proc文件系统

proc文件系统是一种无存储的文件系统，当读其中的文件时，其内容动态生成，当写文件时，文件所关联的写函数被调用。每个proc文件都关联着字节特定的读写函数，因而它提供了另外的一种和内核通信的机制：内核部件可以通过该文件系统向用户空间提供接口来提供查询信息、修改软件行为，因而它是一种比较重要的特殊文件系统。

由于proc文件系统以文件的形式向用户空间提供了访问接口，这些接口可以用于在运行时获取相关部件的信息或者修改部件的行为，因而它是非常方便的一个接口。内核中大量使用了该文件系统。proc文件系统就是一个文件系统，它可以挂载在目录树的任意位置，不过通常挂载在/proc下，它大致包含了如下信息：

- 内存管理
- 每个进程的相关信息
- 文件系统
- 设备驱动程序
- 系统总线
- 电源管理
- 终端
- 系统控制参数
- 网络

主机上的各个进程都是以目录的形态存在于/proc当中，例如systemd进程

```
[root@cloud001 ~]# ll /proc/1
total 0
dr-xr-xr-x. 2 root root 0 May 22 06:58 attr
-rw-r--r--. 1 root root 0 May 22 06:58 autogroup
-r--------. 1 root root 0 May 22 06:58 auxv
-r--r--r--. 1 root root 0 May 22 06:54 cgroup
--w-------. 1 root root 0 May 22 06:58 clear_refs
-r--r--r--. 1 root root 0 May 22 06:54 cmdline
-rw-r--r--. 1 root root 0 May 22 06:54 comm
-rw-r--r--. 1 root root 0 May 22 06:58 coredump_filter
-r--r--r--. 1 root root 0 May 22 06:58 cpuset
lrwxrwxrwx. 1 root root 0 May 22 06:58 cwd -> /
-r--------. 1 root root 0 May 22 06:54 environ
lrwxrwxrwx. 1 root root 0 May 22 06:54 exe -> /usr/lib/systemd/systemd
dr-x------. 2 root root 0 May 22 06:54 fd
dr-x------. 2 root root 0 May 22 06:58 fdinfo
```

cmdline 这个进程启动的命令

envrion 这个进程的环境变量内容

[root@cloud001 1]# cat cmdline

/usr/lib/systemd/systemd--switched-root--system--deserialize21

和整个Linux系统相关的参数如下：

/proc/cmdline 加载kernel时的相关指令与参数
/proc/cpuinfo CPU相关信息，包含频率、类型与运算功能
/proc/devices 记录了系统各个主要设备的主设备号码
/proc/filesystems 记录系统加载的文件系统
/proc/loadavg 平均负载值 top看到就是这个
/proc/meminfo 内存信息，free命令看到就是这个
/proc/modules 系统已经加载的模块
/proc/mounts 系统已经挂载的数据 mount看到就是这个数据
/proc/partitions 系统的分区文件
/proc/version 系统的核心版本 uname -a看到的内容

常见的系统监视命令：

uptime

能够打印系统总共运行了多长时间和系统的平均负载。uptime命令可以显示的信息显示依次为：现在时间、系统已经运行了多长时间、目前有多少登陆用户、系统在过去的1分钟、5分钟和15分钟内的平均负载。

[root@cloud001 ~]# uptime
07:34:03 up 39 min, 1 user, load average: 0.00, 0.01, 0.05

系统平均负载是指在特定时间间隔内运行队列中的平均进程数。如果每个CPU内核的当前活动进程数不大于3的话，那么系统的性能是良好的。如果每个CPU内核的任务数大于5，那么这台机器的性能有严重问题。如果你的linux主机是1个双核CPU的话，当Load Average 为6的时候说明机器已经被充分使用了

free

可以显示当前系统未使用的和已使用的内存数目，还可以显示被内核使用的内存缓冲区。

[root@cloud001 ~]# free -m

	total	used	free	shared	buff/cache	available
Mem:	3935	167	3603	8	165	3539
Swap:	2047	0	2047			

vmstat

命令是最常见的Linux/Unix监控工具，可以展现给定时间间隔的服务器的状态值,包括服务器的CPU使用率，内存使用，虚拟内存交换情况,IO读写情况。

```
[root@cloud001 ~]# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 3689724   1468 167756    0    0    20     1   44   75  0  0 99  0  0
```

