Open Source Software

"EAS for Android" & mainlining update

Linaro Connect
Vancouver 2018

Dietmar Eggemann
20-September-2018
EAS for Android - updated development process

Previous EAS development

- Quarterly releases r1.3, r1.4, r1.5 etc.
- EAS product codeline in Android Common Kernel (ACK)
- Development ran behind LTS kernel

2018- EAS for Android

- Yearly major release aligned with LTS
- Some EAS building blocks in mainline kernel
- **Minimal delta** from mainline kernel (including EAS building blocks on tip/sched/core and LKML)
- Arm's development/testing process closely aligned with Linux mainline
EAS for Android - improved release process

Much closer alignment with mainline Linux

• Goal: earliest creation of Android Common Kernel (ACK) android-X.XX kernel version containing product-quality EAS

• **Single release per year** for ACK

• Longterm goal: minimize differences between LTS kernel an ACK

• Further EAS updates to each android-X.XX release are minimal => easy to catch up

• Development on `android.googlesource.com/kernel/common.git experimental/eas-dev`, tracks latest mainline kernel closely

Fully open workflow – ACK patches discussed on LKML and eas-dev mailing list
EAS for Android - improved release process

Arm mainline integration: [test platforms: Hikey960, Arm Juno r0, Arm Juno r2, Arm TC2]

+EAS core
mainline

rc6
4.18
rc6
4.19
rc6
4.20

+EAS core
+product EAS

experimental/eas-dev
[test platform: Hikey960]

android-mainline-tracking
[testing: Linaro/Google AOSP CI]

MAJOR ANDROID RELEASE
Minor eas updates for 4 months
Bugfixes for 3 years

android-4.19
<table>
<thead>
<tr>
<th>EAS Building Block</th>
<th>Status HKG18</th>
<th>Status YVR18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency and CPU Invariant Engines (FIE/CIE)</td>
<td>merged in v4.15</td>
<td>merged in v4.15</td>
</tr>
<tr>
<td>Idle CPU PELT update (Remote status update) (Linaro)</td>
<td>tip/sched/core</td>
<td>merged in v4.17</td>
</tr>
<tr>
<td>Util Est</td>
<td>tip/sched/core</td>
<td>merged in v4.17</td>
</tr>
<tr>
<td>Misfit Task</td>
<td>v2 on LKML</td>
<td>tip/sched/core (v4.20)</td>
</tr>
<tr>
<td>Dynamic Topology Flag Detection</td>
<td>in development</td>
<td>tip/sched/core (v4.20)</td>
</tr>
<tr>
<td>Energy Aware Scheduling (Energy Model)</td>
<td>v1 on LKML</td>
<td>v7 on LKML (possibly v4.21)</td>
</tr>
<tr>
<td>Util Clamping</td>
<td>v1 on LKML</td>
<td>v4 on LKML</td>
</tr>
</tbody>
</table>
EAS Update – Upstream Linux Kernel – Building Blocks

• Not all features got upstream in LTS – but all have made excellent progress
  • Note: Expected LTS was 4.20 but 4.19 was late and 4.20 is now too far into Dec 30 to be an LTS
  • Misfit task & Dynamic Topology Flag Detection are queued in tip/sched/core
  • Energy Aware Scheduling (Energy Model) (v7) is still being discussed on LKML – positive feedback from Rafael Wysocki

• Will get better idea of merge time for Util Clamping (v4) at LPC 18
  • Previous consensus on design is holding, implementation and review ongoing on LKML
EAS Update – Upstream Linux Kernel – Additional Topics

• Further reduce the delta between mainline and Android Common Kernel
  • Example: Per scheduler class utilization signal
    – [PATCH v7 00/11] **track CPU utilization** (Vincent Guittot, Linaro) (v4.19)
  • Example: One (running) task per CPU
    – [PATCH] sched/fair: **fix 1 task per CPU** (Vincent Guittot, Linaro) (LKML)
  • Max. Frequency Capping -> Introduce Thermal Pressure
    – [PATCH 0/6] **Introduce Thermal Pressure** (Thara Gopinath, Linaro)
EAS Update – Upstream Linux Kernel – Integration & Testing

• Biweekly EAS integration branch
  • Based on git://git.kernel.org/pub/scm/linux/kernel/git/tip/tip.git sched/core
  • Platform support for Juno r0, r2, Hikey960 (64bit) and TC2 (32bit)
  • Published at https://developer.arm.com/open-source/energy-aware-scheduling/eas-mainline-development

