PM constraints/Tegra

Antti P Miettinen

Egotrip: who is Antti P Miettinen?

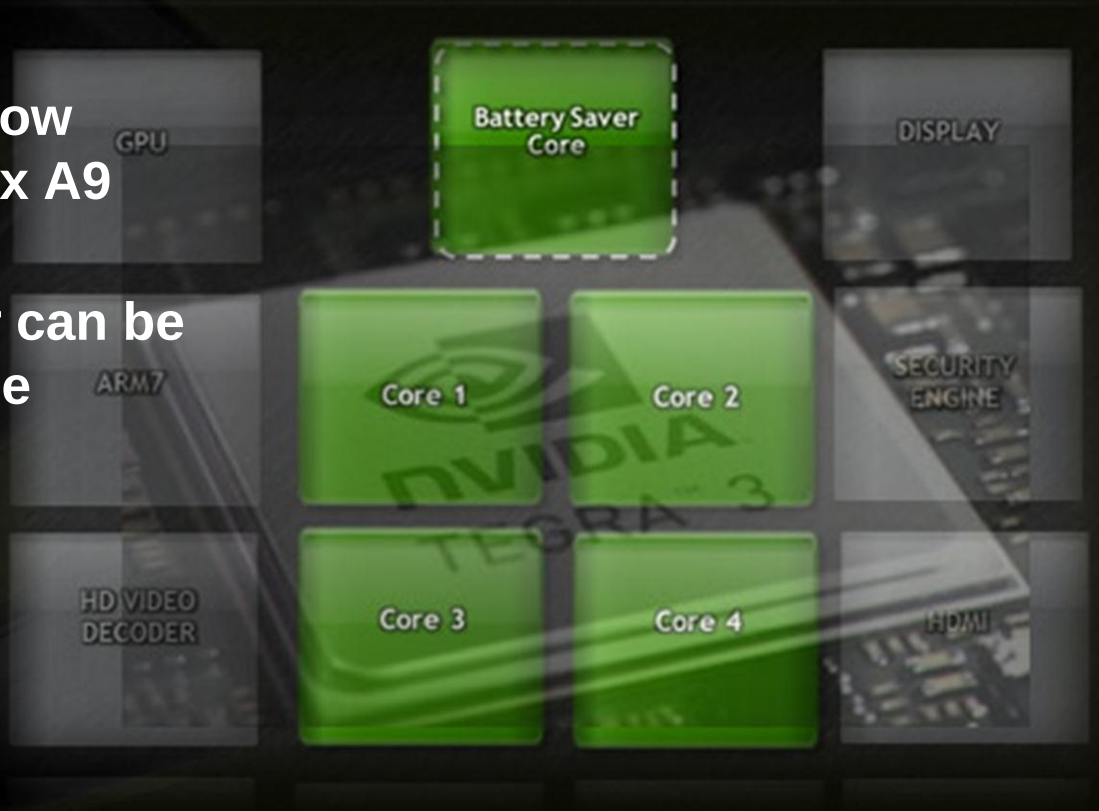
- Not a hockey player
- Energy Efficiency Engineer @ NVIDIA
  - One year with Tegra now
- Used to be a researcher at Nokia Research Center
  - About seven years of EPM research
- Newbie in hard core Linux work
  - But surrounded by great professionals

Why constraints?

- Peak current management
  - Regulators have limited current capacity
  - So does the battery
- Thermal management
  - Silicon temperature, characterization constraints
  - Battery temperature, device skin temperature
- Helping e.g. cpufreq and scheduler
  - Timely performance boost
  - Optimizing energy efficiency
- Platform quirks
  - Tegra 3 companion core impersonates CPU0

Tegra 3, overall

- 4+1 Cortex A9 CPUs
  - G cluster: 4 high performance Cortex A9 cores
  - LP cluster: 1 low leakage Cortex A9 core
  - Only 1 cluster can be active at a time
- GPU, video, etc



Tegra 3, some details

- Two voltage rails: VDD_CPU, VDD_CORE
  - VDD_CPU: quad core CPU cluster
  - VDD_CORE: LP A9 and engines
  - Voltages are not independent
- Shared clock for G cluster CPUs
- LP A9 is CPU0
  - Must quiesce CPU1-3 for cluster migration
- CPU0 power gating is actually VDD_CPU rail gating
  - CPUidle characteristics are different

Constraining in current Tegra kernel

- PM QoS additions for
  - Limiting CPU frequency: both minimum and maximum
  - Number of CPUs online: both minimum and maximum
- Autohotplug/cpuquiet
  - Separate presentation
- Peak current management
  - Implemented in platform code
  - CPU frequency cap
  - Based on number of online cores and temperature
- Issues in current implementation
  - Per core CPU frequency?
  - PM QoS has no differentiation for requests

