arm

EAS in Mainline: Where we are

OSPM 2018

Dietmar Eggemann & Quentin Perret

Agenda

1) Introduction
2) Energy Aware Scheduling
3) Open issues
    a) Interface for the Energy Model / EAS
    b) Overutilization
    c) Algorithm complexity
    d) Impact of idling big CPUs

Agenda

1) **Introduction**
2) Energy Aware Scheduling
3) Open issues
    a) Interface for the Energy Model / EAS
    b) Overutilization
    c) Algorithm complexity
    d) Impact of idling big CPUs

What is EAS?

1.	Energy Model/ EAS wakeup path	Energy-aware task wakeup	v2 on lkml	Quentin/Dietmar
2.	Frequency and Cpu Invariant Engines	Enhance utilization signal quality	v4.15	Dietmar
3.	Idle Cpu PELT update (Remote status update)	Enhance utilization signal quality	v4.17-rc1	Vincent/(Brendan)
4.	Util est	Utilization signal for task at wakeup	v4.17-rc1	Patrick
5.	Util clamp	Userspace QoS for tasks	v1 on lkml	Patrick
6.	Misfit Task	Correct wrong wakeup decisions in load-balance	v2 on lkml	Morten/Valentin
7.	Dynamic topology flag detection	SD_ASYM_CPUCAPACITY	Android v4.14	Morten

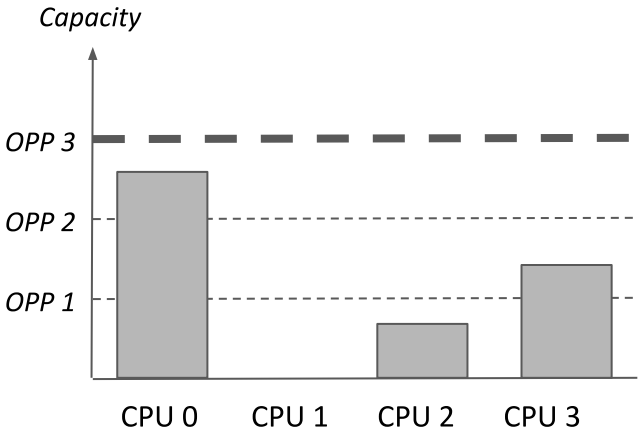
Why abandon the per sched domain Energy Model?

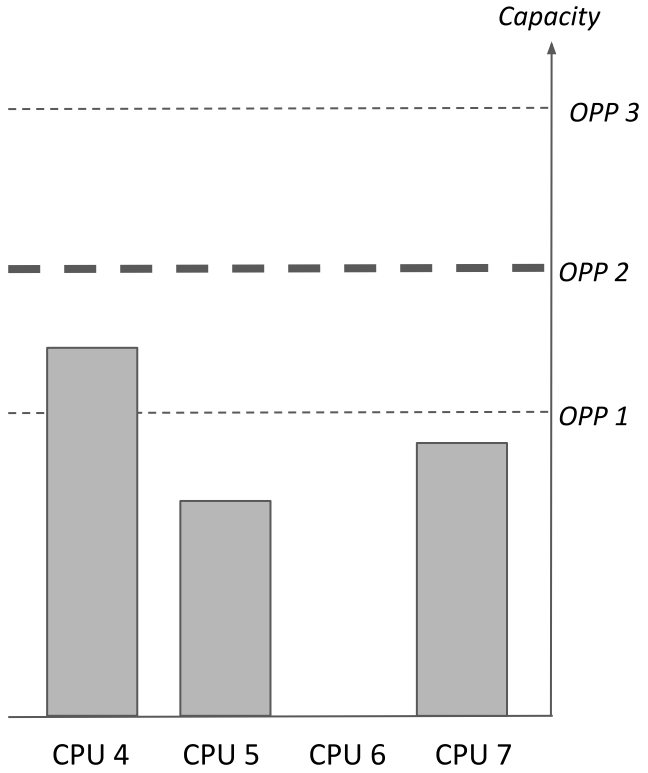
1. Cluster active energy almost neglectable on modern SOCs
2. Biggest contributor to possible energy savings is the difference in uArch
3. Not able to forecast future idle states in a quality needed for the EAS algorithm
4. Mapping sched domains other than MC (core) and DIE (cluster) is not beneficial
5. Scheduler domains are no longer congruent to frequency domains (Arm DynamIQ)
6. Easy deployment strategy for an CPU-only EM for Arm via PM_OPP library and DT CPU property dynamic-power-coefficient

Agenda

1) Introduction
2) **Energy Aware Scheduling**
3) Open issues
    a) Interface for the Energy Model / EAS
    b) Overutilization
    c) Algorithm complexity
    d) Impact of idling big CPUs

