EAS Update

ARM

September 2015

Amit Kucheria – Linaro Technical lead – Power Management
Robin Randhawa – ARM Powersoft Tech lead
Ian Rickards – ARM Powersoft Product Manager

Motivation

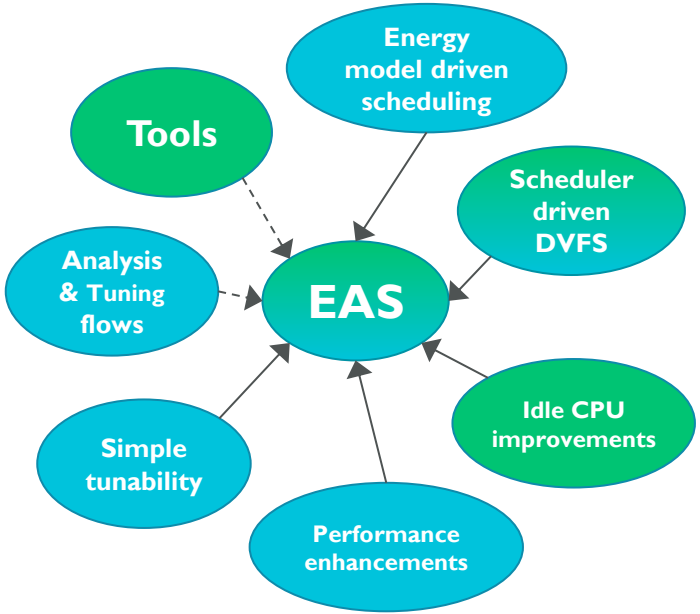
- Hardware topologies are becoming more varied, accommodating different power/performance budgets:
  - SMP, multi-cluster SMP, ARM big.LITTLE technology.
  - Per core/per cluster DVFS (Dynamic Voltage & Frequency Scaling)
- Linux power management frameworks are uncoordinated and hard to tune for different topologies.
- **We need a common upstream solution to minimize software costs.**

All policy, all metrics, all averaging should happen at the scheduler power saving level, in a single place, and then the scheduler should directly drive the new low level idle state driver mechanism.

Goals

Introduce generic energy-awareness in upstream Linux:

1. Integrate Idle, DVFS, scheduler big.LITTLE support
2. Clean design rather than short-cuts.
3. Based on measureable energy model data rather than magic tunables.
4. Support future CPU topologies
5. Maintained in upstream Linux, reduced software maintenance costs.



Linaro

Linaro development

ARM development

ARM

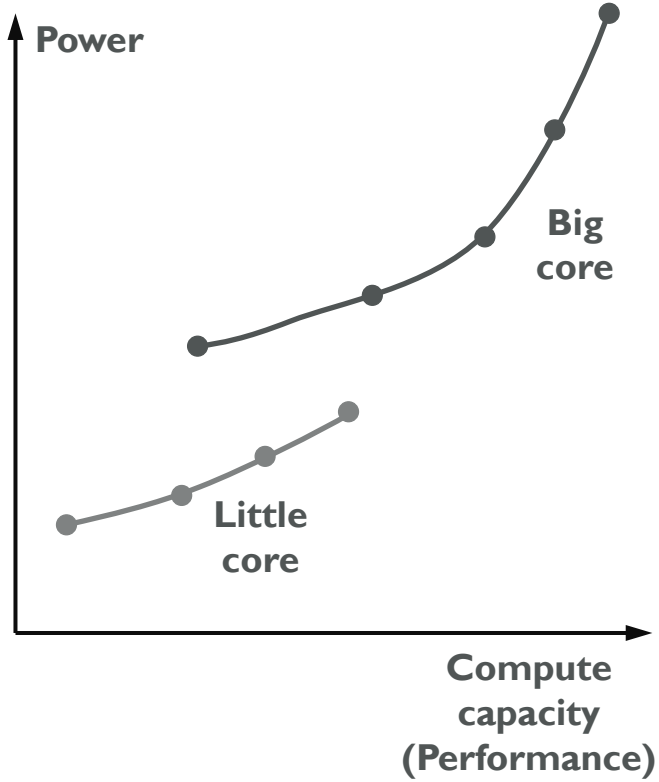
Power Fundamentals

Static Power

- Area of silicon (mm2)
- Threshold voltage (Vt)
  - "Low Vt" implementation faster but more leaky
  - "High Vt" implementation slower
- Temperature

