Thermal control-CPU frequency

The following content applies to the `RDK X3` and `RDK X3 Module` development boards, not to the `RDK Ultra` development board.

1. Thermal control

To avoid overheating of the chip under heavy load, a certain degree of power management is performed at the operating system level.

The SoC has an internal temperature sensor, which is monitored by the thermal subsystem.

- ### Main temperature points

  - **Startup temperature:**The highest temperature when the system starts. If the temperature exceeds this temperature, the CPU and BPU will be downclocked immediately when the system starts.

    You can get the current configuration value through the command `cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_0_temp`. The default value is 80000 (80 degrees Celsius).

  - **Downclocking temperature:**The downclocking temperature point of the system CPU and BPU. When the temperature exceeds this temperature point, the CPU and BPU will reduce the operating frequency to reduce the power consumption of the SoC.

    The CPU is reduced to 240MHz at the lowest and the BPU is reduced to 400MHz at the lowest.

    You can get the current configuration value through the command `cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_1_temp`. The default value is 95000 (95℃).

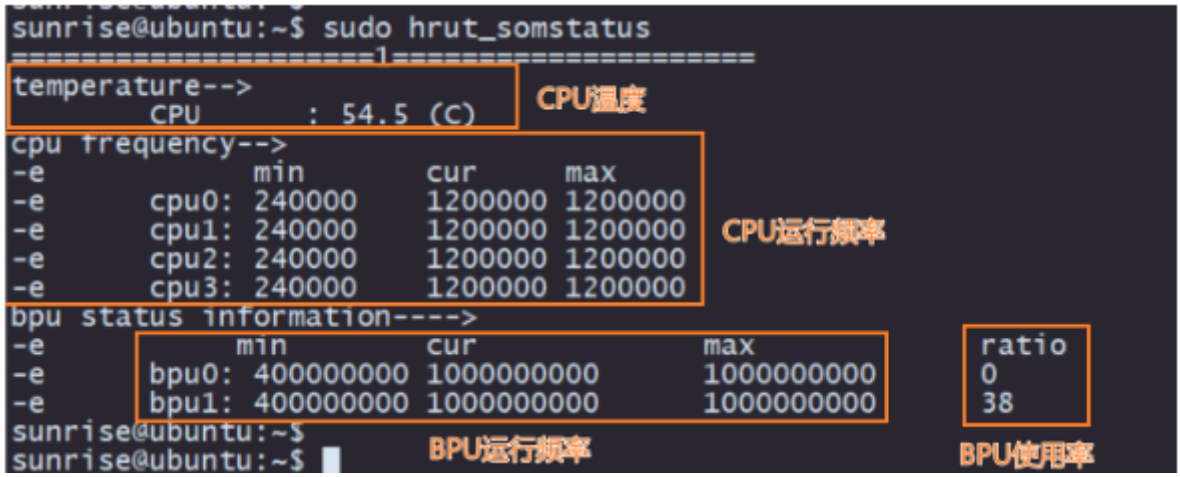
  - **Downtime temperature:** The system downtime temperature point. If the temperature exceeds this temperature, the system will shut down to protect the chip and hardware.

    It is recommended to do a good job of heat dissipation for the device to avoid device downtime, because the device will not automatically restart after downtime.

    The user needs to manually power off the development board and restart it. The current configuration value can be obtained through the command `cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_2_temp`. The default value is 105000 (105℃).

```
sunrise@ubuntu:/$ cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_0_te
mp
80000
sunrise@ubuntu:/$ cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_1_te
mp
95000
sunrise@ubuntu:/$ cat /sys/devices/virtual/thermal/thermal_zone0/trip_point_2_te
mp
105000
sunrise@ubuntu:/$
```

You can use the `sudo hrut_somstatus` command to view the current chip operating frequency, temperature and other status.



- **Set temperature point**

The system frequency reduction and shutdown temperature can be temporarily set through commands.

The set frequency reduction temperature cannot exceed the shutdown temperature, and the shutdown temperature cannot be set to more than 105℃.

For example, set 85 degrees Celsius as the frequency reduction temperature point.

```
echo 85000 > /sys/devices/virtual/thermal/thermal_zone0/trip_point_1_temp
```

For example, to set 105℃ as the shutdown temperature point.

```
echo 105000 > /sys/devices/virtual/thermal/thermal_zone0/trip_point_2_temp
```

The above command setting method will be restored to the default value after the system restarts and needs to be reset.

It can be added to the auto-startup item for automatic configuration.

2. CPU frequency management

In the Linux kernel, there is a built-in cpufreq subsystem to control the CPU frequency and frequency control strategy.

