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**。

本文主要以CentOS 6.3 x64系统为例，介绍sar命令。

**1、sar命令常用格式**

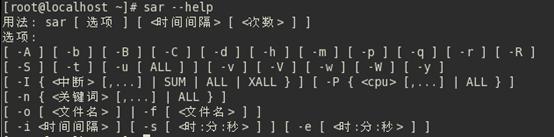
sar [options] [-A] [-o file] t [n]

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：

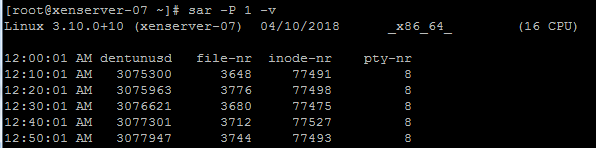


-A：所有报告的总和

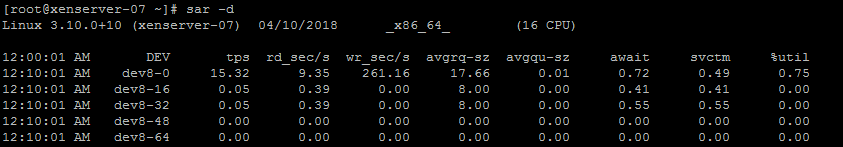
-P：设定CPU

-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

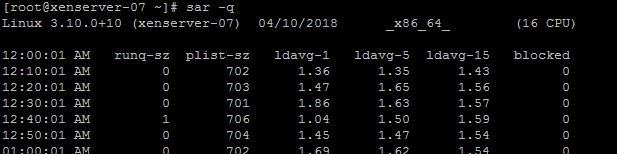


-d：输出每一个块设备的活动信息

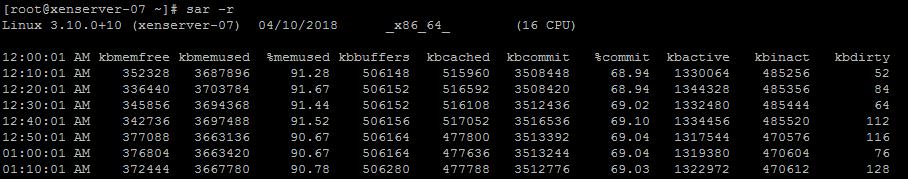


-n：汇报网络情况

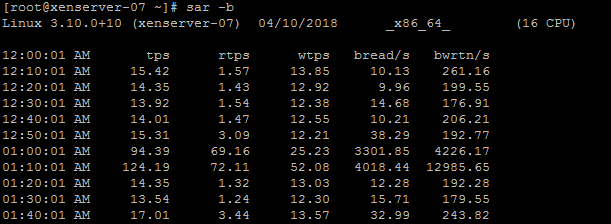
-q：汇报队列长度和负载信息



-r：输出内存和交换空间的统计信息



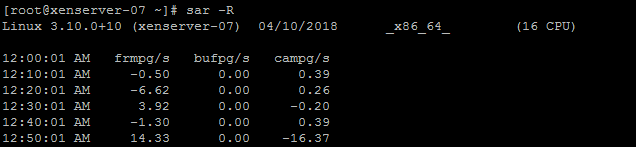
-b：显示I/O和传送速率的统计信息



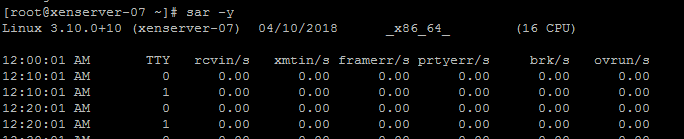
-a：文件读写情况

-c：输出进程统计信息，每秒创建的进程数

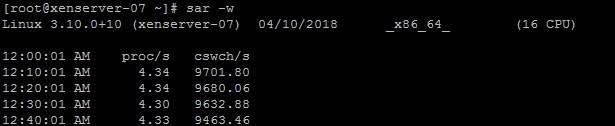
-R：输出内存页面的统计信息



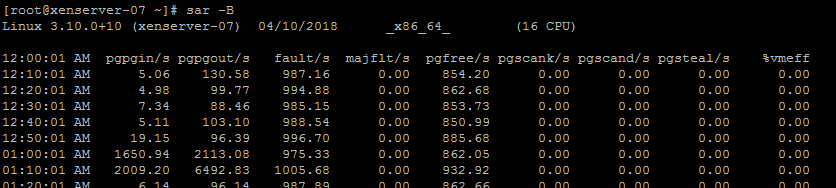
-y：终端设备活动情况



-w：输出系统交换活动信息



-B: 内存分页



**2. 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sys\_info中，需键入如下命令：

sar -u -o sys\_info  10 3

屏幕显示如下：

17:06:16 CPU %user %nice %system %iowait %steal %idle

17:06:26 all 0.00 0.00 0.20 0.00 0.00 99.80

17:06:36 all 0.00 0.00 0.20 0.00 0.00 99.80

17:06:46 all 0.00 0.00 0.10 0.00 0.00 99.90

Average: all 0.00 0.00 0.17 0.00 0.00 99.83

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。

**注意说明：**

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 。

由于sys\_info是二进制文件中的内容， 当你用cat看sys\_info时发现全都是乱码，呵呵，别着急，sar为你准备了-f filename选项，你只要用-f设定要读取的信息存储文件，就可以清晰地读出信息了。

比如sar -f sys\_info。

**如果我的CPU是多核处理器，那么sar能知道某一个核的运行信息么？**

      完全没问题的。有一个选项-P，就是用来为多核处理器而设计的。

      当在使用sar命令而没有设定-P选项时，sar会根据所有核给出一个宏观汇报，也就是平均的值。

       如果使用了-P选项来指定某一个核，那么就会针对这个单独的核给出具体性能信息。

      当使用-P ALL时，sar就会根据每一个核都给出其具体性能信息，然后再给出一个总的性能信息。

       比如，我这里有一个至强处理器的CPU，是双核CPU，看看-P的使用方法吧：

sar -P ALL 1 1

Linux 2.6.9    10/16/2009

10:59:38 PM       CPU     %user     %nice   %system   %iowait     %idle

10:59:39 PM       all      2.12      0.00      2.87      0.00     95.01

10:59:39 PM         0      0.00      0.00      1.98      0.00     98.02

10:59:39 PM         1      9.00      0.00      7.00      0.00     84.00

sar会根据处理器的每一个核给出性能信息。当我们想查看第6个核的信息时，其输出如下：

sar -P 0 1 1

