

# Chapter 11

## 套路篇：如何迅速分析出 系統CPU的瓶頸在哪裡

Winnie 6/3/2020

# 困惑 ???

---

疑問: CPU 的性能指标那么多, CPU 性能分析工具也是一抓一大把, 如何套用到實工作場景?

# CPU 性能指標 --- CPU 使用率

---

- 用戶 CPU 使用率
- 系統 CPU 使用率
- 等待 I/O 的 CPU 使用率
- 軟中斷和硬中斷的 CPU 使用率
- 在虛擬化環境中會用到的竊取 CPU 使用率 (steal) 和客戶 CPU 使用率 (guest)

# CPU 性能指标 --- 平均负载

---

反应系统的整体负载情况

理想情况下，平均负载等于逻辑 CPU 个数

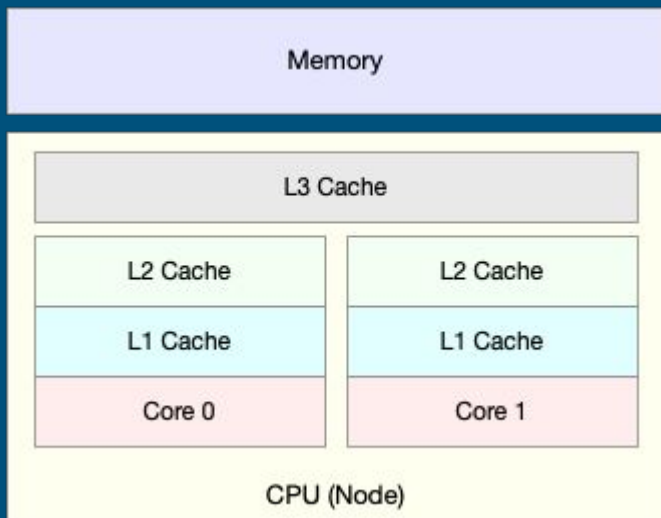
# CPU 性能指標 --- 進程上下文切換

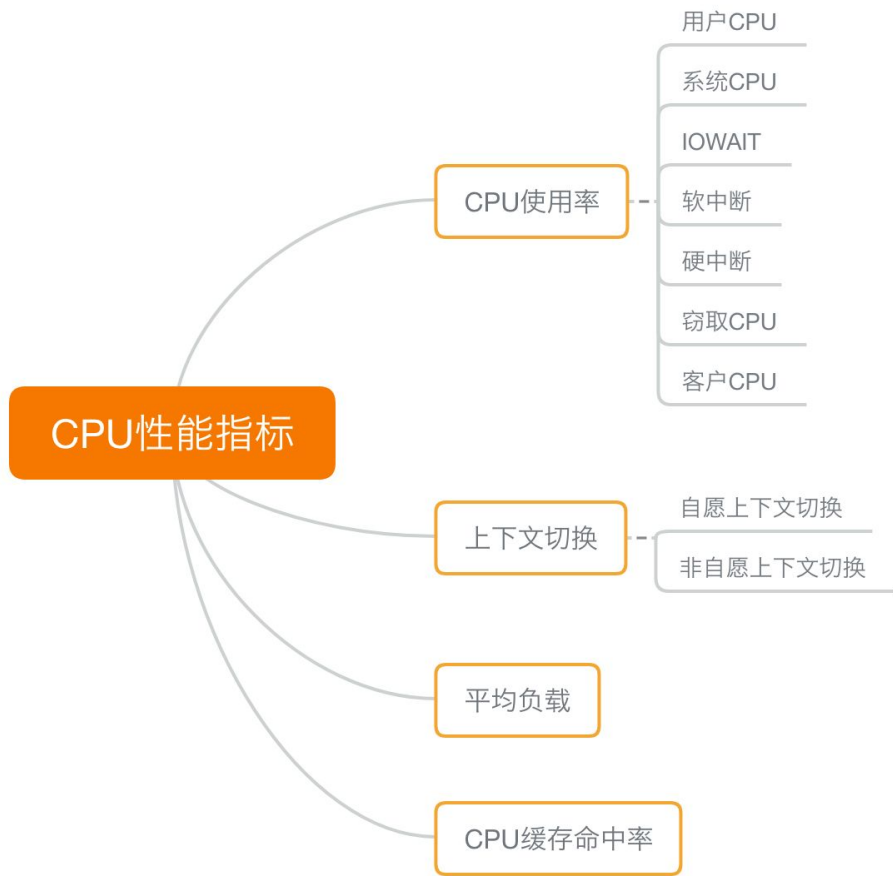
---

- 无法获取资源而导致的自愿上下文切换
- 被系统强制调度导致的非自愿上下文切换

# CPU 性能指標 --- CPU 緩存的命中率

---





# 性能工具

---

uptime

top, perf, sar

mpstat, pidstat, vmstat, dstat, /proc/softirqs

tcpdump

strace

stress, sysbench

execsnoop



活用學活

把性能指标和性能工具联系起来

---

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

## 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

如何迅速分析CPU的性能瓶頸？

---



# 思考題

---

我(很久)以前用過的工具

valgrind

gprof