Thermal Governors: How to pick the right one?

Linaro Connect 2019

Keerthy Jagadeesh

Agenda

- Why do we need thermal management on SoCs?
- What is the exact problem to be solved?
- Linux Thermal framework overview.
- Main parts of the linux thermal framework
- Types of Governors
- Step_wise
- Fair Share
- Bang Bang

Why do we need thermal management on SoCs?

- Multiple Cores packed in One SoC
  - Multi core CPUs
  - GPU
  - IVA/EVE engines
- Shrinking Die size
  - Nanometer shrink
- Higher clocking 1+ GHz
- Device/User safety

Problem to be solved

- Extract the optimum performance from a device.
- Ensure the temperature is adhering to the device operating temperature constraints.
- Ensure user safety as well for handheld devices.

Linux’s solution to the problem

- Thermal framework: Linux runs on most of the high power/performance SoCs. Provides an abstraction for solving the thermal problem. A generic solution to ARM & X_86 world.

Critical parts of the framework

- Thermal Zones: Software abstraction of hot hardware portions on IC(Ex: CPU, GPU, Core etc)
- Thermal sensors: The actual hardware that determines the temperature of a zone(Typically an ADC)
- Cooling Agents: Mitigation agents that cool the device. Ex: Fans, CPUFreq, Shutdown
- Trip points: The temperature points at which mitigating actions are executed.
- **Thermal governors: Policies to manage the overall functionality.**

Types of Governors

- step wise: Open loop control. Temperature threshold and trend based. Walk through each cooling device cooling state, step by step.
- fair share: Weight based. Determine the cooling device state based on assigned weight partitioning.
- bang bang: uses a hysteresis to switch abruptly on or off a cooling device. It is intended to control fans, which can not be throttled but just switched on or off.
- power allocator: Closed loop control. Based on power budget, temperature, and current power consumption of each involved device.
- user space: hand off the control of a thermal zone to user space. Example: thermald and iTux.

Step – wise Governor Logic:

If the temperature is higher than a trip point,

* a. if the trend is THERMAL_TREND_RAISING, use higher cooling state for this trip point
* b. if the trend is THERMAL_TREND_DROPPING, do nothing

If the temperature is Lower than a trip point,

* a. if the trend is THERMAL_TREND_RAISING, do nothing
* b. if the trend is THERMAL_TREND_DROPPING, use lower cooling state for this trip point, if the cooling state already equals lower limit, deactivate the thermal instance

Note: There is no usage of THERMAL_TREND_RAISE_FULL & THERMAL_TREND_DROP_FULL although there are cases mentioned.

Step wise performance: Dual Core A15 DRA74

Use case: cpuloadgen @100% on both A15 cores

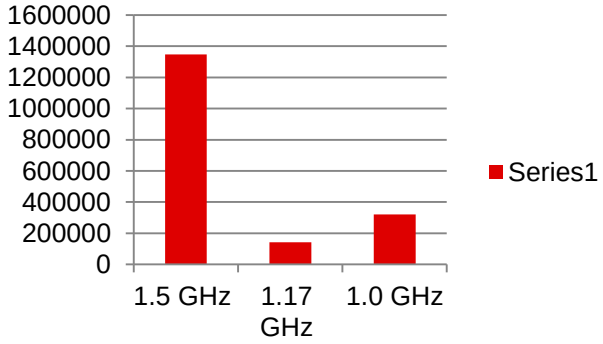
Transition Table From : To(State represents the cooling state 0 being lowest 2 being highest)

	state 0	state 1	state 2
state 0	0	272	0
state 1	272	0	272
state 2	0	272	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

