配置文件优先级

配置文件可位于三个位置：

- **/etc/tmpfiles.d/*.conf**
- **/run/tmpfiles.d/*.conf**
- **/usr/lib/tmpfiles.d/*.conf**

/usr/lib/tmpfiles.d/ 中的文件是由相关 RPM 软件包提供的，不应编辑这些文件。**/run/tmpfiles.d/** 下的文件本身是易失性文件，通常由守护进程用来管理自己的运行时临时文件。**/etc/tmpfiles.d/** 下的文件旨在供管理员配置自定义临时位置，以及覆盖供应商提供的默认值。

如果 **/run/tmpfiles.d/** 中的文件与 **/usr/lib/tmpfiles.d/** 中的文件同名，则系统将使用 **/run/tmpfiles.d/** 中的文件。如果 **/etc/tmpfiles.d/** 中的文件与 **/run/tmpfiles.d/** 或 **/usr/lib/tmpfiles.d/** 中的文件同名，则系统将使用 **/etc/tmpfiles.d/** 中的文件。

给定这些优先级规则后，您将相关文件复制到 **/etc/tmpfiles.d/**，然后编辑该文件，即可轻松覆盖供应商提供的设置。如果以这种方式工作，则确保可从中央配置管理系统轻松管理管理员提供的设置，并且软件包的更新不会覆盖这些设置。

注意

在测试新配置或修改后的配置时，仅从一个配置文件应用命令会十分有用。在命令行中指定配置文件的名称即可实现这一点。

参考文献

systemd-tmpfiles(5)、**tmpfiles.d**(1)、**stat**(2)、**stat**(5) 和 **systemd.timer**(8) man 手册

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

管理临时文件

在本练习中，您将配置 **systemd-tmpfiles** 以更改从 **/tmp** 中删除临时文件的间隔，同时定期清除另一个目录中的文件。

成果

您应能够：

- 配置 **systemd-tmpfiles** 以从 **/tmp** 中删除未使用的临时文件。
- 配置 **systemd-tmpfiles** 以定期清除另一个目录中的文件。

在你开始之前

以 student 身份并使用 student 作为密码登录 workstation。

在 workstation上，运行 **lab scheduling-tempfiles start** 开始本练习。此脚本会创建必要的文件并确保环境设置正确无误。

```
[student@workstation ~]$ lab scheduling-tempfiles start
```

1. 从 workstation，以 student 用户身份打开连接 servera 的 SSH 会话。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

2. 配置 **systemd-tmpfiles** 以清理 **/tmp** 目录，使之不再包含过去五天未使用的文件。确保配置未被任何软件包更新所覆盖。

2.1. 使用 **sudo -i** 命令，切换为 root 用户。

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2.2. 将 **/usr/lib/tmpfiles.d/tmp.conf** 复制到 **/etc/tmpfiles.d/tmp.conf**。

```
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp.conf
```

2.3. 在 **/etc/tmpfiles.d/tmp.conf** 中搜索应用于 **/tmp** 目录的配置行。将该配置行中临时文件的现有期限替换为 **5** 天的新期限。从文件中删除其他所有行，包括注释行。您可以使用 **vim /etc/tmpfiles.d/tmp.conf** 命令来编辑配置文件。**/etc/tmpfiles.d/tmp.conf** 文件应当如下所示：

海量视频题库 mcitpub.com QQ:556462
www.52mvit.com

```
q /tmp 1777 root root 5d
```

上述配置将导致 **systemd-tmpfiles** 确保存在八进制权限设为 **1777** 的目录 **/tmp**。**/tmp** 的用户和组所有权必须为 root。**/tmp** 目录中必须没有过去五天内未使用的临时文件。

2.4. 使用 **systemd-tmpfiles --clean** 命令，验证 **/etc/tmpfiles.d/tmp.conf** 文件中是否包含正确的配置。

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf
```

由于上述命令未返回任何错误，因此可确认配置的设置正确无误。

3. 添加新配置，以确保存在用户和组所有权设为 root 的 **/run/momentary** 目录。目录的八进制权限必须为 **0700**。该配置应清除此目录中在过去 30 秒内未使用的任何文件。

3.1. 创建名为 **/etc/tmpfiles.d/momentary.conf** 并包含以下内容的文件。您可以使用 **vim /etc/tmpfiles.d/momentary.conf** 命令来创建配置文件。

```
d /run/momentary 0700 root root 30s
```

上述配置将导致 **systemd-tmpfiles** 确保存在八进制权限设为 **0700** 的 **/run/momentary** 目录。**/run/momentary** 的用户和组所有权必须为 root。此目录中在过去 30 秒内未使用的任何文件都必须予以清除。

3.2. 使用 **systemd-tmpfiles --create** 命令，验证 **/etc/tmpfiles.d/momentary.conf** 文件中是否包含适当的配置。如果目录 **/run/momentary** 不存在，此命令就会创建该目录。

```
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/momentary.conf
```

由于上述命令未返回任何错误，因此可确认配置的设置正确无误。

3.3. 使用 **ls** 命令，验证创建的 **/run/momentary** 目录是否具有正确的权限、所有者和组所有者。

