[**每天一个linux命令（57）：ss命令**](http://www.cnblogs.com/peida/archive/2013/03/11/2953420.html)

ss是Socket Statistics的缩写。顾名思义，ss命令可以用来获取socket统计信息，它可以显示和netstat类似的内容。但ss的优势在于它能够显示更多更详细的有关TCP和连接状态的信息，而且比netstat更快速更高效。

当服务器的socket连接数量变得非常大时，无论是使用netstat命令还是直接cat /proc/net/tcp，执行速度都会很慢。可能你不会有切身的感受，但请相信我，当服务器维持的连接达到上万个的时候，使用netstat等于浪费 生命，而用ss才是节省时间。

天下武功唯快不破。ss快的秘诀在于，它利用到了TCP协议栈中tcp\_diag。tcp\_diag是一个用于分析统计的模块，可以获得Linux 内核中第一手的信息，这就确保了ss的快捷高效。当然，如果你的系统中没有tcp\_diag，ss也可以正常运行，只是效率会变得稍慢。（但仍然比 netstat要快。）

**1.命令格式:**

ss [参数]

ss [参数] [过滤]

**2.命令功能：**

ss(Socket Statistics的缩写)命令可以用来获取 socket统计信息，此命令输出的结果类似于 netstat输出的内容，但它能显示更多更详细的 TCP连接状态的信息，且比 netstat 更快速高效。它使用了 TCP协议栈中 tcp\_diag（是一个用于分析统计的模块），能直接从获得第一手内核信息，这就使得 ss命令快捷高效。在没有 tcp\_diag，ss也可以正常运行。

**3.命令参数：**

-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        显示计时器信息

-e, --extended       显示详细的套接字（sockets）信息

-m, --memory         显示套接字（socket）的内存使用情况

-p, --processes 显示使用套接字（socket）的进程

-i, --info 显示 TCP内部信息

-s, --summary 显示套接字（socket）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（socket）

-t, --tcp 仅显示 TCP套接字（sockets）

-u, --udp 仅显示 UCP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

      QUERY := {all|inet|tcp|udp|raw|unix|packet|netlink}[,QUERY]

-D, --diag=FILE     将原始TCP套接字（sockets）信息转储到文件

 -F, --filter=FILE   从文件中都去过滤器信息

       FILTER := [ state TCP-STATE ] [ EXPRESSION ]

**4.使用实例：**

**实例1：显示TCP连接**

**命令：**

ss -t -a

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -t -a  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
LISTEN     0      0                                         127.0.0.1:smux                                               \*:\*         
LISTEN     0      0                                                 \*:3690                                               \*:\*         
LISTEN     0      0                                                 \*:ssh                                                \*:\*         
ESTAB      0      0                                   192.168.120.204:ssh                                        10.2.0.68:49368     
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例2：显示 Sockets 摘要**

**命令：**

ss -s

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -s  
Total: 34 (kernel 48)  
TCP:   4 (estab 1, closed 0, orphaned 0, synrecv 0, timewait 0/0), ports 3  
  
Transport Total     IP        IPv6  
\*         48        -         -          
RAW       0         0         0          
UDP       5         5         0          
TCP       4         4         0          
INET      9         9         0          
FRAG      0         0         0          
  
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

列出当前的established, closed, orphaned and waiting TCP sockets

**实例3：**列出所有打开的网络连接端口

**命令：**

ss -l

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -l  
Recv-Q Send-Q                                     Local Address:Port                                         Peer Address:Port     
0      0                                              127.0.0.1:smux                                                    \*:\*         
0      0                                                      \*:3690                                                    \*:\*         
0      0                                                      \*:ssh                                                     \*:\*         
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例4：查看进程使用的socket**

**命令：**

ss -pl

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -pl  
Recv-Q Send-Q                                     Local Address:Port                                         Peer Address:Port     
0      0                                              127.0.0.1:smux                                                    \*:\*        users:(("snmpd",2716,8))  
0      0                                                      \*:3690                                                    \*:\*        users:(("svnserve",3590,3))  
0      0                                                      \*:ssh                                                     \*:\*        users:(("sshd",2735,3))  
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例5：找出打开套接字/端口应用程序**

**命令：**

ss -lp | grep 3306

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -lp|grep 1935  
0      0                            \*:1935                          \*:\*        users:(("fmsedge",2913,18))  
0      0                    127.0.0.1:19350                         \*:\*        users:(("fmsedge",2913,17))  
[root@localhost ~]# ss -lp|grep 3306  
0      0                            \*:3306                          \*:\*        users:(("mysqld",2871,10))  
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例6：显示所有UDP Sockets**

**命令：**

ss -u -a

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss -u -a  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
UNCONN     0      0                                         127.0.0.1:syslog                                             \*:\*         
UNCONN     0      0                                                 \*:snmp                                               \*:\*         
ESTAB      0      0                                   192.168.120.203:39641                                  10.58.119.119:domain   
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例7：显示所有状态为established的SMTP连接**

**命令：**

ss -o state established '( dport = :smtp or sport = :smtp )'

