How to set CPU governor at system boot

blog.sleeplessbeastie.eu/2015/11/09/how-to-set-cpu-governor-at-boot

by Milosz Galazka on November 9, 2015

Currently, I am playing with recent *Ubuntu Wily Werewolf* on my personal notebook. The first thing I did after system installation was to configure CPU governor and set it at system boot.

Install `cpupower` utility.

```
$ sudo apt-get install linux-tools-common linux-tools-$(uname -r)
```

Verify available CPU governors.

```
$ sudo cpupower -c all frequency-info
```

```
analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency is 3.11 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 1:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 1
  CPUs which need to have their frequency coordinated by software: 1
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency is 3.11 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 2:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 2
  CPUs which need to have their frequency coordinated by software: 2
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
```

```
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency is 3.12 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 3:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 3
  CPUs which need to have their frequency coordinated by software: 3
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "performance" may decide which speed to use
                  within this range.
  current CPU frequency is 3.13 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
```

Set CPU governor to the target state (`powersave` in this example).

```
$ sudo cpupower -c all frequency-set -g powersave

Setting cpu: 0
Setting cpu: 1
Setting cpu: 2
Setting cpu: 3
```

Verify that the change is in effect.

```
$ sudo cpupower -c all frequency-info

analyzing CPU 0:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "powersave" may decide which speed to use
                  within this range.
  current CPU frequency is 3.09 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 1:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 1
  CPUs which need to have their frequency coordinated by software: 1
```

```
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "powersave" may decide which speed to use
                  within this range.
  current CPU frequency is 3.14 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 2:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 2
  CPUs which need to have their frequency coordinated by software: 2
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "powersave" may decide which speed to use
                  within this range.
  current CPU frequency is 3.12 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
analyzing CPU 3:
  driver: intel_pstate
  CPUs which run at the same hardware frequency: 3
  CPUs which need to have their frequency coordinated by software: 3
  maximum transition latency: 0.97 ms.
  hardware limits: 500 MHz - 3.20 GHz
  available cpufreq governors: performance, powersave
  current policy: frequency should be within 500 MHz and 3.20 GHz.
                  The governor "powersave" may decide which speed to use
                  within this range.
  current CPU frequency is 3.11 GHz (asserted by call to hardware).
  boost state support:
    Supported: yes
    Active: yes
```

Create *systemd* service file to set CPU governor at system boot.

```
$ cat << EOF | sudo tee /etc/systemd/system/cpupower.service
[Unit]
Description=CPU powersave

[Service]
Type=oneshot
ExecStart=/usr/bin/cpupower -c all frequency-set -g powersave

[Install]
WantedBy=multi-user.target
EOF
```

Notice that I have used **oneshot** process start-up type as I only want to execute an action without keeping active process.

Reload systemd manager configuration.

```
$ sudo systemctl daemon-reload
```

Enable service at boot time.

```
$ sudo systemctl enable cpupower.service
```

Additional notes

Read systemd.service — Service unit configuration and `cpupower` manual page for further information.

Inspect `/sys/devices/system/cpu/cpu?/cpufreq/` directories if you want to directly use `sysfs` virtual file system provided by the Linux kernel.

Milosz Galazka's Picture

About Milosz Galazka

Milosz is a Linux Foundation Certified Engineer working for a successful Polish company as a system administrator and a long time supporter of Free Software Foundation and Debian operating system.

Post navigation

Previous

How to crack password using Nvidia GPU
November 2, 2015

Next

How to mount encrypted LVM logical volume
November 16, 2015