

EAS Product Codeline Past/Present/Future

ARM

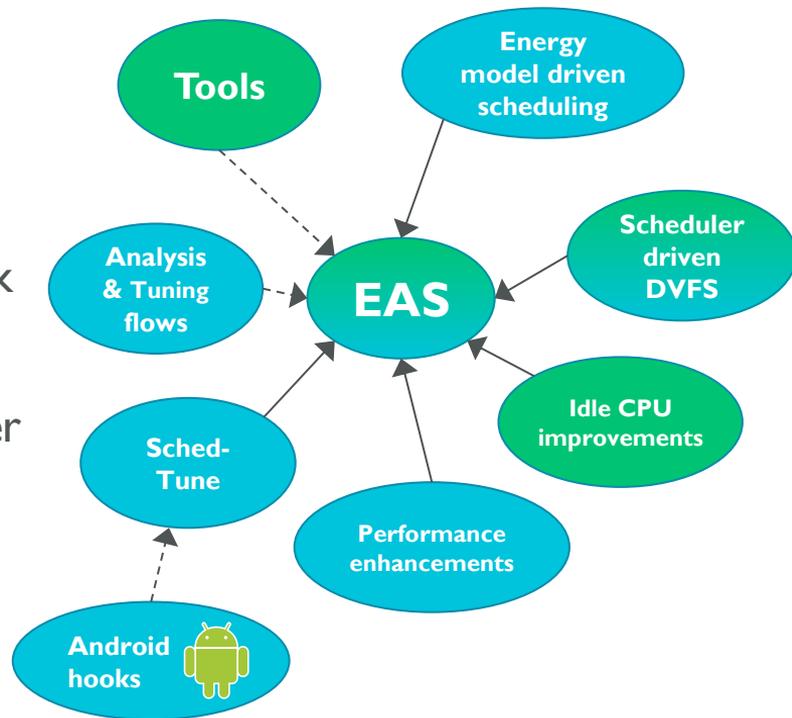
March-2017

Chris Redpath

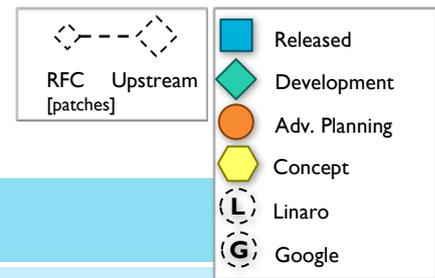
What is 'Product Codeline'?

Use **energy-awareness** in Linux to enable devices:

1. Integrate **Idle, DVFS**, scheduler **big.LITTLE** support with Android Middleware performing task classification
2. Based on measurable **energy model** data rather than magic tunables
3. Code and Tools integrated from many sources
4. Intended for Android products
5. Released in Android Common Kernel