state0	state1	state2
1347113	141418	321367

- Total Transitions: 1088
- Zone temperature: 66200
- trip_point_0_temp: 70000



Step wise performance: Dual Core A15 DRA74

Use case: cpuloadgen @100% on both A15 cores

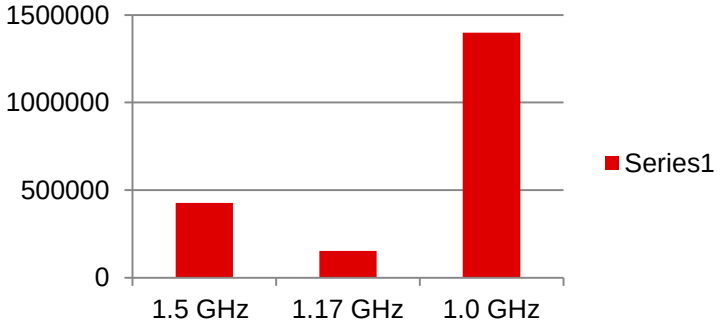
Transition Table From : To(State represents the cooling state 0 being lowest 2 being highest)

	state 0	state 1	state 2
state 0	0	295	0
state 1	294	0	295
state 2	0	294	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

state0	state1	state2
426030	153120	1398287

- Total Transitions: 1178
- Zone temperature: 66200
- trip_point_0_temp: 60000



Step wise performance on Single Core A15 DRA72

- Single Core A15 takes a long time to heat up even when CPU Is loaded 100 percent for more than couple of hours the temperature was hovering around 60C. So relatively safe to assume that step wise works well.

StepWise: On DRA76x 1.8GHz dual core A15

Use case: cpuloadgen @100% on both A15 cores

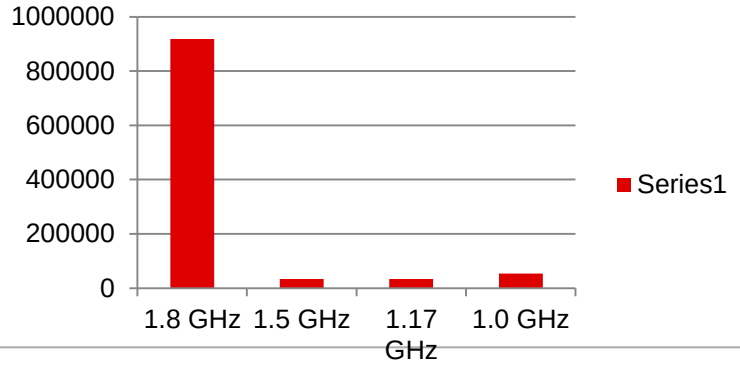
- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	66	0	0
State 1	65	0	66	0
State 2	0	65	0	63
State 3	0	0	62	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
917871	34066	33274	53681

- Total Transitions: 387
- Zone temperature: 112600
- trip_point_0_temp: 100000



StepWise: On DRA76x 1.8GHz dual core A15

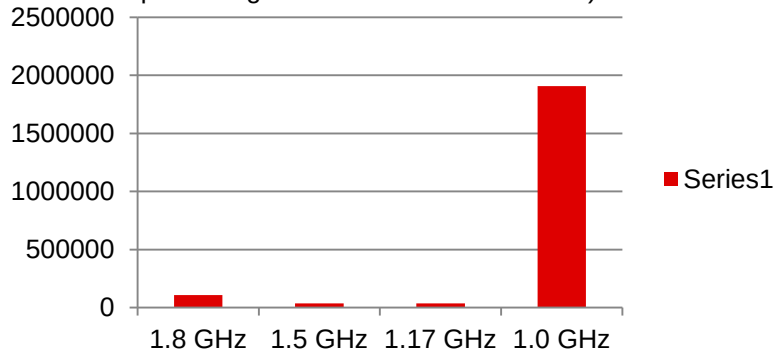
Use case: cpuloadgen @100% on both A15 cores

- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	66	0	0
State 1	65	0	66	0
State 2	0	65	0	63
State 3	0	0	62	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
106361	35143	35096	1906783