参数解释：

-V：显示vmstat版本信息
-n：只在开始时显示一次各字段名称
-a：显示活跃和非活跃内存
-d：显示各个磁盘相关统计信息
-D：显示磁盘总体信息
-p：显示指定磁盘分区统计信息
-s：显示内存相关统计信息及多种系统活动数量
-m：显示slabinfo
-t：在输出信息的时候也将时间一并输出出来
-S：使用指定单位显示。参数有k、K、m、M，分别代表1000、1024、1000000、1048576字节（byte）。默认单位为K（1024bytes）
delay：刷新时间间隔。如果不指定，只显示一条结果
count：刷新次数。如果不指定刷新次数，但指定了刷新时间间隔，这时刷新次数为无穷

vmstat各字段说明：

1、procs
r：表示运行和等待CPU时间片的进程数（就是说多少个进程真的分配到CPU），这个值如果长期大于系统CPU个数，说明CPU不足，需要增加CPU
b：表示在等待资源的进程数，比如正在等待I/O或者内存交换等。
2、memory
swpd：表示切换到内存交换区的内存大小，即虚拟内存已使用的大小（单位KB），如果大于0，表示你的机器物理内存不足了，如果不是程序内存泄露的原因，那么你该升级内存了或者把耗内存的任务迁移到其他机器。
free：表示当前空闲的物理内存
buff：表示buffers cached内存大小，也就是缓冲大小，一般对块设备的读写才需要缓冲
Cache：表示page cached的内存大小，也就是缓存大小，一般作为文件系统进行缓冲，频繁访问的文件都会被缓存，如果cache值非常大说明缓存文件比较多，如果此时io中的bi比较小，说明文件系统效率比较好
3、swap
si：表示从磁盘调入内存，也就是内存进入内存交换区的内存大小；通俗的讲就是 每秒从磁盘读入虚拟内存的大小，如果这个值大于0，表示物理内存不够用或者内存泄露了，要查找耗内存进程解决掉。
so：表示由内存进入磁盘，也就是由内存交换区进入内存的内存大小。

注意：一般情况下si、so的值都为0，如果si、so的值长期不为0，则说明系统内存不足，需要增加系统内存

4、io

bi：表示从块设备每秒读取的块数量

bo：表示每秒写到块设备的块数量

注意：如果bi+bo的值过大，且wa值较大，则表示系统磁盘IO瓶颈

5、system

in：表示每秒的中断数，包括时钟

cs：表示每秒产生的上下文切换次数，例如我们调用系统函数，就要进行上下文切换，线程的切换，也要进程上下文切换，这个值要越小越好，太大了，要考虑调低线程或者进程的数目,例如在apache和nginx这种web服务器中，我们一般做性能测试时会进行几千并发甚至几万并发的测试，选择web服务器的进程可以由进程或者线程的峰值一直下调，压测，直到cs到一个比较小的值，这个进程和线程数就是比较合适的值了。系统调用也是，每次调用系统函数，我们的代码就会进入内核空间，导致上下文切换，这个是很耗资源，也要尽量避免频繁调用系统函数。上下文切换次数过多表示你的CPU大部分浪费在上下文切换，导致CPU干正经事的时间少了，CPU没有充分利用，是不可取的。

注意：

这两个值越大，则由内核消耗的CPU就越多

6、CPU

us：表示用户进程消耗的CPU时间百分比，us值越高，说明用户进程消耗CPU时间越多，如果长期大于50%，则需要考虑优化程序或者算法

sy：表示系统内核进程消耗的CPU时间百分比，一般来说us+sy应该小于80%，如果大于80%，说明可能存在CPU瓶颈

id：表示CPU处在空间状态的时间百分比

wa：表示IP等待所占用的CPU时间百分比，wa值越高，说明I/O等待越严重，根据经验wa的参考值为20%，如果超过20%，说明I/O等待严重，引起I/O等待的原因可能是磁盘大量随机读写造成的，也可能是磁盘或者监控器的贷款瓶颈（主要是块操作）造成的