Dynamic Power

- Toggling nodes x capacitance x voltage



ARM

OS task scheduling – throughput policy

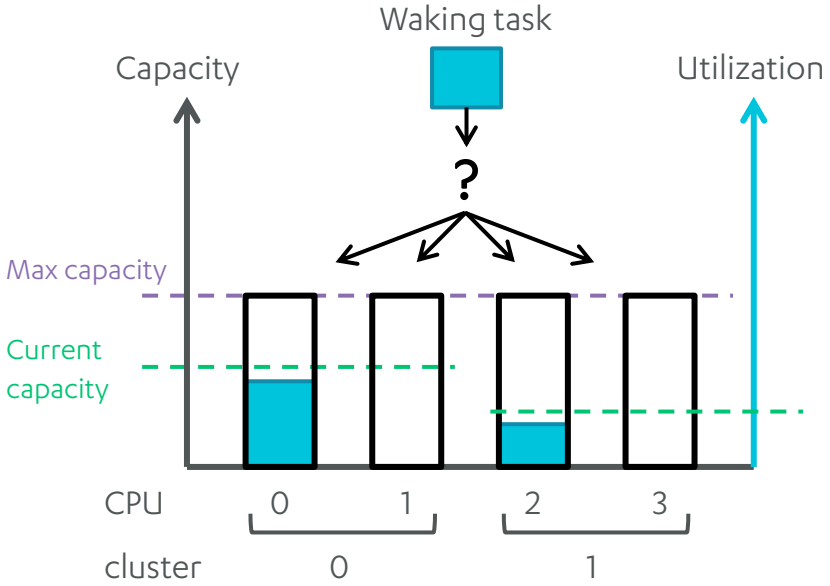
Scheduling policy decides task placement

- Affect performance and energy consumption.

Mainline Linux policy is 'work preserving'

- Considers only maximizing throughput.
- DVFS and idle-states controlled by independent policy governors.

Designed for SMP, not energy-aware



ARM

OS task scheduling – energy-aware policy

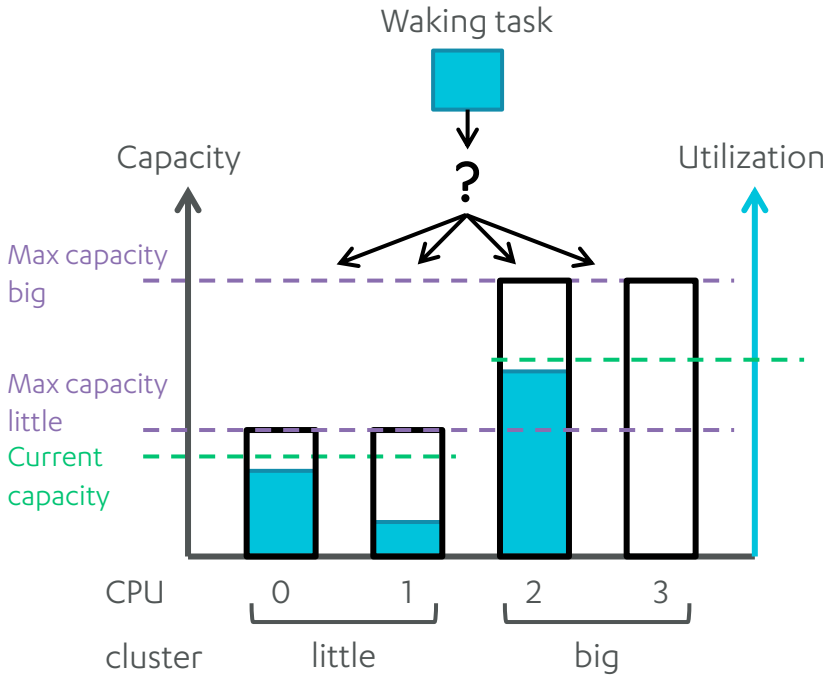
Energy-Aware Scheduling (EAS) policy:

- Pick CPU with sufficient spare capacity and smallest energy impact.