- Total Transitions: 405
- Zone temperature: 71400
- trip_point_0_temp: 60000

StepWise: On DRA76x 1.8GHz dual core A15 with 3 trips

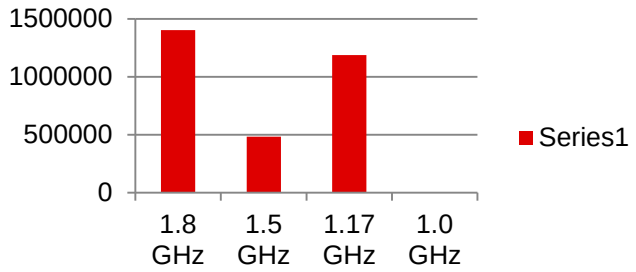
Use case: cpuloadgen @100% on both A15 cores

- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	925	0	0
State 1	924	0	925	0
State 2	0	924	0	0
State 3	0	0	0	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
1402154	482303	1187768	0



- Total Transitions: 3698
- Zone temperature: 94600
- trip_point_0_temp: 90000
- trip_point_1_temp: 100000

Step – wise Governor Pros & Cons

Pros

- Simple.
- Transitions are less when trip temperatures are significantly lower than the shutdown temperatures. Sufficient cooling with the cooling states.
- Handles cooling devices with multiple states.

Cons

- Transitions are High when trip temperatures are close to critical shutdown temperatures. Insufficient cooling with the cooling states.
- Potentially can lead to thermal runaway when passive trip points are closer to critical shutdown temperatures.
- **No hysteresis involved which means no cooling as soon as temperature is below trip temp. Should that be incorporated?**

Fair Share Governor: Logic

Three parameters to calculate the new throttle state

Parameters used for Throttling:

* P1. max_state: Maximum throttle state exposed by the cooling device.

  Ex: In case of cpufreq the highest OPP Number.

* P2. percentage[i]/100:

  How 'effective' the 'i'th device is, in cooling the given zone.
  This comes in to picture when there are multiple cooling devices for a given zone

* P3. cur_trip_level/max_no_of_trips:

  This describes the extent to which the devices should be throttled.

  We do not want to throttle too much when we trip a lower temperature,

  whereas the throttling is at full swing if we trip critical levels.

New state of the cooling device = P3 * P2 * P1

Fair Share: DRA76 (2 Trip points)

Use case: cpuloadgen @100% on both A15 cores

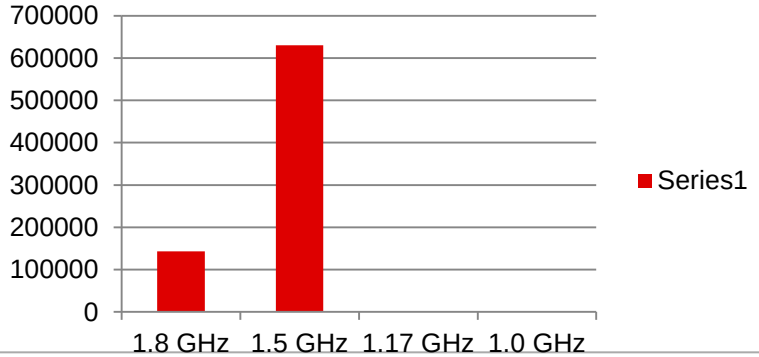
- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	27	0	0
State 1	26	0	0	0
State 2	0	0	0	0
State 3	0	0	0	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
142981	630082	0	0

- Total Transitions: 53
- Zone temperature:115400
- trip_point_0_temp: 100000



Fair Share: DRA76 (3 Trip points)

Use case: cpuloadgen @100% on both A15 cores

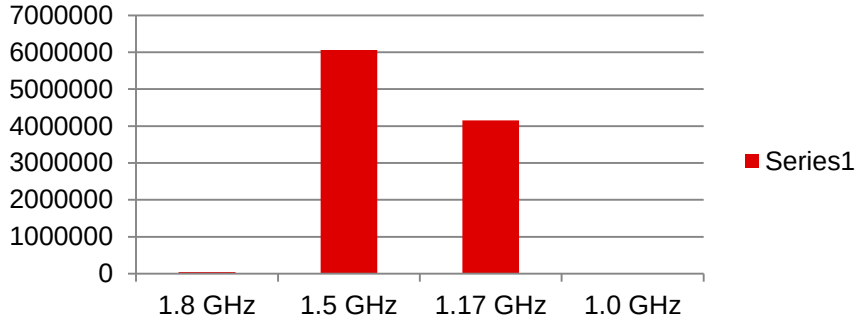
- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	12	0	0
State 1	11	0	15	0
State 2	0	15	0	0
State 3	0	0	0	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
47449	1367263	13772	0