• Biweekly Regression testing on all supported platforms
  • Based on Lisa Test Framework
    1. EAS task placement and latency tests (Generic)
    2. PELT Load Tracking tests (Load-Tracking)
    3. Asymmetric CPU capacity load balancing tests (Misfit)
  • Published at https://developer.arm.com/open-source/energy-aware-scheduling/eas-mainline-development (Juno r0)
EAS Update – Upstream Linux Kernel – Integration & Testing


1. Latest source code
2. Current integration changes
3. EAS main features
4. Latest integration test results from Juno r0
EAS in android-4.14

android-specific

- find_best_target()
- Use of idle states

Upstream-targeted

- Dynamic Topology Flag Detection
- Frequency and CPU Invariant Engines
- Idle CPU PELT update (Remote status update)

- Util Est
- Misfit Task
- Energy Aware Scheduling (Energy Model)
EAS for android-4.19

android-specific
- find_best_target()
- Load balance tweaks

Sync Wakeups
- Trace & Debug
- Max. Freq. Capping
- Use of idle states

Upstream-targeted
- Dynamic Topology Flag Detection (v4.20)
- Misfit Task (v4.20)
- Energy Aware Scheduling (Energy Model) (v4.21 ?)

Upstream-merged
- Util Est (v4.17)
- Frequency and CPU Invariant Engines (v4.15)
- Idle CPU PELT update (Remote status update) (v4.17)
- Track CPU utilization (Rt-PELT) (v4.19)

Util Clamping
Schedtune

Dynamic Topology Flag Detection (v4.20)
Misfit Task (v4.20)
Energy Aware Scheduling (Energy Model) (v4.21 ?)
Util Est (v4.17)
Frequency and CPU Invariant Engines (v4.15)
Idle CPU PELT update (Remote status update) (v4.17)
Track CPU utilization (Rt-PELT) (v4.19)
# EAS – product code size reduction v4.14 -> v4.19

<table>
<thead>
<tr>
<th>EAS Building Block</th>
<th>android-4.14</th>
<th>android-4.19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency and CPU Invariant Engines (FIE/CIE)</td>
<td>v4.15</td>
<td>+ 1053 / - 15</td>
</tr>
<tr>
<td>Idle CPU PELT update (Remote status update) (Linaro)</td>
<td>v4.17</td>
<td>+ 677 / - 447</td>
</tr>
<tr>
<td>Util Est</td>
<td>v4.17</td>
<td>+ 265 / - 20</td>
</tr>
<tr>
<td>Misfit Task</td>
<td>v4.20</td>
<td>+ 168 / - 32</td>
</tr>
<tr>
<td>Dynamic Topology Flag Detection</td>
<td>v4.20</td>
<td>+ 78 / - 9</td>
</tr>
<tr>
<td>Energy Aware Scheduling (Energy Model)</td>
<td>v4.21 (?)</td>
<td>+ 1149 / - 80</td>
</tr>
<tr>
<td>(Schedtune /) Util Clamping</td>
<td>?</td>
<td>+ 1688 / -14</td>
</tr>
<tr>
<td>Track CPU utilization (Rt-PELT) (Linaro)</td>
<td>v4.19</td>
<td>+ 635 / - 429</td>
</tr>
</tbody>
</table>
## EAS in AOSP (HKG18 Recap)

### EAS Code Size by Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Android-specific</th>
<th>WALT</th>
<th>Upstreamable Features</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>3797</td>
<td>1470</td>
<td>2785</td>
<td>1153</td>
</tr>
</tbody>
</table>

### Target EAS Code Size by Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android-specific</td>
<td>824</td>
</tr>
<tr>
<td>WALT</td>
<td>0</td>
</tr>
<tr>
<td>Upstreamable Features</td>
<td>2785</td>
</tr>
<tr>
<td>Documentation</td>
<td>1153</td>
</tr>
</tbody>
</table>
Progress since HKG2018

![Bar chart showing progress since HKG2018 with categories: Android-specific, Schedtune, WALT, Upstreamable-Features, Documentation.](chart.png)
EAS Size Targets

ANDROID-SPECIFIC EAS CODE SIZE

EAS Code size in android-4.14: 100%
EAS Code size in android-4.19: 40%
Android Specific Code Target (rough estimate if everything goes to plan, mid-2019): 15%
Thanks.

Any questions?
The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks