



**Linaro
connect**
Vancouver 2018

LCG: Lightning Talks

LCG Team



Agenda

- Project Sharp - Tom Gall
- AOSP and external Upstreams - John Stultz
- Clang Master updates - Bero
- My experiments with remote lkft lab - Sumit Semwal





Linaro
connect
Vancouver 2018

Project Sharp

LCG Lightning Talks - September 17 2018

Director Linaro Consumer Group : Tom Gall



Project Sharp

Mission: Detect kernel regressions in mainline, stable and LTS kernels before they reach downstream kernels

1. Patch Gathering for stable kernels
2. Running Tests and Triaging results on participating hardware across mainline, -next, stable, 4.4, 4.9 & 4.14 kernels when running OE and Android
3. Expand coverage and maintain test suites in use
4. Dev boards in support of validation efforts



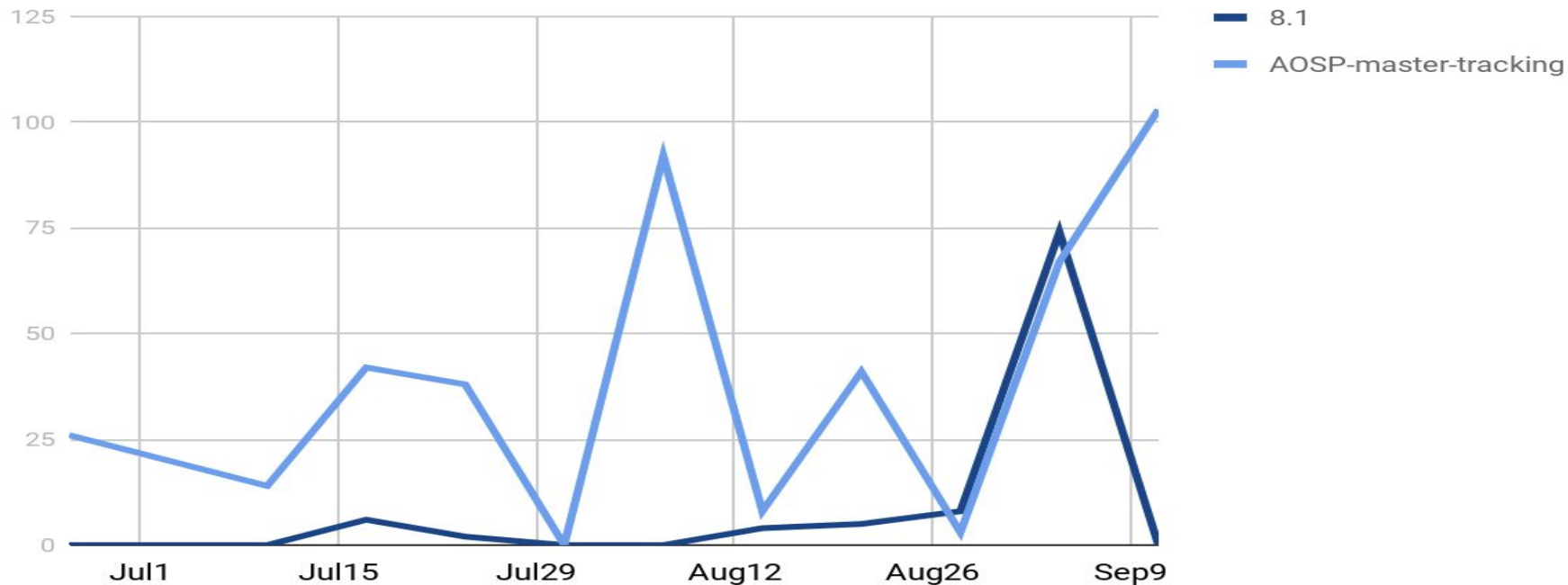
Deployed Boards, Kernels

Board	Lab Status	OE - Linux	AOSP - Android Common
HiSilicon Hikey 6220 - 64bit	Deployed	4.4, 4.9, 4.14, 4.19, Stable, Mainline, -next	4.4, 4.9, 4.14, 4.19
TI X15 - 32bit	Deployed		4.14, 4.19
Arm Juno-r2 - 64bit	Deployed		N/A
QEMU Arm (32bit and 64bit)	Deployed		N/A
QEMU x86 (64bit and 32bit)	Deployed		N/A
X86 - 64bit	Deployed		N/A



Regressions over time...

"Failures" Detected



What makes up the failures?