综上所述，如果评估CPU，需要重点关注procs项的r列值和CPU的us、sy、wa列的值

使用实践：

一般vmstat工具的使用是通过两个数字参数来完成的，第一个参数是采样的时间间隔数，单位是秒，第二个参数是采样的次数，如：

```
[root@cloud001 ~]# vmstat 2 2
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 3647276   1468 192288    0    0    14     1   35   55  0  0 100  0  0
 0  0      0 3647260   1468 192288    0    0     0     0  110  164  0  0 99  0  0
```

mpstat

是Multiprocessor Statistics的缩写，是实时系统监控工具。其报告与CPU的一些统计信息，这些信息存放在/proc/stat文件中。在多CPUs系统里，其不但能查看所有CPU的平均状况信息，而且能够查看特定CPU的信息。mpstat最大的特点是：可以查看多核心cpu中每个计算核心的统计数据；而类似工具vmstat只能查看系统整体cpu情况。

需要安装：yum install -y sysstat

参数：

-P {|ALL} 表示监控哪个CPU， cpu在[0,cpu个数-1]中取值

internal 相邻的两次采样的间隔时间、

count 采样的次数，count只能和delay一起使用

当没有参数时，mpstat则显示系统启动以后所有信息的平均值。有interval时，第一行的信息自系统启动以来的平均信息。从第二行开始，输出为前一个interval时间段的平均信息。

实例：每2秒查看一次，共查看5次

```
[root@cloud001 ~]# mpstat 2 5
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

08:38:55 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:38:57 AM  all    0.00    0.00    0.25    0.00    0.00    0.00    0.00    0.00    0.00   99.75
08:38:59 AM  all    0.00    0.00    0.25    0.00    0.00    0.00    0.00    0.00    0.00   99.75
08:39:01 AM  all    0.50    0.00    0.50    0.25    0.00    0.00    0.00    0.00    0.00   98.75
08:39:03 AM  all    0.00    0.00    0.25    0.00    0.00    0.00    0.00    0.00    0.00   99.75
08:39:05 AM  all    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
Average:     all    0.10    0.00    0.25    0.05    0.00    0.00    0.00    0.00    0.00   99.60
```

查看每个cpu核心的详细当前运行状况信息，输出如下：

```
[root@cloud001 ~]# mpstat -P ALL 2 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

08:41:01 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:41:03 AM  all    0.25    0.00    0.25    0.00    0.00    0.00    0.00    0.00    0.00   99.50
08:41:03 AM    0    0.00    0.00    0.50    0.00    0.00    0.00    0.00    0.00    0.00   99.50
08:41:03 AM    1    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00

08:41:03 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:41:05 AM  all    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
08:41:05 AM    0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
08:41:05 AM    1    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00

08:41:05 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
08:41:07 AM  all    0.00    0.00    0.25    0.00    0.00    0.00    0.00    0.00    0.00   99.75
08:41:07 AM    0    0.50    0.00    0.50    0.00    0.00    0.00    0.00    0.00    0.00   99.00
08:41:07 AM    1    0.00    0.00    0.50    0.00    0.00    0.00    0.00    0.00    0.00   99.50

Average:     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
Average:     all    0.08    0.00    0.17    0.00    0.00    0.00    0.00    0.00    0.00   99.75
Average:       0    0.17    0.00    0.33    0.00    0.00    0.00    0.00    0.00    0.00   99.50
Average:       1    0.00    0.00    0.17    0.00    0.00    0.00    0.00    0.00    0.00   99.83
```

