

Oracle Linux 10

Optimize Performance and Power Consumption With TuneD and PowerTOP



G24424-02
October 2025



Oracle Linux 10 Optimize Performance and Power Consumption With TuneD and PowerTOP,

G24424-02

Copyright © 2025, Oracle and/or its affiliates.

Documentation License

The content in this document is licensed under the [Creative Commons Attribution–Share Alike 4.0](https://creativecommons.org/licenses/by-sa/4.0/) (CC-BY-SA) license. In accordance with CC-BY-SA, if you distribute this content or an adaptation of it, you must provide attribution to Oracle and retain the original copyright notices.

Contents

Preface

1 About TuneD

2 Installing TuneD

3 Reviewing TuneD Global Settings

Selecting Static or Dynamic Tuning	1
Selecting Daemon or No-Daemon Mode	2

4 Reviewing TuneD Profiles

5 TuneD Command Reference

Listing TuneD Profiles	1
Querying TuneD for a Recommended Profile	2
Selecting an Active TuneD Profile	2
Unloading the Current Active Profile	3

6 Creating a Custom TuneD Profile

7 About PowerTOP

8 Installing PowerTOP

9 PowerTOP Command Reference

Using PowerTOP in Interactive Mode	1
------------------------------------	---

Overview	2
Idle Stats	3
Frequency Stats	3
Tunables	4
Wakeup	4
Using PowerTOP to Generate CSV Reports	5

10 Installing powertop2tuned

11 Using powertop2tuned to Create a TuneD Profile

Preface

[Oracle Linux 10: Optimize Performance and Power Consumption With TuneD and PowerTOP](#) describes how to install and use the TuneD and PowerTOP tools to monitor performance and reduce power consumption.

1

About TuneD

TuneD can be used to optimize the system's performance and power consumption for different use cases, and consists of the following main components:

TuneD Profiles

A profile is a set of predefined optimizations for system components, such as CPUs, and disk devices. Some profiles, such as `latency-performance` and `network-latency`, are configured to prioritize performance, whilst others, such as `powersave`, are configured to use less power. Select the profile to use on a server according to the workload and the required power-performance balance of those tasks. Custom TuneD profiles with extra optimizations for a specific system can also be used.

`tune-adm` command line utility

Use the `tune-adm` command to manage profiles on the server, list available profiles, and select the current system profile.

For more information, see the `tuned-adm(8)` and `tuned(8)` manual pages.

2

Installing TuneD

Install the `tuned` package on Oracle Linux 10.

To install `tuned` package and start the `tuned` service, follow these steps:

1. Install the `tuned` package by running the following command:

```
sudo dnf install tuned
```

2. Enable and start the `tuned` service by running the following command:

```
sudo systemctl enable --now tuned
```

The `tuned` package is installed, and the service is enabled and running. To check the profiles installed and set up on the system, follow the steps in [Listing TuneD Profiles](#) .

3

Reviewing TuneD Global Settings

This following section provides an overview of global settings that configure the tuned service.

TuneD Global Configuration File

TuneD can be configured globally by editing the `/etc/tuned/tuned-main.conf` file. The following sample provides an example configuration:

```
# Global tuned configuration file.

# Whether to use daemon. Without daemon it just applies tuning. It is
# not recommended, because many functions don't work without daemon,
# e.g. there will be no D-Bus, no rollback of settings, no hotplug,
# no dynamic tuning, ...
daemon = 1

# Dynamically tune devices, if disabled only static tuning will be used.
dynamic_tuning = 0

# How long to sleep before checking for events (in seconds)
# higher number means lower overhead but longer response time.
sleep_interval = 1

# Update interval for dynamic tunings (in seconds).
# It must be multiply of the sleep_interval.
update_interval = 10

# Recommend functionality, if disabled "recommend" command will be not
# available in CLI, daemon will not parse recommend.conf but will return
# one hardcoded profile (by default "balanced").
recommend_command = 1
...
```

The following are sample cases for which you can configure parameters in `/etc/tuned/tuned-main.conf`:

Selecting Static or Dynamic Tuning

With the `dynamic_tuning` parameter, you can select whether system tuning is static or dynamic.

