Energy-aware Scheduling

국민대학교 임베디드 연구실
경 주 현

Outline

- Summarize two documents
- “Energy-aware Scheduling”
- “EAS Overview and Integration Guide”

Motivation

- **Hardware topologies are becoming more varied**
- **accommodating different power/performance budgets:**
  - SMP, multi-cluster SMP, ARM big.LITTLE technology.
  - Per core/per cluster DVFS
- **Linux power management frameworks are uncoordinated**
  - Hard to tune for different topologies.

Power Saving Scheduling

- Measure all CPUs' load
- Choose a CPU from available CPUs
- Decide the idlest CPU
- Migrate tasks from the idlest CPU to the chosen CPU

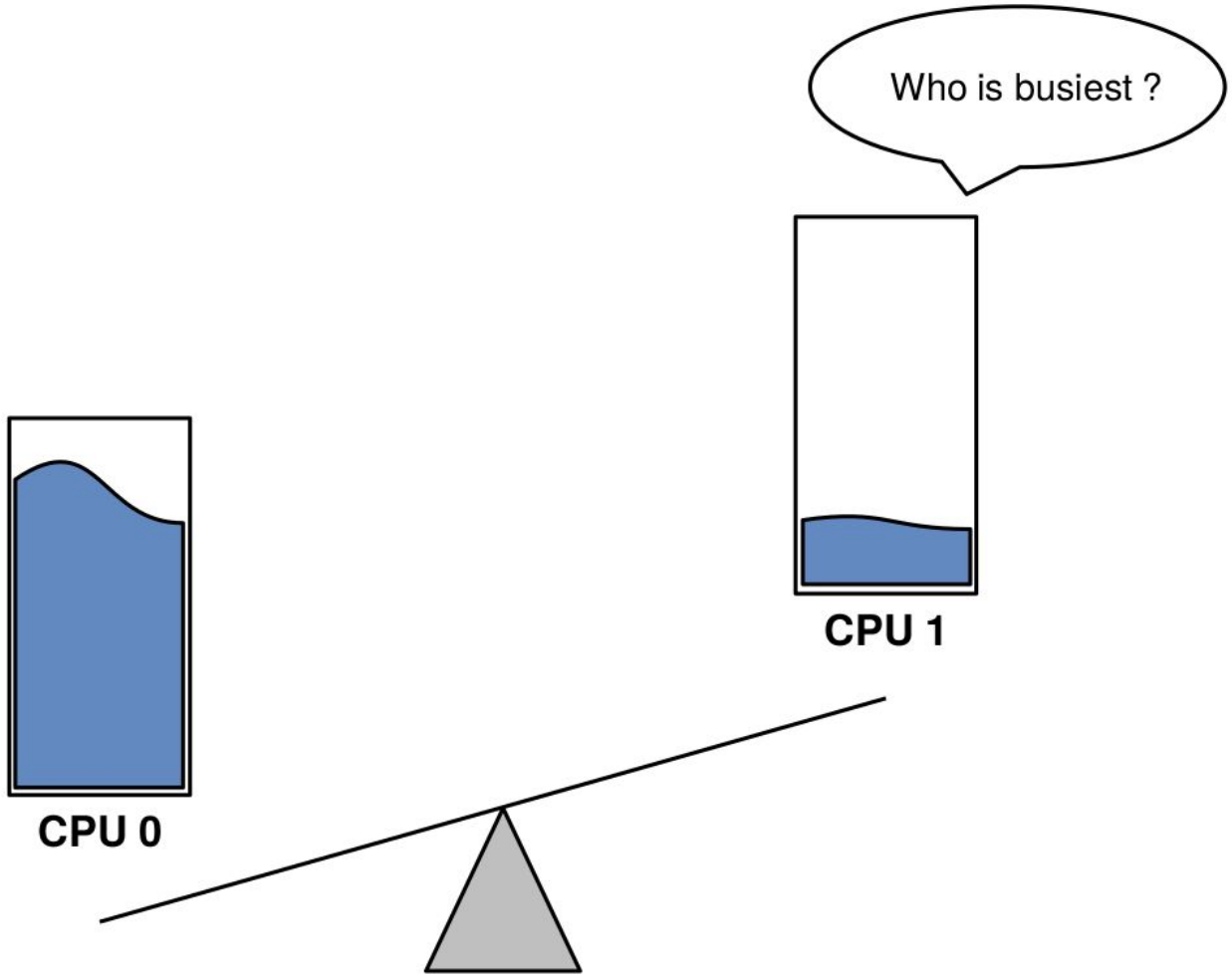
State of the art document for EAS.

- "Energy Aware Scheduling", Byungchul Park, LG Electronic
- “EAS Overview and Integration Guide”, ARM

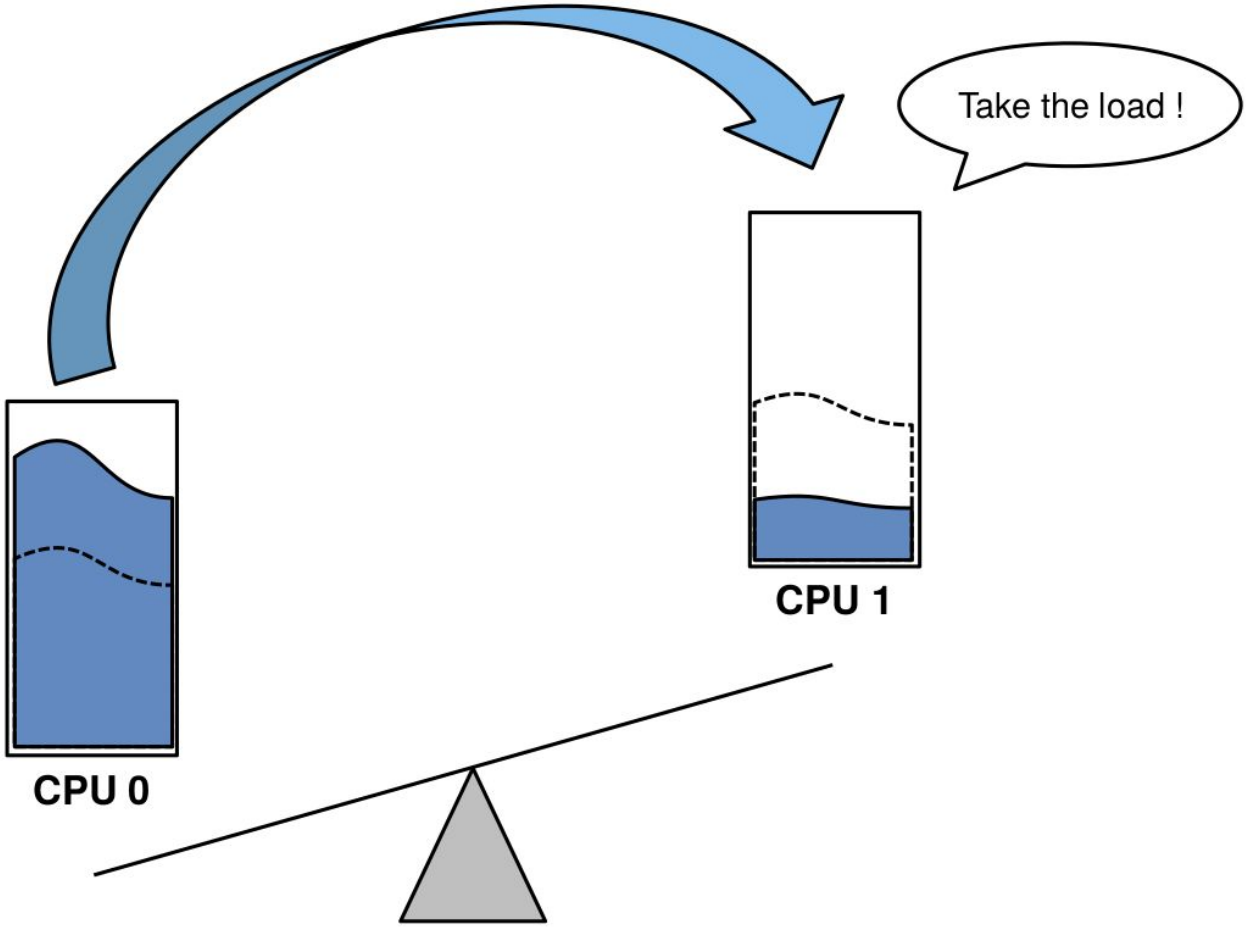
"Energy Aware Scheduling",
Byungchul Park,
LG Electronic

