KBA-170921005535

CPUFreq mitigation with LMH DCVSh.

Applicable platform:

MSM8998, SDM845, SDM660, SDM670……

Issue/problem description:

In MSM8998/SDM660 chipset, you would find there is no CPUfreq update of scaling_cur_freq or scaling_max_freq when CPU frequency throttling kicks in. below explains the reason and shows alternative debug method.

Issue Analysis:

KTM(msm_thermal) decides how the cpufreq mitigation request should be propagated.

- **Without LMH DCVSh (legacy)**
  - Vote scaling max and min frequency to cpufreq.
- **With LMH DCVSh**
  - Scaling max frequency is voted via LMH DCVSh h/w override vote.
  - Scaling min frequency is voted via both LMh DCVSh h/w and cpufreq.
    - No register in OSM to aggregate the min request. (Will be featurized in SDM845)
    - Once featurized in h/w (in SDM845), the min frequency will be voted to LMH DCVSh h/w only.

In MSM8998/SDM660, KTM will use LMH(**80-NM328-108**) DCVSh to vote for frequency, that is why mitigation is unknown to cpufreq driver. So no update of either scaling_cur_freq or scaling_max_freq in tsens log when CPU frequency throttling kicks in.

msm_thermal.c API: **update_cpu_freq**

```
/*
 * If LMH DCVS is available, we update the hardware directly
 * for faster response. However, the LMH DCVS does not aggregate
 * min freq correctly - cpufreq could be voting for a min
 * freq lesser than what we desire and that would be honored.
 * Update cpufreq, so the min freq remains consistent in the hw.
 */
if (lmh_dcvs_available) {
    msm_lmh_dcvs_update(cpu);
    if (changed & FREQ_LIMIT_MIN)
        cpufreq_update_policy(cpu);
} else {
    cpufreq_update_policy(cpu);
}
```

Instead, we export cpu cooling device(cpu_cooling.c) to mitigate cpu scaling max freq.

Below is replacement methods:

1. [**Recommend**]Via cooling devices thermal-cpufreq-0/1:
   - /sys/class/thermal/cooling_device*/type<cur_state>, to show lmh max limit levels/frequencies.

With latest tsens script, you would get **thermal-cpufreq-0** and **thermal-cpufreq-1** column in tsens log.

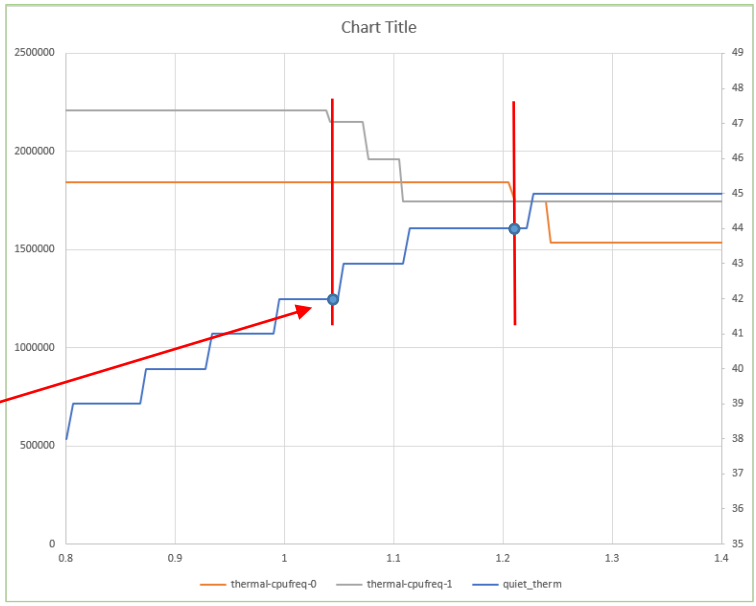
Test thermal-engine.conf as instance:

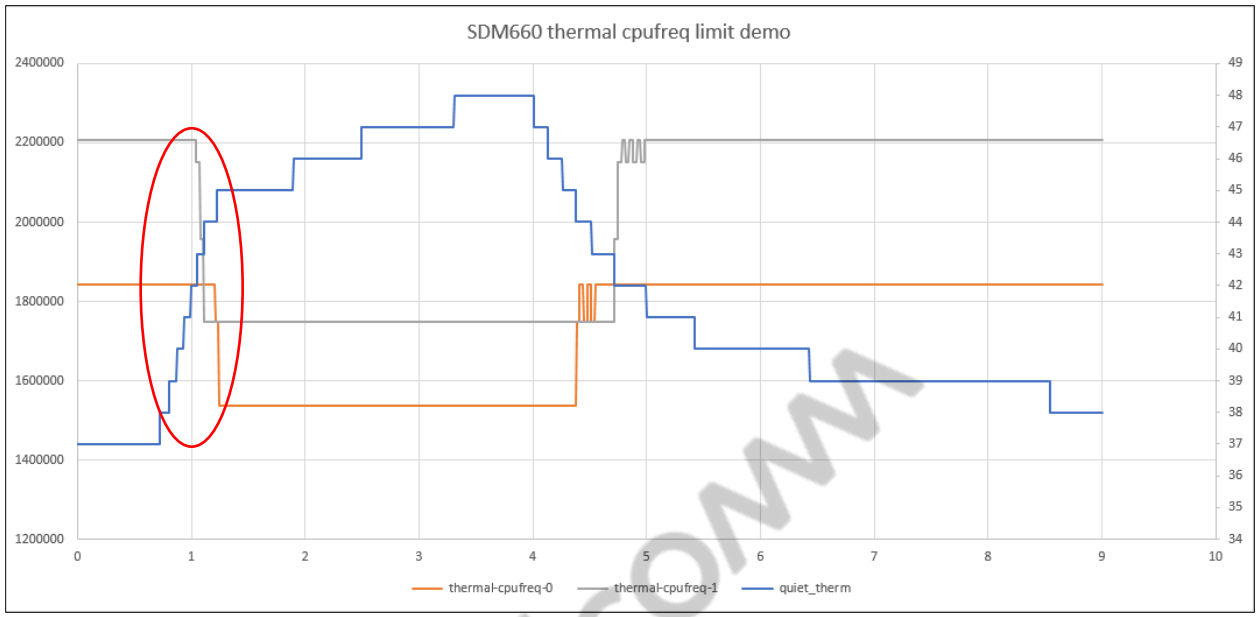
thermal-cpufreq-1(cluster1) start throttling after quiet_therm up to 42C;

thermal-cpufreq-0(cluster0) start throttling after quiet_therm keep rise to 44C.

```
[SS-QUIET-THERM-SILVER]
#algo_type ss
sampling 2000
sensor quiet_therm
device cluster0
set_point 44000
set_point_clr 41000
time_constant 0
device_max_limit 1536000

[SS-QUIET-THERM-GOLD]
#algo_type ss
sampling 2000
sensor quiet_therm
device cluster1
set_point 42000
set_point_clr 39000
time_constant 0
device_max_limit 1747200
```





2. Via clk node:
   - /d/clk/perfcl_clk/clk_measure
   - /d/clk/pwrcl_clk/clk_measure
3. [Optional] in SDM845/SDM670, you stil couold get the mitigation frequency via:
   - CPU0~3: /sys/devices/platform/soc/17d41000.qcom,cpucc/17d41000.qcom,cpucc:qcom,limits-dcvs@0/lmh_freq_limit
   - CPU4~7: /sys/devices/platform/soc/17d41000.qcom,cpucc/17d41000.qcom,cpucc:qcom,limits-dcvs@1/lmh_freq_limit
4. you can use the trace event: sched_get_task_cpu_cycles to see what the scheduler's view of the current frequency is.

Event to capture ftrace:

echo "cpufreq_interactive:* power:cpu_idle power:cpu_frequency power:clock_set_rate power:clock_enable power:clock_state power:clock_disable sched:sched_switch sched:sched_task_load sched:sched_wakeup **thermal:*** sched:sched_wakeup_new irq:* kgsl:kgsl_clk msm_low_power:* msm_bus:bus_update_request sched:sched_migrate_task sched:sched_cpu_load_cgroup sched:sched_cpu_load_lb sched:sched_cpu_load_wakeup power:wakeup_source_deactivate sched:sched_enq_deq_task power:wakeup_source_activate **sched:sched_get_task_cpu_cycles** power:bw_hwmon_meas power:bw_hwmon_update" > set_event;

“Freq=” is actual HW frequency, that is calculated by “cycle/execution time”. “legacy_freq” reflects current freq from cpufreq driver.

```
<idle>-0    [002] dnh4   382.529947: sched_wakeup: comm=FMOD stream thr pid=4080 prio=120 target_cpu=002
<idle>-0    [002] dnh2   382.529950: irq_handler_exit: irq=5 ret=handled
<idle>-0    [002] dnh3   382.529955: sched_get_task_cpu_cycles: cpu=2 event=5 cycles=57420000000 exec_time=70886 freq=810033 legacy_freq=1728000
<idle>-0    [002] .n.2   382.529958: cpu_idle: state=4294967295 cpu_id=2
<idle>-0    [002] dn.3   382.529965: sched_get_task_cpu_cycles: cpu=2 event=0 cycles=57420000000 exec_time=70886 freq=810033 legacy_freq=1728000
<idle>-0    [002] dn.3   382.529968: sched_get_task_cpu_cycles: cpu=2 event=1 cycles=57420000000 exec_time=70886 freq=810033 legacy_freq=1728000
<idle>-0    [002] d..3   382.529971: sched_switch: prev_comm=swapper/2 prev_pid=0 prev_prio=120 prev_state=R ==> next_comm=FMOD stream thr next_pid=
```

Thx.