```
[root@servera ~]# ls -ld /run/momentary
drwx------. 2 root root 40 Mar 21 16:39 /run/momentary
```

请注意，**/run/momentary** 的八进制权限集是 **0700**，而用户和组所有权则设为 root。

4. 验证 **/run/momentary** 目录下在过去 30 秒内未使用的所有文件均已被删除，具体取决于目录的 **systemd-tmpfiles** 配置。

4.1. 使用 **touch** 命令，创建一个名为 **/run/momentary/testfile** 的文件。

```
[root@servera ~]# touch /run/momentary/testfile
```

4.2. 使用 **sleep** 命令，配置 shell 提示符在 30 秒内不要返回。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
[root@servera ~]# sleep 30
```

4.3. 在 shell 提示符返回后，使用 **systemd-tmpfiles --clean** 命令从 **/run/momentary** 中清理陈旧文件，具体取决于 **/etc/tmpfiles.d/momentary.conf** 中所述的规则。

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/momentary.conf
```

上述命令将删除 **/run/momentary/testfile**，因为该文件在 30 秒内未使用，根据 **/etc/tmpfiles.d/momentary.conf** 中所述的规则，应予以删除。

4.4. 使用 **ls -l** 命令验证 **/run/momentary/testfile** 文件已不存在。

```
[root@servera ~]# ls -l /run/momentary/testfile
ls: cannot access '/run/momentary/testfile': No such file or directory
```

4.5. 从 root 用户的 shell 退出，以返回到 student 用户的 shell。

```
[root@servera ~]# exit
logout
[student@servera ~]$
```

4.6. 从 servera 注销。

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 workstation 上，运行 **lab scheduling-tempfiles finish** 来完成本练习。此脚本将删除在实验过程中创建的文件，并确保环境清理干净。

```
[student@workstation ~]$ lab scheduling-tempfiles finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

小测验

计划将来的任务

选择以下问题的正确答案。

1. 哪个命令可以显示当前计划作为延迟作业运行的所有用户作业？
   a. `atq`
   b. `atrm`
   c. `at -c`
   d. `at --display`

2. 哪个命令可以删除具有作业编号 5 的延迟用户作业？
   a. `at -c 5`
   b. `atrm 5`
   c. `at 5`
   d. `at --delete 5`

3. 哪个命令可以显示为当前登录用户计划的所有周期性用户作业？
   a. `crontab -r`
   b. `crontab -l`
   c. `crontab -u`
   d. `crontab -V`

4. 哪种作业格式可以从周一到周五每天的上午 9 点到晚上 6 点每小时执行一次 `/usr/local/bin/daily_backup`？
   a. `00 ***Mon-Fri/usr/local/bin/daily_backup`
   b. `* */9 * * Mon-Fri /usr/local/bin/daily_backup`
   c. `00 */18 * * * /usr/local/bin/daily_backup`
   d. `00 09-18 * * Mon-Fri /usr/local/bin/daily_backup`

5. 哪个目录中包含要每天运行的 shell 脚本？
   a. `/etc/cron.d`
   b. `/etc/cron.hourly`
   c. `/etc/cron.daily`
   d. `/etc/cron.weekly`

海量视频题库 myitpub.com QQ:556462
www.52myit.com

▶ 6. 哪个配置文件定义了每天、每周和每月运行的系统作业的设置？

a. **/etc/crontab**

b. **/etc/anacrontab**

c. **/etc/inittab**

d. **/etc/sysconfig/crond**

▶ 7. 哪个 systemd 单元可以定期触发临时文件的清理操作？

a. systemd-tmpfiles-clean.timer

b. systemd-tmpfiles-clean.service

c. dnf-makecache.timer

d. unbound-anchor.timer

海量视频题库 myitpub.com QQ:556462
www.52myit.com

解决方案

计划将来的任务

选择以下问题的正确答案。

1. 哪个命令可以显示当前计划作为延迟作业运行的所有用户作业?

a. **atq**
b. atrm
c. at -c
d. at --display

2. 哪个命令可以删除具有作业编号 5 的延迟用户作业?

a. at -c 5
b. **atrm 5**
c. at 5
d. at --delete 5

3. 哪个命令可以显示为当前登录用户计划的所有周期性用户作业?

a. crontab -r
b. **crontab -l**
c. crontab -u
d. crontab -V

4. 哪种作业格式可以从周一到周五每天的上午 9 点到晚上 6 点每小时执行一次 /usr/local/bin/daily_backup?

a. 00 ***Mon-Fri/usr/local/bin/daily_backup
b. * */9 * * Mon-Fri /usr/local/bin/daily_backup
c. 00 */18 * * * /usr/local/bin/daily_backup
d. **00 09-18 * * Mon-Fri /usr/local/bin/daily_backup**

5. 哪个目录中包含要每天运行的 shell 脚本?

a. /etc/cron.d
b. /etc/cron.hourly
c. **/etc/cron.daily**
d. /etc/cron.weekly

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

▶ 6. 哪个配置文件定义了每天、每周和每月运行的系统作业的设置？

a. /etc/crontab

b. **/etc/anacrontab**

c. /etc/inittab

d. /etc/sysconfig/crond

▶ 7. 哪个 systemd 单元可以定期触发临时文件的清理操作？

a. systemd-tmpfiles-clean.timer

b. systemd-tmpfiles-clean.service

c. dnf-makecache.timer

d. unbound-anchor.timer

海量视频题库 myitpub.com QQ:556462
www.52myit.com

总结

在本章中，您学到了：

- 计划在将来运行一次的作业称为延迟作业或任务。
- 周期性用户作业可按重复的计划执行用户任务。
- 周期性系统作业则是按重复的计划完成影响整个系统的管理任务。
- systemd 定时器单元既可以执行延迟作业，也可以执行周期性作业。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

章 3

调优系统性能

目标

通过设置调优参数和调整进程的调度优先级来提高系统性能。

培训目标

- 通过选择由 tuned 守护进程管理的调优配置文件来优化系统性能。
- 通过 nice 和 renice 命令，对特定进程进行优先排序或取消其优先排序。

章节

- 调整调优配置文件（及引导式练习）
- 影响进程调度（及引导式练习）

实验

调优系统性能

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

调整调优配置文件

培训目标

学完本节后，您应能够通过选择由 tuned 守护进程管理的调优配置文件来优化系统性能。

调优系统

系统管理员可以基于多种用例工作负载来调整各种设备设置，以此优化系统性能。tuned 守护进程会利用反映特定工作负载要求的调优配置文件，以静态和动态两种方式应用调优调整。

配置静态调优

tuned 守护进程会在服务启动时或选择新的调优配置文件时应用系统设置。静态调优会对配置文件中由 **tuned** 在运行时应用的预定义 kernel 参数进行配置。对于静态调优而言，内核参数是针对整体性能预期而设置的，不会随着活跃度的变化而进行调整。

配置动态调优

对于动态调优而言，tuned 守护进程会监视系统活动，并根据运行时行为的变化来调整设置。从所选调优配置文件中声明的初始设置开始，动态调优会不断进行调优调整以适应当前工作负载。

例如，存储设备在启动和登录期间的使用率会比较高，但如果用户工作负载的内容是使用 Web 浏览器和电子邮件客户端，则活动量会很少。同样，CPU 和网络设备在整个工作日的高峰使用期间会出现活动增加的情况。tuned 守护进程会监视这些组件的活动并调整参数设置，以最大限度提升高活动量期间的性能，并在低活动量期间降低设置值。tuned 守护进程将使用预定义调优配置文件中提供的性能参数。

安装并启用 TUNED

默认情况下，红帽企业 Linux 8 的最小安装中包含并启用了 tuned 软件包。要手动安装并启用该软件包：

```
[root@host ~]$ yum install tuned
[root@host ~]$ systemctl enable --now tuned
Created symlink /etc/systemd/system/multi-user.target.wants/tuned.service → /usr/
lib/systemd/system/tuned.service.
```

选择调优配置文件

Tuned 应用提供的配置文件分为以下几个类别：

- 节能型配置文件
- 性能提升型配置文件

性能提升型配置文件中包括侧重于以下方面的配置文件：

- 存储和网络的低延迟
- 存储和网络的高吞吐量
- 虚拟机性能

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

· 虚拟化主机性能

随红帽企业 Linux 8 分发的调优配置文件

调优配置文件	用途
均衡	非常适合需要在节能和性能之间进行折衷的系统。
desktop	从 balanced 配置文件衍生而来。加快交互式应用响应速度。
throughput-performance	调优系统，以获得最大吞吐量。
latency-performance	非常适合需要牺牲能耗来获取低延迟的服务器系统。
网络延迟	从 latency-performance 配置文件衍生而来。它可以启用额外的网络调优参数，以提供低网络延迟。
网络吞吐量	从 throughput-performance 配置文件衍生而来。应用其他网络调优参数，以获得最大网络吞吐量。
节能	调优系统，以最大程度实现节能。
Oracle	基于 throughput-performance 配置文件，针对 Oracle 数据库负载进行优化。
virtual-guest	当系统在虚拟机上运行时，调优系统以获得最高性能。
virtual-host	当系统充当虚拟机的主机时，调优系统以获得最高性能。

从命令行管理配置文件

tuned-adm 命令可用于更改 tuned 守护进程的设置。**tuned-adm** 命令可以查询当前设置、列出可用的配置文件、为系统推荐调优配置文件、直接更改配置文件或关闭调优。

系统管理员使用 **tuned-adm active** 来确定当前活动的调优配置文件。

```
[root@host ~]# tuned-adm active
Current active profile: virtual-guest
```

tuned-adm list 命令列出所有可用的调优配置文件，包括内置的配置文件和系统管理员创建的自定义调优配置文件。

```
[root@host ~]# tuned-adm list
Available profiles:
- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- powersave
- sap
- throughput-performance
```

海量视频题库 myitpub.com QQ:556462
www.52myit.com

```
- virtual-guest
- virtual-host
Current active profile: virtual-guest
```

使用 **tuned-adm profile profilename** 可以将活动的配置文件切换为更符合系统当前调优要求的其他配置文件。

```
[root@host ~]$ tuned-adm profile throughput-performance
[root@host ~]$ tuned-adm active
Current active profile: throughput-performance
```

tuned-adm 命令可以为系统推荐调优配置文件。该机制用于在安装后确定系统的默认配置文件。

```
[root@host ~]$ tuned-adm recommend
virtual-guest
```

> **注意**
> **tuned-adm recommend** 的输出基于各种系统特征，包括系统是否为虚拟机以及在系统安装期间选择的其他预定义类别。

要恢复当前配置文件所做的设置更改，可切换到另一个配置文件或停用 tuned 守护进程。利用 **tuned-adm off** 关闭 tuned 调优活动。

```
[root@host ~]$ tuned-adm off
[root@host ~]$ tuned-adm active
No current active profile.
```

通过 WEB 控制台管理配置文件

要通过 Web 控制台管理系统性能配置文件，请以具有特权的访问权限登录。单击 Reuse my password for privileged tasks 选项。这将允许用户以 sudo 权限执行修改系统性能配置文件的命令。

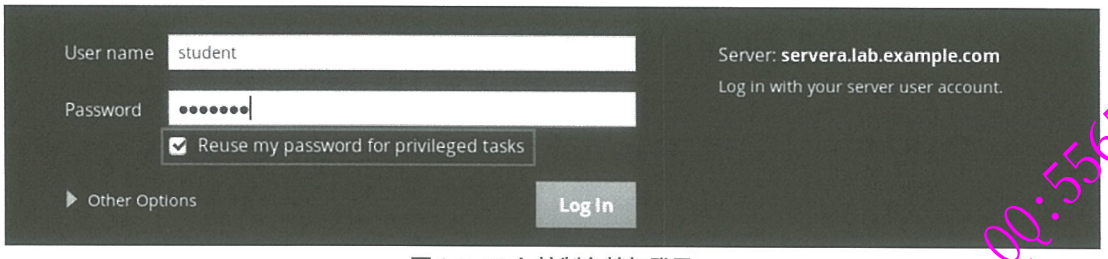

图 3.1: Web 控制台特权登录

以特权用户的身份单击左侧导航栏中的 Systems 菜单选项。当前活动的配置文件将显示在 Performance Profile 字段中。要选择其他配置文件，请单击活动配置文件链接。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

