International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

The Thermal Framework in the Linux Kernel

Anish Kumar

Email: *yesanishhere[at]gmail.com*

Abstract: ***The thermal framework in the Linux kernel provides a flexible and extensible architecture for managing temperature - related issues in computer systems. This paper provides an overview of the thermal framework's key components, including thermal zones, trip points, cooling devices, and thermal governors. We examine how the framework enables temperature monitoring, implements thermal policies, and coordinates cooling actions across hardware devices. The device tree bindings, sysfs interfaces, and various thermal governors are discussed in detail, along with their pros and cons. Case studies on different SoCs are presented to illustrate the performance of different governors.***

Keywords: Thermal framework, Linux kernel, cooling devices, thermal governors

1. Introduction

As modern computer systems become more powerful and compact, thermal management has become increasingly critical for ensuring reliable operation and preventing damage from overheating. The Linux kernel's thermal framework provides a standardized approach for monitoring temperatures, defining thermal policies, and coordinating cooling actions across various hardware components.

2. Motivation for Thermal Management

The need for robust thermal management in SoCs arises from several factors:

- Multiple cores packed in one SoC (multi - core CPUs, GPUs, IVA/EVE engines)
- Shrinking die sizes and nanometer - scale manufacturing
- Higher clock frequencies exceeding 1 GHz
- Device and user safety concerns, especially for handheld devices

The primary goals of thermal management are:

- Extracting optimal performance from a device
- Ensuring temperature adherence to device operating constraints
- Maintaining user safety for handheld devices

3. Thermal Framework Architecture

The thermal framework consists of several key components that work together to implement thermal management policies:

a) Thermal Zones

Thermal zones represent physical areas of a system where temperature is monitored, such as a CPU die or battery. Each thermal zone is associated with one or more temperature sensors.

b) Trip Points

Trip points define temperature thresholds at which specific actions should be taken. Common types include:

- Active: Triggers active cooling (e. g., fan speed increase)
- Passive: Initiates passive cooling (e. g., CPU frequency reduction)
- Critical: Indicates a critically high temperature requiring immediate shutdown

c) Cooling Devices

Cooling devices represent hardware or software mechanisms that can reduce temperature, such as fans, CPU frequency scaling, or GPU throttling.

d) Thermal Governors

Thermal governors implement the policy logic for responding to temperature changes. The framework provides several built - in governors:

- Step - wise: Incrementally increases cooling in steps as temperature rises
- Fair - share: Distributes cooling actions fairly across multiple devices
- Bang - bang: Uses hysteresis to switch cooling devices on or off
- Power allocator: Manages cooling based on power consumption
- User space: Hands off control to user space tools like thermald and iTux

4. Device Tree Bindings

The thermal framework can be configured through device tree bindings. Key properties include:

- Thermal - sensors: Specifies temperature sensors for a thermal zone
- Trips: Defines temperature thresholds and actions
- Cooling - maps: Associates cooling devices with specific trip points

5. SYSFS Interface

The thermal framework exposes configuration and status information through sysfs under /sys/class/thermal/. Key attributes include:

a) Thermal Zone Attributes

- type: Type of thermal zone (e. g., "acpitz" for ACPI thermal zone)
- temp: Current temperature in millidegrees Celsius
- mode: "enabled" or "disabled"
- policy: Current thermal governor
- trip point [0 - *] temp: Trip point temperatures
- trip point [0 - *] type: Trip point types (e. g., critical, passive, active)
- emul temp: Emulated temperature for debugging

b) Cooling Device Attributes
- type: Type of cooling device (e. g., "Fan", "Processor")
- max state: Maximum cooling state
- cur state: Current cooling state
- stats/: Directory with cooling statistics

6. Thermal Governors

a) Step - wise Governor
The step - wise governor uses an open - loop control system based on temperature thresholds and trends. It walks through each cooling device state step by step.

Logic:
If temperature is higher than a trip point:
- If trend is THERMAL TREND RAISING, use higher cooling state
- If trend is THERMAL TREND DROPPING, do nothing

If temperature is lower than a trip point:
- If trend is THERMAL TREND RAISING, do nothing
- If trend is THERMAL TREND DROPPING, use lower cooling state or deactivate if at lowest state

Pros:
- Simple implementation
- Handles cooling devices with multiple states
- Effective when trip temperatures are significantly lower than shutdown temperatures

Cons:
- High transitions when trip temperatures are close to critical shutdown temperatures
- Potential thermal runaway when passive trip points are close to critical temperatures
- Lacks hysteresis, which may lead to rapid state changes

b) Fair - share Governor
The fair - share governor uses a weight - based approach to determine cooling device states based on assigned weight partitioning.

