

Ubuntu/Debian 系统的低延时调优指南(Low Latency)

工厂日常

Exported on 09/09/2025

Table of Contents

1	Ubuntu操作系统低延时调优	4
1.1	安装依赖软件包	4
1.2	使用NetworkManager管理网络	4
1.3	系统服务优化	5
1.4	设置Tuned Profile	5
1.5	优化内核参数	5
2	Debian 操作系统低延时调优	7
2.1	安装依赖软件包	7
2.2	系统服务优化	8
2.3	优化Tuned Profile	8
2.4	优化内核参数	8
3	安装低延时网卡	9
3.1	Solarflare 低延时卡驱动安装和优化	9
3.2	Exanic低延时网卡驱动安装和优化	11
4	优化后系统检查	12

• 文档信息

分发范围	客户	
使用范围	Ubuntu 18.04.x Ubuntu 20.04.x Ubuntu 22.04.x Ubuntu 24.04.x	
作者	@曹永乐	
初始版本	将原始文档转换到此文档	
v1.0	新增关闭ufw和irq服务 调整tuned的profile 新增2个软件包: mlocate lm-sensors	2024/09/12
v1.1	增加debian 12.11	2025/07/08
v1.2	增加优化后系统检查	2025/09/08

• 概要

此文档用于Ubuntu重新安装后的调优和设置。

备注：应先完成硬件层低延时调优，对于中鑫服务器在出厂是硬件层均已完成。

1 Ubuntu操作系统低延时调优

1.1 安装依赖软件包

```
# For Ubuntu 24.4.x
root@ubuntu:~# apt install linux-headers-$(uname -r) automake build-essential g++
cmake libssl-dev dosfstools libcap-dev linux-tools-common linux-tools-generic lsscsi
tuned alien linux-source plocate lm-sensors

# For ubuntu 22.04.x/20.04.x
root@ubuntu:~# apt install linux-headers-$(uname -r) automake build-essential g++
cmake libssl-dev dosfstools libcap-dev linux-tools-common linux-tools-generic lsscsi
tuned alien linux-source mlocate lm-sensors

# For ubuntu 18.04.x
root@ubuntu:~# apt install linux-headers-$(uname -r) automake build-essential g++
cmake libssl-dev dosfstools libcap-dev linux-tools-common linux-tools-generic lsscsi
tuned alien mlocate lm-sensors

# 安装失败时更新升级yum源即可解决
root@ubuntu:~# apt upgrade
root@ubuntu:~# apt update

# 如果需要onload_tcpdump功能
root@ubuntu:~# apt install libcap-dev libnl-3-dev libnl-genl-3-dev pkg-config

# 为确保后续安装不会内核和相关开发包，建议禁止内核自动升级，强烈建议操作
root@ubuntu:~# apt-mark hold linux-generic linux-headers-generic
```

1.2 使用NetworkManager管理网络

```
# 切换到root帐号
ubuntu@ubuntu:~$ sudo -i
# 安装network-manager工具
root@ubuntu:~# apt install -y network-manager
# 编辑网卡配置文件，在最后一行添加renderer: NetworkManager参数
cat > /etc/netplan/00-installer-config.yaml <<EOF
network:
  version: 2
  renderer: NetworkManager
EOF

# 使刚才添加的参数生效（记住，此操作后如是dhcp分配的地址可能会切换新地址）
root@ubuntu:~# netplan apply
```

```
# 开启NetworkManager服务, 并设置开机启动
root@ubuntu:~# systemctl start NetworkManager
root@ubuntu:~# systemctl enable NetworkManager
```

1.3 系统服务优化

```
# 关闭一下服务, 使用root帐号执行
systemctl disable ufw.service
systemctl disable irqbalance.service
systemctl disable postfix.service
systemctl disable cpupower.service
```

1.4 设置Tuned Profile

```
# 使用network-latency
root@ubuntu:~# systemctl enable tuned
root@ubuntu:~# tuned-adm profile network-latency
```

1.5 优化内核参数

```
# 核心隔离的内核参数
# 前2个核心隔离给system使用, 2-17留给程序使用, 这里的数据根据实际核心数量手动调整
root@ubuntu:~# sed -i 's/GRUB_CMDLINE_LINUX="/&intel_idle.max_cstate=0
processor.max_cstate=0 idle=poll pcie_aspm.policy=performance mce=ignore_ce
ipmi_si.force_kipmid=0 nmi_watchdog=0 noht nohz=off nohalt nosoftlockup splash
irqaffinity=0-1 isolcpus=2-17 nohz_full=2-17 rcu_nocbs=2-17 selinux=0 audit=0
tsc=reliable/' /etc/default/grub

# 非核心隔离的内核参数
root@ubuntu:~# sed -i 's/GRUB_CMDLINE_LINUX="/&intel_idle.max_cstate=0
processor.max_cstate=0 idle=poll pcie_aspm.policy=performance mce=ignore_ce
ipmi_si.force_kipmid=0 nmi_watchdog=0 noht nohz=off nohalt nosoftlockup splash
selinux=0 audit=0 tsc=reliable/' /etc/default/grub

# 编译grub引导参数
root@ubuntu:~# grub-mkconfig -o /boot/grub/grub.cfg

# 重启生效
```

```
root@ubuntu:~# reboot
```

2 Debian 操作系统低延时调优

- 目前验证版本：
 - Debian 12.11

2.1 安装依赖软件包

在安装软件包前，需要先配置软件包源

- 添加仓库源

```
# 添加源
mv /etc/apt/sources.list /etc/apt/sources.list.old
cat > /etc/apt/sources.list << EOF
deb https://deb.debian.org/debian/ bookworm main contrib non-free non-free-firmware
deb-src https://deb.debian.org/debian/ bookworm main contrib non-free non-free-
firmware
deb https://deb.debian.org/debian/ bookworm-updates main contrib non-free non-free-
firmware
deb-src https://deb.debian.org/debian/ bookworm-updates main contrib non-free non-
free-firmware
deb https://deb.debian.org/debian/ bookworm-backports main contrib non-free non-free-
firmware
deb-src https://deb.debian.org/debian/ bookworm-backports main contrib non-free non-
free-firmware
deb https://deb.debian.org/debian-security/ bookworm-security main contrib non-free
non-free-firmware
deb-src https://deb.debian.org/debian-security/ bookworm-security main contrib non-
free non-free-firmware
EOF

# 更新源
apt update
```

- 安装软件包

```
# 安装软件包
apt install linux-headers-$(uname -r) automake build-essential g++ cmake libssl-dev
dosfstools libcap-dev lsscsi tuned alien mlocate lm-sensors libnl-3-dev libnl-genl-3-
dev pkg-config
```

2.2 系统服务优化

```
systemctl disable ufw.service
systemctl disable irqbalance.service
systemctl disable postfix.service
systemctl disable cpupower.service
```

2.3 优化Tuned Profile

```
tuned-adm profile network-latency
```

2.4 优化内核参数

```
# 核心隔离的内核参数
# 前2个核心隔离给system使用，2-17留给程序使用，这里的数据根据实际核心数量手动调整
sed -i 's/GRUB_CMDLINE_LINUX="&intel_idle.max_cstate=0 processor.max_cstate=0
idle=poll pcie_aspm.policy=performance mce=ignore_ce ipmi_si.force_kipmdd=0
nmi_watchdog=0 noht nohz=off nohalt nosoftlockup splash irqaffinity=0-1 isolcpus=2-17
nohz_full=2-17 rcu_nocbs=2-17 selinux=0 audit=0 tsc=reliable/' /etc/default/grub

# 非核心隔离的内核参数
sed -i 's/GRUB_CMDLINE_LINUX="&intel_idle.max_cstate=0 processor.max_cstate=0
idle=poll pcie_aspm.policy=performance mce=ignore_ce ipmi_si.force_kipmdd=0
nmi_watchdog=0 noht nohz=off nohalt nosoftlockup splash selinux=0 audit=0
tsc=reliable/' /etc/default/grub

# 编译grub引导参数
grub-mkconfig -o /boot/grub/grub.cfg

# 重启生效
reboot
```


3 安装低延时网卡

3.1 Solarflare 低延时卡驱动安装和优化

备注：Ubuntu18.04需要在BIOS关闭 secure boot，其余版本不需操作

- **Onload V7版本**
- **建议操作系统版本**
 - Ubuntu Server LTS 18.04.x

```
# 解压onload
root@ubuntu:~# tar -zxvf onload-7.1.3.202.tgz
# 进入安装目录
root@ubuntu:~# cd onload-7.1.3.202/scripts
# 编译安装onload
root@ubuntu:~/onload-7.1.3.202/scripts# ./onload_install
root@ubuntu:~/onload-7.1.3.202/scripts# onload_tool reload
```