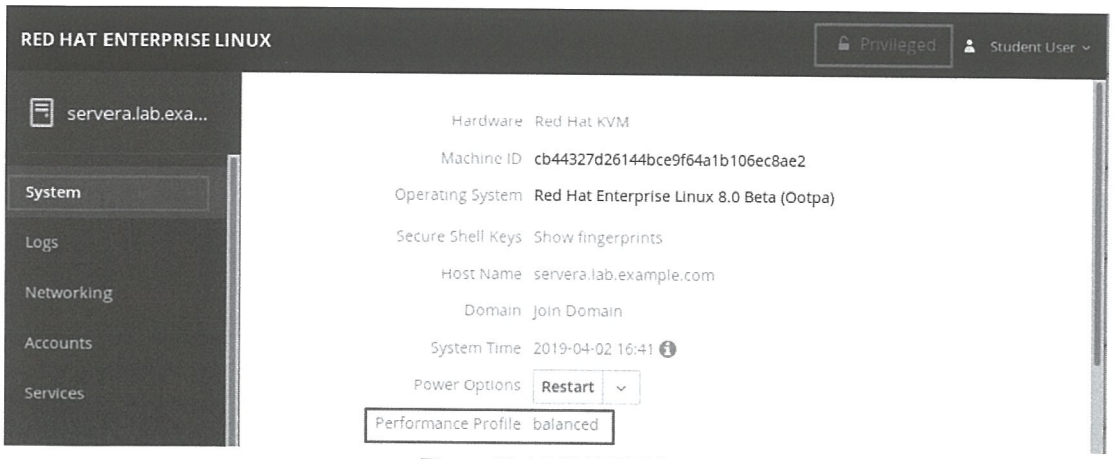

图 3.2: 活动的性能配置文件

在 Change Performance Profile 用户界面中，滚动配置文件列表以选择最适合系统用途的配置文件。

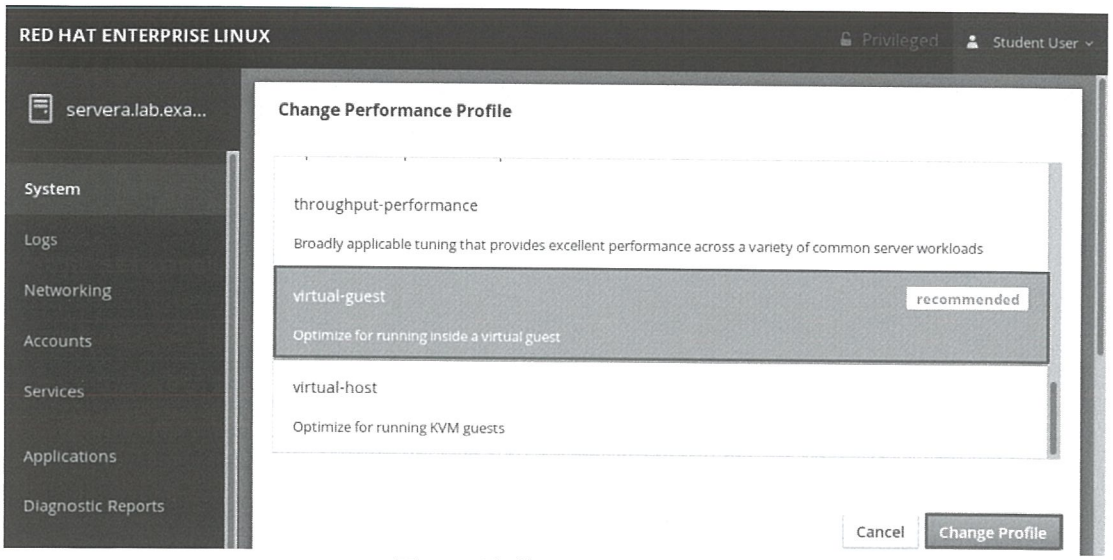

图 3.3: 选择首选的性能配置文件

要验证更改，请返回主 System 页面并确认 Performance Profile 字段中是否显示活动的配置文件。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

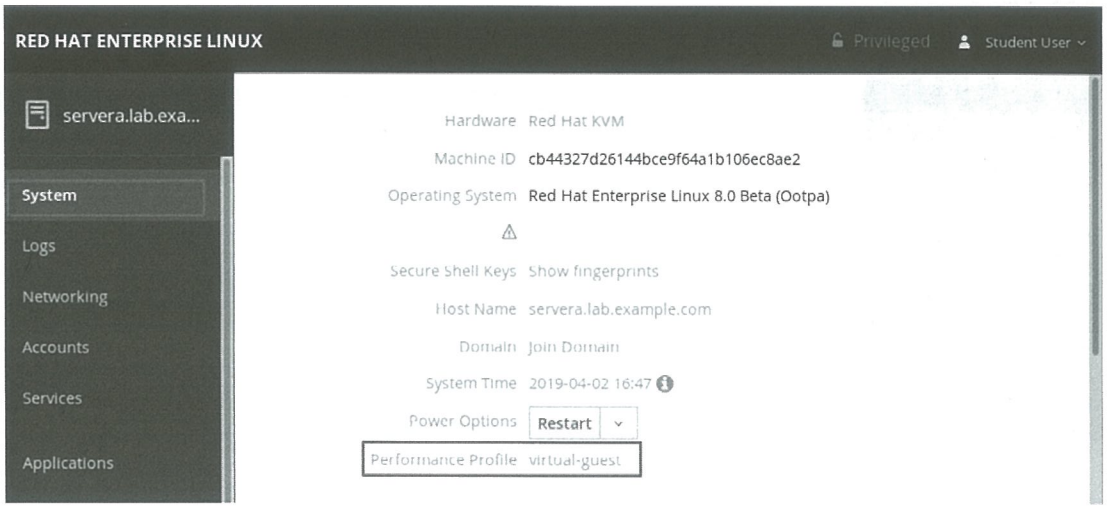

图 3.4: 验证活动的性能配置文件

参考文献

tuned(8)、**tuned.conf**(5)、**tuned-main.conf**(5) 和 **tuned-adm**(1) man page

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

指导练习

调整调优配置文件

在本练习中，您将通过激活 tuned 服务并应用调优配置文件来调优服务器的性能。

成果

您应能够配置系统以使用调优配置文件。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab tuning-profiles start** 命令。该命令将运行一个起始脚本，以确定 servera 主机是否可从网络访问。

```
[student@workstation ~]$ lab tuning-profiles start
```

1. 从 workstation，使用 SSH 以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

2. 验证是否安装、启用和启动了 tuned 软件包。

2.1. 使用 **yum** 确认是否安装了 tuned 软件包。

```
[student@servera ~]$ yum list tuned
...output omitted...
Installed Packages
tuned.noarch                    2.10.0-15.el8                    @anaconda
```

2.2. **systemctl is-enabled tuned; systemctl is-active tuned** 命令可显示其启用和运行状态。

```
[student@servera ~]$ systemctl is-enabled tuned; systemctl is-active tuned
enabled
active
```

3. 列出可用的调优配置文件并确定活动的配置文件。如果 sudo 提示输入密码，请在提示符后面输入 student。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
[student@servera ~]$ sudo tuned-adm list
[sudo] password for student: student
Available profiles:
- 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
- powersave                   - Optimize for low power consumption
- 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
```

4. 将当前活动的调优配置文件更改为 powersave，然后确认结果。如果 sudo 提示输入密码，请在提示符后面输入 student。

4.1. 更改当前活动的调优配置文件。

```
[student@servera ~]$ sudo tuned-adm profile powersave
```

4.2. 确认 powersave 为活动的调优配置文件。

```
[student@servera ~]$ sudo tuned-adm active
Current active profile: powersave
```

5. 从 servera 退出。

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 workstation 上，运行 **lab tuning-profiles finish** 脚本来结束本练习。

```
[student@workstation ~]$ lab tuning-profiles finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

