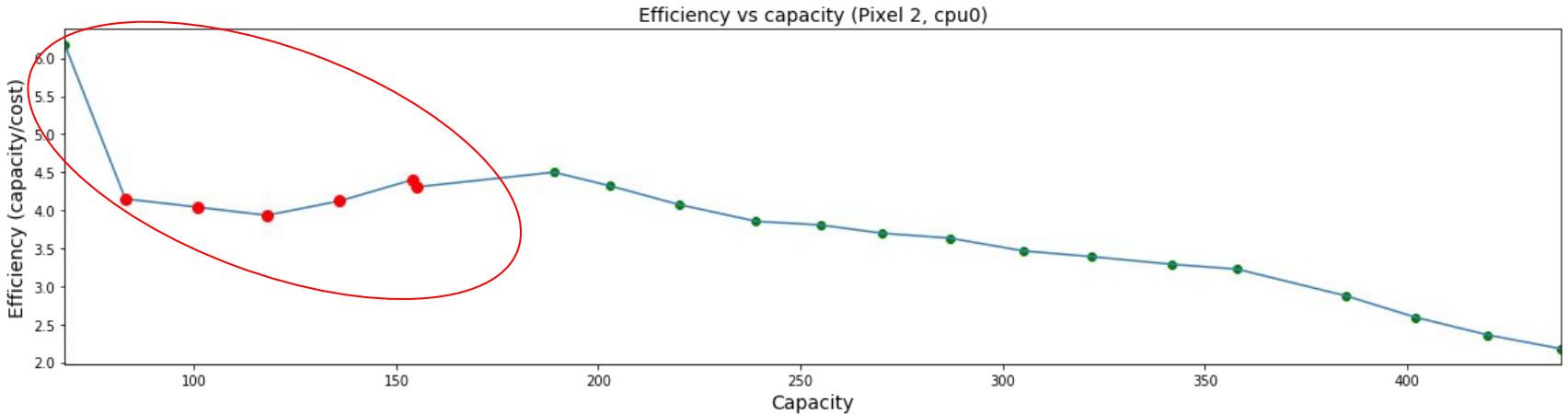
Let's make SchedUtil Energy Aware

OSPM 2019

Douglas Raillard

What's the problem on hand ?

- Higher frequencies (boosting) at a controlled power cost
  - When it’s cheap
  - When it’s useful
- Skip inefficient frequencies on some platforms



https://android.googlesource.com/kernel/msm/+/refs/heads/android-msm-wahoo-4.4-oreo-dr1/arch/arm/boot/dts/qcom/msm8998.dtsi#282

arm

What can we use to improve frequency selection ?

- Shiny new Energy Model framework to get frequencies' cost
  - Allows skipping inefficient OPPs (Operating Power Points)

- sugov_cpu_is_busy() heuristic to boost frequency when it could be useful
  - Updated to work for shared policies

[RFC PATCH 0/7] sched/cpufreq: Make schedutil energy aware
LKML: https://lkml.org/lkml/2019/5/8/805
Branch: eas/next/integration-20190503
Git repo: git://linux-arm.org/linux-power.git

When and how much can we boost?

The heuristic deciding of the boosting needs to satisfy some criteria

- Not boosting all the time
  - Only when ramping up in frequency
  - Other useful scenarios ?

- Energy-aware: bounded power cost
  - Whole performance domain/cpufreq policy affected !
  - Heuristic may trigger spuriously

sugov_cpu_is_busy(): How does it fare ?