reference link: Linux Foundation Events

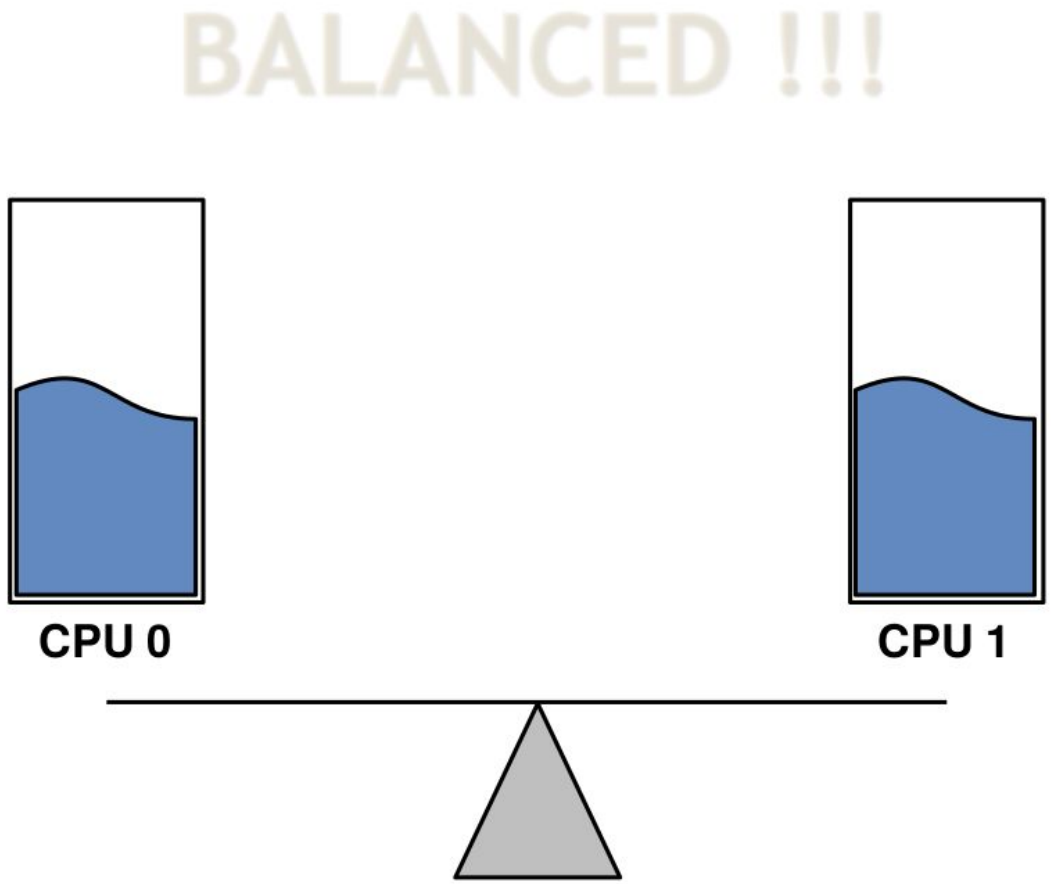
Load Balancing - migration



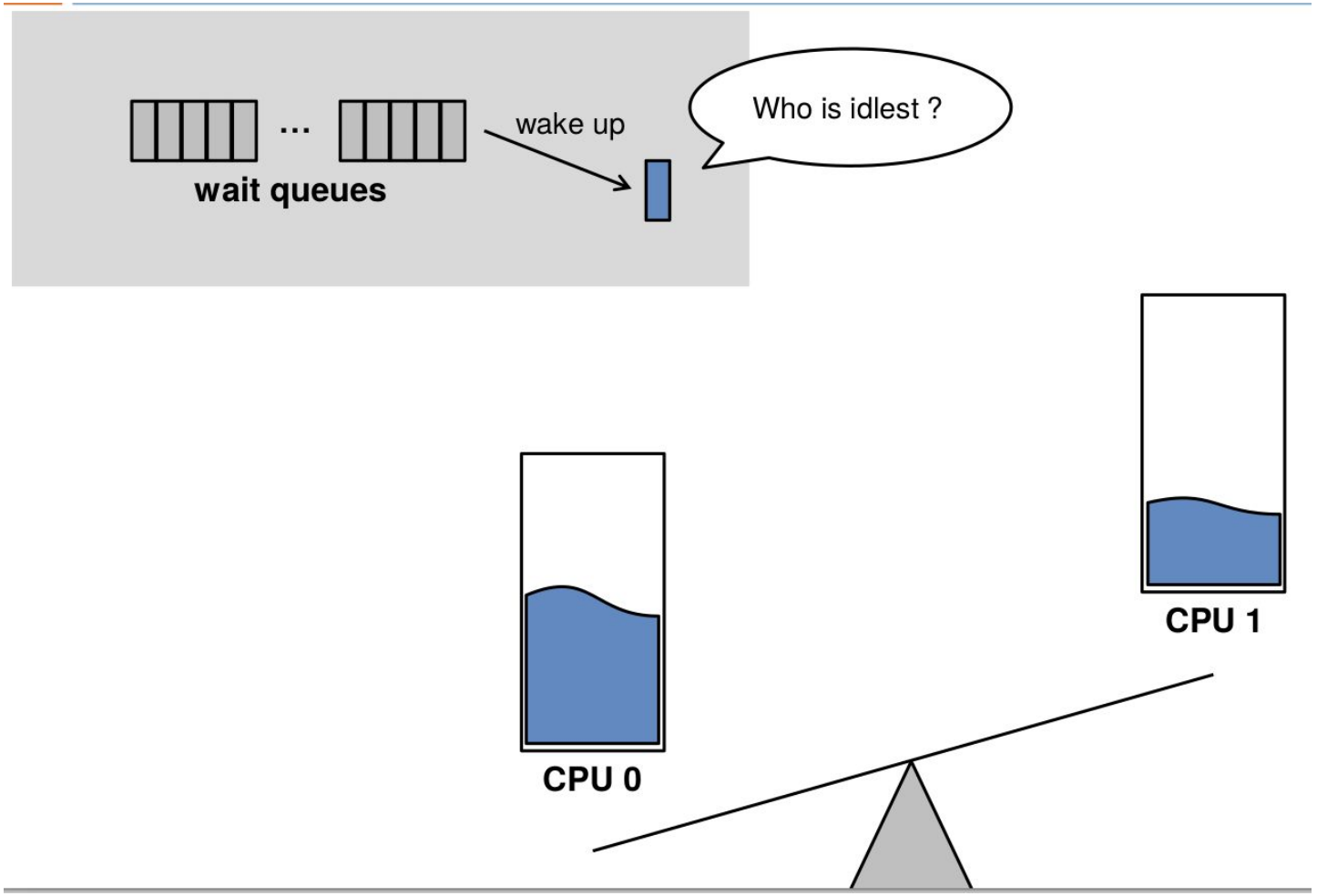
Load Balancing - migration



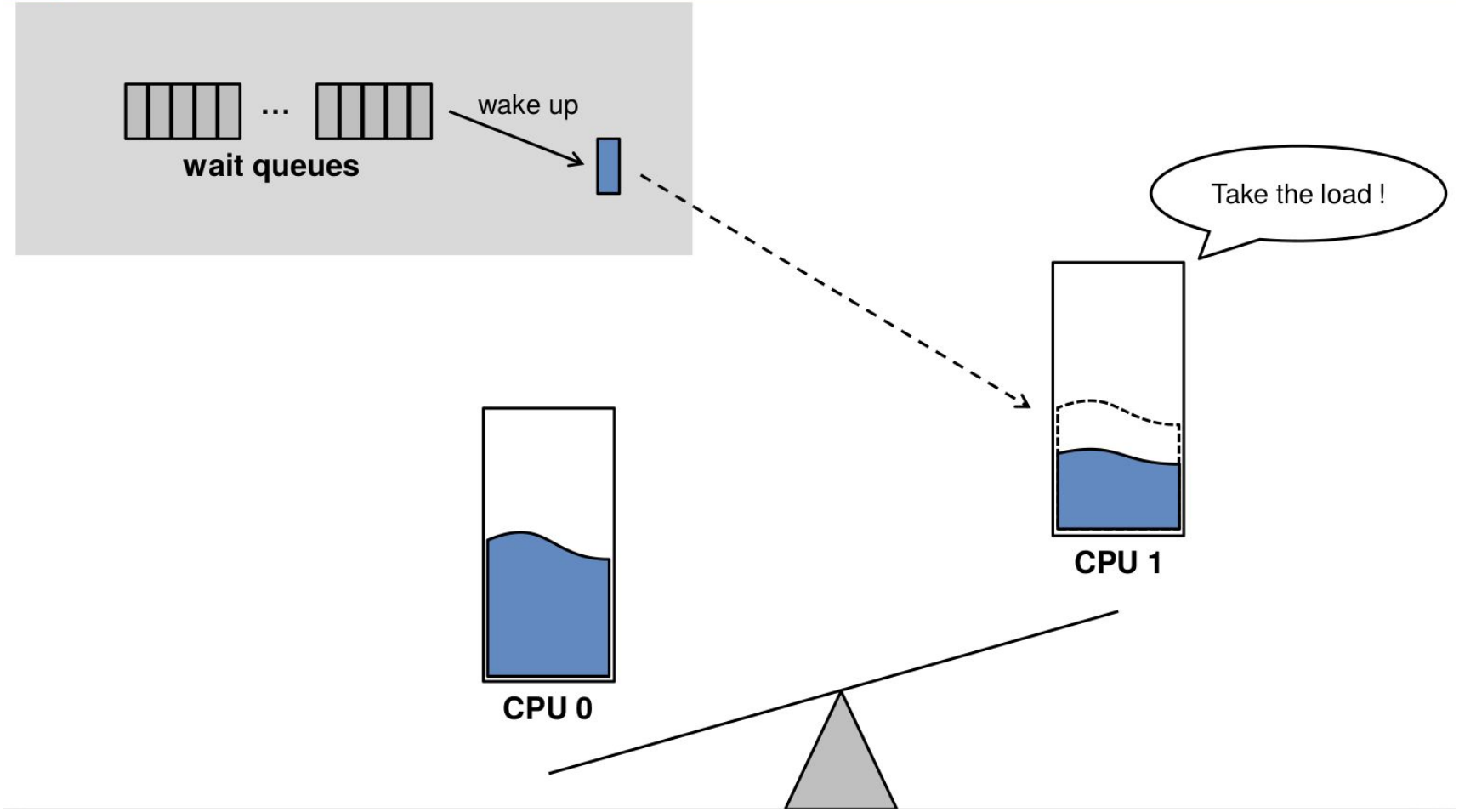
Load Balancing - migration



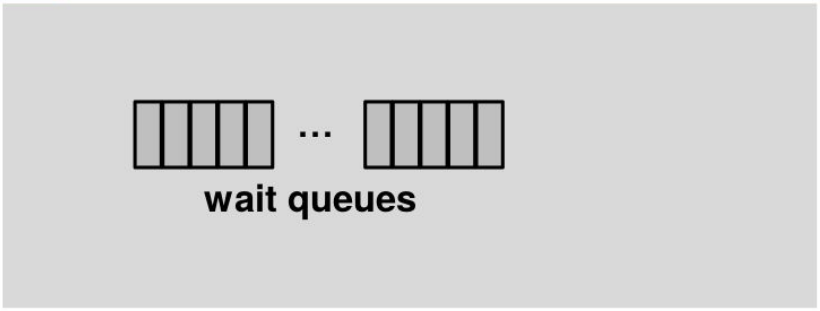
Load Balancing - wake-up



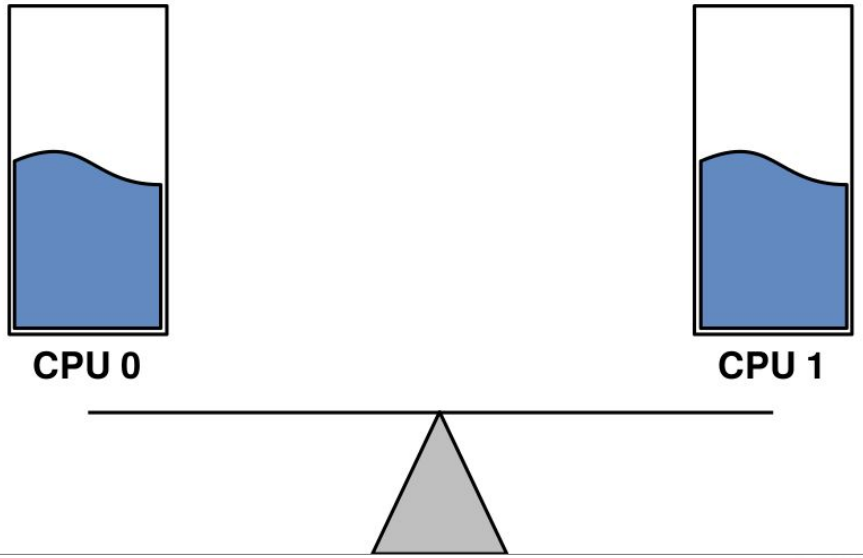
Load Balancing - wake-up



Load Balancing - wake-up



BALANCED !!!

