

理解 Linux 的平均负载和性能监控

内容

在本文中，我们将解释 Linux 系统中最关键的管理任务之一——关于系统 / CPU 的 ^{load} 负载。

Load average
和 平均负载 的性能监控。

首先来看所有的类 UNIX 系统中两个重要的表述：

系统负载 / CPU 负载 – 衡量 Linux 系统的 CPU 过载或利用率低的指标，即处于运算状态或等待状态的 CPU 核心数。

平均负载 – 通过固定的时间周期如 1、5、15 分钟计算出的平均的系统负载。

Linux 中，平均负载一般指在内核运行队列中被标记为运行或不可打断状态的进程的平均数。

注意：

几乎没有 Linux 或类 Unix 系统不为用户展示平均负载的值。

完全空闲的 Linux 系统平均负载为 0，不包括空闲进程。

绝大多数类 Unix 系统只统计运行和等待状态的进程。但是在 Linux 中，平均负载也包括处于不可打断的睡眠状态的进程——它们是在等待其它系统资源如磁盘 I/O 等的进程。

如何监测 Linux 系统平均负载

有诸多方式监测系统平均负载，如 uptime，它会展示系统运行时间、用户数量及平均负载：

```
$ uptime
```

```
07:13:53 up 8 days, 19 min, 1 user, load average: 1.98, 2.15, 2.21
```

平均负载的数字从左到右的含义依次为：

最近 1 分钟的平均负载为 1.98

最近 5 分钟的平均负载为 2.15

最近 15 分钟的平均负载为 2.21

高平均负载意味着系统是过载的：许多进程在等待 CPU 时间。

下一节将介绍平均负载和 CPU 核数的关系。此外，常用的工具 top 和 glances 可以实时显示 Linux 系统的运行状态：

Top 命令

```
$ top
```

显示运行中的 Linux 进程：

```
top - 12:51:42 up 2:11, 1 user, load average: 1.22, 1.12, 1.26
```

```
Tasks: 243 total, 1 running, 242 sleeping, 0 stopped, 0 zombie
```

```
%Cpu(s): 17.4 us, 2.9 sy, 0.3 ni, 74.8 id, 4.6 wa, 0.0 hi, 0.0 si, 0.0 st
```

```
KiB Mem : 8069036 total, 388060 free, 4381184 used, 3299792 buff/cache
```

```
KiB Swap: 3906556 total, 3901876 free, 4680 used. 2807464 avail Mem
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
```

```
6265 tecmint 20 0 1244348 170680 83616 S 13.3 2.1 6:47.72 Headset
```

```
2301 tecmint 9 -11 640332 13344 9932 S 6.7 0.2 2:18.96 pulseaudio
```

```
2459 tecmint 20 0 1707692 315628 62992 S 6.7 3.9 6:55.45 cinnamon
```

```
2957 tecmint 20 0 2644644 1.035g 137968 S 6.7 13.5 50:11.13 firefox
```

```
3208 tecmint 20 0 507060 52136 33152 S 6.7 0.6 0:04.34 gnome-terminal-
```

```
3272 tecmint 20 0 1521380 391324 178348 S 6.7 4.8 6:21.01 chrome
```

```
6220 tecmint 20 0 1595392 106964 76836 S 6.7 1.3 3:31.94 Headset
```

```

1 root 20 0 120056 6204 3964 S 0.0 0.1 0:01.83 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:00.10 ksoftirqd/0
5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
....

