Linux Plumbers Conference 2022

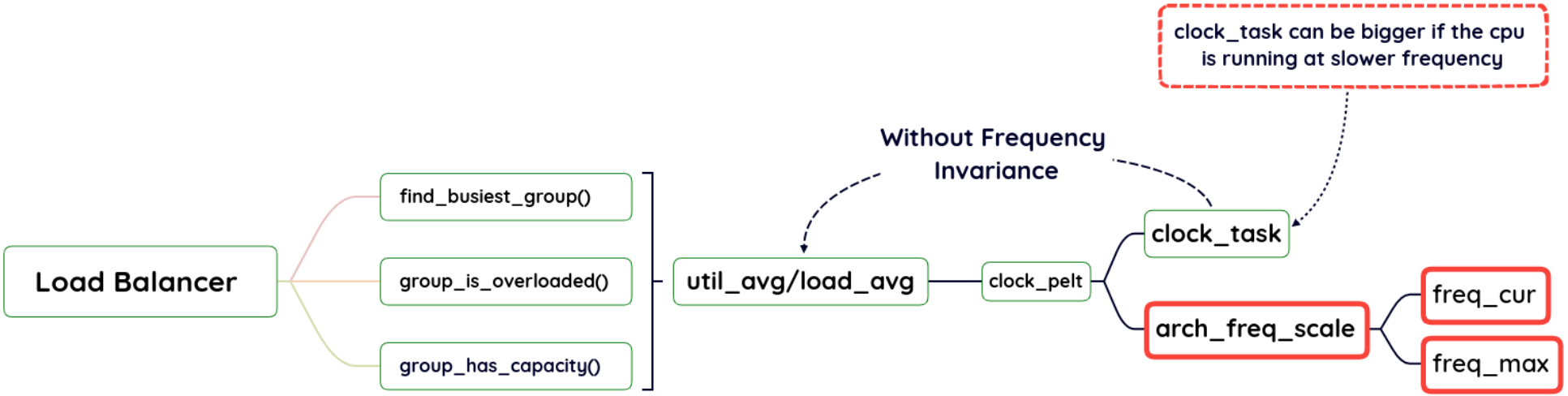
>> Dublin, Ireland / September 12-14, 2022

