NAME

thermal-conf.xml − Configuration file for thermal daemon

SYNOPSIS

$(TDCONFDIR)/etc/thermald/thermal-conf.xml

DESCRIPTION

thermal-conf.xml is a configuration file for the thermal daemon. It is used to configure thermal sensors, zone and cooling devices.The location of this file depends on the configuration option used during build time.

The terminology used in this file conforms to "Advanced Configuration and Power Interface Specification". The ACPI thermal model is based around conceptual platform regions called thermal zones that physically contain devices, thermal sensors, and cooling controls. For example of a thermal zone can be a CPU or a laptop cover. A zone can contain multiple sensors for monitoring temperature. A cooling device provides interface to reduce the temperature of a source device, which causes increase in the temperature. An example of a cooling device is a FAN or some Linux driver which can throttle the source device.

A thermal zone configuration includes one or more trip points. A trip point is a temperature at which a cooling device needs to be activated.

A cooling device can be either active or passive. An example of an active device is a FAN, which will not reduce performance at the cost of consuming more power and noise. A passive device uses performance throttling to control temperature. In addition to cooling devices present in the thermal sysfs, the following cooling devices are built into the thermald, which can be used as valid cooling device type:

- rapl_controller
- intel_pstate
- cpufreq
- LCD

The thermal sysfs under Linux (/sys/class/thermal) provides a way to represent per platform ACPI configuration. The kernel thermal governor uses this data to keep the platform thermals under control. But there are some limitations, which thermald tries to resolve. For example:

- If the ACPI data is not optimized or buggy. In this case thermal-conf.xml can be used to correct the behavior without change in BIOS.

  - There may be thermal zones exposed by the thermal sysfs without associated cooling actions. In this case thermal conf.xml can be used to tie the cooling devices to those zones.

  - The best cooling method may not be in the thermal sysfs. In this case thermal-conf.xml can be used to bind a zone to an external cooling device.

  - Specify thermal relationships. A zone can be influenced by multiple source devices with varying degrees. In this case thermal-conf.xml can be used to define the relative influence for apply compensation.

