THE STATE OF ARM64

WOOKEY, WOOKEY@WOOKWARE.ORG
WHO IS WOOKEY?

- Debian Developer, ARM porter
  Embedded/multiarch/crossbuild/bootstrap
- Linaro, seconded from ARM
Ramble Through State

- Supposed to be BOF-\( y \)
- Built, working, optimised, missing
- Hardware availability
- Benchmarking, what next
MOSTLY WORKING!

- Nearly everything builds
I'm a build engineer
If it builds, it's done.
So we are DONE :-}
MOSTLY DONE

• Things people care about work
• Some things still need porting
BUT HOW WELL DO THEY WORK?

- Some things may not work at all...
- Popular things are optimised
- Many things are not
- Need feedback from real users
SOME HISTORY

- Started 2010 - toolchains
- 'Fast' Model 2011
- Qemu 2013
- Hardware March 2014
5 BOOTSTRAPS

- Crossbuilt internal bootstrap 2011
- Crossbuilt ubuntu bootstrap 2012
- Ubuntu rebootstrap 2013/14
- Debian ports rebootstrap 2014
- Debian official rebootstrap 2014/15
NOW

- 96% built (11205 packages)
- 314 failed
- 370 not for us
Languages

- Optimised
  - C, C++, Java (8&9), Python, Perl, Ocaml, Javascript (v8), Haskell (ghc), Lisp (SBCL) (2 months)
- Ported
  - Lua, R, Rust, Golang, Julia (3 days), Perl6 (3 months), Javascript, Pascal (fpc) (just now!),
- Missing
  - Mono, libphobos (D), Luajit, ?
STILL MISSING

- Mono (23)
- libphobos (D-libraries) (11)
MONO

- Blocks 23 packages
- Done but not released
- Does anyone care enough?
- Coreclr instead? (not in Debian yet)
FREE PASCAL (FPC)

- Blocks 22 packages
- Port Done
- Uploaded yesterday :-)
NOT MISSING

- Nodejs - libv8
- Julia - uploaded 3 days ago
- golang - (base support in v1.5)
- Lua (but no luajit)
- Ocaml (native in 4.02, 2015.07)
- SBCL (2015.10)
- Perl6 (2015.10)
OPTIMISATION

- C fallback
- Assembler, Intrinsics, Neon
ASSEMBLER IN PACKAGES

1000 bits of assembler:

- https://wiki.linaro.org/Platform/DevPlatform/ArmSoftwareList
- http://performance.linaro.org/

Removing often better than 'fixing' (e.g. alsalib)
WHAT’S OPTIMISED

- GCC (compilers)
- LLVM (no linker) (compiler)
- Java 8 & 9
- openSSL
- libV8 (Javascript jit)
- fftw3 (FFT library) Neon support
- gnu-mp (gmp) (Feb 2013)
- hadoop (crc, using HWCAP)
- ceph (crc, using HWCAP)
- Kernel (raid6, crypto)
- Xen, (loads less code on arm64)
NOT OPTIMISED

- Ionmonkey (Mozilla JIT)
- Golang (being worked on)
- OpenCV
- ffmpeg/libav (parts)
- Openoffice
- Lua (luajit)
- R?
- What else?
BLOCKING PACKAGES

- mono (23), fpc (22), libphobos(11)
- dietlibc (5)
- insighttoolkit (3)
- timblserver (2)
- umview (3)
- kexec-tools (2)
NEVER BUILT
(‘AUTO NOT FOR US’)

- For other arches
  - nvidia-support
  - nvramtool
  - powerpc-utils
  - raxml
- Probably should be fixed
  - openafs
  - rootstrap
  - scm
  - qtemu
  - darktable
  - lots more
I HAVE NO HARDWARE!

ONLINE

- OVH (Runabove.com) ThunderX
- OBS (Open Build System)
- Debian porter boxes
I HAVE NO HARDWARE!

HARDWARE

- HP Moonshot
- ARM juno
- APM C1
- Softiron 3000
- Gigabyte MP30
- Hikey (96boards)
- Dragonboard 410c(96boards)
- Anaconda
- Pine64
Juno (ARM A57)

$6000
Moonshot (HP)

$10000+$
Softiron 3000 (AMD seattle SOC)

$2500
Gigabyte MP30 (X-gene SOC)

€950
Hikey (96boards, Hisilicon/Lemaker)

$75 (1G) $99 (2G)
Dragonboard 410c (96boards, Qualcomm)

$75
Pine64 (Allwinner A64)

May 2016 $15 (0.5GB) - $29 (2G)
Benchmarking across architectures is difficult. What are good tests?

Sometimes it's obvious (botch)

<table>
<thead>
<tr>
<th>Arch</th>
<th>Build Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>amd64:</td>
<td>37m</td>
</tr>
<tr>
<td>arm64 (generic ocaml):</td>
<td>4hrs 52m</td>
</tr>
<tr>
<td>arm64 (native ocaml):</td>
<td>1hr 15</td>
</tr>
</tbody>
</table>
BENCHMARKING

Find good tests?
Broadly equivalent platforms
Look for changes over time
SO, SUGGESTIONS?