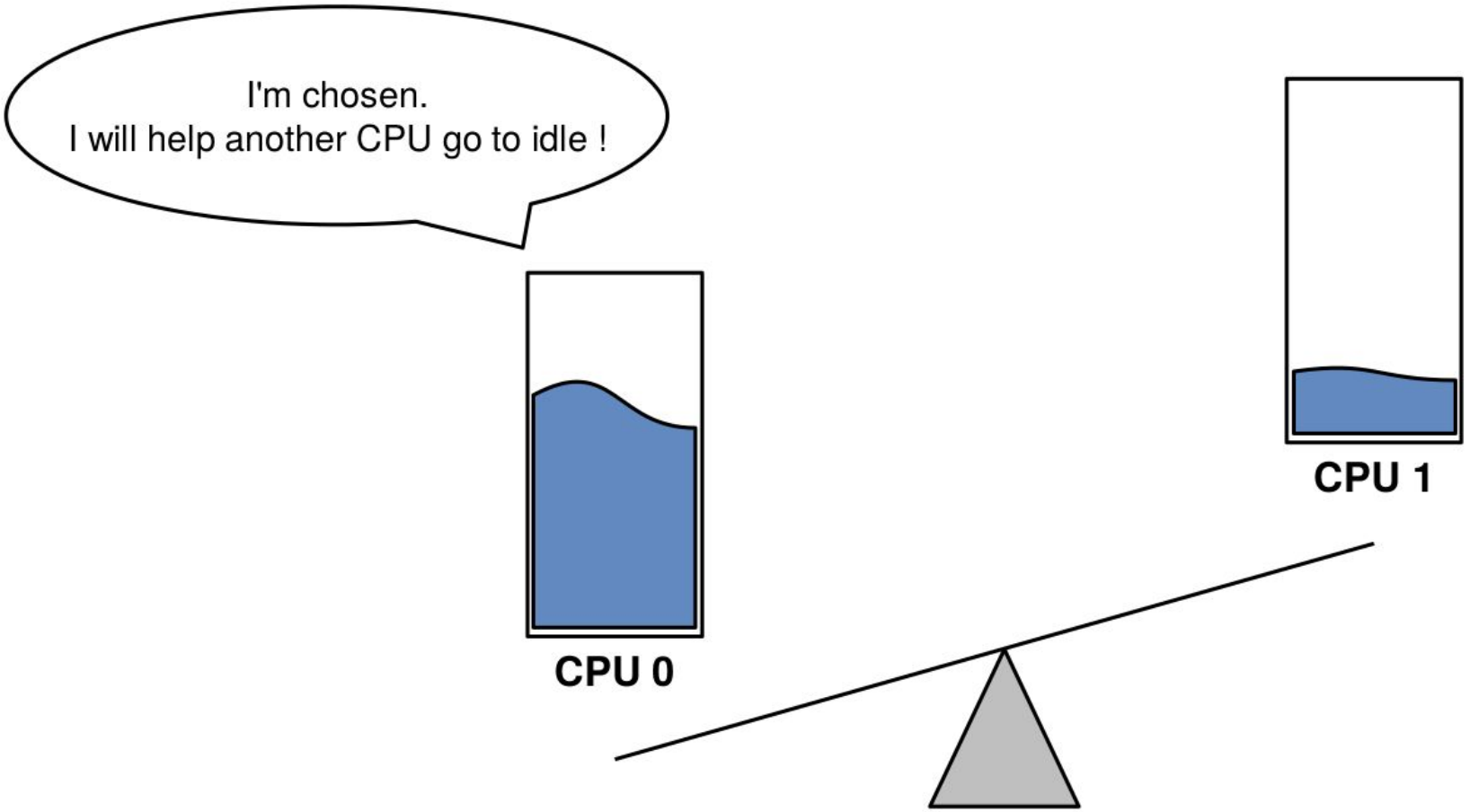


Problem

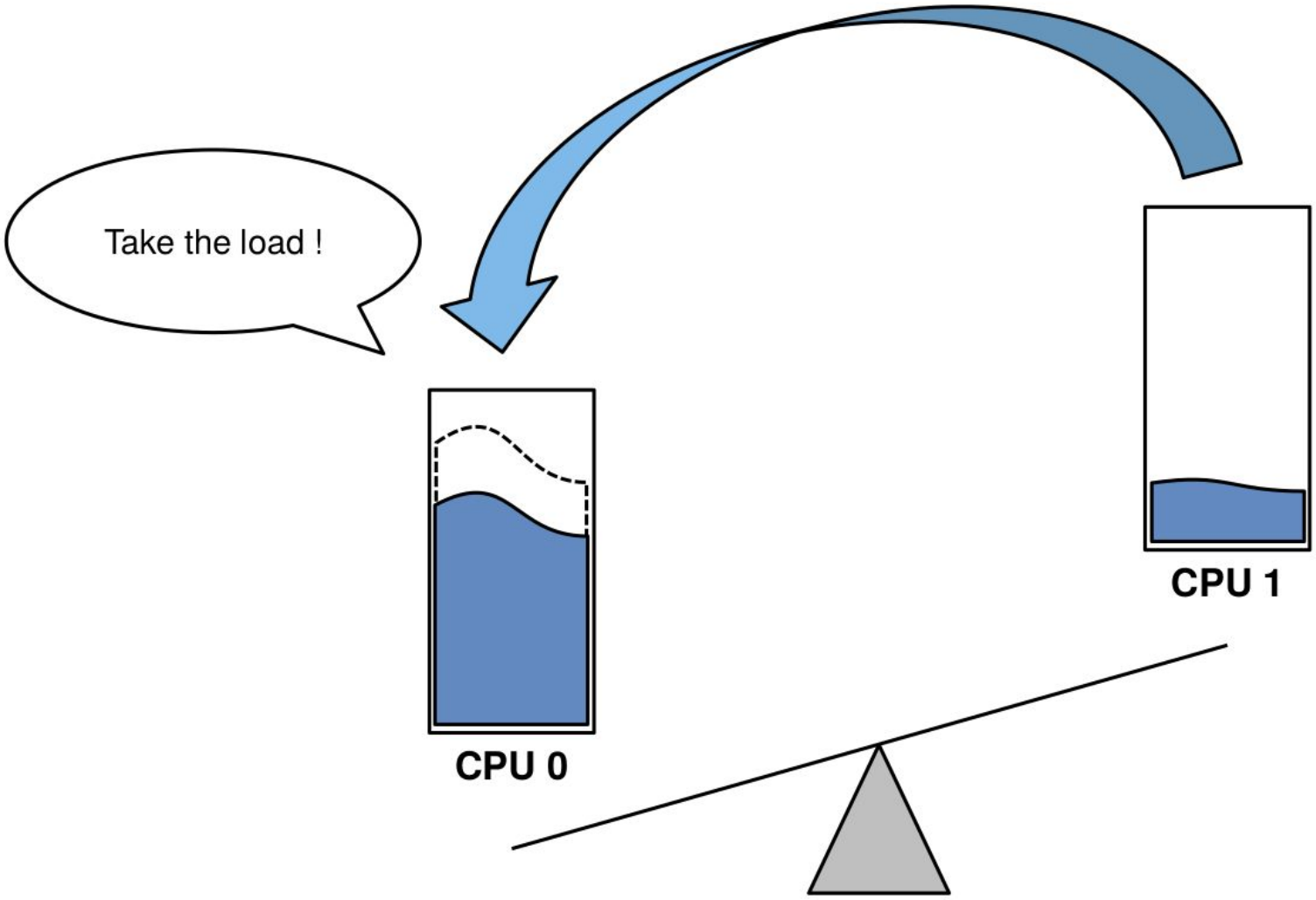
- **Only focused on performance**

- **Spreading approach**
  - Guarantee maximum performance and minimum latency

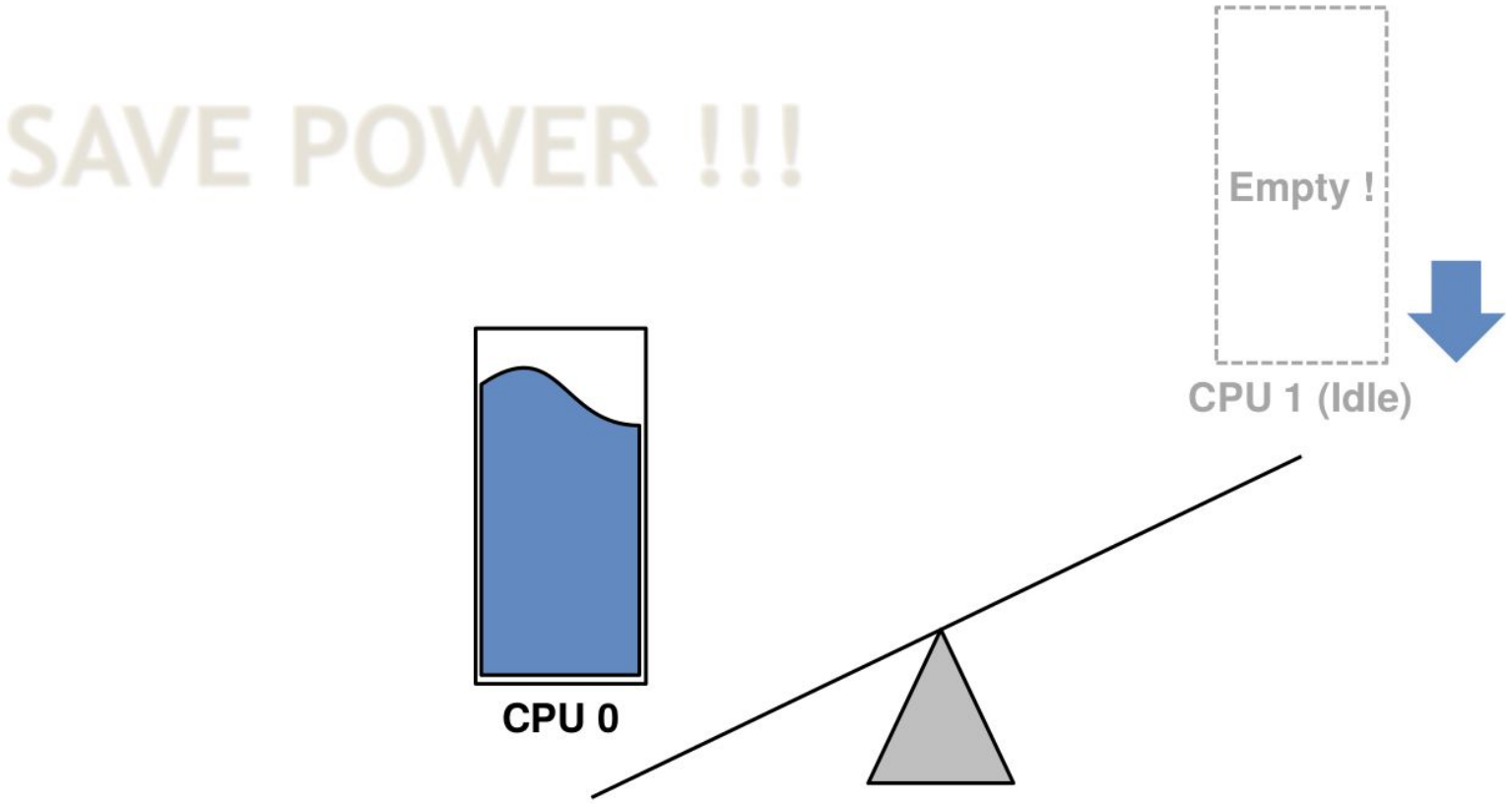
Power Saved Load Balancing - migration



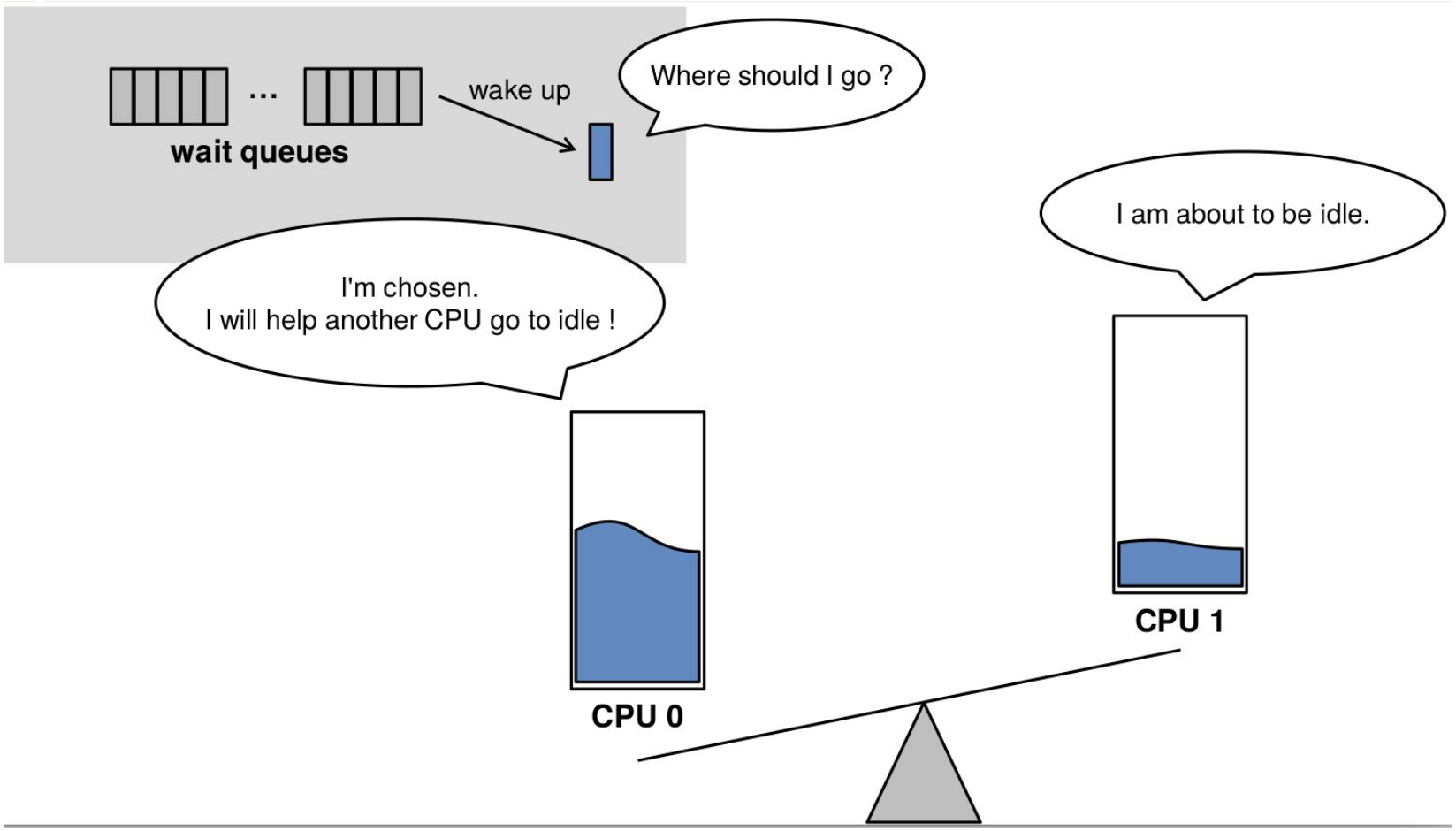
Power Saved Load Balancing - migration



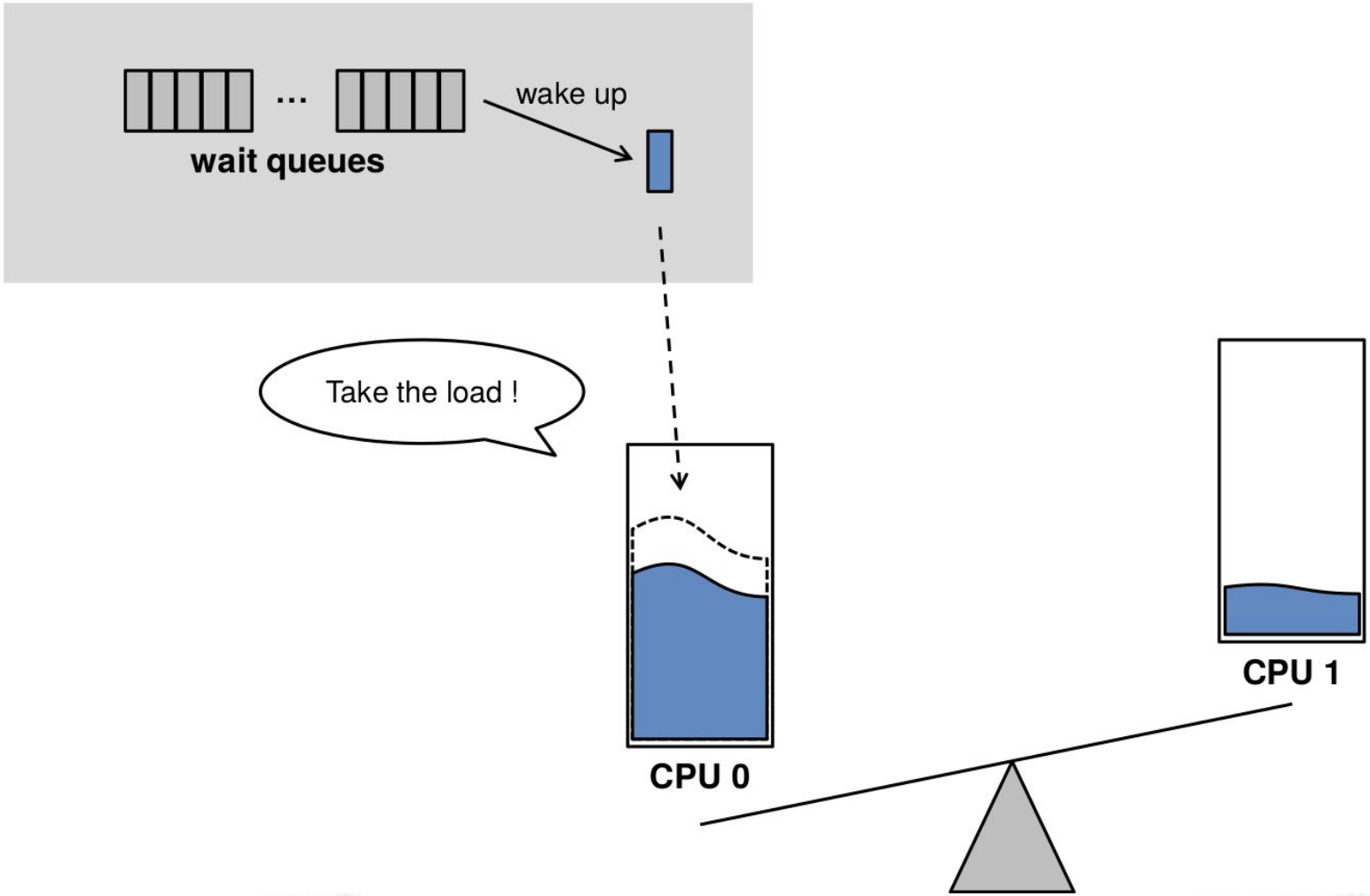
Power Saved Load Balancing - migration



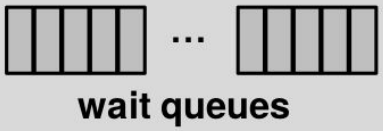
Power Saved Load Balancing - wake-up

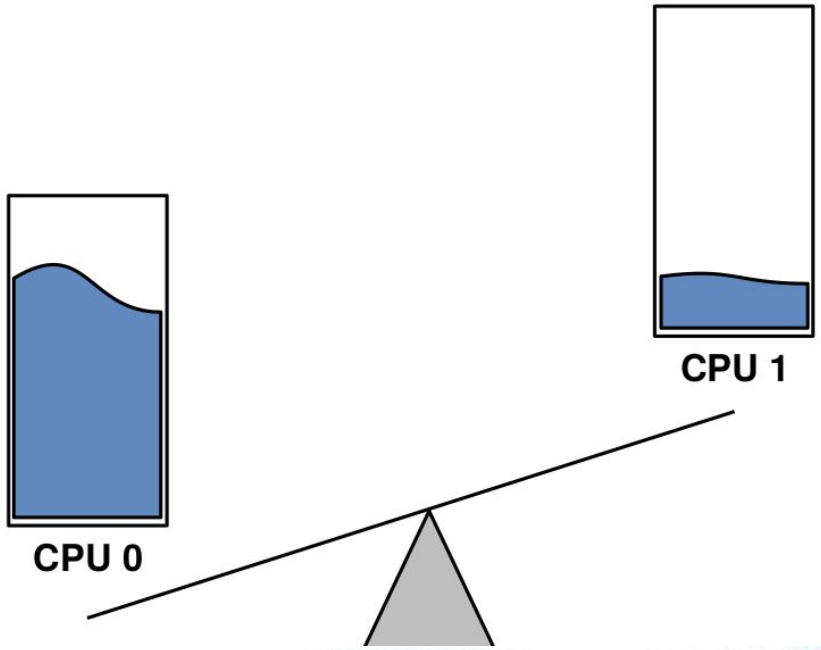


Power Saved Load Balancing - wake-up

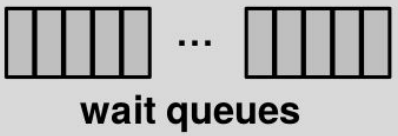


Power Saved Load Balancing - wake-up

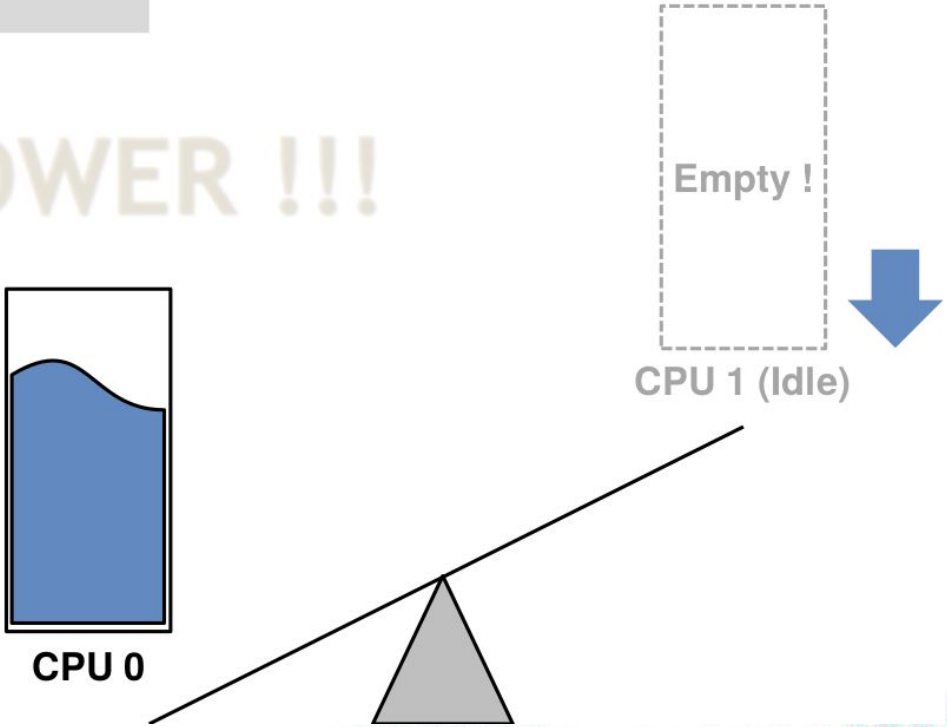




Power Saved Load Balancing - wake-up

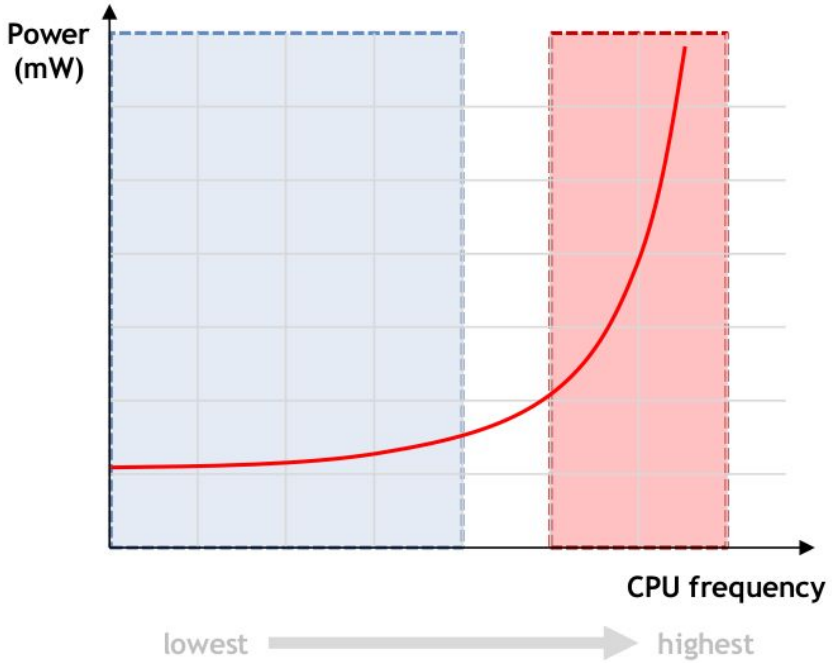


SAVE POWER !!!