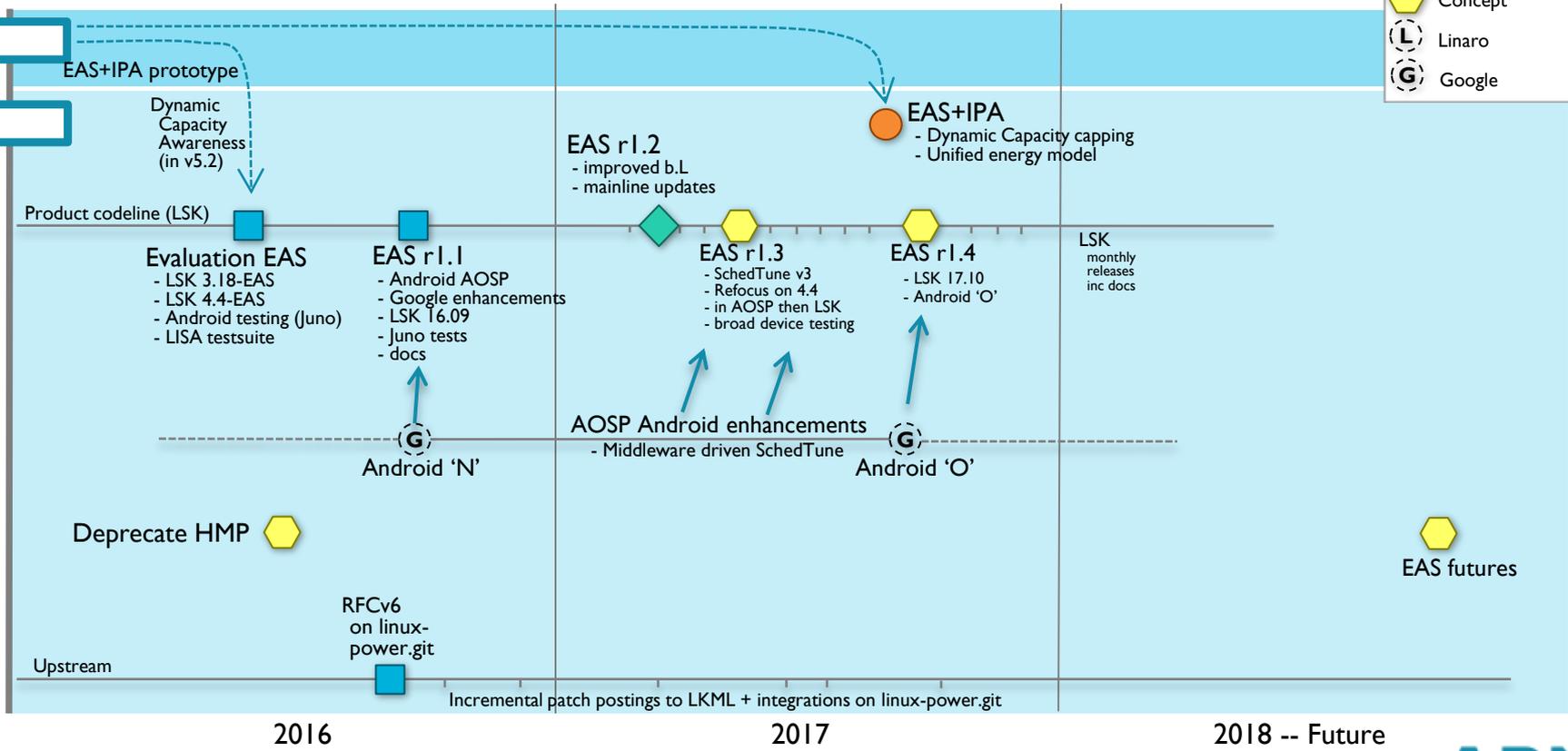


Power Software



IPA

EAS



Note: Upstream dates are subject to community acceptance and kernel version release schedule

EAS Origins (2014 – 2016)

- EAS first public appearance May 2014 as an RFC for mainline
 - <https://lwn.net/Articles/600135/>
- EAS released by ARMLT LSK
 - <https://git.linaro.org/landing-teams/working/arm/kernel-release.git/>
 - First tag was lsk-3.18-armlt-20151102-eas-test
- Derived from the RFCv5 posting
 - RFC v5 posted July 2015 <https://lkml.org/lkml/2015/7/7/754>
- EAS became the default LSK LT heterogeneous support replacing HMP in 16.09.

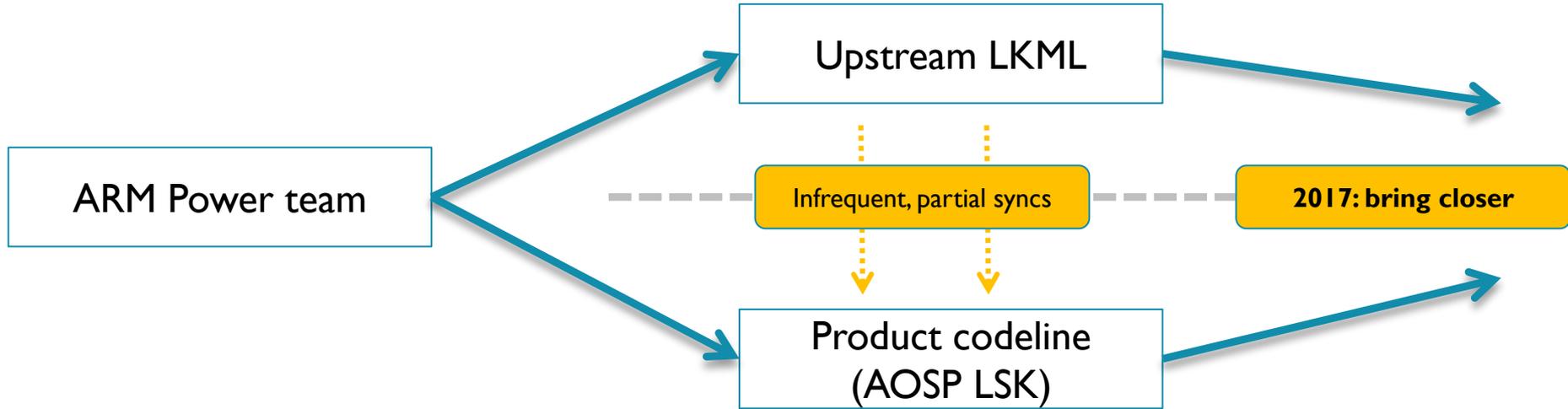
EAS v5.2 (2014/2015)

