

## Linux性能优化（十）——CPU性能分析工具-爱代码爱编程

category: 性能优化 category: linux category: cpu性能分析工具 2022-02-23

# 一、CPU性能指标

## 1、CPU使用率

CPU使用率是最常见的一个性能指标，描述了非空闲时间占总CPU 时间的百分比，根据CPU上运行任务的不同，分为用户CPU、系统CPU、等待IO CPU、软中断和硬中断等。

用户CPU使用率包括用户态CPU使用率（user）和低优先级用户态CPU使用率（nice），表示CPU在用户态运行的时间百分比。用户CPU使用率高，通常说明有应用程序比较繁忙。

系统CPU使用率，表示CPU在内核态运行的时间百分比（不包括中断）。系统CPU使用率高，说明内核比较繁忙。

等待IO的CPU使用率，即iowait，表示等待IO的时间百分比。iowait 高，通常说明系统与硬件设备的IO交互时间比较长。

软中断和硬中断的CPU使用率，分别表示内核调用软中断处理程序、硬中断处理程序的时间百分比。中断CPU使用率高，通常说明系统发生大量的中断。

窃取CPU使用率（steal）和客户CPU使用率（guest），分别表示被其它虚拟机占用的CPU时间百分比，和运行客户虚拟机的CPU 时间百分比。

## 2、CPU平均负载

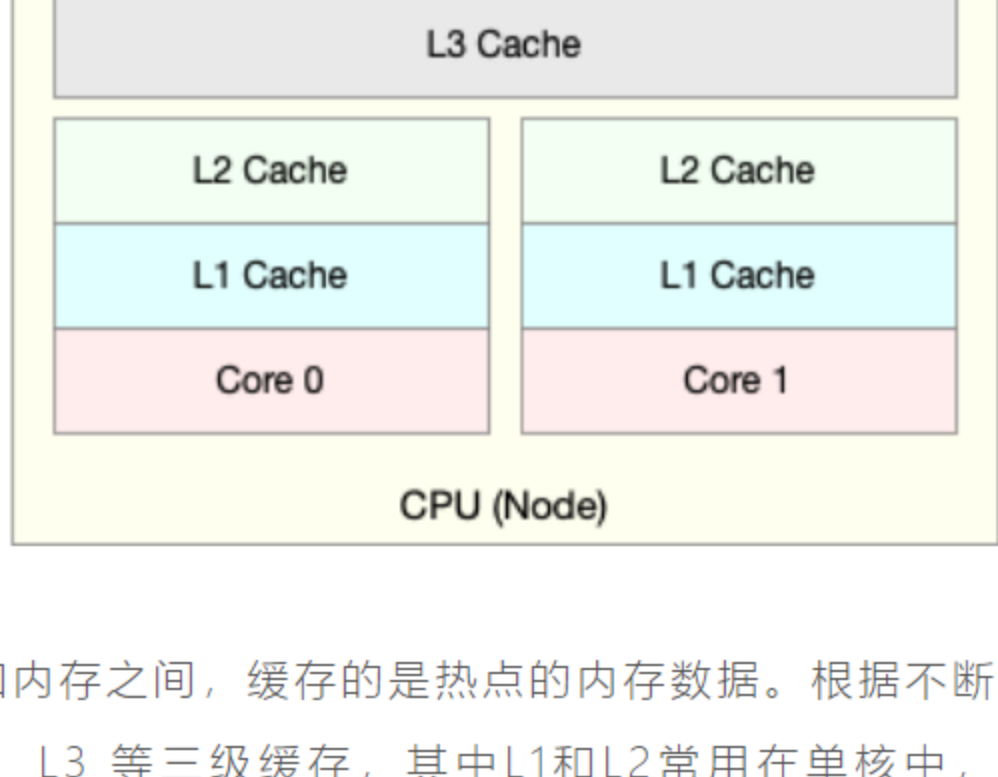
CPU平均负载是系统的平均活跃进程数，反应了系统的整体负载情况，主要包括三个数值，分别指过去1分钟、过去5分钟和过去15分钟的平均负载。理想情况下，平均负载等于逻辑CPU个数，表示每个CPU都恰好被充分利用。如果平均负载大于逻辑CPU个数，表示负载比较重。

## 3、进程上下文切换

进程上下文切换包括无法获取资源而导致的自愿上下文切换和被系统强制调度导致的非自愿上下文切换。上下文切换是保证Linux正常运行的一项核心功能，但过多上下文切换会将原本运行进程的CPU时间，消耗在寄存器、内核栈以及虚拟内存等数据的保存和恢复上，缩短进程真正运行的时间，成为性能瓶颈。