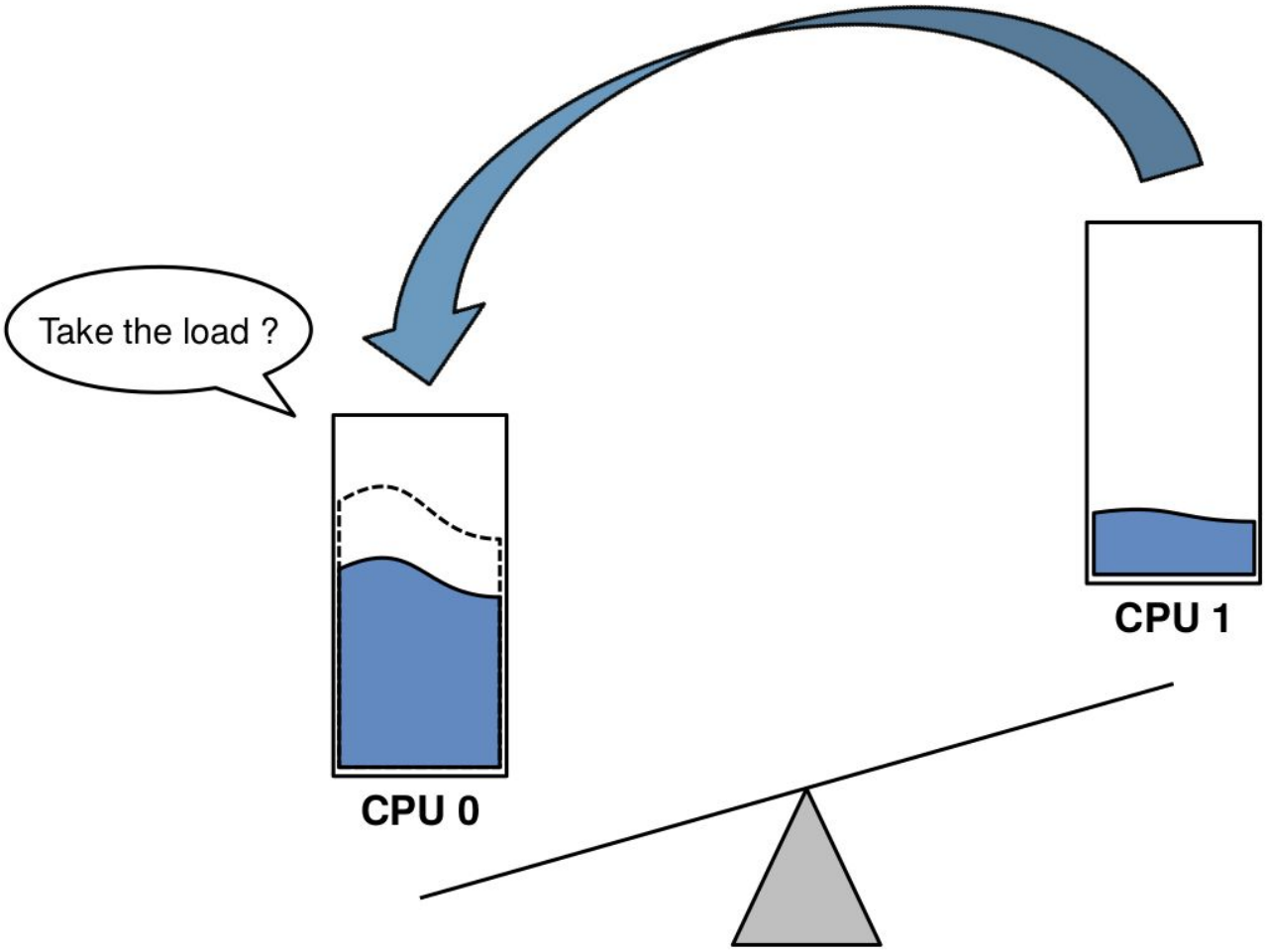


Problem

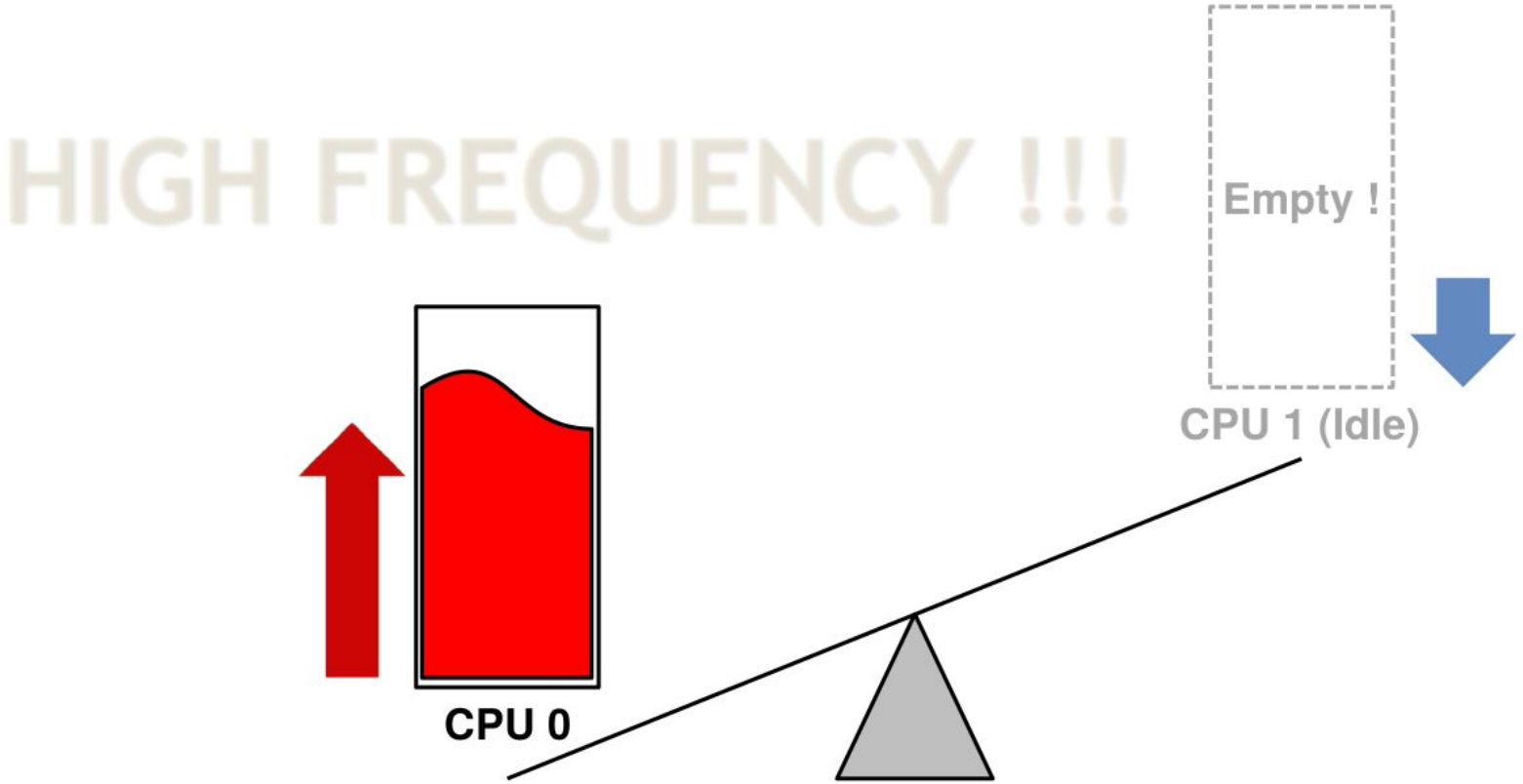
- **Frequency**



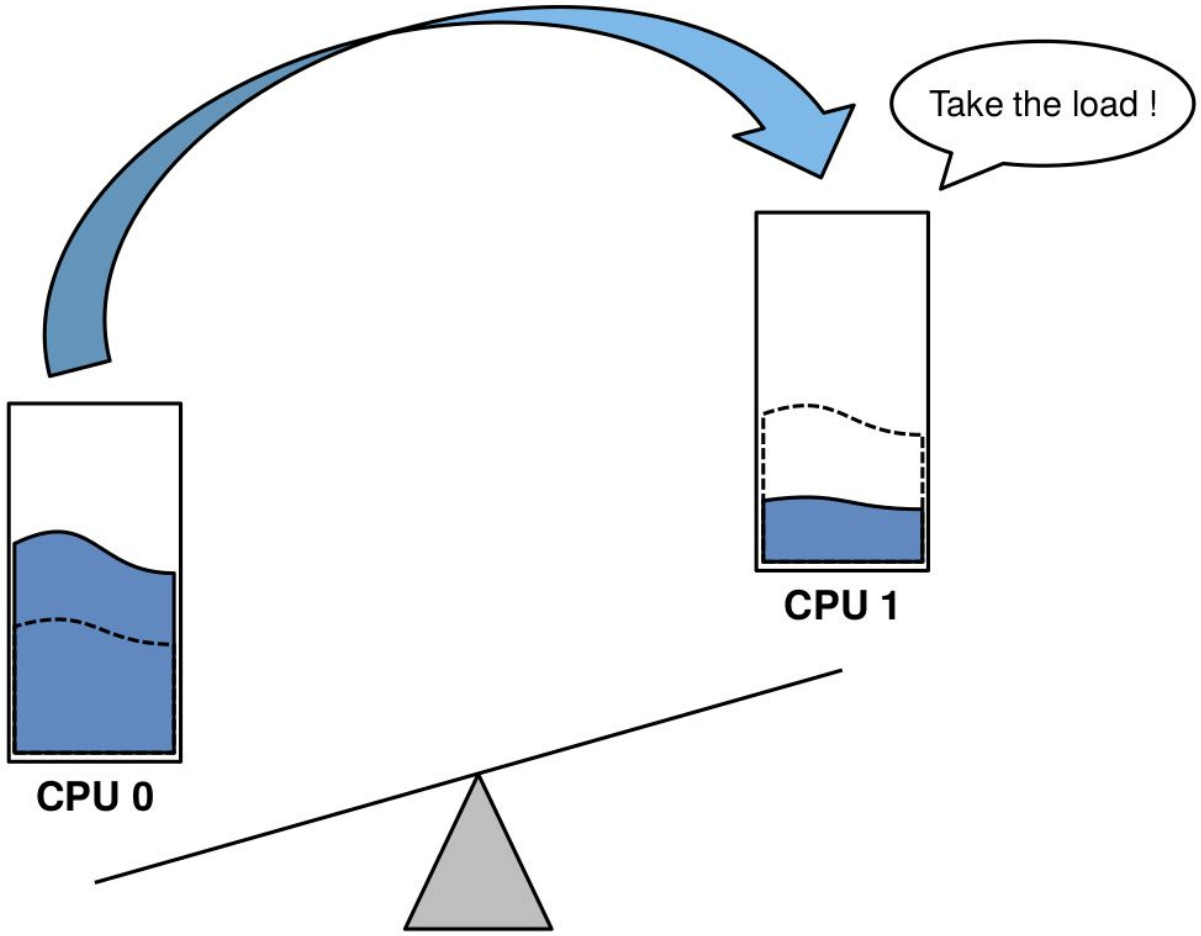
Problem - Frequency



Problem - Frequency

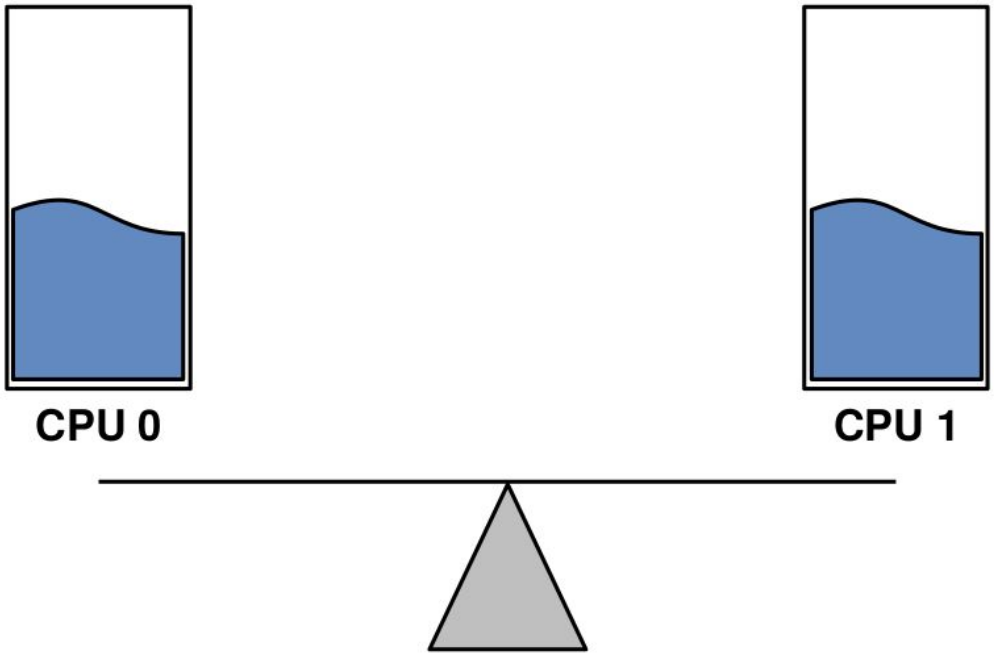


Solution



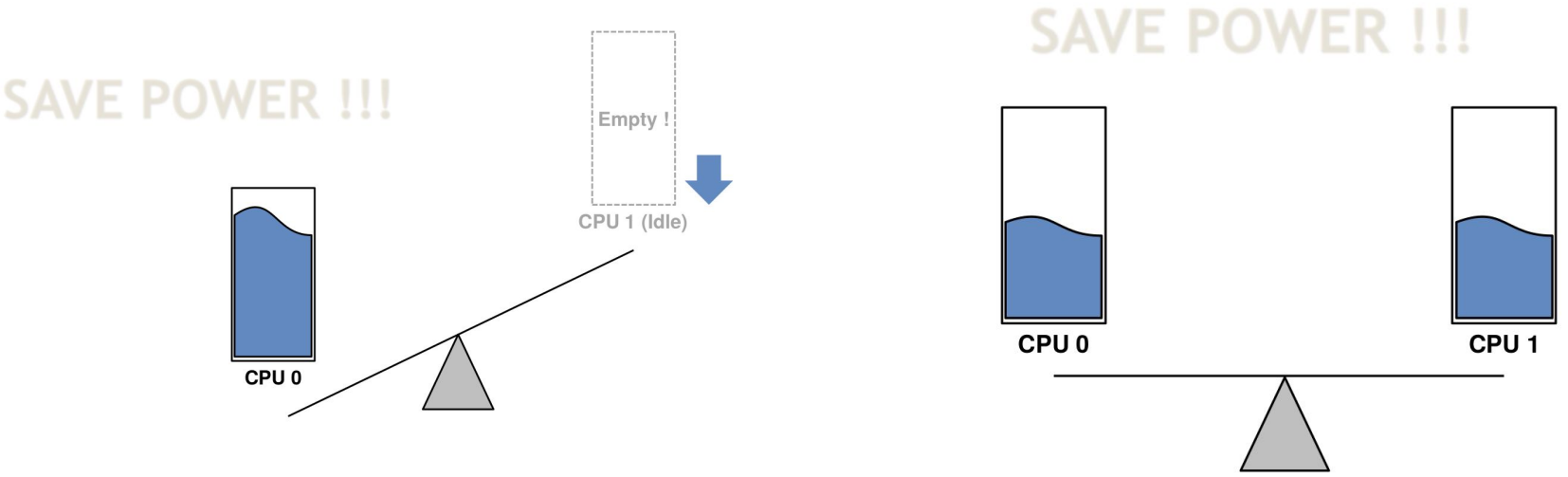
Solution

SAVE POWER !!!



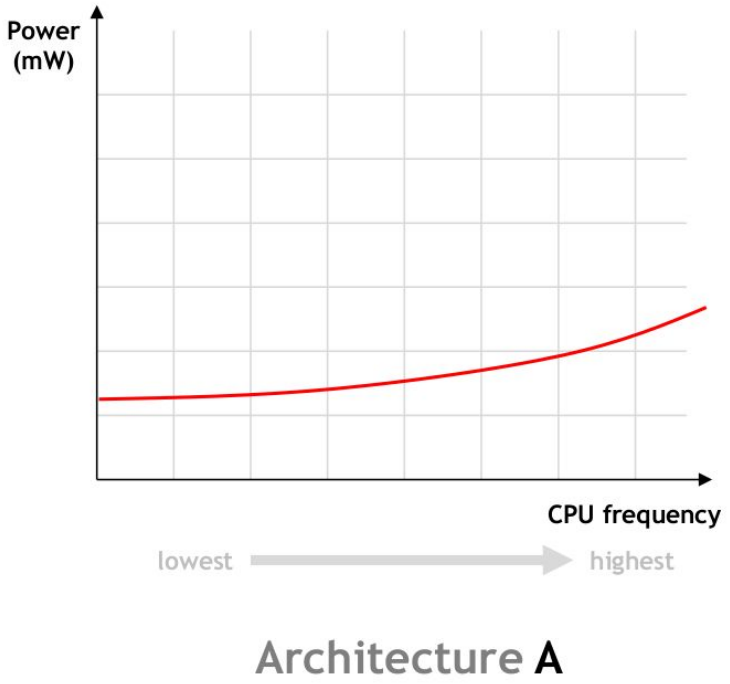
Problem 1

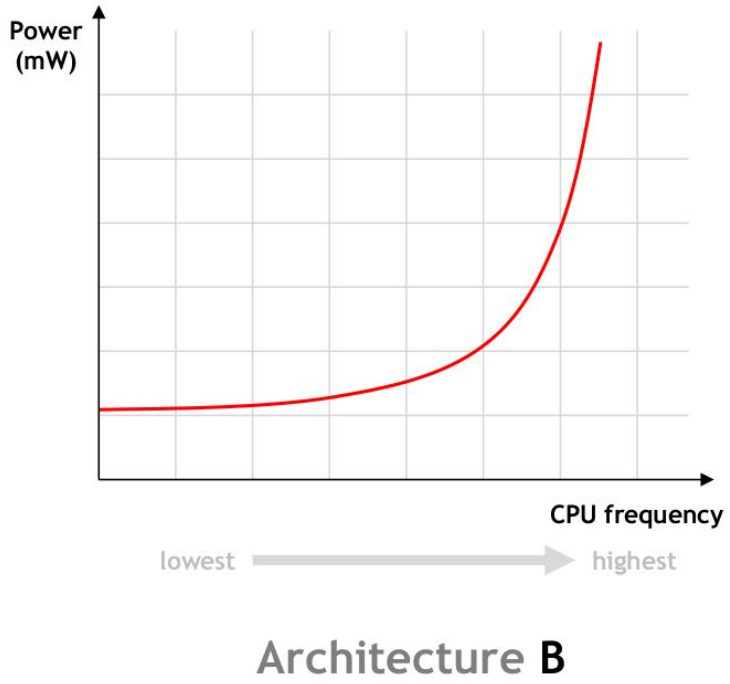
- **CPU idle? CPU frequency ?**



Problem 2

- **Architecture A? Architecture B?**



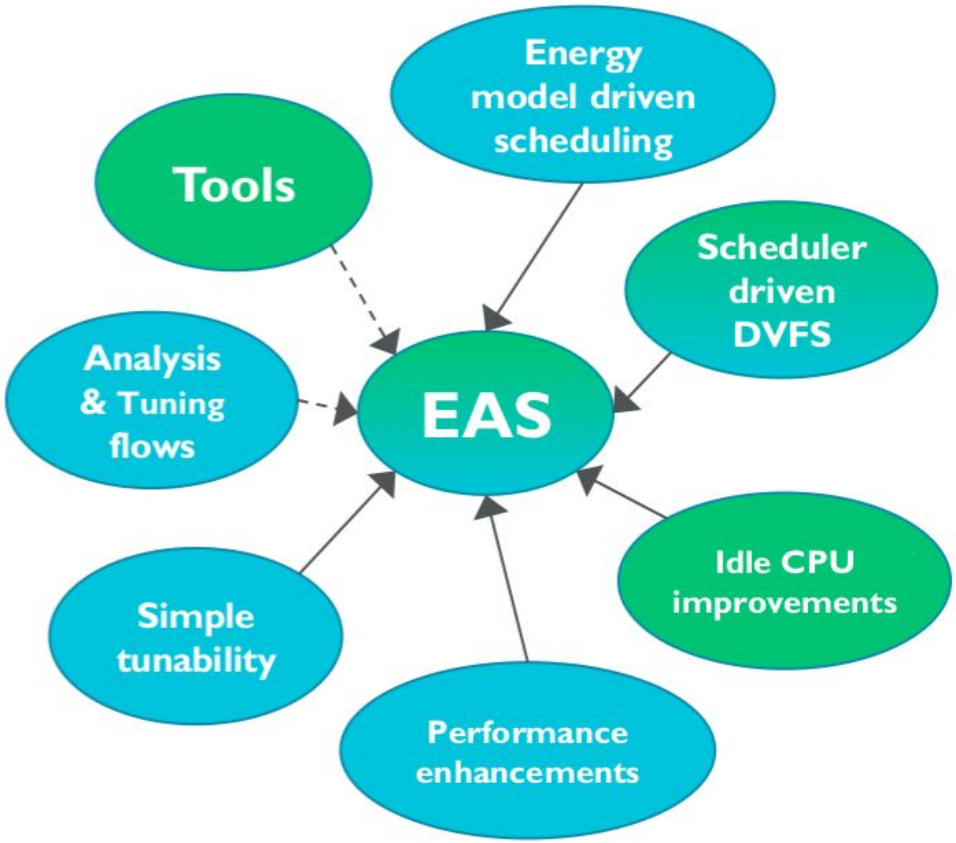


Solution

- **Use Energy Cost Model**

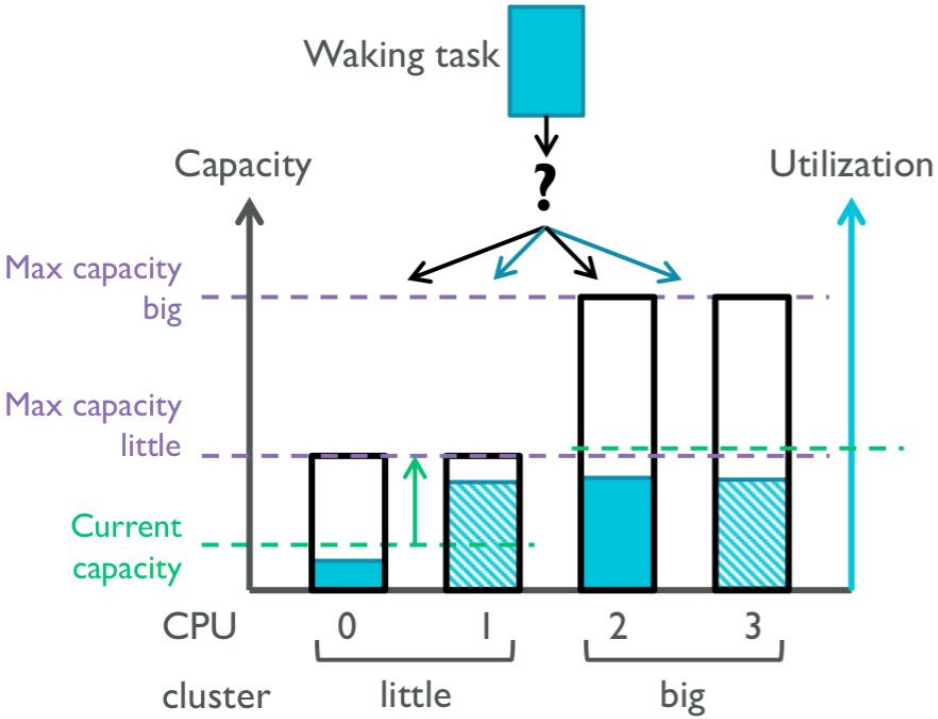
- **For saving power consumption,**
  - Is it better to pack tasks to fewer CPUs ?
  - Is it better to spread tasks to all CPUs ?

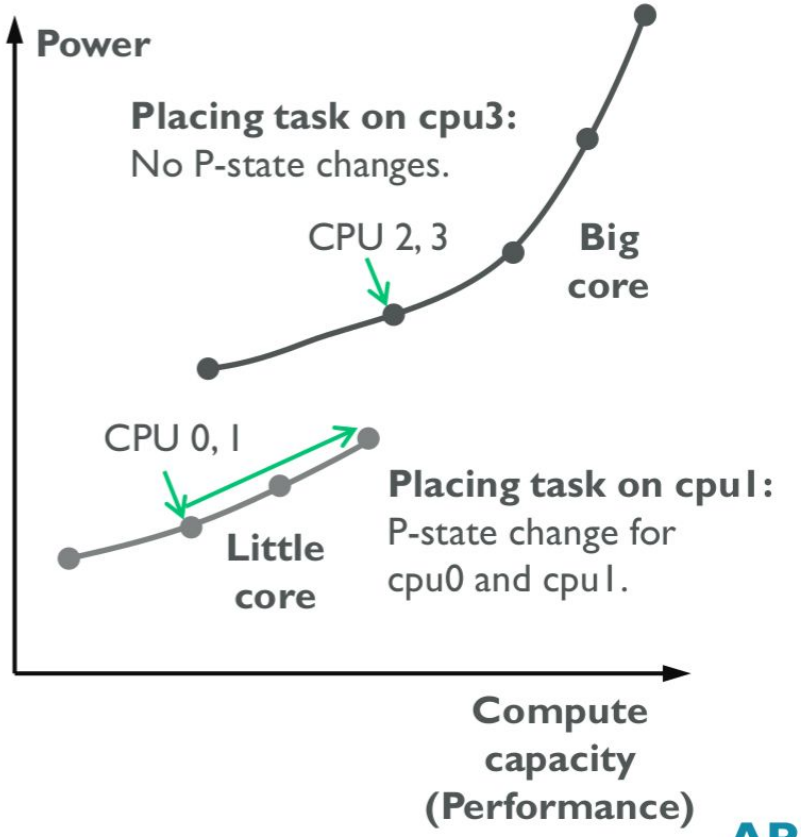
Solution


Energy model driven scheduling
Tools
Scheduler driven DVFS
Analysis & Tuning flows
EAS
Idle CPU improvements
Simple tunability
Performance enhancements
Linaro development
ARM development

"EAS Overview and Integration Guide”
ARM

What is EAS – the energy model





ARM

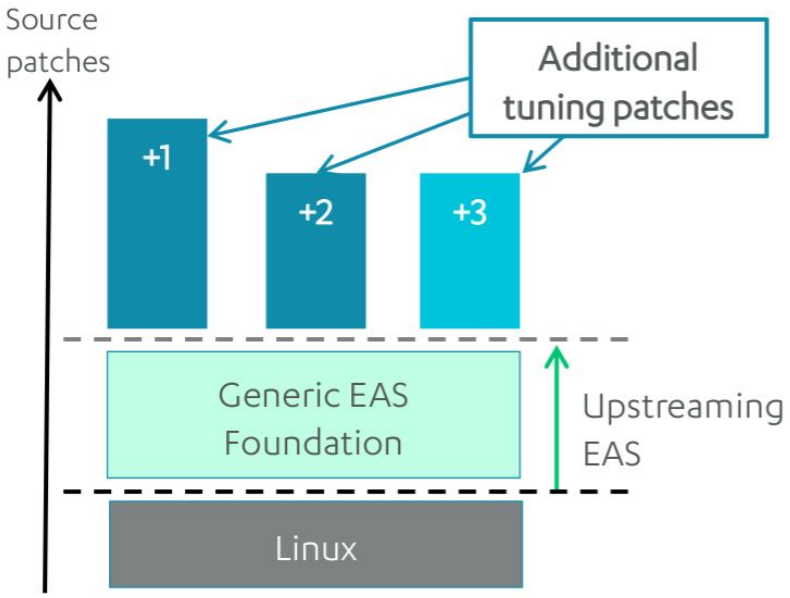
"EAS Update" 2015 september ARM

EAS vs big.LITTLE HMP(GTS)

EAS

New **Energy Aware Scheduling**

- Generic energy model based approach fits all platforms and topologies.
- Foundation for further enhancements.

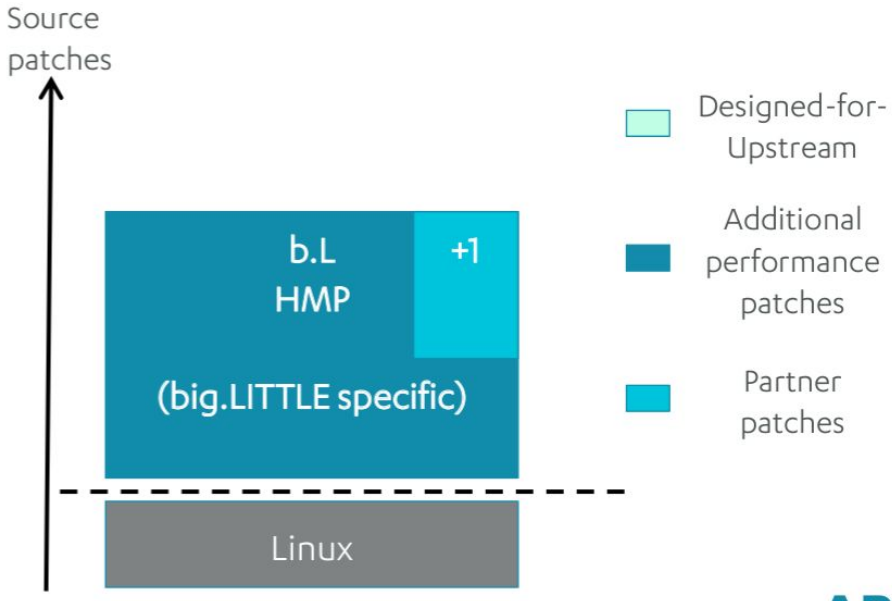


vs

big.LITTLE HMP

Existing **Heterogeneous MP** patchset

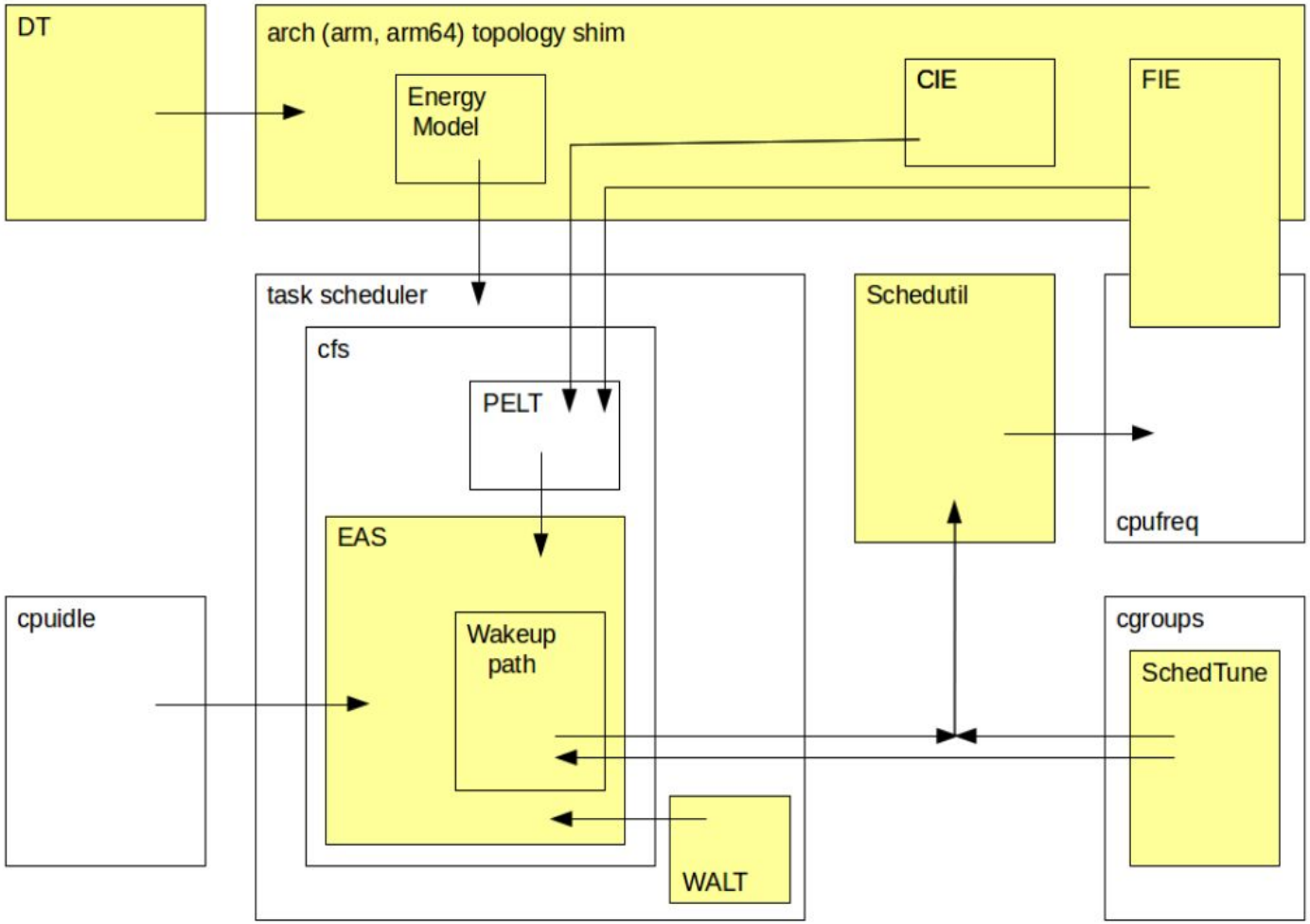
- big.LITTLE topology only.
- Hard coded behaviors.
- In Linaro LSK kernels (not mainline).



ARM

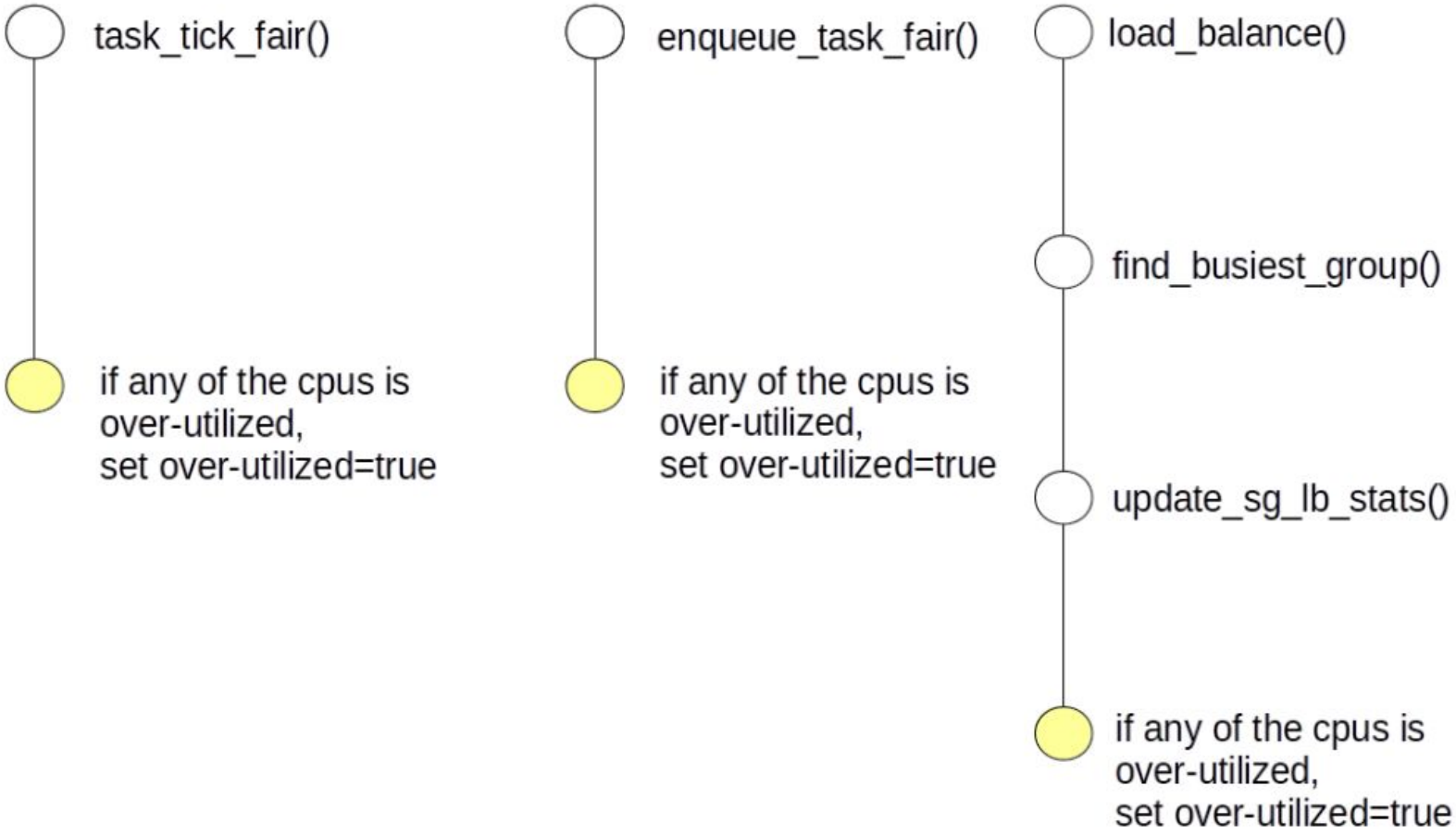
"EAS Update" 2015 september ARM

EAS building blocks



"EAS Overview and Integration Guide", ARM TR

Over-utilized



"EAS Overview and Integration Guide", ARM TR

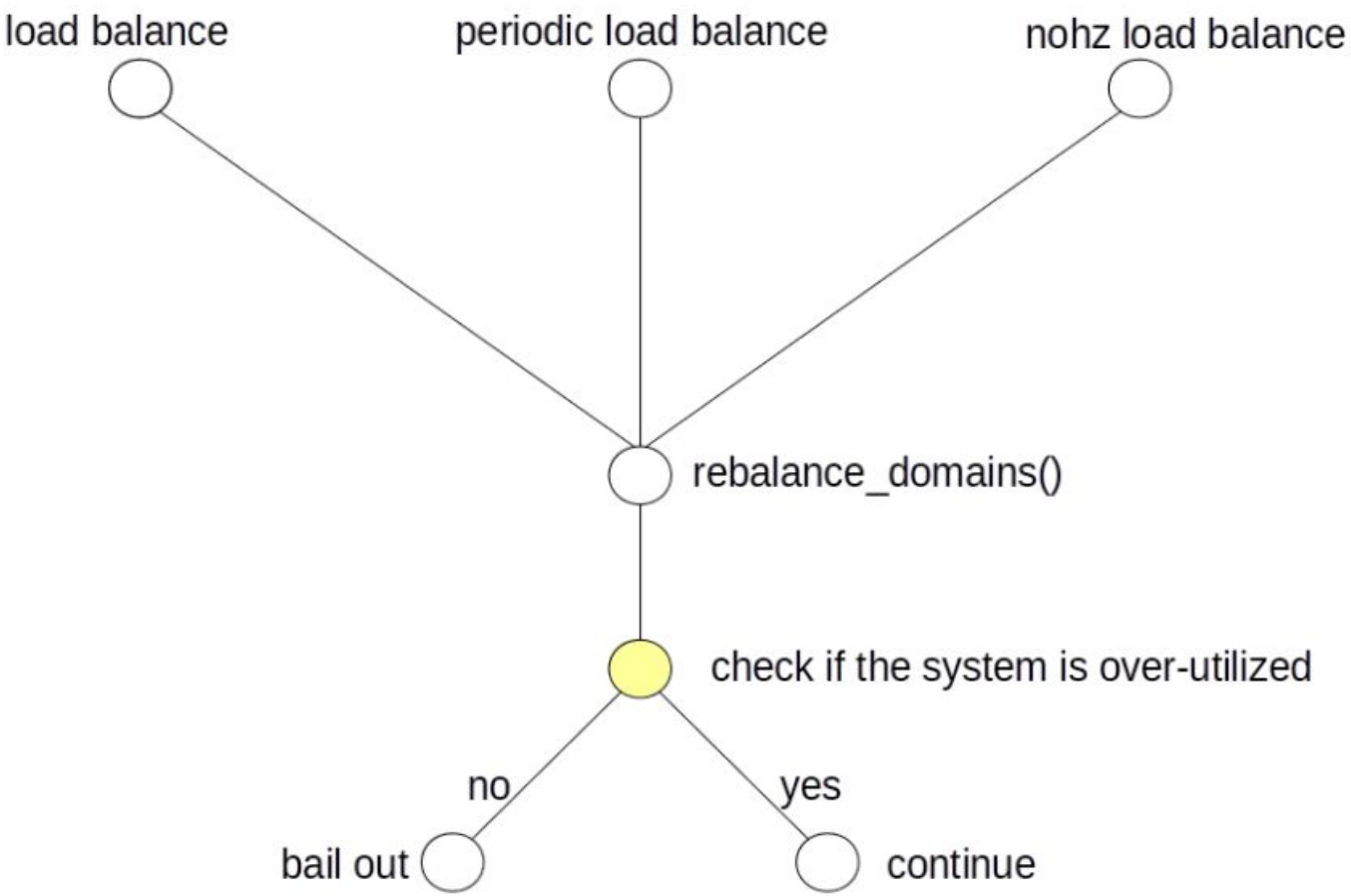
cpu_overutilized

```
static bool cpu_overutilized(int cpu)
{
    return (capacity_of(cpu) * 1024) < (cpu_util(cpu) * capacity_margin);
}
```

capacity_margin = capacity_margin = 1280; /* ~20% margin */

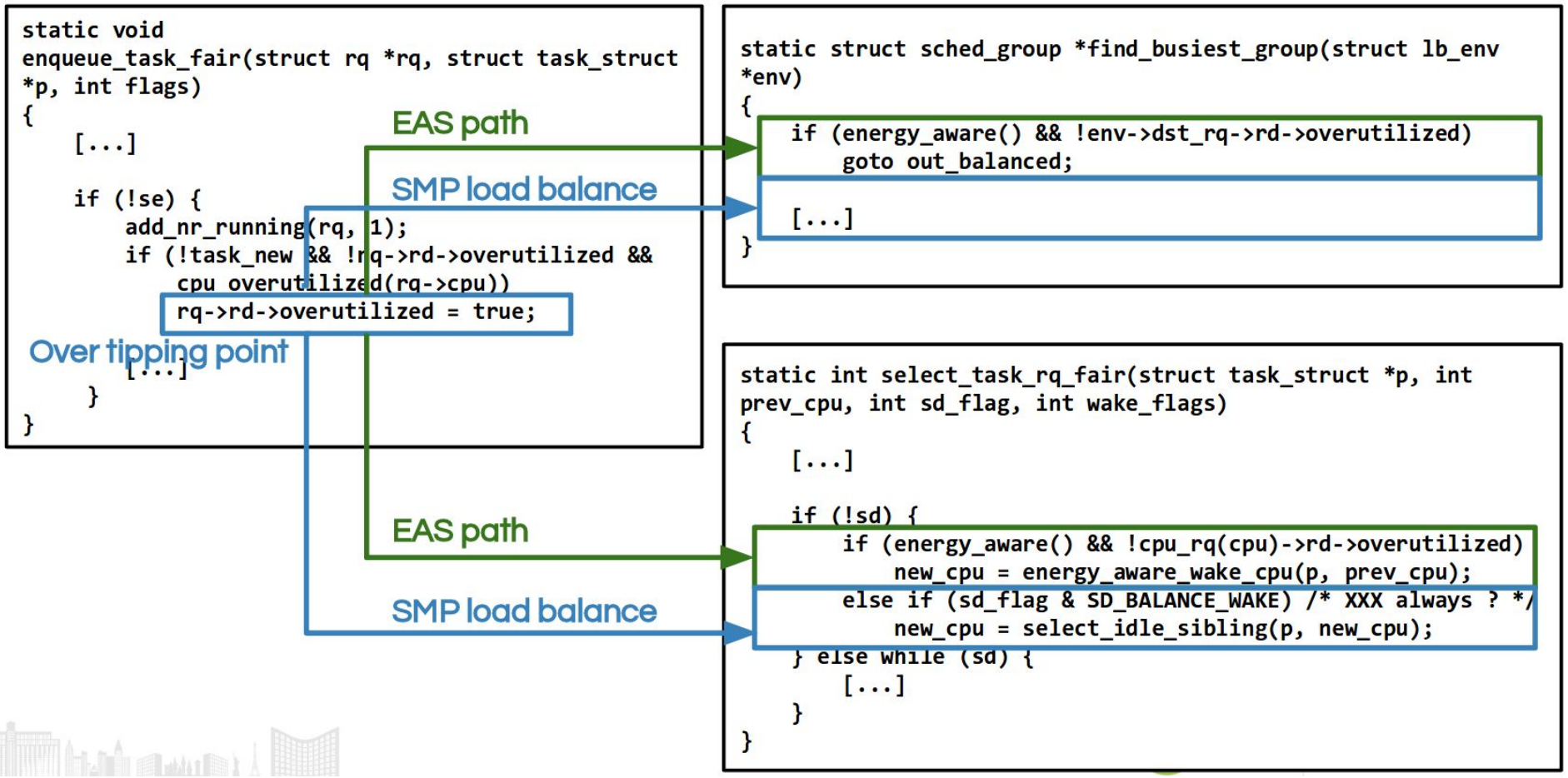
```
cpu_util = static inline unsigned long __cpu_util(int cpu, int delta)
{
    unsigned long util = cpu_rq(cpu)->cfs.avg.util_avg;
}
```

over-utilized

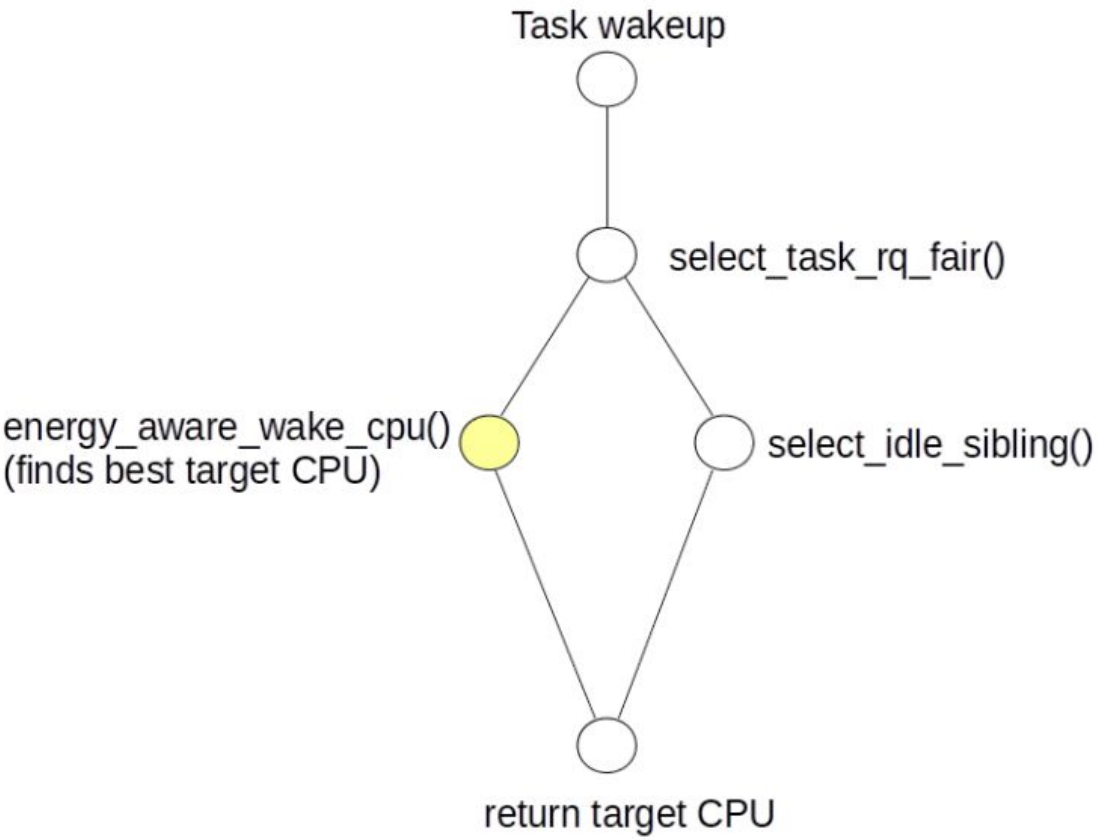


"EAS Overview and Integration Guide", ARM TR

over-utilized



energy_aware_wake_cpu



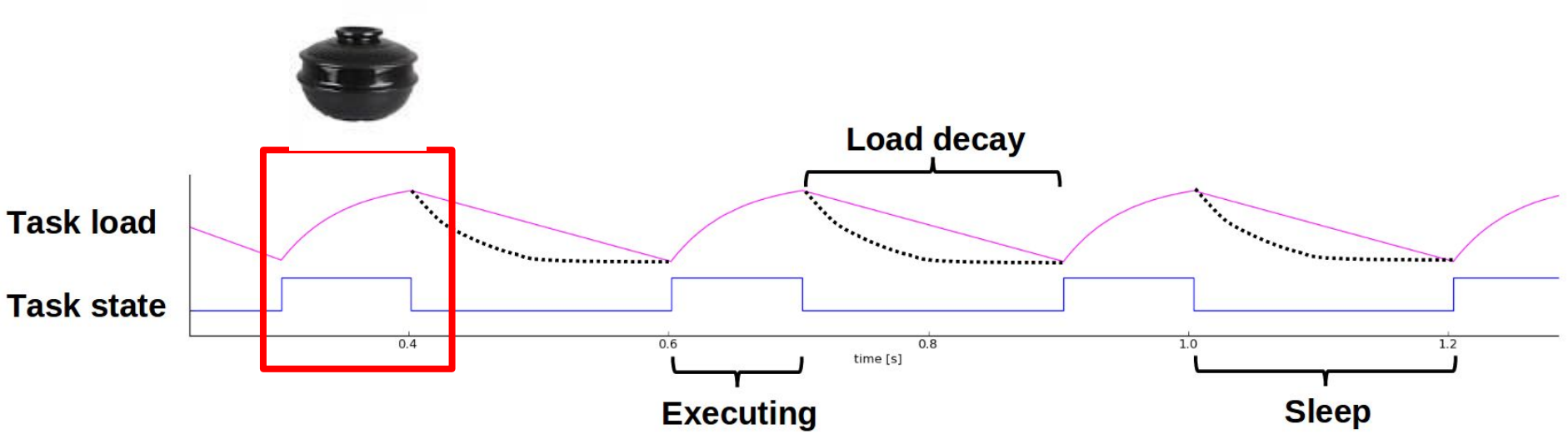
"EAS Overview and Integration Guide", ARM TR

select_task_rq_fair()

kernel/sched/fair.c

Problem - PELT+EAS

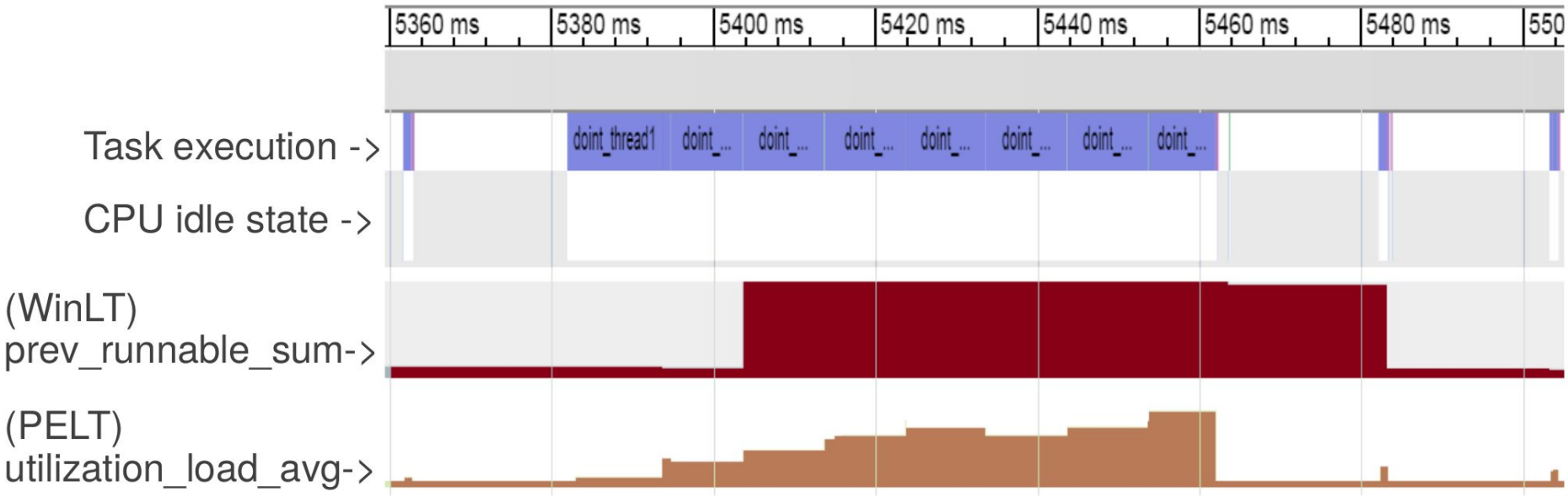
- Responsiveness problem.
- Consider web browsing in mobile world.
- The load does not rise quickly enough.



Solution

- **Use Window Aissisted Load Tracking scheme(WALT)**

PELT and WinLT



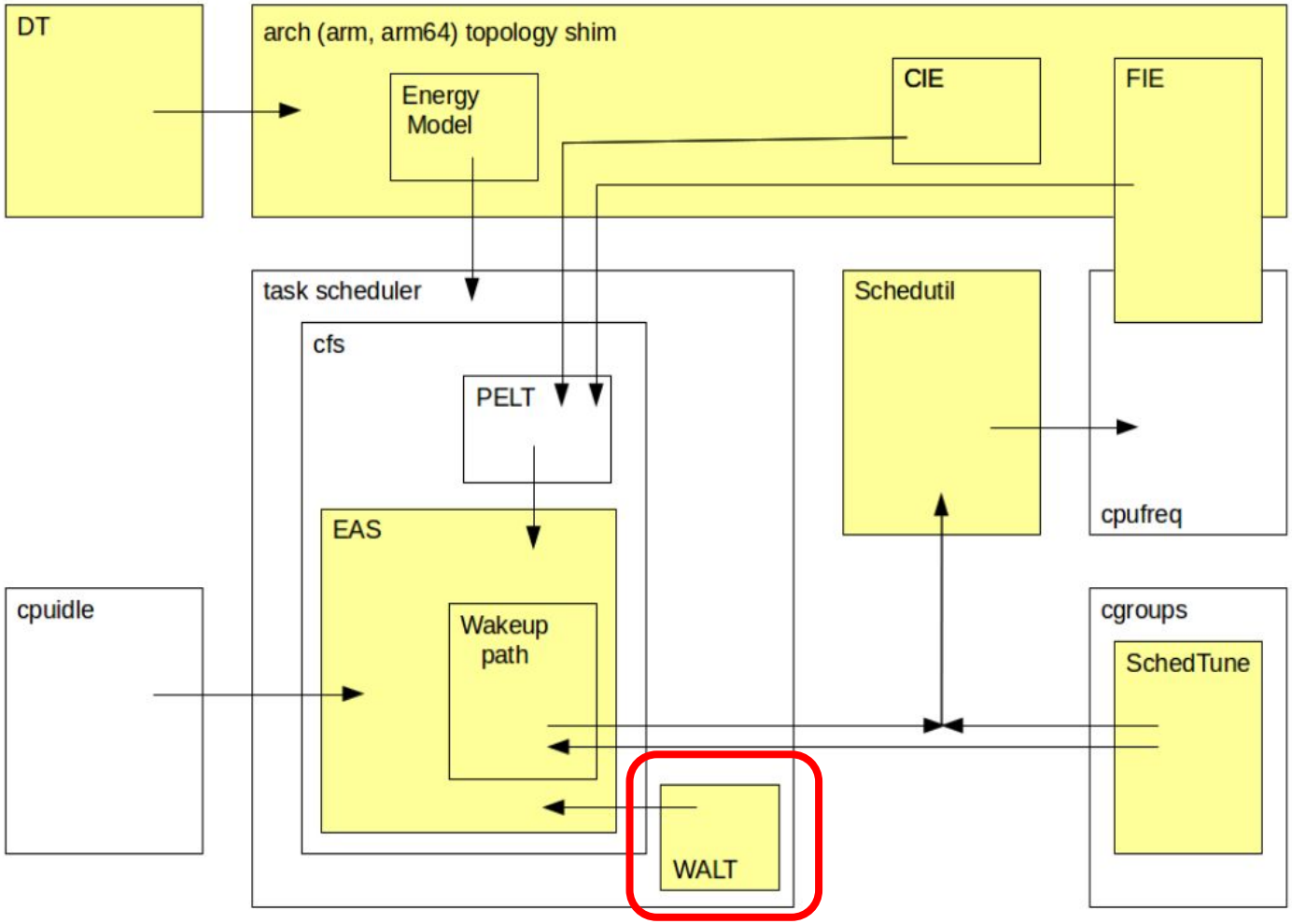
Solution

- **Use Window Aissisted Load Tracking scheme(WALT)**

- **WALT**
  - Problem : Lock + Cache coherency

- **Solution**
  - new PELT Load tracking algorithm

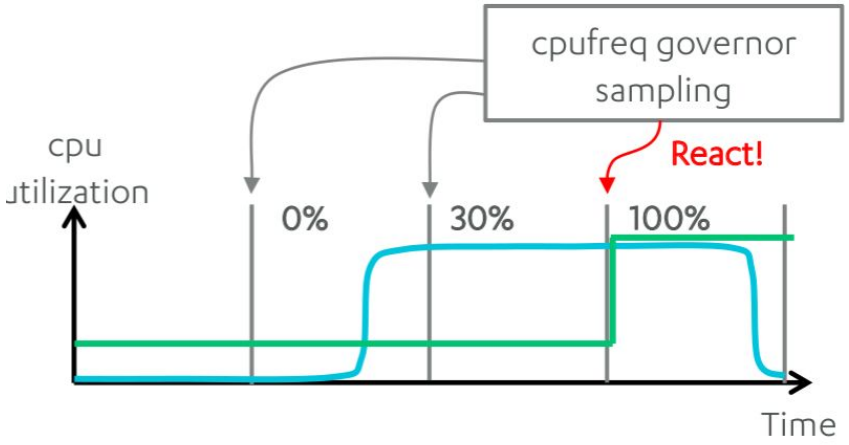
EAS building blocks

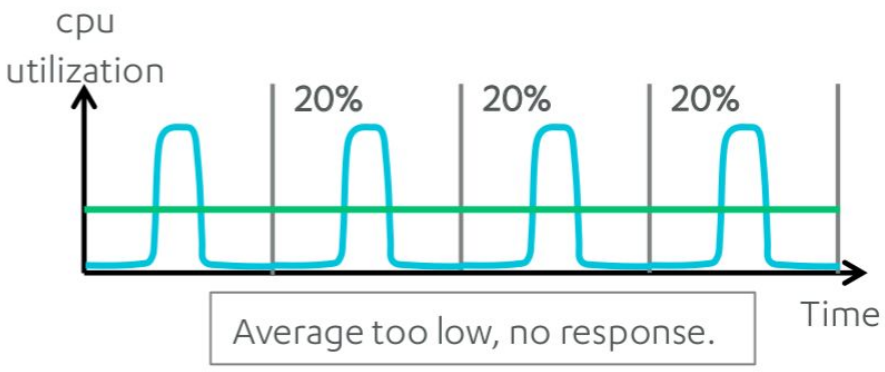


"EAS Overview and Integration Guide", ARM TR

Problem - Timer-based DVFS

- **Problem**
  - Sampling based governors are slow to respond and hard to tune.
- **Sampling too fast**
  - Freq changes for small utilization spikes.
- **Sampling too slow**
  - Average too low

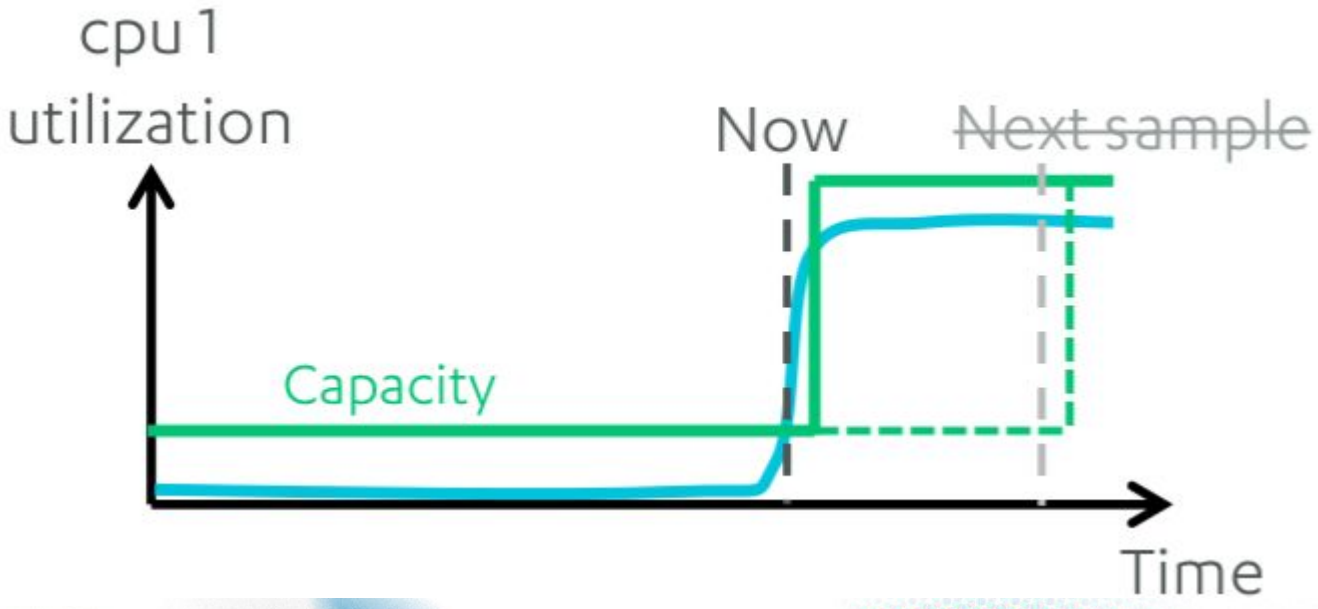




"EAS Update", Amit Kucheria et. ,al. ARM

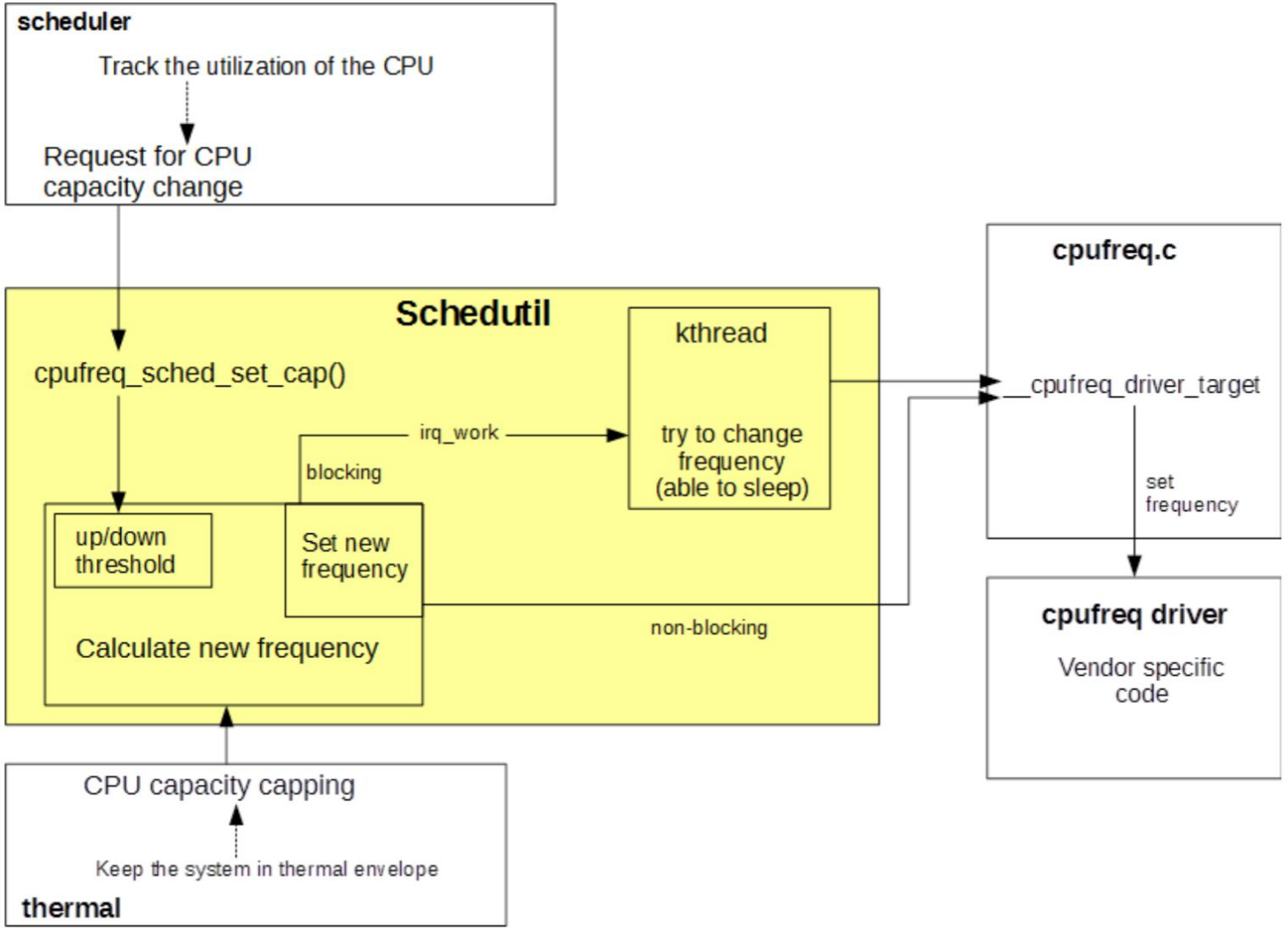
Solution

- **Use scheduler-based DVFS**
- **Scheduler task utilization tracking DVFS**
- **Immediately when CPU utilization changes**
  - improved responsiveness.



"EAS Update", Amit Kucheria et. ,al. ARM

Schedutil



"EAS Overview and Integration Guide", ARM TR

Problem - Android

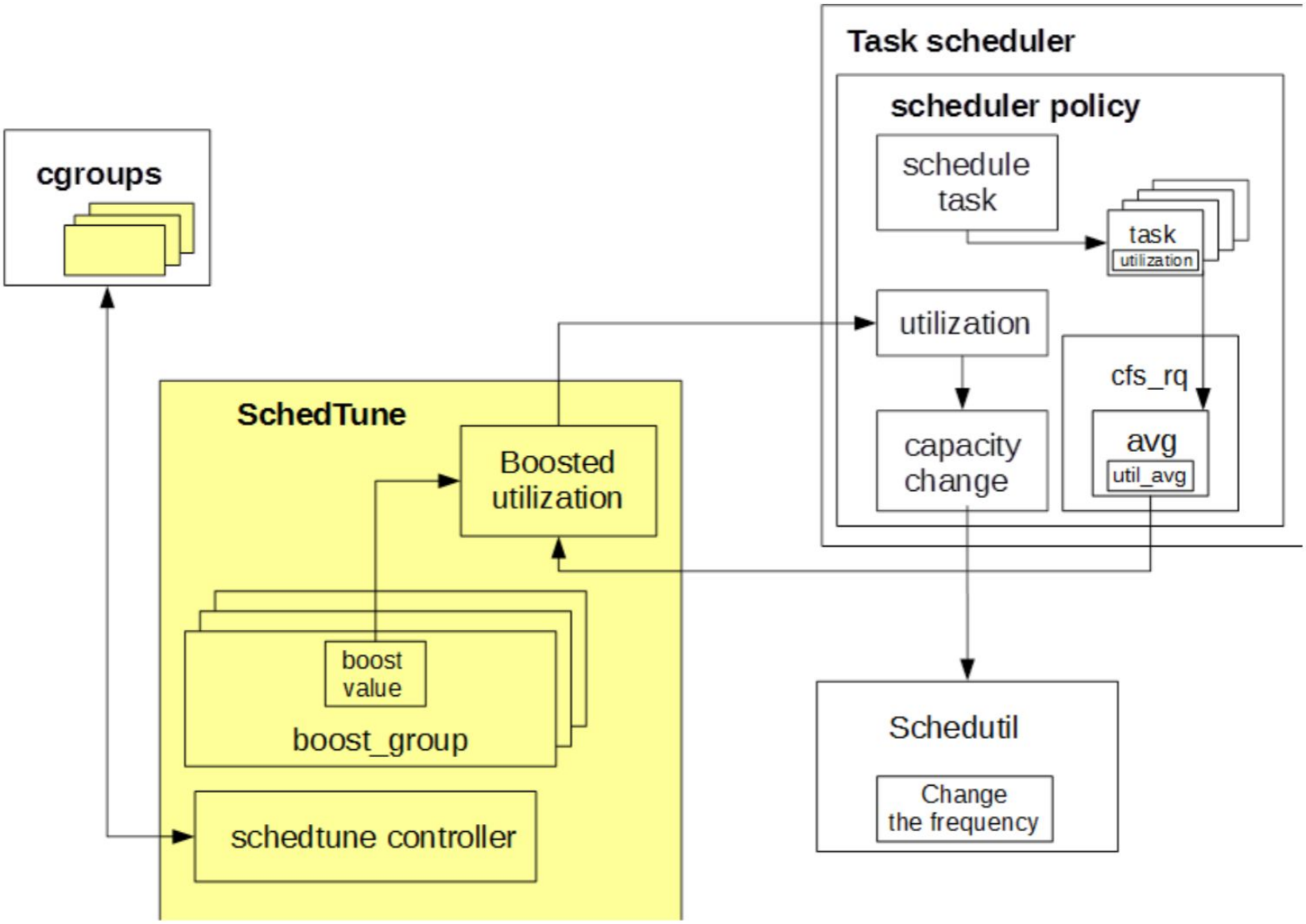
- **Sometimes response time is crucial to the users.**

- **Scheduler**
  - Cannot give user's experience information.

Solution

- **Use SchedTune**
- **A way of to provide user's experience information.**
- **Implemented as a control-group controller.**
- **Each control group has a single tunable knob.**
  - schedtune.boost

SchedTune


Task scheduler
scheduler policy
schedule task
task
utilization
utilization
cfs_rq
capacity change
avg
util_avg
cgroups
SchedTune
Boosted utilization
boost value
boost_group
schedtune controller
Schedutil
Change the frequency

Create energy-aware scheduler tuning nodes

```
mkdir /dev/stune
mount cgroup none /dev/stune schedtune
mkdir /dev/stune/foreground
mkdir /dev/stune/background
mkdir /dev/stune/top-app
chown system system /dev/stune
chown system system /dev/stune/foreground
chown system system /dev/stune/background
chown system system /dev/stune/top-app
chown system system /dev/stune/tasks
chown system system /dev/stune/foreground/tasks
chown system system /dev/stune/background/tasks
chown system system /dev/stune/top-app/tasks
chmod 0664 /dev/stune/tasks
chmod 0664 /dev/stune/foreground/tasks
chmod 0664 /dev/stune/background/tasks
chmod 0664 /dev/stune/top-app/tasks
```

Reference

- https://pdos.csail.mit.edu/6.828/2016/schedule.html
- http://web.mit.edu/6.033
- http://www.rdrop.com/~paulmck/
- "**Is Parallel Programming Hard, And If So, What Can You Do About It?"**
- Davidlohr Bueso. 2014. Scalability techniques for practical synchronization primitives. *Commun. ACM* 58

http://queue.acm.org/detail.cfm?id=2698990

- **"CPUFreq and The Scheduler Revolution in CPU Power Management", Rafael J. Wysocki**
- **https://sites.google.com/site/embedwiki/oses/linux/pm/pm-qos**
- **https://intl.aliyun.com/forum/read-916**
- **User-level threads : co-routines**

  http://www.gamedevforever.com/291

  https://www.youtube.com/watch?v=YYtzQ355_Co

- **Scheduler Activations**
  - https://cgi.cse.unsw.edu.au/~cs3231/12s1/lectures/SchedulerActivations.pdf
- **https://en.wikipedia.org/wiki/FIFO_(computing_and_electronics)**
- **http://jake.dothome.co.kr/**
- **http://www.linuxjournal.com/magazine/completely-fair-scheduler?page=0,0**
- **https://www2.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/6_CPU_Scheduling.html**
- **"Energy Aware Scheduling", Byungchul Park, LG Electronic**
- **"Update on big.LITTLE scheduling experiments", ARM**
- **"EAS Update" 2015 september ARM**
- **"EAS Overview and Integration Guide", ARM TR**
- **"Drowsy Power Management", Matthew Lentz, SOSP 2015**
- **https://www.slideshare.net/nanik/learning-aosp-android-hardware-abstraction-layer-hal**
- https://www.youtube.com/watch?v=oTGQXqD3CNI
- https://www.youtube.com/watch?v=P80NcKUKpuo
- https://lwn.net/Articles/398470/
- "SCHED_DEADLINE: It's Alive!", ARM, 2017