Linux Thermal: User Kernel Interface

Objective

- Avoid polling
- Fast actions from user space thermal solution
- Differentiate between temperature reporting and trip updates
- Add new notification mechanism
- Additional custom attributes

Temperature thresholds

- Optional temperature thresholds

```
sys/class/thermal/
├── thermal_zone_*
│   ├── temp_notify_threshold_high
│   ├── temp_notify_threshold_hyst
│   ├── temp_notify_threshold_low
│   └── ...
└── ...
```

Kernel-User notifications

- Only active when zone is enabled and user space gov
- Common notify on a char_dev

```
sys/class/thermal/
├── thermal_zone_*
└── thermal_notify---->/dev/thermal_notify
```

- A kfifo based
- User space can select/poll
- A generic structure with
  - thermal_zone_id
  - notification type
  - notification data

Thermal Notification codes

- Notifications
  - THERMAL_ZONE_CREATE
  - THERMAL_ZONE_DELETE
  - THERMAL_ZONE_DISABLED
  - THERMAL_ZONE_ENABLED
  - THERMAL_TEMP_LOW_THRES
  - THERMAL_TEMP_HIGH_THRES
  - THERMAL_TRIP_UPDATE
  - THERMAL_TRIP_ADD
  - THERMAL_TRIP_DELETE

Custom Attributes

- Per zone and cdev custom attributes/attribute group
  - Example

    Get: running_average_temperature

    Set: load conversion_tables to firmware
- Similar to

```
struct cpufreq_driver xx_driver = {
        ..
        .attr = private_attributes,
}
```

Handle Critical/Hot Trip

- Kernel driver powers off even for user space governor
  - Problem with transient temperature spikes

Thermal Zone Mode Control

Zhang Rui

Issues 1 (initialization)

- thermal_zone_device_update() invoked immediately during thermal zone device registration, and .get_temp() may be not ready
- status:
  - workaround in of_thermal code by setting dummy get_temp()

Issues 2 (initialization)

- issue:
  - themal_zone_device_register()/[devm]_thermal_zone_of_sensor_register() needs to be called first to get thermal_zone_device structure
  - request driver specific IRQ handler
  - thermal->chip->control() (IRQ can be fired then)
  - We need a mechanism to make sure get_temp() is not poked before thermal->chip->control() and is ready to work right after it.
- solution for DT thermal:
  - https://patchwork.kernel.org/patch/10645813/
  - split register, enable and update
  - Mark thermal zone as ready but don't update thermal zone from thermal/of_thermal core code before step 3
  - Update thermal zone from platform thermal driver explicitly after step 3

Proposal to fix initialization issues

- introduce tz->enable
- thermal_zone_device_register() don't call thermal_zone_device_update(), just register the sysfs and data structure
- thermal_zone_device_enable() checks the driver callbacks and set tz->enable to true.
- thermal_zone_device_update() no change, invoked by platform thermal.
- thermal_zone_set_mode() set/clear tz->enable
- then we don't the dummy callbacks and __thermal_zone->mode in of_thermal?

Issues 3

- Polling timer always running, even for a disabled thermal zone
- Status:
  - workaround in of_thermal code by setting polling delay to 0 when disable thermal zones
- Proposal:
  - check tz->enabled and don't rearm the polling timer in thermal_zone_device_update()

Issues 4 (userspace)

- Userspace tool always pokes temp sysfs attributes directly and get error return value and error messages
- Proposal:
  - always check “mode” sysfs attribute before poking the other sysfs attributes?
  - register/unregister hwmon sysfs I/F when thermal zone is enabled/disabled