

20 常用性能分析工具

常用性能分析工具

CPU、内存、网络、磁盘是影响计算机性能的主要硬件，当程序或者系统出现问题时，应该根据需求对这四个硬件进行排查。

在Linux系统中，很多高效的性能分析工具可以帮助分析定位问题。

常用的性能分析工具包括vmstat、sar、iostat、netstat、free、ps、top等，其中sar、iostat、dstat、iotop、netstat、ifconfig、iftop等工具需要额外安装。

```
yum install -y net-tools sysstat iotop dstat iftop
```

命令	CPU	内存	磁盘	进程	网络连接	安装方式
top	√	√		√		
vmstat	√	√				
ps				√		
free		√				
df			√			
ip					√	
ss					√	
sar	√	√	√	√		CentOS源sysstat包
iostat	√		√			CentOS源sysstat包
dstat	√	√	√		√	CentOS源dstat包
iotop			√			CentOS源iotop包
netstat					√	CentOS源net-tools包
ifconfig					√	CentOS源net-tools包
iftop					√	EPEL源iftop包

重点掌握

- 命令功能：注意对比类似命令间的差异
- 常用参数

一、综合性能分析工具

1.1 top命令——综合性能监控

`top` 命令是性能监控工具，可以在很多Linux/UNIX版本下使用。它也是 Linux运维人员经常使用的监控系统性能的工具。

`top` 命令可以定期显示（动态）CPU使用、内存使用、交换内存、缓存大小、缓冲区大小、进程、用户等内容，并将它们更新到列表中。

`top` 命令便于用户对Linux系统进行监控。

`top` 命令的语法格式为：`top [选项]`

`top` 命令选项如下表所示。

参数	说明
d	指定每两次屏幕信息刷新之间的时间间隔，也可以使用s交互命令来改变。
P	通过指定监控进程ID来仅仅监控某个进程的状态。
q	该选项将使top没有任何延迟的进行刷新，如果调用程序有超级用户权限，那么top将以尽可能高的优先级运行。
S	指定累计模式。
s	使top命令在安全模式中运行，避免交互式。
i	使top命令不显示任何闲置或僵死进程。
c	显示整个命令而不只是显示命令名。

```

1 [root@Shell ~]# top
2 top - 20:41:41 up 39 min,  1 user,  load average: 0.09, 0.12, 0.13
3 Tasks: 370 total,  3 running, 367 sleeping,  0 stopped,  0 zombie
4 %Cpu(s):  1.4 us,  1.7 sy,  0.0 ni, 96.9 id,  0.0 wa,  0.0 hi,  0.0 si,  0
.0 st
5 KiB Mem : 1868660 total, 1347588 free,  272828 used,  248244 buff/cach
e
6 KiB Swap: 2097148 total, 2097148 free,          0 used. 1425156 avail Mem
7
8      PID USER      PR  NI   VIRT   RES   SHR S %CPU %MEM    TIME+  COMMAND
9      2842 mysql    20   0 1174548 114948  8768 S  1.7  6.2   0:10.30 mysqld
10       1 root     20   0  46340   6800  4056 S  0.3  0.4   0:02.78 systemd
11      643 root     20   0      0      0      0 R  0.3  0.0   0:00.91 xfsail
d/dm-0
12      712 root     20   0  38996   5912  5596 S  0.3  0.3   0:01.10 systemd-journal
13      850 root     20   0 293688   6244  5464 S  0.3  0.3   0:00.28 rsyslogd
14     16828 root     20   0 148436   2292  1432 R  0.3  0.1   0:00.07 top
15       2 root     20   0      0      0      0 S  0.0  0.0   0:00.02 kthrea
dd
16       3 root     20   0      0      0      0 S  0.0  0.0   0:00.01 ksofti
rqd/0
17       5 root      0 -20      0      0      0 S  0.0  0.0   0:00.00 kworke
r/0:0H
18       6 root     20   0      0      0      0 S  0.0  0.0   0:00.23 kworke
r/u256:0
19       7 root     rt    0      0      0      0 S  0.0  0.0   0:00.00 migrat
ion/0
20       8 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcu_bh
21       9 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
0
22      10 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
1
23      11 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
2
24      12 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
3
25      13 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
4
26      14 root     20   0      0      0      0 S  0.0  0.0   0:00.00 rcuob/
5
27

```

28	6	15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/
29	7	16	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/
30	8	17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/
31	9	18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/
32	10	19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/

`top` 命令结果的字段解释如表所示。

字段	含义
VIRT	虚拟内存（virtual memory usage），进程“需要的”虚拟内存大小，包括进程使用的库、代码、数据等。例如：进程申请100m的内存，但实际只使用了10m，那么它会增长100m，而不是实际的使用量。
RES	常驻内存（resident memory usage），进程当前使用的内存大小，统计加载的库文件所占内存大小，但不包括swap out，包含其他进程的共享。例如：如果申请100m的内存，实际使用10m，它只增长10m，与VIRT相反。
SHR	共享内存（shared memory）除了自身进程的共享内存，也包括其他进程的共享内存。
DATA	数据占用的内存，如果top没有显示，按f键可以显示出来，是真正在运行中要使用的，真正的该程序要求的数据空间。
PID	（Process ID进程标志号。
PPID	父进程id。
RUSER	Real user name。
UID	进程所有者的用户id。
GROUP	进程所有者的组名。
TTY	启动进程的终端名，不是从终端启动的进程则显示为？
PR	进程的优先级。
USER	进程所有者用户名。
NI	进程优先级别nice值，负值表示高优先级，正值表示低优先级。
P	最后使用的CPU，仅在多CPU环境下有意义。
SWAP	进程使用的虚拟内存中，被换出的大小，单位kb。
CODE	可执行代码以外占用的物理内存大小，单位kb。

nFLT	页面错误次数。
nDRT	最后一次写入到现在，被修改过的页面数。
WCHAN	若该进程在睡眠，则显示睡眠中的系统函数名。
Flags	任务标志。
%CPU	进程占用的CPU使用率。
%MEM	进程占用的物理内存和总内存的百分比。
COMMAND	进程启动执行的命令。
TIME+	进程所有的CPU时间总计。
S	进程状态。
D=	不可中断的睡眠状态。
R=	运行
S=	睡眠
T=	跟踪/停止
Z	僵尸进程

默认情况下仅显示比较重要的PID、USER、PR、NI、VIRT、RES、SHR、S、%CPU、%MEM、TIME+、COMMAND列。可以通过下面的快捷键来更改显示内容。

(1) 按 **o** 键可以改变列的显示顺序，按大写的A-Z可以将相应的列向左移动，小写的a-z可以将相应的列向右移动，最后按回车键确定。

(2) 按f键可以选择显示的内容，按f键之后会显示出列的列表，按 **a-z** 即可显示或隐藏对应的列，最后按回车键确定。

(3) 按大写的 **F** 或 **O** 键，按 **a-z** 可以将进程按照相应的列进行排序，而大写的R键可以将当前的排序倒转。

top 命令运行中，可使用快捷键对进程的显示方式进程控制。

top 命令的快捷键如表所示。

命令	说明
s	改变画面更新频率
l	关闭或开启第一部分第一行top信息的表示

t	关闭或开启第一部分第二行Tasks和第三行Cpus信息的表示
m	关闭或开启第一部分第四行Mem和第五行Swap信息的表示
N	以PID的大小的顺序排列表示进程列表
P	以CPU占用率大小的顺序排列进程列表
M	以内存占用率大小的顺序排列进程列表
h	显示帮助
n	设置在进程列表所显示进程的数量
q	退出top
s	改变画面更新周期

`top` 命令是Linux上进行**系统监控的首选命令**，但 `top` 命令有很大的局限性，有时候却达不到的要求，通常用 `ps` 和 `netstat` 两个命令来补充 `top` 的不足。

1.2 vmstat命令——系统监控工具

`vmstat` 命令是常见的Linux/UNIX监控工具，可以通过给定时间间隔来展示服务器的状态，包括**CPU使用率、内存使用率、虚拟内存交换情况、IO读写情况**。

`vmstat` 命令格式为：`vmstat [选项] [采样时间间隔 [采样次数]]`

`vmstat` 命令的参数有以下三种情况。

- 无参数：表示采样1次。
- 1个参数：参数表示采样时间间隔，采样次数无限，执行 `CTRL+C` 中断。
- 2个参数：第一个参数 `采样时间间隔` 的单位是秒，第二个参数为 `采样次数`。

```

1 [root@Shell ~]# vmstat
2 procs -----memory----- --swap-- -----io---- -system-- -----cp
  u-----
3  r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs us sy i
  d wa st
4  1  0       0 742208    948 139652    0    0    2    0   12   15  0  0 10
  0  0  0
5 [root@Shell ~]# vmstat 5
6 procs -----memory----- --swap-- -----io---- -system-- -----cp
  u-----
7  r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs us sy i
  d wa st
8  2  0       0 742208    948 139652    0    0    2    0   12   15  0  0 10
  0  0  0
9  0  0       0 742192    948 139652    0    0    0    0   13   13  0  0 10
  0  0  0
10 0  0       0 742192    948 139652    0    0    0    0   12   14  0  0 10
  0  0  0
11 ^C
12 [root@Shell ~]# vmstat 5 2
13 procs -----memory----- --swap-- -----io---- -system-- -----cp
  u-----
14 r  b   swpd   free   buff  cache   si   so   bi   bo   in   cs us sy i
  d wa st
15 2  0       0 742224    948 139652    0    0    2    0   12   15  0  0 10
  0  0  0
16 0  0       0 742208    948 139652    0    0    0    0   12   12  0  0 10
  0  0  0

```

vmstat 命令结果的字段解释如表所示。

参数	描述
r	等待运行的进程数，多少个进程分到了cpu，一般不超过cpu个数是正常的值。
b	处于非中断睡眠状态的进程数，即在等待资源分配的进程数，阻塞状态。
swpd	虚拟内存已使用的大小（KB），如果大于0，表示机器的物理内存不足，如果不是程序内存泄漏的原因，那么就要升级内存或者把消耗内存的任务迁移到其他机器。
free	空闲的物理内存的大小（KB）。
buff	用作缓存的内存数，缓存的是文件目录基本内容，在磁盘中的位置。权限等（KB）。
cache	用作文件缓存的内存数，对打开的文件做缓存，提高执行效率和使用性能（KB）。
si	从磁盘交换到内存的交换页数量，即每秒使用的虚拟内存数量（KB/s）。

so	从内存交换到磁盘的交换页数量（KB/s）。
bi	发送到块设备一般即磁盘的块数（块/s）。
bo	从块设备接收到的块数（块/s）。
in	每秒中断次数，包括时钟中断。
cs	每秒的上下文切换次数。
us	用户cpu使用时间。
sy	系统cpu使用时间，如进行I/O操作等。
id	空闲时间。
wt	等待I/O的cpu时间，一般为0。

相比 `top` 命令，通过 `vmstat` 命令，用户可以看到**系统整体** CPU、内存、I/O的使用情况，而不是单单看到各个进程的CPU使用率和内存使用率，两者使用的场景不一样。

1.3 sar命令——系统活动报告

`sar` (System Activity Reporter，系统活动情况报告)是Linux全面的系统性能分析工具之一，可以从多方面对系统活动进行报告，包括**文件的读写情况、系统调用的使用情况、磁盘I/O、CPU效率、内存使用状况、进程活动及IPC 有关活动**等，可以连续对系统取样，获得大量的取样数据。取样数据和分析的结果都**可以存入文件**，而且所需的负载很小。

`sar` 命令的语法格式为：`sar [选项] [采样时间间隔] [采样次数]`

`sar` 命令与 `vmstat` 在使用方法上有一定相似性。区别在于

- **`vmstat` 一次性显示多个统计量**
- **`sar` 每次只显示一类统计量**，默认显示**cpu利用率统计**，如果显示其他类别统计量，需要指定选项。


```

1  # 默认统计CPU利用率
2  [root@Shell ~]# sar
3  Linux 3.10.0-327.el7.x86_64 (Shell)      02/07/2022      _x86_64_
   (1 CPU)
4
5  11:10:01 PM      CPU      %user      %nice      %system      %iowait      %steal
   %idle
6  11:20:01 PM      all        0.01        0.00        0.09        0.00        0.00
   99.89
7  11:30:01 PM      all        0.01        0.00        0.09        0.01        0.00
   99.90
8  Average:         all        0.01        0.00        0.09        0.01        0.00
   99.90
9  [root@Shell ~]# sar -u
10 Linux 3.10.0-327.el7.x86_64 (Shell)      02/07/2022      _x86_64_
   (1 CPU)
11
12 11:10:01 PM      CPU      %user      %nice      %system      %iowait      %steal
   %idle
13 11:20:01 PM      all        0.01        0.00        0.09        0.00        0.00
   99.89
14 11:30:01 PM      all        0.01        0.00        0.09        0.01        0.00
   99.90
15 Average:         all        0.01        0.00        0.09        0.01        0.00
   99.90
16 # -r统计内存
17 [root@Shell ~]# sar -r
18 Linux 3.10.0-327.el7.x86_64 (Shell)      02/07/2022      _x86_64_
   (1 CPU)
19
20 11:10:01 PM kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit
   %commit kbactive  kbinact  kbdirty
21 11:20:01 PM    668668    332664    33.22        948    160244    246228
   7.95    138428    81952        0
22 11:30:01 PM    668712    332620    33.22        948    160280    246228
   7.95    138480    81940        0
23 Average:         668690    332642    33.22        948    160262    246228
   7.95    138454    81946        0
24 # -d统计块设备
25 [root@Shell ~]# sar -d
26 Linux 3.10.0-327.el7.x86_64 (Shell)      02/07/2022      _x86_64_
   (1 CPU)
27
28 11:10:01 PM      DEV          tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz
   await      svctm      %util

```

29	11:20:01 PM	dev8-0	0.07	0.00	0.51	7.58	0.00
			1.32	0.80	0.01		
30	11:20:01 PM	dev253-0	0.07	0.00	0.51	6.89	0.00
			1.27	0.73	0.01		
31	11:20:01 PM	dev253-1	0.00	0.00	0.00	0.00	0.00
			0.00	0.00	0.00		
32	11:30:01 PM	dev8-0	0.03	0.11	0.25	11.37	0.00
			2.11	1.95	0.01		
33	11:30:01 PM	dev253-0	0.04	0.11	0.25	9.82	0.00
			1.91	1.68	0.01		
34	11:30:01 PM	dev253-1	0.00	0.00	0.00	0.00	0.00
			0.00	0.00	0.00		
35	Average:	dev8-0	0.05	0.05	0.38	8.80	0.00
			1.58	1.17	0.01		
36	Average:	dev253-0	0.06	0.05	0.38	7.86	0.00
			1.48	1.05	0.01		
37	Average:	dev253-1	0.00	0.00	0.00	0.00	0.00
			0.00	0.00	0.00		

sar 命令参数详解如表所示。

参数	描述
-A	所有报告的总和。
-u	输出cpu使用情况的统计信息。
-v	输出inode，文件和其他内核表的统计信息。
-d	输出每一个块设备的活动信息。
-f	输出内存和交换空间的统计信息。
-b	显示I/O和传送速率的统计信息。
-a	文件读写情况。
-c	输出进程统计信息，每秒创建的进程数。
-R	输出内存页面的统计信息。
-y	终端设备活动情况。
-w	输出系统交换活动信息。
-g	输出串口的使用情况。
-h	输出关于buffer使用的统计数据。
-m	输出IPC消息队列和信号量的使用情况。

-n	输出命令cache的使用情况。
-q	输出运行队列和交换队列的平均长度。
-r	输出没有使用的内存页面和硬盘块。
-y	输出TTY设备活动状况。
-B	输出附加的缓存的使用情况。
-p	输出调页活动的使用情况。

1.4 iostat 命令——监控CPU的使用情况和输入输出设备统计信息

iostat 用于报告CPU统计信息和整个系统、适配器、tty设备、磁盘和 CD-ROM的输入/输出统计信息。

默认显示CPU利用统计信息和磁盘IO统计信息，其中CPU利用统计信息与 **vmstat** 命令相同。

iostat 命令的格式如下：**iostat** [选项] [采样时间间隔 [采样次数]]

iostat 命令选项详解如下表所示。

参数	说明
-c	显示CPU使用情况
-d	显示磁盘使用情况
-k	以KB为单位显示
-m	以M为单位显示
-N	显示磁盘陈列（LVM）信息
-n	显示NFS使用情况
-p	显示磁盘和分区的情况
-t	显示终端和CPU的信息
-x	显示详细信息
-V	显示版本信息

```

1  # 默认显示CPU利用统计信息和磁盘I/O统计信息
2  [root@Shell ~]# iostat
3  Linux 3.10.0-327.el7.x86_64 (Shell)      02/09/2022      _x86_64_
   (1 CPU)
4
5  avg-cpu:  %user   %nice %system %iowait  %steal   %idle
6             0.08    0.00    0.27    0.10    0.00   99.55
7
8  Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
9  sda                 0.94         10.51          1.28       91949       11241
10 # 显示CPU利用统计
11 [root@Shell ~]# iostat -c
12 Linux 3.10.0-327.el7.x86_64 (Shell)      02/09/2022      _x86_64_
   (1 CPU)
13
14 avg-cpu:  %user   %nice %system %iowait  %steal   %idle
15             0.08    0.00    0.27    0.10    0.00   99.55
16 # 显示磁盘I/O统计信息
17 [root@Shell ~]# iostat -d
18 Linux 3.10.0-327.el7.x86_64 (Shell)      02/09/2022      _x86_64_
   (1 CPU)
19
20 Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
21 sda                 0.94         10.49          1.28       91949       11241
22 # 显示磁盘I/O统计信息, 以MB为单位
23 [root@Shell ~]# iostat -dm
24 Linux 3.10.0-327.el7.x86_64 (Shell)      02/09/2022      _x86_64_
   (1 CPU)
25
26 Device:            tps    MB_read/s    MB_wrtn/s    MB_read    MB_wrtn
27 sda                 0.94          0.01          0.00         89         10
28 # 设置采样时间和次数
29 [root@Shell ~]# iostat 2 1
30 Linux 3.10.0-327.el7.x86_64 (Shell)      02/09/2022      _x86_64_
   (1 CPU)
31
32 avg-cpu:  %user   %nice %system %iowait  %steal   %idle
33             0.07    0.00    0.27    0.10    0.00   99.55
34
35 Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
36 sda                 0.94         10.48          1.28       91949       11241
37

```

1.5 dstat命令——动态显示系统负载

dstat 命令工具默认情况下会**动态**显示 **CPU、disk、net、page、system 负载情况**，每秒收集1次。

dstat 命令的格式：**dstat [-afv] [选项] [采样时间间隔 [采样次数]]**

dstat 命令选项详解如表所示。

参数	含义
-c	--cpu, 统计CPU状态, 包括user、system、idle（空闲等待时间百分比）、wait（等到磁盘IO）、hardware interrupt（硬件中断）、software interrupt（软件中断）等
-d	--disk 统计磁盘读写状态
-D total	sda统计指定磁盘或汇总信息
-l	--load统计系统负载情况, 包括1分钟、5分钟、15分钟平均值
-m	--mem统计系统物理内存使用情况, 包括used、buffers、cache、free
-s	--swap统计已使用和剩余量
-n	-net统计网络使用情况, 包括接收和发送数据
-N eth1,total	统计eth1接口汇总流量
-r	--io 统计I/O请求, 包括读写请求
-p	--proc统计进程信息, 包括runnable、uninterruptible、new
-y	--sys统计系统信息, 包括中断、上下文切换
-t	显示统计时间, 对分析历史数据非常有用
--fs	统计文件打开数和inodes数
-a	此为默认选项, 等同于-cdngy
--ipc	IPC状态（消息队列、信号、共享内存）
--lock	文件锁状态（posix、flock、read、write）
--raw	原始套接字信息
--socket	套接字信息（所有的、tcp、udp、原始的、IP片段的）
--tcp	tcp状态（listen、established、syn、time_wait、close）
--udp	udp状态（listen、active）
--unix	Unix接口状态（datagram、stream、listen、active）
--vm	虚拟内存信息（hard、pagefaults、softpagefaults、allocated、free）硬页面错误、较页面错误、分配的、未分配的

delay	两次输出之间的时间间隔，默认是1s
count	报告输出的次数，默认是没有限制，一直输出知道ctrl+c

▼

Bash | 复制代码

```
1  #默认显示CPU、磁盘IO、网络IO等统计信息
2  [root@Shell ~]# dstat
3  You did not select any stats, using -cdngy by default.
4  ----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
5  usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int  csw
6    0   0 100   0   0   0|8453B 1071B|   0    0 |   0   0 |  17   24
7    0   1  99   0   0   0|   0    0 | 60B  378B|   0   0 |  39   62
8    0   0 100   0   0   0|   0    0 | 60B  170B|   0   0 |  32   52
9    0   0 100   0   0   0|   0    0 | 60B  170B|   0   0 |  34   49
10 ^C
11 #dstat也支持采用时间间隔和采样次数
12 [root@Shell ~]# dstat 2 1
13 You did not select any stats, using -cdngy by default.
14 ----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
15 usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int  csw
16    0   0 100   0   0   0|8415B 1068B|   0    0 |   0   0 |  17   25
17    0   0 100   0   0   0|   0    0 | 60B  219B|   0   0 |  31   51
18 #dstat可通过参数指定统计类型
19 [root@Shell ~]# dstat -c 2 1
20 ----total-cpu-usage----
21 usr sys idl wai hiq siq
22    0   0 100   0   0   0
23    0   0 100   0   0   0
```

`dstat` 命令显示结果含义如下。

参数	含义
usr	用户进程消耗的CPU时间百分比，usr的值比较高时，说明用户进程消耗的CPU时间多，但是如果长期超过50%的使用，那么就要考虑优化程序进行加速。
sys	内核进程消耗的CPU时间百分比，sys的值高时，说明系统内核消耗的CPU资源多，
idl	CPU处在空闲状态时间百分比
wai	IO等待消耗的CPU时间百分比，wai的值高时，说明IO等待比较严重，这可能由于磁盘大量作随机访问造成，也可能是磁盘的带宽出现瓶颈。
hiq	硬中断
siq	软中断

read	磁盘读操作
writ	磁盘写操作数
recv	接受请求数
send	发送请求数
in	每秒产生的中断次数
out	系统分页
int	系统中断次数
csw	每秒上下文切换次数
分页统计	系统的分页活动。分页指的是一种内存管理技术用于查找系统场景，一个较大的分页表明系统正在使用大量的交换空间，或者说内存非常分散，大多少情况下希望看到page in(换入)和page out(换出)的值是0
r	运行的和等待（CPU时间片）运行的进程数
b	处于不可中断状态的进程数，常见的情况是由IO引起的
swpd	切换到交换内存上的内存（默认以KB为单位）如果swpd的值不为0，但si，so的值长期为0，也不会影响系统性能
free	空闲的物理内存
buff	作为buffer cache的内存，对块设备的读写进行缓冲
cache	作为page cache的内存，文件系统的cache。如果cache的值大的时候，说明cache中文件数多，如果频繁访问到的文件都能被cache中，那么磁盘的读IO bi会非常小
si	交换内存使用，由磁盘调入内存。内存够用的使用，si和so值为0，如果值长期大于0时，系统性能会受到影响，磁盘IO和CPU资源都会被消耗。
so	交换内存使用，由内存调入磁盘
bi	从块设备读入的数据总量（读磁盘）（KB/s）
bo	写入到块设备的数据总量（写磁盘）（KB/s）