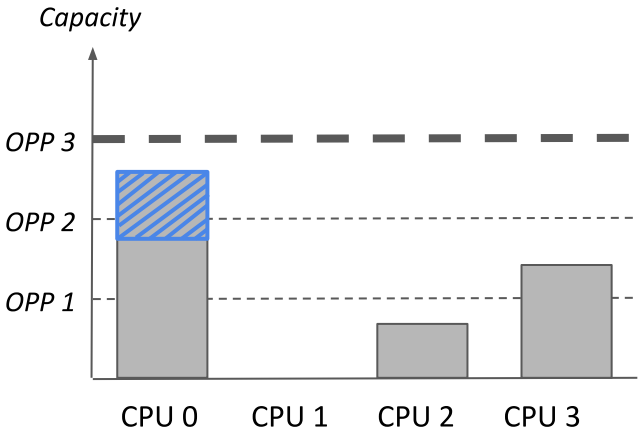
Energy Aware Scheduling

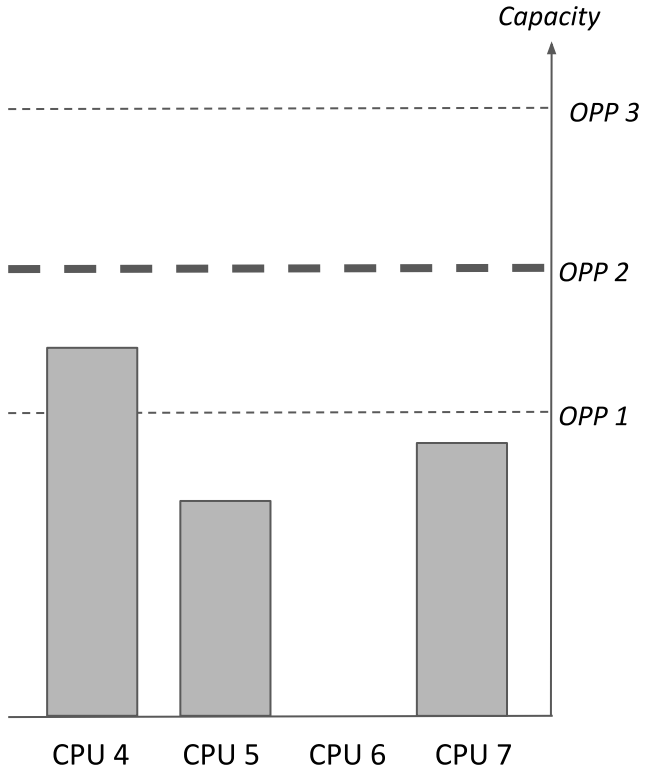




arm

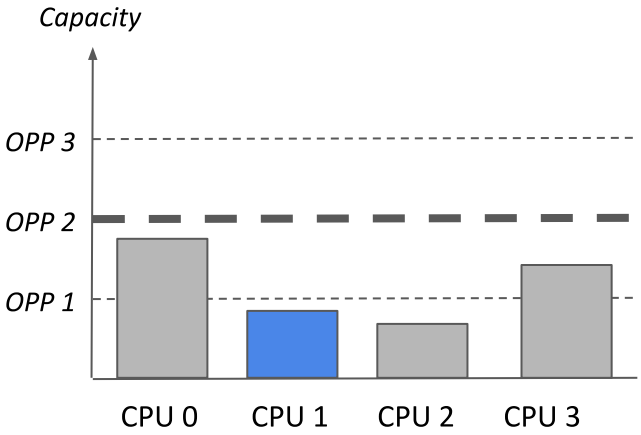
Energy Aware Scheduling

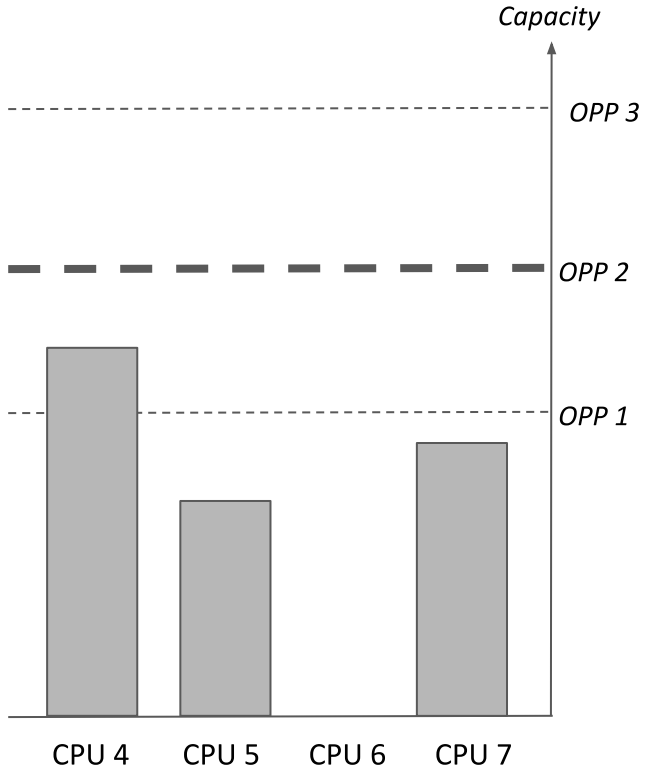




arm

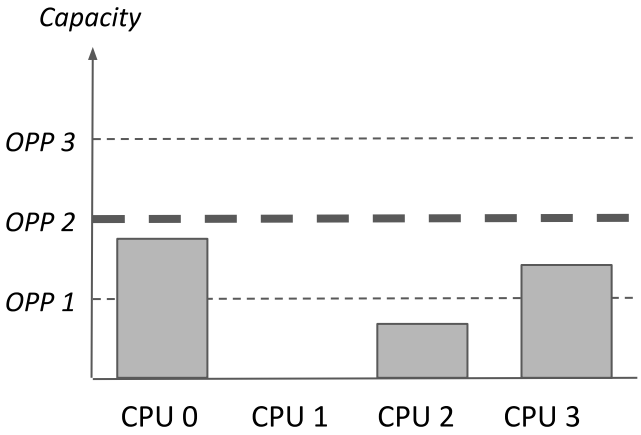
Energy Aware Scheduling

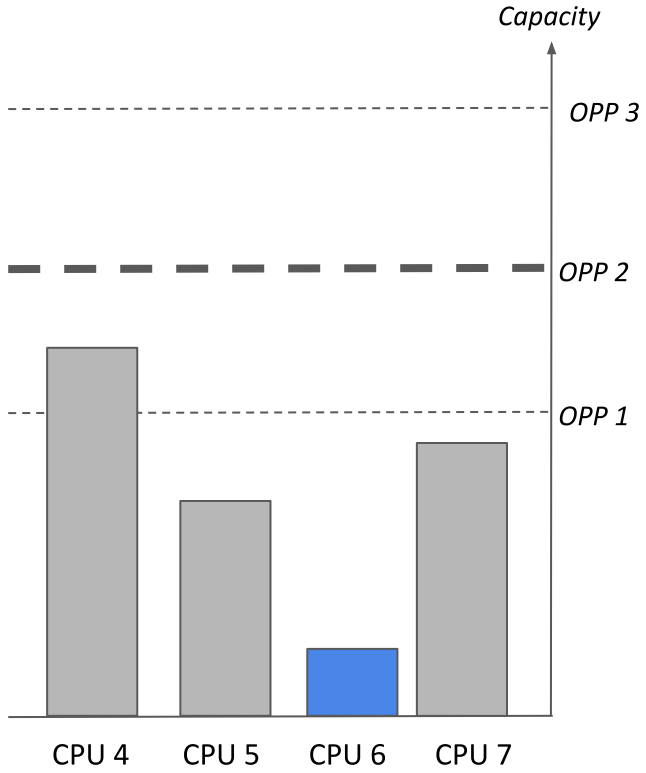




arm

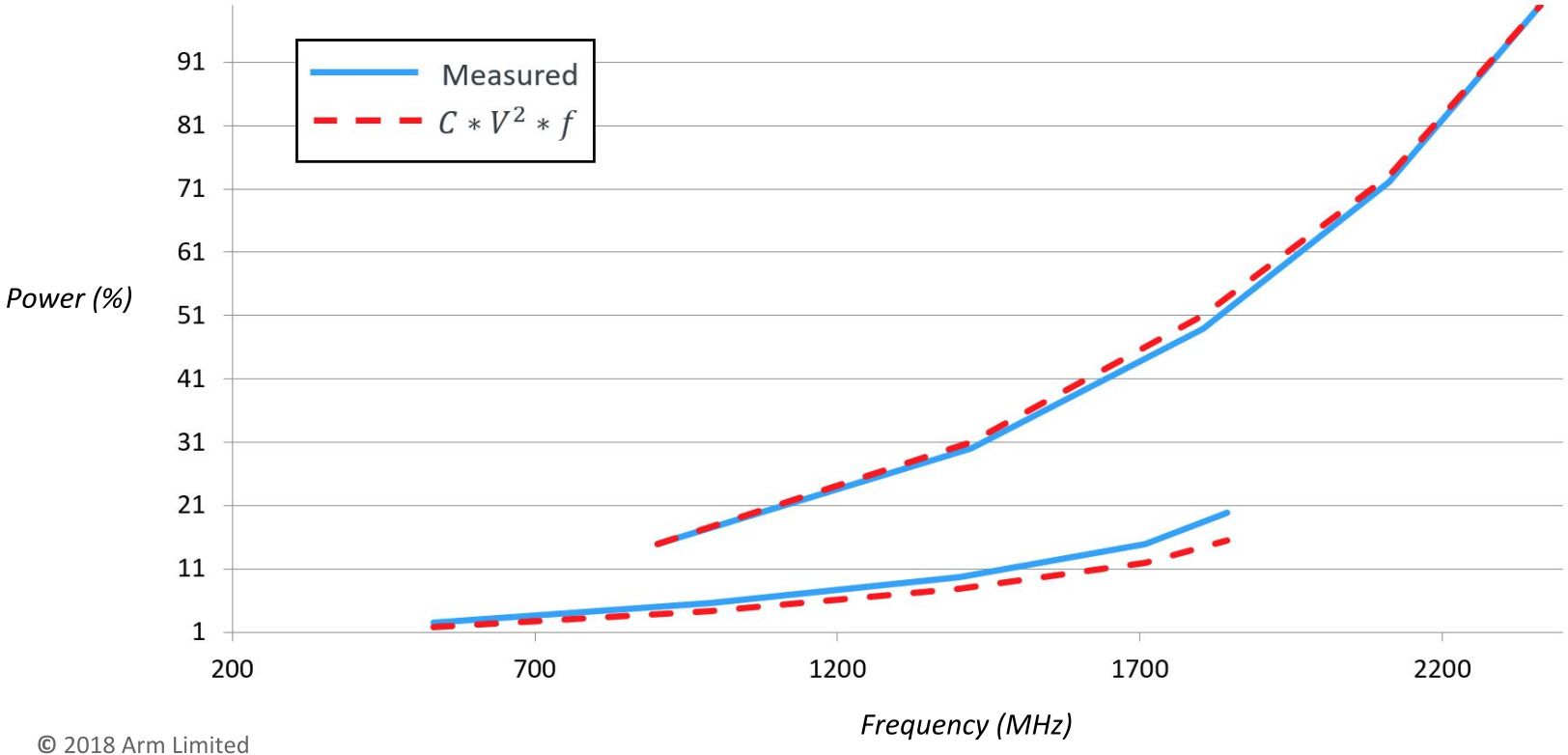
Energy Aware Scheduling





arm

Energy Aware Scheduling

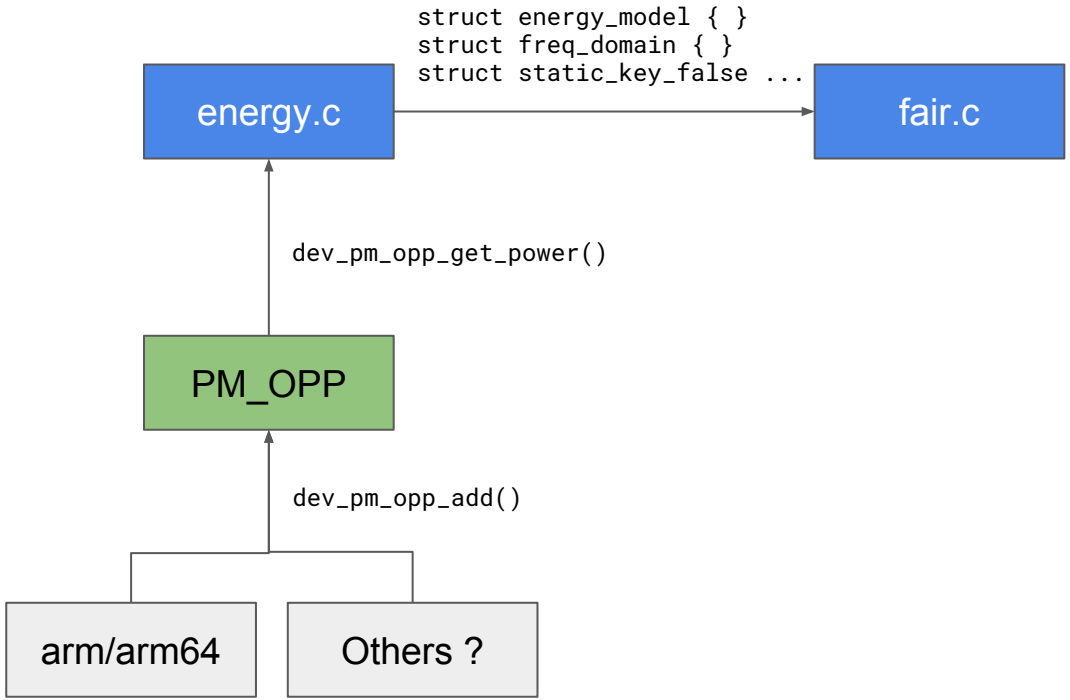


arm

Agenda

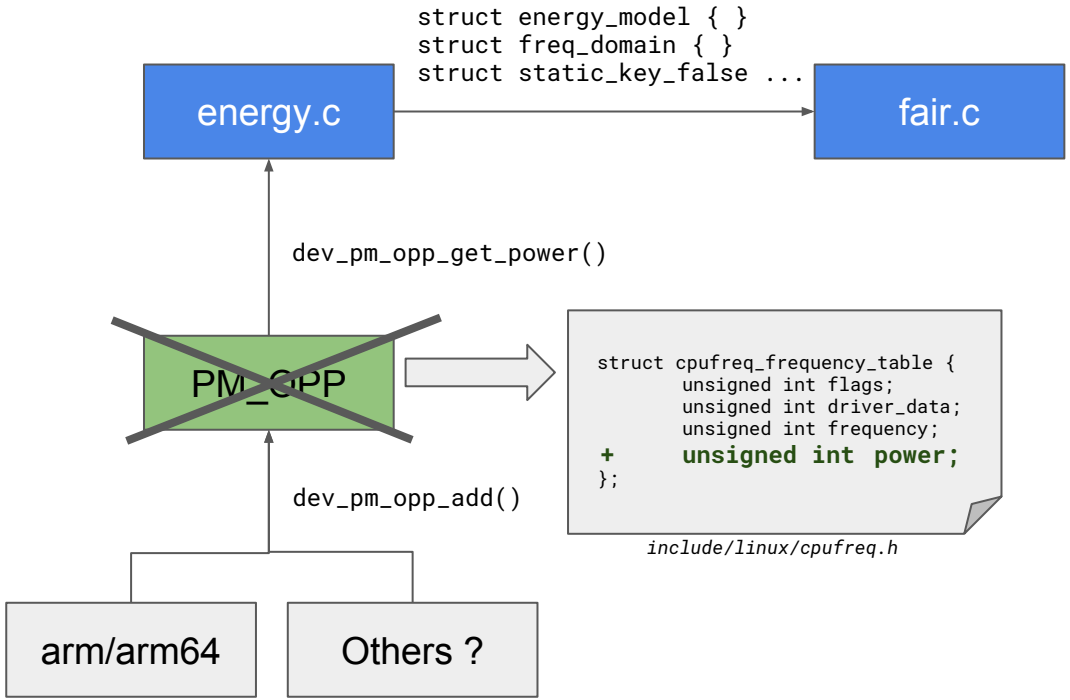
1) Introduction
2) Energy Aware Scheduling
3) **Open issues**
    a) **Interface for the Energy Model / EAS**
    b) Overutilization
    c) Algorithm complexity
    d) Impact of idling big CPUs

Issue 1: Interface for the energy model / EAS



arm

Issue 1: Interface for the energy model / EAS

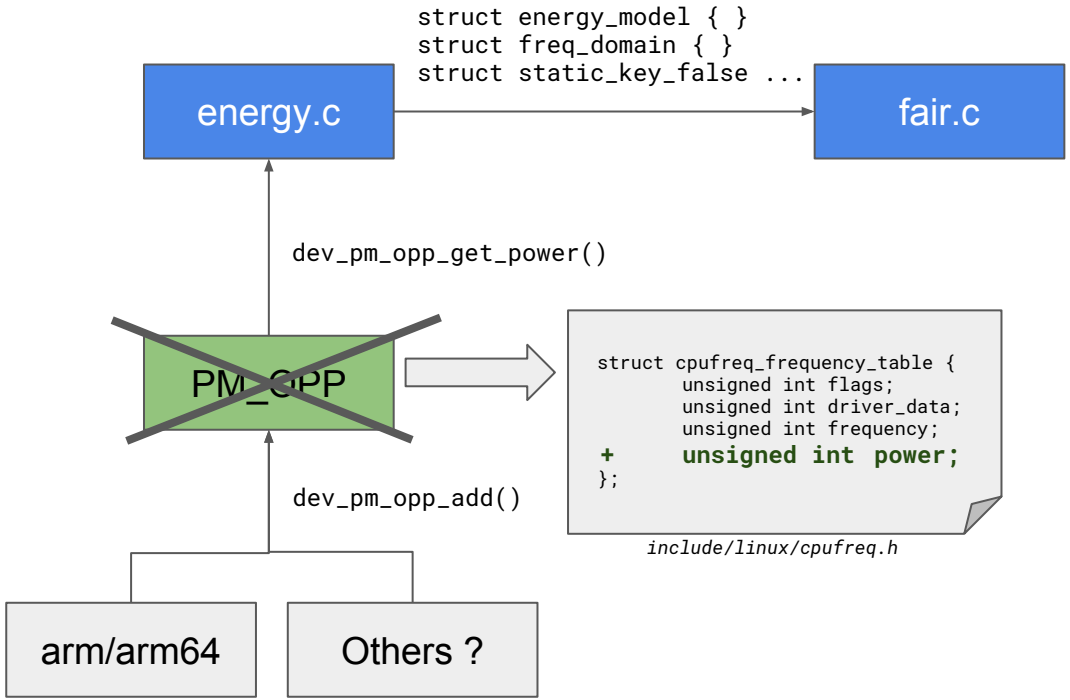


arm

Issue 1: Interface for the energy model / EAS

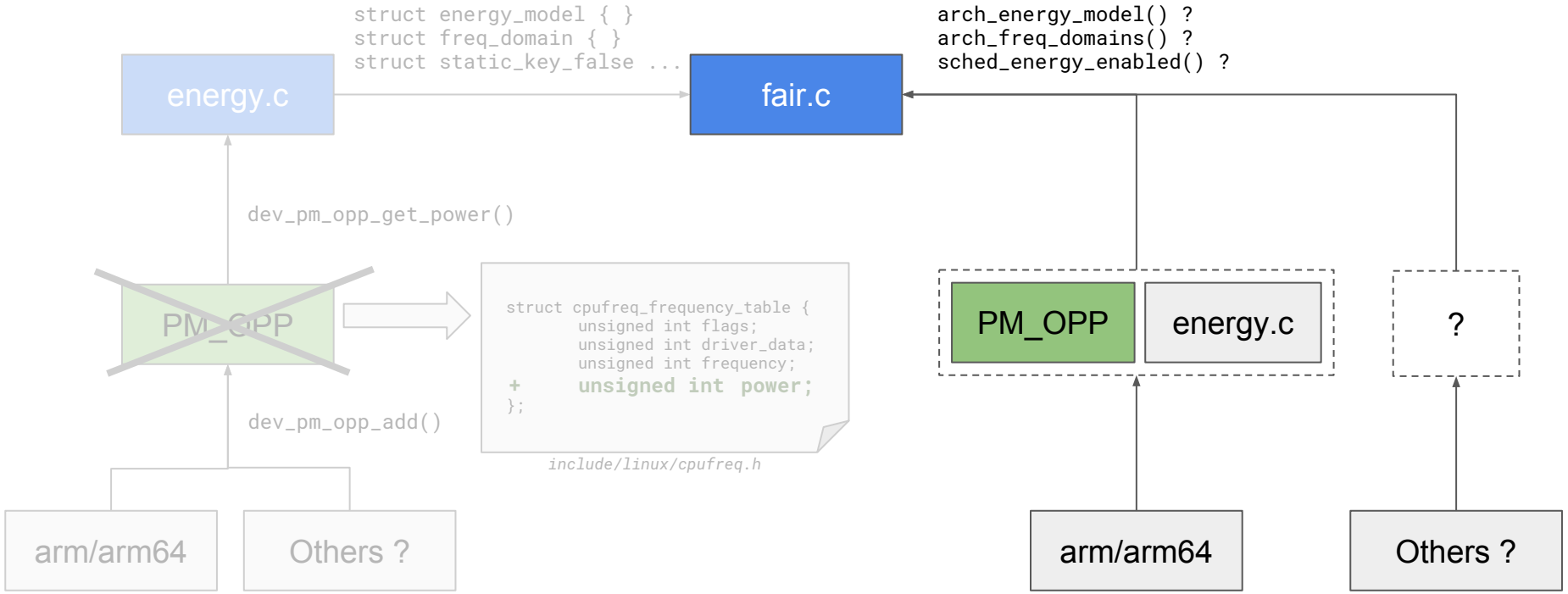
- Factor out part of `cpufreq_schedutil.c::get_next_freq()`
- Forecast OPP with `cpufreq_table_find_index_l()` (as in `cpufreq_driver_resolve_freq()`)
- Index `policy->freq_table` directly from the scheduler ?

Issue 1: Interface for the energy model / EAS



arm

Issue 1: Interface for the energy model / EAS



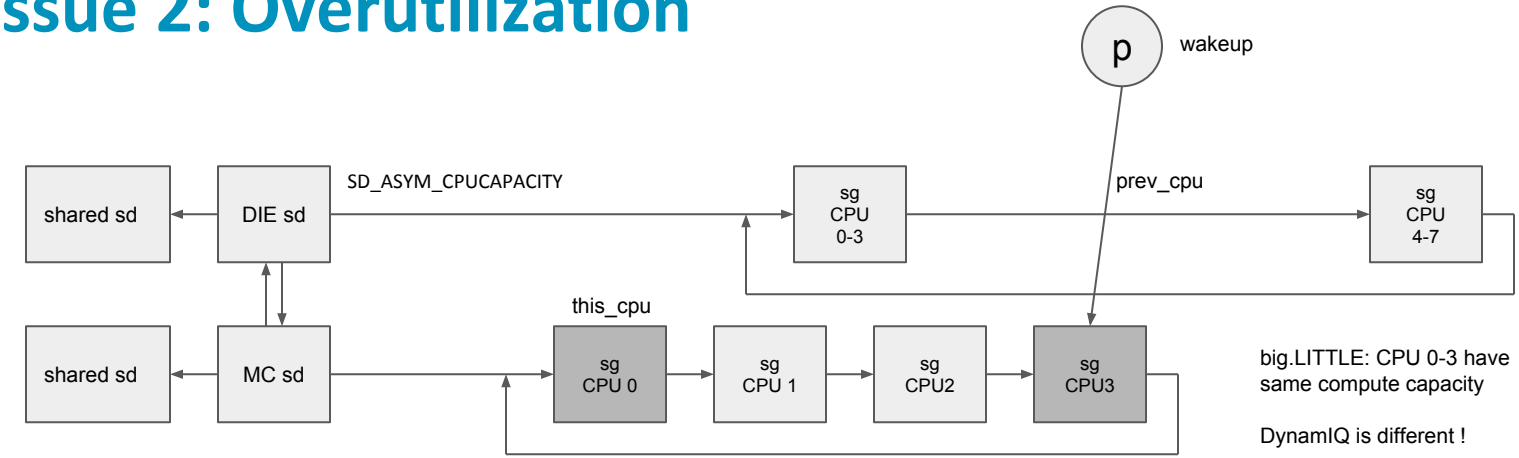
arm

Agenda

1) Introduction
2) Energy Aware Scheduling
3) **Open issues**
    a) Interface for the Energy Model / EAS
    b) Overutilization
    c) Algorithm complexity
    d) Impact of idling big CPUs

arm

Issue 2: Overutilization



select_task_rq_fair():

(1) wake_energy()

```
if ( … is_overutilized ( cpu_rq ( prev_cpu ) -> sd ) )
    return false
```

(2) for_each_domain (..., tmp)

```
if ( …. ! is_overutilized ( tmp ) )
    energy_sd = tmp
```

(3) find_energy_efficient_cpu (**energy_sd**, ...)

How to implement “overutilization”?

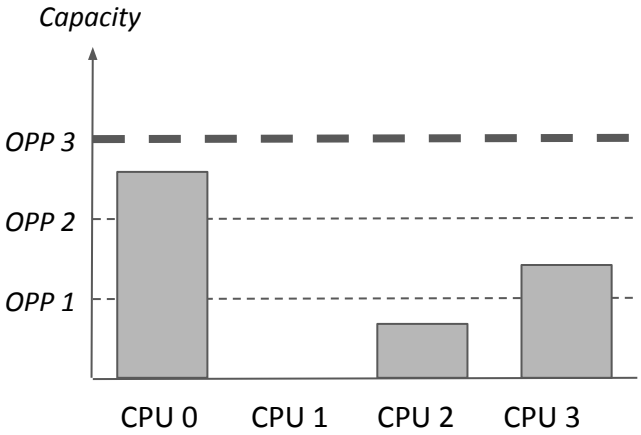
1. Hierarchical per-sd
2. Selective per-sd (cached sd_eas) -> (sis() and sd_llc)
3. System-wide (per root-domain)