Requirements:

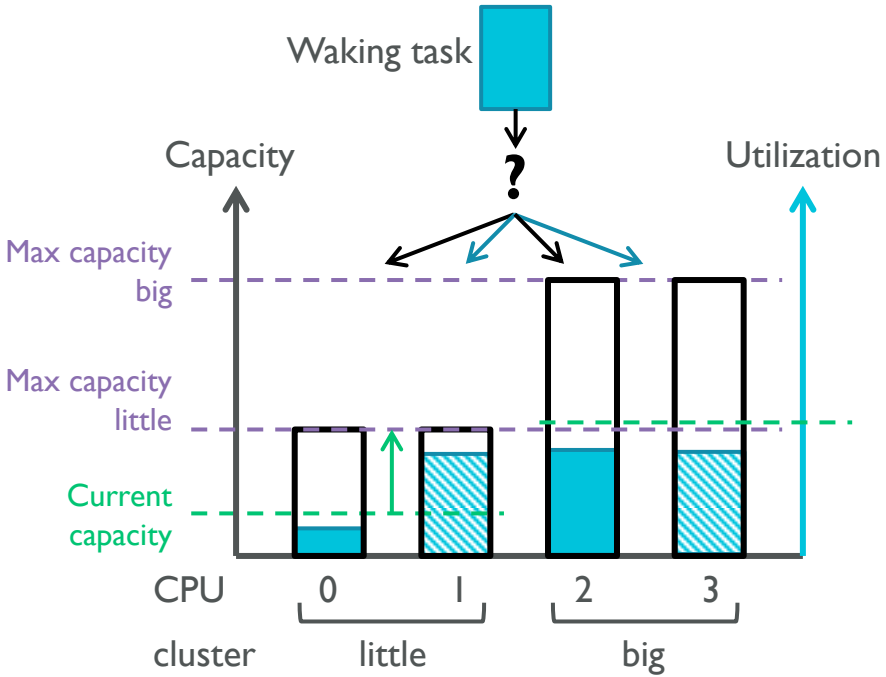
1. Tracking of task utilization.
2. Platform energy model.

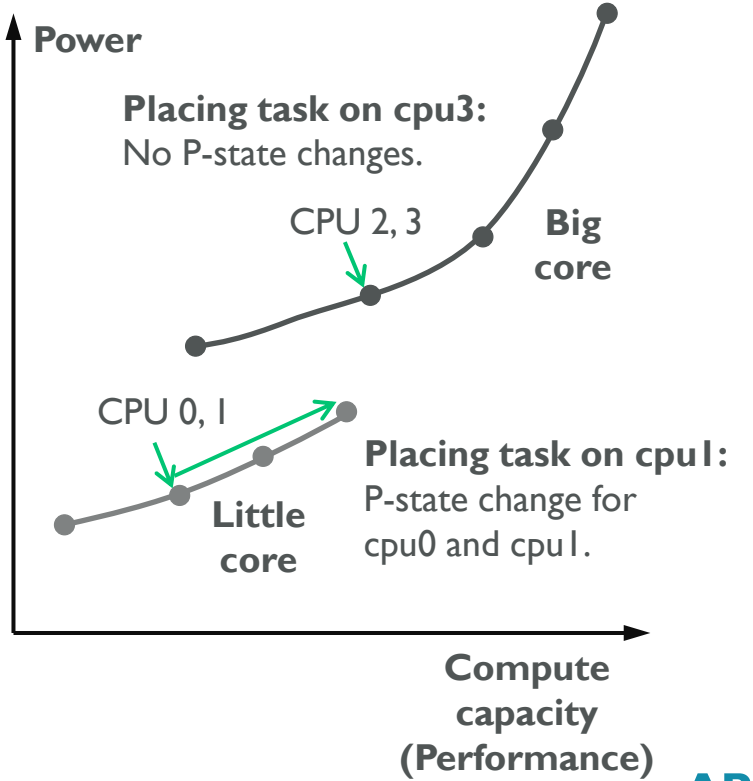
Supports all topologies, including SMP and big.LITTLE.



