Generic Thermal Sysfs driver How To

Written by Sujith Thomas <sujith.thomas@intel.com>, Zhang Rui
<rui.zhang@intel.com>

Updated: 2 January 2008

0. Introduction

The generic thermal sysfs provides a set of interfaces for thermal zone devices (sensors) and thermal cooling devices (fan, processor...) to register with the thermal management solution and to be a part of it.

This how-to focuses on enabling new thermal zone and cooling devices to participate in thermal management.
This solution is platform independent and any type of thermal zone devices and cooling devices should be able to make use of the infrastructure.

The main task of the thermal sysfs driver is to expose thermal zone attributes as well as cooling device attributes to the user space.
An intelligent thermal management application can make decisions based on inputs from thermal zone attributes (the current temperature and trip point temperature) and throttle appropriate devices.

[0-*]   denotes any positive number starting from 0
[1-*]   denotes any positive number starting from 1

1. thermal sysfs driver interface functions

1.1 thermal zone device interface

1.1.1 struct thermal_zone_device *thermal_zone_device_register(char *name, int trips, void *devdata, struct thermal_zone_device_ops *ops)

This interface function adds a new thermal zone device (sensor) to /sys/class/thermal folder as thermal_zone[0-*]. It tries to bind all the thermal cooling devices registered at the same time.

name: the thermal zone name.
trips: the total number of trip points this thermal zone supports.
devdata: device private data
ops: thermal zone device call-backs.
- .bind: bind the thermal zone device with a thermal cooling device.
- .unbind: unbind the thermal zone device with a thermal cooling device.
- .get_temp: get the current temperature of the thermal zone.
- .get_mode: get the current mode (user/kernel) of the thermal zone.
  - "kernel" means thermal management is done in kernel.
  - "user" will prevent kernel thermal driver actions upon trip points so that user applications can take charge of thermal management.
- .set_mode: set the mode (user/kernel) of the thermal zone.
- .get_trip_type: get the type of certain trip point.
- .get_trip_temp: get the temperature above which the certain trip point will be fired.

1.1.2 void thermal_zone_device_unregister(struct thermal_zone_device *tz)

    This interface function removes the thermal zone device.
    It deletes the corresponding entry form /sys/class/thermal folder and
    unbind all the thermal cooling devices it uses.

1.2 thermal cooling device interface
1.2.1 struct thermal_cooling_device *thermal_cooling_device_register(char *name,
                void *devdata, struct thermal_cooling_device_ops *)

    This interface function adds a new thermal cooling device
(fan/processor/...)
    to /sys/class/thermal/ folder as cooling_device[0-*]. It tries to bind
itself
    to all the thermal zone devices register at the same time.
    name: the cooling device name.
    devdata: device private data.
    ops: thermal cooling devices call-backs.
        .get_max_state: get the Maximum throttle state of the cooling device.
        .get_cur_state: get the Current throttle state of the cooling device.
        .set_cur_state: set the Current throttle state of the cooling device.

1.2.2 void thermal_cooling_device_unregister(struct thermal_cooling_device
*cdev)

    This interface function remove the thermal cooling device.
    It deletes the corresponding entry form /sys/class/thermal folder and
    unbind itself from all the thermal zone devices using it.

1.3 interface for binding a thermal zone device with a thermal cooling device
1.3.1 int thermal_zone_bind_cooling_device(struct thermal_zone_device *tz,
                int trip, struct thermal_cooling_device *cdev);

    This interface function bind a thermal cooling device to the certain trip
    point of a thermal zone device.
    This function is usually called in the thermal zone device .bind callback.
    tz: the thermal zone device
    cdev: thermal cooling device
    trip: indicates which trip point the cooling devices is associated with
          in this thermal zone.

1.3.2 int thermal_zone_unbind_cooling_device(struct thermal_zone_device *tz,
                int trip, struct thermal_cooling_device *cdev);

    This interface function unbind a thermal cooling device from the certain
    trip point of a thermal zone device. This function is usually called in
    the thermal zone device .unbind callback.
    tz: the thermal zone device
    cdev: thermal cooling device
    trip: indicates which trip point the cooling devices is associated with
          in this thermal zone.

2. sysfs attributes structure

RO   read only value

RW   read/write value

Thermal sysfs attributes will be represented under /sys/class/thermal.
Hwmon sysfs I/F extension is also available under /sys/class/hwmon
if hwmon is compiled in or built as a module.

Thermal zone device sys I/F, created once it's registered:
/sys/class/thermal/thermal_zone[0-*]:

```
    |---type:                    Type of the thermal zone
    |---temp:                    Current temperature
    |---mode:                    Working mode of the thermal zone
    |---trip_point_[0-*]_temp:   Trip point temperature
    |---trip_point_[0-*]_type:   Trip point type
```

Thermal cooling device sys I/F, created once it's registered:
/sys/class/thermal/cooling_device[0-*]:

```
    |---type:                    Type of the cooling device(processor/fan/...)
    |---max_state:               Maximum cooling state of the cooling device
    |---cur_state:               Current cooling state of the cooling device
```

Then next two dynamic attributes are created/removed in pairs. They represent
the relationship between a thermal zone and its associated cooling device.
They are created/removed for each successful execution of
thermal_zone_bind_cooling_device/thermal_zone_unbind_cooling_device.