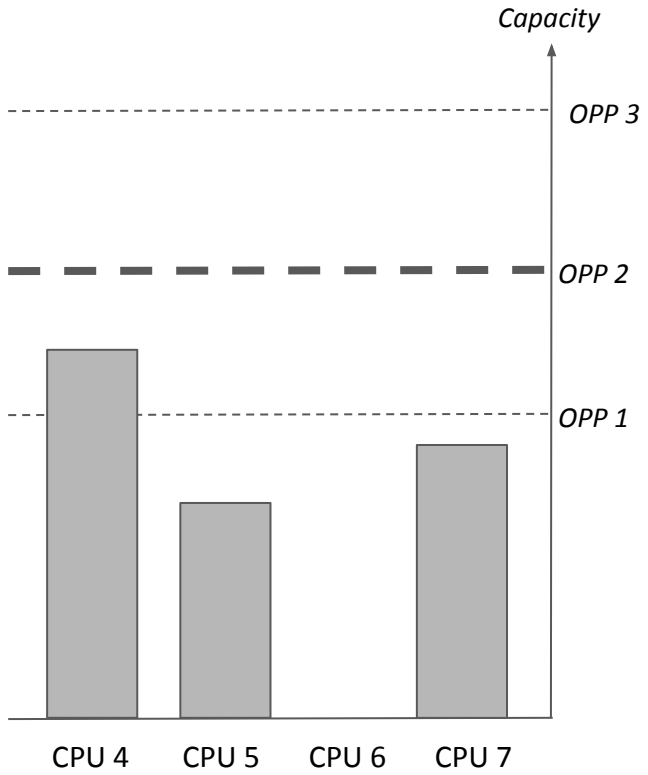
arm

Agenda

1) Introduction
2) Energy Aware Scheduling
3) **Open issues**
   a) Interface for the Energy Model / EAS
   b) Overutilization
   c) **Algorithm complexity**
   d) Impact of idling big CPUs