By default, static tuning is operative in the system. Static tuning applies settings that have been predefined for `sysctl` and `sysfs` commands, or those that are set for configuration tools at the moment these tools are activated. Thereafter, no more tuning is performed.

Dynamic tuning instead is performed continuously. TuneD monitors the system at intervals throughout the system's up time. Based on the information gathered at a specific interval, TuneD optimizes the system. The interval at which TuneD monitors and optimizes components is configured by the value of the `uptime_interval`, which by default is set to 10 seconds.

Selecting Daemon or No-Daemon Mode

With the `daemon` parameter, you can set the mode for system tuning.

By default, the functionalities of TuneD are active if the daemon is running. If TuneD is switched to run with the daemon disabled, then TuneD applies the profile settings and then exits. This mode isn't recommended because some TuneD functionalities, such as compatibility with Dbus, hotplugging, and rollback of settings, are absent if the daemon is disabled.

4

Reviewing TuneD Profiles

The following sections provide an overview of TuneD profiles and how they're configured in their respective `tuned.conf` configuration files.

Predefined Profiles

The following list provides a summary of profiles that are commonly provided for use with TuneD:

balanced

The `balanced` profile provides a balance between performance and power consumption. The profile uses automatic scaling and automatic tuning when possible. A possible drawback is increased latency.

powersave

The `powersave` profile provides maximum power saving performance. The profile can minimize actual power consumption by throttling performance.

Note

In some instances, the `balanced` profile is more efficient than the `powersave` profile and therefore, a better choice. For example, consider a workload that includes idle periods between resource-intensive tasks. A system running in a higher performance mode uses more energy to complete the tasks, but it completes them more quickly, and returns its components to idle power saving states for longer. In such situations, the `balanced` profile can be a better option.

throughput-performance

The `throughput-performance` profile disables power-saving mechanisms and enables `sysctl` settings to improve the throughput performance of the disk and network IO.

latency-performance

The `latency-performance` profile is optimized for low latency. The profile disables power-saving mechanisms and enables `sysctl` settings to improve latency.

network-latency

The `network-latency` profile provides low latency network tuning. The `network-latency` profile inherits from the `latency-performance` profile, and in addition, includes several network-related `sysctl` settings. The profile also disables transparent huge pages and automatic NUMA balancing.

network-throughput

The `network-throughput` profile is used for optimizing throughput network tuning. The `network-throughput` is based on the `throughput-performance` profile, and in addition, includes `sysctl` settings to increase kernel network buffer sizes.

virtual-guest

The `virtual-guest` profile is designed for virtual guests and is based on the `throughput-performance` profile. In addition, this profile decreases virtual memory swappiness and increases the `dirty_ratio` setting.

virtual-host

The `virtual-host` profile is designed for virtual hosts and is based on the `throughput-performance` profile. In addition, this profile sets a more aggressive value for dirty pages writeback.

desktop

The `desktop` profile is optimized for desktop environments and is based on the `balanced` profile. In addition, this profile sets scheduler `autogroups` for better response of interactive applications.

Note

Different types of instances of Oracle Linux can have different TuneD profiles installed by default. For example, on an Oracle Linux instance that's running in Oracle Cloud Infrastructure, the list would include extra profiles, such as the following:

oci-busy-polling

The `oci-busy-polling` profile enables Busy Polling conditionally in OCI.

oci-cpu-power

The `oci-cpu-power` profile sets processor power management parameters in OCI.

oci-nic

The `oci-nic` profile increases combined channels to 16 on NICs with `bnxt_en` driver on BM shapes in OCI.

oci-rps-xps

The `oci-rps-xps` profile enables RPS/XPS conditionally in OCI

To get a complete list of `tuned-profile` packages that are available for installation, run the following command:

```
sudo dnf list tuned-profiles*
```

For more information about profiles, see the manual page for `tuned-profiles(7)`.

Custom Profiles

The predefined profiles included with TuneD cover a range of use cases. TuneD can also run custom profiles in cases where further optimization is required. One way of creating a custom profile is to copy an existing profile and then customizing that profile as required. For more information about how to create and activate custom profiles, see [Creating a Custom TuneD Profile](#).