- Infrastructure (network seems to hit about once a month)
- Flaky Testcases
 - Ex: VtsKernelQtaguidTest.SocketTagUsrSpaceTest.* - kernel config off in some combos, testcases don't test for it
 - Ex: Thread issue - <https://android-review.googlesource.com/c/kernel/tests/+/696462>
- Real Regressions
 - Ex: Tradefed Change
<https://android-review.googlesource.com/c/platform/test/vts/+/712372>
 - Ex: kselftest - seccomp_bpf https://bugs.linaro.org/show_bug.cgi?id=3671



Thoughts

- All data is public - test results via email
 - weekly summary available for members
- Email reports -> Triage -> Bugs -> Fixes
- Fixing more testcases than kernel regressions
- AOSP-master-tracking is most chaotic, but we expect that
 - Changes going on in Kernel, Testcases and Frameworks
- Qa-reports
 - Delta/historical data most useful
 - **Getting to tradefed output is painful and what Android engineers WANT TO SEE**





**Linaro
connect**
Vancouver 2018

AOSP & External Upstreams

John Stultz
<john.stultz@linaro.org>



Lots of good work going on

Android developers have a new focus on upstreaming kernel changes!

Work ongoing in keeping AOSP/master synced w/ external freedesktop.org projects.

- drm_hwcomposer
- libdrm
- mesa



New friction points


AOSP tends to not worry much about compatibility with older Android releases

- External upstreams want latest build to work w/ at least all active Android releases: Lollipop(2014) on.

AOSP has increasing requirements for using Blueprints instead of Makefiles

- Often not backwards compatible

Treble requirements creating new strict rules about what code and libraries can access and do.

- A push to compile out some library dependencies
 - Upstream may not be receptive/accomodating to these rules
- 

TODO

- Need to foster communication and interaction between Android and upstream developers (not just with the kernel community!)
- Help share / improve understanding of each sides concerns
- Emphasize the benefit of working together
- Avoid getting to a place where forking is the easiest answer





Linaro
connect
Vancouver 2018

Clang master updates

Bero



Clang master

Currently on ci.linaro.org:

- Building nightly snapshots of LLVM+Clang+Polly+OpenMP+Compiler-RT master
- Trying to build AOSP with it
- Fix any bugs in AOSP detected by this; report bugs in clang upstream



News on clang master builds

- Now using a variant of AOSP's toolchain build scripts - this makes it easier to keep in sync and generate the non-standard files (symbol maps etc.) the AOSP build system relies on
- This will also make it possible to build NDK toolchains more easily



Possible future projects

This is just ideas that may or may not happen, depending mostly on member input:

- More testing on clang release branches (this is currently being done manually)
- Build toolchains for the NDK
- Enable support for extra languages (having Swift available in an NDK build should be interesting to app developers trying to port an app written for that other mobile OS)
- Build aarch64 native compilers



Possible future projects

This is just ideas that may or may not happen, depending mostly on member input:

- Add binutils (currently using AOSP's version, may be nice to test 2.31.1 if not nightly snapshots of binutils along with the compiler)
- Go beyond “it compiles, therefore it works”
- Automatically “git bisect” new failures





**Linaro
connect**
Vancouver 2018

My experiments with Remote Ikft lab

Sumit Semwal



What is Remote lkft lab?

- Allows using own devices to run tests and report to LKFT
- Uses Lava's distributed instance setup
- Chase Qi did amazing work in creating and documenting the remote lab setup
 - <https://github.com/Linaro/lkft-remote-lab>
 - Uses Intel laptop as dispatcher, RPi 3 + relays as Power and OTG control



Motivation? Arm on Arm

- I wanted to experiment to use ARM devices only
 - ARM as dispatcher
 - ARM as PDU control (using set of relays, controlled via GPIO)
 - ARM device under test



1st idea - RPi3 as both

- RPi 3 is already used as power control to control relays, why not use the same board as dispatcher?
- Issues:
 - Lava setup requires 64-bit OS; RPi default is 32-bit userspace
 - **Solution:** use 64-bit Debian Userspace - not straight forward
 - USB connections reset while toggling GPIOs at the same time
 - Before root-causing it, I got the 2nd idea, so ...



2nd idea - db410c

- Already has excellent debian support
- Lava-dispatcher prep straight-forward
- Issues:
 - GPIOs drive at 1.8V, so couldn't control 5v relays (Didn't know about the level-shifters in audio mezzanine ...)
 - **Solution:** Carbon with usb-relay
 - Thanks to Mani for suggestion, and Daniel Thompson for writing usb-relay
 - **Added bonus: 96boards-only solution!**
 - Slow: Just about Ok for one device, might not scale for more devices under control



3rd idea - Macchiatobin

- Decently powerful - quad core A72, 4GB Ram
- excellent mainline / debian story
- Some setup quirks, but manageable
 - UEFI not default, but needed to install debian



Misc Issues

- My relay-board ^[1] had inverted high and low signals
 - Inverted them in the targetctl shell script
- 'pulse' command from ocm3 didn't work well
 - Used 'on off' sequence for reboot in targetctl

[1]: https://www.amazon.in/gp/product/B0787G3W15/ref=oh_aui_search_detailpage?ie=UTF8&psc=1



Next steps

- Tested locally so far; need to connect with 'main' lkft server
 - Idea was to harden it locally, then add to the main server
- Blog post very soon!



Thank you!