%user　　在internal时间段里，用户态的CPU时间(%)，不包含nice值为负进程 (usr/total)*100

%nice　　在internal时间段里，nice值为负进程的CPU时间(%)　(nice/total)*100

%sys　　在internal时间段里，内核时间(%)　　(system/total)*100

%iowait　在internal时间段里，硬盘IO等待时间(%) (iowait/total)*100

%irq　　在internal时间段里，硬中断时间(%)　(irq/total)*100

%soft　　在internal时间段里，软中断时间(%)　(softirq/total)*100

%idle　　在internal时间段里，CPU除去等待磁盘IO操作外的因为任何原因而空闲的时间闲置时间(%) (idle/total)*100

iostat

能查看到系统IO状态信息，从而确定IO性能是否存在瓶颈

```
[root@cloud001 ~]# iostat
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.14    0.00    0.39    0.03    0.00   99.44

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
scd0              0.00         0.01         0.00         44          0
sda               1.73        37.00        17.61     253769     120759
sdb               0.11         0.73         0.00       4992          0
sde               0.04         0.27         0.00       1832          0
sdd               0.05         0.33         0.00       2252          0
sdc               0.03         0.19         0.00       1324          0
dm-0              1.57        35.92        17.30     246364     118691
dm-1              0.02         0.16         0.00       1068          0
md2               0.01         0.07         0.00        456          0
dm-2              0.01         0.07         0.00        456          0
dm-3              0.03         0.13         0.00        908          0
dm-4              0.02         0.07         0.00        456          0
dm-5              0.00         0.01         0.00         60          0
dm-6              0.02         0.07         0.00        456          0
dm-7              0.01         0.07         0.00        456          0
```

tps：该设备每秒的传输次数（Indicate the number of transfers per second that were issued to the device.）。“一次传输”意思是“一次I/O请求”。多个逻辑请求可能会被合并为“一次I/O请求”。“一次传输”请求的大小是未知的。

kB_read/s：每秒从设备（drive expressed）读取的数据量；

kB_wrtn/s：每秒向设备（drive expressed）写入的数据量；

kB_read：读取的总数据量；

kB_wrtn：写入的总数量数据量；

常见用法：

iostat -d -k 1 10　　#查看TPS和吞吐量信息

iostat -d -x -k 1 10　　#查看设备使用率（%util）、响应时间（await）

iostat -c 1 10　　#查看cpu状态

sar

sar（System Activity Reporter系统活动情况报告）是目前 Linux 上最为全面的系统性能分析工具之一，可以从多方面对系统的活动进行报告，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有关的活动等。

sar命令常用格式

sar [options] [-A] [-o file] t [n]

其中：

t为采样间隔，n为采样次数，默认值是1；

-o file表示将命令结果以二进制格式存放在文件中，file 是文件名。

options 为命令行选项，sar命令常用选项如下：

-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

-r：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

实例：

1. CPU资源监控

例如，每10秒采样一次，连续采样3次，观察CPU 的使用情况，并将采样结果以二进制形式存入当前目录下的文件test中，需键入如下命令：

sar -u -o test 10 3

屏幕显示如下：

```
[root@cloud001 tmp]# sar -u -o test 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:12:33 AM     CPU     %user     %nice   %system   %iowait    %steal     %idle
09:12:43 AM     all      0.00      0.00      0.25      0.00      0.00     99.75
```

输出项说明：

CPU：all 表示统计信息为所有 CPU 的平均值。

%user：显示在用户级别(application)运行使用 CPU 总时间的百分比。

%nice：显示在用户级别，用于nice操作，所占用 CPU 总时间的百分比。

%system：在核心级别(kernel)运行所使用 CPU 总时间的百分比。

%iowait：显示用于等待I/O操作占用 CPU 总时间的百分比。

%steal：管理程序(hypervisor)为另一个虚拟进程提供服务而等待虚拟 CPU 的百分比。

%idle：显示 CPU 空闲时间占用 CPU 总时间的百分比。

1. 若 %iowait 的值过高，表示硬盘存在I/O瓶颈

2. 若 %idle 的值高但系统响应慢时，有可能是 CPU 等待分配内存，此时应加大内存容量

3. 若 %idle 的值持续低于1，则系统的CPU处理能力相对较低，表明系统中最需要解决的资源是 CPU

如果要查看二进制文件test中的内容，需键入如下sar命令：

sar -u -f test

2. inode、文件和其他内核表监控

例如，每10秒采样一次，连续采样3次，观察核心表的状态，需键入如下命令：

sar -v 10 3

屏幕显示如下：

```
[root@cloud001 tmp]# sar -v 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:14:23 AM dentunusd   file-nr  inode-nr    pty-nr
09:14:33 AM     13080      1024     23370         2
09:14:43 AM     13080      1024     23370         2
09:14:53 AM     13080      1024     23370         2
Average:        13080      1024     23370         2
```