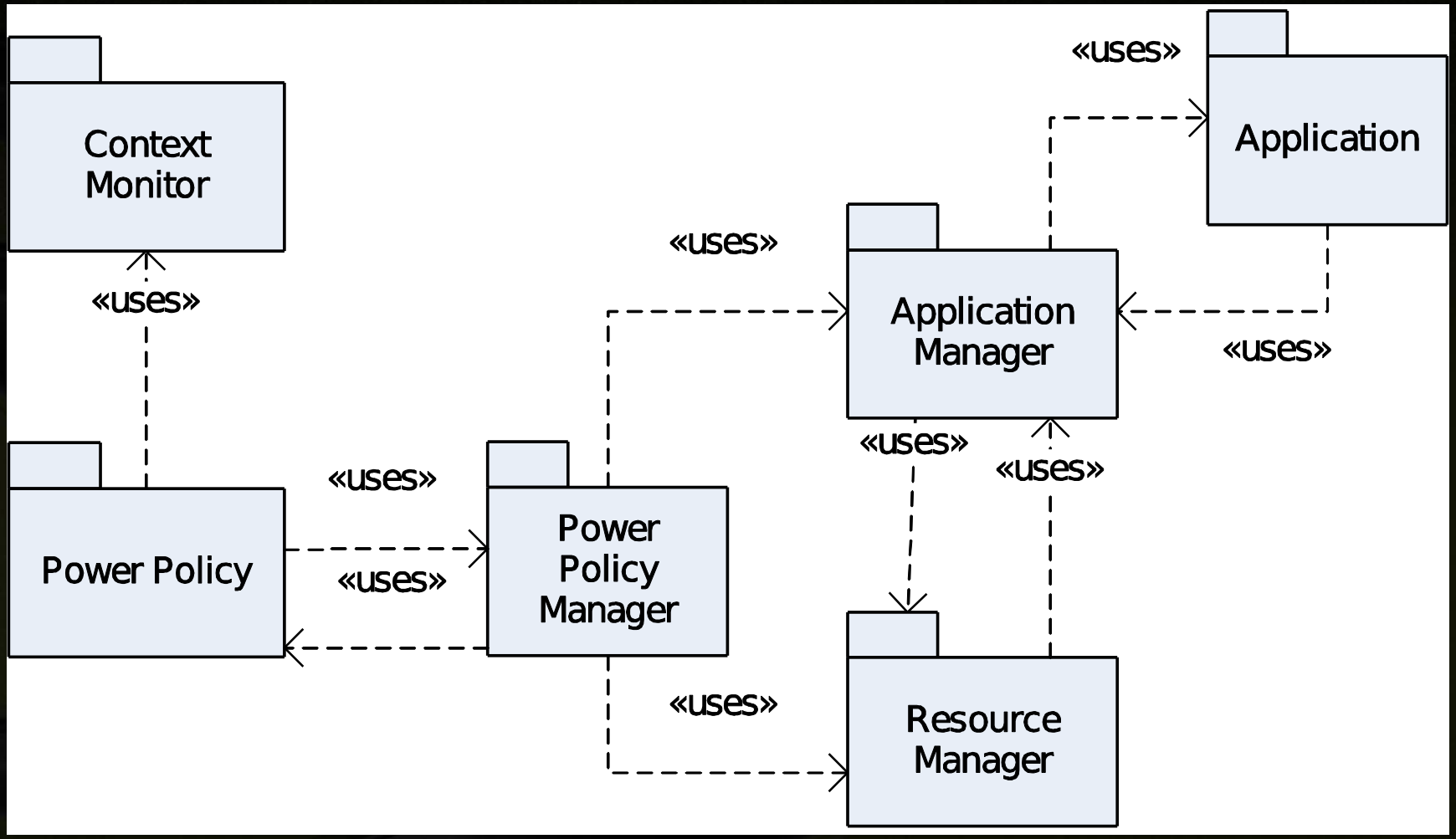
Requirements: user space interface

- Kernel tries it's best to manage the HW
  - But trading power vs performance is difficult
- Application/middleware often knows better
- Currently identified needs
  - Minimum CPU frequency for ensuring performance
  - Maximum CPU frequency for ensuring energy efficiency
  - Minimum number of online CPUs for performance
  - Maximum number of online CPUs for energy efficiency
- Under study
  - Blocks other than CPU
- Notification interface?

Requirements: in kernel constraints

- Thermal and peak currents
  - CPU frequency
  - Number of online CPUs
  - Possibly other blocks
  - Current capacity at different levels: regulator, battery
- Cluster switching requires CPU activity management
  - Only CPU0 can be active for migration to LP
- Limiting number of online cores: cpuquiet
  - Separating policy and mechanism
- Differentiation mechanism needed
  - User space minimum frequency may be just a wish
  - Peak current frequency cap is a hard limit

An idea once upon a time



From G. Bosch, P. Niska, *System-Level Power Management for Mobile Devices*, IEEE CIT 2007

PM constraints, general thoughts

- Scope of constraints framework?
  - Many things could be modeled as constraints
  - LP cluster has one core, fast cluster has 4
  - Voltage domains have dependencies
  - Clocks have dependencies
- Application interface is essential
  - Per CPU vs misc device minors?
- Differentiation is essential
  - Or separate request and enforcement layers
- Configurability is essential
  - Chip variants may want to impose different limits/policies
- The TLA? P2C? (Power/Perf Constraints)

Related issues

- Asymmetric CPU idle states
- Cluster awareness
  - E.g. perf tool
- Runtime PM: multiple states
  - Clock gating vs power gating
- CPU hotplug is slow
  - State between online and offline?
  - Or just make hotplug fast enough?
- CPU affinity vs hotplug
  - Virtual CPU IDs?
- Scheduler, timers, workqueues, cpufreq