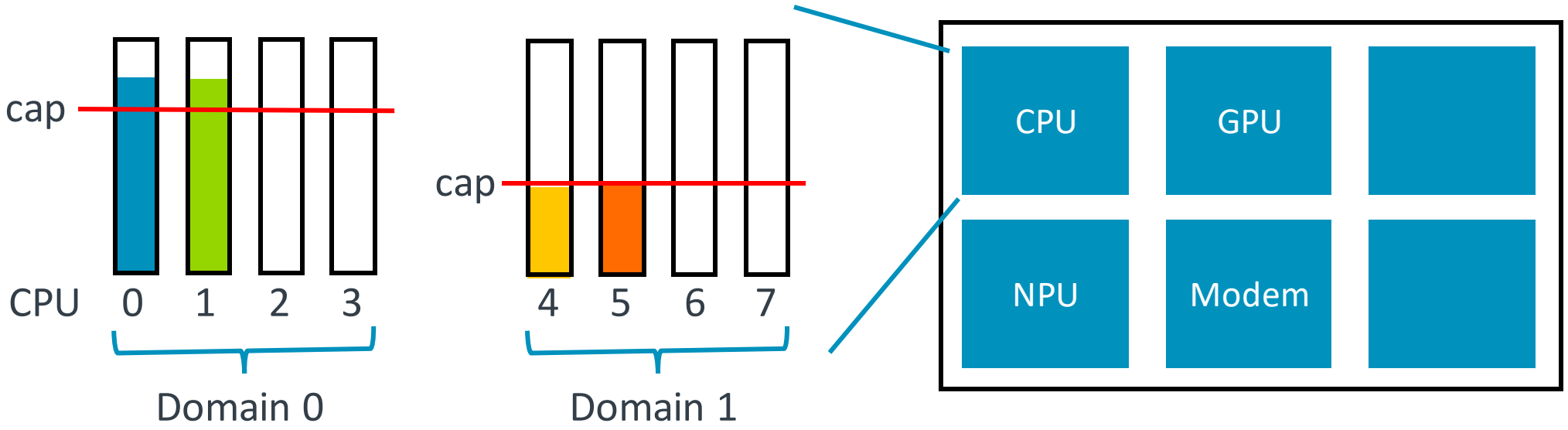
Performance guarantees under thermal pressure

Morten Rasmussen <morten.rasmussen@arm.com>

Linux Plumbers Conference 2019, 9-11 September, Lisbon

Unpredictable compute bandwidth

- Compute bandwidth is increasingly becoming unpredictable.
- In mobile systems performance capping is a common scenario.
- User-space has no information about minimum compute bandwidth.



arm

Is best effort compute enough?

- Current Linux kernel model:
  - User creates tasks, the kernel and platform delivers as good performance as it can.
  - User can tweak task placement and cpufreq governors (including util_clamp) but kernel and/or platform can override most of it.
- SCHED_DEADLINE is reservation-based and implies a bandwidth guarantee.
- Reservations are not cleared with thermal framework and could be impossible to fulfill.

- Should we have a “guaranteed” performance level that SCHED_DEADLINE could use for admission control?
- What level of “guarantee” should it provide?
- Who should provide it? DT, ACPI?

