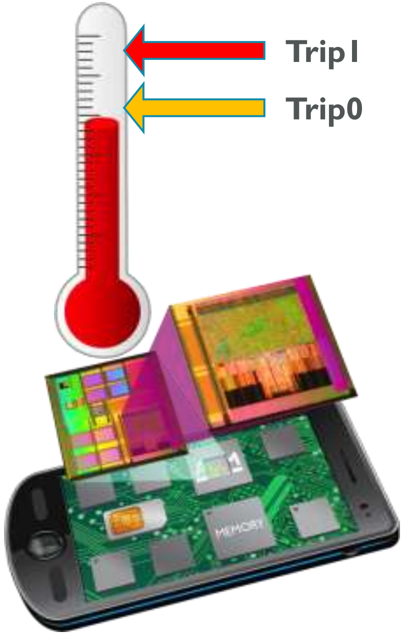
Intelligent Power Allocation for Consumer & Embedded Thermal Control

ARM

Ian Rickards
ARM Ltd, Cambridge UK

ELC San Diego
5-April-2016

Existing Linux Thermal Framework



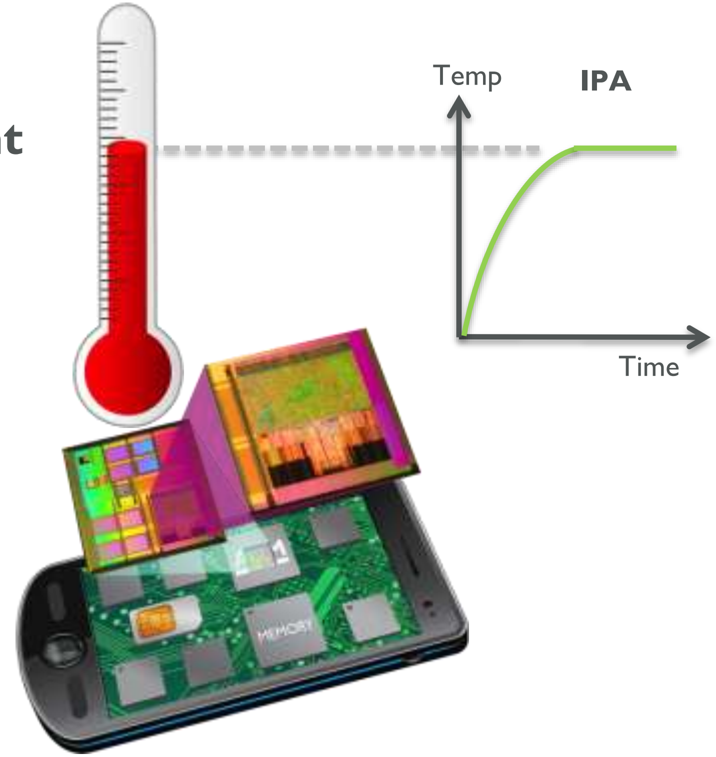
- Thermal trip mechanism using cooling devices
- Originally designed to turn on fans.
  'step_wise' governor adjusts MHz based on temp change
- **Reactive** on temperature overshoot
- **Fixed power partitioning** between different parts of SoC: GPU, CPU, DSP, video
  - does not adapt to current workload

Solutions currently used in mobile
Linux thermal framework
Custom firmware
Hotplug

ARM

IPA advancements

- **Proactive** vs. Reactive **Thermal Management**
  - Continuously adapting response based on power consumption and thermal headroom
  - Closed-loop control uses **PID algorithm for accurate temperature control**