ARM

What is EAS – the energy model



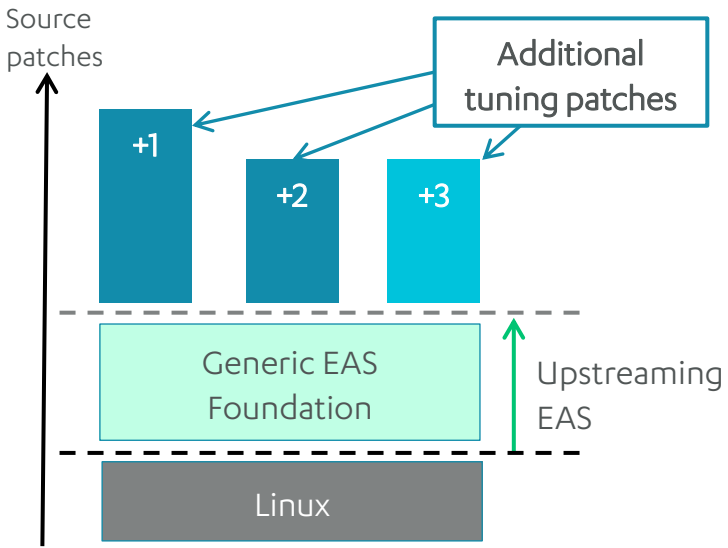


ARM

EAS vs big.LITTLE HMP

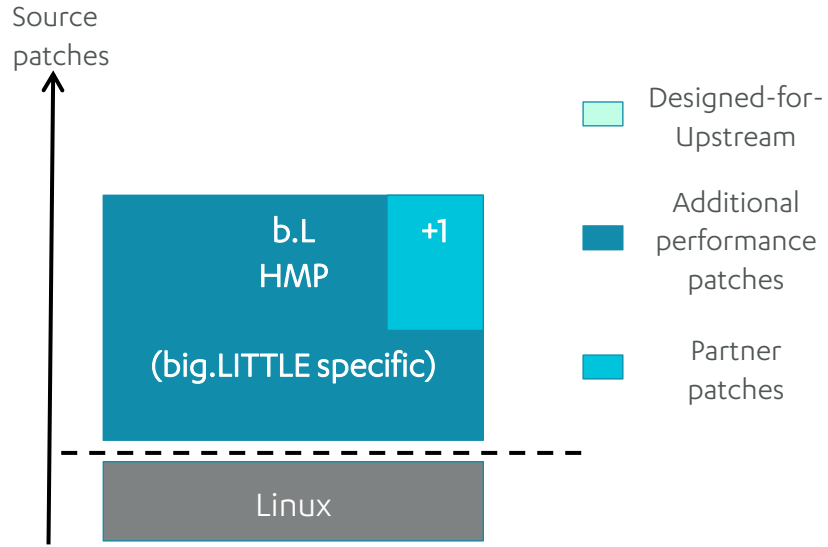
New **Energy Aware Scheduling**

- Generic energy model based approach fits all platforms and topologies.
- Foundation for further enhancements.



Existing **Heterogeneous MP** patchset

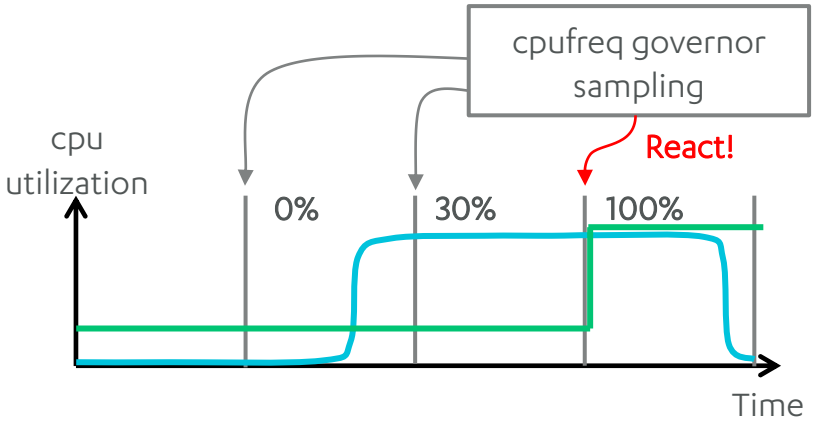
- big.LITTLE topology only.
- Hard coded behaviors.
- In Linaro LSK kernels (not mainline).