二、单项性能分析工具

2.1 free命令 —— 显示系统内存情况

`free` 命令能够显示系统中**物理上的空闲内存、已用内存**，还有**交换内存**，同时也能显示被内核使用的**缓冲和缓存**。这些信息是通过解析文件 `/proc/meminfo` 而收集到的。

`free` 命令的格式为 `free [选项]`。

`free` 命令的选项如下。

参数	含义
-h	以人类可读的方式输出统计结果
-k	以KB为单位输出统计结果
-m	以MB为单位输出统计结果
-g	以GB为单位输出统计结果
-t	使用该选项会多显示一行标题为Total的统计信息，该行统计的是（used、free、total、的总和）此Total与total不同
-o	禁止显示第二行的缓冲区调整值（-/+buffers/cache）
-s	每个多少秒自动刷新结果
-c	与-s配合使用，控制刷新结果次数
-l	显示高低内存的统计详情
-a	显示可用内存
-V	显示版本号


```

1  # free命令默认以KB形式显示系统内存情况
2  [root@Shell ~]# free
3      total          used          free      shared  buff/cache   avai
4  Mem:      1001332    116672      669332       6784     215328      7
5  Swap:      2097148         0      2097148
6  [root@Shell ~]# free -k
7      total          used          free      shared  buff/cache   avai
8  Mem:      1001332    116652      669332       6784     215348      7
9  Swap:      2097148         0      2097148
10 # free命令如果指定不太恰当的单位，可能导致信息不容易观察
11 [root@Shell ~]# free -g
12      total          used          free      shared  buff/cache   avai
13 Mem:           0           0           0           0           0
14 Swap:           1           0           1
15 # 推荐使用-h选项以方便阅读的方式显示内容内存信息
16 [root@Shell ~]# free -h
17      total          used          free      shared  buff/cache   avai
18 Mem:           977M        113M        653M         6.6M        210M
19 Swap:          2.0G         0B         2.0G

```

输出项目的说明信息如下。

选项	含义
total	内存总数
used	已经使用的内存数
free	空闲的内存数，等于total-used-buff/cache
shared	多个进程共享的内存总额
buffers/cache	磁盘缓存的大小
available	可用的内存数

2.2 ps命令——进程监控工具

`top` 命令是对进程**实时监控**的命令。`ps` 命令**静态**显示进程的状态。`ps` 命令可以确定**进程运行的状态、进程是否结束、有没有僵尸进程、哪些进程占用了过多的资源等**。`ps` 命令是**最常用的后台进程监控工具**。

`ps` 命令的语法格式如下: `ps [选项]`

`ps` 命令的选项的详细信息如下。

参数	说明
-a	显示同一终端下的所有程序
-A	显示所有进程 (等价于-e)
-w	显示加宽可以显示较多的信息
-au	显示较详细的信息
aux	显示所有包含其他使用者的进程
-d	显示所有进程, 但省略所有的会话引线
-e	等于“-A”
-f	全部列出, 通常和其他选项联用, 如: <code>ps -fa</code> or <code>ps -fx</code>
f	显示程序间的关系 (树形结构, <code>pstree</code>)
-x	显示没有控制终端的进程, 同时显示各个命令的具体路径, <code>dx</code> 不可合用
-N	反向选择
r	显示当前终端的进程
T	显示当前终端的所有程序
u	指定用户的所有进程
-t	指定终端编号。并列出于该终端机的程序的状况
-p	pid父进程id
-u	uid or username 选择有效的用户id或者是用户名
-g	gid or groupname 显示组的所有进程
-L	参数, 后面加上特定的PID显示特定进程的线程
-l	以长格式显示进程信息

对进程进行监测和控制 `ps aux` 命令或者 `ps -ef` 命令可以获得**终端上所有用户有关进程的信息**, 这个也是平时用的最多命令之一。

由于进程信息较多，因此通常与 `grep` 命令配合使用。

▼Shell | 复制代码

```
1 [root@Shell ~]# ps -ef | grep ssh
2 root      1468      1  0 Feb07 ?        00:00:00 /usr/sbin/sshd -D
3 root      2611    1468  0 Feb07 ?        00:00:00 sshd: root@pts/0
4 root      2851    2615  0 00:13 pts/0    00:00:00 grep --color=auto ssh
```

`ps`命令输出的类别信息如下。

参数	含义
%CPU	进程的cpu占有率
%MEM	进程的内存占有率
RSS	进程使用的驻留集大小或者实际内存的大小
TTY	与进程关联的终端
STAT	检查的状态
R	running正在运行或准备运行
S	sleeping 睡眠，休眠中，受阻，在等待某个条件的形成或接受到信号。
I	idle空闲
Z	僵死（zombie）进程已终止，但进程描述符存在，直到父进程调用wait4()系统调用释放
D	不可中断的睡眠。通常是I/O，收到信号不唤醒和不可运行，进程必须等待直到有中断发生
P	等待交换页
W	换出，表示当前页面不在内存
N	低优先级任务
T	terminate 终止，进程收到SIGSTOP，SIGSTP，SIGTIN，SIGTOU信号后停止运行
STRT	进程启动时间和日期
TIME	进程使用的总cpu时间
COMMAND	正在执行的命令行命令
NI	（nice）优先级
PRI	进程优先级编号

PPID	父进程的进程ID
SID	会话ID (session id)
WCHAN	进程正在睡眠的内核函数，该函数的名称是从/root/system.map文件中获得的
FLAGS	与进程相关的数字标识
UID	用户ID (User ID)
X	死掉的进程
<	高优先级进程，高优先序的进程
L	内存锁页 (Lock) 有记忆体分页分配并缩在记忆体内
s	进程的领导者 (在它之下有子进程)
+	位于后台的进程组
l	多进程的 (使用CLONE_THREAD, 类似NPTL pthreads)

2.3 df命令——查看磁盘占用

df 命令是用来检查Linux服务器的**磁盘空间占用情况**，用该命令获取硬盘被占用了多少空间，还剩余多少空间等信息。

df 命令的语法格式为：**df** [选项] [文件]

如果没有指定文件名，则显示所有当前被挂载的文件系统的可用空间。

默认情况下，磁盘空间将以**KB**为单位进行显示。

当指定环境变量 POSIXLY_CORRECT时，将以512字节为单位进行显示。

df 命令常见参数详解如表所示。

参数	含义
-a	全部文件系统列表
-B	--block-size指定单位大小，如1k、1m等
-h	以人类易读格式显示，如GB、MB、KB等
-H	和“-h”一样，但计算式为：1k=1000，而不是1k=1024
-i	显示inode信息
-k	区块为1024字节，以KB的容量显示各文件系统，相当于--block-size=1k
-l	只显示本地文件系统