影响进程调度

培训目标

学完本节后，您应能够通过 **nice** 和 **renice** 命令对特定进程进行优先排序或取消其优先排序。

LINUX 进程调度和多任务

现代计算机系统中既包含只有单个 CPU 且任何时候都只能执行单个指令的低端系统，也包含高性能超级计算机，这些超级计算机每台配备数百个 CPU，且每个 CPU 上有数十甚至数百个处理核心，可并行执行大量指令。所有这些系统还有一个共同点：它们需要运行的进程线程数量超出了其 CPU 数量。

通过使用称为时间片或多任务的技术，Linux 和其他操作系统可运行超出其处理单元数的进程。操作系统进程调度程序在单个核心上的进程之间快速切换，从而给人一种有多个进程在同时运行的印象。

相对优先级

不同进程的重要性水平各不相同。进程调度程序可以配置为针对不同的进程采用不同的调度策略。常规系统上运行的大多数进程所使用的调度策略称为 SCHED_OTHER（也称为 SCHED_NORMAL），但还有一些其他策略可满足各种各样的工作负载需求。

由于并非所有进程都同等重要，因此可为采用 SCHED_NORMAL 策略运行的进程指定相对优先级。此优先级称为进程的 nice 值，对于任何进程，都可以组织成 **40** 种不同级别的 nice 值。

nice 级别值的范围介于 -20（最高优先级）到 19（最低优先级）之间。默认情况下，进程将继承其父进程的 nice 级别，通常为 0。nice 级别越高，表示优先级越低（该进程容易将其 CPU 使用量让给其他进程）；nice 级别越低，表示优先级越高（该进程更加不倾向于让出 CPU）。如果不存在资源争用（例如，当活动进程数少于可用 CPU 核心数时），即使 nice 级别高的进程也将仍使用它们可使用的所有可用 CPU 资源。但当请求 CPU 时间的进程数超过可用核心数时，nice 级别较高的进程将比 nice 级别较低的进程收到更少的 CPU 时间。

设置 NICE 级别和权限

为很占 CPU 资源的进程设置较低的 nice 级别可能对会同一系统上运行的其他进程的性能造成负面影响，所以只有 root 用户可以降低进程的 nice 级别。

非特权用户的权限仅限于提高自己进程的 nice 级别。他们不能降低自己进程的 nice 级别，也不能修改其他用户进程的 nice 级别。

报告 NICE 级别

有几个工具可以显示正在运行的进程的 nice 级别。进程管理工具（如 top）默认显示 nice 级别。其他工具（如 **ps** 命令）可在使用适当选项的情况下显示 nice 级别。

使用 top 显示 nice 级别

使用 **top** 命令可通过交互方式查看和管理进程。默认配置显示的是关于 nice 级别与优先级的两列。**NI** 列显示进程的 nice 值，而 **PR** 列则显示其调度的优先级。在 **top** 界面中，nice 级别映射至

海量视频题库 mvitpub.com QQ:556462
www.52mvit.com

内部系统优先级队列，如下图所示。例如，nice 级别 -20 映射至 **PR** 列中的 0。nice 级别 19 映射至 **PR** 列中的优先级 39。

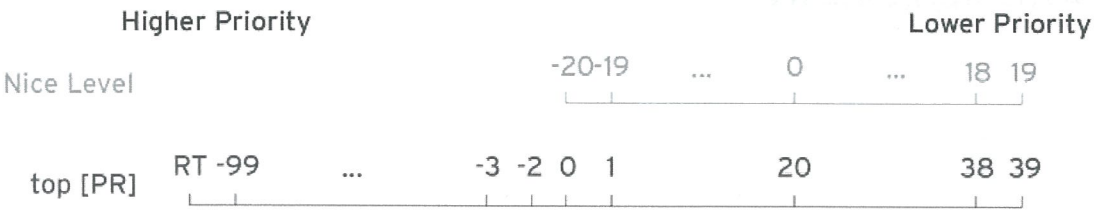

图 3.5: top 报告的 nice 级别

通过命令行显示 nice 级别

ps 命令可显示进程的 nice 级别，但仅限在包含正确格式化选项的情况下。

以下 **ps** 命令列出所有进程，包括其 PID、进程名称、nice 级别和调度类，按 nice 级别降序排列。在 **CLS** 调度类这一列中显示 **TS** 的进程将依照 SCHED_NORMAL 调度策略运行。nice 级别显示为短划线 (**-**) 的进程将依照其他调度策略运行，并被调度程序解读为具有较高的优先级。其他调度策略的详细信息已超出了本课程的范畴。

```
[user@host ~]$ ps axo pid,comm,nice,cls --sort=-nice
  PID COMMAND          NI CLS
   30 khugepaged       19  TS
   29 ksmd              5  TS
    1 systemd           0  TS
    2 kthreadd          0  TS
    9 ksoftirqd/0       0  TS
   10 rcu_sched         0  TS
   11 migration/0       -  FF
   12 watchdog/0        -  FF
...output omitted...
```

启动具有不同 NICE 级别的进程

在进程创建过程中，进程会继承其父级的 nice 级别。从命令行启动某个进程时，该进程将从启动它的 shell 进程那里继承 nice 级别。通常，这将导致新进程以 nice 级别 0 运行。