输出项说明：

dentunusd：目录高速缓存中未被使用的条目数量

file-nr：文件句柄（file handle）的使用数量

inode-nr：索引节点句柄（inode handle）的使用数量

pty-nr：使用的pty数量

3. 内存和交换空间监控

例如，每10秒采样一次，连续采样3次，监控内存分页：

sar -r 10 3

屏幕显示如下：

```
[root@cloud001 ~]# sar -r 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:14:53 AM kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit  kbactive   kbinact   kbdirty
09:15:03 AM   3689440    340976      8.46      1468    102956    353536      5.77     95172     75904         4
09:15:13 AM   3689440    340976      8.46      1468    102956    353536      5.77     95180     75904         4
09:15:23 AM   3689440    340976      8.46      1468    102956    353536      5.77     95180     75904         0
Average:      3689440    340976      8.46      1468    102956    353536      5.77     95177     75904         3
```

输出项说明：

kbmemfree：这个值和free命令中的free值基本一致,所以它不包括buffer和cache的空间.
kbmemused：这个值和free命令中的used值基本一致,所以它包括buffer和cache的空间.
%memused：这个值是kbmemused和内存总量(不包括swap)的一个百分比.
kbbuffers和kbcached：这两个值就是free命令中的buffer和cache.
kbcommit：保证当前系统所需要的内存,即为了确保不溢出而需要的内存(RAM+swap).
%commit：这个值是kbcommit与内存总量(包括swap)的一个百分比.
4. 内存分页监控
例如，每10秒采样一次，连续采样3次，监控内存分页：
sar -B 10 3
屏幕显示如下：

```
[root@cloud001 ~]# sar -B 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:16:13 AM  pgpgin/s pgpgout/s   fault/s  majflt/s  pgfree/s pgscank/s pgscand/s pgsteal/s    %vmeff
09:16:23 AM      0.00      0.00      7.00      0.00      8.90      0.00      0.00      0.00      0.00
09:16:33 AM      3.20      0.00    181.10      0.00     97.90      0.00      0.00      0.00      0.00
09:16:43 AM      0.00      0.20      3.90      0.00     11.20      0.00      0.00      0.00      0.00
Average:         1.07      0.07     64.00      0.00     39.33      0.00      0.00      0.00      0.00
```

输出项说明：
pgpgin/s：表示每秒从磁盘或SWAP置换到内存的字节数(KB)
pgpgout/s：表示每秒从内存置换到磁盘或SWAP的字节数(KB)
fault/s：每秒钟系统产生的缺页数,即主缺页与次缺页之和(major + minor)
majflt/s：每秒钟产生的主缺页数.
pgfree/s：每秒被放入空闲队列中的页个数
pgscank/s：每秒被kswapd扫描的页个数
pgscand/s：每秒直接被扫描的页个数
pgsteal/s：每秒钟从cache中被清除来满足内存需要的页个数
%vmeff：每秒清除的页(pgsteal)占总扫描页(pgscank+pgscand)的百分比
5. I/O和传送速率监控
例如，每10秒采样一次，连续采样3次，报告缓冲区的使用情况，需键入如下命令：
sar -b 10 3

```
[root@cloud001 tmp]# sar -b 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:16:44 AM       tps      rtps      wtps   bread/s   bwrtn/s
09:16:54 AM      0.00      0.00      0.00      0.00      0.00
09:17:04 AM      0.20      0.00      0.20      0.00      1.60
09:17:14 AM      0.20      0.00      0.20      0.00      1.20
Average:         0.13      0.00      0.13      0.00      0.93
```