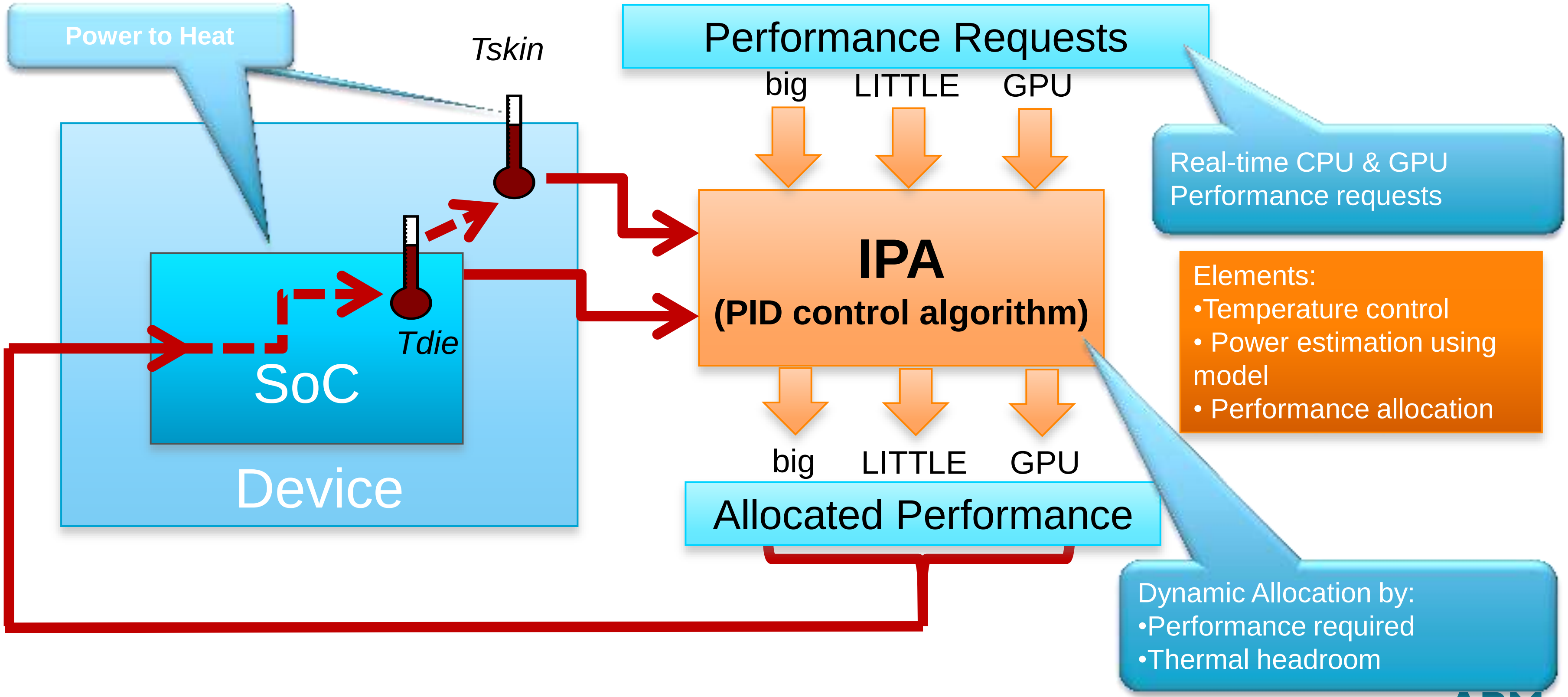


- **Dynamic Partitioning** vs. Fixed
  - Optimally allocates power to CPU or GPU depending on current workload

- **Merged in Linux-4.2**
  - Will benefit all operating systems based on Linux, no patches required