Enter the directory `/sys/devices/system/cpu/cpufreq/policy0`, and `ls`, you will see the following files in the directory.

```
```

```
affected_cpus                       // The CPU cores currently affected by the
control (CPUs in offline state are not displayed)
cpuinfo_cur_freq                    // Current CPU frequency (unit: KHz)
cpuinfo_max_freq                    // The highest frequency available to the CPU
under the current frequency adjustment strategy (unit: KHz)
cpuinfo_min_freq                    // The lowest frequency available to the CPU
under the current frequency adjustment strategy (unit: KHz)
cpuinfo_transition_latency          // The time required for the processor to
switch frequency (unit: ns)
related_cpus                        // CPU cores are affected by this control
strategy (including all CPUs online+offline)
scaling_available_frequencies       // Main frequency list supported by PU (unit:
KHz)
scaling_available_governors         // All governor types supported by the
current kernel
scaling_boost_frequencies           // List of CPU main frequencies supported in
boost mode (unit: KHz)
scaling_cur_freq                    // Saves the current CPU frequency cached by
the cpufreq module. No checks are performed on the CPU hardware registers.
scaling_disable_freq                // The CPU frequency cannot be set. Only one
can be set.
scaling_driver                      // Current FM drive in use
scaling_governor                    // Governor (frequency regulation) strategy
scaling_max_freq                    // The highest frequency available to the CPU
under the current frequency adjustment strategy (read from the cpufreq module
cache)
scaling_min_freq                    // The lowest frequency available to the CPU
under the current frequency scaling policy (read from the cpufreq module cache)
scaling_setspeed                    // You need to switch the governor to
userspace to use it. Echo the value in this file to switch the frequency
```

The Linux kernel used by the RDK system supports the following frequency adjustment strategies.

- Performance: Always put the CPU in the state of highest energy consumption and highest performance, that is, the highest frequency supported by the hardware.
- Powersave: Always put the CPU in the state of lowest energy consumption and worst performance, that is, the lowest frequency supported by the hardware.
- Ondemand: Check the load regularly and adjust the frequency according to the load. When the load is low, adjust to the lowest frequency that can just meet the current load requirements. When the load is high, immediately increase to the highest performance state.
- Conservative: Similar to the ondemand strategy, check the load regularly and adjust the frequency according to the load. When the load is low, adjust to the lowest frequency that can just meet the current load requirements. But when the load is high, it is not immediately set to the highest performance state, but the main frequency is gradually increased.
- Userspace: Open the control interface to users through sysfs, and users can customize the strategy and manually adjust the frequency in user space.
- Scheduling information (schedutil): This is a strategy introduced since Linux-4.7. Its principle is to adjust the frequency based on the CPU utilization information provided by the scheduler. The effect is similar to the ondemand strategy, but it is more accurate and natural (because the scheduler has the best CPU usage).

Users can control the CPU frequency adjustment policy by controlling the corresponding settings in the directory `/sys/devices/system/cpu/cpufreq/policy0`.

For example, to make the CPU run in performance mode.

```
sudo bash -c "echo performance >
/sys/devices/system/cpu/cpufreq/policy0/scaling_governor"
```

Or control the CPU to run at a fixed frequency (1GHz).

```
sudo bash -c "echo userspace >
/sys/devices/system/cpu/cpufreq/policy0/scaling_governor"
sudo bash -c "echo 1000000 >
/sys/devices/system/cpu/cpufreq/policy0/scaling_setspeed"
```

- ## CPU Overclocking

  The development board uses the CPU Freq driver to manage the CPU working state. The default mode is the `ondemand` mode. At this time, the CPU working frequency will be dynamically adjusted according to the load to save power.

  Users can change to the `performance` mode to make the CPU always run at the highest frequency. The command is as follows.

  ```
  sudo bash -c 'echo performance >
  /sys/devices/system/cpu/cpufreq/policy0/scaling_governor'
  ```

  The development board provides an overclocking function in the system, which can increase the maximum CPU frequency from 1.2GHz to 1.5GHz.

  The configuration command is as follows.

  ```
  sudo bash -c 'echo 1 > /sys/devices/system/cpu/cpufreq/boost'
  ```

  The CPU frequency configured using the above command is only effective during the current operation.

  If the device is restarted, the default configuration will be restored.

  Note: CPU overclocking will increase the power consumption and heat generation of the chip.

  If stability problems occur, the overclocking function can be turned off using the following command.

  ```
  sudo bash -c 'echo 0 > /sys/devices/system/cpu/cpufreq/boost'
  ```