**3. 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：

17:10:49 dentunusd file-nr inode-nr pty-nr

17:10:59 6301 5664 12037 4

17:11:09 6301 5664 12037 4

17:11:19 6301 5664 12037 4

Average: 6301 5664 12037 4

**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

**4. 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：

10:02:52 PM kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit   %commit

10:03:02 PM   2289016   1632096     41.62    204860   1218352    333068      8.49

10:03:12 PM   2288388   1632724     41.64    204860   1218352    333068      8.49

10:03:22 PM   2288544   1632568     41.64    204860   1218352    333068      8.49

Average:        2288649   1632463     41.63    204860   1218352    333068      8.49

**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).

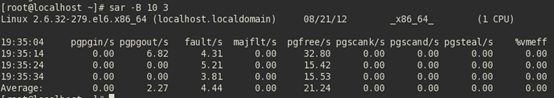
%commit：这个值是kbcommit与内存总量(包括swap)的一个百分比.

**5. 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：



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

**6. 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

屏幕显示如下：

18:51:05 tps rtps wtps bread/s bwrtn/s

18:51:15 0.00 0.00 0.00 0.00 0.00

18:51:25 1.92 0.00 1.92 0.00 22.65

18:51:35 0.00 0.00 0.00 0.00 0.00

Average: 0.64 0.00 0.64 0.00 7.59

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

**7. 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：

19:25:50 runq-sz plist-sz ldavg-1 ldavg-5 ldavg-15

19:26:00 0 259 0.00 0.00 0.00

19:26:10 0 259 0.00 0.00 0.00

19:26:20 0 259 0.00 0.00 0.00

Average: 0 259 0.00 0.00 0.00

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

**8. 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：

19:39:50 pswpin/s pswpout/s

19:40:00 0.00 0.00

19:40:10 0.00 0.00

19:40:20 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

**9. 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：

17:45:54    DEV    tps    rd\_sec/s    wr\_sec/s    avgrq-sz    avgqu-sz    await    svctm    %util

17:46:04    scd0    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

17:46:04    sda    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

17:46:04    vg\_livedvd-lv\_root    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