ARM

ARM Intelligent Power Allocation



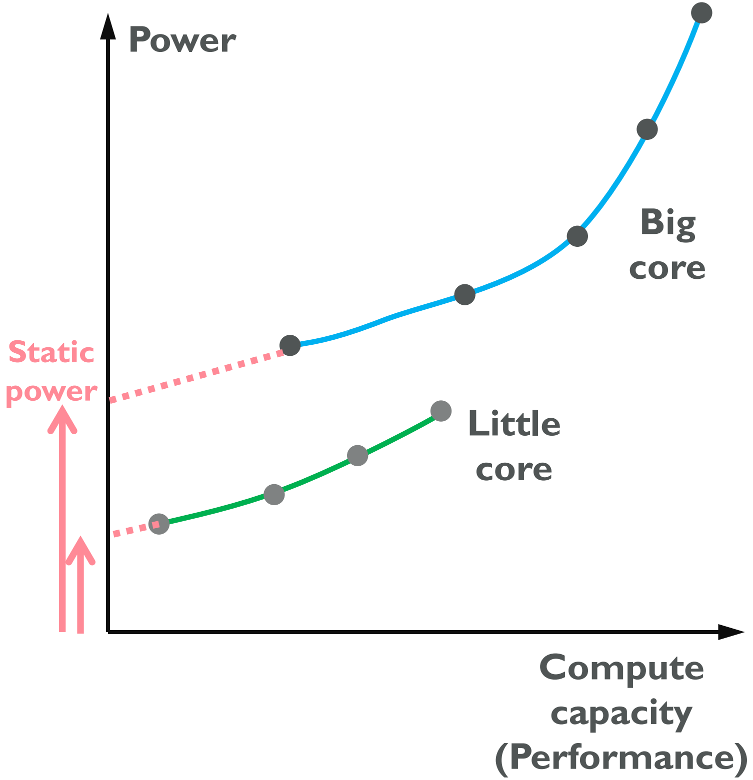
CMOS SoC Power Fundamentals

Static power

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

Dynamic power

- Pipeline depth/complexity
- Toggling nodes x capacitance x voltage^2



ARM

Requirements to use IPA

- **At least one on-chip temperature sensor**

- **Ability to passively control power of key System-on-Chip (SoC) IP blocks**
    - Blocks are ‘big’, ‘LITTLE’, ‘GPU’ – their power is controlled so they are ‘cooling devices’
    - Set CPU power cap using max frequency/voltage via cpufreq
    - [optional: set other device frequency/voltage via devfreq, e.g. GPU]

- **Power models of SoC IP blocks: Frequency/voltage & ‘Utilization’**
    - Dynamic power model
    - Static power model [optional]

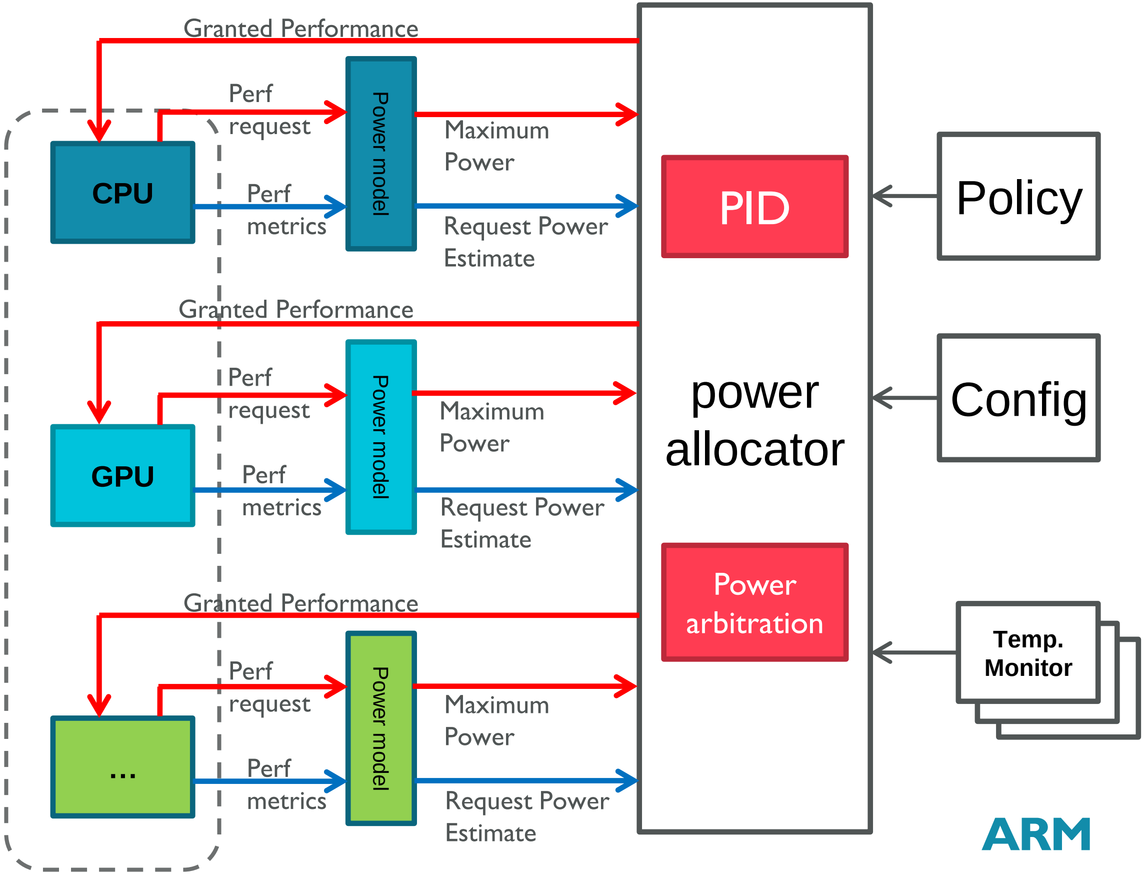
- **Translate power <–> performance cap**

IPA Algorithm - Overview

Governor performs two tasks:

1. Keeps system within thermal envelope
   - Controls total power budget
   - Exploits thermal headroom

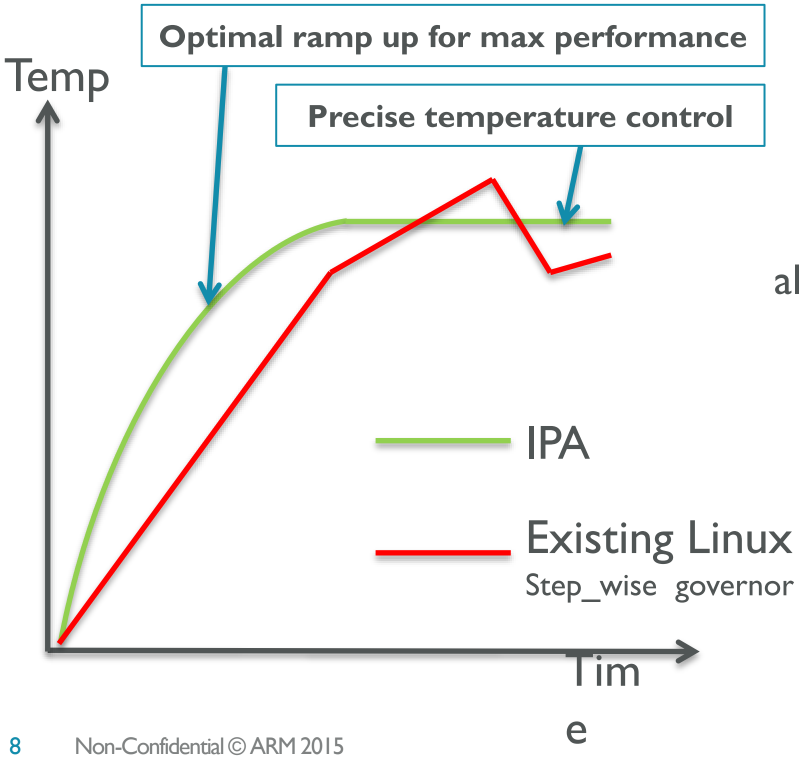
2. **Dynamic power allocation per device**
   - Performance demand & power models
   - Power divided based on what each device requested. Anything left over is distributed among the devices, up to their maximum.



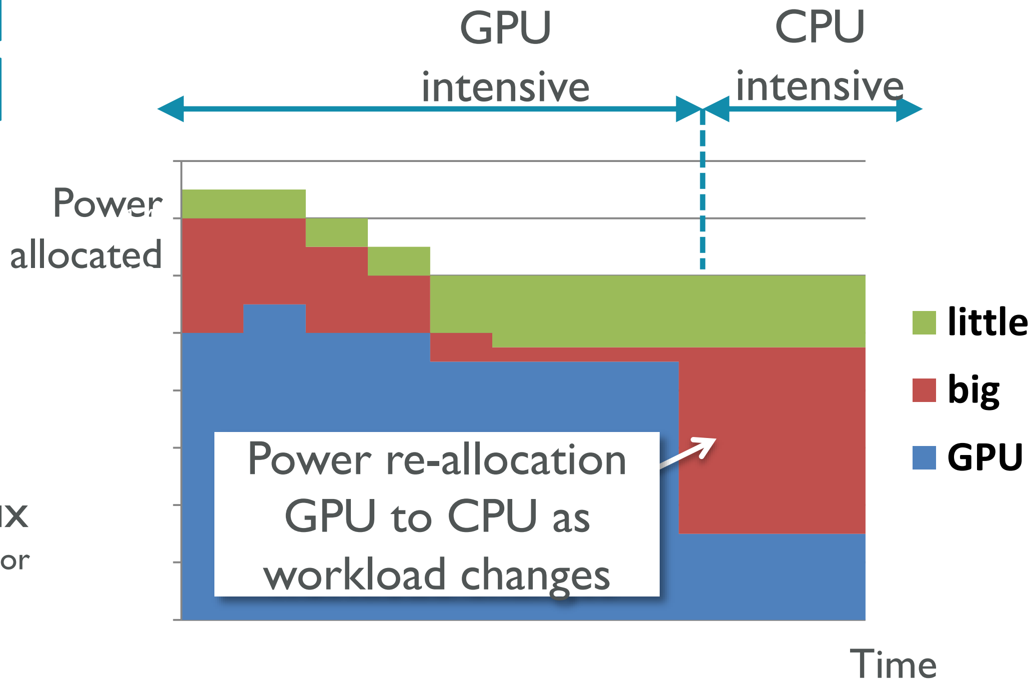
ARM

IPA benefits illustration

PID accurate temp control



Dynamic power allocation



Setup

Porting parameters

sysfs location: /sys/class/thermal/thermal_zoneX/:

struct element	DT name	sysfs (writeable)	
switch_on temperature	trip-point@0	trip_point_0_temp	Temperature above which IPA starts operating (first passive trip point – trip 0)
desired_temperature	trip-point@1	trip_point_1_temp	Target temperature (last passive trip point – trip 1)
weight	contribution	cdevX_weight	Weight for the cooling device
sustainable_power	sustainable-power	sustainable_power	Max sustainable power
k_po	*[not h/w property]*	k_po	Proportional term constant during temperature overshoot periods
k_pu	*[not h/w property]*	k_pu	Proportional term constant during temperature undershoot periods
k_i	*[not h/w property]*	k_i	PID loop's integral term constant (compensates for long-term drift) When the temperature error is below 'integral_cutoff', errors are accumulated in the integral term
k_d	*[not h/w property]*	k_d	PID loop's derivative term constant (typically 0)
integral_cutoff	*[not h/w property]*	integral_cutoff	Typically 0 so cutoff not used

DT registering thermal_zone and trip points

```
thermal-zones {
        skin {
                polling-delay = <1000>;
                polling-delay-passive = <100>;
                sustainable-power = <2500>;

                thermal-sensors = <&scpi_sensor0 3>;

                trips {
                        threshold: trip-point@0 {
                                temperature = <55000>;
                                hysteresis = <1000>;
                                type = "passive";
                        };
                        target: trip-point@1 {
                                temperature = <65000>;
                                hysteresis = <1000>;
                                type = "passive";
                        };
                };

                cooling-maps {
                        map0 {
                                trip = <&target>;
                                cooling-device = <&cluster0 0 4>;
                                contribution = <1024>;
                        };
                        map1 {
                                trip = <&target>;
                                cooling-device = <&cluster1 0 4>;
                                contribution = <2048>;
                        };
                        map2 {
                                trip = <&target>;
                                cooling-device = <&gpu 0 4>;
                                contribution = <1024>;
                        };
```

Power allocator

2 passive trip points
trip-point@0
trip-point@1

Switch_on
55degC

Target temp
65degC

Currently only trip-points, sustainable-power and weights can be specified in DT as these are direct h/w properties

When using DT, boot happens with defaults and userspace can change it by writing to sysfs files.

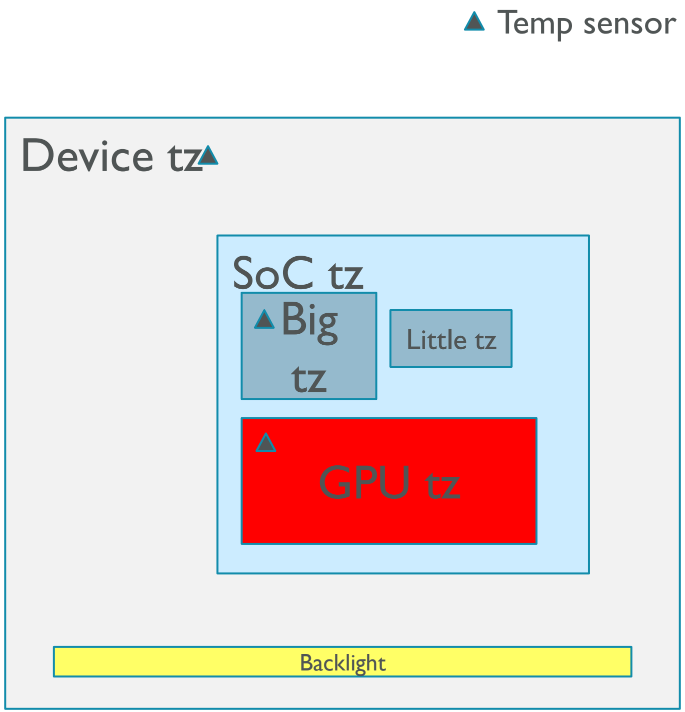
ARM

Thermal zone hierarchy

- Thermal zones structured as tree (supports multiple temp sensors)
- Power restricted by cooling devices
- Allocated power is **minimum** of that requested by all thermal zones