Logic:
New state = (cur trip level / max no of trips) * (percentage [i] / 100) * max state
Where:
- max state: Maximum throttle state of the cooling device
- percentage [i]: Effectiveness of the i - th device in cooling the zone
- cur trip level: Current trip level
- max no of trips: Total number of trip points

Pros:
- Supports multiple cooling agents
- Allows weight assignment to cooling devices
- Handles multiple trip points

Cons:
- May result in thermal runaway with fewer trips and multiple cooling levels
- Requires careful thermal characterization for choosing trip points
- Includes critical trips in the num trips calculation

c) Bang - bang Governor
The bang - bang governor uses hysteresis to switch cooling devices on or off abruptly.

Logic:
- If temperature > trip point && cooling is off: switch on cooling
- If temperature < (trip point - hysteresis) && cooling is on: switch off cooling

Pros:
- Simple implementation
- Ideal for binary cooling devices like fans without speed control

Cons:
- Limited to binary cooling states
- Requires careful hysteresis setting to avoid rapid toggling

7. Case Studies

a) DRA74 Dual - Core A15
- Performance analysis of the step - wise governor on a DRA74 dual - core A15 SoC showed [4]:
- Total transitions: 1088 (with 70°C trip point)
- Zone temperature: 66.2°C
- Time spent in different cooling states varied based on trip point settings

b) DRA76 Dual - Core A15
- Comparison of step - wise and fair - share governors on a DRA76 dual - core A15 SoC revealed [4]:
- Step - wise governor: 387 transitions, zone temperature 112.6°C (100°C trip point)
- Fair - share governor: 53 transitions, zone temperature 115.4°C (100°C trip point)
- Fair - share governor showed fewer transitions but higher temperatures

8. Conclusion

The Linux kernel's thermal framework provides a flexible and extensible architecture for implementing thermal management policies. By standardizing interfaces between temperature sensors, cooling devices, and policy logic, it enables more robust and portable thermal solutions across diverse hardware platforms. The choice of thermal governor depends on the specific hardware characteristics and thermal requirements of the system.

References

[1] Linux Kernel Documentation, "Linux thermal framework," [Online]. Available: https: //www.kernel. org/doc/Documentation/thermal/sysfs - api. Txt
[2] T. Ram, "Linux Kernel Thermal Framework," in Embedded Linux Conference, 2020.
[3] Linux Kernel Documentation, "CPU cooling APIs How To," [Online]. Available: https: //www.kernel. org/doc/Documentation/thermal/cpu - cooling - api. Txt
[4] K. Jagadeesh, "Thermal Governors: How to pick the

right one?" Linaro Connect, 2019.
[5] Texas Instruments, "DRA7 Public TRM," [Online]. Available: ti. com/lit/ug/sprui30f/sprui30f. pdf
[6] L. Zhang, "Linux Thermal Framework Overview," in Proc. Embedded Linux Conference, 2018.
[7] V. Pallipadi and A. Belay, "cpuidle: Do nothing, efficiently," in Proc. Linux Symposium, 2007, pp.119 - 125.
[8] E. Rotem et al., "Power management architecture of the Intel microarchitecture code - named Sandy Bridge," IEEE Micro, vol.32, no.2, pp.20 - 27, 2012.
[9] Z. Toprak - Deniz et al., "Distributed power delivery for energy efficient and low power systems," in Proc.49th ACM/EDAC/IEEE Design Automation Conference (DAC), 2012, pp.914 - 919.
[10] K. Skadron et al., "Temperature - aware microarchitecture: Modeling and implementation," ACM Transactions on Architecture and Code Optimization, vol.1, no.1, pp.94 - 125, 2004.
[11] Linux Kernel Documentation, "Linux Thermal Framework," [Online]. Available: https: //www.kernel. org/doc/html/latest/driver - api/thermal/
[12] R. Minnich, "The Linux ACPI Thermal Driver," in Proc. Ottawa Linux Symposium, 2001.
[13] N. Vallina - Rodriguez and J. Crowcroft, "Energy Management Techniques in Modern Mobile Handsets," IEEE Communications Surveys & Tutorials, vol.15, no.1, pp.179 - 198, 2013.