**输出：**

[root@localhost ~]# ss -o state established '( dport = :smtp or sport = :smtp )'   
Recv-Q Send-Q                                     Local Address:Port                                         Peer Address:Port     
[root@localhost ~]#

**说明：**

**实例8：显示所有状态为Established的HTTP连接**

**命令：**

ss -o state established '( dport = :http or sport = :http )'

**输出：**

[root@localhost ~]# ss -o state established '( dport = :http or sport = :http )'   
Recv-Q Send-Q                                     Local Address:Port                                         Peer Address:Port     
0      0                                              75.126.153.214:2164                                        192.168.10.42:http      
[root@localhost ~]#

**说明：**

**实例9：列举出处于 FIN-WAIT-1状态的源端口为 80或者 443，目标网络为 193.233.7/24所有 tcp套接字**

**命令：**

ss -o state fin-wait-1 '( sport = :http or sport = :https )' dst 193.233.7/24

**输出：**

**说明：**

**实例10：用TCP 状态过滤Sockets:**

**命令：**

ss -4 state FILTER-NAME-HERE

ss -6 state FILTER-NAME-HERE

**输出：**

[root@localhost ~]#ss -4 state closing   
Recv-Q Send-Q                                                  Local Address:Port                                                      Peer Address:Port   
1      11094                                                  75.126.153.214:http                                                      192.168.10.42:4669

**说明：**

FILTER-NAME-HERE 可以代表以下任何一个：

established

syn-sent

syn-recv

fin-wait-1

fin-wait-2

time-wait

closed

close-wait

last-ack

listen

closing

all : 所有以上状态

connected : 除了listen and closed的所有状态

synchronized :所有已连接的状态除了syn-sent

bucket : 显示状态为maintained as minisockets,如：time-wait和syn-recv.

big : 和bucket相反.

**实例11：匹配远程地址和端口号**

**命令：**

ss dst ADDRESS\_PATTERN

ss dst 192.168.1.5

ss dst 192.168.119.113:http

ss dst 192.168.119.113:smtp

ss dst 192.168.119.113:443

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss dst 192.168.119.113  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:20229     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:61056     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:61623     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:60924     
ESTAB      0      0                                   192.168.119.103:16050                                192.168.119.113:43701     
ESTAB      0      0                                   192.168.119.103:16073                                192.168.119.113:32930     
ESTAB      0      0                                   192.168.119.103:16073                                192.168.119.113:49318     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:3844      
[root@localhost ~]# ss dst 192.168.119.113:http  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
[root@localhost ~]# ss dst 192.168.119.113:3844  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
ESTAB      0      0                                   192.168.119.103:16014                                192.168.119.113:3844      
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例12：匹配本地地址和端口号**

**命令：**

ss src ADDRESS\_PATTERN

ss src 192.168.119.103

ss src 192.168.119.103:http

ss src 192.168.119.103:80

ss src 192.168.119.103:smtp

ss src 192.168.119.103:25

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss src 192.168.119.103:16021  
State      Recv-Q Send-Q                                Local Address:Port                                    Peer Address:Port     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:63054     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:62894     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:63055     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:2274      
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:44784     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:7233      
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.103:58660     
ESTAB      0      0                                   192.168.119.103:16021                                192.168.119.201:44822     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:56737     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:57487     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:56736     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:64652     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:56586     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:64653     
ESTAB      0      0                                   192.168.119.103:16021                                     10.2.1.206:56587     
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

**实例13：将本地或者远程端口和一个数比较**

**命令：**

ss dport OP PORT

ss sport OP PORT

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# ss  sport = :http   
[root@localhost ~]# ss  dport = :http   
[root@localhost ~]# ss  dport \> :1024   
[root@localhost ~]# ss  sport \> :1024   
[root@localhost ~]# ss sport \< :32000   
[root@localhost ~]# ss  sport eq :22   
[root@localhost ~]# ss  dport != :22   
[root@localhost ~]# ss  state connected sport = :http   
[root@localhost ~]# ss \( sport = :http or sport = :https \)   
[root@localhost ~]# ss -o state fin-wait-1 \( sport = :http or sport = :https \) dst 192.168.1/24

[复制代码](javascript:void(0);)

**说明：**

ss dport OP PORT 远程端口和一个数比较；ss sport OP PORT 本地端口和一个数比较。

**OP 可以代表以下任意一个:**

<= or le : 小于或等于端口号

>= or ge : 大于或等于端口号

== or eq : 等于端口号

!= or ne : 不等于端口号

< or gt : 小于端口号

> or lt : 大于端口号

**实例14：ss 和 netstat 效率对比**

**命令：**

time netstat -at

time ss

**输出：**

[复制代码](javascript:void(0);)

[root@localhost ~]# time ss     
real    0m0.739s  
user    0m0.019s  
sys     0m0.013s  
[root@localhost ~]#   
[root@localhost ~]# time netstat -at  
real    2m45.907s  
user    0m0.063s  
sys     0m0.067s  
[root@localhost ~]#

[复制代码](javascript:void(0);)

**说明：**

用time 命令分别获取通过netstat和ss命令获取程序和概要占用资源所使用的时间。在服务器连接数比较多的时候，netstat的效率完全没法和ss比。