TuneD Profile Configuration Files

Profiles are automatically stored in the following directories:

- `/usr/lib/tuned/profile_name` contains predefined profiles.

- `/etc/tuned/profile_name` contains custom profiles.

Each profile's rules are contained in a corresponding `tuned.conf` file. For example, for the latency-performance profile, the rules are defined in `/usr/lib/tuned/latency-performance/tuned.conf`, while the rules for the powersave profile are defined in `/usr/lib/tuned/powersave/tuned.conf`.

The following extract shows an example configuration file for the latency-performance profile:

```
[main]
summary=Optimize for deterministic performance at the cost of increased power
consumption

[cpu]
force_latency=cstate.id_no_zero:1|3
governor=performance
energy_perf_bias=performance
min_perf_pct=100

[acpi]
platform_profile=performance

[sysctl]
# If a workload mostly uses anonymous memory and it hits this limit, the
entire
# working set is buffered for I/O, and any more write buffering would require
# swapping, so it's time to throttle writes until I/O can catch up. Workloads
# that mostly use file mappings may be able to use even higher values.
#
# The generator of dirty data starts writeback at this percentage (system
default
# is 20%)
vm.dirty_ratio=10

# Start background writeback (via writeback threads) at this percentage
(system
# default is 10%)
vm.dirty_background_ratio=3

# The swappiness parameter controls the tendency of the kernel to move
# processes out of physical memory and onto the swap disk.
# 0 tells the kernel to avoid swapping processes out of physical memory
# for as long as possible
# 100 tells the kernel to aggressively swap processes out of physical memory
# and move them to swap cache
vm.swappiness=10

[scheduler]
runtime=0
...
```

The previous example shows how profile configuration files often include sections such as `[cpu]` and `[sysctl]`, where different types of component and performance configurations are defined.

Profiles can inherit settings from other profiles. For example, the following sample extract shows how the configuration file for the `network-performance` profile uses the `include=` option to inherit settings from the `latency-performance` profile:

```
[main]
summary=Optimize for deterministic performance at the cost of increased power
consumption, focused on low latency network performance
include=latency-performance

[vm]
transparent_hugepages=never

[sysctl]
net.core.busy_read=50
net.core.busy_poll=50
net.ipv4.tcp_fastopen=3
kernel.numa_balancing=0
kernel.hung_task_timeout_secs = 600
kernel.nmi_watchdog = 0
vm.stat_interval = 10
kernel.timer_migration = 0

[bootloader]
cmdline_network_latency=skew_tick=1 tsc=reliable
rcupdate.rcu_normal_after_boot=1
...
```

For more information about configuring TuneD profiles, see the `tuned.conf(5)` manual page.

5

Tuned Command Reference

The following table provides an overview of commands provided by the `tuned-adm` commands utility.

Action	Command	Description
List the active profile.	<code>tuned-adm active</code>	Outputs the active Tuned profile. Note: There can be more than one profile active on a server.
List all available profiles.	<code>tuned-adm list</code>	Outputs a list of all available profiles, followed by the profile that's active.
Query Tuned for recommended profile.	<code>tuned-adm recommend</code>	Outputs the profile recommended for the system. By default, Tuned recommends profiles targeted to the best performance.
Set active profile.	<code>tuned-adm profile profile1 [profile2 profile3 ...]</code>	Switches to the specified profile. If more than one profile is specified, the profiles are merged and the resulting profile is applied. In case of conflicting settings, the setting from the last profile is used.
Verify that the system has been optimized to match the settings as defined in the active profile.	<code>tuned-adm verify</code>	Outputs information on whether system settings match current profile.
Unload the Tuned tunings.	<code>tuned-adm off</code>	Deactivates Tuned tunings and unloads active profiles.

Listing Tuned Profiles

Use the `list` and `active` commands provided by `tuned-adm` to review the Tuned profiles on a server.