Busy ~= no idle time since last frequency increase

Pros

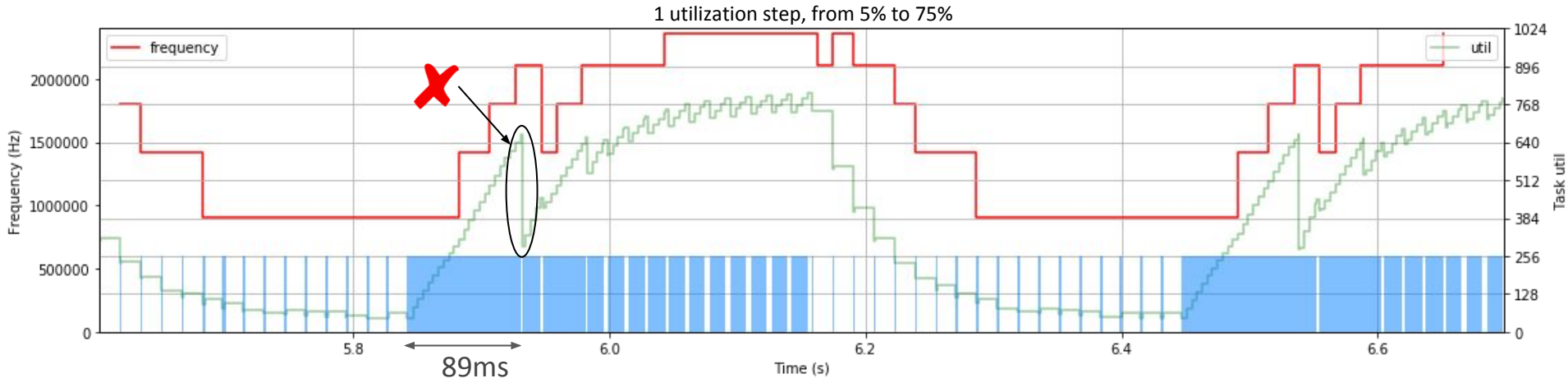
- Boosting improves utilization settle time & time to reach needed frequency

Cons

- May trigger too often
- Events ordering (freq & utilization changes) is important, which may be brittle

Frequency selection: without EM-Boost (mainline)

- Frequency
- Task utilization (se->util_avg)
- Rtapp activations



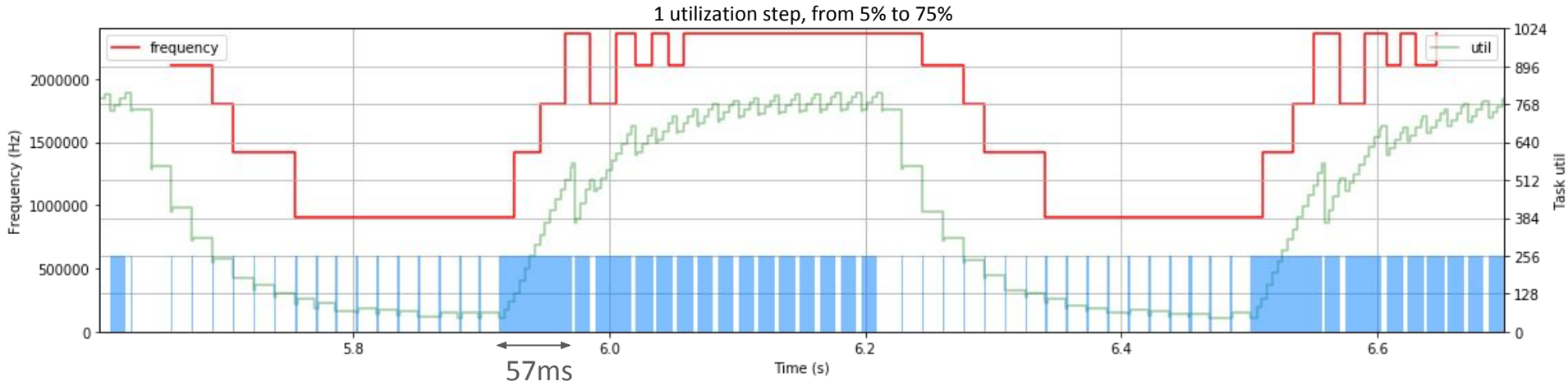
Kernel based on tip/sched/core (as of 2019/04/26)
with additional patches needed for hikey960, full tree available here:
git://linux-arm.org/linux-power.git d3855f96e038249416e12aab854397fdad29d85e

