



KooshaYeganehGnuLinuxSoftwares

|<[][]\$#/-\ `/[E-(+_+[-|X|E-# (_+|X||_| _!|X|_|_)|C \$[]]='|'\|//-\V/2[-\$

Tuned

powerful daemon for dynamically auto-tuning Linux server performance

Tuned uses two types of plugins: *monitoring plugins* and *tuning plugins*.

Monitoring plugins are used to get information from a running system.

Currently, the following monitoring plugins are implemented:

Disk

Gets disk load (number of IO operations) per device and measurement interval.

Net

Gets network load (number of transferred packets) per network card and measurement interval.

Load

Gets CPU load per CPU and measurement interval.

The output of the monitoring plugins can be used by tuning plugins for dynamic tuning. Currently implemented dynamic tuning algorithms try to balance the performance and powersave and are therefore disabled in the performance profiles (dynamic tuning for individual plugins can be enabled or disabled in the tuned profiles). Monitoring plugins are automatically

instantiated whenever their metrics are needed by any of the enabled tuning plugins. If two tuning plugins require the same data, only one instance of the monitoring plugin is created and the data is shared.

Each tuning plugin tunes an individual subsystem and takes several parameters that are populated from the tuned profiles. Each subsystem can have multiple devices (for example, multiple CPUs or network cards) that are handled by individual instances of the tuning plugins. Specific settings for individual devices are also supported. The supplied profiles use wildcards to match all devices of individual subsystems (for details on how to change this, refer to Section 3.1.3, “Custom Profiles”), which allows the plugins to tune these subsystems according to the required goal (selected profile) and the only thing that the user needs to do is to select the correct tuned profile.

Cpu

Sets the CPU governor to the value specified by the *governor* parameter and dynamically changes the PM QoS CPU DMA latency according to the CPU load. If the CPU load is lower than the value specified by the *load_threshold*

parameter, the latency is set to the value specified by the *latency_high* parameter, otherwise it is set to value specified by *latency_low*. Also the latency can be forced to a specific value without being dynamically changed further. This can be accomplished by setting the *force_latency* parameter to the required latency value.

eeepc_she

Dynamically sets the FSB speed according to the CPU load; this feature can be found on some netbooks and is also known as the Asus Super Hybrid Engine. If the CPU load is lower or equal to the value specified by the *load_threshold_powersave* parameter, the plugin sets the FSB speed to the value specified by the *she_powersave* parameter (for details about the FSB frequencies and corresponding values, see the kernel documentation, the provided defaults should work for most users). If the CPU load is higher or equal to the value specified by the *load_threshold_normal* parameter, it sets the FSB speed to the value specified by the *she_normal* parameter. Static tuning is not supported and the plugin is transparently disabled if the hardware support for this feature is not detected.

net

Configures wake-on-lan to the values specified by the *wake_on_lan* parameter (it uses same syntax as the *ethtool* utility). It also dynamically changes the interface speed according to the interface utilization.

sysctl

Sets various sysctl settings specified by the plugin parameters. The syntax is name=value, where name is the same as the name provided by the sysctl tool. Use this plugin if you need to change settings that are not covered by other plugins (but prefer specific plugins if the settings are covered by them).

usb

Sets autosuspend timeout of USB devices to the value specified by the *autosuspend* parameter. The value 0 means that autosuspend is disabled.

vm

Enables or disables transparent huge pages depending on the Boolean value of the *transparent_hugepages* parameter.

audio

Sets the autosuspend timeout for audio codecs to the value specified by the *timeout* parameter. Currently *snd_hda_intel* and *snd_ac97_codec* are supported. The value 0 means that the autosuspend is disabled. You can also enforce the controller reset by setting the Boolean parameter *reset_controller* to true.

Disk

Sets the elevator to the value specified by the *elevator* parameter. It also sets ALPM to the value specified by the *alpm* parameter, ASPM to the value

specified by the *aspm* parameter, scheduler quantum to the value specified by the *scheduler_quantum* parameter, disk spindown timeout to the value specified by the *spindown* parameter, disk readahead to the value specified by the *readahead* parameter, and can multiply the current disk readahead value by the constant specified by the *readahead_multiply* parameter. In addition, this plugin dynamically changes the advanced power management and spindown timeout setting for the drive according to the current drive utilization. The dynamic tuning can be controlled by the Boolean parameter *dynamic* and is enabled by default.

Mounts

Enables or disables barriers for mounts according to the Boolean value of the *disable_barriers* parameter.