Frequency-invariance gaps in current kernel

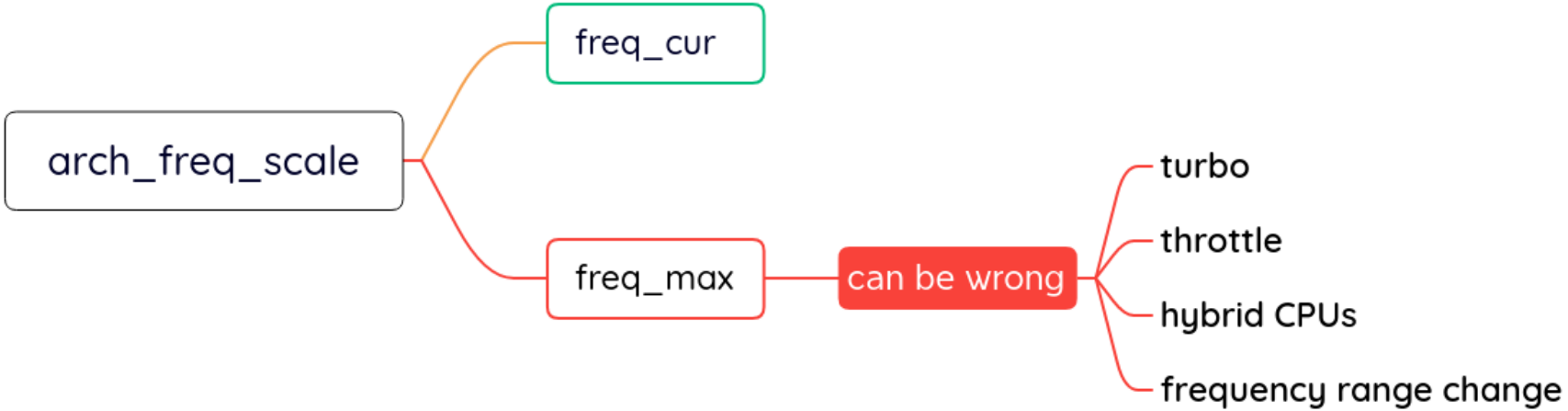
Zhang Rui <rui.zhang@intel.com>

Chen Yu <yu.c.chen@intel.com>

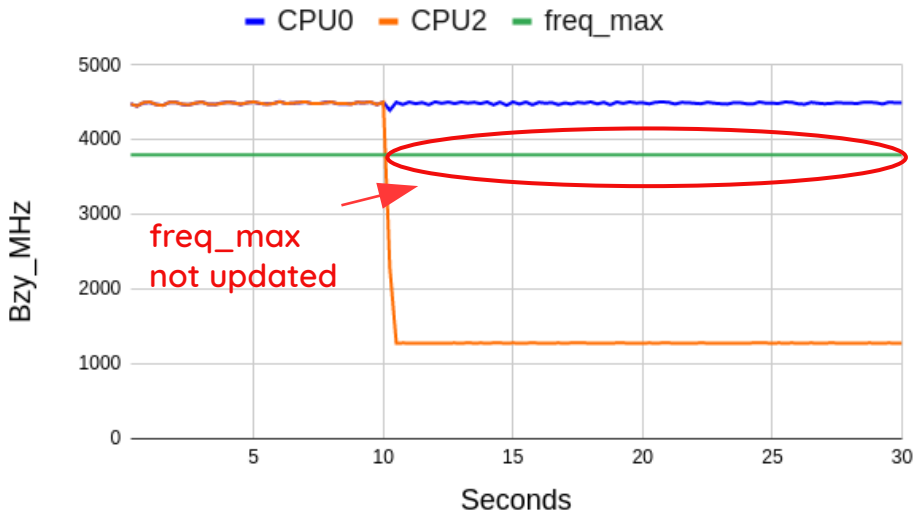
Frequency-Invariance background