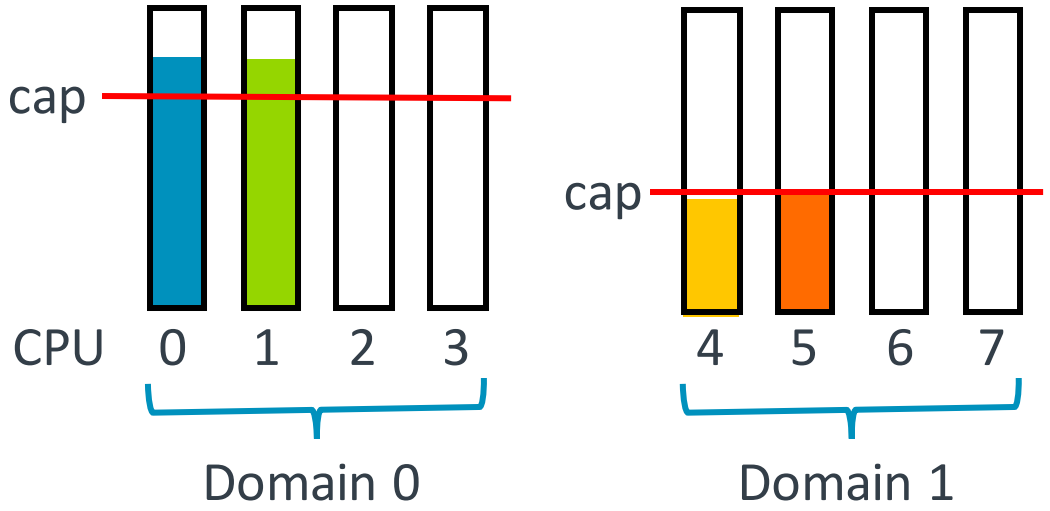
Task-centric thermal management

Morten Rasmussen <morten.rasmussen@arm.com>

Linux Plumbers Conference 2019, 9-11 September, Lisbon

CPU thermal management in Linux

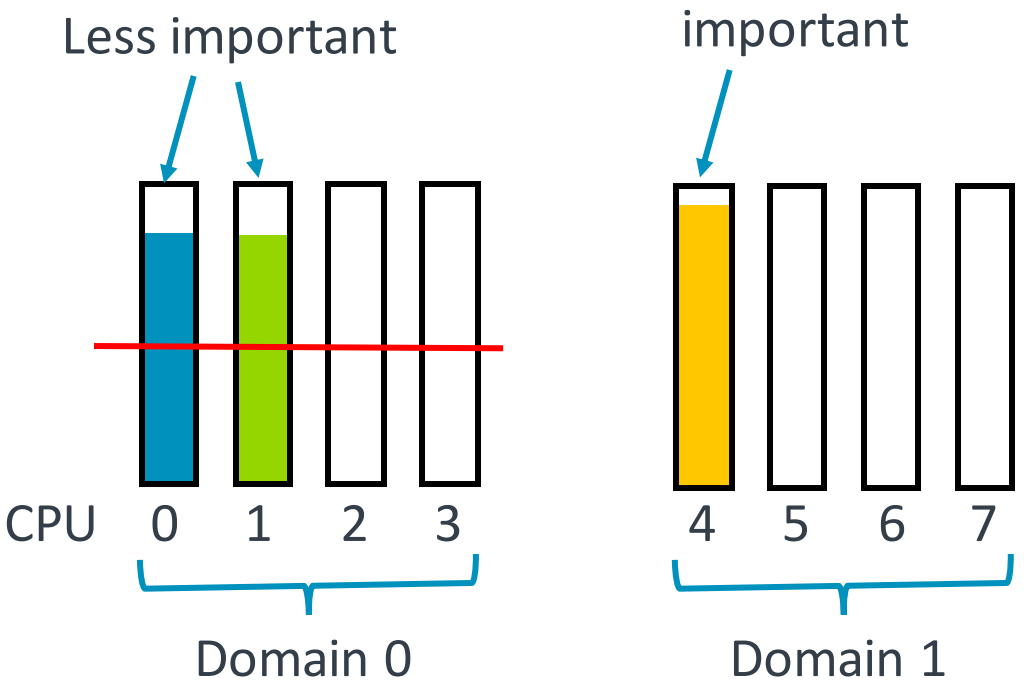
- Most systems control thermally unsustainable compute demand by performance capping.
- Control is performance domain-centric (clock domain) capping one or more CPUs together.
  - No knowledge about relative importance of tasks on the CPUs in each performance domain.
  - No knowledge about how to best spend the thermal budget.
  - Scheduler might help tasks to “escape” capped CPUs moving the problem somewhere else.



arm

Task-centric thermal management

- Ideally thermal management should maintain the budget while minimizing the perceived performance impact.
- Middleware/application is best positioned to decide the budget split.
- If tasks would self-adapt their compute demand, it would be even better.



arm

Power management mechanisms

- Applications
  - Self-adapting apps[1], util_clamp capping.
- Middleware:
  - cgroup bandwidth controller, (util_clamp capping).
- Kernel:
  - IPA, cpufreq (DVFS), ~~hot plugging~~, (cgroup bandwidth controller).
- Firmware:
  - Frequency capping.