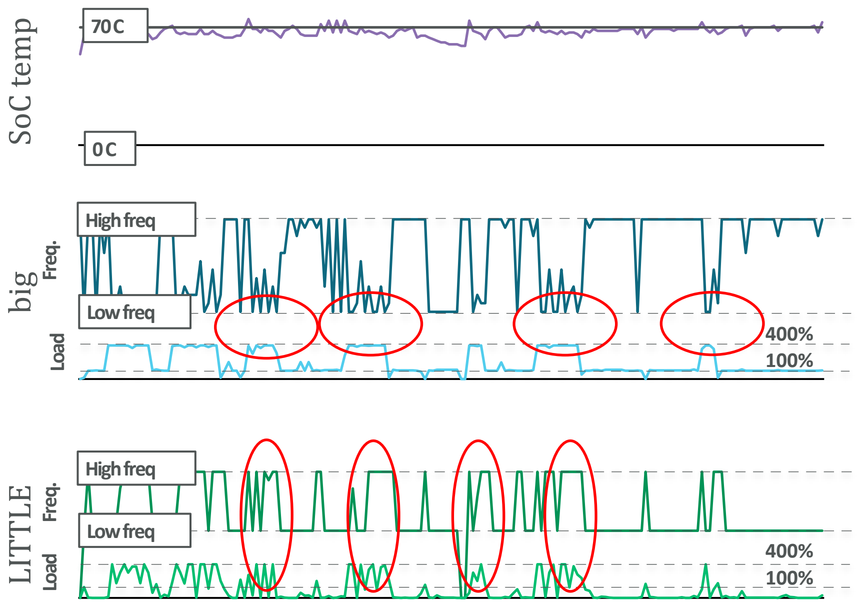
⇒ Rapid limiting from CPU cluster temp sensor

⇒ Additional limiting from motherboard sensor



ARM

IPA in action on 4x4 big.LITTLE



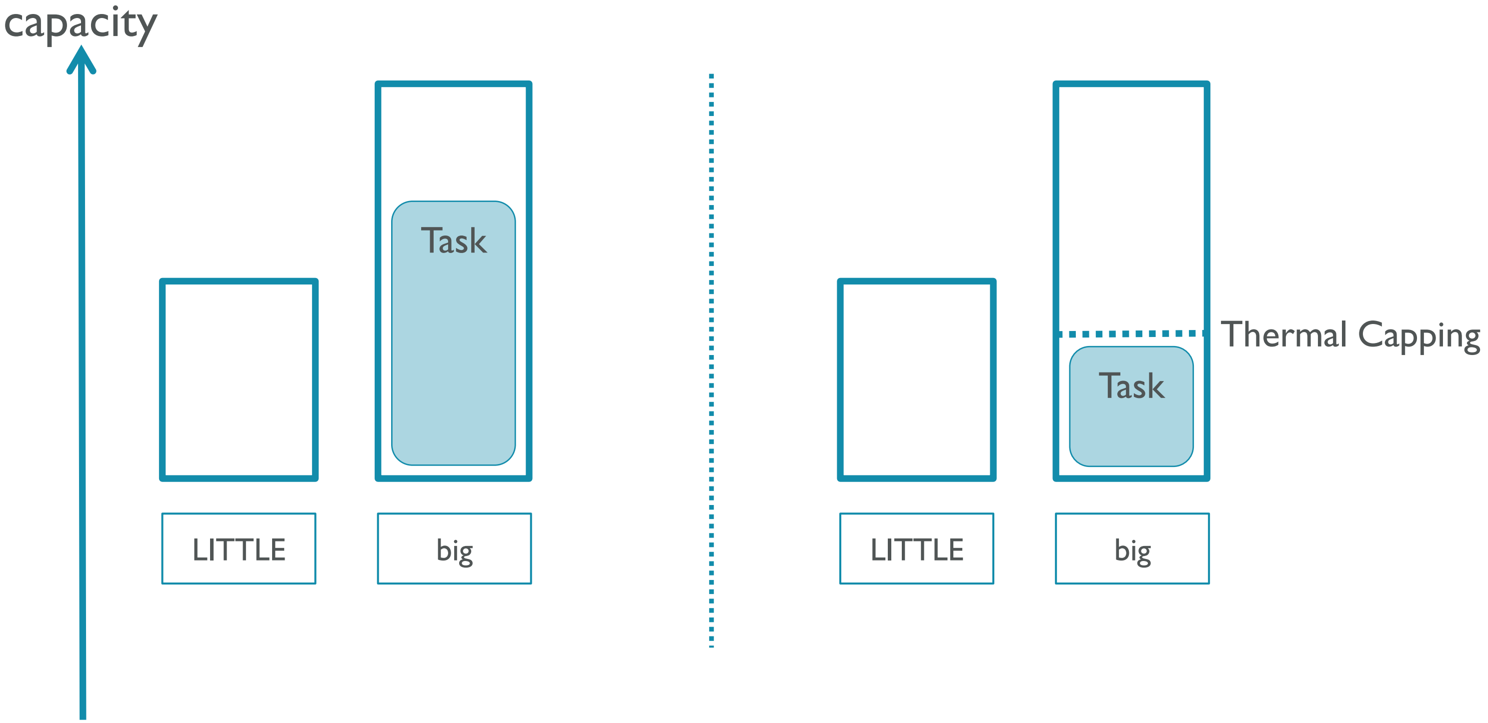
- GeekBench alternates between single and multithreaded phases