## 4、CPU缓存命中率

由于CPU发展的速度远快于内存的发展，CPU的处理速度就比内存的访问速度快得多。CPU在访问内存时，需要要等待内存的响应。



CPU缓存的速度介于CPU和内存之间，缓存的是热点的内存数据。根据不断增长的热点数据，CPU缓存按照大小不同分为L1、L2、L3 等三级缓存，其中L1和L2常用在单核中，L3则用在多核中。从L1到L3，三级缓存的大小依次增大，性能依次降低。CPU缓存命中率用于衡量CPU缓存的复用情况，命中率越高，则表示性能越好。

# 二、CPU性能分析工具

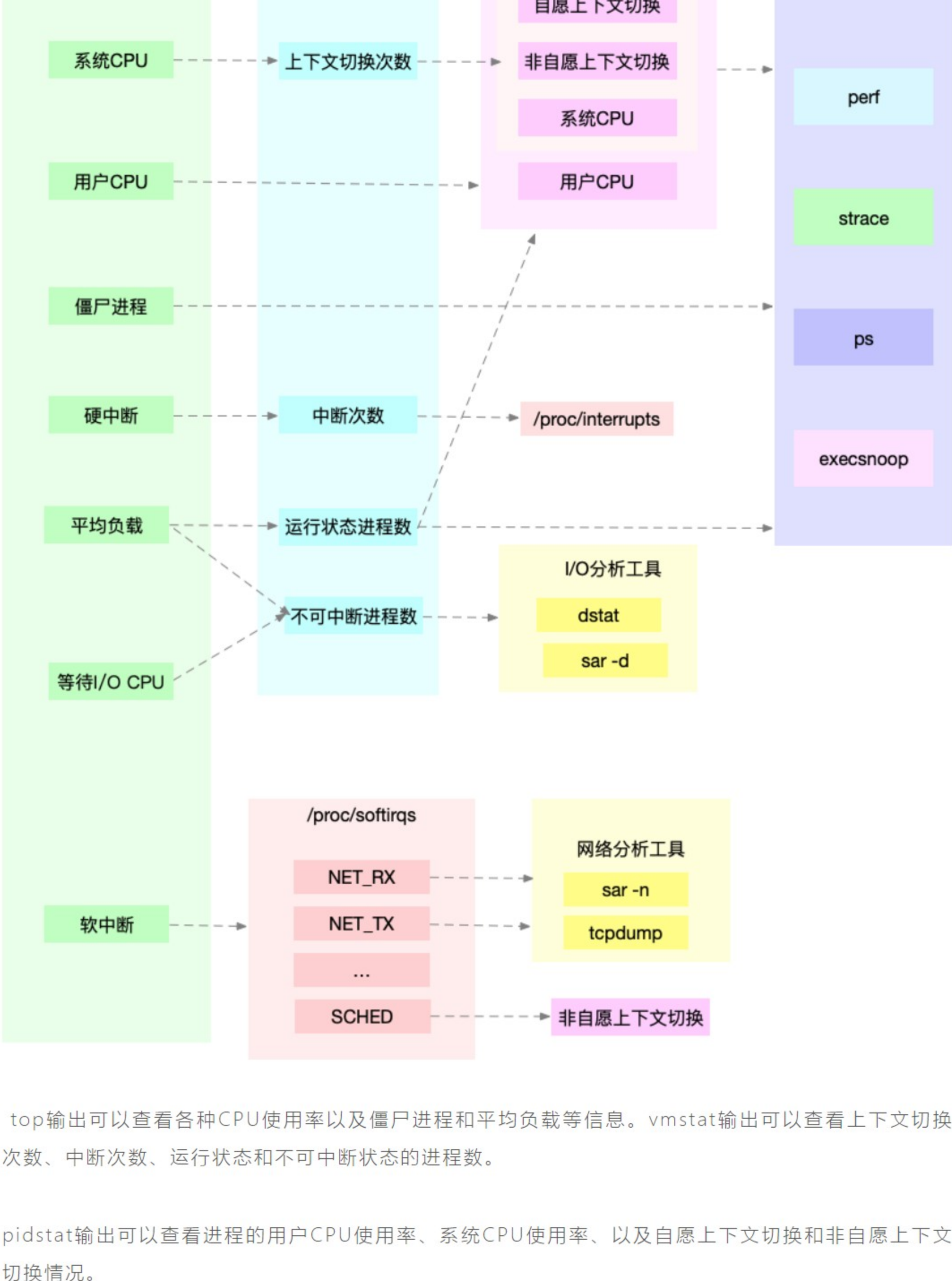
CPU性能指标查看可以使用工具：

根据指标找工具（CPU性能）		
性能指标	工具	说明
平均负载	uptime top	uptime最简单； top提供了更全的指标
系统整体CPU使用率	vmstat mpstat top sar /proc/stat	top、vmstat、mpstat 只可以动态查看， 而 sar 还可以记录历史数据  /proc/stat是其他性能工具的数据来源
进程CPU使用率	top pidstat ps htop atop	top和ps可以按CPU使用率给进程排序， 而pidstat只显示实际用了CPU的进程  htop和atop以不同颜色显示更直观
系统上下文切换	vmstat	除了上下文切换次数， 还提供运行状态和不可中断状态进程的数量
进程上下文切换	pidstat	注意加上 -w 选项
软中断	top /proc/softirqs mpstat	top提供软中断CPU使用率， 而/proc/softirqs和mpstat提供了各种软 中断在每个CPU上的运行次数
硬中断	vmstat /proc/interrupts	vmstat提供总的中断次数， 而/proc/interrupts提供各种中断在每个 CPU上运行的累积次数
网络	dstat sar tcpdump	dstat和sar提供总的网络接收和发送情况， 而tcpdump则是动态抓取正在进行的网络 通讯
I/O	dstat sar	dstat和sar都提供了I/O的整体情况
CPU 个数	/proc/cpuinfo lscpu	lscpu更直观
事件剖析	perf execsnoop	perf可以用来分析CPU的缓存以及内核调用 链，execsnoop用来监控短时进程

性能分析工具可以查看的CPU性能指标：

根据工具查指标（CPU性能）	
性能工具	CPU性能指标
uptime	平均负载