Script

This plugin can be used for the execution of an external script that is run when the profile is loaded or unloaded. The script is called by one argument which can be start or stop (it depends on whether the script is called during the profile load or unload). The script file name can be specified by the *script* parameter. Note that you need to correctly implement the stop action in your script and revert all setting you changed during the start action, otherwise the roll-back will not work. For your convenience, the functions Bash helper script is installed by default and allows you to import and use various functions defined in it. Note that this functionality is provided mainly for backwards compatibility and it is recommended that you use it as the last resort and prefer other plugins if they cover the required settings.

Sysfs

Sets various sysfs settings specified by the plugin parameters. The syntax is name=value, where name is the sysfs path to use. Use this plugin in case you need to change some settings that are not covered by other plugins (please prefer specific plugins if they cover the required settings).

video

Sets various powersave levels on video cards (currently only the Radeon cards are supported). The powersave level can be specified by using the

radeon_powersave parameter. Supported values are: default, auto, low, mid, high, and dynpm. For details, refer to http://www.x.org/wiki/RadeonFeature#KMS_Power_Management_Options. Note that this plugin is experimental and the parameter may change in the future releases.

Bootloader

Adds parameters to the kernel boot command line. This plugin supports the legacy GRUB 1, GRUB 2, and also GRUB with Extensible Firmware Interface (EFI). Customized non-standard location of the grub2 configuration file can be specified by the *grub2_cfg_file* option. The parameters are added to the current grub configuration and its templates. The machine needs to be rebooted for the kernel parameters to take effect.

USE

Currently the default profile is selected according the following customizable rules:

Throughput-performance

This is pre-selected on Red Hat Enterprise Linux 7 operating systems which act as compute nodes. The goal on such systems is the best throughput performance.

Virtual-guest

This is pre-selected on virtual machines. The goal is best performance. If you are not interested in best performance, you would probably like to change it to the `balanced` or `powersave` profile (see bellow).

Balanced

This is pre-selected in all other cases. The goal is balanced performance and power consumption

```
sudo systemctl start tuned
```

```
sudo systemctl enable tuned
```

To view the available installed profiles, run:

```
tuned-adm list
```

To view the currently activated profile, run:

```
tuned-adm active
```

To select or activate a profile, run:

```
tuned-adm profile profile
```

For example:

tuned-adm profile powersave

As an experimental feature it is possible to select more profiles at once. The tuned application will try to merge them during the load. If there are conflicts the settings from the last specified profile will take precedence. This is done automatically and there is no checking whether the resulting combination of parameters makes sense. If used without thinking, the feature may tune some parameters the opposite way which may be counterproductive. An example of such situation would be setting the disk for the high throughput by using the throughput-performance profile and concurrently setting the disk spindown to the low value by the spindown-disk profile. The following example optimizes the system for run in a virtual machine for the best performance and concurrently tune it for the low power consumption while the low power consumption is the priority:

tuned-adm profile virtual-guest powersave

To let tuned recommend you the best suitable profile for your system without changing any existing profiles and using the same logic as used during the

installation, run the following command:

```
tunedadm recommend
```

Custom Profiles

Distribution-specific profiles are stored in the `/usr/lib/tuned/` directory. Each profile has its own directory. The profile consists of the main configuration file called `tuned.conf`, and optionally other files, for example helper scripts.

If you need to customize a profile, copy the profile directory into the `/etc/tuned/` directory, which is used for custom profiles. If there are two profiles of the same name, the profile included in `/etc/tuned/` is used.

You can also create your own profile in the `/etc/tuned/` directory to use a profile included in `/usr/lib/tuned/` with only certain parameters adjusted or overridden.

The `tuned.conf` file contains several sections. There is one `[main]` section. The other sections are configurations for plugins instances. All sections are optional including the `[main]` section. Lines starting with the hash sign (`#`) are comments.

The `[main]` section has the following option:

`include=profile`

The specified profile will be included, e.g. `include=powersave` will include the powersave profile.

Sections describing plugins instances are formatted in the following way:

```
[NAME]
type=TYPE
devices=DEVICES
```

NAME is the name of the plugin instance as it is used in the logs. It can be an arbitrary string. *TYPE* is the type of the tuning plugin. For a list and descriptions of the tuning plugins refer to Section 3.1.1, “Plug-ins”. *DEVICES* is the list of devices this plugin instance will handle. The devices line can contain a list, a wildcard (*), and negation (!). You can also combine rules. If there is no devices line all devices present or later attached on the system of the TYPE will be handled by the plugin instance. This is same as using `devices=*`. If no instance of the plugin is specified, the plugin will not be enabled. If the plugin supports more options, they can be also specified in the plugin section. If the option is not specified, the default value will be used (if not previously specified in the included

plugin). For the list of plugin options refer to Section 3.1.1, “Plug-ins”).

