



**Linaro
connect**
San Francisco 2017

LMG Lightning Talks

LMG





**Linaro
connect**
San Francisco 2017

linaro-android kernel topic branch updates

Amit Pundir





Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

linaro-android kernel updates

- lsk-v3.18-android
 - Not actively maintained by LMG.
- lsk-v4.4-android
 - Weekly/Bi-weekly android-4.4 updates.
- linaro-android-llct
 - Rebased to v4.14-rc2 tracking android-4.9 patchset.
 - EAS patchset is dropped and is tracked separately by Google/Arm/Linaro developers.



v4.14-rc2..llct-v4.14-rc2 git diff stats

- 336 files changed, 31753 insertions(+), 526 deletions(-)
- ~18% Filesystem
 - ~15% sdcardfs
 - [LPC 2017: SDCardFS Upstreaming](#)
 - Misc fuse, epoll, pstore et.al changes, fidtrim, private anonymous memory tagging support ..
- ~16% Networking
 - ~15% netfilter (xt_qtaguid, idletimer)
 - [LPC 2017: Replacing xt_qtaguid with an upstream eBPF implementation](#)
 - Idletimer notifications from nflog
 - PPPoPNS/PPPoLAC
 - [Linaro Wiki: Make Android use upstream PPP VPN code](#)
 - Paranoid Networking ..



v4.14-rc2..llct-v4.14-rc2 git diff stats

- ~14% USB
 - ~12% usb gadget functions
 - [Coming Soon\(?\): MTP/PTP moving to FunctionFS](#)
 - Dual role sysfs class, OTG wakelock support ..
- ~10% Staging
 - ~6% FIQ debugger
 - ~4% Goldfish emulator changes
- ~7% Input
 - Keycombo/Keyreset, gpio input drivers
- ~6% Arch
 - Appended DTB support, cmdline extension support, ranchu defconfigs etc



v4.14-rc2..llct-v4.14-rc2 git diff stats

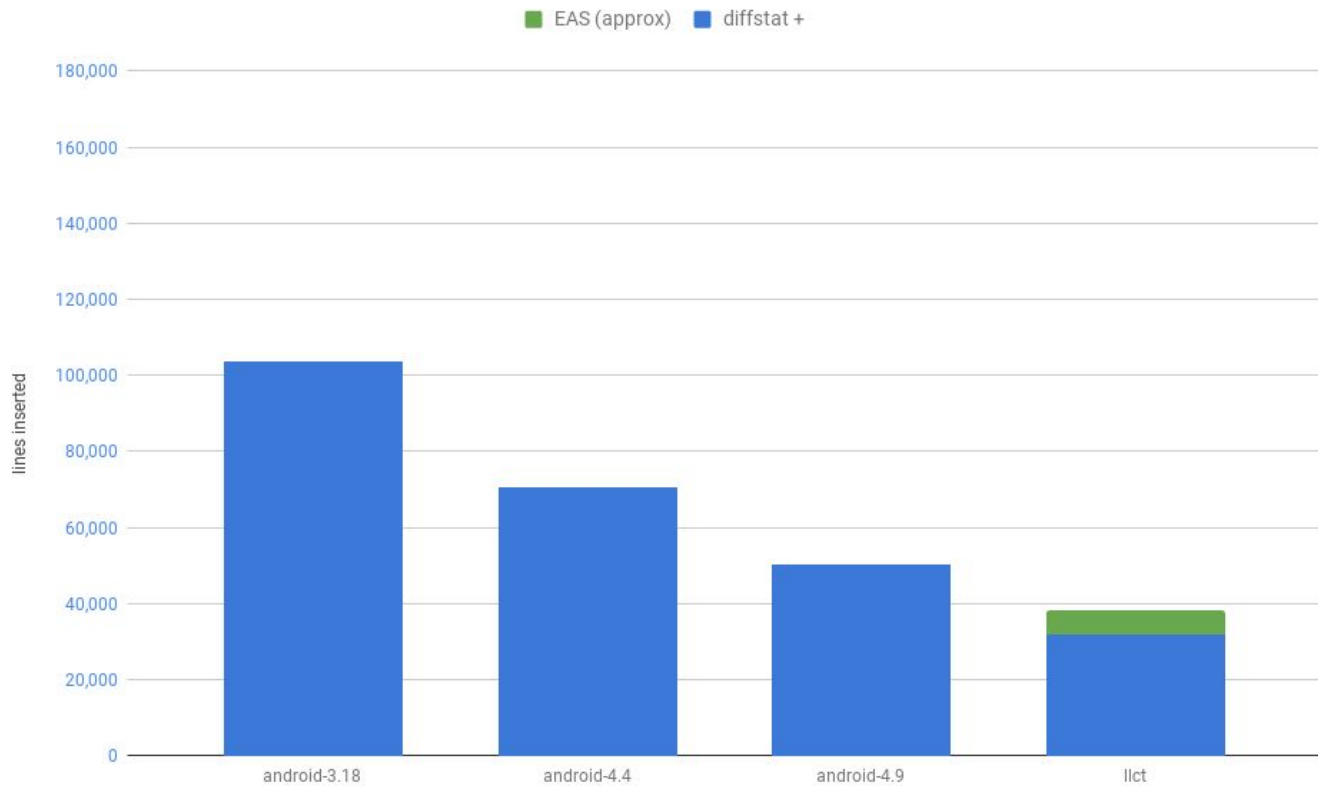
- ~4% Device Mapper
 - Dm-android-verity (A/B update support)
- ~3% UID sys stats
 - Per UID system statistics exported to /proc
- And other misc tracing, mmc-sdio, wakeup reason changes et.al.