top	平均负载、运行队列、整体的CPU使用率以及每个进程的状态和CPU使用率
htop	top增强版，以不同颜色区分不同类型的进程，更直观
atop	CPU、内存、磁盘和网络等各种资源的全面监控
vmstat	系统整体的CPU使用率、上下文切换次数、中断次数，还包括处于运行和不可中断状态的进程数量
mpstat	每个CPU的使用率和软中断次数
pidstat	进程和线程的CPU使用率、中断上下文切换次数
/proc/softirqs	软中断类型和在每个CPU上的累积中断次数
/proc/interrupts	硬中断类型和在每个CPU上的累积中断次数
ps	每个进程的状态和CPU使用率
pstree	进程的父子关系
dstat	系统整体的CPU使用率
sar	系统整体的CPU使用率，包括可配置的历史数据
strace	进程的系统调用
perf	CPU性能事件剖析，如调用链分析、CPU缓存、CPU调度等
execsnoop	监控短时进程

CPU性能指标比较多，很多CPU性能指标间都有一定的关联。用户 CPU使用率高，应该排查进程的用户态而不是内核态。因为用户 CPU使用率反映用户态的CPU使用情况，而内核态的CPU使用情况只会反映到系统CPU使用率上。



top输出可以查看各种CPU使用率以及僵尸进程和平均负载等信息。vmstat输出可以查看上下文切换次数、中断次数、运行状态和不可中断状态的进程数。

pidstat输出可以查看进程的用户CPU使用率、系统CPU使用率、以及自愿上下文切换和非自愿上下文切换情况。

pidstat输出结果中的进程用户CPU使用率升高，会导致top输出的用户CPU使用率升高。当发现top输出的用户CPU使用率有问题时，可以跟pidstat的输出做对比，观察是否是某个进程导致的问题。而找出导致性能问题的进程后，使用进程分析工具来分析进程的行为，比如使用strace分析系统调用情况，使用perf分析调用链中各级函数的执行情况。

top输出结果中平均负载升高，可以跟vmstat输出的运行状态和不可中断状态的进程数做对比，观察是哪种进程导致的负载升高。如果不可中断进程数增多，那么就需要做IO分析，使用dstat或sar等工具，进一步分析IO情况。如果运行状态进程数增多，使用top和 pidstat，找出处于运行状态的到底是什么进程，然后再用进程分析工具，做进一步分析。

当发现top输出结构中软中断CPU使用率升高时，可以查看 /proc/softirqs文件中各种类型软中断的变化情况，确定到底是哪种软中断出的问题。如果是网络接收中断导致的问题，使用网络分析工具sar和tcpdump来分析。