-m	区块为1048576字节，以KB的容量显示各文件系统，相当于—block-size=1m
--no-sync	忽略sync命令
-P	输出格式为POSIX
--sync	在取得磁盘信息前，先执行sync命令
-T	文件系统类型
--block-size	指定区块大小
-t<文件系统类型>	只显示选定文件系统的磁盘信息
-x<文件系统类型>	不显示选定文件系统的磁盘信息
--help	显示帮助信息
--version	显示版本信息

```

1  #类似于free, -h选项以人类易读形式显示容量(size)
2  [root@Shell ~]# df -h
3  Filesystem                Size      Used Avail Use% Mounted on
4  /dev/mapper/centos-root    18G       989M    17G   6% /
5  devtmpfs                   479M       0     479M   0% /dev
6  tmpfs                       489M       0     489M   0% /dev/shm
7  tmpfs                       489M   6.7M    483M   2% /run
8  tmpfs                       489M       0     489M   0% /sys/fs/cgroup
9  /dev/sda1                   497M    125M    373M  25% /boot
10 tmpfs                       98M       0      98M   0% /run/user/0
11 # 显示文件系统类型
12 [root@Shell ~]# df -Th
13 Filesystem                Type              Size      Used Avail Use% Mounted on
14 /dev/mapper/centos-root    xfs                18G       1.7G    16G  10% /
15 devtmpfs                   devtmpfs           903M       0     903M   0% /dev
16 tmpfs                       tmpfs              913M       0     913M   0% /dev/shm
17 tmpfs                       tmpfs              913M   8.7M    904M   1% /run
18 tmpfs                       tmpfs              913M       0     913M   0% /sys/fs/cgroup
19 /dev/sda1                   xfs                497M    125M    373M  26% /boot
20 tmpfs                       tmpfs              183M       0     183M   0% /run/user/0
21 #指定容量(size)显示单位大小
22 [root@Shell ~]# df -B1G
23 Filesystem                1G-blocks    Used Available Use% Mounted on
24 /dev/mapper/centos-root          18         1         17    6% /
25 devtmpfs                          1         0         1    0% /dev
26 tmpfs                             1         0         1    0% /dev/shm
27 tmpfs                             1         1         1    2% /run
28 tmpfs                             1         0         1    0% /sys/fs/cgroup
29 /dev/sda1                         1         1         1   25% /boot
30 tmpfs                             1         0         1    0% /run/user/0
31 [root@Shell ~]# df -B10G
32 Filesystem                10G-blocks    Used Available Use% Mounted on
33 /dev/mapper/centos-root           2         1         2    6% /
34 devtmpfs                         1         0         1    0% /dev
35 tmpfs                            1         0         1    0% /dev/shm
36 tmpfs                            1         1         1    2% /run
37 tmpfs                            1         0         1    0% /sys/fs/cgroup
38 /dev/sda1                       1         1         1   25% /boot
39 tmpfs                            1         0         1    0% /run/user/0

```

2.4 iotop命令——监视磁盘I/O状态

`iotop` 命令是一个**监视磁盘I/O使用情况的** **top** 类工具。

`iotop` 具有与 `top` 相似的UI, 其中包括PID、用户、I/O、进程等相关信息。

Linux下的IO统计工具如 `iostat` 等大多数是只能统计到每个设备的读写情况，若想确定哪个进程产生了I/O，就需要利用 `iotop` 工具进行查看。

`iotop` 命令可以查看I/O统计信息排序，追踪到具体的进程，显示当前进程或者线程的使用率。

`iotop` 命令的语法格式为：`iotop [选项]`

`iotop` 命令参数详解如表所示。

参数	含义
-v	--version显示版本号
-h	--help显示帮助用法
-o	--only只显示正在产生I/O的进程活线程，除了传参，可以在运行过程中按o生效
-b	--batch非交互模式，一般用来记录日志
-n NUM	--iter=NUM设置监测的次数，默认无限，一般在非交互式下使用
-d SEC	--delay=SEC 设置每次监测的间隔，默认1秒
-p PID	--pid=PID 指定监测的进程或线程
-u USER	--user=USER 指定监测某个用户产生的I/O
-p	--process仅显示进程，默认iotop显示所有线程
-a	--accumulated 显示累积的I/O，而不是带宽
-k	--kilobytes使用KB单位，在非交互式模式下，脚本编程有用
-t	--time 加上时间戳，非交互式模式下
-q	--quiet禁止头几行，非交互式模式，有三种指定方式。-q表示只在第一次监测时显示列名，-qq永远不显示列名，-qqq永远不显示I/O汇总。

`iotop` 常用的快捷键如表所示。

参数	含义
左右箭头	改变排序方式，默认是IO排序
r	改变排列顺序
o	只显示有IO输出的进程
p	进程或线程的显示方式的切换
a	显示累积使用量
q	退出

```

1 [root@Shell ~]# iotop
2 Total DISK READ :      0.00 B/s | Total DISK WRITE :      0.00 B/s
3 Actual DISK READ:      0.00 B/s | Actual DISK WRITE:      0.00 B/s
4 TID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO>   COMMAND

5  1 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % systemd --switched
   -root --system --deserialize 21
6  2 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [kthreadd]
7  3 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [ksoftirqd/0]
8  6 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [kworker/u256:0]
9  7 rt/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [migration/0]
10 8 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [rcu_bh]
11 # -o参数只显示IO操作进程
12 [root@Shell ~]# iotop -o
13 Total DISK READ :      0.00 B/s | Total DISK WRITE :      0.00 B/s
14 Actual DISK READ:      0.00 B/s | Actual DISK WRITE:      0.00 B/s
15 TID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO>   COMMAND

16 3193 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.01 % [kworker/0:2]
17 # -u参数显示root用户的IO进程
18 [root@Shell ~]# iotop -u root
19 3193 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.01 % [kworker/0:2]
20 Total DISK READ :      0.00 B/s | Total DISK WRITE :      0.00 B/s
21 Actual DISK READ:      0.00 B/s | Actual DISK WRITE:      0.00 B/s
22 TID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN      IO>   COMMAND

23 3193 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.06 % [kworker/0:2]
24  1 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % systemd --switched
   -root --system --deserialize 21
25  2 be/4 root          0.00 B/s    0.00 B/s  0.00 %    0.00 % [kthreadd]

```

2.5 netstat命令——显示网络状态

`netstat` 命令用于显示各种网络相关信息，如网络连接，路由表，接口状态 (Interface Statistics) 等等。

`netstat` 命令的格式为： `netstat [选项]`

`netstat` 命令选项详解如表所示。

选项	含义
-a	显示所有socket，包括正在监听的。
-c	每隔1秒就重新显示一遍，直到用户中断它。
-i	显示所有网络接口的信息。格式“netstat -i”。
-n	以网络IP地址代替名称，显示出网络连接情形。
-r	显示核心路由表，格式同“route -e”。
-t	显示TCP协议的连接情况。
-u	显示UDP协议的连接情况。
-v	显示正在进行的工作。
-p	显示建立相关连接的程序名和PID。
-b	显示在创建每个连接或侦听端口时涉及的可执行程序。
-e	显示以太网统计。此选项可以与-s选项结合使用。
-f	显示外部地址的完全限定域名（FQDN）。
-o	显示与网络计时器相关的信息。
-s	显示每个协议的统计。
-x	显示NetworkDirect连接、侦听器 and 共享端点。
-y	显示所有连接的TCP连接模板。无法与其他选项结合使用。
interval	重新显示选定的统计，各个显示间暂停的间隔秒数。按CTRL+C停止重新显示统计。如果省略，则netstat将打印当前的配置信息一次。

案例：

```

1  # 显示网络接口列表
2  [root@Shell ~]# netstat -i
3  Kernel Interface table
4  Iface          MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DR
P TX-OVR Flg
5  eno16777736    1500     20982      0      0 0          7108      0
0      0 BMRU
6  lo             65536      0      0      0 0          0      0
0      0 LRU
7  #
8  [root@Shell ~]# netstat -an
9  Active Internet connections (servers and established)
10 Proto Recv-Q Send-Q Local Address           Foreign Address         Stat
11 tcp      0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
12 tcp      0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
13 tcp      0      52 192.168.149.3:22        192.168.149.1:21594     ESTABLISHED
14 tcp6     0      0 :::22                   :::*                     LISTEN
15 tcp6     0      0 :::1:25                  :::*                     LISTEN
16 Active UNIX domain sockets (servers and established)
17 Proto RefCnt Flags   Type       State       I-Node     Path
18 unix  2      [ ACC ] STREAM    LISTENING   13588      /run/lvm/lvmpo
lld.socket
19 unix  2      [ ]     DGRAM     8218        /run/systemd/n
otify
20 unix  2      [ ACC ] STREAM    LISTENING   8236        /run/systemd/j
ournal/stdout
21 unix  5      [ ]     DGRAM     8239        /run/systemd/j
ournal/socket
22 unix  2      [ ACC ] STREAM    LISTENING   17712       /var/run/Netwo
rkManager/private
23 unix  2      [ ACC ] STREAM    LISTENING   16172       /var/run/dbus/
system_bus_socket
24 unix  11     [ ]     DGRAM     8241        /dev/log
25 unix  2      [ ACC ] STREAM    LISTENING   20452       public/cleanup
26 unix  2      [ ACC ] STREAM    LISTENING   20455       public/qmgr
27 unix  2      [ ACC ] STREAM    LISTENING   20473       private/rewrit
e
28 unix  2      [ ACC ] STREAM    LISTENING   20476       private/bounce
29 unix  2      [ ACC ] STREAM    LISTENING   20479       private/defer