To list all the profiles that are available for the system, run the following command:

```
tuned-adm list
```

The command output displays a list of all the available profiles. By default, the list is followed by the current active profile, as shown in the following sample output:

```
Available profiles:
- accelerator-performance      - Throughput performance based tuning with
disabled higher latency STOP states
- balanced                      - General non-specialized tuned profile
- desktop                       - Optimize for the desktop use-case
...
```

```

- latency-performance          - Optimize for deterministic performance at the
cost of increased power consumption
- network-latency             - Optimize for deterministic performance at the
cost of increased power consumption, focused on low latency network
performance
- network-throughput          - Optimize for streaming network throughput,
generally only necessary on older CPUs or 40G+ networks
...
- throughput-performance      - Broadly applicable tuning that provides
excellent performance across a variety of common server workloads
- virtual-guest                - Optimize for running inside a virtual guest
- virtual-host                 - Optimize for running KVM guests
Current active profile: virtual-guest

```

The list of profiles that are displayed varies depending on whether you're running Oracle Linux on self-hosted physical infrastructure or as an instance in Oracle Cloud Infrastructure.

To show only the current active profile, use the `active` command as follows:

```
tuned-adm active
```

Querying TuneD for a Recommended Profile

Use the `recommend` command provided by `tuned-adm` to see the profile recommended by TuneD.

If the `recommend` parameter is enabled in the `/etc/tuned/tuned-main.conf` file, TuneD can recommend an optimization profile for the system.

To see the profile that TuneD recommends, run the following command:

```
tuned-adm recommend
```

```
balanced
```

```
balanced
```

Selecting an Active TuneD Profile

Use the `profile` command provided by `tuned-adm` to select the active profile.

When `tuned` is enabled on a system or an instance, a default TuneD profile becomes automatically active. However, you can select a different profile and activate it so that system optimization is performed based on that profile. Use the following syntax:

```
sudo tuned-adm profile profile1 [profile2 profile3 ...]
```

The following is a suggested method of changing an active profile:

1. Optionally, check the current active profile by running the following command:

```
tuned-adm active
```

The command outputs the current active profile, balanced in this example:

```
Current active profile: balanced
```

2. Use the profile command to activate a different profile. For example, to select powersave as the active profile, run the following command:

```
sudo tuned-adm profile powersave
```

3. Confirm the active profile has changed by running the following command:

```
tuned-adm active
```

The command outputs the profile selected in the previous step, powersave in this example:

```
Current active profile: powersave
```

4. Verify that the system has been optimized to match the settings as defined in the active profile you have set by running the following command:

```
tuned-adm verify
```

The command output states whether the verification has succeeded, as shown in the following sample output:

```
Verification succeeded, current system settings match the preset profile.  
See TuneD log file ('/var/log/tuned/tuned.log') for details.
```

The active profile has been set and the system settings have been verified as matching the newly set profile.

Unloading the Current Active Profile

Use the `off` command provided by `tuned-adm` to unload active profiles and deactivate TuneD tunings.

To temporarily deactivate tunings without stopping the `tuned.service`, run the following steps:

1. To check the current active profile, run the following command:

```
tuned-adm active
```

The output shows the active profile, for example:

```
Current active profile: balanced
```

2. Unload the active profile and deactivate TuneD tunings by running the following command:

```
sudo tuned-adm off
```

3. Verify that no profile is now active by running the following command:

```
tuned-adm active
```

The command output shows that no profile is active:

```
No current active profile.
```

The active profile has been unloaded and TuneD tunings have been deactivated. To set an active profile again follow the steps outlined in [Selecting an Active TuneD Profile](#).

6

Creating a Custom TuneD Profile

Create a custom TuneD profile with optimization rules customized for a specific system.

TuneD stores predefined profiles in the `/usr/lib/tuned` directory. For custom profiles, TuneD provides the `/etc/tuned` directory. If two profiles share the same name, the profile under `/etc/tuned` takes precedence.

The following steps show one way of creating a custom TuneD profile:

1. Copy a directory for a predefined profile, such as `virtual-guest`, to the directory location for custom profiles.

```
sudo cp -R /lib/tuned/virtual-guest/ /etc/tuned
```

2. Rename the directory that you copied. For example, to create a profile called `mycustomprofile`, run the following command:

```
sudo mv /etc/tuned/virtual-guest /etc/tuned/mycustomprofile
```

3. Customize the `tune.conf` file in the custom directory. For example, for the `mycustomprofile` directory, customize the `/etc/tuned/mycustomprofile/tuned.conf` file.

