# Splitting gcc's binary packages

Thibaut Girka

November 25, 2012

# Splitting what?

Currently, src:gcc-4.7 produces...

- libgcc1 but no libgcc*-dev
- libobjc4 but no libobjc*-dev
- libgfortran3 but no libgfortran*-dev
- etc.

# Splitting what?

So, were are those files?

- libgcc's development files are in gcc-4.7
- libobjc's in gobjc-4.7
- libgfortran's in gfortran-4.7
- etc.

# Why?

- doko decided it was required to fix #678623.

# Why?

- doko decided it was required to fix #678623.
- it reduces conflicts between cross-compilers

# Why?

- doko decided it was required to fix #678623.
- it reduces conflicts between cross-compilers
- it is useful for other compilers (clang)

# #678623

- g++-4.7-arm-linux-gnueabihf depends on libstc++6-4.7-dev:armhf
- libstdc++6-4.7-dev:armhf depends on g++-4.7:armhf
- g++-4.7:armhf conflicts with g++-4.7:native
- Solution: drop the dependency

# Conflicts between cross-compilers

- gcc-4.7:armhf and gcc-4.7-arm-linux-gnueabihf have common files
- moreover, such conflicts are not easily expressed
- Solution: move those common files to a M-A: same package

# Other compilers

- clang links against libgcc and other runtime libs
- libgcc dev files are in gcc-4.7
- Solution: move dev files out of gcc-4.7

# The situation now

Development files have been (in experimental) split out of the frontends.

- gcc-$VER $\mapsto$ gcc-$VER + libgcc-$VER-dev
- gobjc-$VER $\mapsto$ gobjc-$VER + libobjc-$VER-dev
- gfortran-$VER $\mapsto$ gfortran-$VER + libgfortran-$VER-dev

[TODO: Thank Daniel Schepler]