```

30	unix	2	[ACC]	STREAM	LISTENING	20482	private/trace
31	unix	2	[ACC]	STREAM	LISTENING	20488	public/flush
32	unix	2	[ACC]	STREAM	LISTENING	20485	private/verify
33	unix	2	[ACC]	STREAM	LISTENING	20503	public/showq
34	unix	2	[ACC]	STREAM	LISTENING	20491	private/proxym
35	ap						
	unix	2	[ACC]	STREAM	LISTENING	20494	private/proxyw
36	rite						
	unix	2	[ACC]	STREAM	LISTENING	20497	private/smtp
37	unix	2	[ACC]	STREAM	LISTENING	20459	private/tlsmgr
38	unix	2	[ACC]	STREAM	LISTENING	20500	private/relay
39	unix	2	[ACC]	STREAM	LISTENING	20506	private/error
40	unix	2	[ACC]	STREAM	LISTENING	20509	private/retry
41	unix	2	[ACC]	STREAM	LISTENING	20512	private/discar
42	d						
	unix	2	[ACC]	STREAM	LISTENING	20515	private/local
43	unix	2	[ACC]	STREAM	LISTENING	20518	private/virtua
44	l						
	unix	2	[ACC]	STREAM	LISTENING	20521	private/lmtp
45	unix	2	[ACC]	STREAM	LISTENING	20524	private/anvil
46	unix	2	[ACC]	STREAM	LISTENING	20527	private/scache
47	unix	2	[ACC]	STREAM	LISTENING	20448	public/pickup
48	unix	2	[ACC]	SEQPACKET	LISTENING	13732	/run/udev/cont
49	rol						
	unix	2	[ACC]	STREAM	LISTENING	13530	/run/systemd/p
50	private						
	unix	2	[ACC]	STREAM	LISTENING	13544	/run/lvm/lvmet
51	ad.socket						
	unix	2	[]	DGRAM		13814	/run/systemd/s
52	hutdownd						
	unix	3	[]	STREAM	CONNECTED	17788	
53	unix	3	[]	STREAM	CONNECTED	14565	
54	unix	3	[]	STREAM	CONNECTED	20520	
55	unix	3	[]	STREAM	CONNECTED	16305	
56	unix	3	[]	STREAM	CONNECTED	20516	
57	unix	3	[]	STREAM	CONNECTED	20499	
58	unix	3	[]	STREAM	CONNECTED	20454	
59	unix	3	[]	STREAM	CONNECTED	20067	/var/run/dbus/
60	system_bus_socket						
	unix	3	[]	STREAM	CONNECTED	20484	
61	unix	3	[]	STREAM	CONNECTED	16304	
62	unix	3	[]	STREAM	CONNECTED	20495	
63	unix	3	[]	STREAM	CONNECTED	17374	
64	unix	3	[]	STREAM	CONNECTED	18046	/var/run/dbus/
65	system_bus_socket						
	unix	2	[]	DGRAM		25911	
66	unix	3	[]	STREAM	CONNECTED	20480	
67	unix	3	[]	STREAM	CONNECTED	20508	

68	unix 3 []	STREAM	CONNECTED	16306	/var/run/dbus/
69	system_bus_socket				
70	unix 3 []	STREAM	CONNECTED	20472	
71	unix 2 []	DGRAM		14107	
72	unix 3 []	STREAM	CONNECTED	20447	
73	unix 2 []	DGRAM		21129	
74	unix 3 []	STREAM	CONNECTED	20519	
75	unix 3 []	STREAM	CONNECTED	20450	
	unix 3 []	STREAM	CONNECTED	17375	/run/systemd/j
	ournal/stdout				
76	unix 3 []	STREAM	CONNECTED	20523	
77	unix 3 []	STREAM	CONNECTED	20504	
78	unix 3 []	STREAM	CONNECTED	20487	
79	unix 3 []	STREAM	CONNECTED	14566	/run/systemd/j
	ournal/stdout				
80	unix 3 []	STREAM	CONNECTED	19032	
81	unix 3 []	STREAM	CONNECTED	16607	/run/systemd/j
	ournal/stdout				
82	unix 3 []	STREAM	CONNECTED	20453	
83	unix 3 []	STREAM	CONNECTED	20483	
84	unix 3 []	DGRAM		14627	
85	unix 3 []	STREAM	CONNECTED	19178	
86	unix 3 []	STREAM	CONNECTED	18042	
87	unix 3 []	STREAM	CONNECTED	20528	
88	unix 3 []	STREAM	CONNECTED	20511	
89	unix 3 []	STREAM	CONNECTED	19179	/run/systemd/j
	ournal/stdout				
90	unix 3 []	STREAM	CONNECTED	20471	
91	unix 3 []	STREAM	CONNECTED	17789	/var/run/dbus/
	system_bus_socket				
92	unix 3 []	STREAM	CONNECTED	20496	
93	unix 3 []	STREAM	CONNECTED	18062	
94	unix 3 []	STREAM	CONNECTED	14277	
95	unix 3 []	DGRAM		14626	
96	unix 3 []	STREAM	CONNECTED	20446	
97	unix 3 []	STREAM	CONNECTED	20507	
98	unix 3 []	STREAM	CONNECTED	20492	
99	unix 3 []	STREAM	CONNECTED	20475	
100	unix 2 []	DGRAM		16866	
101	unix 3 []	STREAM	CONNECTED	16606	
102	unix 3 []	STREAM	CONNECTED	20449	
103	unix 3 []	STREAM	CONNECTED	20486	
104	unix 3 []	STREAM	CONNECTED	20505	
105	unix 3 []	STREAM	CONNECTED	17714	
106	unix 3 []	STREAM	CONNECTED	16237	
107	unix 3 []	STREAM	CONNECTED	20522	
108	unix 3 []	STREAM	CONNECTED	16880	/var/run/dbus/
	system_bus_socket				