以下示例将从 shell 启动进程并显示该进程的 nice 值。注意在 **ps** 中使用 PID 选项以指定所请求的输出。

```
[user@host ~]$ sha1sum /dev/zero &
[1] 3480
[user@host ~]$ ps -o pid,comm,nice 3480
  PID COMMAND          NI
 3480 sha1sum           0
```

所有用户都可以使用 **nice** 命令来启动具有默认或更高 nice 级别的命令。在不带选项的情况下，**nice** 命令将启动一个默认 nice 值为 10 的进程。

以下示例将启动 **sha1sum** 命令作为具有默认 nice 级别的后台作业，并显示该进程的 nice 级别：

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

```
[user@host ~]$ nice sha1sum /dev/zero &
[1] 3517
[user@host ~]$ ps -o pid,comm,nice 3517
  PID COMMAND          NI
 3517 sha1sum          10
```

利用 **-n** 选项可以将用户定义的 nice 级别应用于启动的进程。默认设置是将 10 添加到进程的当前 nice 级别中。以下示例将启动某个命令作为具有用户定义的 nice 值的后台作业，并显示该进程的 nice 级别：

```
[user@host ~]$ nice -n 15 sha1sum &
[1] 3521
[user@host ~]$ ps -o pid,comm,nice 3521
  PID COMMAND          NI
 3521 sha1sum          15
```

重要

非特权用户只能在其当前值的基础上提高 nice 值，最大可提高至 19。一旦提高 nice 值，非特权用户便无法降低该值以恢复之前的 nice 级别。`root` 用户可以在当前的任何级别基础上降低 nice 级别，最低可降至 -20。

更改现有进程的 NICE 级别

可以使用 **renice** 命令来更改现有进程的 nice 级别。以下示例将使用上一示例中的 PID 标识符将当前的 nice 级别从 15 更改为所需的 nice 级别 19。

```
[user@host ~]$ renice -n 19 3521
3521 (process ID) old priority 15, new priority 19
```

也可以使用 **top** 命令更改进程的 nice 级别。在 **top** 交互式界面中，按下 **r** 选项以访问 **renice** 命令，后跟要更改的 PID 和新的 nice 级别。

参考文献

nice(1)、**renice**(1)、**top**(1) 和 **sched_setscheduler**(2) man page。

海量视频题库 myitpub.com QQ:556546 2
www.52myit.com

指导练习

影响进程调度

在本练习中，您将使用 **nice** 和 **renice** 命令来调整进程的调度优先级，并观察它对进程执行的影响。

成果

您应该能够调整进程的调度优先级。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab tuning-procscheduling start** 命令。该命令将运行一个起始脚本，以确定 servera 主机是否可从网络访问。

```
[student@workstation ~]$ lab tuning-procscheduling start
```

1. 从 workstation，使用 SSH 以 student 用户身份登录 servera。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

2. 确定 servera 上的 CPU 核心数，然后为每个核心启动 **sha1sum /dev/zero &** 命令的两个实例。

   2.1. 利用 **grep**，从 **/proc/cpuinfo** 文件中解析现有虚拟处理器（CPU 核心）的数量。

```
[student@servera ~]$ grep -c '^processor' /proc/cpuinfo
2
```

   2.2. 使用循环命令启动 **sha1sum /dev/zero &** 命令的多个实例。针对上一步中找到的每个虚拟处理器启动两个实例。在本例中将是四个实例。输出中的 PID 值将与示例有所不同。

```
[student@servera ~]$ for i in $(seq 1 4); do sha1sum /dev/zero & done
[1] 2643
[2] 2644
[3] 2645
[4] 2646
```

3. 验证每个 sha1sum 进程都有后台作业正在运行。

海量视频题库 myitpub.com QQ:565462
www.52myit.com

```
[student@servera ~]$ jobs
[1]    Running                 sha1sum /dev/zero &
[2]    Running                 sha1sum /dev/zero &
[3]-   Running                 sha1sum /dev/zero &
[4]+   Running                 sha1sum /dev/zero &
```

4. 使用 **ps** 和 **pgrep** 命令显示每个 sha1sum 进程的 CPU 使用百分比。

```
[student@servera ~]$ ps u $(pgrep sha1sum)
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
student   2643 49.8  0.0 228360  1744 pts/0    R    11:15   6:09 sha1sum /dev/zero
student   2644 49.8  0.0 228360  1780 pts/0    R    11:15   6:09 sha1sum /dev/zero
student   2645 49.8  0.0 228360  1748 pts/0    R    11:15   6:09 sha1sum /dev/zero
student   2646 49.8  0.0 228360  1780 pts/0    R    11:15   6:09 sha1sum /dev/zero
```

5. 终止所有 sha1sum 进程，然后验证没有正在运行的作业。

5.1. 使用 **pkill** 命令终止名称模式为 sha1sum 的所有正在运行的进程。

```
[student@servera ~]$ pkill sha1sum
[2]    Terminated              sha1sum /dev/zero
[4]+   Terminated              sha1sum /dev/zero
[1]-   Terminated              sha1sum /dev/zero
[3]+   Terminated              sha1sum /dev/zero
```