- **Onload V8版本**
- **建议操作系统版本：**
 - Ubuntu Server LTS 20.04.x
 - Ubuntu Server LTS 22.04.x
 - Ubuntu Server LTS 24.04.x
 - Debian 12.11

```
# 解压onload
root@ubuntu:~# tar -zxvf onload-8.1.3.40.tgz
# 进入安装目录
root@ubuntu:~# cd onload-8.1.3.40/scripts
# 编译安装onload
root@ubuntu:~/onload-8.1.3.40/scripts# ./onload_install
root@ubuntu:~/onload-8.1.3.40/scripts# onload_tool reload
```

- **特殊处理**
- Ubuntu Server LTS 24.04.x

需要特别关注，sfc等相关模块未放置到initram中，需要额外执行一下命令以确保重新引导后驱动可以生效

```
add_onload_initramfs() {
    cp /etc/initramfs-tools/modules /etc/initramfs-tools/modules.bak
    for i in /usr/lib/modules/$(uname -r)/extra/sfc.ko /usr/lib/modules/$(uname -r)/
extra/sfc_char.ko /usr/lib/modules/$(uname -r)/extra/sfc_resource.ko /usr/lib/
modules/$(uname -r)/extra/sfc_onload.ko
do
```

```

        grep $i /etc/initramfs-tools/modules >/dev/null && echo $i, 已包含 || echo $i
>> /etc/initramfs-tools/modules
    done
    update-initramfs -u
}

add_onload_initramfs

```

• 网卡固件升级

```

# 安装网卡固件包
root@ubuntu:~# dpkg -i sfutils_8.3.3.1000-2_amd64.deb
# 升级网卡固件
root@ubuntu:~# sfupdate --write
root@ubuntu:~# sfupdate

# 此外也可以使用ISO引导系统完成升级

```

• 网卡优化

```

# 开启网卡超低延时模式
# 注意替换命令中的网卡名
sfboot --adapter=eth2 firmware-variant=ultra-low-latency
sfboot --adapter=eth3 firmware-variant=ultra-low-latency

# 如果需要改回原来模式，使用如下命令
# 注意替换命令中的网卡名
sfboot --adapter=eth2 firmware-variant=full-feature
sfboot --adapter=eth3 firmware-variant=full-feature

# 禁用低延时网卡PXE
sfboot boot-type=disabled

# 设置发送队列和接收队列
ethtool -G 网卡名 rx 4096 tx 2048
# 关闭自适应
ethtool -C 网卡名 rx-usecs 0 rx-usecs-irq 0 adaptive-rx off

# 将设置命令添加到/etc/rc.local
# 注意替换命令中的网卡名
cat >> /etc/rc.local << EOF
ethtool -G eth2 rx 4096 tx 2048
ethtool -G eth3 rx 4096 tx 2048

ethtool -C eth2 rx-usecs 0 rx-usecs-irq 0 adaptive-rx off
ethtool -C eth3 rx-usecs 0 rx-usecs-irq 0 adaptive-rx off
EOF

```

```
# 设置开机自启动，使用root账户执行
chmod a+x /etc/rc.local
systemctl enable rc-local
```

3.2 Exanic低延时网卡驱动安装和优化

```
# 解压软件包
root@ubuntu:~# tar zxvf exanic-2.7.4.tar.gz
# 进入安装目录
root@ubuntu:~# cd exanic*
# 编译安装
root@ubuntu:~# make
root@ubuntu:~# make install
```

4 优化后系统检查

调优全部完成后，建议重启机器后做确认检查

- 检查项目条目和命令

```
# 内存检查，主要核对：内存容量和内存频率
free -h
dmidecode -t 17

# CPU检查，只要核对CPU主频和温度
lscpu
turbostat -i 1

# 内核参数检查
cat /proc/cmdline

# 服务检查
systemctl list-unit-files --state=enabled

# 网卡优化检查，注意替换命令中的网卡名
sfboot
ethtool -g eth2
ethtool -g eth3

ethtool eth2
ethtool eth3

# Solarflare驱动和onload检查
modinfo sfc
modinfo onload
onload ping 127.0.0.1

# Exanic驱动和exasock
modinfo exanic
modinfo exasock
exasock ping 127.0.0.1
```