FILE FORMAT

The configuration file format conforms to XML specifications. A set of tags defined to define platform, sensors, zones, cooling devices and trip points.

```
<ThermalConfiguration>
  <Platform>
    <Name>Example Platform Name</Name>
    <!-- UUID is optional, if present this will be matched. Both product
         name and UUID can contain wild card "*", which matches any
         platform. -->
```

```
<UUID>Example UUID</UUID>
<!-- configuration file format conforms to XML specifications. A
     set of tags defined to define platform, sensors, zones, cooling
     devices and trip points. -->
<ProductName>Example Product Name</ProductName>
<Preference>QUIET|PERFORMANCE</Preference>
<!-- Quiet mode will only use passive cooling device. "PERFORMANCE"
     will only select active devices. -->
<ThermalSensors>
  <ThermalSensor>
    <!-- New Sensor with a type and path -->
    <Type>example_sensor_1</Type>
    <Path>/some_path</Path>
    <AsyncCapable>0</AsyncCapable>
  </ThermalSensor>
  <ThermalSensor>
    <!-- Already present in thermal sysfs, enable this or
         add/change config For example, here we are indicating
         that sensor can do async events to avoid polling. -->
    <Type>example_thermal_sysfs_sensor</Type>
    <!-- If async capable, then we don’t need to poll. -->
    <AsyncCapable>1</AsyncCapable>
  </ThermalSensor>
</ThermalSensors>
<ThermalZones>
  <ThermalZone>
    <Type>Example Zone type</Type>
    <TripPoints>
      <TripPoint>
        <SensorType>example_sensor_1</SensorType>
        <!-- Temperature at which to take action. -->
        <Temperature> 75000 </Temperature>
        <!-- max/passive/active If a MAX type is specified, then
             daemon will use PID control to aggressively throttle
             to avoid reaching this temp. -->
        <type>max</type>
        <!-- SEQUENTIAL | PARALLEL. When a trip point temp is
             violated, then number of cooling devices can be
             activated. If control type is SEQUENTIAL then, it
             will exhaust first cooling device
             before trying next. -->
        <ControlType>SEQUENTIAL</ControlType>
        <CoolingDevice>
          <index>1</index>
          <type>example_cooling_device</type>
          <!-- Influence will be used order cooling devices. First
               cooling device will be used, which has highest
               influence. -->
          <influence> 100 </influence>
          <!-- Delay in using this cdev, this takes some time too
               actually cool a zone. -->
          <SamplingPeriod> 12 </SamplingPeriod>
          <!-- Set a specific state of this cooling device when this
               trip is violated. -->
```

```
              <TargetState> 6 </TargetState>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
    <CoolingDevices>
      <CoolingDevice>
        <!-- Cooling device can be specified by a type and optionally
             a sysfs path. If the type is already present in thermal
             sysfs, there is no need of a path. Compensation can use
             min/max and step size to increasing cool the system.
             Debounce period can be used to force a waiting period
             for action. -->
        <Type>example_cooling_device</Type>
        <MinState>0</MinState>
        <IncDecStep>10</IncDecStep>
        <ReadBack> 0 </ReadBack>
        <MaxState>50</MaxState>
        <DebouncePeriod>5000</DebouncePeriod>
        <!-- If there are no PID parameters, compensation increase step
             wise and exponentially (if single step is not able to
             change trend).
             Alternatively a PID parameters can be specified then next
             step will use PID calculation using provided PID
             constants. -->
        <PidControl>
          <kp>0.001</kp>
          <kd>0.0001</kd>
          <ki>0.0001</ki>
        </PidControl>
        <!-- Write some prefix attached to state value, like below the
             prefix is "level ". It will preserve spaces as entered
             when writing to sysfs. -->
        <WritePrefix>level </WritePrefix>
      </CoolingDevice>
    </CoolingDevices>
  </Platform>
</ThermalConfiguration>
```

EXAMPLE CONFIGURATIONS

Example 1: This is a very simple configuration, to change the passive limit on the CPU. Instead of default, this new temperature 86C in the configuration is used. This will start cooling, once the temperature reaches 86C.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
    <Name>Override CPU default passive</Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>cpu</Type>
        <TripPoints>
```

```
          <TripPoint>
            <Temperature>86000</Temperature>
            <type>passive</type>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
  </Platform>
</ThermalConfiguration>
```

Example 2: In this configuration, we are controlling backlight when some sensor "SEN2" reaches 60C. Here "LCD" is a standard cooling device, which uses Linux backlight sysfs interface. "LCD_Zone" is a valid thermal zone in Linux thermal sysfs on the test platform, hence we don’t need to provide path for sysfs for "LCD_Zone". The Linux thermal sysfs is already parsed and loaded by thermald program.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
    <Name>Change Backlight</Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>LCD_Zone</Type>
        <TripPoints>
          <TripPoint>
            <SensorType>SEN2</SensorType>
            <Temperature>60000</Temperature>
            <type>passive</type>
            <CoolingDevice>
              <Type>LCD</Type>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
  </Platform>
</ThermalConfiguration>
```

Example 3: In this example Lenovo Thinkpad X220 and fan speed is controlled. Here a cooling device "_Fan", can be controlled via sysfs /sys/devices/platform/thinkpad_hwmon/pwm1. When the x86_pkg_temp reaches 45C, Fan is started with increasing speeds, if the temperature can’t be controlled at 45C.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
    <Name>Lenovo ThinkPad X220</Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>x86_pkg_temp</Type>
        <TripPoints>
          <TripPoint>
            <SensorType>x86_pkg_temp</SensorType>
```