- LKML “RFCv5” => product versions (“v5.2+” with product features – not posted)
 - Evolving a full-featured 'product version' and a cleaner 'RFC version'
 - Based on RFCv5 with .2 addition used to indicate additional features.
 - <http://www.linaro.org/blog/core-dump/energy-aware-scheduling-eas-progress-update/>
 - Energy model & Energy-driven-Scheduling
 - Scheduler Idle state awareness
 - Sched-DVFS
 - Capacity & Frequency Invariance
 - Schedtune
 - Many tooling things (rt-app, workload automation, trappy, idlestat)
- Switch to ‘EAS **r1.x**’ terminology for full-featured product
 - EAS v5.2+ becomes EAS **r1.0**

EAS r1.1 (Aug-2016)

- Partnered with Google & Qualcomm to productize EAS r1.1 for Pixel devices
 - EAS r1.1 released in AOSP kernel/common for 3.18 and 4.4.
- Improvements
 - Significant upgrades to SchedTune
 - Some fixes/changes to sched-dvfs
 - Use of WALT for utilization/load signals
 - Alternative wakeup-cpu-selection code (find_best_target) optimised for low-latency on smp-like platforms (the 'is_big_little' sysctl)
 - Close integration with Android middleware for classifying tasks
 - Minimal latency preferred to maximal energy saving

Bring EAS closer to mainline



Divergence in mainline features from the product codeline thanks to:

- significant rate of change in the mainline scheduler
- EAS mainline postings kept simple for easier merging
- ChromeOS /Android significant inputs into product codeline during 2016

In 2017:

- **Develop EAS product codeline against AOSP (=> pulled into LSK-Android)**
- **Align upstream & product codeline**

EAS r1.2 (Mar-2017)

- An upstream catch-up release with some product compromises made more generic
 - Removed `is_big_little`
 - EAS restructured to be much closer to mainline version, but retain latency focus from EAS r1.1
 - Added backport of `schedutil` from 4.7
 - Incorporated upstream CFS fixes (**massive thanks to Vincent**)
 - Added upstream capacity-based-scheduling
 - Switchable WALT vs. PELT

Upstream vs. product

LKML Patches posted	linux-power.git on linux- arm.org	Product additions AOSP
Capacity Awareness (merged)	+EAS-core	WALT option
DT capacity arch topology driver (pending)	Debug & tracing inc PELT tracing	Schedutil hooks
SchedTune v3 Cgroup controller	Frequency/capacity invariance (next LKML posting)	Performance improvements
Schedutil fixes		CPU dynamic capacity capping (thermal)
		DT Energy Model

EAS r1.2 (2017)

- “Pre-alpha” experimental branch made available for Pixel (3.18 kernel)
 - <https://android.googlesource.com/kernel/msm.git/+android-msm-marlin-3.18-nougat-mr1-eas-experimental>
 - Using a commercially-available platform with optimised user space available for evaluation
 - Currently evaluating performance, looking at traces & identifying upstream commits to cherry-pick
 - Pixel will be the only 3.18 source for evaluation
(we don't expect to support AOSP kernel-3.18, moving to AOSP kernel-4.4 and 4.9)

