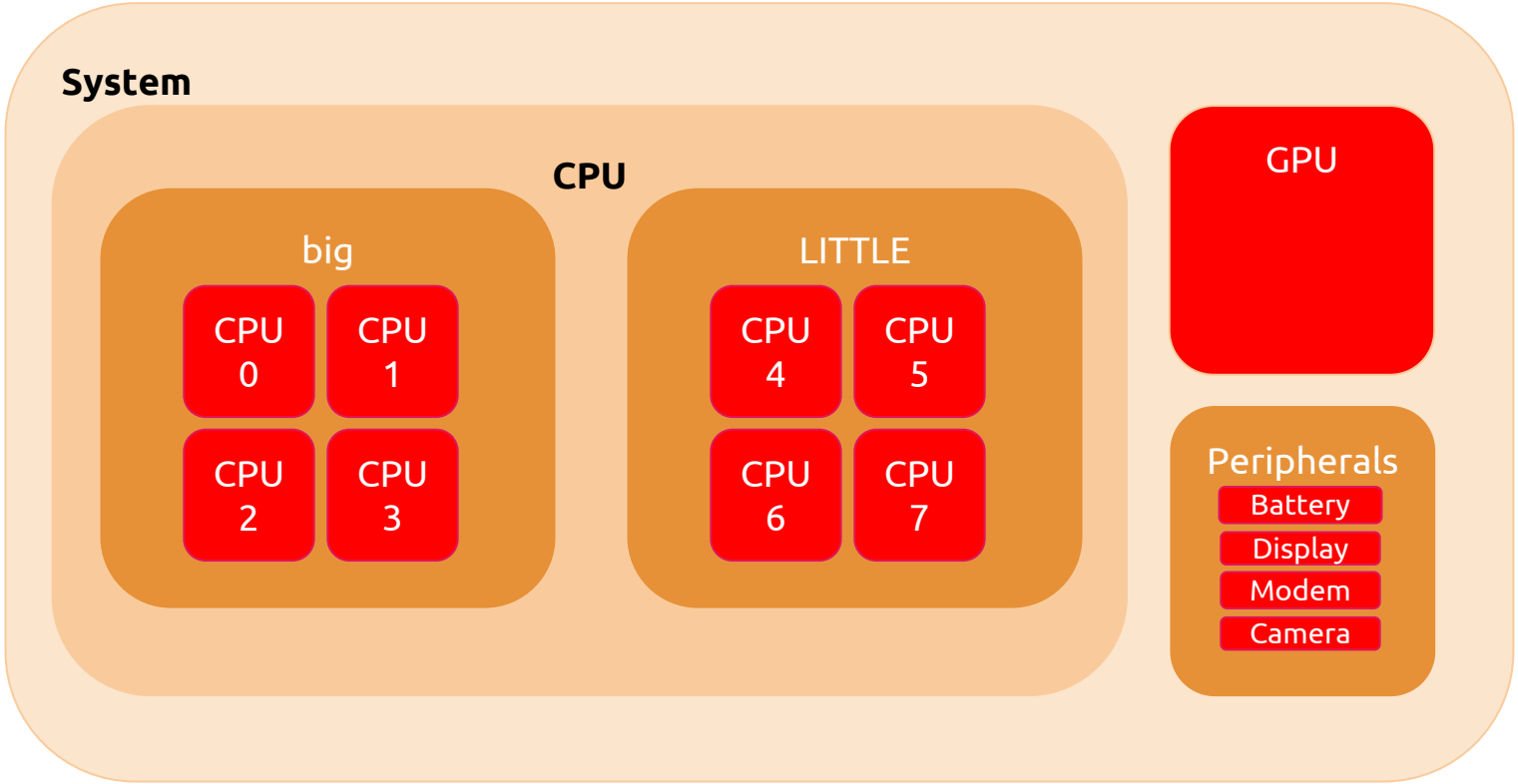
Embedded Thermal Use Cases (how to handle them?)

Amit Kucheria

13th Nov 2018
Linux Plumbers conference

LEADING
COLLABORATION
IN THE ARM
ECOSYSTEM

Typical SoC


System
CPU
big
CPU 0
CPU 1
CPU 2
CPU 3
LITTLE
CPU 4
CPU 5
CPU 6
CPU 7
GPU
Peripherals
Battery
Display
Modem
Camera

Capturing hotspots

- Lots of sensors (20+ in some mobile SoCs)
- Need to capture hotspots quickly and mitigate
  - 30-40℃ rise in under 10ms
  - Even within a CPU cluster, >5℃ difference between hot CPU and ambient cluster temperature
- But some mitigation actions apply to a group
  - E.g. cpufreq acts on the cluster
  - Hence need for aggregation

Modeling an Octacore CPU with 8 sensors

- *Option 1:* One thermal zone per sensor
  - No way to track cluster temperature trend
  - If the info was available, possibility to do smarter scheduling, idle injection
- *Option 2:* 8 sensors in single thermal zone using coefficients for a linear equation
  - Not possible today, needs fixing first (of-thermal.c)
  - Q: Will end up throttling all CPUs, how to handle hotspots?
- *Proposal:* model the hierarchical topology of SoC
  - Hierarchical thermal zones (per-CPU zone, per-cluster aggregation)
  - Possibility of different governors at each level
  - Extend to model other devices on SoC
  - Q: What would DT/ACPI representation look like?

Winter is coming...

- No way to ensure lower thermal threshold
  - Needed in very cold environments, to ensure circuit closures
  - Downstream solution: a “heating” governor
- Potential solutions:
  - Multiple governors per zone (downstream)
    - One for <0°C, another for >0°C
    - Regulator “cooling” devices (heating) to raise floor voltages
  - Range-based governor (more elegant?)

Other downstream hacks for mobile

- Sensors spanning thermal zones (patches submitted for review)
  - Single sensor between two hotspots needing different governors
- Multiple governors associated with a thermal zone
  - E.g. userspace and step-wise, so you could directly influence the application
  - E.g. one for hot and one for cold temperatures
- Corner cases:
  - Initialise thermal framework earlier?
    - Problem: Core at fs_initcall() but several cpufreq drivers at late_initcall() or device_initcall()
    - Booting faster
  - Allow different governors for each thermal zone at boot
    - Patch rejected to keep in line with other frameworks that don't allow governors to be specified in DT

In summary

- Ultimate goal is to avoid throttling for as long as possible
  - Power allocator governor can help in smoothing out the sawtooth
  - Smarter scheduling + idle injection should help
- More flexible sensor/thermal zone topology desirable
  - Multiple sensors within a thermal zone
  - Sensors spanning thermal zones
  - Aggregator thermal zones (avg, min, max)
  - Hierarchical thermal zones
  - Q: All of the above?
- Range-based governor