```

Glances 工具

\$ glances

Glances – Linux 系统监测工具：

```

TecMint (LinuxMint 18 64bit / Linux 4.4.0-21-generic) Uptime: 2:16:06
CPU 16.4% nice: 0.1% LOAD 4-core MEM 60.5% active: 4.90G SWAP 0.1%
user: 10.2% irq: 0.0% 1 min: 1.20 total: 7.70G inactive: 2.07G total: 3.73G
system: 3.4% iowait: 2.7% 5 min: 1.16 used: 4.66G buffers: 242M used: 4.57M
idle: 83.6% steal: 0.0% 15 min: 1.24 free: 3.04G cached: 2.58G free: 3.72G
NETWORK Rx/s Tx/s TASKS 253 (883 thr), 1 run, 252 slp, 0 oth sorted automatically by
cpu_percent, flat view
enp1s0 525Kb 31Kb
lo 2Kb 2Kb CPU% MEM% VIRT RES PID USER NI S TIME+ IOR/s IOW/s Command
wlp2s0 0b 0b 14.6 13.3 2.53G 1.03G 2957 tecmint 0 S 51:49.10 0 40K /usr/lib/firefox/firefox
7.4 2.2 1.16G 176M 6265 tecmint 0 S 7:08.18 0 0 /usr/lib/Headset/Headset --type=renderer
--no-sandbox --primordial-pipe-token=879B36514C6BEDB183D3E4142774D1DF --lan
DISK I/O R/s W/s 4.9 3.9 1.63G 310M 2459 tecmint 0 R 7:12.18 0 0 cinnamon --replace
ram0 0 0 4.2 0.2 625M 13.0M 2301 tecmint -11 S 2:29.72 0 0 /usr/bin/pulseaudio --start --
log-target=syslog
ram1 0 0 4.2 1.3 1.52G 105M 6220 tecmint 0 S 3:42.64 0 0 /usr/lib/Headset/Headset
ram10 0 0 2.9 0.8 409M 66.7M 6240 tecmint 0 S 2:40.44 0 0 /usr/lib/Headset/Headset --
type=gpu-process --no-sandbox --supports-dual-gpus=false --gpu-driver-bug-
workarounds=7,2
ram11 0 0 2.9 1.8 531M 142M 1690 root 0 S 6:03.79 0 0 /usr/lib/xorg/Xorg :0 -audit 0 -auth
/var/lib/mdm/:0.Xauth -nolisten tcp vt8
ram12 0 0 2.6 0.3 79.3M 23.8M 9651 tecmint 0 R 0:00.71 0 0 /usr/bin/python3
/usr/bin/glances
ram13 0 0 1.6 4.8 1.45G 382M 3272 tecmint 0 S 6:25.30 0 4K /opt/google/chrome/chrome
...

```

这些工具中的平均负载是从 /proc/loadavg 文件中读取的，也可以直接使用 cat 命令查看：

\$ cat /proc/loadavg

2.48 1.69 1.42 5/889 10570

想要图形样式监测平均负载，请戳：ttyload – 终端中颜色编码图形显示 Linux 平均负载。

在桌面计算机中，可以使用图形用户接口工具查看系统平均负载。

理解系统平均负载和 CPU 核心数的关系

考虑了 CPU 核心数的影响，才能解释系统负载。

多处理器 Vs 多核处理器

多处理器 – 一个计算机系统中集成两个或多个物理 CPU

多核处理器 – 单个物理 CPU 有两个或多个单独的核并行工作（也叫处理单元）。双核意味着有两个处理单元，4 核有 4 个处理单元，以此类推。

此外，Intel 引入了超线程技术用来提高并行计算能力。

通过超线程技术，在操作系统中，单个物理 CPU 表现的和两个逻辑 CPU 一样。（实际在硬件上只有一个 CPU）。

注意，单个 CPU 核同一时间只能执行一个任务，于是产生了多 CPU/处理器、多核 CPU，以及多线程技术。

多 CPU 时，多个程序可以同时执行。如今的 Intel CPU 使用了多核心和超线程技术。

可以使用 `nproc` 或 `lscpu` 命令查看系统中的处理器单元数量。

```
$ nproc
```

```
4
```

或者

```
lscpu
```

也可以使用 `grep` 命令：

```
$ grep 'model name' /proc/cpuinfo | wc -l
```

```
4
```

为了进一步理解系统负载，需要做一些假设。假设系统负载如下：

```
23:16:49 up 10:49, 5 user, load average: 1.00, 0.40, 3.35
```

在单核系统中意味着：

CPU 被充分利用（100%）；最近的 1 分钟有 1 个进程在运行。

CPU 有 60% 处于空闲状态；在最近的 5 分钟没有进程等待 CPU 时间。

CPU 平均过载了 235%；最近的 15 分钟平均有 2.35 个进程在等待 CPU 时间。

在双核系统中意味着：

有一个 CPU 处于完全空闲状态，另一个 CPU 被使用；最近的 1 分钟没有进程等待 CPU 时间。

CPU 平均 160% 处于空闲状态；最近的 5 分钟没有进程等待 CPU 时间。

CPU 平均过载了 135%；最近的 15 分钟有 1.35 个进程等待 CPU 时间。

也许你还会喜欢：

20 个监控系统性能的命令行工具（一）

13 个 Linux 性能监控工具（二）

Perf：一个 Linux 上的性能监控分析工具

使用 Nmon 监控 Linux 的系统性能

总而言之，如果你是系统管理员，你应该关注高的平均负载。平均负载高于 CPU 核心数意味着需要增加 CPU，反之则意味着 CPU 未被充分利用。