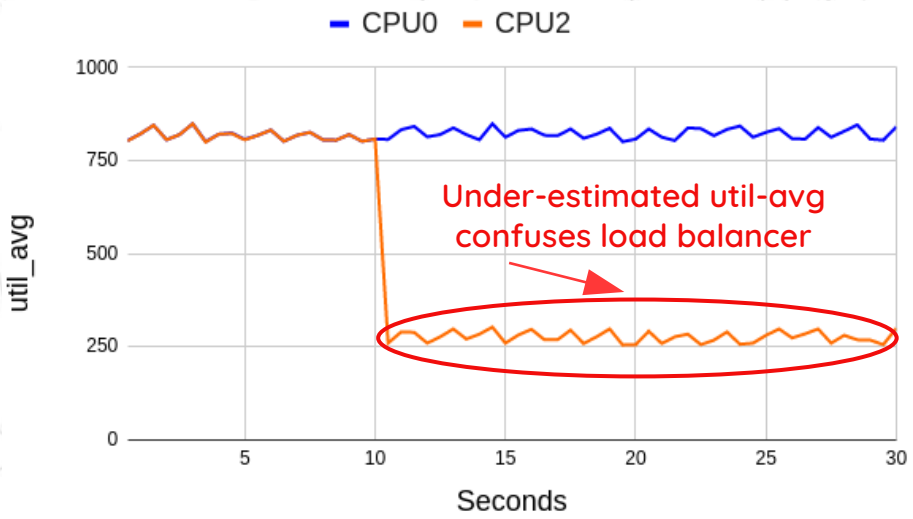


freq_max can be wrong



Bogus freq_max when frequency range changed



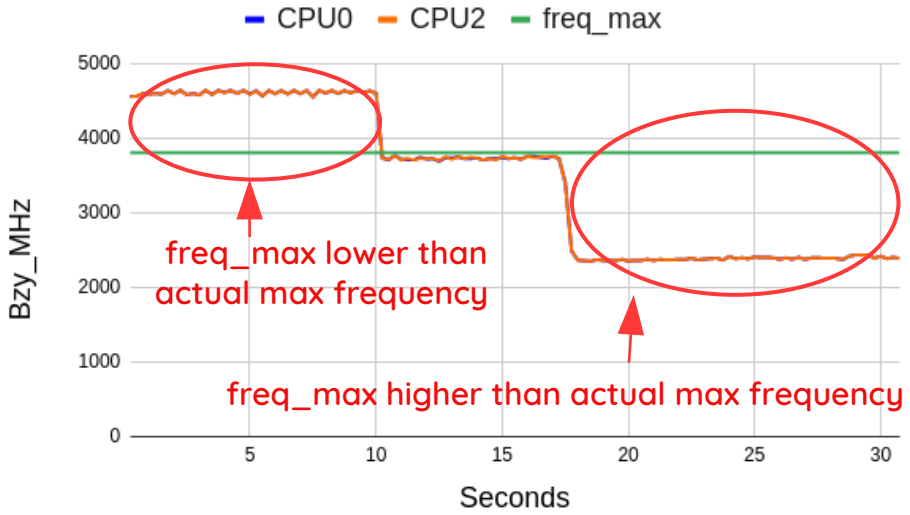


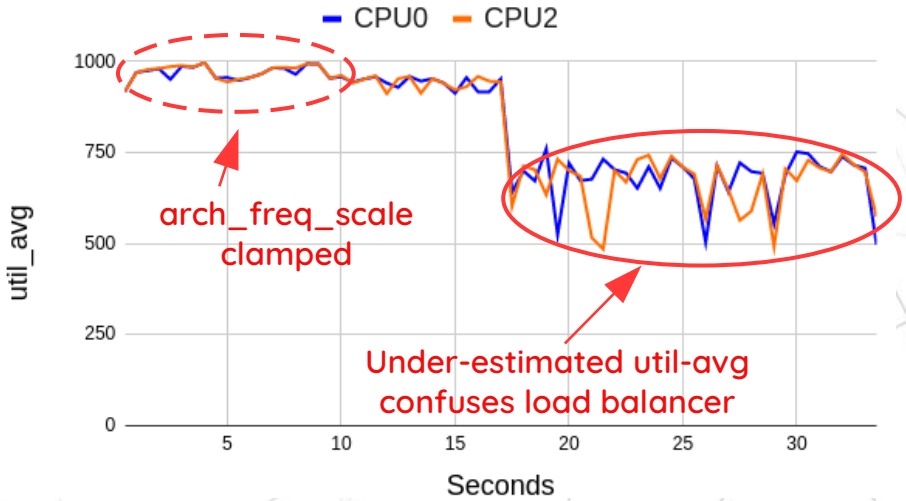
Workload:

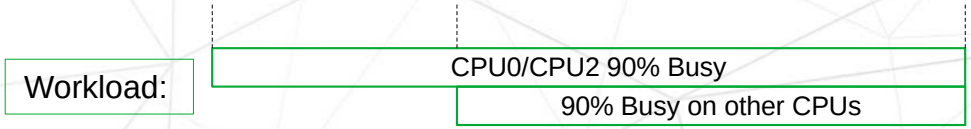
80% Busy on CPU0/CPU2

Limit CPU2 frequency via sysfs

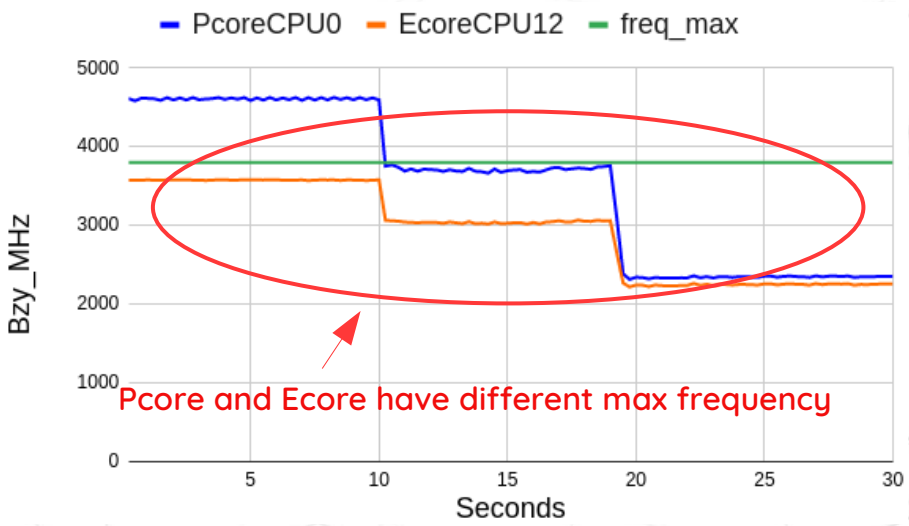
Bogus freq_max when turbo/throttled

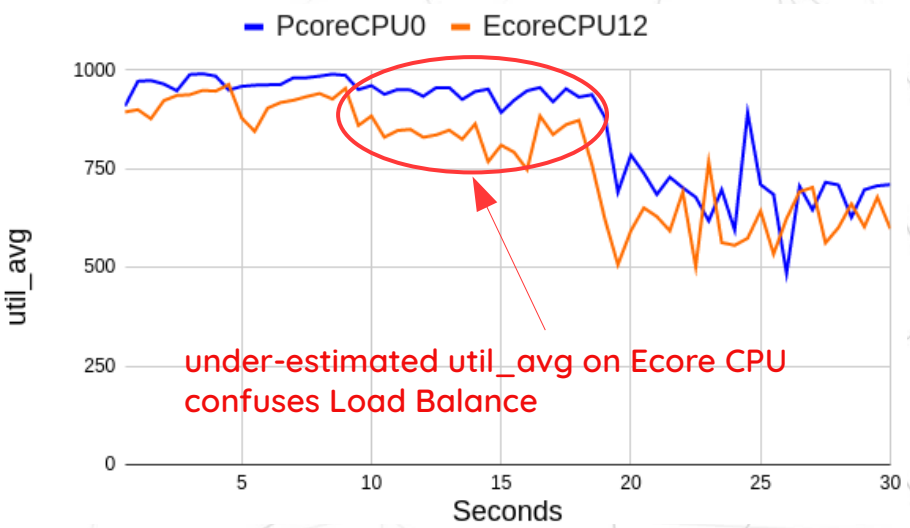


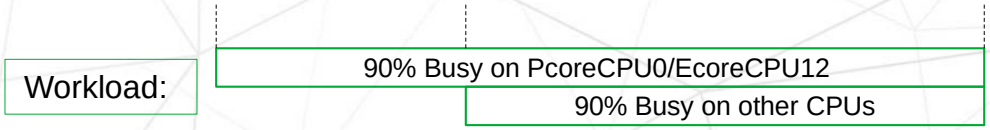




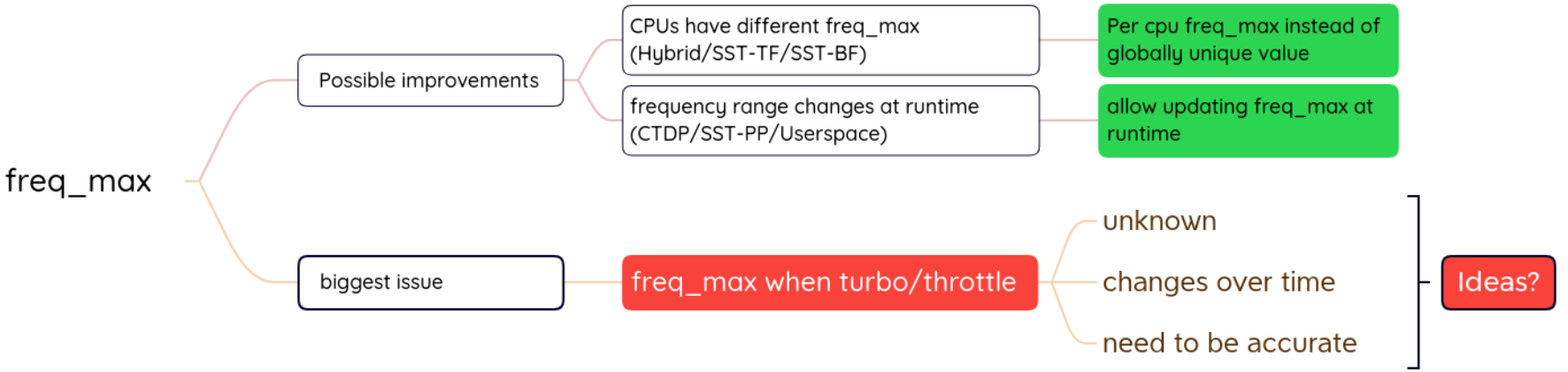
Bogus freq_max on hybrid CPUs



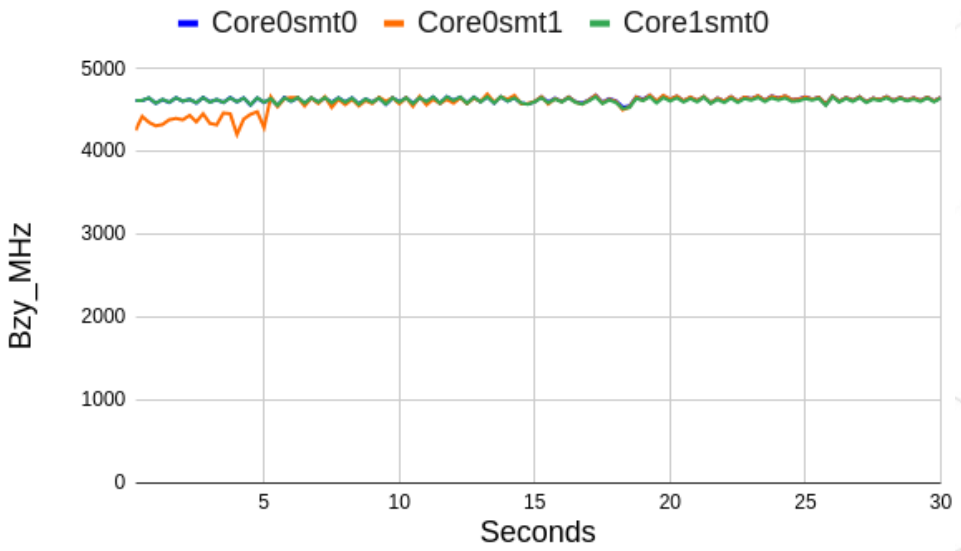