```
            <Temperature>45000</Temperature>
            <type>passive</type>
            <ControlType>SEQUENTIAL</ControlType>
            <CoolingDevice>
              <index>1</index>
              <type>_Fan</type>
              <influence> 100 </influence>
              <SamplingPeriod> 12 </SamplingPeriod>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
    <CoolingDevices>
      <CoolingDevice>
        <Type>_Fan</Type>
        <Path>/sys/devices/platform/thinkpad_hwmon/pwm1</Path>
        <MinState>0</MinState>
        <IncDecStep>30</IncDecStep>
        <ReadBack> 0 </ReadBack>
        <MaxState>255</MaxState>
        <DebouncePeriod>5</DebouncePeriod>
      </CoolingDevice>
    </CoolingDevices>
  </Platform>
</ThermalConfiguration>
```

Example 4: The following example shows how PID can be used. Here once temperature exceeds 80C, compensation is calculated using PID using 80C as set point of PID. The compensation depends on error from the set point. Here the default built in processor cooling device is used with min state as 0 and max state as 10.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
    <Name>Use PID param </Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>x86_pkg_temp</Type>
        <TripPoints>
          <TripPoint>
            <SensorType>x86_pkg_temp</SensorType>
            <Temperature>80000</Temperature>
            <type>passive</type>
            <ControlType>SEQUENTIAL</ControlType>
            <CoolingDevice>
              <type>Processor</type>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
    <CoolingDevices>
```

```
      <CoolingDevice>
        <Type>Processor</Type>
        <PidControl>
          <kp>0.0002</kp>
          <kd>0</kd>
          <ki>0</ki>
        </PidControl>
      </CoolingDevice>
    </CoolingDevices>
  </Platform>
</ThermalConfiguration>
```

Example 5: The following example shows how to control Fan when the sysfs expects some string prefix. For example instead of just write a number to fan control sysfs, the interface requires "level " in front of the speed index value.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
    <Name>Use Fan control first then CPU throttle </Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>x86_pkg_temp</Type>
        <TripPoints>
          <TripPoint>
            <SensorType>x86_pkg_temp</SensorType>
            <Temperature>80000</Temperature>
            <type>passive</type>
            <ControlType>SEQUENTIAL</ControlType>
            <CoolingDevice>
              <type>_fan_</type>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
    <CoolingDevices>
      <CoolingDevice>
        <Type>_fan_</Type>
        <Path>/proc/acpi/ibm/fan</Path>
        <WritePrefix>level </WritePrefix>
        <MinState>0</MinState>
        <MaxState>5</MaxState>
        <DebouncePeriod>10</DebouncePeriod>
      </CoolingDevice>
    </CoolingDevices>
  </Platform>
</ThermalConfiguration>
```

Example 6: Similar to example 5, but write different speeds at different temperatures.

```
<?xml version="1.0"?>
<ThermalConfiguration>
  <Platform>
```

```
    <Name>Use Fan control first then CPU throttle </Name>
    <ProductName>*</ProductName>
    <Preference>QUIET</Preference>
    <ThermalZones>
      <ThermalZone>
        <Type>x86_pkg_temp</Type>
        <TripPoints>
          <TripPoint>
            <SensorType>x86_pkg_temp</SensorType>
            <Temperature>80000</Temperature>
            <type>passive</type>
            <CoolingDevice>
              <type>_fan_</type>
              <TargetState>1</TargetState>
            </CoolingDevice>
          </TripPoint>
          <TripPoint>
            <SensorType>x86_pkg_temp</SensorType>
            <Temperature>85000</Temperature>
            <type>passive</type>
            <CoolingDevice>
              <type>_fan_</type>
              <TargetState>2</TargetState>
            </CoolingDevice>
          </TripPoint>
        </TripPoints>
      </ThermalZone>
    </ThermalZones>
    <CoolingDevices>
      <CoolingDevice>
        <Type>_fan_</Type>
        <Path>/proc/acpi/ibm/fan</Path>
        <WritePrefix>level </WritePrefix>
        <MinState>0</MinState>
        <MaxState>5</MaxState>
        <DebouncePeriod>10</DebouncePeriod>
      </CoolingDevice>
    </CoolingDevices>
  </Platform>
</ThermalConfiguration>
```