EAS 1.2 overview

97 patches & counting

Patch stack walkthrough

- **0001-0024** - patches from eas_r1.1 which went into ACK that did not get merged back into the msm kernel

```
0001-ANDROID-sched-tune-__pcpu_scope_cpu_boost_groups-can.patch
0002-ANDROID-sched-tune-schedtune_allow_attach-can-be-sta.patch
0003-sched-tune-Fix-lacking-spinlock-initialization.patch
0004-Revert-WIP-UTIL_EST-use-estimated-utilization-on-loa.patch
0005-Revert-WIP-UTIL_EST-use-estimated-utilization-on-ene.patch
0006-Revert-WIP-UTIL_EST-sched-fair-use-estimated-utiliza.patch
0007-Revert-WIP-UTIL_EST-switch-to-usage-of-tasks-s-estim.patch
0008-sched-revert-UTIL_EST-usage-from-commit-6bf72ca7f1.patch
0009-Revert-WIP-UTIL_EST-sched-core-fair-add-support-to-u.patch
0010-Revert-WIP-UTIL_EST-sched-fair-add-support-for-estim.patch
0011-FIXUP-sched-tune-update-accounting-before-CPU-capacit.patch
0012-FIX-sched-tune-move-schedtune_normalize_energy-into-.patch
0013-eas-sched-fair-Fixing-comments-in-find_best_target.patch
0014-sched-fair-missing-parts-of-optimize-idle-cpu-select.patch
0015-sched-fair-Fix-uninitialised-variable-in-idle_balanc.patch
0016-Revert-UTIL_EST-code-from-fix-set_cfs_cpu_capacity-w.patch
0017-Unify-whitespace-layout-with-android-3.18.patch
0018-schedtune-Guarding-against-compile-errors.patch
0019-sched-walt-use-do_div-instead-of-division-operator.patch
0020-sched-Fix-sysctl_sched_cfs_boost-type-to-be-int.patch
0021-sched-tune-backport-fix-accounting-for-runnable-task.patch
0022-sched-walt-Drop-arch-specific-timer-access.patch
0023-Revert-DEBUG-UTIL_EST-sched-update-tracepoint-to-rep.patch
0024-sched-This-kernel-expects-sched_cfs_boost-to-be-sign.patch
```

- **0025-0035** - backport schedutil to 3.18 and enable for marlin

```
0025-sched-cpufreq-fix-tunables-for-schedfreq-governor.patch
0026-kthread-allow-to-cancel-kthread-work.patch
0027-sched-backport-cpufreq-hooks-from-4.9-rc4.patch
0028-sched-backport-schedutil-governor-from-4.9-rc4.patch
0029-cpufreq-schedutil-move-slow-path-from-workqueue-to-S.patch
0030-sched-cpufreq-use-rt_avg-as-estimate-of-required-RT-.patch
0031-sched-cpufreq-make-schedutil-use-WALT-signal.patch
0032-trace-sched-add-rq-utilization-signal-for-WALT.patch
0033-cpufreq-schedutil-add-up-down-frequency-transition-r.patch
0034-schedutil-Fix-linkage-of-schedutil-and-walt.patch
0035-config-Update-marlin_defconfig-to-include-schedutil-.patch
```

- **0036-0052** - backport mainline capacity awareness

0036-Revert-WIP-sched-Consider-spare-cpu-capacity-at-task.patch
0037-Partial-Revert-WIP-sched-Add-cpu-capacity-awareness.patch
0038-sched-core-Fix-power-to-capacity-renaming-in-comment.patch
0039-sched-fair-Make-the-use-of-prev_cpu-consistent-in-th.patch
0040-sched-fair-Optimize-find_idlest_cpu-when-there-is-no.patch
0041-sched-core-Remove-unnecessary-NULL-pointer-check.patch
0042-sched-core-Introduce-SD_ASYM_CPUCAPACITY-sched_domai.patch
0043-sched-core-Pass-child-domain-into-sd_init.patch
0044-sched-core-Enable-SD_BALANCE_WAKE-for-asymmetric-cap.patch
0045-sched-fair-Let-asymmetric-CPU-configurations-balance.patch
0046-sched-fair-Compute-task-cpu-utilization-at-wake-up-c.patch
0047-sched-fair-Consider-spare-capacity-in-find_idlest_gr.patch
0048-sched-fair-Add-per-CPU-min-capacity-to-sched_group_c.patch
0049-sched-fair-Avoid-pulling-tasks-from-non-overloaded-h.patch
0050-sched-fair-Fix-incorrect-comment-for-capacity_margin.patch
0051-Experimental-arm64-Set-SD_SHARE_CAP_STATES-sched_dom.patch
0052-Experimental-sched-fair-Do-not-force-want_affine-eq.patch