+ On big cores frequency is adjusted automatically based on parallel load
    - When 4 cores are active frequency is lower

+ On LITTLE cores frequency follows load
    - When 4 cores are active frequency is lower

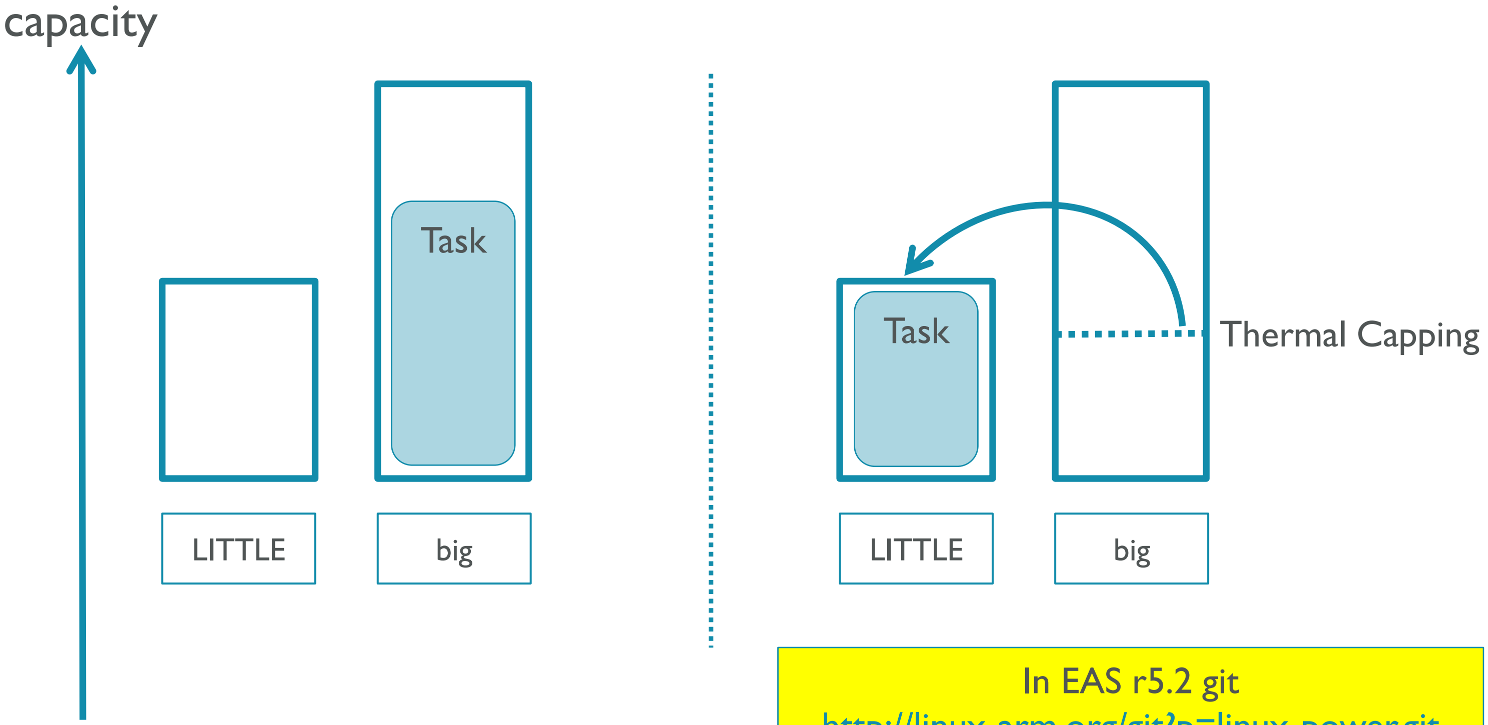
ARM

Changing Capacities



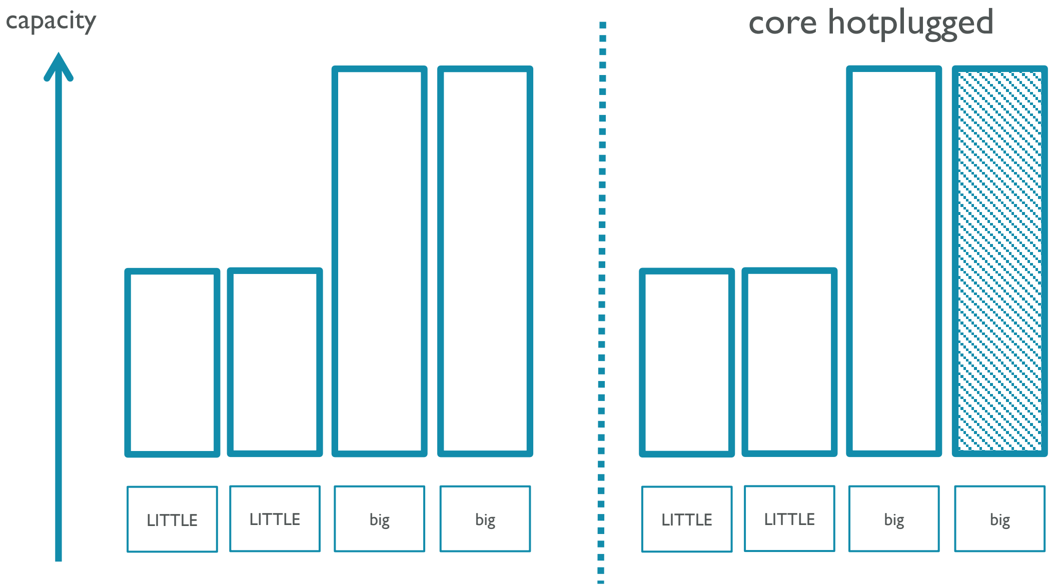
ARM

Changing Capacities – Energy Aware Scheduling



In EAS r5.2 git
http://linux-arm.org/git?p=linux-power.git
LKML posting planning for EAS RFCv6

Static Power - Hotplug



- Turn core OFF
  - Start with 'big'
- Progressively hotplug more cores
- Issues
  - Latency of thermal response
  - Ping-pong of core due to large discrete steps in power reduction

Summary

- 'Intelligent Power Allocation' now available for the Linux kernel

- IPA is designed to **maximise performance in the thermal envelope**:
  - **Proactively** adjusts available power budget, based on device temperature
  - Allocates power **dynamically** between CPU/GPU, based on workload

- Results show:
  - **More accurate thermal control**
  - **Better performance** than existing governors in Linux thermal framework

- Future: Core idling / Unified energy model

ARM

Demo - Technical info

- **Odroid-XU3 Exynos 5422   (4xCortex-A15 + 4xCortex-A7)**

- **Automation: ARM 'Workload Automation'**
- **ipython analysis and tuning flow using TRAPpy**
  - See Patrick Bellasi presentation "LISA & friends" talk @ 4.20pm
  - ARM "Ice Cave" demo => GPU intensive
  - Antutu – varying CPU/GPU load

- **IPA released in Linux 4.2, Linaro LSK 3.10 & 3.18**
  - Tooling on https://github.com/arm-software

Thank you