✗ Task util drop after 1st idle

arm

Frequency selection: with EM-Boost

- Frequency
- Task utilization (se->util_avg)
- Rtapp activations



tip sched/core (2019/04/26) + EAS integration series

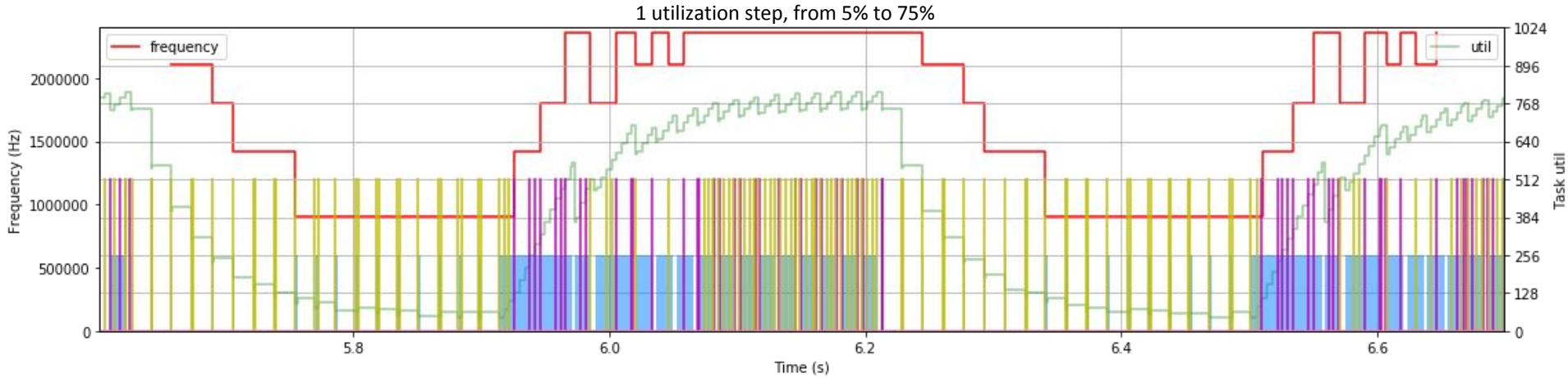
Branch: eas/next/integration-20190503
Git repo: http://linux-arm.org/git?p=linux-power.git;a=summary

- Freq curve still above util one
- Reduced drop at first idle

arm

Frequency selection: with EM-Boost

- Frequency
- Task utilization (se->util_avg)
- No boost
- Boost
- Rtapp activations



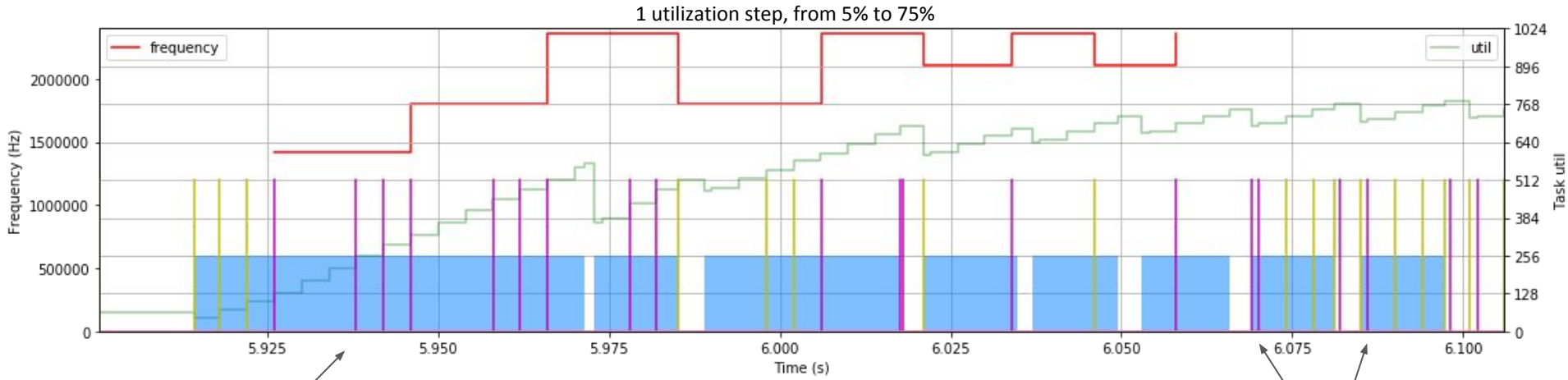
- No boosting when idle
- But boosting after ramp up