android-4.9 vs llct-v4.14-rc2

- Lowmemorykiller dropped
- Sync reworked
- Interactive gov and scheduler idle notifiers dropped
- Bcmdhd wlan platform headers dropped
- W1 ds2482 patch, n/w bridge hack dropped
- Android config fragments dropped







**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Hibernate-Resume on android-4.4

- Member interest in hibernate-resume support
- arm64-hibernate kernel feature backported on linux-linaro-lsk-v4.4 kernel
- Rebased lsk-v4.4 patches to android-hikey-linaro-4.4
 - Hikey story
 - 1GB RAM not enough for Android
 - CPU lockups while trying to enable Mali clk
 - Hikey960 story
 - Missing Suspend/Resume support in Kirin960 DRM driver
 - Verified with VirtualFB display support and screenrecord



**Linaro
connect**
San Francisco 2017

AOSP build with upstream clang

MinSeong Kim



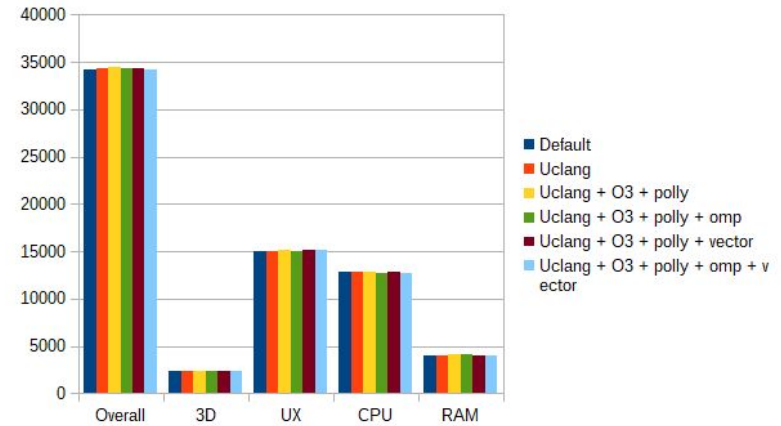
Upstream Clang for building AOSP

- Continuously watching upstream clang (trunk) for building AOSP master
 - Prebuilt clang in AOSP lag behind in time 3 to 6 months.
 - The latest Android clang version 5.0 (based on 4/12)
 - Upstream clang version 6.0.0 (now)
 - Purpose
 - Check clang performance on AOSP
 - Help clang to soft-land on AOSP
- AOSP and upstream clang at 9/13
 - No visible performance increase compared to prebuilt clang