5.2. 验证没有正在运行的作业。

```
[student@servera ~]$ jobs
[student@servera ~]$
```

6. 启动 **sha1sum /dev/zero &** 的多个实例，然后启动另外一个 nice 级别为 10 的 **sha1sum /dev/zero &** 实例。启动至少与系统虚拟处理器数目一样多的实例。在本例中，我们启动了 3 个常规实例，外加一个具有更高 nice 级别的实例。

6.1. 使用循环命令启动 **sha1sum /dev/zero &** 的三个实例。

```
[student@servera ~]$ for i in $(seq 1 3); do sha1sum /dev/zero & done
[1] 1947
[2] 1948
[3] 1949
```

6.2. 使用 **nice** 命令启动 nice 级别为 10 的第四个实例。

```
[student@servera ~]$ nice -n 10 sha1sum /dev/zero &
[4] 1953
```

7. 使用 **ps** 和 **pgrep** 命令显示每个进程的 PID、CPU 使用百分比、nice 值和可执行名称。nice 值为 10 的实例应显示比其他实例更低的 CPU 使用百分比。

海量视频题库 mvitpub.com QQ:556462
www.52mvit.com

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
  PID %CPU  NI COMMAND
 1947 66.0   0 sha1sum
 1948 65.7   0 sha1sum
 1949 66.1   0 sha1sum
 1953  6.7  10 sha1sum
```

8. 使用 **sudo renice** 命令降低上一步中进程的 nice 级别。请注意 nice 级别为 10 的进程实例的 PID 值。使用该进程 PID 将其 nice 级别降至 5。

```
[student@servera ~]$ sudo renice -n 5 1953
[sudo] password for student: student
1953 (process ID) old priority 10, new priority 5
```

9. 重复 **ps** 和 **pgrep** 命令，重新显示 CPU 百分比和 nice 级别。

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
  PID %CPU  NI COMMAND
 1947 63.8   0 sha1sum
 1948 62.8   0 sha1sum
 1949 65.3   0 sha1sum
 1953  9.1   5 sha1sum
```

10. 从 servera 退出。

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

完成

在 workstation 上，运行 **lab tuning-procscheduling finish** 脚本来结束本练习。

```
[student@workstation ~]$ lab tuning-procscheduling finish
```

本引导式练习到此结束。

海量视频题库 myitpub.com QQ:5565462
www.52myit.com

开放研究实验

调优系统性能

性能检查清单

在本实验中，您将应用特定的调优配置文件，并调整具有高 CPU 使用率的现有进程的调度优先级。

成果

您应能够：

- 为计算机系统激活特定的调优配置文件。
- 调整某个进程的 CPU 调度优先级。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab tuning-review start** 命令。该命令将运行一个起始脚本，以确定 serverb 主机是否可从网络访问。

```
[student@workstation ~]$ lab tuning-review start
```

1. 将 serverb 的当前调优配置文件更改为 balanced，这是一个常规的非专用调优配置文件。
2. serverb 上两个进程的 CPU 使用百分比较高。请将每个进程的 nice 级别调整为 10，以便为其他进程腾出更多 CPU 时间。

评估

在 workstation 上，运行 **lab tuning-review grade** 命令来确认是否成功完成本实验练习。

```
[student@workstation ~]$ lab tuning-review grade
```

完成

在 workstation 上，运行 **lab tuning-review finish** 脚本来完成本练习。

```
[student@workstation ~]$ lab tuning-review finish
```

本实验到此结束。

海量视频题库 myitpub.com QQ:556462
www.52myit.com

解决方案

调优系统性能

性能检查清单

在本实验中，您将应用特定的调优配置文件，并调整具有高 CPU 使用率的现有进程的调度优先级。

成果

您应能够：

- 为计算机系统激活特定的调优配置文件。
- 调整某个进程的 CPU 调度优先级。

在你开始之前

在 workstation 上，以 student 用户身份并使用密码 student 进行登录。

从 workstation，运行 **lab tuning-review start** 命令。该命令将运行一个起始脚本，以确定 serverb 主机是否可从网络访问。

```
[student@workstation ~]$ lab tuning-review start
```

1. 将 serverb 的当前调优配置文件更改为 balanced，这是一个常规的非专用调优配置文件。

1.1. 从 workstation，以 student 用户身份打开连接 serverb 的 SSH 会话。系统已配置为使用 SSH 密钥来进行身份验证，因此不需要提供密码。

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

1.2. 使用 **yum** 确认是否安装了 tuned 软件包。

```
[student@serverb ~]$ yum list tuned
...output omitted...
Installed Packages
tuned.noarch                    2.10.0-15.el8                    @anaconda
```

1.3. 使用 **systemctl is-enabled tuned** 命令显示 tuned 服务的启用状态。

```
[student@serverb ~]$ systemctl is-enabled tuned
enabled
```

1.4. 列出所有可用的调优配置文件及其说明。请注意，当前活动的配置文件为 virtual-guest。

海量视频题库 myitpub.com QQ:556462
www.52myit.com