109	unix 3 []	STREAM	CONNECTED	19033	/run/systemd/j
	ournal/stdout				
110	unix 2 []	DGRAM		20630	
111	unix 2 []	DGRAM		17674	
112	unix 2 []	DGRAM		17916	
113	unix 3 []	STREAM	CONNECTED	17316	
114	unix 3 []	STREAM	CONNECTED	20501	
115	unix 3 []	STREAM	CONNECTED	20529	
116	unix 3 []	STREAM	CONNECTED	18043	/run/systemd/j
	ournal/stdout				
117	unix 3 []	STREAM	CONNECTED	20478	
118	unix 3 []	STREAM	CONNECTED	20514	
119	unix 3 []	STREAM	CONNECTED	14278	/run/systemd/j
	ournal/stdout				
120	unix 2 []	DGRAM		20364	
121	unix 3 []	STREAM	CONNECTED	18063	/var/run/dbus/
	system_bus_socket				
122	unix 3 []	STREAM	CONNECTED	20525	
123	unix 3 []	STREAM	CONNECTED	20474	
124	unix 3 []	STREAM	CONNECTED	20510	
125	unix 3 []	STREAM	CONNECTED	20457	
126	unix 2 []	DGRAM		16963	
127	unix 3 []	STREAM	CONNECTED	20493	
128	unix 3 []	STREAM	CONNECTED	20502	
129	unix 3 []	STREAM	CONNECTED	16238	/run/systemd/j
	ournal/stdout				
130	unix 3 []	STREAM	CONNECTED	20489	
131	unix 3 []	STREAM	CONNECTED	17715	/var/run/dbus/
	system_bus_socket				
132	unix 3 []	STREAM	CONNECTED	20498	
133	unix 3 []	STREAM	CONNECTED	20481	
134	unix 3 []	STREAM	CONNECTED	16779	/run/systemd/j
	ournal/stdout				
135	unix 3 []	STREAM	CONNECTED	16194	
136	unix 3 []	STREAM	CONNECTED	20066	
137	unix 3 []	STREAM	CONNECTED	20517	
138	unix 3 []	STREAM	CONNECTED	20477	
139	unix 3 []	STREAM	CONNECTED	16879	
140	unix 3 []	STREAM	CONNECTED	18045	
141	unix 2 []	DGRAM		14587	
142	unix 3 []	STREAM	CONNECTED	20513	
143	unix 3 []	STREAM	CONNECTED	20456	
144	unix 3 []	STREAM	CONNECTED	20490	
145	unix 3 []	STREAM	CONNECTED	17317	/var/run/dbus/
	system_bus_socket				
146	unix 2 []	DGRAM		18096	
147	unix 2 []	DGRAM		16080	
148	unix 3 []	STREAM	CONNECTED	20526	

```

149 unix 3 [ ] STREAM CONNECTED 16778
150 #列出所有监听tcp服务
151 [root@Shell ~]# netstat -tlnp
152 Active Internet connections (only servers)
153 Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
154 tcp        0      0 0.0.0.0:ssh              0.0.0.0:*               LISTEN      16778/sshd

```

2.6 ss命令——显示网络状态

查看服务器连接数一般都使用 `netstat` 命令。`ss` 命令的优势在于它能够显示更多、更详细关于TCP和连接状态的信息，而且比 `netstat` 更快速、更高效。

`ss` 是Socket Statistics的缩写。顾名思义，`ss` 命令可以用来获取socket统计数据，它可以显示PACKET套接字、TCP套接字、UDP套接字、DCCP套接字、RAW套接字、UNIX域套接字等的统计信息，而且允许显示和netstat 类似的内容。

`ss` 命令的语法格式为：`ss [选项] [过滤]`

`ss` 命令选项详解如表所示：

选项	含义
-h	--help 帮助信息
-V	--version 程序版本信息
-n	--numeric 不解析服务名称
-r	--resolve 解析主机名
-a	--all 显示所有套接字 (sockets)
-l	--listening 显示监听状态的套接字 (sockets)
-o	--options 显示计数器信息
-e	--extended 显示详细的套接字 (sockets) 信息
-m	--memory 显示套接字 (sockets) 的内存使用情况
-p	--processes 显示使用套接字 (sockets) 的进程
-i	--info 显示TCP内部信息
-s	--summary 显示套接字 (sockets) 使用概况
-4	--ipv4 仅显示IPv4的套接字 (sockets)
-6	--ipv6 仅显示IPv6的套接字 (sockets)
-0	--packet 显示PACKET套接字 (sockets)

-t	--tcp 仅显示 TCP套接字 (sockets)
-u	--udp 仅显示UDP套接字 (sockets)
-d	--dccp 仅显示 DCCP套接字 (sockets)
-w	--raw 仅显示 RAW套接字 (sockets)
-x	--unix 仅显示 Unix套接字 (sockets)
-f	--family=FAMILY 显示 FAMILY类型的套接字 (sockets) , FAMILY可选, 支持unix, inet, inet6, link, netlink
-A	--query=QUERY, --socket=QUERY 查看某种类型 QUERY : ={all inet tcp udp raw unix packet netlink}[,QUERY]
-D	--diag=FILE 将原始TCP套接字 (sockets) 信息存储到文件
-F	--filter=FILE 使用此参数指定的过滤规则文件, 过滤某种状态的连接FILTER :=[stat TCP-STATE][EXPRESSION]

```

1  #-a 显示本地打开的所有端口
2  [root@Shell ~]# ss -a
3  Netid State      Recv-Q Send-Q           Local Address      Peer Address
   s:Port
4  nl      UNCONN      0      0
   rtnl:-1304427624
   *
5  nl      UNCONN      0      0
   rtnl:-721419368
   *
6  nl      UNCONN      0      0
   rtnl:kernel
   *
7  #-at显示所有tcp socket
8  [root@Shell ~]# ss -at
9  State      Recv-Q Send-Q           Local Address      Peer Address
   ss:Port
   s:Port
10 LISTEN      0      128
   *:ssh
   *:
11 LISTEN      0      100
   0.1:smtp
   *:
12 ESTAB       0      52
   9.3:ssh
   1:21594
   192.168.14
   192.168.149.
13 LISTEN      0      128
   :::ssh
   :::*
14 LISTEN      0      100
   ::1:smtp
   :::*
15 [root@Shell ~]# ss -tpl
16 State      Recv-Q Send-Q           Local Address      Peer Address
   ss:Port
   s:Port
17 LISTEN      0      128
   *:ssh
   *:
   users:(("sshd",pid=1479,fd=3))
18 LISTEN      0      100
   0.1:smtp
   *:
   users:(("master",pid=2492,fd=13))
19

```



```
20  LISTEN      0      128
    :::ssh
    :::*
    users: (("sshd",pid=1479,fd=4))
    LISTEN      0      100
21  :::1:smtp
    :::*
    users: (("master",pid=2492,fd=14))
22  [root@Shell ~]# ss -nltp
    State      Recv-Q  Send-Q
23  ress:Port
    ss:Port
    LISTEN      0      128
24  *:22
    *:~
    users: (("sshd",pid=1479,fd=3))
    LISTEN      0      100
25  0.0.1:25
    *:~
    users: (("master",pid=2492,fd=13))
    LISTEN      0      128
26  :::22
    :::*
    users: (("sshd",pid=1479,fd=4))
    LISTEN      0      100
    :::1:25
    :::*
```

2.7 ifconfig命令——显示或设置网络设备信息

ifconfig 命令的英文全称是network interfaces configuring，即用于配置和显示Linux内核中网络接口的网络参数。

用 ifconfig 命令配置的网卡信息，在网卡重启后机器重启后，配置就不存在了。

要想将上述的配置信息永远的存的电脑里，那就要修改网卡的配置文件了。

ifconfig 命令的格式为：ifconfig [网络设备] [参数]

Ifconfig 命令参数详解如表所示。

参数	含义
up	启动指定网络设备/网卡
down	关闭指定网络设备/网卡。该参数可以有效地阻止通过指定接口的IP信息流，如果想永久地关闭一个接口，还需要从核心路由表中将该接口的路由信息全部删除
arp	设置指定网卡是否支持ARP协议
-promisc	设置是否支持网卡的promiscuous模式，如果选择此参数，网卡将接收网络中发给它所有的数据包
-allmulti	设置是否支持多播模式，如果选择此参数，网卡将接收网络中所有的多播数据包
-a	显示全部接口信息

-s	显示摘要信息（类似于netstat -i）
add	给指定网卡配置IPv6地址
del	删除指定网卡的IPv6地址
mtu<字节数>	设置网卡的最大的传输单元（bytes）
netmask<子网掩码>	设置网卡的子网掩码，掩码可以是有前缀0x的32位十六进制数，也可以是用点分开的4个十进制数，如果不打算将网络分成子网，可以不管这一选项，如果要使用子网，那么网络中每一个系统必须有相同子网掩码
tunnel	建立隧道
dstaddr	设定一个远端地址，建立点对点通信
-broadcast <地址>	为指定网卡设置广播协议
-pointtopoint <地址>	为网卡设置点对点通讯协议
multicast	为网卡设置组播标志
address	为网卡设置IPv4地址
txqueuelen<长度>	为网卡设置传输队列的长度