Controlling compute demand

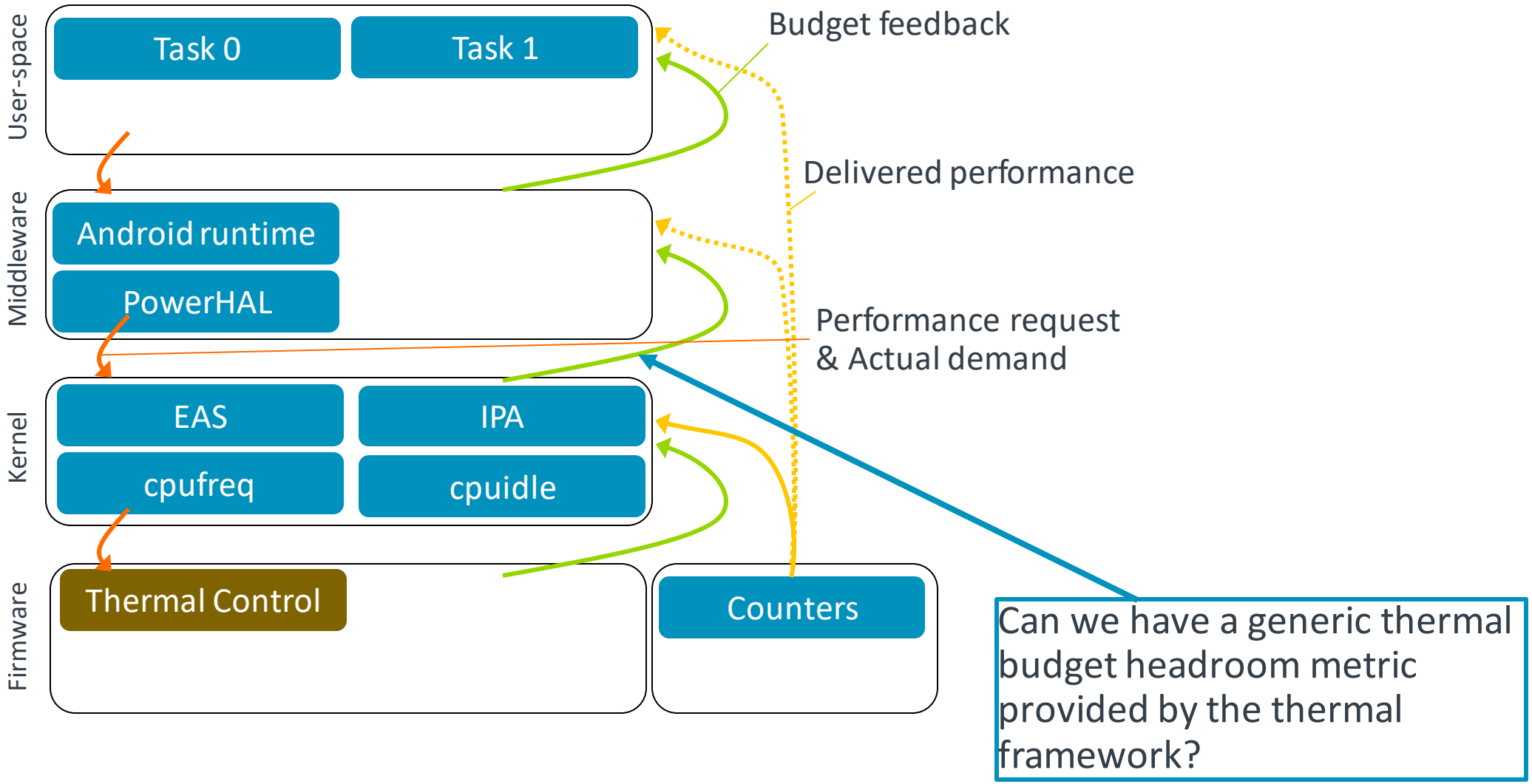
Controlling compute bandwidth

- Adapting demand is preferable to reducing compute bandwidth.
- Middleware or Kernel steps in controlling the bandwidth if applications don't behave.

1 https://blogs.unity3d.com/2019/04/01/higher-fidelity-and-smoother-frame-rates-with-adaptive-performance/

arm

Performance management hierarchy



arm