- Total Transitions: 53
- Zone temperature: 99000 crossed 100000
- trip_point_0_temp: 90000
- trip_point_1_temp: 100000



Fair Share: DRA76 (4 Trip points)

Use case: cpuloadgen @100% on both A15 cores

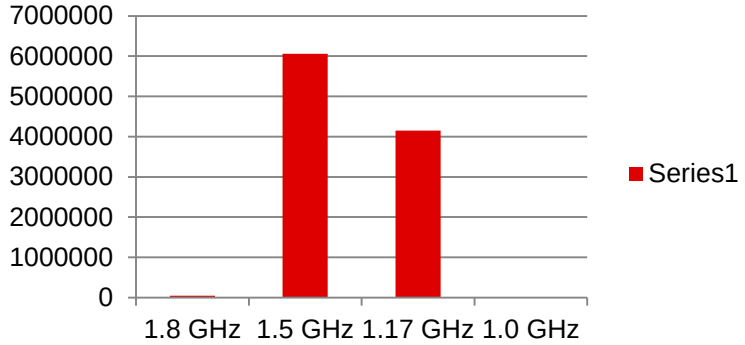
- Transition Table: From : To(State represents the cooling state 0 being lowest 3 being highest)

	State 0	State 1	State 2	State 3
State 0	0	6	0	0
State 1	5	0	2561	0
State 2	0	2650	0	0
State 3	0	0	0	0

- Transition Time for each state in ms(State represents the cooling state and the time cpu cooling was in that state since boot)

State 0	State 1	State 2	State 3
45986	6059733	4149313	0

- Total Transitions: 5132
- Zone temperature: 92200
- trip_point_0_temp: 80000
- trip_point_1_temp: 90000
- trip_point_1_temp: 100000



Fair Share performance on Single Core A15 DRA72

- Single Core A15 takes a long time to heat up even when CPU Is loaded 100 percent for more than couple of hours the temperature was hovering around 60C. So relatively safe to assume that Fair Share works well.

Fair Share Pros & Cons

Pros

- Caters to multiple cooling agents.
- Transitions are less when trip temperatures are significantly lower than the shutdown temperatures.
- One can associate weights with each cooling device so that target cooling states are chosen accordingly.
- Handles multiple trips.

Cons

- Fewer trips with multiple cooling levels. Always the target state chosen is low & results in thermal runaway.
- Thermal characterization for choosing multiple trip points is necessary.
- **Num_trips in the formula also include critical trips. Ideally it should be total number of passive trips.**

Bang Bang Governor: Logic

If the zone temperature is higher than a trip point & cooling is currently off then switch on the cooling

If the zone temperature is lower than the (trip point – hysteresis value) & cooling is currently on then switch off the cooling.

Bang Bang Governor Continued

- This could be used for Only binary cooling even with CPUfreq can be analogous to switch On/Off of fan → Switch Off/On Higher OPP.
- Hysteresis needs to be carefully set otherwise **default value is 0.** This means there will be tremendous toggling when temperature reaches the trip point.
- Multiple cooling states will not be used even if defined the max cooling state is 1.
- Ideal for Fans which do not have speed control or multiple levels.
- Can be used on SoCs that support 2 OPPs.

References

- DRA7 Public TRM: ti.com/lit/ug/sprui30f/sprui30f.pdf
- https://www.kernel.org/ Linux Mainline kernel
- https://blog.linuxplumbersconf.org/2015/ocw//system/presentations/2613/original/thermal-framework-status-no-transitioning.pdf

Thank You