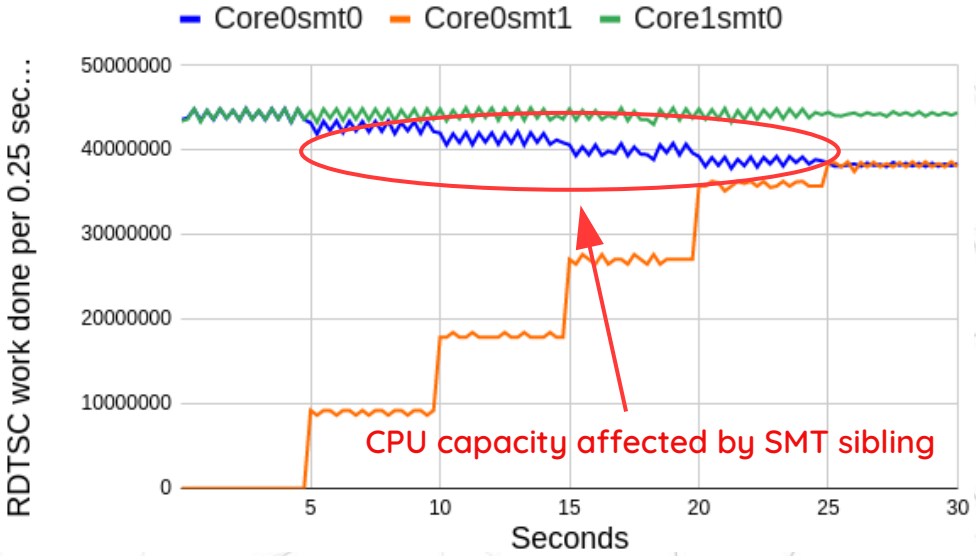


Next steps ?


freq_max
Possible improvements
CPUs have different freq_max (Hybrid/SST-TF/SST-BF)
Per cpu freq_max instead of globally unique value
frequency range changes at runtime (CTDP/SST-PP/Userspace)
allow updating freq_max at runtime
biggest issue
freq_max when turbo/throttle
unknown
changes over time
need to be accurate
Ideas?