Upstream Clang with more stuff for AOSP

- LLD (LLVM Linker: a drop-in replacement for GNU linkers)
 - Current default: gold for aarch64 build
 - Found a few issues (3 missing options and some compatibility issues)
- Souper (Peep-hole optimizer on LLVM IR)
 - Missed simplifications on the final LLVM IR.
 - Applied to bionic and benchmarked - no visible performance increase
- Polly (Loop and data-locality optimizer)
 - 3 major options to turn on its optimizations
 - Scalar, OpenMP, and Vectorizer
 - No visible performance increase (e.g Antutu 6.3)
- For a good candidate for application
 - Please contact minseong.kim@linaro.org
 - with new stuff and benchmark





**Linaro
connect**
San Francisco 2017

LCR- Linaro Confectionary Release

Yonqin Liu



Features with LCR

- Requirement based releases with latest stable android tag as reference
 - Kernels based on LSK or from AOSP
 - New features list, optimizations list
 - Update on deployment instructions, build scripts, etc
- Performance patches integrated
 - Optimizations on memory, boot time, bionic, powermanager, etc
 - External projects version update
- Security feature supported
 - SELinux, OP-TEE
- Experiment with new toolchains
 - Build with Gcc7.x, Gcc6.x
 - Build Kernel with Clang prebuilt, with Clang master



Features with LCR

- Tests to ensure quality
 - Normal boot test to ensure basic functions
 - Benchmark tests to check performance regression
 - CTS/MTS for compatibility regression check
 - Other topics tests like boot time, memory usage test
- Premerge ci for patch checking
 - Help to verify the same change on different platforms
 - Make sure the changes tested on LAVA with the same environment, which means the tests result is table and reproducible

Working on the Oreo based builds with LAVA V2 Test support now, more features are coming soon....



Potential improvement

- Upstream master version might have some good change
 - Using master version directly sometimes breaks stable version build
 - Could help to make master compatible with old android version
- Change to use one common kernel for all platforms
 - Linaro generic build
 - AOSP Treble project
- Making adding new platform and checking result more easily
 - Automated
 - Visualizable
- Reference android projects
 - Usb Camera enabling
 - OpenCV utilizing projects
 - Projects works with IOT(zephyr, Android Things, etc)
 - Etc.





**Linaro
connect**

San Francisco 2017

Building AOSP on aarch64 hosts

Bernhard “Bero” Rosenkränzer





**Linaro
connect**

San Francisco 2017

Why would we want to do this?

- Some aarch64 build machines are at least as powerful as their x86 counterparts - 96-core boxes exist
- Test environment for build boxes
- First step towards totally native build environment: Build Android (and Android apps) inside Android for an easier development environment

ENGINEERS
AND DEVICES
WORKING
TOGETHER





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Current status

- ✓ Compilers (both clang and gcc based toolchains) ready
 - ✓ Build tools (soong, ninja, acp, zipalign, ...) built for aarch64
 - ✓ Build tools with x86 hardcodes (soong) ported to detect CPU type
 - ✓ All related work ported to 8.0.0_r12 release
 - ✓ Easy to rebase patchset
 - ✗ Builds working
- Current problem: Generated ninja files differ from their counterparts created on x86 boxes for reasons yet to be found. “diff -u” shows thousands of lines of differences



Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Current status

- Most differences look irrelevant; some cause
error: external/llvm/tools/gold/Android.bp:20:1: module "LLVMgold" variant "linux_arm64_shared": depends on disabled module "libLLVM"
error: development/vndk/tools/vtable-dumper/Android.bp:17:1: module "vndk-vtable-dumper" variant "linux_arm64": depends on disabled module "libLLVM"
error: frameworks/compile/libbcc/tools/bcc_strip_attr/Android.bp:19:1: module "bcc_strip_attr" variant "linux_arm64": depends on disabled module "libLLVM"
error: external/llvm/tools/llvm-ar/Android.bp:5:1: module "llvm-ar" variant "linux_arm64": depends on disabled module "libLLVM"
error: bionic/tools/versioner/src/Android.bp:1:1: module "versioner" variant "linux_arm64": depends on disabled module "libclang"
error: bionic/tools/versioner/src/Android.bp:1:1: module "versioner" variant "linux_arm64": depends on disabled module "libLLVM"
error: external/spirv-llvm/lib/SPIRV/Android.bp:22:1: module "libSPIRV" variant "linux_arm64_shared": depends on disabled module "libLLVM"



Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Current status

- Working around that (“FORCE_BUILD_LLVM_COMPONENTS=true”) currently results in more errors - duplicated definitions etc.
- Probably there’s not much more missing - all the prebuilts are there and seem to work well
- But getting past this one problem can take some more time





**Linaro
connect**
San Francisco 2017

AOSP zstd integration

Satish Patel/Bernhard “Bero” Rosenkränzer





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Why would we want to do this?

- zstd is a new compression system that can give speeds comparable to lz4, while also providing file sizes comparable to LZMA/xz
- Given that, it should be able to speed up the boot process considerably if we put /system on a zstd compressed squashfs as opposed to an xz compressed squashfs filesystem (while not using the amount of disk space we'd have to use with uncompressed or LZ4 compressed squashfs or ext4)





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER



Status

- Zstd (de)compression added to the kernel (for kernel image and initramfs image): DONE for armv7hl (in armv8, decompression happens in the bootloader...)
- Zstd decompression added to squashfs
- Zstd compression added to squashfs userland utilities
- (Independently, done upstream) zstd compression added to btrfs

Speed on Hikey

	1st iteration of					
	First boot (sec)	Next 5 boot time avg	sq_find (sec)	sq_find (sec)	sq_read (sec)	avg. read using dd
squashfs - zstd (block API)	31.97	15.64	21.99	1.27	0.2	26445468 bytes/sec
squashfs - lz4	27.93	14.48	11.62	1.28	0.2	28482735 bytes/sec
squashfs - zstd (block API, Block size 256K, compression level 16)	49.97	14.54	17.52	1.17	0.2	29502580 byte/sec
squashfs - zstd (steam API)		14.97	16.79	1.18	0.2	



Speed on X15

	First boot (sec)	Next 5 boot time avg	1st iteration of sq_find (sec)	sq_find (sec)	sq_read (sec)	avg. read using dd	Kernel Size (MB)	system.img (MB)
squashfs - lz4	20.06	9.9646	10.25	0.97	0.3	27627146 bytes/sec	7.765342712	581.1102562
squashfs - zstd (block API, Block size 256K, compression level 16)	20.08	9.3656	8.72	0.91	0.32	27072135 byte/sec	7.468711853	271.0351944





Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Conclusions

- Certainly worth it -- system.img shrinking to less than half its previous size with slightly improved performance
- Should keep the patches in sync with future upstream zstd releases - both compression and speed are likely to get better
- Are there other places where we might be able to benefit from zstd?





**Linaro
connect**
San Francisco 2017

U-Boot upstreaming progress

Sam Protsenko





Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

1st iteration (page 1)

- Goal: initial support of fastboot and Android eMMC boot for AM57x EVM (X15) and DRA7 boards
- 1st iteration includes my merged patches for period from 2016-02-16 to 2017-02-13
- Total count: 23
- Complete list is here:

[http://git.denx.de/?p=u-boot.git&a=search&h=HEAD
&st=author&s=Protsenko](http://git.denx.de/?p=u-boot.git&a=search&h=HEAD&st=author&s=Protsenko)





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

1st iteration (page 2)

- Fastboot
 - Fix g_dnl: Correct recovery path for register
 - Add CONFIG_FASTBOOT_USB_DEV option
 - Respect USB controller number in fastboot
 - am57xx: Enable download gadget
 - am57xx: Enable fastboot
 - am57xx: Set serial# variable
- Android
 - Add Android partition table





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

1st iteration (page 3)

