Debian ARM ports
and the new ABI

Wookey
Aleph One Ltd, Embedded Debian
wookey@aleph1.co.uk
http://www.aleph1.co.uk/

Intro

Debian and its ports

Debian infrastructure and mechanisms

EABI changes

New armel port

Debian - The universal OS

- Handhelds to Mainframes
- Release architectures:
  - alpha, amd64, arm, hppa, i386, ia64, mips, mipsel, powerpc, s390, sparc
- Non released official architectures:
  - hurd-386, m68k
- debian-installer, emdebian
- Unofficial builds adds:
  - armeb, kfreebsd-amd64, kfreebsd-i386, m32r
- External projects:
  - nexenta

Some Statistics

unstable has

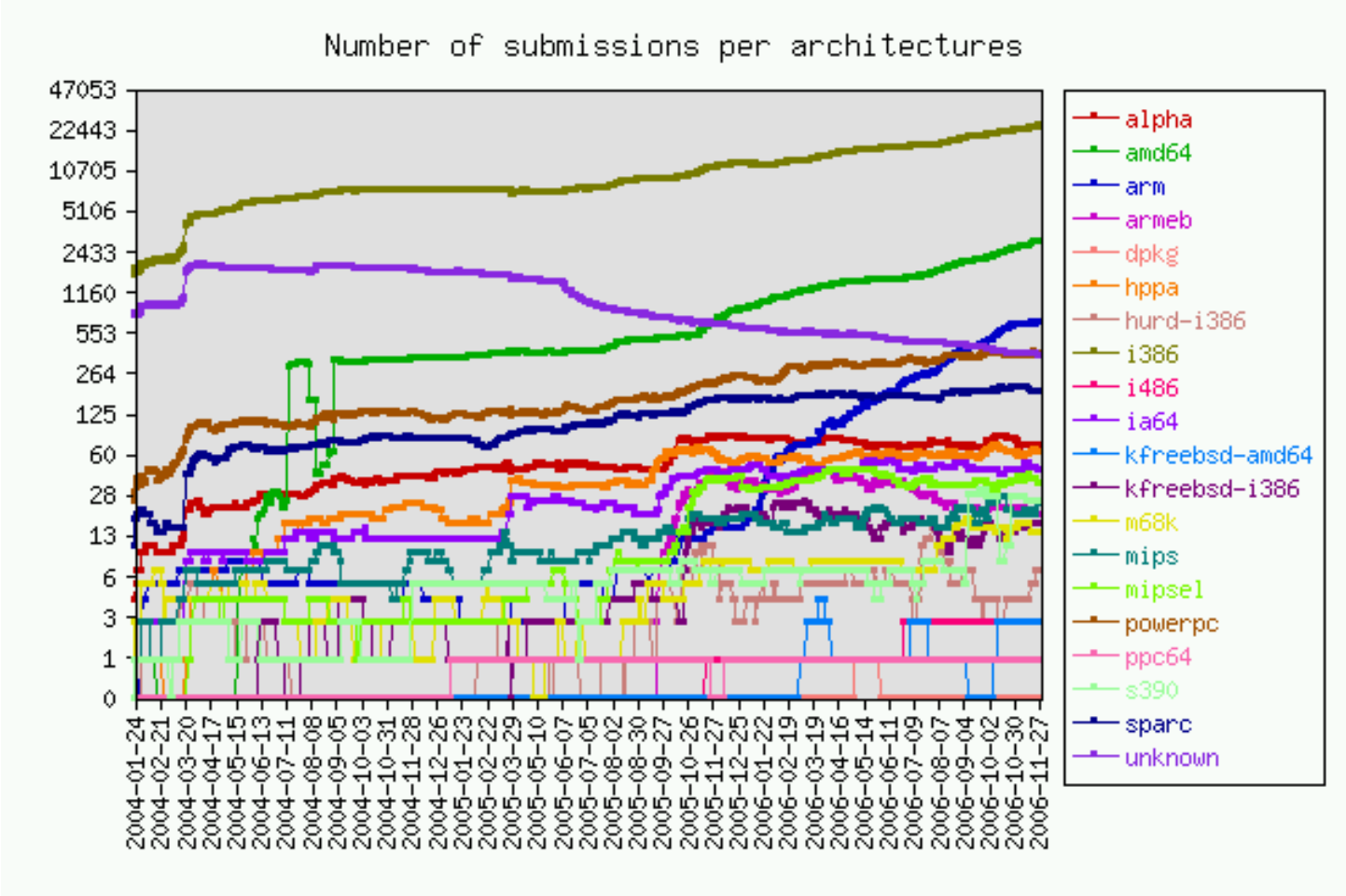
- 10,783 source packages
- 18,317 binary (arm) packages
- 2,042,254 files
- (316 source, 356 binary in contrib/non-free)

99% of suitable packages build for arm
96% of i386 packages

about 1000 developers

13GB per architecture - 21 CDs

ARM port popularity



Debian ARM port

- ARM-Linux started 1998
- Debian-arm started in 2000
- Netwinder
- RiscPC, Cats: 2001
- Lart, Bast: 2003
- Iyonix, Manga: 2004
- NSLU2, Thecus: 2005/6

Many others without debian-installer support

Architecture Release Criteria

- Available in the market without NDA
- 50 users, 5 developers
- Working installer
- Upstream and debian porter support, esp toolchain
- 95% of archive built
- Must keep up with 2 buildds (relaxed for arm and m68k)
- Veto by release team, security team.

Debian Suites

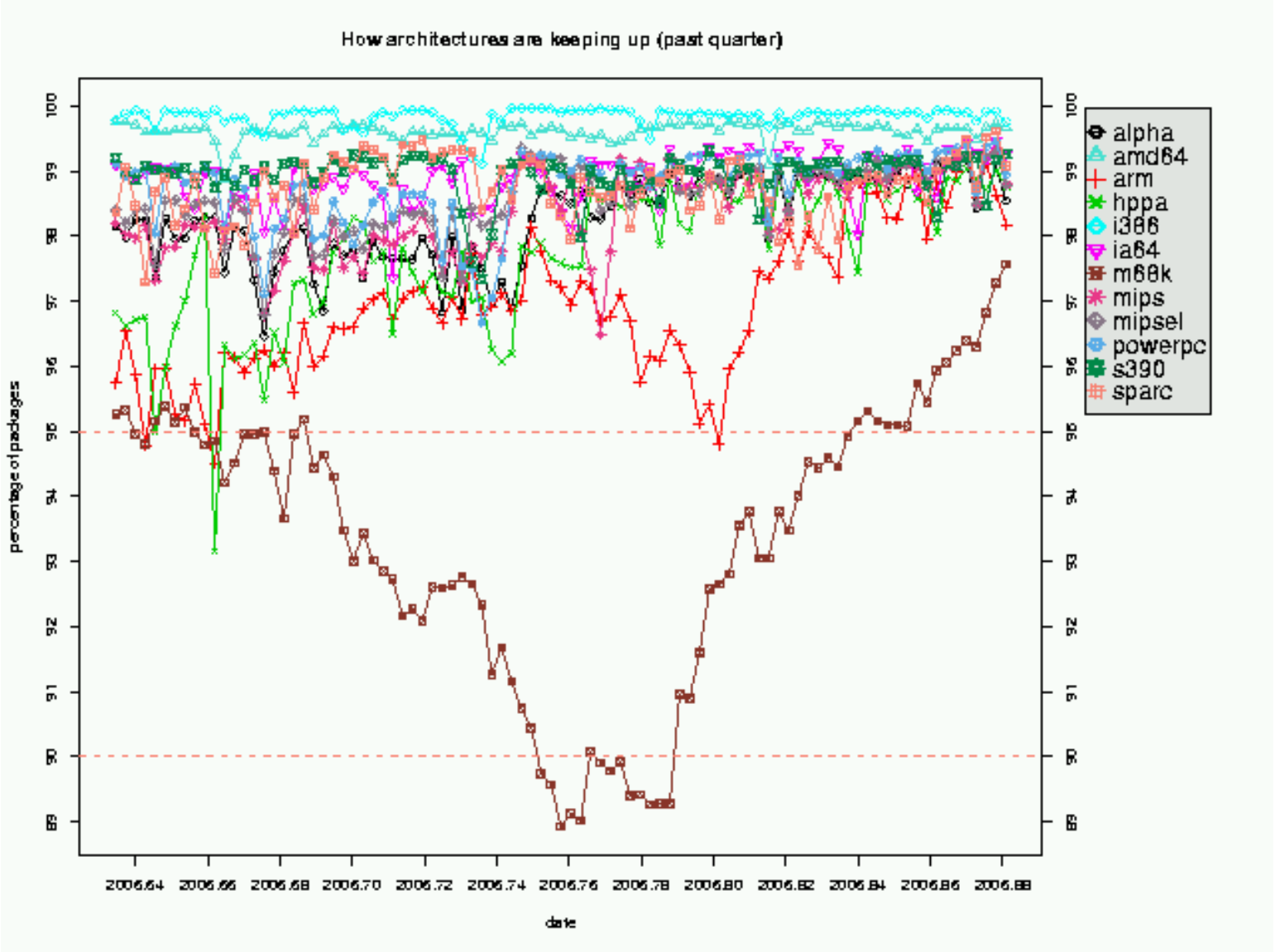
- Experimental
- Unstable (sid)
- Testing (etch)
- Stable (sarge)
- Oldstable (woody)

Process

Everything is autobuilt, except uploaded package.