A new custom profile is created in the `/etc/tuned` directory and is ready to be activated.

7

About PowerTOP

PowerTOP can be used to analyze CPU and power usage on a system, and provide suggestions for power management improvements. The statistics that PowerTOP provides include CPU usage and wakeup calls for each process, kernel worker, and device in the system.

PowerTOP can be started in interactive mode by running `sudo powertop` without passing any options to the command. In interactive mode, PowerTOP displays the data in real time in a tabbed interface and lets you make nonpersistent changes to various power settings for testing purposes. The tabbed interface contains a **Tunables** tab that lists recommendations on power-tuning settings.

Use the `powertop` command with the `--csv` and `--html` options to generate CSV and HTML reports for reporting and data analysis.

For more information, see the `powertop(8)` manual page.

About `powertop2tuned`

The `powertop2tuned` utility can be used to create a customized Tuned profile that combines optimizations from the current active profile with recommended optimizations from PowerTOP. By default, `powertop2tuned` creates the new profile configuration file with the PowerTOP recommendations commented out so that you can review and decide which ones to include in the new profile.

For more information, run the `powertop2tuned --help` command.

8

Installing PowerTOP

Install the `powertop` package on Oracle Linux 10 so that you can use the PowerTOP program to analyze power and CPU usage on the system.

To install the `powertop` package, run the following command:

```
sudo dnf install powertop
```

The `powertop` package is installed and you can use the PowerTOP program.

9

PowerTOP Command Reference

The following table provides an overview of the `powertop` command and the options that can be used with it.

Action	Command	Description
Run PowerTOP in interactive mode.	<code>sudo powertop</code>	PowerTOP is started in interactive mode with a tabbed interface. Tabs include Overview , Idle stats , Frequency stats , and Tunables .
Generate an HTML report.	<code>sudo powertop -- html=<i>filename</i></code>	A HTML report is created in the current directory. If a file name is not specified, then the default name <code>powertop.html</code> is used.
Generate a CSV report.	<code>sudo powertop -- csv=<i>filename</i></code>	A CSV report is created in the current directory. If a file name is not specified, then the default name <code>powertop.csv</code> is used.
Generate HTML reports by running a specified number of tests (<i>iterations</i>) at specified time intervals (<i>seconds</i>)	<code>sudo powertop -- html=<i>filename</i> -- iteration=<i>iterations</i> -- time=<i>seconds</i></code>	A HTML report is created for each iteration. Each HTML file is named using the name format <code>filename-yyyyMMdd-hhmmss.html</code> , for example: <code>mypowertop-20250129-132713.html</code> .
Generate CSV reports by running a specified number of tests (<i>iterations</i>) at specified time intervals (<i>seconds</i>)	<code>sudo powertop -- csv=<i>filename</i> -- iteration=<i>iterations</i> -- time=<i>seconds</i></code>	A CSV report is created for each iteration. Each CSV file is named using the name format <code>filename-yyyyMMdd-hhmmss.csv</code> , for example: <code>mypowertop-20250129-133102.csv</code> .

Using PowerTOP in Interactive Mode

Use PowerTOP in interactive mode by running the `powertop` command without any options.

To start PowerTOP in interactive mode, run the following command:

```
sudo powertop
```

PowerTOP starts in interactive mode with a tabbed interface, with the Overview tab active, as illustrated in the following sample output:

```

Overview  Idle stats  Frequency stats  Device stats  Tunables
WakeUp

Summary: 1937.4 wakeups/second, 0.0 GPU ops/seconds, 0.0 VFS ops/sec and
18.2% CPU use

          Usage      Events/s  Category  Description
          119.9 ms/s  1088.4   Process   [PID 52069] /usr/libexec/
qemu-kvm...
          52.2 ms/s   663.9    Process   [PID 52068] /usr/
libexec/qemu-kvm -name ...
          1.2 ms/s    60.4     Process   [PID 1793] /usr/
libexec/platform-pyth...
          3.5 ms/s    24.7     Timer     apic_timer_fn
          80.0 µs/s   18.8     Process   [PID 962] [xfsaild/
dm-0]
          60.1 µs/s   18.8     Process   [PID 1459] [xfsaild/
dm-1]
          34.4 µs/s   10.9     Process   [PID 15] [rcu_sched]
          66.8 µs/s   9.9      kWork    kvmclock_update_fn
          670.7 µs/s  7.9      Timer     tick_sched_timer
...
          2.0 pkts/s                Device    nic:vnet0
          1.0 pkts/s                Device    nic:virbr0
          6.8 pkts/s                Device    Network interface:
enp0s5 (virtio_net)
...