▼

Bash | 复制代码

```

1 [root@Shell ~]# ifconfig
2 eno16777736: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
3     inet 192.168.149.3 netmask 255.255.255.0 broadcast 192.168.149.2
4     55
5     inet6 fe80::20c:29ff:fed6:98a0 prefixlen 64 scopeid 0x20<link>
6     ether 00:0c:29:d6:98:a0 txqueuelen 1000 (Ethernet)
7     RX packets 20864 bytes 23888880 (22.7 MiB)
8     RX errors 0 dropped 0 overruns 0 frame 0
9     TX packets 7096 bytes 1098486 (1.0 MiB)
10    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
11 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
12     inet 127.0.0.1 netmask 255.0.0.0
13     inet6 ::1 prefixlen 128 scopeid 0x10<host>
14     loop txqueuelen 0 (Local Loopback)
15     RX packets 0 bytes 0 (0.0 B)
16     RX errors 0 dropped 0 overruns 0 frame 0
17     TX packets 0 bytes 0 (0.0 B)
18     TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

ifconfig 如果不接受任何参数，就会输出当前网络接口的情况。

lo是表示主机的回环地址，它一般是用于测试一个网络程序，但又不想让局域网或外网的用户查看，只能在这台主机上运行和查看所用的网络接口。

`ifconfig` 输出结果每行分别表示为。

- 第一行:连接类型:Ethernet(以太网)HWaddr(硬件MAC地址)。
- 第二行:网卡的IP地址、子网、掩码。
- 第三行:UP(表示网卡开启状态)，RUNNING表示网卡的网线被接上，MULTICAST表示支持组播，MTU:1500表示最大的传输单元为1500字节。
- 第四、五行:接收、发送数据包情况统计。
- 第六、七行:接收、发送数据字节数统计信息。

2.8 iftop命令——实时流量监控工具

`iftop` 是一款实时流量监控工具、监控TCP/IP连接、显示端口、反向解析IP地址等，但 `iftop` 有个缺点是**必须以 `root` 身份才能运行**，且没有报表功能，

注意：`iftop`命令需要通过epel源安装，`yum -y install iftop`。

`iftop` 命令格式为：`iftop [选项]`

`iftop` 命令参数详解如表所示。

参数	含义
-i	设定监测的网卡，例如： <code>iftop -i eth1</code>
-B	以bytes为单位显示流量（默认是bits） ，例如： <code>iftop -B</code>
-n	使host信息默认显示IP：例如： <code>iftop -n</code> ，不进行DNS解析
-N	使端口信息默认显示端口号，例如： <code>iftop -N</code> ，不显示服务名称
-F	显示特定网段的进出流量，例如： <code>iftop -F 10.10.1.0/24</code> 或 <code>iftop -F 10.10.1.0/255.255.255.0</code>
-h	显示参数帮助信息
-p	使用这个参数后，中间的列表显示本地主机信息，出现了本机以外的IP信息
-b	使流量图形条默认显示
-f	过滤包
-P	默认显示host信息及端口信息
-m	设置界面最上边的刻度的最大值，刻度分五个大段显示，例如： <code>iftop -m 100M</code>

```

1  #默认监控第一块网卡的流量
2  [root@Shell ~]# iftop
3  interface: eno16777736
4  IP address is: 192.168.149.3
5  MAC address is: 00:0c:29:d6:98:a0
6      12.5Kb      25.0Kb      37.5Kb      50.0Kb
7  62.5Kb
8  +-----+-----+-----+-----+
9  bogon => bogon 0.98Kb 1.41Kb 1
10 .66Kb
11 <= 184b 221b
12 245b
13 bogon => cache1-ny 0b 0b
14 145b
15 <= 0b 0b
16 185b
17 -----
18 TX: cum: 2.70KB peak: 3.75Kb rates: 0.98Kb 1.41Kb 1
19 .80Kb
20 RX: 646B 1.45Kb 184b 221b
21 431b
22 TOTAL: 3.33KB 5.20Kb 1.16Kb 1.62Kb 2
23 .22Kb
24 #-n选项直接显示IP, 不进行DNS反解析
25 [root@Shell ~]# iftop -n
26 interface: eno16777736
27 IP address is: 192.168.149.3
28 MAC address is: 00:0c:29:d6:98:a0
29      12.5Kb      25.0Kb      37.5Kb      50.0Kb
30 62.5Kb
31 +-----+-----+-----+-----+
32 192.168.149.3 => 192.168.149.1 1.05Kb 1.70Kb 1
33 .70Kb
34 <= 184b 276b
35 276b
36 -----
37 TX: cum: 1.70KB peak: 2.91Kb rates: 1.05Kb 1.70Kb 1
38 .70Kb
39 RX: 276B 368b 184b 276b
40 276b

```

```
29 TOTAL: 1.96Kb 3.27Kb 1.23Kb 1.96Kb 1
30 .96Kb
31 #-i选项指定监控某个接口的流量
32 [root@Shell ~]# iftop -i eno16777736
33 interface: eno16777736
34 IP address is: 192.168.149.3
35 MAC address is: 00:0c:29:d6:98:a0
36 12.5Kb 25.0Kb 37.5Kb 50.0Kb
37 62.5Kb
38 +-----+-----+-----+-----+
39 bogon => bogon 2.59Kb 2.75Kb 2
40 .75Kb
41
42 <= 368b 368b
43 bogon => cache1-ny 0b 434b
44 434b
45 <= 0b 556b
46 556b
47 -----
48 TX: cum: 1.59KB peak: 3.75Kb rates: 2.59Kb 3.17Kb 3
49 .17Kb
50 RX: 462B 1.45Kb 368b 924b
51 924b
52 TOTAL: 2.04KB 5.20Kb 2.95Kb 4.08Kb 4
53 .08Kb
```

`iftop` 界面操作命令详解如表所示。

命令	含义
h	按h切换是否显示帮助
n	按n切换显示本机的IP或主机名
s	按s切换是否显示本机的host信息
d	按d切换显示远端目标主机的host信息
t	按t切换显示格式为2行/1行/只显示发送流量/只显示接收流量
N	按N切换显示端口或端口服务名称
S	按S切换是否显示本机的端口信息
D	按D切换是否显示远端目标主机的端口信息

p	按p切换是否显示端口信息
b	按b切换是否显示平均流量图形条
B	按B切换计算2秒或10秒或40秒内的平均流量
T	按T切换是否显示每个连接的总流量
l	按l打开屏幕过滤功能，输入要过滤的字符，比如ip，按回车后，屏幕就只显示这个IP相关的流量信息
L	按L切换显示画面上边的刻度，刻度不同，流量图形条会有变化
j或k	按j或按k可以向上或向下滚动屏幕显示的连接记录
1或2或3	按1或2或3可以根据右侧显示的三列流量数据进行排序
<	按<根据左边的本机名或IP排序
>	按>根据远端目标主机的主机名或IP排序
o	按o切换是否固定只显示当前的连接
f	按f可以编辑过滤代码，
!	按!可以使用shell命令
q	按q退出监控

小结

- 综合性能分析工具
 - top
 - vmstat
 - sar
 - iostat
 - dstat
- 单项性能分析工具
 - free -h
 - ps aux | grep
 - df -Th
 - iotop
 - netstat -nltp
 - ss -an
 - ifconfig
 - iftop

课程目标

- 知识目标：熟练掌握常用性能分析工具的应用。
- 技能目标：能够利用性能分析工具查找系统性能瓶颈。

课外拓展

- 进一步了解性能分析工具的应用场景

参考资料

- 《Linux系统命令及Shell脚本实践指南》，王军，机械工业出版社
- 《跟老男孩学Linux运维：Shell编程实战》，老男孩，机械工业出版社