MOVIAL

IDEAS IN MOTION

Scratchbox 2.0

Bringing crosscompiling to Debian

Riku Voipio <riku.voipio@movial.fi>

Movial Corporation

May 17, 2006

Part I

History

Where Scratchbox came from

- Build environment for a custom OpenEmbedded style embedded Linux distribution
- Broken `./configure`, ad-hoc build systems
- Fixing packages to crosscompile was tedious and boring
- Raw often packages buggy and/or missing target arch patches

What developers did

- The created CPU transparency!
- And some other workarounds needed to make `./configure` think it's compiling natively
- The environment was christened Scratchbox
- Opened possibility to build "unmodified" Debian sources

Part II

What is Scratchbox

What is Scratchbox?

"Cross-compiling environment"

"Cross-compiling toolkit"

"Cross-development tool"

"Self-contained mini-distribution"

"Linux From Scratch-sandbox"

And in practice?

- CPU Transparency
- Host tools and target libs
- Binary redirection
- Fakeroot integration
- GCC wrapper
- Other tweaks (uname returns target arch)
- Hubris

CPU transparency

- Ability to execute target binaries while crosscompiling
- Broken `./configure` tests
- Tools that are built as part of package build process to be used later in the same build (monolithic X)..
- Originally used SSH+NFS
- SSH was replaced with SBRSH
- Now also QEMU userland emulation

Host tools

- Build time is not only gcc - flex,doxygen,unpacking sources waste lots ot time too
- Host tools are located under /scratchbox
- `--rpath` and `--prefix` set + patches and hacks
- Compiled specifically for Scratchbox
- Means that Scratchbox already comes with all the bells and whistles you need for development.

Target libs and filesystem

- in /lib and /usr/lib instead of /opt/foo/target-libs or /usr/lib/arch-xx/
- no need to mangle where to install stuff
- apt-get can be used to manage target arch libs and packages!
- multiarch would make this unneeded
- sbox target root is almost ready rootfs/nfs-root

Binary redirection

- debian/rules: `#!/usr/bin/make -f`
- redirect exec of /usr/bin/make to /scratchbox/tools/bin/make
- beats hardcoded paths
- unfortunately leads to complications when compiling perl/python modules with C code

CroCoDiLe

- A tool that allows bootstrapping an arch from sources only
- Installs core libs (libc6,libgcc,sdtc++) from toolchain
- Avoids build-dep loops by using sbox host tools.
  - make,gcc,flex,bzip2,docbook-foo,tetex,doxygen..
- Tested to crosscompile debian base from scratch on combinations like arm-eabi/glibc to cris/uclibc
- ..since Scratchbox host tools are still at sarge level, only bootstrapping sarge possible at the moment.

Part III

Alternatives?

Alternatives: Native compiling

- Compiling natively on a target arch computer
  - + No crosscompiling related problems (duh)
  - – Really slow
  - – On slow arch's where everyone uses crosscompilers, native compilers might be completely untested!
- Example: Debian buildd's

Alternatives: fix buildsystems

- fix autotools tests, workaround exotic buildsystems
- pass proper `--host=i486-linux-gnu --build=arm-linux-gnueabi -L.. flags`
  - + The proper$^{\text{TM}}$ way
  - + OpenEmbedded etc. people have fixed many many packages already
  - + Many exotic buildsystems are getting extinct (imake, ya)
  - – Managing target libs/tools hard (dpkg-cross..).
  - – Tedious if you work with lots of packages
  - – Not always easy - dynamic code generation for example.
  - – New NiH buildsystems are appearing, scons, cmake..
- Example: OpenEmbedded

Alternatives: distcc from native host

- dpkg-buildpackage runs on target arch
- when gcc is called, distcc is used to execute crosscompiler on fast system instead of slow local gcc
  - + Crosscompiles pretty much anything without effort
  - – Most time building a Debian package is often not consumed in gcc
  - – Doesn't help solving build dependency loops
- Example: Debian Armeb buildd, linksys nslu-2 to a distcc host
- Example: see "MythTV on Philips Nexperia^{TM} PNX8550" presentation

Part IV

Scratchbox 2.0

What is Scratchbox 2

- from `exec()` redirect to `open()` redirect
- apt-get install host tools also!
- Scratchbox gets small enough to be included in Debian!
- ...unfortunately not as ready as I hoped

The Pain

- "Sarge is old, why doesn's sbox have etch version of tools?"
- Manual work to update host tools
- No interest to maintain such a huge pile (at least for free)
- "Why can't we just apt-get host tools" ?

The solution

- Make unmodified applications run from /scratchbox/
- Execute host binaries with ld.so –library-path /scratchbox/lib:..

DONE By Toni Timonen

- Make applications read their config/data files from under /scratchbox

WiP

- Thus the need for open redirection

Open redirection #1

- For example /etc/latex2html.conf needs to go to /scratchbox/etc/latex2html
- First implementation: LD_PRELOAD based library.
- based on modified fakechroot

DONE at extremadura (*Me And Lauri Leukkunen*)

- Second implementation: fuse engine
- based on mirrorfs

DONE at extremadura(*Toni Timonen*)

Open redirect #2

- For hacking purposes lua scripting used for redirect config
- In /scratchbox/redir_scripts
- Now left the hard questions, what should we redirect and how
- Probably ”As little as possible” to avoid unexpected side effects

Open redirection #3

```
function sbox_translate_path(binary_name,
func_name, orig_path)

    if (binary_name == "latex2html" &&
        orig_path == "/etc/latex2html.conf") then
        return "/scratchbox/"+orig_path
    end

    return orig_path
end
```

Fuse vs LD_PRELOAD

- Fuse
  - + fast
  - + robust
  - − requires a recent kernel
  - − knows less context

Fuse vs LD_PRELOAD

- LD_PRELOAD
  - + works on any kernel
  - + can get complete context of process
  - + LD_PRELOAD might be needed anyway
  - − slower
  - − more fragile

Revolution?

- Completely new support scripts
- pbuilder-style ?

  Scratchbox2 create toolchain /var/foo

  Scratchbox2 update /var/foo

  Scratchbox2 build /var/foo xfree86.x.x.dsc

  Scratchbox2 login

  ...

  + can be uploaded to Debian quickly
  − will take long to reach the stage of current Scratchbox

Or evolution?

- Add open redirection to current Scratchbox Apophis

  inside Scratchbox: host-apt-get install foobar

  replace custom sbox tools with vanilla tools as needed
  - + can be made work really soon
  - + doesn't break working habits of current users
  - − will take very long until it's slim enough for Debian

Questions?

- ?

Further links

- http://scratchbox.org/
- http://scratchbox.org/wiki/Scratchbox2
- http://scratchbox.org/wiki/Crocodile
- http://scratchbox.org/cgi-bin/darcsweb.cgi