ss

比 netstat 好用的socket统计信息，iproute2 包附带的另一个工具，允许你查询 socket 的有关统计信息

补充说明

ss命令 用来显示处于活动状态的套接字信息，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。

语法

```
ss [参数]
ss [参数] [过滤]
```

选项

```
-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 ]
```

实例

```
ss -t -a    # 显示TCP连接
ss -s       # 显示 Sockets 摘要
ss -l       # 列出所有打开的网络连接端口
ss -pl      # 查看进程使用的socket
ss -lp | grep 3306  # 找出打开套接字/端口应用程序
ss -u -a    显示所有UDP Sockets
ss -o state established '( dport = :smtp or sport = :smtp )' # 显示所有状态为established的SMTP连接
ss -o state established '( dport = :http or sport = :http )' # 显示所有状态为Established的HTTP连接
ss -o state fin-wait-1 '( sport = :http or sport = :https )' dst 193.233.7/24  # 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

# ss 和 netstat 效率对比
time netstat -at
time ss

# 匹配远程地址和端口号
# 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

# 匹配本地地址和端口号
# 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
```

将本地或者远程端口和一个数比较

```
# 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
ss  sport = :http
ss  dport = :http
ss  dport \> :1024
```

```
ss  sport \> :1024
ss sport \< :32000
ss  sport eq :22
ss  dport != :22
ss  state connected sport = :http
ss \( sport = :http or sport = :https \)
ss -o state fin-wait-1 \( sport = :http or sport = :https \) dst
192.168.1/24
```

用TCP 状态过滤Sockets

```
ss -4 state closing
# ss -4 state FILTER-NAME-HERE
# ss -6 state FILTER-NAME-HERE
# 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.
```

显示ICP连接

```
[root@localhost ~]# ss -t -a
State       Recv-Q Send-Q                                 Local Address:Port
Peer Address:Port
LISTEN      0      0                                            *:3306
*:*
LISTEN      0      0                                            *:http
*:*
LISTEN      0      0                                            *:ssh
*:*
LISTEN      0      0                                        127.0.0.1:smtp
*:*
ESTAB       0      0                                    112.124.15.130:42071
42.156.166.25:http
ESTAB       0      0                                    112.124.15.130:ssh
121.229.196.235:33398
```

显示 Sockets 摘要

```
[root@localhost ~]# ss -s
Total: 172 (kernel 189)
TCP:   10 (estab 2, closed 4, orphaned 0, synrecv 0, timewait 0/0), ports 5

Transport Total     ip        IPv6
*         189       -         -
RAW       0         0         0
UDP       5         5         0
```

```
TCP       6         6         0
INET      11        11        0
FRAG      0         0         0
```

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

列出所有打开的网络连接端口

```
[root@localhost ~]# ss -l
Recv-Q Send-Q                                  Local Address:Port
Peer Address:Port
0      0                                                *:3306
*:*
0      0                                                *:http
*:*
0      0                                                *:ssh
*:*
0      0                                           127.0.0.1:smtp
*:*
```

查看进程使用的socket

```
[root@localhost ~]# ss -pl
Recv-Q Send-Q                                        Local Address:Port
Peer Address:Port
0      0                                                          *:3306
*:*        users:(("mysqld",1718,10))
0      0                                                          *:http
*:*        users:(("nginx",13312,5),("nginx",13333,5))
0      0                                                          *:ssh
*:*        users:(("sshd",1379,3))
0      0                                                   127.0.0.1:smtp
*:*        us
```

找出打开套接字/端口应用程序

```
[root@localhost ~]# ss -pl | grep 3306
0      0                            *:3306                          *:*
users:(("mysqld",1718,10))
```

显示所有UDP Sockets

```
[root@localhost ~]# ss -u -a
State       Recv-Q Send-Q                                     Local
Address:Port                                    Peer Address:Port
UNCONN      0      0
*:syslog                                                *:*
UNCONN      0      0
112.124.15.130:ntp                                                 *:*
UNCONN      0      0
10.160.7.81:ntp                                                  *:*
```

```
UNCONN      0      0                                                    *:*
127.0.0.1:ntp
UNCONN      0      0
*:ntp                                                          *:*
```

出所有端口为 22（ssh）的连接

```
ss state all sport = :ssh

Netid State      Recv-Q Send-Q     Local Address:Port
Peer Address:Port
tcp   LISTEN     0      128                    *:ssh
*:*
tcp   ESTAB      0      0          192.168.0.136:ssh
192.168.0.102:46540
tcp   LISTEN     0      128                   :::ssh
:::*
```