A generic energy model description

Linux Plumbers Conference 2022
Power management micro-conference

Daniel Lezcano

Description of the energy model

- Device tree
  - No clear definition of the energy model
  - Need to be computed from the dynamic power coefficient
  - No static leakage
  - No energy cost for performance transitions
  - Some platforms have performance indexes, no frequencies, no voltages, ~~$P = C \times f \times V^2$~~
  - Adaptative Voltage Scaling

- SCMI
  - Power per performance level only

Linux Energy model

- The energy model exists
- Implementation initially for CPUs, then GPU
  - Does not really appears as generic
  - Focused on performance domain
- However, the energy model is embedded per struct device
  - Accessible from everywhere
- An energy model is more complex as it depends on :
  - the temperature
  - the usage of the device

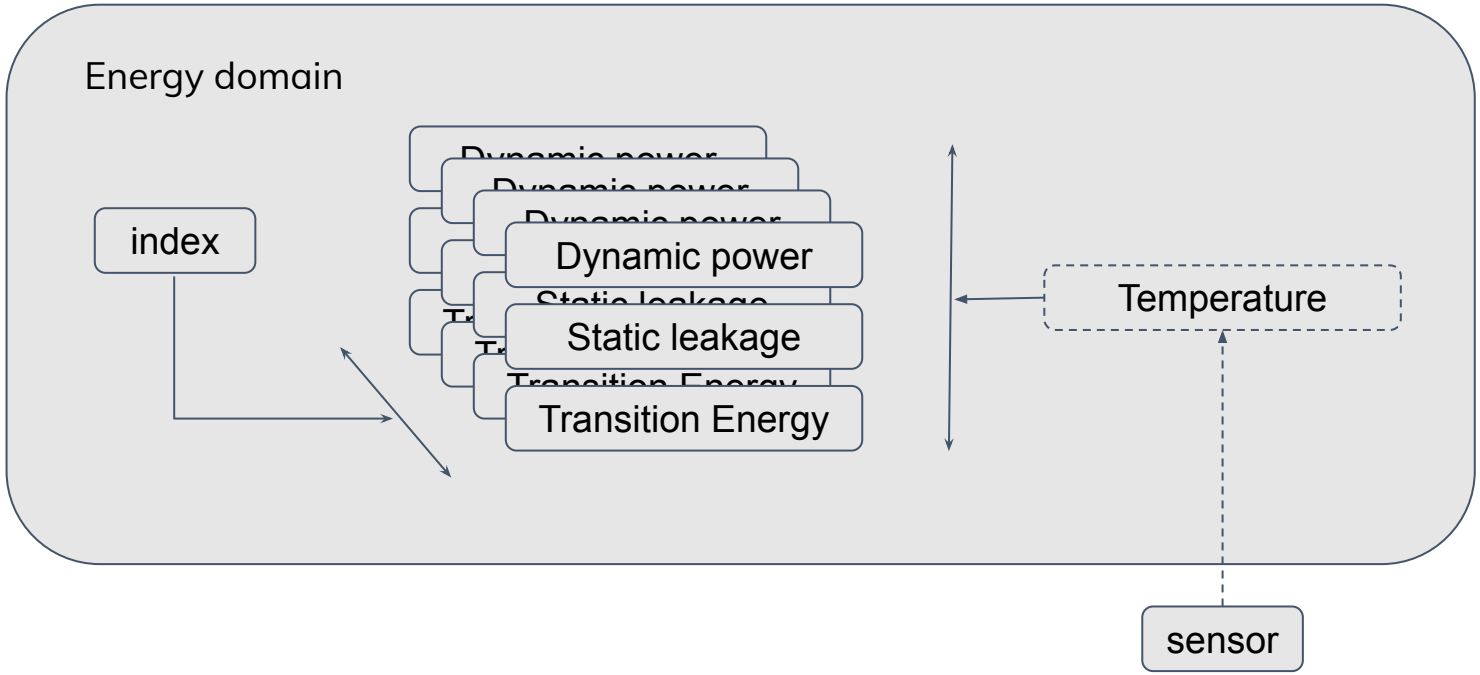
A paradox

- SoC vendors are reluctant to share the energy information
- But they want an Energy Aware Scheduler
- Power based control (thermal, capping, EAS)
- Hardware can provide energy counters
- Out of tree kernels have their own energy model implementation

An invitation

- With the EAS, the power capping and the thermal power:
  - idea of sharing some power numbers moved forward
  - power numbers help to be more energy efficient
- New platforms are providing energy counters, power meters in different SoC places
- Chrome OS is looking forward for more power numbers
- Shall we propose a generic energy model description and in kernel handling to give the opportunity to provide the power numbers?

Proposal


Energy domain
index
Dynamic power
Static leakage
Transition Energy
Temperature
sensor

Proposal

No inference: all values are clearly expressed

- index : the performance index which is common to all devices
  - Not a frequency
- dynamic power (uW) :
  - Not deduced
- static leakage in (uW) :
- Transition energy cost (uJ)

All these values are expressed in function of the temperature:

- Linear regression coefficients
- Array of pair <temp, value>
- Constant if not depending on the temperature