thanks to Johannes Schauer
Satisfiability of Build-Depends

- 3000 packages can satisfy cross Build-Depends
- [http://bootstrap.debian.net/cross{,_all}.html](http://bootstrap.debian.net/cross{,_all}.html)
  - Toolchain dependencies need translation
  - Multi-Archify more packages

---

4 Thanks to Johannes Schauer
Toolchain Dependency Translation

- Build-Depends: g++ (>= 4:5)
Toolchain Dependency Translation

- Build-Depends: g++:$\text{HOST} \ (>= \ 4:5)
  means native compiler for host arch
Toolchain Dependency Translation

- Build-Depends: g++:$HOST (>= 4:5) means native compiler for host arch
- Want g++-$HOST:$BUILD (>= 4:5)
Toolchain Dependency Translation

- **Build-Depends:** `g++:HOST` (>= 4:5) means native compiler for host arch
- **Want** `g++-$HOST:$BUILD` (>= 4:5)
- **Future solution:** B-D: `g++-for-host` (>= 4:5)
Toolchain Dependency Translation

- Build-Depends: g++:HOST (>= 4:5)
  means native compiler for host arch
- Want g++-$HOST:$BUILD (>= 4:5)
- Future solution: B-D: g++-for-host (>= 4:5)
- PoC: src:gcc-cross-support
Multi-Arch

- Apply more Multi-Arch: \{same,foreign\}
Multi-Arch

- Apply more Multi-Arch: \{\text{same,foreign}\}
- Multi-Arch interpreter issue
Multi-Arch Interpreter Issue

application [Arch: any]  
    ↓              perl-module [Arch: all]  
    ↓                    perl-extension [Arch: any]
Multi-Arch Interpreter Issue

- application [Arch: any]
  - perl-module [Arch: all]
    - perl-extension [Arch: any, M-A: same]
Multi-Arch Interpreter Issue

```
application [Arch:any]

↓

perl-module [Arch:any, M-A:same]

↓

perl-extension [Arch:any, M-A:same]
```
Multi-Arch Interpreter Issue

```plaintext
application [Arch:any] / src:package
  ↓
perl-module [Arch:any, M-A:same]
  ↓
perl-extension [Arch:any, M-A:same]
```
Partial Cross Archive Rebuild

- 1100 packages cross built on debomatic\(^5\)

\(^5\)Thanks to Luca Falavigna and Wavecon GmbH (hardware)
Partial Cross Archive Rebuild

- 1100 packages cross built on debomatic\(^5\)
- build/host = amd64/arm64

\(^5\)Thanks to Luca Falavigna and Wavecon GmbH (hardware)
Partial Cross Archive Rebuild

- 1100 packages cross built on debomatic
- build/host = amd64/arm64
- 450 successful

---

5Thanks to Luca Falavigna and Wavecon GmbH (hardware)
Cross Compilation Problems

- Using build arch toolchain
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python means host arch Python
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python: any
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python: any
- Executing host arch binaries
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python:any
- Executing host arch binaries
  - help2man
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python: any
- Executing host arch binaries
  - help2man
  - Needing oneself during build (e.g. cracklib2, nss)
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python:any
- Executing host arch binaries
  - help2man
  - Needing oneself during build (e.g. cracklib2, nss)
  - Native/outdated configure checks
Cross Compilation Problems

- Using build arch toolchain
- Build-Depends: python:any
- Executing host arch binaries
  - help2man
  - Needing oneself during build (e.g. cracklib2, nss)
  - Native/outdated configure checks
  - Maintain config.site
Bootstrap Overview

- Build cross toolchain
Bootstrap Overview

- Build cross toolchain
- Cross build build-essential packages
Bootstrap Overview

- Build cross toolchain
- Cross build build-essential packages
- Continue native builds
Bootstrap Overview

- Build cross toolchain
- Cross build build-essential packages
- Continue native builds
- Break dependency cycles
Package Selection

- Essential: yes and build-essential
Package Selection

- **Essential:** yes and build-essential
- and their Depends
Package Selection

- **Essential**: yes and build-essential
- and their Depends
- and their Build-Depends
Package Selection

- Essential: yes and build-essential
- and their Depends
- and their Build-Depends
- transitively
Package Selection

- Essential: yes and build-essential
- and their Depends
- and their Build-Depends
- transitively

> 500 source packages
Build Profiles

- A way to build different “flavors” of a package

6 Thanks to Guillem Jover and Johannes Schauer
Build Profiles

- A way to build different “flavors” of a package
- Emitted binary packages conditional to profiles

6 Thanks to Guillem Jover and Johannes Schauer
Build Profiles

- A way to build different “flavors” of a package
- Emitted binary packages conditional to profiles
- Build-Depends conditional to profiles

---

6 Thanks to Guillem Jover and Johannes Schauer
Build Profiles

- A way to build different “flavors” of a package
- Emitted binary packages conditional to profiles
- Build-Depends conditional to profiles
- Available since jessie\(^6\)

\(^6\)Thanks to Guillem Jover and Johannes Schauer
Build Profiles

- A way to build different “flavors” of a package
- Emitted binary packages conditional to profiles
- Build-Depends conditional to profiles
- Available since jessie\(^6\)

Example

```xml
<cross>

Extra dependencies for cross compilation
```

\(^6\)Thanks to Guillem Jover and Johannes Schauer
Dependency Reduction

- Use Build-Depends-Indep
Dependency Reduction

- Use Build-Depends-Indep
- Use <nocheck> with DEB_BUILD_OPTIONS=nocheck
Dependency Reduction

- Use `Build-Depends-Indep`
- Use `<nocheck>` with `DEB_BUILD_OPTIONS=nocheck`
- Reduced functionality using `<stage1>`
Dependency Reduction

- Use Build-Depends-Indep
- Use `<nocheck>` with `DEB_BUILD_OPTIONS=nocheck`
- Reduced functionality using `<stage1>`
  - e.g. `src:openldap` and `src:cyrus-sasl2`
musl Port\(^7\)

Another C library

\(^7\)Thanks to Shawn Landden and Szabolcs Nagy
musl Port\textsuperscript{7}

- Another POSIX C library

\textsuperscript{7}Thanks to Shawn Landden and Szabolcs Nagy
musl Port\textsuperscript{7}

- Another POSIX C library
- musl-linux-any

\textsuperscript{7}Thanks to Shawn Landden and Szabolcs Nagy
musl Port\textsuperscript{7}

- Another POSIX C library
- musl-linux-any
- Highlights issues such as missing includes

\textsuperscript{7}Thanks to Shawn Landden and Szabolcs Nagy
Unsolved Problems

- Toolchain dependency translation
Unsolved Problems

- Toolchain dependency translation
- Multi-Arch interpreter issue
Unsolved Problems

- Toolchain dependency translation
- Multi-Arch interpreter issue
- Avoid using host arch for M-A:foreign
Unsolved Problems

- Toolchain dependency translation
- Multi-Arch interpreter issue
- Avoid using host arch for M-A:foreign
- src:perl
Unsolved Problems

- Toolchain dependency translation
- Multi-Arch interpreter issue
- Avoid using host arch for M-A:foreign
- src:perl

Tomorrow: 11:00 Stockholm BoF