- **0036-0052** - backport mainline capacity awareness

0036-Revert-WIP-sched-Consider-spare-cpu-capacity-at-task.patch
0037-Partial-Revert-WIP-sched-Add-cpu-capacity-awareness.patch
0038-sched-core-Fix-power-to-capacity-renaming-in-comment.patch
0039-sched-fair-Make-the-use-of-prev_cpu-consistent-in-th.patch
0040-sched-fair-Optimize-find_idlest_cpu-when-there-is-no.patch
0041-sched-core-Remove-unnecessary-NULL-pointer-check.patch
0042-sched-core-Introduce-SD_ASYM_CPUCAPACITY-sched_domai.patch
0043-sched-core-Pass-child-domain-into-sd_init.patch
0044-sched-core-Enable-SD_BALANCE_WAKE-for-asymmetric-cap.patch
0045-sched-fair-Let-asymmetric-CPU-configurations-balance.patch
0046-sched-fair-Compute-task-cpu-utilization-at-wake-up-c.patch
0047-sched-fair-Consider-spare-capacity-in-find_idlest_gr.patch
0048-sched-fair-Add-per-CPU-min-capacity-to-sched_group_c.patch
0049-sched-fair-Avoid-pulling-tasks-from-non-overloaded-h.patch
0050-sched-fair-Fix-incorrect-comment-for-capacity_margin.patch
0051-Experimental-arm64-Set-SD_SHARE_CAP_STATES-sched_dom.patch
0052-Experimental-sched-fair-Do-not-force-want_affine-eq.patch

- **0053-0060** - merge mainline-focussed wakeup path into EAS r1.1

0053-Experimental-sched-fair-Decommission-energy_aware_wa.patch
0054-Experimental-sched-fair-Add-energy_diff-dead-zone-ma.patch
0055-Experimental-sched-fair-Energy-aware-wake-up-task-pl.patch
0056-Fixup-sched-fair.c-Set-SchedTune-specific-struct-ene.patch
0057-Experimental-EAS-sched-fair-Re-integrate-honor-sync-.patch
0058-Experimental-sched-fair-Code-is_big_little-path-into.patch
0059-Experimental-sched-Remove-sysctl_sched_is_big_little.patch
0060-sched-core-Remove-remnants-of-commit-fd5c98da1a42.patch

- **0061-0069** - remove is_big_little and refactor EAS r1.1 wakeup path to be topology-agnostic

0061-Experimental-sched-core-Add-first-cpu-w-max-min-orig.patch
0062-Experimental-sched-fair-Change-cpu-iteration-order-i.patch
0063-sched-fair-Simplify-backup_capacity-handling-in-find.patch
0064-Fixup-sched-fair-Simplify-target_util-handling-in-fi.patch
0065-Fixup-sched-fair-Simplify-idle_idx-handling-in-find_.patch
0066-Fixup-sched-fair-Refactor-min_util-new_util-in-find_.patch
0067-Fixup-sched-fair-Simplify-idle_idx-handling-in-selec.patch
0068-Fixup-Return-first-idle-cpu-for-prefer_idle-task-imm.patch
0069-Fixup-sched-fair-No-need-to-and-current-cpu-w-online.patch