The following example will match everything starting with sd, such as sda or sdb, and does not disable barriers on them:

```
[data_disk]
type=disk
devices=sd*
disable_barriers=false
```

The following example will match everything except sda1 and sda2:

```
[data_disk]
type=disk
devices=!sda1, !sda2
disable_barriers=false
```

In cases where you do not need custom names for the plugin instance and there is only one definition of the instance in your configuration file, Tuned supports the following short syntax:

```
[TYPE]
```

```
devices=DEVICES
```

in this case, it is possible to omit the type line. The instance will then be referred to with a name, same as the type. The previous example could be then rewritten into:

```
[disk]
devices=sdb*
disable_barriers=false
```

If the same section is specified more than once using the include option, then the settings are merged. If they cannot be merged due to a conflict, the last conflicting definition overrides the previous settings in conflict. Sometimes, you do not know what was previously defined. In such cases, you can use the replace boolean option and set it to true. This will cause all the previous definitions with the same name to be overwritten and the merge will not happen.

You can also disable the plugin by specifying the `enabled=false` option. This has the same effect as if the instance was never defined. Disabling the plugin can be useful if you are redefining the previous definition from the include option and do not want the plugin to be active in your custom profile.

The following is an example of a custom profile that is based on the balanced profile and extends it the way that ALPM for all devices is set to the maximal powersaving.

```
[main]
include=balanced

[disk]
alpm=min_power
```

The following is an example of a custom profile that adds `isolcpus=2` to the kernel boot command line:

```
[bootloader]
cmdline=isolcpus=2
```

The machine needs to be rebooted after the profile is applied for the changes to take effect.

Tuned-Adm

A detailed analysis of a system can be very time-consuming. Red Hat Enterprise Linux 7 includes a number of predefined profiles for typical use cases that you can easily activate with the `tuned-adm` utility. You can also create, modify, and delete profiles.

To list all available profiles and identify the current active profile, run:

```
tuned-adm list
```

To only display the currently active profile, run:

```
tuned-adm active
```

To switch to one of the available profiles, run:

```
tuned-adm profile profile_name
```

for example:

```
tuned-adm profile latency-performance
```

To disable all tuning:

```
tuned-adm off
```

Balanced

The default power-saving profile. It is intended to be a compromise between performance and power consumption. It tries to use auto-scaling and auto-tuning whenever possible. It has good results for most loads. The only drawback is the increased latency. In the current tuned release it enables the CPU, disk, audio and video plugins and activates the conservative governor. The *radeon_powersave* is set to auto

Powersave

A profile for maximum power saving performance. It can throttle the performance in order to minimize the actual power consumption. In the current tuned release it enables USB autosuspend, WiFi power saving and ALPM power savings for SATA host adapters. It also schedules multi-core power savings for systems with a low wakeup rate and activates the ondemand governor. It enables AC97 audio power saving or, depending on your system, HDA-Intel power savings with a 10 seconds timeout. In case your system contains supported Radeon graphics card with enabled KMS it configures it to automatic power saving. On Asus Eee PCs a dynamic Super Hybrid Engine is enabled.

Throughput-performance

A server profile optimized for high throughput. It disables power savings mechanisms and enables sysctl settings that improve the throughput performance of the disk, network IO and switched to the deadline scheduler. CPU governor is set to performance

Latency-performance

A server profile optimized for low latency. It disables power savings mechanisms and enables sysctl settings that improve the latency. CPU governor is set to performance and the CPU is locked to the low C states (by PM QoS).

Network-latency

A profile for low latency network tuning. It is based on the latency-performance profile. It additionally disables transparent hugepages, NUMA balancing and tunes several other network related sysctl parameters.

Network-throughput

Profile for throughput network tuning. It is based on the throughput-performance profile. It additionally increases kernel network buffers.

virtual-guest

A profile designed for Red Hat Enterprise Linux 7 virtual machines as well as VMware guests based on the enterprise-storage profile that, among other tasks, decreases virtual memory swappiness and increases disk readahead values. It does not disable disk barriers.

virtual-host

A profile designed for virtual hosts based on the enterprise-storage profile that, among other tasks, decreases virtual memory swappiness, increases disk readahead values and enables a more aggressive value of dirty pages.

oracle

A profile optimized for Oracle databases loads based on throughput-performance profile. It additionally disables transparent huge

pages and modifies some other performance related kernel parameters.