Boost events

arm

Frequency selection: EM-Boost limitations

- Task utilization
- Frequency
- No boost
- Boost
- Rtapp activations



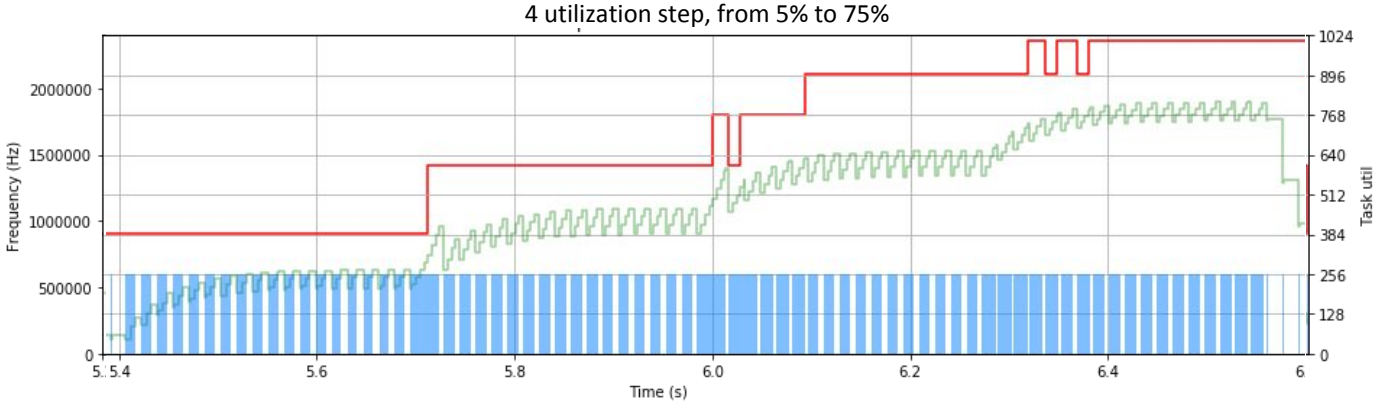
Zoom-in ++

boost when missing activations

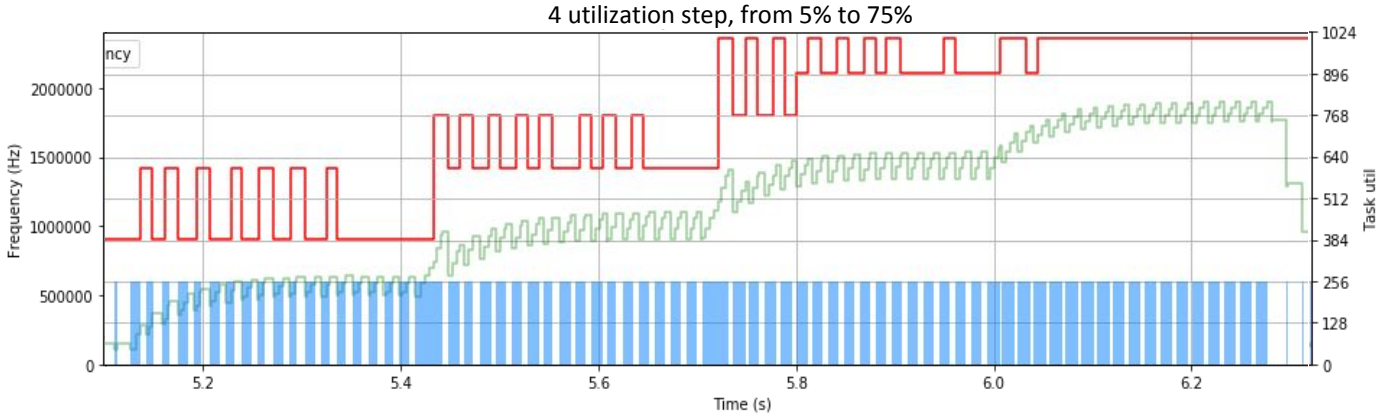
Spurious unwanted boost

arm

MultiStep utilization increase



Mainline



EM-Boost

arm

Frequency Boosting

Better heuristics than sugov_cpu_is_busy() ?

What next ?

- Testing with range of workloads
- (real) energy measurements

- Experiment with other boosting heuristics
- Userspace knob to adjust power-frequency tradeoff ?
- ...

Thanks !

MultiStep utilization increase

More steps

arm