```

By default, the data is refreshed every 20 seconds. To change the refresh rate, press the letter `s` on the keyboard, enter a new integer value, and then press `Enter`.

You can navigate between the tabs by using the `Tab` and `Shift+Tab` keys, and scroll vertically and horizontally by using the `Arrow` keys.

The following list summarizes the purpose of each tab in the interactive mode interface.

Overview

The **Overview** lists processes and components in descending order based on CPU and resource usage. The **Overview** tab contains the following columns:

- **Usage**
Shows the resource usage of each process or component in the system.
Process usage is shown in CPU time allocated per second (for example 119.9 ms/s) whilst network interface usage is indicated by the rate at which packets are processed (for example 2.0 pkts/s).
- **Events/s**
The number of wakeup events per second. The higher the number of wakeups, the greater the power consumption.
- **Category**

Describes the type of component. For example, a process, timer, or a device.

- **Description**

Describes the component. For example, a network interface might be described as follows:
Network interface: enp0s5 (virtio_net).

Idle Stats

The **Idle stats** tab displays the time the CPU components spend in different C-States.

A C-State of C0 corresponds to an active component, whilst all other C-States, C1, C2...Cn, correspond to idle sleep states (the higher the C-number, the deeper the sleep mode). The C-States available depend upon the processors being used in the system.

The following output provides an example of the type of information displayed:

Pkg(HW)	Core(HW)	CPU(OS) 0	CPU(OS) 64
	C0 active	0.0%	0.0%
0.0% 0.0 ms	POLL	0.0%	0.0 ms
	C1	0.0%	0.0 ms
0.0% 0.0 ms	C1E	0.0%	0.5 ms
C2 (pc2) 52.0%			
0.0% 0.0 ms			
C3 (pc3) 0.0%	C3 (cc3) 0.0%		
C6 (pc6) 0.0%	C6 (cc6) 92.4%	C6 99.9%	59.4 ms 100.0%
134.7 ms			
C7 (pc7) 0.0%	C7 (cc7) 0.0%		
...			

Frequency Stats

The **Frequency stats** tab displays the frequency statistics for the processor components. Lower frequency statistics can often reflect lower power consumption. The following output provides an example of the type of information displayed:

Package	Core	CPU 0	CPU 64
		Average 3.2 GHz	1.7 GHz
Idle	Idle	Idle	
800 MHz	800 MHz	800 MHz	
3.40 GHz	3.40 GHz	3.40 GHz	
	Core	CPU 1	CPU 65
		Average 1.9 GHz	2.0 GHz
	Idle	Idle	
	800 MHz	800 MHz	
	3.40 GHz	3.40 GHz	
...			

Tunables

The **Tunables** tab lists power-tuning settings with a status value of **Good** or **Bad**. The following output provides an example of the type of information that's displayed:

```
PowerTOP 2.15      Overview  Idle stats  Frequency stats  Device stats
Tunables  WakeUp

>> Bad           Enable SATA link power management for
host1
  Bad           Enable SATA link power management for host2
  Bad           Enable SATA link power management for host3
  Bad           VM writeback timeout
  Bad           Enable SATA link power management for host0
...
  Good          Autosuspend for USB device UHCI Host Controller [usb2]
  Good          Autosuspend for USB device UHCI Host Controller [usb4]
  Good          Autosuspend for USB device EHCI Host Controller [usb1]
...
```

To switch between **Bad** and **Good**, move the cursor to the setting, using the `Arrow` keys, and hit the `Enter` key.

Note

Changes made in the **Tunables** tab are only intended for testing purposes, so they are not preserved between reboots.