- DD builds, signs, uploads
- Buildd network processes:
  - Checks dependencies
  - Hands out for build
  - Classified as ’built-OK’, ’maybe-successful’, ’failed’.
- Packages migrate when:
  - All arches are done
  - No release-critical bugs filed
  - Ready for 10 days in unstable
  - Dependencies satisfied
  - Build failure on any release arch will block.

Architectures - percent-built



Maintaining arm port

Big job. Few people. Help welcome
QEMU helps

Issues:

- Java
- Mono
- C++ link timouts
  - Mozilla
- Haskell

Debian infrastructure

- packages.qa.debian.org
- buildd pages and logs
- ’Why is package X not in testing’
- Bugs database
- Wiki
- Porter machines

EABI changes

- Structure packing
  - Old ABI had min structure packing size of 4 bytes
  - EABI has no minimum - packing is determined by type sizes
- Argument alignment
  - 8-byte stack alignment at public function entry points (was 4)
  - 64-bit data types (e.g. long long) are 8-byte aligned (was 4)
- Enums
  - EABI allows enums to have variable type size (-mabi=aapcs)
  - Not used on GNU/Linux - they remain as 4-bytes. (-mabi=aapcs-linux)
- Floating point
  - Mixed-endian LE format goes away
  - Can mix GCC softfloat and FPU hardfloat/emulation

New syscall convention

- More efficient on harvard architecture
- Changed in kernel 2.6.15 - mainline 2.6.16
- Kernels retain old syscall compatibility so they can run older binaries.
- Speed gain is not realised unless compatibility disabled
- glibc 2.3.6 uses old syscall interface via shims
- Shims removed in glibc 2.4 and 2.3.7 - no longer supported

Why do we care?

Pros

- Standardisation across toolchains, debuggers
- Most arm wierdness removed (FP formats, packing, C++ exceptions)
- Hard/soft float interworking
- Thumb interworking
- Interchangeable binaries (PalmOS, GNU/Linux, Symbian OS)
- More efficient syscall convention

Cons

- Almost total incompatibility

Timeline

- new ABI published Dec 2003

- Code sourcery 1st cross-tools q3 2005 v 3.4.4

- 2005: Early Linux adopters (montavista, nokia) - shimmed glibc
- Kernel syscalls changed during 2.6.15 - Feb 2006
- Debian port started q1 2006
- Code sourcery gcc4.1 cross-tools Q1 2006
- Angstrom OE EABI Aug 2006

Tools

- GCC
  - work done by Code Sourcery
  - eabi support in CS gcc3.4.4 (with -mabi=aapcs-linux)
  - From 4.1: different arch
    - Old ABI is called linux-arm-none-gnu
    - EABI is called linux-arm-none-gnueabi
- Glibc
  - shims in 2.3.6
  - new syscalls in 2.4 and 2.3.7
  - shm broken in 2.4 - fixed in 2.5
- Kernel
  - support from 2.6.16
- QEMU
  - support from 0.8.1

Tools (2)

All-new toolchain needs:

- gcc-4.1,
- glibc-2.4+glibc-ports-2.4, (or 2.3.7)
- binutils-2.16.91.0.7 (or similar) and
- linux-2.6.16.
- This can be compiled using crosstool-0.42
- May 2006, or Oct for glibc2.5

Debian port

- Worth changing to
  - Avoid obsolescence
  - Fix the FP problem
  - Build stuff that never worked
- Binary compatibility not an issue for free software, but still convenient.
  - (e.g. commerical debuggers).
- Incompatibility with existing port a problem.
- How to make the change?...

Rename all library packages

Pros

- Can do apt-get dist-upgrade

Cons

- Every single library package needs to be renamed
- Will take a long time, during which unstable will be broken for all arches (6months for C++) - 2yrs?
- Not popular due to large hassle for other arches
- Will lose v3, may lose v4 support.

