6th International Linux-Kongress 1999

The Debian Build Daemon

Roman Hodek

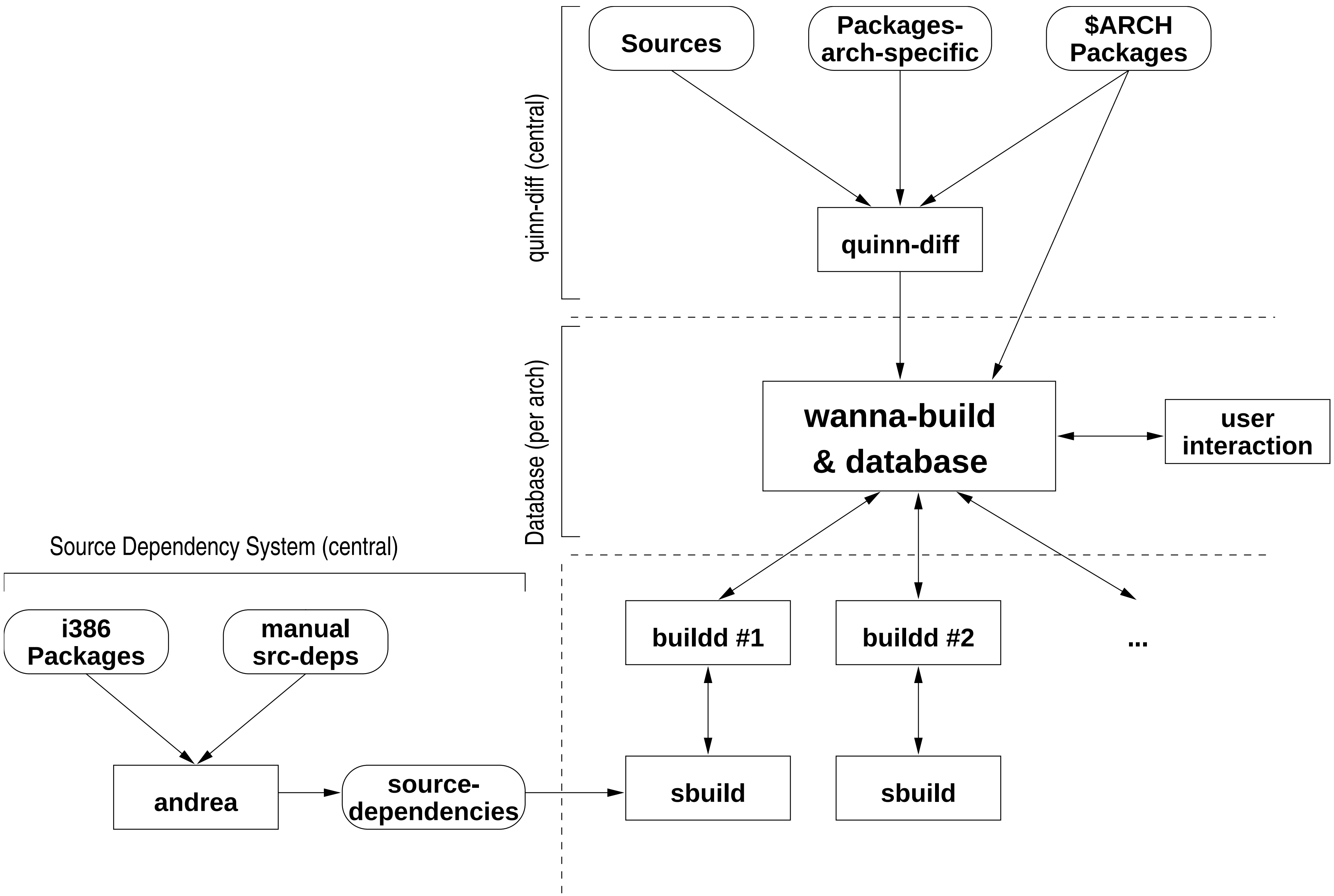
<Roman.Hodek@informatik.uni-erlangen.de>

- Debian is available for 7 architectures now. But most of the package maintainers can provide only i386 binaries.

- Thus all packages have to be recompiled for non-i386 architectures whenever a new source version comes out.

- The build daemon automates this recompilation process as far as possible.

