CPU-frequency scaling on Slackware

Introduction

CPU frequency scaling is carried out by the kernel, there are currently two main ways of thinking about CPU Frequency Scaling, and these are based on the frequency scaling driver which is in use. By default "p-states" implements CPU Frequency Scaling and cannot be compiled out of the Kernel. This has been the case for Intel for quite some time, and recently (as of 2023) "p-states" are also implemented for AMD (but is not the default driver). It is however possible to disable p-states, which will set the controlling driver to "acpi-cpufreq".

"P-states" use the default scaling mechanisms of the CPU and in many cases leave more decisions to the CPU ("hardware managed p-states" aka HPW). There are two frequency scaling "governors" for this mode, "performance" and "powersave". If your CPU does not support HWP it is put into "passive" mode, which you can also do manually. If you change the scaling driver to acpi-cpufreq, you can disable the p-states of the CPU. ACPI is an "industry standard" and related to this topic you can use it to scale your CPU in a different way by using the acpi-cpufreq driver and scaling governors. Until somewhat recently there were 5 scaling governors for acpi-cpufreq, **"performance"**, **"powersave"**, **"userspace"**, **"ondemand"** and **"conservative"**, with "ondemand" often being the default. A 6th governor **"schedutil"** was added and made the default option in the Kernel and Slackware. It might seem complicated, but you only really have to consider the two main modes of CPU Frequency Scaling, which is p-state and acpi-cpufreq. Slackware by default comes with all requirements necessary to use these features.

For older computers with "legacy" scaling implementations and with a new Kernel, these functions have been incorporated into acpi-cpufreq, and alongside AMD it should be your default choice. For very old and new AMD CPU's (zen2 onwards), other options exists and should be considered.

Available governors and the **currently used driver** can be seen by using the cpufreq command

```
cpufreq-info
```

Kernel Configuration

Both Kernels (huge/generic) for Slackware 15 includes the necessary modules for all the CPU Frequency Scaling functions.
If you want to use a newer Kernel series or your own Kernel, the relevant options are found in the section **"Power management and ACPI options" under "CPU Frequency scaling"**.

With AMD you should normally use the acpi-cpufreq driver, but exceptions apply. With Intel you will expose 3 different drivers: **intel_pstate**, **intel_cpufreq** (passive/acpi) and **acpi-cpufreq**. Generally speaking you could say that Intel P-state is performance biased in general and that acpi-cpufreq gives you the maximum control (including performance bias). There are several options you can pass to the Kernel at boot, but if you want to disable p-states and use the acpi-cpufreq driver you pass

intel_pstate=disable to the Kernel at boot time. These two options are the safe options, while other options could prevent a boot with certain CPU's. However, you might want to use another option depending on your cpu.

You can pass these options at boot time to the Kernel command line with the bootloader, like Grub2, you add it after the Kernel line, the standard way of passing Kernel parameters at boot.

/etc/rc.d/rc.cpufreq and /etc/default/cpufreq

The main way to adjust frequency scaling in Slackware is to do so at boot time by making /etc/rc.d/rc.cpufreq executable.

```
chmod +x /etc/rc.d/rc.cpufreq
```

And by selecting the governor to use by editing the file /etc/default/cpufreq.

```
nano /etc/default/cpufreq
```

```
# To select a particular CPU governor option for /etc/rc.d/rc.cpufreq,
# uncomment the line below and edit it to select your choice:
#SCALING_GOVERNOR=ondemand
```

> The necessary setup is done in /etc/rc.d/rc.cpufreq and it reads /etc/default/cpufreq to select the governor. These things do depend on what options are available to it, and those options depend on what choice you make at boot time and which driver is in use. If you use the acpi-cpufreq driver, you can select any of the available scaling governors, if intel_pstate is in use, it will ignore the choice if it cannot be parsed.

After the initial setup of CPU frequency scaling at boot, scaling can also be controlled manually in a variety of ways, as described below.

Handling of CPU frequency scaling with cpufrequtils and /sys

Slackware comes with the `cpufrequtils` package which helps to configure frequency scaling in an appropriate manner.

Checking the CPU frequency settings

You can check the settings on your system with the command

```
cpufreq-info
```

Or if you don't have the package installed:

```
cat /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
cat /proc/cpuinfo
```

where "cpu*" is the CPU you want the info on, for example cpu0.

Setting the CPU frequency settings

The `cpufreq-set` command can be used to set the appropriate governor.

You can set the scaling governor manually if you want, per cpu core.

```
cpufreq-set --cpu 0 --governor conservative
cpufreq-set --cpu 1 --governor conservative
```

The same can be done using sysfs:

```
echo "conservative" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo "conservative" > /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
```

If you have Turbo Boost enabled in the UEFI/BIOS, you can turn boost on (1) or off (0) by running:

```
echo 1 > /sys/devices/system/cpu/cpufreq/boost
```

as long as the interface exists.

Further reading

The manpages for `cpufreq-info` and `cpufreq-set` are useful.

The kernel `Documentation/cpu-freq` directory contains detailed explanations of CPU frequency drivers, governors, and other settings.

A decent explanation of P states and how they are handled in the kernel:
https://plus.google.com/+ArjanvandeVen/posts/dLn9T4ehywL

Sources

- Originally written by Markus Hutmacher
- Updated by metaschima
- Updated by zeebra

* Additional resources and sources: https://www.kernel.org/doc/html/v4.12/admin-guide/pm/intel_pstate.html

* Additional resources and sources: https://www.kernel.org/doc/Documentation/cpu-freq/user-guide.txt

* Additional resources and sources: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

howtos, CPU, frequency, Laptop, cpu frequency scaling