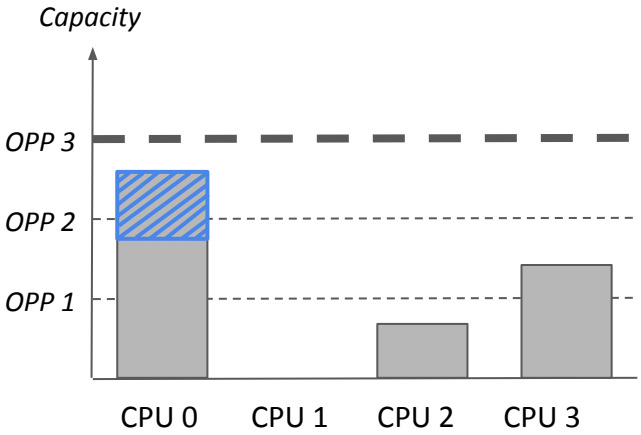
Issue 3: Algorithm complexity

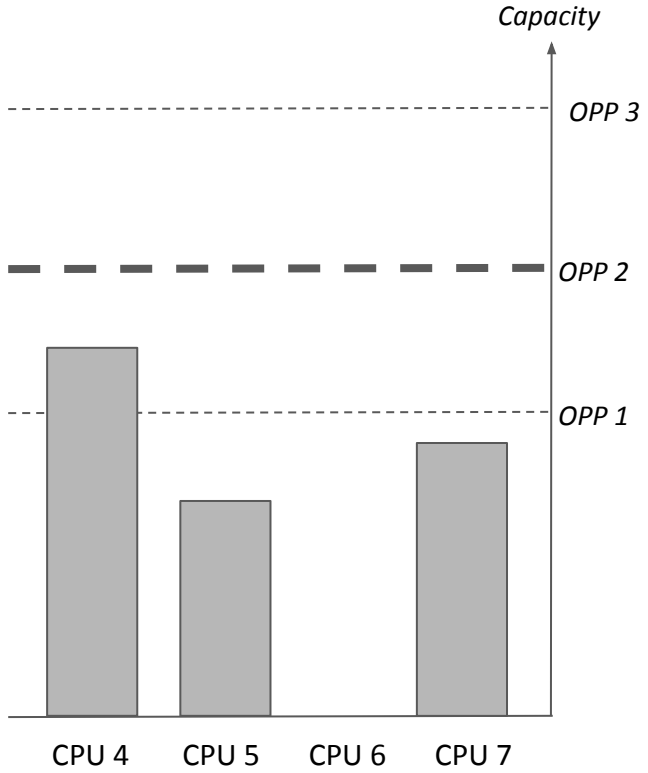




arm

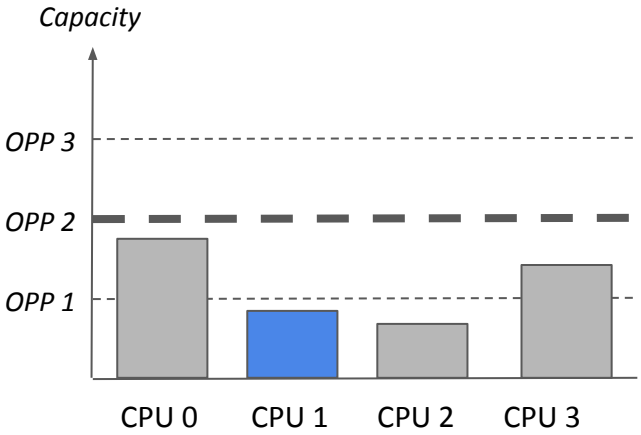
Issue 3: Algorithm complexity

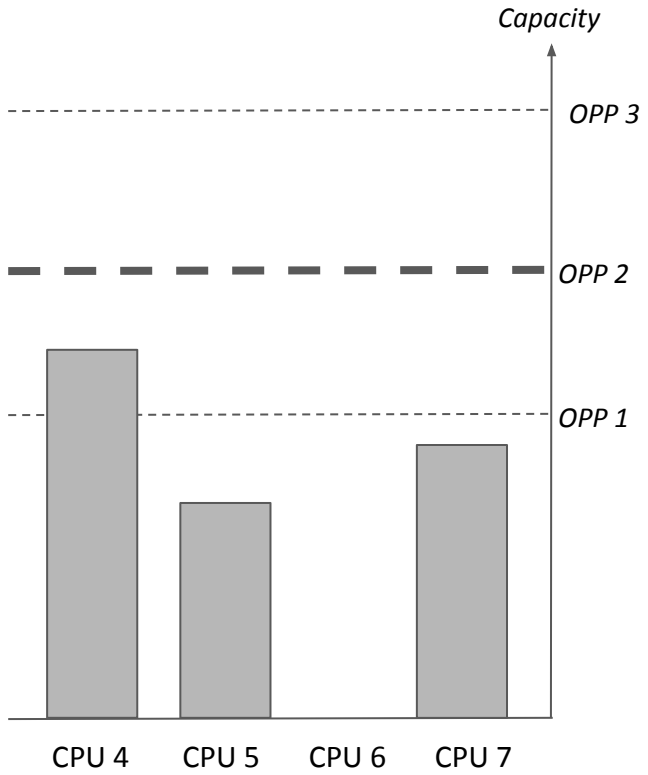




arm

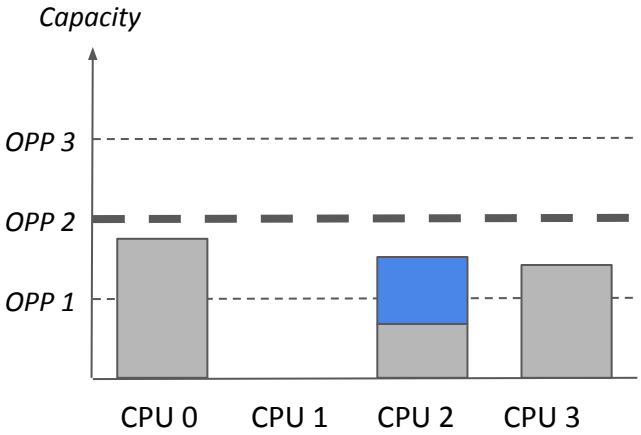
Issue 3: Algorithm complexity

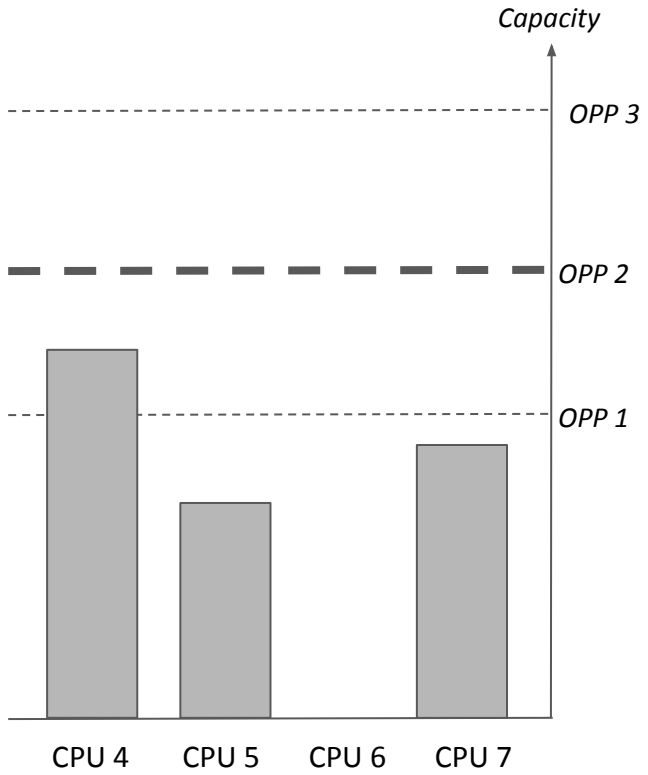




arm

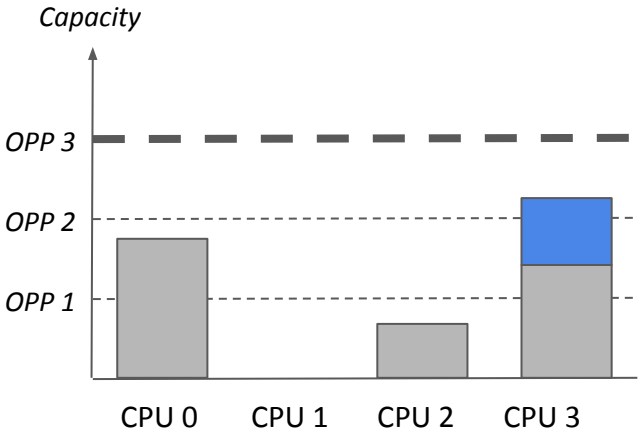
Issue 3: Algorithm complexity

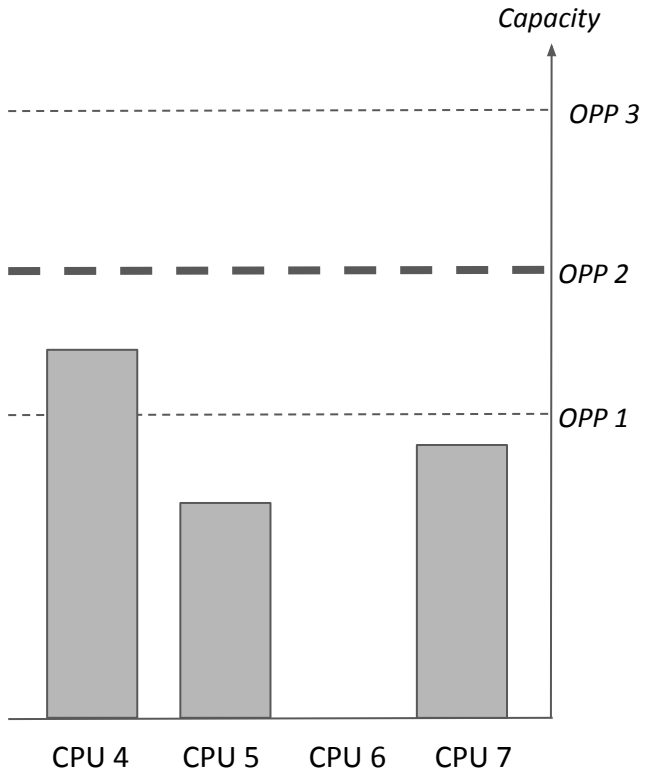




arm

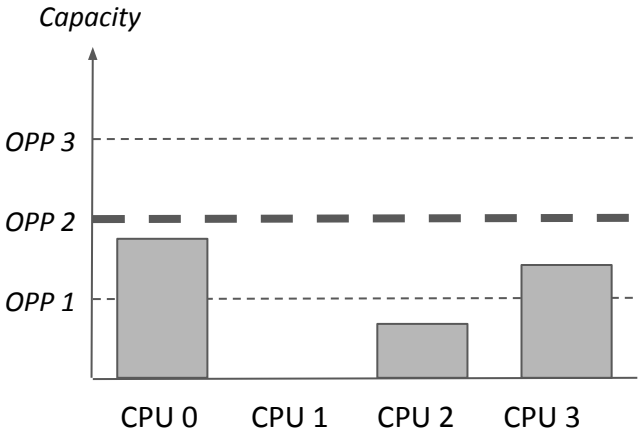
Issue 3: Algorithm complexity