输出项说明：
tps：每秒钟物理设备的 I/O 传输总量
rtps：每秒钟从物理设备读入的数据总量

wtps：每秒钟向物理设备写入的数据总量

bread/s：每秒钟从物理设备读入的数据量，单位为 块/s

bwrtn/s：每秒钟向物理设备写入的数据量，单位为 块/s

6. 进程队列长度和平均负载状态监控

例如，每10秒采样一次，连续采样3次，监控进程队列长度和平均负载状态：

sar -q 10 3

屏幕显示如下：

```
[root@cloud001 tmp]# sar -q 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:18:01 AM   runq-sz  plist-sz   ldavg-1   ldavg-5  ldavg-15   blocked
09:18:11 AM         0       153      0.00      0.01      0.05         0
09:18:21 AM         0       153      0.00      0.01      0.05         0
09:18:31 AM         0       153      0.00      0.01      0.05         0
Average:            0       153      0.00      0.01      0.05         0
```

输出项说明：

runq-sz：运行队列的长度（等待运行的进程数）

plist-sz：进程列表中进程（processes）和线程（threads）的数量

ldavg-1：最后1分钟的系统平均负载（System load average）

ldavg-5：过去5分钟的系统平均负载

ldavg-15：过去15分钟的系统平均负载

7. 系统交换活动信息监控

例如，每10秒采样一次，连续采样3次，监控系统交换活动信息：

sar - W 10 3

屏幕显示如下：

```
[root@cloud001 ~]# sar -W 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:18:37 AM  pswpin/s pswpout/s
09:18:47 AM      0.00      0.00
09:18:57 AM      0.00      0.00
09:19:07 AM      0.00      0.00
Average:         0.00      0.00
```

输出项说明：

pswpin/s：每秒系统换入的交换页面（swap page）数量

pswpout/s：每秒系统换出的交换页面（swap page）数量

8. 设备使用情况监控

例如，每10秒采样一次，连续采样3次，报告设备使用情况，需键入如下命令：

sar -d 10 3 –p

屏幕显示如下：

```
[root@cloud001 ~]# sar -d 10 3
Linux 3.10.0-514.el7.x86_64 (cloud001)  05/22/2018      _x86_64_        (2 CPU)

09:20:17 AM       DEV       tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz     await     svctm     %util
09:20:27 AM   dev8-32      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM   dev8-16      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM    dev8-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM   dev8-48      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM   dev8-64      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM   dev11-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-0      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-1      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM    dev9-2      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-2      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-3      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-4      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-5      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-6      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
09:20:27 AM  dev253-7      0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00
```

其中：

参数-p可以打印出sda,hdc等磁盘设备名称,如果不用参数-p,设备节点则有可能是dev8-0,dev22-0

tps:每秒从物理磁盘I/O的次数.多个逻辑请求会被合并为一个I/O磁盘请求,一次传输的大小是不确定的.

rd_sec/s:每秒读扇区的次数.

wr_sec/s:每秒写扇区的次数.

avgrq-sz:平均每次设备I/O操作的数据大小(扇区).

avgqu-sz:磁盘请求队列的平均长度.

await:从请求磁盘操作到系统完成处理,每次请求的平均消耗时间,包括请求队列等待时间,单位是毫秒(1秒=1000毫秒).

svctm:系统处理每次请求的平均时间,不包括在请求队列中消耗的时间.

%util:I/O请求占CPU的百分比,比率越大,说明越饱和.

1. avgqu-sz 的值较低时，设备的利用率较高。
2. 当%util的值接近 1% 时，表示设备带宽已经占满。

要判断系统瓶颈问题，有时需几个 sar 命令选项结合起来

怀疑CPU存在瓶颈，可用 sar -u 和 sar -q 等来查看

怀疑内存存在瓶颈，可用 sar -B、sar -r 和 sar -W 等来查看

怀疑I/O存在瓶颈，可用 sar -b、sar -u 和 sar -d 等来查看

iotop

实时观察磁盘io情况，可以观察到哪个进程占用I/O

参数：

-o：只显示有io操作的进程

-b：批量显示，无交互，主要用作记录到文件。

-n NUM：显示NUM次，主要用于非交互式模式。

-d SEC：间隔SEC秒显示一次。

-p PID：监控的进程pid。

-u USER：监控的进程用户。