- **0070-0079** - backport upstream CFS fixes judged important

0070-sched-fair-Initiate-a-new-task-s-util-avg-to-a-bound.patch
0071-sched-fair-Apply-more-PELT-fixes.patch
0072-sched-fair-Improve-PELT-stuff-some-more.patch
0073-sched-factorize-attach-entity.patch
0074-sched-factorize-PELT-update.patch
0075-sched-fix-hierarchical-order-in-rq-leaf_cfs_rq_list.patch
0076-sched-propagate-load-during-synchronous-attach-detac.patch
0077-sched-propagate-asynchronous-detach.patch
0078-sched-fair-Fix-effective_load-to-consistently-use-sm.patch
0079-sched-Multiple-upstream-load-tracking-changes.patch

- **0080-0097** - various minor modifications and tweaks to some signal handling and configuration. *some of these are not tested properly yet*

```
0080-sched-EAS-single-cpu-per-cluster-cpu-hotplug-interop.patch
0081-sched-walt-kill-min-max-_capacity.patch
0082-ANDROID-sched-walt-fix-build-failure-if-FAIR_GROUP_S.patch
0083-Revert-cgroup-Fix-issues-in-allow_attach-callback.patch
0084-DEBUG-sched-fair-Fix-missing-sched_load_avg_cpu-even.patch
0085-DEBUG-sched-fair-Fix-sched_load_avg_cpu-events-for-t.patch
0086-sched-core-Fix-PELT-jump-to-max-OPP-upon-util-increa.patch
0087-config-marlin-Turn-off-WALT-by-default.patch
0088-Experimental-sched-fair-Add-eas-cas-specific-rq-sd-a.patch
0089-trace-sched-Make-util_avg-in-load_avg-trace-reflect-.patch
0090-sched-fair-remove-task-util-from-own-cpu-when-placin.patch
0091-sched-fair-use-correct-signal-in-cpu_util_wake.patch
0092-cpufreq-schedutil-Fix-schedutil-s-default-governor-m.patch
0093-Revert-config-marlin-Turn-off-WALT-by-default.patch
0094-sched-walt-Add-CONFIG_USE_WALT-to-change-default-usa.patch
0095-EXPERIMENTAL-sched-walt-set-walt_disabled-flag-accor.patch
0096-sched-fair-avoid-subtracting-unrelated-signals-when-.patch
0097-EXPERIMENTAL-sched-fair-ensure-signals-are-synchroni.patch
```

EAS r1.3 (2Q2017)

- Will mostly contain fixes and tweaks to EAS r1.2 (to be identified)
 - Schedutil still a little less nicely-behaved than sched-dvfs for Android, but not much
 - WALT may be removed, depending upon energy & performance of PELT
 - Backport of CFS fixes has made PELT behave much better
 - Updates to Schedtune
 - Schedtune v3 update, redesigned following upstream feedback [if possible, otherwise EAS 1.4]
 - <https://lkml.org/lkml/2017/2/28/355>
 - Looks possible to include both in one kernel so long as we don't use both at once.
 - SCHED_DEADLINE updates
 - Mainline development will be included where necessary

Summary of 2017 Product Codeline Changes



- Planning Android Common Kernel updates throughout 2017
 - Support 4.4 and 4.9 (future). Drop support for 3.18