- There may be more than one build daemon for one architecture (work distribution) and it is still possible to manually rebuild packages.


quinn-diff (central)
Sources
Packages-
arch-specific
$ARCH
Packages
quinn-diff
Database (per arch)
wanna-build
& database
user
interaction
buildd #1
buildd #2
...
sbuild
sbuild
Source Dependency System (central)
i386
Packages
manual
src-deps
andrea
source-
dependencies

Major Components:

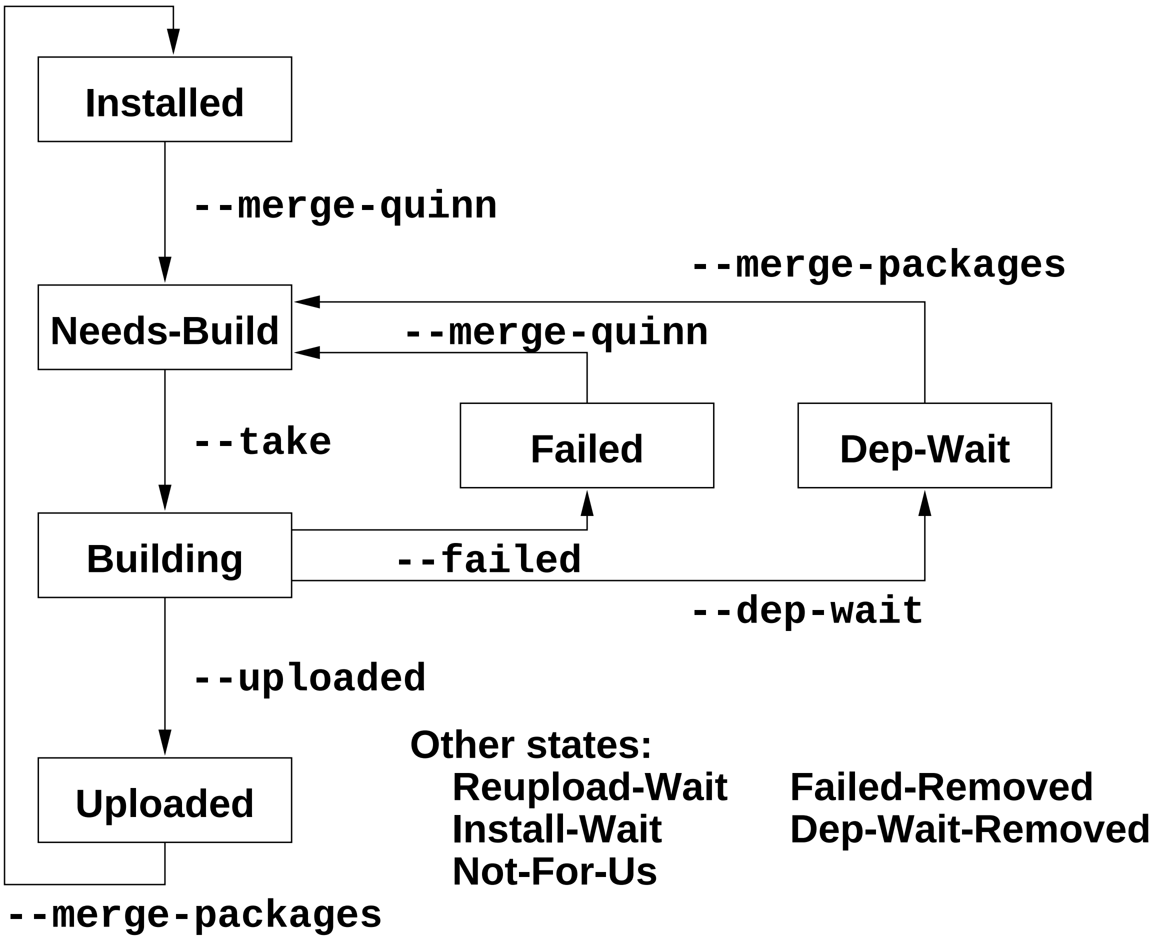
wanna-build provides a database for package states, versions, . . .

buildd looks in the database for packages in state `Needs-Build`, takes care of their recompilation and uploads them after acknowledge by its administrator.

sbuild does the actual compilation. It uses Debian standard tools for this, but additionally knows *source dependencies* and other tricks useful for mass builds.

quinn-diff compares source versions with binary versions of an architecture. Packages with differences are fed into the database and become `Needs-Build`.

andrea generates part of source dependencies automatically and merges the manual source dependencies.