17:46:04    vg\_livedvd-lv\_swap    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 100% 时，表示设备带宽已经占满。

**10、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

**11、利用sar来做个后台监控程序，可以实时汇报机器性能情况。sar能放后台运行么**

借助linux的后台符就可以了，别忘了把标准输出重定向哦，方法是这样的：

sar -o monitor.res interval count >/dev/null 2>&1 &

记得把interval和count都替换为你想要的间隔和次数。这样所有的性能信息就全都存入二进制格式的数据文件monitor.res里了。当读取时使用-f就可以了。

**12、sar分析网卡流量**

sar -n { DEV | EDEV | NFS | NFSD | SOCK | ALL }

sar 提供六种不同的语法选项来显示网络信息。-n选项使用6个不同的开关：DEV | EDEV | NFS | NFSD | SOCK | ALL 。DEV显示网络接口信息，EDEV显示关于网络错误的统计数据，NFS统计活动的NFS客户端的信息，NFSD统计NFS服务器的信息，SOCK显示套接字信息，ALL显示所有5个开关。它们可以单独或者一起使用。

如果你使用DEV关键字，那么sar将汇报和网络设备相关的信息，如lo，eth0或eth1等，例如

**#sar -n DEV 2 10**

[Linux](http://www.2cto.com/os/linux/) 2.6.18-53.el5PAE (localhost.localdomain) 03/29/2009

01:39:40 AM IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcst/s

01:39:42 AM lo 0.00 0.00 0.00 0.00 0.00 0.00 0.00

01:39:42 AM eth1 131.34 104.98 119704.48 36110.45 0.00 0.00 0.00

01:39:42 AM sit0 0.00 0.00 0.00 0.00 0.00 0.00 0.00

01:39:42 AM IFACE rxpck/s txpck/s rxbyt/s txbyt/s rxcmp/s txcmp/s rxmcst/s

01:39:44 AM lo 0.00 0.00 0.00 0.00 0.00 0.00 0.00

01:39:44 AM eth1 168.00 165.50 114496.50 83938.50 0.00 0.00 0.00

01:39:44 AM sit0 0.00 0.00 0.00 0.00 0.00 0.00 0.00

IFACE：LAN接口，网络设备的名称

rxpck/s：每秒钟接收的数据包

txpck/s：每秒钟发送的数据包

rxbyt/s：每秒钟接收的字节数

txbyt/s：每秒钟发送的字节数

rxcmp/s：每秒钟接收的压缩数据包

txcmp/s：每秒钟发送的压缩数据包

rxmcst/s：每秒钟接收的多播数据包

使用EDEV关键字，那么会针对网络设备汇报其失败情况，例如：

**#sar -n EDEV 2 10**

Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/29/2009

01:42:18 AM IFACE rxerr/s txerr/s coll/s rxdrop/s txdrop/s txcarr/s rxfram/s rxfifo/s txfifo/s

01:42:20 AM lo 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

01:42:20 AM eth1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

01:42:20 AM sit0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

IFACE：LAN接口网络设备的名称

rxerr/s：每秒钟接收的坏数据包

txerr/s：每秒钟发送的坏数据包

coll/s：每秒冲突数

rxdrop/s：因为缓冲充满，每秒钟丢弃的已接收数据包数

txdrop/s：因为缓冲充满，每秒钟丢弃的已发送数据包数

txcarr/s：发送数据包时，每秒载波错误数

rxfram/s：每秒接收数据包的帧对齐错误数

rxfifo/s：接收的数据包每秒FIFO过速的错误数

txfifo/s：发送的数据包每秒FIFO过速的错误数

使用SOCK关键字，则会针对socket连接进行汇报：

#sar -n SOCK 2 10

Linux 2.6.18-53.el5PAE (localhost.localdomain) 03/29/2009

  www.2cto.com

01:44:32 AM totsck tcpsck udpsck rawsck ip-frag

01:44:34 AM 243 9 8 0 0

01:44:36 AM 242 9 7 0 0

01:44:38 AM 238 9 7 0 0

01:44:40 AM 238 9 7 0 0

totsck:使用的套接字总数量

tcpsck:使用的TCP套接字数量

udpsck:使用的UDP套接字数量

rawsck:使用的raw套接字数量

ip-frag:使用的IP段数量

如果你使用FULL关键字，相当于上述DEV、EDEV和SOCK三者的综合