- Changes have two priorities:
 1. Get closer to mainline
 2. Improve Android

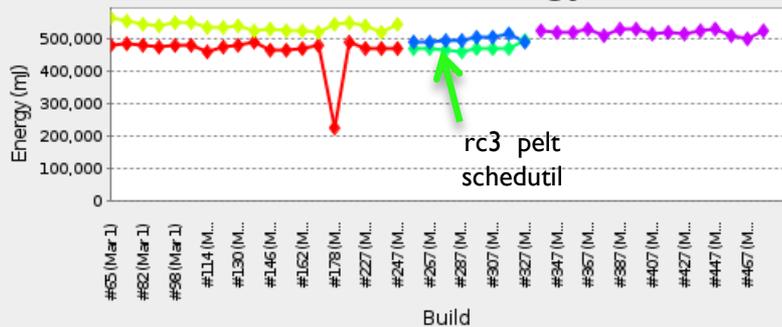
- Work in open to allow contribution

- Implemented CI and auto testing

LSK – EAS strategy: evolving product towards mainline

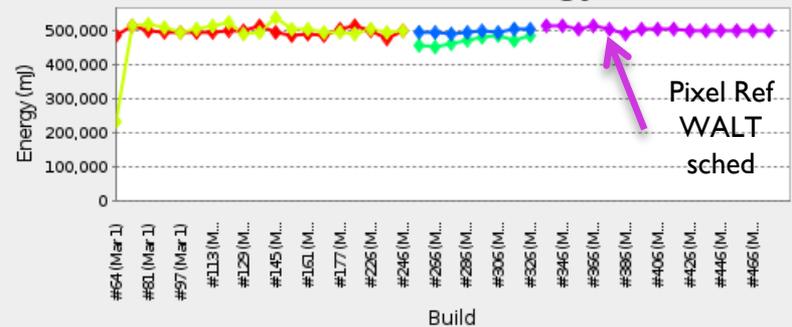
- Keep LSK close to Android Common Kernel but use LSK to evolve improvements
 - Use LSK topic branches to allow evaluation of new features, and evolution to upstream
 - Linaro & others are able to contribute
 - When changes sufficiently well-developed, push to Android Common Kernel
- Technical detail:
 - Goal to make PELT/util-est equivalent to WALT, need upstream load tracking solution... Create util-est experimental branch for LSK-3.18 (to compare with WALT)
[note – track PELT changes happening in mainline, to keep util-est relevant to upstream]
 - find_best_target improvements or make other scheduling classes more energy aware (DEADLINE improvements at choosing best CPU)
 - Schedutil replacing interactive (put sched-tune on top of schedutil)...
 - Userspace hints to SchedTune...

Vellamo - PELT - Energy



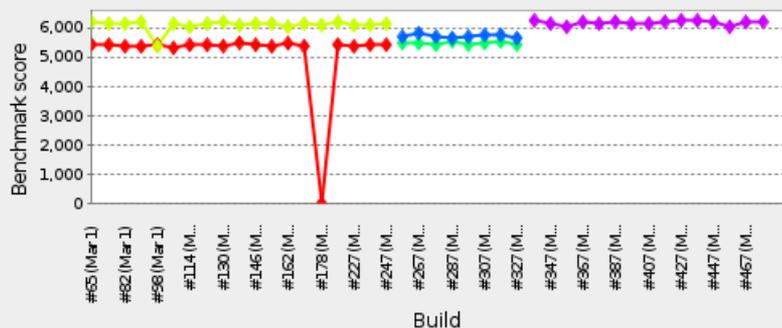
- ◆ eas-r1.2.0_rc2_pelt_schedutil
- ◆ eas-r1.2.0_rc2_pelt_sched
- ◆ eas-r1.2.0_rc3_pelt_schedutil
- ◆ eas-r1.2.0_rc3_pelt_sched
- ◆ android-7.1.1_r0.20_pelt_sched

Vellamo - WALT - Energy



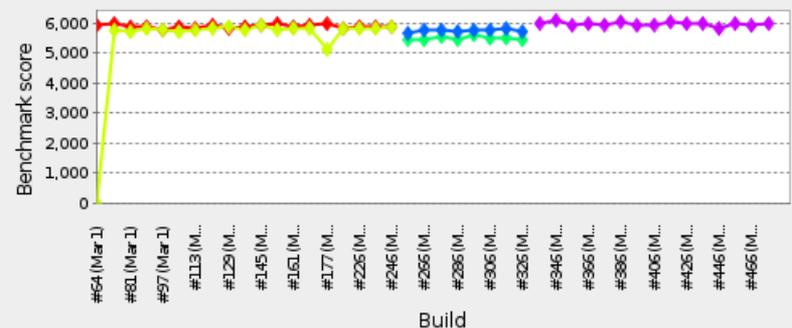
- ◆ eas-r1.2.0_rc2_walt_schedutil
- ◆ eas-r1.2.0_rc2_walt_sched
- ◆ eas-r1.2.0_rc3_walt_schedutil
- ◆ eas-r1.2.0_rc3_walt_sched
- ◆ android-7.1.1_r0.20_walt_sched