New architecture

Pros

- Fits with gcc approach
- Does not affect non-arm arches
- Can keep ’arm’ for v3 and maybe v4 machines
- Can be done relatively quickly as no interaction with other arches/releases

Cons

- Current arm users don’t have easy upgrade path
- Need archive space for new arch

ABI: field in control file

Suggested as part of multiarch proposal

Pros

- Reflects ABI correctly, would help other transitions too

Cons

- No existing implementation
- No consensus on including it yet
- Questions over resolving dependencies and how it fits into archive

New Arch was chosen

- Called armel - decided at Emdebian extamadura meet
  - Nokia then used same name for Maemo
- Introduced in etch +1
- May have armeb too
- Existing arm phased out in etch+2

Issues for port

Instruction set choice:

- EABI problematic on v3/v4
- Thumb interworking
- GCC versions: 4.1.0 broken for v4t

Glibc version

- 2.3.6 in etch
- 2.3.999 (now 2.5) in experimental

Thumb interworking

- EABI alows thumb/arm mixing at function level granularity

Current GCC:

- -march=armv4: mov pc,lr
  - v4 onwards, only interworking-safe from v7
- -march=armv4t: bx lr
  - v4t onwards, interworking-safe

Modified GCC:

- tst lr, #1; moveq pc, lr; bx lr
  - v4 onwards, interworking on v4t onwards. extra instructions
- ldm/ldr:
  - v4 onwards, interworking on v5t onwards.

- Debian maximises device coverage, not speed

Debian port process

- Get working toolchain
- Get working kernel
- Get working Rootfs
- Patch/build armel from debian sources
- Debootstrap Buildd

Bootstrapping Debian is hard

- Not designed to be built from scratch
- No docs for a reason!
- Circular dependencies (libc6 gcc-4.1)
  - Doc-building: groff, tetex, dvi, ps2html
  - gettext wants java
- Patches needed to simplify
- 29 essential packages
- 124 base and required packages
- 16 build-essential packages
- 400-odd build dependencies

Bootstrapping mechanisms

- Plain Crossbuilding not suitable
- Scratchbox+crocodile. Possible, but problems.
- OE angstrom - dependencies, busybox, minimal versions.
- Maemo - old glibc/gcc but works well enough

Using QEMU and mpcore board

3-stage build process

- 1. Bodge a working rootfs to build in
  - Build etch armel packages - tainted but adequate
  - Quite a small set of patches needed
- 2. Debootstrap armel packages
  - Rebuild kosher packages
- 3. Debootstrap buildd to rebuild world
- Making use of old syscall compatibility in kernels

Current Status

- 96 out of 124 needed packages built
- along with 274 build dependencies built
- Several more every day but have some tricky ones left:
- libc6, gcc-4.1, perl, python
- Repository at http://ftp.uk.debian.org/debian-armel/

Base

Needs:

- base-files base-passwd bash bsdutils coreutils debconf debconf-i18n debianutils diff dpkg dselect e2fslibs e2fsprogs findutils gcc-4.1-base grep gzip hostname initscripts libacl1 libattr1 libblkid1 libc6 libcap1 libcomerr2 libdb4.3 libdevmapper1.02 libgcc1 liblocale-gettext-perl libncurses5 libpam0g libpam-modules libpam-runtime libselinux1 libsepol1 libslang2 libss2 libstdc++6 libtext-charwidth-perl libtext-iconv-perl libtext-wrapi18en-perl libuuid1 login lsb-base makedev mawk mktemp mount ncurses-base ncurses-bin passwd perl-base procps sed sysvinit sysv-rc tar tzdata util-linux zlib1g

Still pending:

- bsdutils dpkg dselect gcc-4.1 libc6 libstdc++ login mount passwd perl-base

Required

Needs:

- adduser apt aptitude apt-utils bsdmainutils cpio cron dhcp-client ed gettext-base gnupg groff-base ifupdown info iptables iputils-ping klogd laptop-detect libbz2-1.0 libconsole libdb4.2 libdb4.4 libgcrypt11 libgdbm3 libgnutls13 libgpg-error0 libldap2 libldap-2.3-0 liblzo1 liblzo2-2 libncursesw5 libnewt0.52 libopencdk8 libpopt0 libreadline5 libsasl2 libsigc++-1.2-5c2 libsigc++-2.0-0c2a libssl0.9.8 libtasn1-3 libtasn1-3-bin libusb-0.1-4 libwrap0 logrotate man-db manpages module-init-tools modutils nano netbase netcat net-tools openbsd-inetd readline-common sysklogd tasksel tasksel-data tcpd traceroute vim-common vim-tiny wget whiptail

Still pending:

- adduser apt aptitude apt-utils gnupg libgnutls13 libgpg-error0 libldap2 libsasl2 libsigc++ manpages netbase tasksel vim-common vim-tiny whiptail

Build-essential

Build-essential needs:

- binutils cpp cpp-4.1 dpkg-dev g++ g++-4.1 gcc gcc-4.1 libc6-dev libssp0 libstdc++6-4.1-dev linux-kernel-headers make patch perl perl-modules

Still pending:

- gcc-4.1 libssp0, patch, perl

That’s all folks