Backup1
CPU capacity affected by SMT sibling



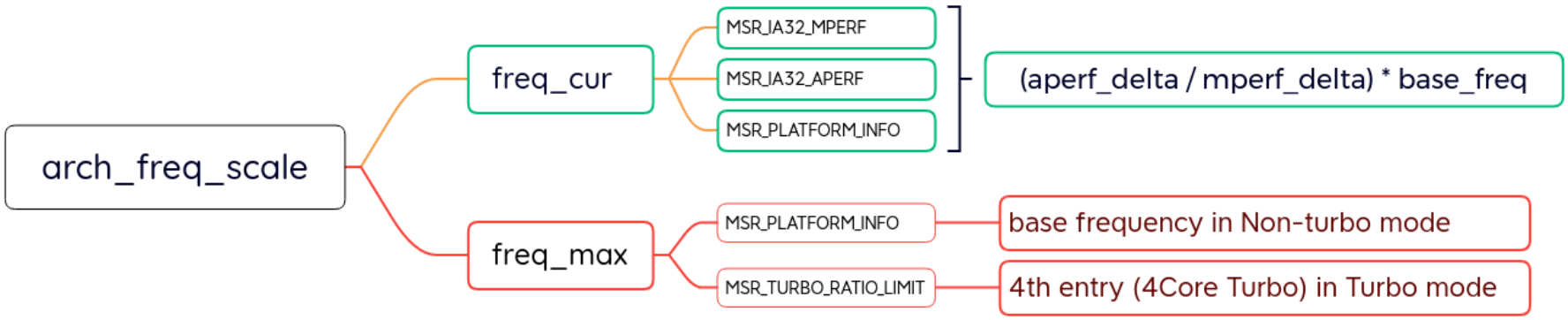


100% Busy on Core0smt0/Core1smt0

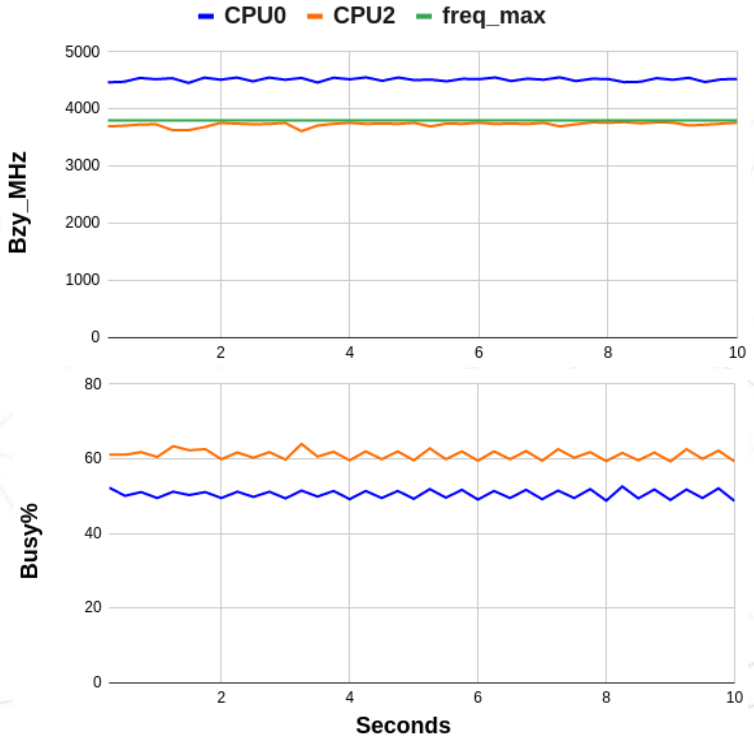
Increasing workload on Core0smt1

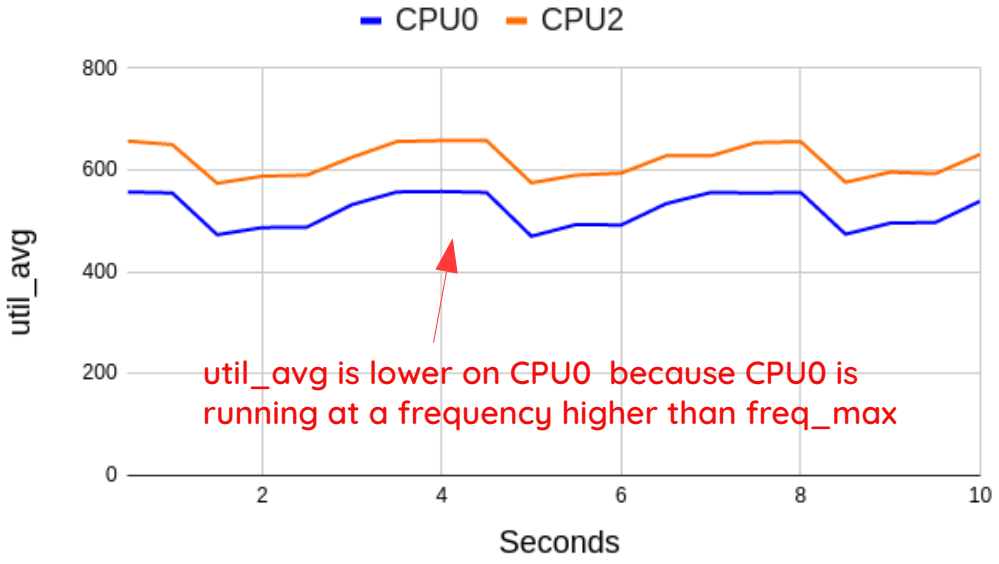
cannot be measured by frequency scaling. Ideas?

Backup2
frequency invariance on Intel



Backup 3
Lose accuracy when freq_max is low

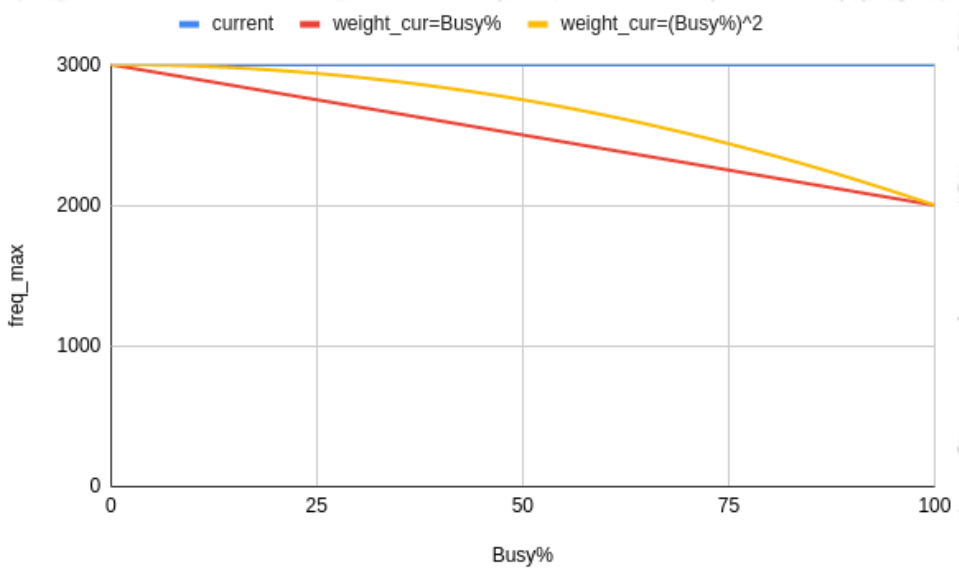




Workload:	Fixed task size on CPU0/CPU2 (yogini rate50)
	CPU0: performance. CPU2: schedutil

Backup 4
utilization based freq_max estimation

- Assumption
  - either firmware or software is targeting for higher frequency when CPU is busier
  - Under-estimated util_avg does not impact much on CPUs with high Idle residency
- Solution
  - Weight current frequent in freq_max calculation
  - Weight is a variant based on Busy% (CPU utilization)
  - Busy% = mperf_delta / tsc_delta



Estimated freq_max value when CPU is throttled from 3G to 2G