Vellamo - PELT - Performance



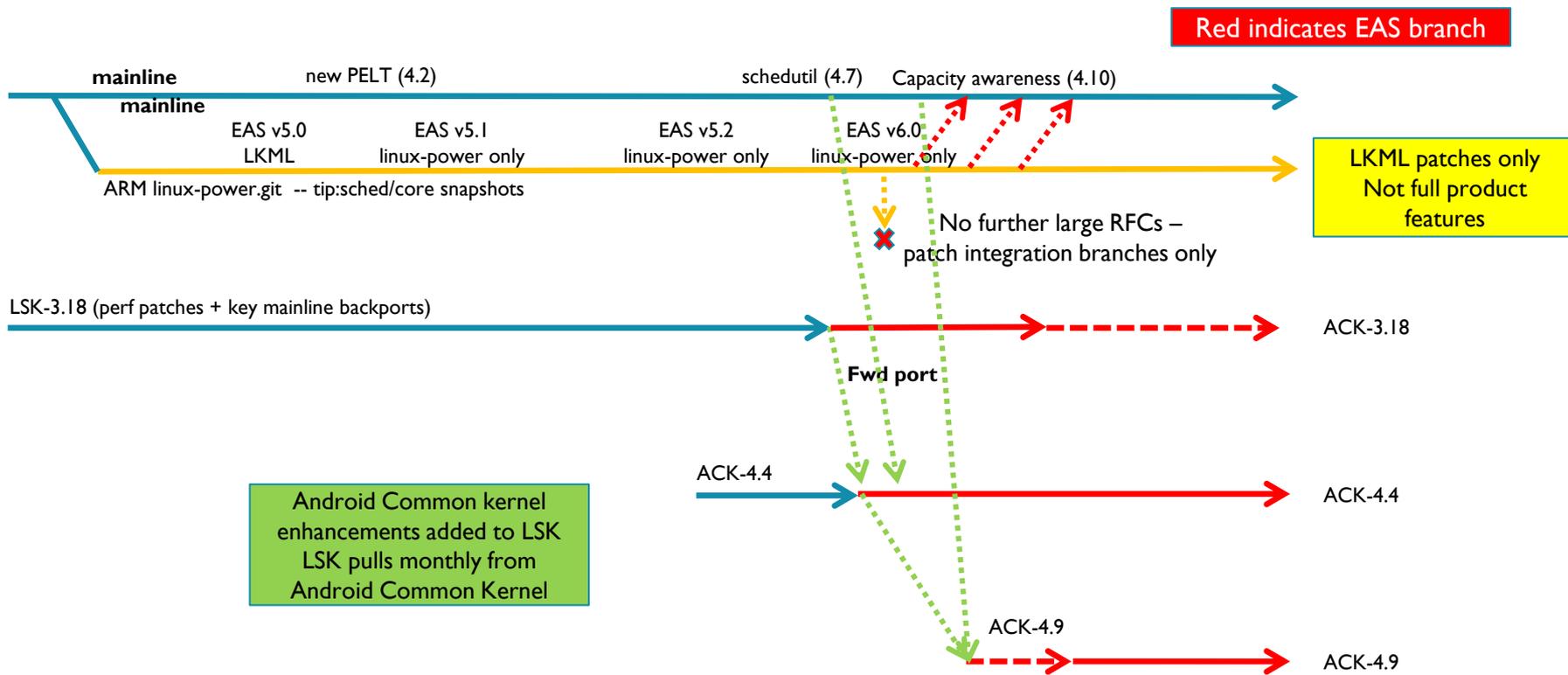
- ◆ eas-r1.2.0_rc2_pelt_schedutil
- ◆ eas-r1.2.0_rc2_pelt_sched
- ◆ eas-r1.2.0_rc3_pelt_schedutil
- ◆ eas-r1.2.0_rc3_pelt_sched
- ◆ android-7.1.1_r0.20_pelt_sched

Vellamo - WALT - Performance



- ◆ eas-r1.2.0_rc2_walt_schedutil
- ◆ eas-r1.2.0_rc2_walt_sched
- ◆ eas-r1.2.0_rc3_walt_schedutil
- ◆ eas-r1.2.0_rc3_walt_sched
- ◆ android-7.1.1_r0.20_walt_sched

Product codeline strategy



Thank you

ARM

The trademarks featured in this presentation are registered and/or unregistered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

All other marks featured may be trademarks of their respective owners.

Copyright © 2017 ARM Limited