Installed
--merge-quinn
--merge-packages
Needs-Build
--merge-quinn
--take
Failed
Dep-Wait
Building
--failed
--dep-wait
--uploaded
Other states:
Reupload-Wait
Install-Wait
Not-For-Us
Failed-Removed
Dep-Wait-Removed
Uploaded
--merge-packages

Package States in the Database:

Installed: is installed and up-to-date

Needs-Build: there is a newer source version and the package has to be recompiled

Building: someone is currently rebuilding the package (`Builder:`)

Uploaded: package built and uploaded

Failed: there were build errors that require a new source version (`Failed:`)

Dep-Wait: other packages needed for building which aren't available yet (`Depends:`)

Not-For-Us: should be completely ignored for this architecture

Failed-Removed, Dep-Wait-Removed: package disappeared from quinn-diff and previous state was `Failed` or `Dep-Wait`, resp.; it isn't actually deleted to avoid loosing additional infos

Reupload-Wait, Install-Wait: same version of the package has been built in another distribution

Typical Scenario:

- quinn-diff finds a new source version and the package is marked `Needs-Build` in the database.
- A `buildd` takes the package and starts `sbuild` to build it.
- The build log is mailed to the admin, who will acknowledge (PGP-sign) it and send it back if everything is ok.
- If there were errors, the admin sends some reply to the daemon what should happen: set the package to `Failed` or `Dep-Wait`, retry it, . . .
- Acknowledged packages are uploaded to master regularly.
- When the package is installed, it will appear in the `Packages` file and its state will change to `Installed`.

Applications:

- Currently there are 6 build daemons running for 5 architectures (m68k, powerpc, sparc, alpha, and i386; m68k has two).
- The i386 `buildd` is for maintainers which have no i386 machine.
- m68k has the slowest machines, but one `buildd` can keep the distribution up-to-date and still has some idle time.
- `buildd` might also be used for maintaining distribution variants, like a Pentium-optimized i386 distribution.
- The concept of `buildd` is *not* suitable for bootstrapping a new architecture or compiling everything from scratch.

Planned New Features:

- Automatically use freshly built packages for during later builds. Problem: which packages?

- Detect relevant changes in library packages (`shlibs` file) and automatically downgrade them if needed for building e.g. stable packages on an unstable system.

- Make `sbuild` able to set environment variables either for all packages or on a per-package basis; this might be useful for using special compilers and/or options.

- Finally make a Debian package for `buildd` & Co. . . .

`sbuild`

- Fetches and unpacks sources (multiple locations possible), then `dpkg-buildpackage -B`.
- Installs source dependencies before build and removes them again afterwards.
- Synchronizes simultaneous builds wrt. source dependencies.
- Keeps a database for time and space requirements of builds.
- Can automatically apply patches.
- Removes `debian/files` if present (common bug).
- Timeout for builds
- Ensures sufficient disk space before build.
- Can check if certain binaries have been used (for addition to manual source dependencies)

Source Dependencies:

- Currently central approach used, because only that provided quick realization.
- New, policy-conforming source dependencies in packages already supported, too.
- Features: version relations, conflicts (`!`), inline arch specifications, special dependencies (`*`), global patches (`**`)
- `andrea` generates part of the dependencies automatically from binary dependencies (mapping from shlib to –dev packages)
- Manually defined dependencies added
- Special dependencies can conditionally execute shell scripts before and after a build.
- Also patches possible as special dependencies.
- Global patches: are applied to *all* packages; example: big-endian `ltconfig` bug.

Packages:

```
Package: foo
Depends: libncurses4 (>= 4.2-3.1), libfoo, bar
```

sd_map-unstable:

```
libncurses4 => libncurses4-dev
libfoo => libfoo-dev
```

result of andrea:

```
foo: libncurses4-dev (>= 4.2-3.1), libfoo-dev
```

madd-unstable:

```
foo: bison, %libfoo, *ETC-FOO, !xxx, baz [m68k]
```

madd-special:

```
*ETC-FOO:
cond { ! [ -d /etc/foo ] }
pre  { sudo mkdir /etc/foo }
post { sudo rmdir /etc/foo }
```

result after merger:

```
foo: libncurses4-dev (>= 4.2-3.1), bison, \
     *ETC-FOO, !xxx, baz [m68k]
```