Wakeup

The **WakeUp** tab lists devices with their wakeup settings (**Enabled** or **Disabled**). The following output provides an example of the type of information displayed:

```
>> Disabled      Wake status for USB device
usb3
  Disabled       Wake status for USB device usb1
  Disabled       Wake status for USB device 1-1
  Disabled       Wake status for USB device usb4
  Disabled       Wake status for USB device usb2
```

To switch between **Disabled** and **Enabled**, move the cursor to the setting using the `Arrow` keys and hit the `Enter` key.

Note

Changes made in the **WakeUp** tab are only intended for testing purposes, so they are not preserved between reboots.

Using PowerTOP to Generate CSV Reports

Use PowerTOP to generate CSV reports for reporting and data analysis.

To generate a CSV report, run the `powertop` command with the `--csv` option. For example, to generate a CSV report called `my_csv_report.csv` in the current directory, run the following command:

```
sudo powertop --csv=my_csv_report.csv
```

If you don't specify a file name, then the report is named `powertop.csv` by default.

You can also use the `--iteration` and `--time` options to generate a particular number of reports at a specified interval. For example, to generate three new CSV reports at a 60 second interval, run the following command:

```
sudo powertop --iteration=3 --time=60 --csv=my_csv_report.csv
```

The output is similar to the following:

```
...
Loaded 0 prior measurements
RAPL device for cpu 0
RAPL device for cpu 0
...
Preparing to take measurements
Taking 3 measurement(s) for a duration of 60 second(s) each.
PowerTOP outputting using base filename my_csv_report-20250117-135852.csv
PowerTOP outputting using base filename my_csv_report-20250117-135952.csv
PowerTOP outputting using base filename my_csv_report-20250117-140052.csv
...
```

10

Installing powertop2tuned

Install `powertop2tuned` on Oracle Linux 10.

The `powertop2tuned` utility is part of the `tuned-utils` package. The `powertop2tuned` utility can be used to create a customized TuneD profile configuration file that combines optimizations from the current active profile with recommended settings returned by PowerTOP.

To install the `tuned-utils` package, run the following command:

```
sudo dnf install tuned-utils
```

The `tuned-utils` package is installed and the `powertop2tuned` utility is ready to use.

11

Using powertop2tuned to Create a TuneD Profile

Use `powertop2tuned` to create a TuneD profile that includes the current active TuneD profile with the addition of PowerTOP recommendations.

Perform the following steps:

1. Run the `powertop2tuned` command with the name of the profile you're creating. For example, to create a profile called `mypowertuneprofile` run the following command:

```
sudo powertop2tuned mypowertuneprofile
```

The command outputs a message as it starts PowerTOP to generate the recommendations, as shown in the following sample output:

```
Running PowerTOP, please wait...
Generating shell script /etc/tuned/mypowertuneprofile/script.sh
Generating TuneD config file /etc/tuned/mypowertuneprofile/tuned.conf
...
```

2. Open and customize the `/etc/tuned/mypowertuneprofile/tuned.conf` file generated in the preceding step.

As shown in the following example file extract, by default, the PowerTOP recommendations are initially commented out. Uncomment entries according to system requirements.

```
#Automatically generated by powertop2tuned tool

[main]
include=virtual-guest

# Enable SATA link power management for host2
#/sys/class/scsi_host/host2/
link_power_management_policy=med_power_with_dipm

# Enable SATA link power management for host3
#/sys/class/scsi_host/host3/
link_power_management_policy=med_power_with_dipm

# Enable SATA link power management for host4
#/sys/class/scsi_host/host4/
link_power_management_policy=med_power_with_dipm

...

# Wake status for USB device usb1
#/sys/bus/usb/devices/usb1/power/wakeup=enabled
```

```
# Wake status for USB device usb2
#/sys/bus/usb/devices/usb2/power/wakeup=enabled

[sysctl]
# VM writeback timeout
#vm.dirty_writeback_centisecs=1500
...
```

The profile with PowerTOP recommendations and the current active TuneD profile settings is created.

You can also use `powertop2tuned` to create a TuneD profile using an existing HTML PowerTOP report. For example, to use a PowerTOP report in file called `myreport.html` to create a profile called `mypowertuneprofile`, run the following command:

```
sudo powertop2tuned --input myreport.html mypowertuneprofile
```