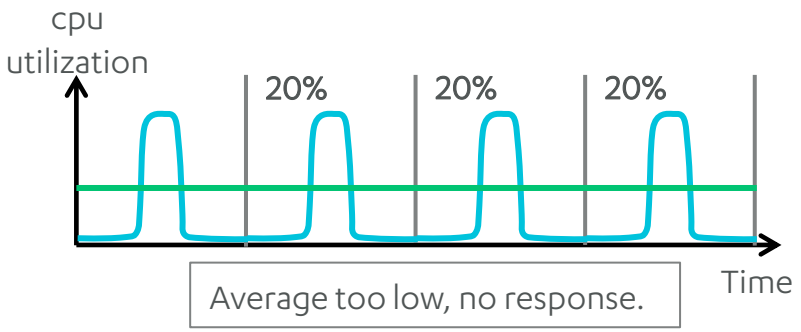


ARM

DVFS in Linux (cpufreq)

- Sampling based governors are slow to respond and hard to tune.
- **Sampling too fast:** OPP changes for small utilization spikes.
- **Sampling too slow:** Sudden burst of utilization might not get the necessary OPP change in time.

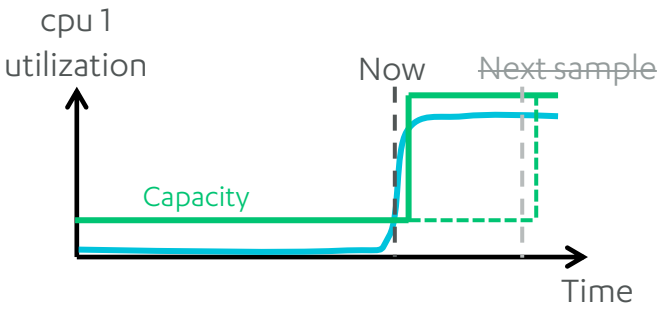


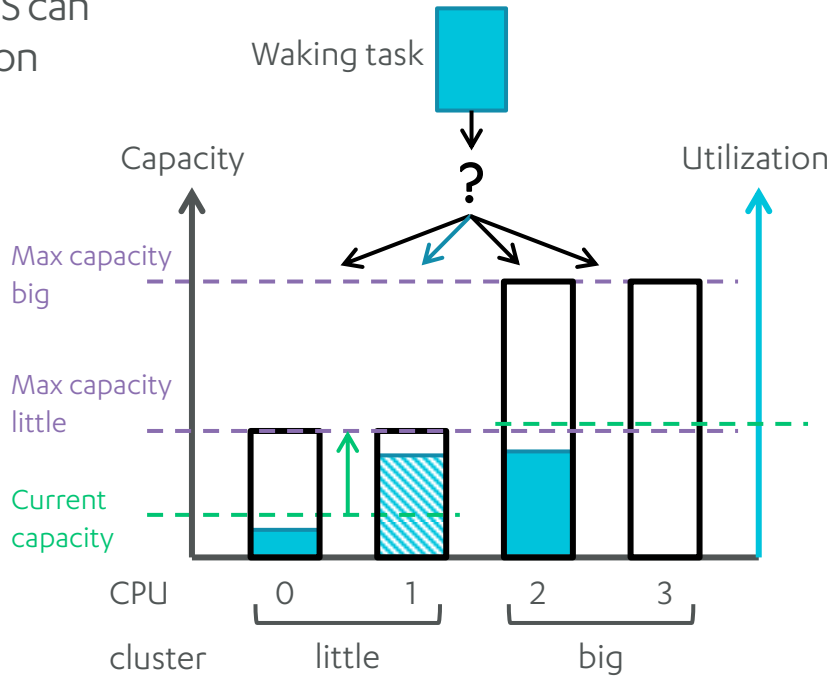


ARM

Scheduler-driven DVFS

- With scheduler task utilization tracking DVFS can be notified **immediately** when CPU utilization changes = improved responsiveness.





ARM

SchedTune

Current:

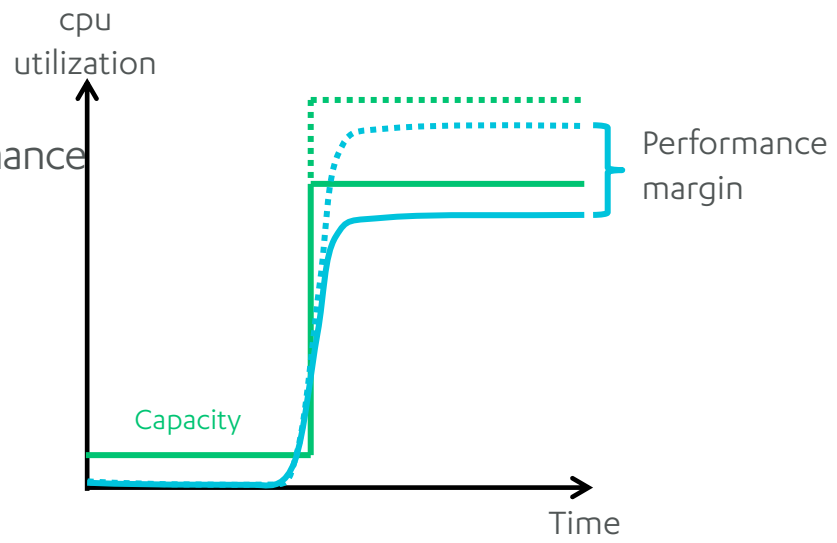
- A set of governor-specific tunables.

Goal:

- Single tunable to bias the energy/performance trade-off.

Prototypes:

- Global boost tunable:
  /proc/sys/kernel/sched_cfs_boost
- Task group (cgroup) based tuning:
  /sys/fs/cgroup/stune/<group>/schedtune.boost



ARM

Tunability improvements

Existing CFS with HMP

HMP tunables	hmp_domains, up_threshold, down_threshold, packing_enable, packing_limit, frequency_invariant_load_scale
Interactive governor	min_sample_time, hispeed_freq, go_hispeed_load, above_hispeed_delay, timer_rate, input_boost, boost, boostpulse

EAS

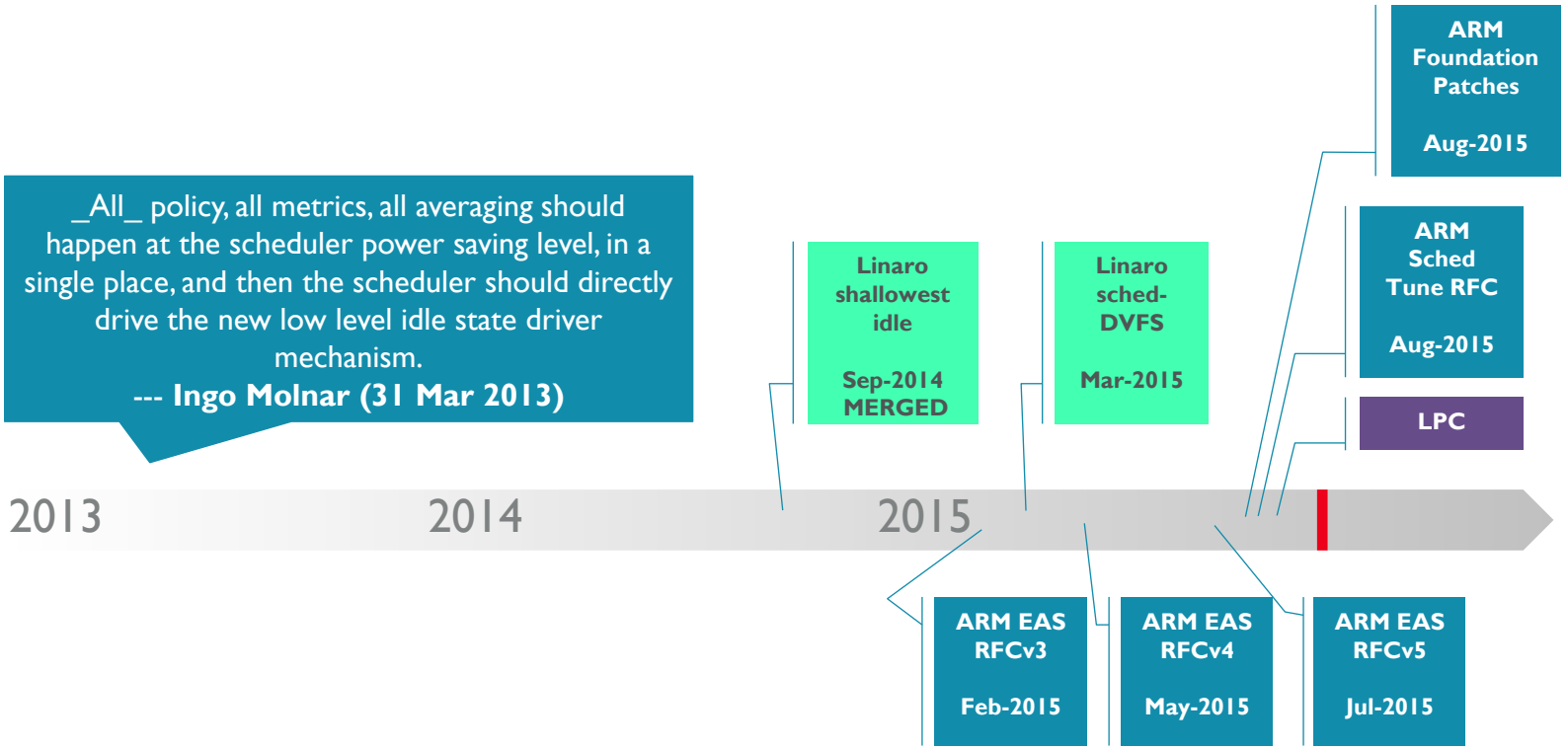
EAS tunables	**NONE - energy model only**
SchedTune	'boost' margin { *boost, boostpulse* }

Analysis tools

Tool name / function	Location
rt-app/ WorkloadGen (Linaro) Variable-intensity workload generator for Linux	https://wiki.linaro.org/WorkingGroups/PowerManagement/Resources/Tools/WorkloadGen
workload-automation (ARM) Automating benchmark runs and ftrace log capture (Linux, Android, ChromeOS)	https://github.com/ARM-software/workload-automation
TRAPpy (ARM) Python-based visualization tool to help analyze ftrace data generated on a device. Uses ipython & javascript	https://github.com/ARM-software/trappy
BART (ARM) Behavior Analysis Regression Testing Thread residency checker, used as the framework for regression testing for EAS.	https://github.com/ARM-software/bart
Idlestat (Linaro) Idlestat uses kernel ftrace to monitor and capture C-state and P-state transitions of CPUs over a time interval.	https://wiki.linaro.org/WorkingGroups/PowerManagement/Resources/Tools/Idlestat
kernelshark X11/GTK tool for analysis of ftrace data, useful for detailed scheduler analysis but does not offer the API capability of 'trappy' above.	http://people.redhat.com/srostedt/kernelshark/HTML/

Upstream progress

All policy, all metrics, all averaging should happen at the scheduler power saving level, in a single place, and then the scheduler should directly drive the new low level idle state driver mechanism.
--- Ingo Molnar (31 Mar 2013)



ARM

Current patchsets for review/testing

Patchset	URL
Scheduler driven DVFS PATCH v3	https://lkml.org/lkml/2015/6/26/620
EAS RFCv5	https://lkml.org/lkml/2015/7/7/754
SchedTune proposal	https://lkml.org/lkml/2015/8/19/419
Foundational Patches (frequency and microarchitecture contribution to capacity/utilization, split out from RFCv5) (***already queued for merging!***)	https://lkml.org/lkml/2015/8/14/296
Yuyang Du PELT rewrite v10 containing ARM enhancements to utilization calculation (***already queued for merging!***)	https://lkml.org/lkml/2015/7/15/159

- Request reviewers to send 'tested-by' or 'acked-by'