/sys/class/thermal/thermal_zone[0-*]:

```
    |---cdev[0-*]:               [0-*]th cooling device in current thermal zone
    |---cdev[0-*]_trip_point:    Trip point that cdev[0-*] is associated with
```

Besides the thermal zone device sysfs I/F and cooling device sysfs I/F,
the generic thermal driver also creates a hwmon sysfs I/F for each _type_
of thermal zone device. E.g. the generic thermal driver registers one hwmon
class device and build the associated hwmon sysfs I/F for all the registered
ACPI thermal zones.

/sys/class/hwmon/hwmon[0-*]:

```
    |---name:                    The type of the thermal zone devices
    |---temp[1-*]_input:         The current temperature of thermal zone [1-*]
    |---temp[1-*]_critical:      The critical trip point of thermal zone [1-*]
```

Please read Documentation/hwmon/sysfs-interface for additional information.

****
* Thermal zone attributes *
****

type
        Strings which represent the thermal zone type.
        This is given by thermal zone driver as part of registration.
        E.g: "acpitz" indicates it's an ACPI thermal device.
        In order to keep it consistent with hwmon sys attribute; this should
        be a short, lowercase string, not containing spaces nor dashes.
        RO, Required

temp

Current temperature as reported by thermal zone (sensor).
Unit: millidegree Celsius
RO, Required

mode
One of the predefined values in [kernel, user].
This file gives information about the algorithm that is currently managing the thermal zone. It can be either default kernel based algorithm or user space application.
kernel = Thermal management in kernel thermal zone driver.
user = Preventing kernel thermal zone driver actions upon trip points so that user application can take full charge of the thermal management.
RW, Optional

trip_point_[0-*]_temp
The temperature above which trip point will be fired.
Unit: millidegree Celsius
RO, Optional

trip_point_[0-*]_type
Strings which indicate the type of the trip point.
E.g. it can be one of critical, hot, passive, active[0-*] for ACPI thermal zone.
RO, Optional

cdev[0-*]
Sysfs link to the thermal cooling device node where the sys I/F for cooling device throttling control represents.
RO, Optional

cdev[0-*]_trip_point
The trip point with which cdev[0-*] is associated in this thermal zone; -1 means the cooling device is not associated with any trip point.
RO, Optional

passive
Attribute is only present for zones in which the passive cooling policy is not supported by native thermal driver. Default is zero and can be set to a temperature (in millidegrees) to enable a passive trip point for the zone. Activation is done by polling with an interval of 1 second.
Unit: millidegrees Celsius
Valid values: 0 (disabled) or greater than 1000
RW, Optional

****
* Cooling device attributes *
****

type
String which represents the type of device, e.g:
- for generic ACPI: should be "Fan", "Processor" or "LCD"
- for memory controller device on intel_menlow platform:
should be "Memory controller".

RO, Required

max_state
The maximum permissible cooling state of this cooling device.
RO, Required

cur_state
The current cooling state of this cooling device.
The value can any integer numbers between 0 and max_state:
- cur_state == 0 means no cooling
- cur_state == max_state means the maximum cooling.
RW, Required

3. A simple implementation

ACPI thermal zone may support multiple trip points like critical, hot, passive, active. If an ACPI thermal zone supports critical, passive, active[0] and active[1] at the same time, it may register itself as a thermal_zone_device (thermal_zone1) with 4 trip points in all.
It has one processor and one fan, which are both registered as thermal_cooling_device.

If the processor is listed in _PSL method, and the fan is listed in _AL0 method, the sys I/F structure will be built like this:

/sys/class/thermal:

```
|thermal_zone1:
    |---type:                   acpitz
    |---temp:                   37000
    |---mode:                   kernel
    |---trip_point_0_temp:      100000
    |---trip_point_0_type:      critical
    |---trip_point_1_temp:      80000
    |---trip_point_1_type:      passive
    |---trip_point_2_temp:      70000
    |---trip_point_2_type:      active0
    |---trip_point_3_temp:      60000
    |---trip_point_3_type:      active1
    |---cdev0:                  --->/sys/class/thermal/cooling_device0
    |---cdev0_trip_point:       1       /* cdev0 can be used for passive */
    |---cdev1:                  --->/sys/class/thermal/cooling_device3
    |---cdev1_trip_point:       2       /* cdev1 can be used for active[0]*/

|cooling_device0:
    |---type:                   Processor
    |---max_state:              8
    |---cur_state:              0

|cooling_device3:
    |---type:                   Fan
    |---max_state:              2
    |---cur_state:              0
```

/sys/class/hwmon:

```
|hwmon0:
    |---name:                   acpitz
    |---temp1_input:            37000
    |---temp1_crit:             100000
```