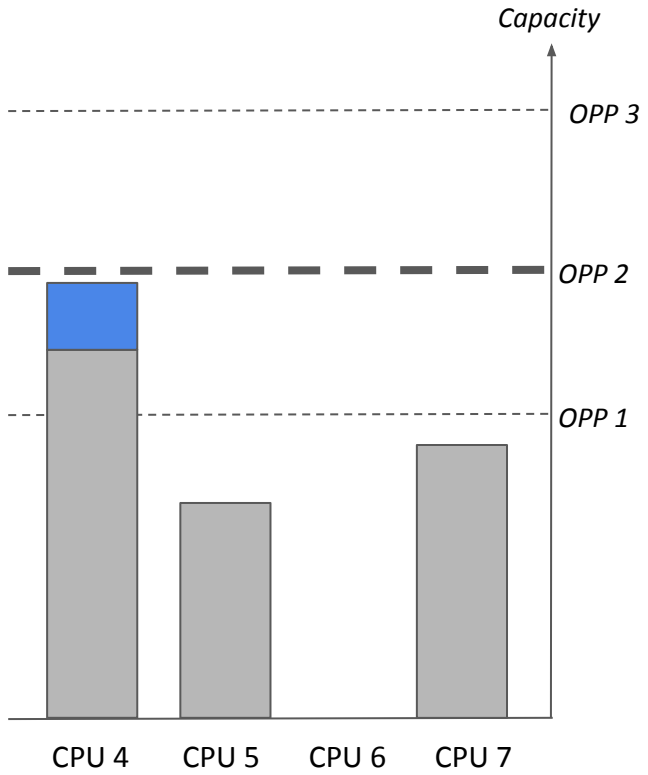




arm

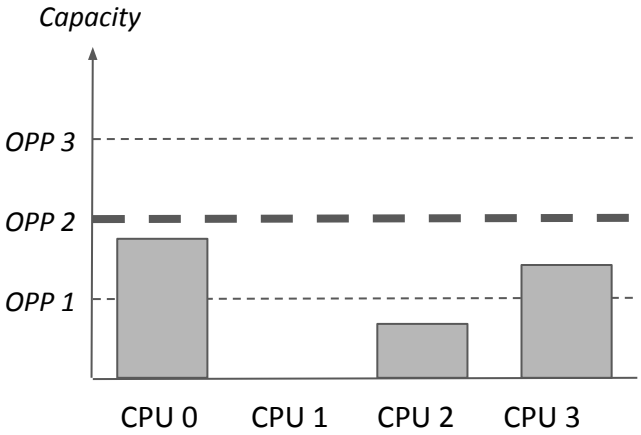
Issue 3: Algorithm complexity

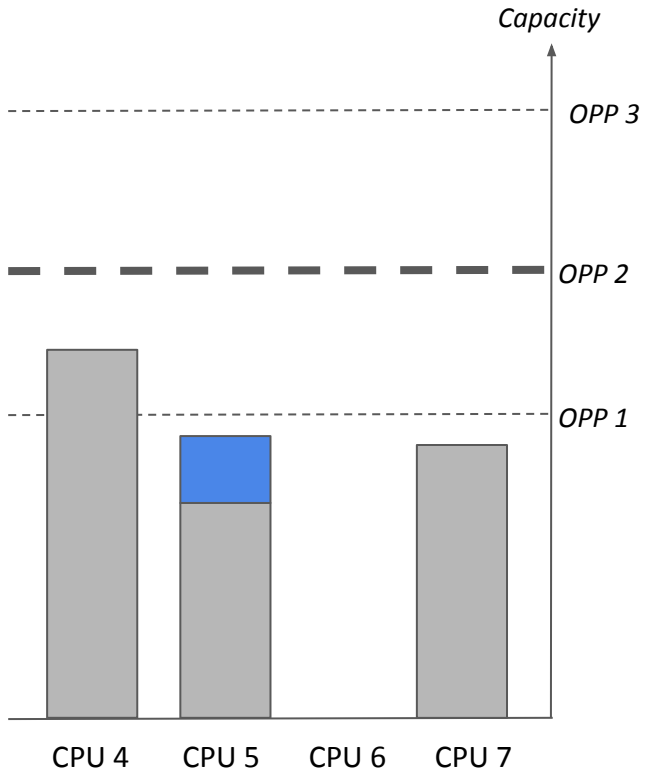




arm

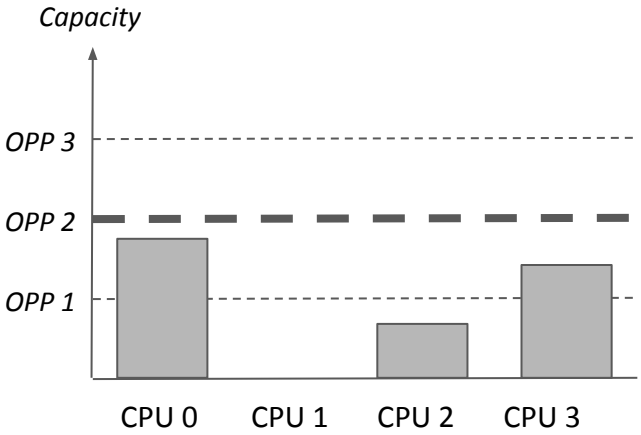
Issue 3: Algorithm complexity

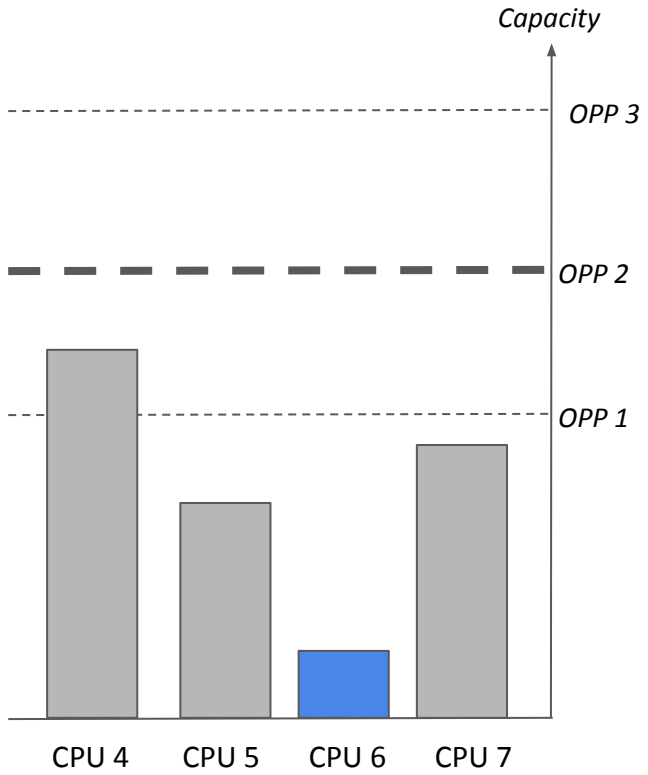




arm

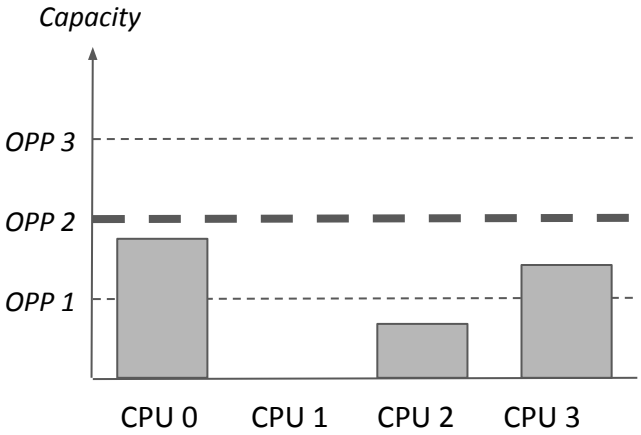
Issue 3: Algorithm complexity

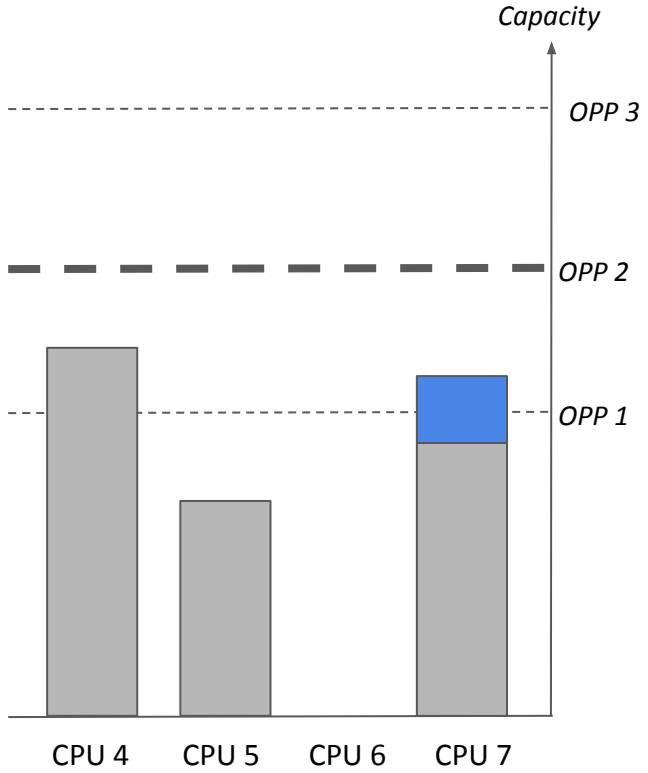




arm

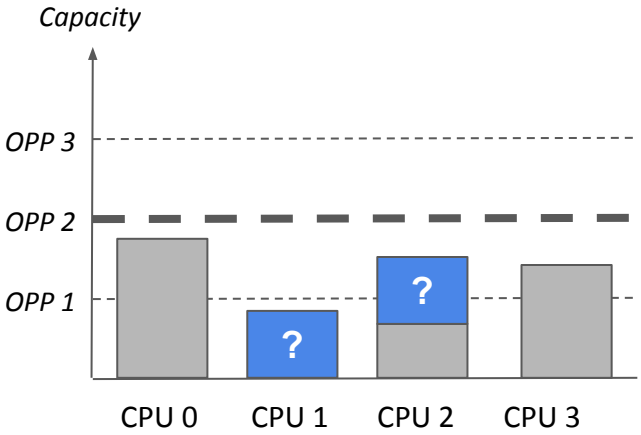
Issue 3: Algorithm complexity

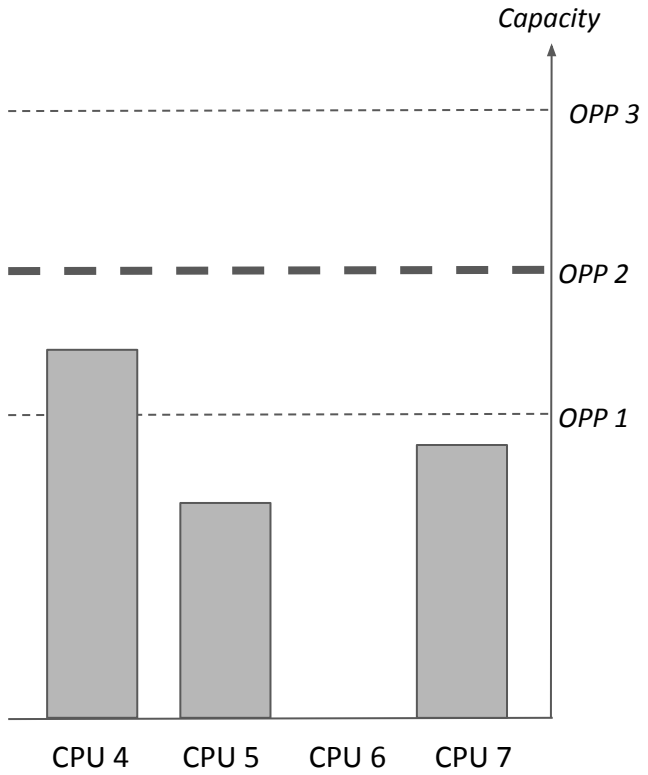




arm

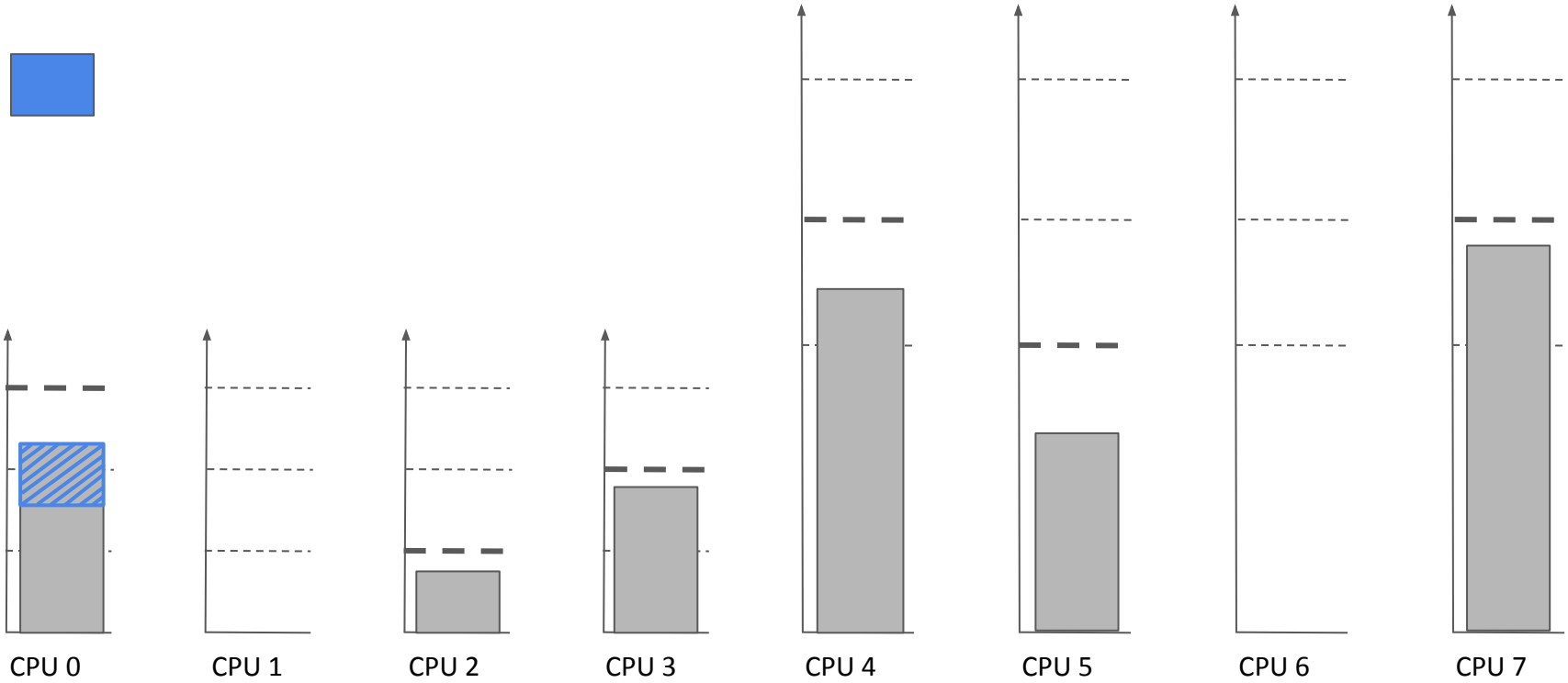
Issue 3: Algorithm complexity





arm

Issue 3: Algorithm complexity



arm

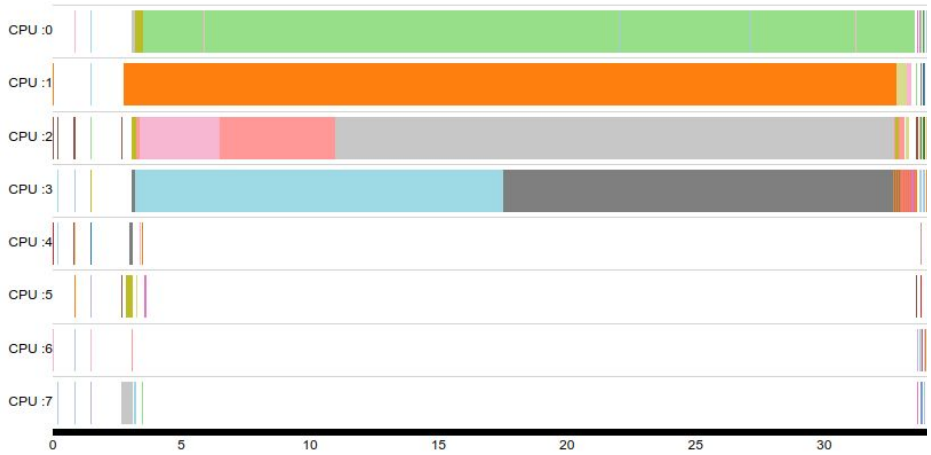
Agenda

1) Introduction
2) Energy Aware Scheduling
3) **Open issues**
   a) Interface for the Energy Model / EAS
   b) Overutilization
   c) Algorithm complexity
   d) **Impact of idling big CPUs**

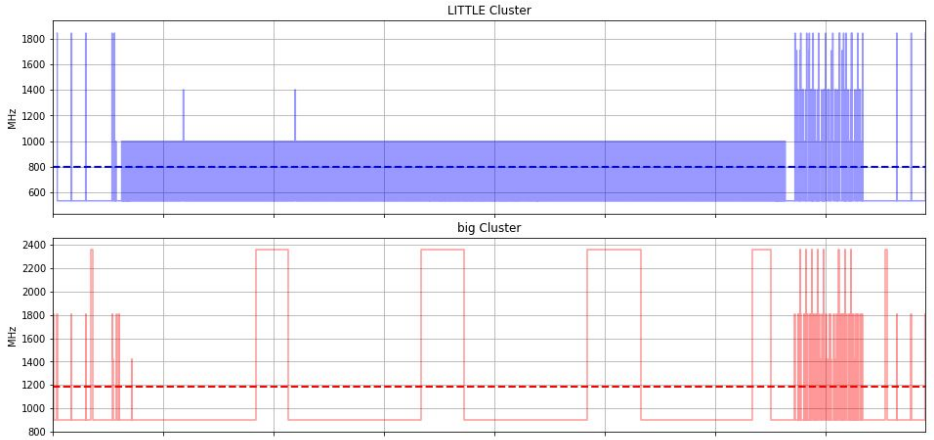
arm

Issue 4: Impact of idling big CPUs

Hikey960 (CPU0-3 LITTLE, CPU4-7 big)



Task placement



Frequency selection

10 RTApp tasks, 5% duty cycle, 30 seconds

arm