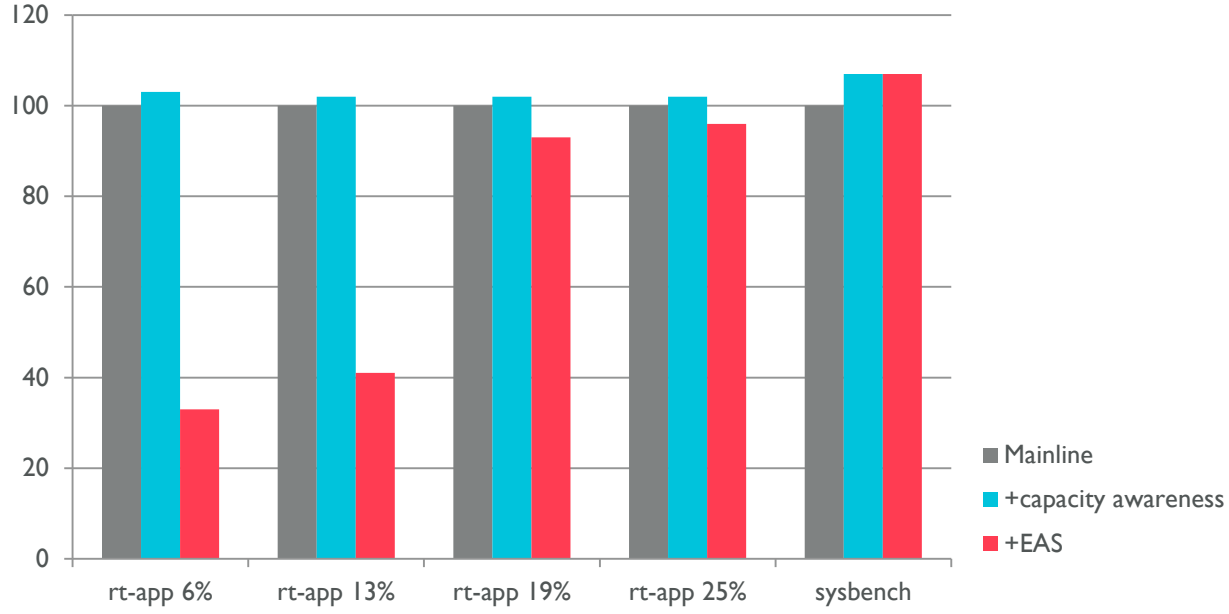
EAS RFCv5 update – posted 7-Jul-2015

- Linaro sched-DVFS integration + ARM improvements
- Maps all 6 HMP behaviors
- Landed on ChromeOS
  - (Linux 3.18 kernel)
- SchedTune equivalent to 'interactive'

HMP behaviors	Equivalent in EAS?
Wake migration	**Yes** – from wakeup pathways
Fork migration	**Yes** – new task initialized to max load
Forced migration	**Yes** – from periodic load balance
Idle-pull migration	**Yes** – from idle load balance
Offload migration	**Yes** – one task per CPU
Small task packing	**Yes** – built into design. From energy model.

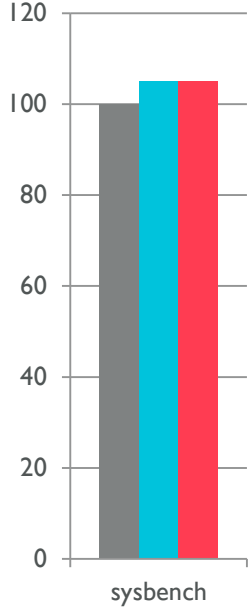
Results – RFCv5 @ ARM TC2

Energy



Lower is better.

Performance



Higher is better.

ARM

EAS Near-term Plans

LSK 3.18

- (Linaro – targeting 15.10) allowing direct HMP vs. EAS comparisons

Testing

- Use-cases (**ChromeOS** then **Android**)
- More platforms – can you help test?

Android testing & tuning targeting December 2015

- Starting with HMP sched_tests, migrating to 'bart' tests

Productization

- Analysis tools / test suites / tuning flow & documentation
- Energy model flow (based on power/perf measurements of dhrystone or sysbench)

EAS Needs YOU!

Now get on the list and ACK some patches!

Linaro

ARM