- Collateral patches
 - Move CONFIG_USB_GADGET to Kconfig
 - Move CONFIG_USB_GADGET_VBUS_DRAW to Kconfig
 - Move CONFIG_USB_GADGET_DUALSPEED to Kconfig
 - Move CONFIG_USB_GADGET_DOWNLOAD to Kconfig
 - Move CONFIG_USB_DWC3_GADGET/HOST to Kconfig
 - Move CONFIG_USB_DWC3_OMAP to Kconfig
 - Move CONFIG_USB_DWC3_PHY_* to Kconfig
 - Move CONFIG_G_DNL_* to Kconfig
 - Move CONFIG_SYS_MMCSL_RAW_MODE_U_BOOT_SECTOR to Kconfig
 - Remove CONFIG_SYS_U_BOOT_MAX_SIZE_SECTORS





Linaro
connect

San Francisco 2017

2nd iteration (page 1)

- Goal: complete fastboot support and Android eMMC boot for AM57x EVM (X15) board
- 2nd iteration includes my merged patches for period from 2017-05-18 to 2017-09-01
- Total count: 15
- Complete list is here:

[http://git.denx.de/?p=u-boot.git&a=search&h=HEAD
&st=author&s=Protsenko](http://git.denx.de/?p=u-boot.git&a=search&h=HEAD&st=author&s=Protsenko)

ENGINEERS
AND DEVICES
WORKING
TOGETHER





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

2nd iteration (page 2)

- **Fastboot**
 - Add “fastboot flash zImage” support
 - Setting fastboot variables (for fastboot.sh script)
 - Fix g_dnl: sync internal variable with serial# from env
- **Android**
 - Keep U-Boot environment on eMMC (rather than on SD card)
 - Add command for booting Android (from eMMC)
 - Add kernel cmdline for Android boot
 - Fix Android partition table
- **Collateral patches**
 - Refactoring: Unify get_device_type() function
 - Extract OMAP5 boot environment to separate file
 - Move CONFIG_BOOTARGS to Kconfig





**Linaro
connect**

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Pending patches

Patches for booting Linux from eMMC are still pending on ML:

- arm: am57xx: Fix Linux boot from eMMC
- arm: dra7xx: Fix Linux boot from eMMC
- env: ti: boot: Extract command for eMMC Linux boot
- env: ti: boot: Show boot status information
- omap: Omit fastboot.userdata_size related errors
- disk: Provide API to get partition by name for specific type
- omap: Fix warning when looking for userdata part





Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Upstreaming learnings (page 1)

- Sometimes you'll be asked to do some preceding (collateral) work first, e.g.:
 - Convert some options to Kconfig
 - Preceding refactoring
- You need to consider impact on other platforms:
 - Reverted: "ti_armv7_common.h: Fix U-Boot location on eMMC"
- Sometimes you'll notice some associated bug or shortcoming, and you need to step up and take an action (see technical debt)
- Even when doing some work for one particular platform, it often benefits others:
 - It usually involves generic code modifications as well
 - Can be used as template for others





Linaro
connect

San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Upstreaming learnings (page 2)

- You may come up with some useful tools along the way; be sure to share it, as it can be helpful for others
 - My scripts for semi-manual migrating U-Boot options to Kconfig:
<https://github.com/joe-skb7/uboot-convert-scripts>
 - My tool for tracing which #ifdef's your define is wrapped into:
<https://gitlab.com/joeskb7/find-ifdefs>
- Internalize commonly used practices and tools:
 - tools/buildman/buildman: useful for automatic building for selected patches and architecture
 - Use existing .travis.yml file to run remote build on Travis CI
 - tools/moveconfig.py: automatic migration of config option to Kconfig
- Get acquainted with maintainers
 - Discuss things early
 - Sometimes IRC is better than mail





**Linaro
connect**

San Francisco 2017

Further directions

- Android-O support
 - DTB overlays with FIT images
 - Android Verified Boot 2.0 (AVB) support
 - “vendor” partition
- Switching from ttyO to ttyS for OMAP platforms
(to be discussed with maintainers)

ENGINEERS
AND DEVICES
WORKING
TOGETHER





**Linaro
connect**
San Francisco 2017

Thank You

#SFO17

SFO17 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org