This profile is provided by `tuned-profiles-oracle` package. It is available in Red Hat Enterprise Linux 6.8 and later.

desktop

A profile optimized for desktops, based on the balanced profile. It additionally enables scheduler autogroups for better response of interactive applications.

cpu-partitioning

The `cpu-partitioning` profile partitions the system CPUs into isolated and housekeeping CPUs. To reduce jitter and interruptions on an isolated CPU, the profile clears the isolated CPU from user-space processes, movable kernel threads, interrupt handlers, and kernel timers.

A housekeeping CPU can run all services, shell processes, and kernel threads.

You can configure the `cpu-partitioning` profile in the `/etc/tuned/cpu-partitioning-variables.conf` file. The configuration options are

`isolated_cores=cpu-list`

Lists CPUs to isolate. The list of isolated CPUs is comma-separated or the user can specify the range. You can specify a range using a dash, such as 3-5. This option is mandatory. Any CPU missing from this list is automatically considered a housekeeping CPU.

```
no_balance_cores=cpu-list
```

Lists CPUs which are not considered by the kernel during system wide process load-balancing. This option is optional. This is usually the same list as `isolated_cores`.

Additional predefined profiles can be installed with the `tuned-profiles-compat` package available in the Optional channel. These profiles are intended for backward compatibility and are no longer developed. The generalized profiles from the base package will mostly perform the same or better. If you do not have specific reason for using them, please prefer the above mentioned profiles from the base package. The compat profiles are following:

```
default
```

This has the lowest impact on power saving of the available profiles and only enables CPU and disk plugins of tuned.

desktop-powersave

A power-saving profile directed at desktop systems. Enables ALPM power saving for SATA host adapters as well as the CPU, Ethernet, and disk plugins of tuned.

laptop-ac-powersave

A medium-impact power-saving profile directed at laptops running on AC. Enables ALPM powersaving for SATA host adapters, Wi-Fi power saving, as well as the CPU, Ethernet, and disk plugins of tuned.

laptop-battery-powersave

A high-impact power-saving profile directed at laptops running on battery. In the current tuned implementation it is an alias for the powersave profile.

Spindown-disk

A power-saving profile for machines with classic HDDs to maximize spindown time. It disables the tuned power savings mechanism, disables USB autosuspend, disables Bluetooth, enables Wi-Fi power saving,

disables logs syncing, increases disk write-back time, and lowers disk swappiness. All partitions are remounted with the noatime option.

Enterprise-storage

A server profile directed at enterprise-class storage, maximizing I/O throughput. It activates the same settings as the throughput-performance profile, multiplies readahead settings, and disables barriers on non-root and non-boot partitions.

To enable the tuned profiles for Red Hat Enterprise Linux Atomic Host, install the `tuned-profiles-atomic` package. Run, as root, the following command:

```
yum install tuned-profiles-atomic
```

The two tuned profiles for Red Hat Enterprise Linux Atomic Host are:

A profile optimized for Red Hat Enterprise Linux Atomic Host, when used as a host system on a bare-metal server, using the throughput-performance profile. It additionally increases SELinux AVC

cache, PID limit, and tunes netfilter connections tracking.

```
atomic-guest
```

A profile optimized for Red Hat Enterprise Linux Atomic Host, when used as a guest system based on the virtual-guest profile. It additionally increases SELinux AVC cache, PID limit, and tunes netfilter connections tracking.

To enable the realtime profile, install the tuned-profiles-realtime package. Run, as root, the following command:

```
yum install tuned-profiles-realtime
```

To enable the realtime-virtual-host and realtime-virtual-guest profiles, install the tuned-profiles-nfv package. Run, as root, the following command:

```
yum install tuned-profiles-nfv
```

Power2tuned

The `powertop2tuned` utility is a tool that allows you to create custom tuned profiles from the PowerTOP suggestions.

To install the `powertop2tuned` application, run the following command as root:

```
yum install tuned-utils
```

To create a custom profile, run the following command as root:

```
powertop2tuned new_profile_name
```

By default it creates the profile in the `/etc/tuned` directory and it bases it on the currently selected tuned profile. For safety reasons all PowerTOP tunings are initially disabled in the new profile. To enable them uncomment the tunings of your interest in the `/etc/tuned/profile/tuned.conf`. You can use the `--enable` or `-e` option that will generate the new profile with most of the tunings suggested by PowerTOP enabled. Some dangerous tunings like the USB autosuspend will still be disabled. If you really need them you will have to uncomment them manually. By default, the new profile is not activated. To activate it run the following command:

```
tuned-adm profile new_profile_name
```

For a complete list of the options `powertop2tuned` supports, type in the following command:

