Linux网络状态工具ss命令使用详解

ss命令用于显示socket状态. 他可以显示PACKET sockets, TCP sockets, UDP sockets, DCCP sockets, RAW sockets, Unix domain sockets等等统计. 它比其他工具展示等多tcp和state信息. 它是一个非常实用、快速、有效的跟踪IP连接和sockets的新工具.SS命令可以提供如下信息：

- 所有的TCP sockets
- 所有的UDP sockets
- 所有ssh/ftp/ttp/https持久连接
- 所有连接到Xserver的本地进程
- 使用state（例如：connected, synchronized, SYN-RECV, SYN-SENT,TIME-WAIT）、地址、端口过滤
- 所有的state FIN-WAIT-1 tcpsocket连接以及更多

很多流行的Linux发行版都支持ss以及很多监控工具使用ss命令.熟悉这个工具有助于您更好的发现与解决系统性能问题.本人强烈建议使用ss命令替代netstat部分命令,例如netsat -ant/lnt等.

展示他之前来做个对比,统计服务器并发连接数

```
 netstat
# time netstat -ant | grep EST | wc -l
3100

real 0m12.960s
user 0m0.334s
sys 0m12.561s
# time ss -o state established | wc -l
3204

real 0m0.030s
user 0m0.005s
sys 0m0.026s
```

结果很明显ss统计并发连接数效率完败netstat,在ss能搞定的情况下, 你还会在选择netstat吗, 还在犹豫吗, 看以下例子,或者跳转到帮助页面.

常用ss命令：

```
ss -l 显示本地打开的所有端口
ss -pl 显示每个进程具体打开的socket
ss -t -a 显示所有tcp socket
ss -u -a 显示所有的UDP Socekt
ss -o state established '( dport = :smtp or sport = :smtp )' 显示所有已建立的SMTP连接
ss -o state established '( dport = :http or sport = :http )' 显示所有已建立的HTTP连接
ss -x src /tmp/.X11-unix/* 找出所有连接X服务器的进程
ss -s 列出当前socket详细信息:
```

显示sockets简要信息

列出当前已经连接，关闭，等待的tcp连接

```
# ss -s
Total: 3519 (kernel 3691)
TCP: 26557 (estab 3163, closed 23182, orphaned 194, synrecv 0, timewait 23182/0), ports 1452

Transport Total IP IPv6
* 3691 - -
RAW 2 2 0
UDP 10 7 3
TCP 3375 3368 7
INET 3387 3377 10
FRAG 0 0 0
```

列出当前监听端口

```
# ss -l
Recv-Q Send-Q Local Address:Port Peer Address:Port
0 10 :::5989 :::*
0 5 *:rsync *:*
0 128 :::sunrpc :::*
0 128 *:sunrpc *:*
0 511 *:http *:*
0 128 :::ssh :::*
0 128 *:ssh *:*
0 128 :::35766 :::*
0 128 127.0.0.1:ipp *:*
0 128 ::1:ipp :::*
0 100 ::1:smtp :::*
0 100 127.0.0.1:smtp *:*
0 511 *:https *:*
0 100 :::1311 :::*
0 5 *:5666 *:*
0 128 *:3044 *:*
```

ss列出每个进程名及其监听的端口

```
# ss -pl
```

ss列所有的tcp sockets

```
# ss -t -a
```

ss列出所有udp sockets

```
# ss -u -a
```

ss列出所有http连接中的连接

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

·以上包含对外提供的80，以及访问外部的80

·用以上命令完美的替代netstat获取http并发连接数，监控中常用到

ss列出本地哪个进程连接到x server

```
# ss -x src /tmp/.X11-unix/*
```

ss列出处在FIN-WAIT-1状态的http、https连接

```
# ss -o state fin-wait-1 '( sport = :http or sport = :https )'
```

ss常用的state状态：

```
established
syn-sent
syn-recv
fin-wait-1
fin-wait-2
time-wait
closed
close-wait
last-ack
listen
closing
all : All of the above states
connected : All the states except for listen and closed
synchronized : All the connected states except for syn-sent
bucket : Show states, which are maintained as minisockets, i.e. time-wait and syn-recv.
big : Opposite to bucket state.
```

ss使用IP地址筛选

```
ss src ADDRESS_PATTERN
src：表示来源
ADDRESS_PATTERN：表示地址规则

如下：
ss src 120.33.31.1 # 列出来之20.33.31.1的连接

#  列出来至120.33.31.1,80端口的连接
ss src 120.33.31.1:http
ss src 120.33.31.1:80
```

ss使用端口筛选

```
ss dport OP PORT
OP:是运算符
PORT：表示端口
dport：表示过滤目标端口、相反的有sport
```

OP运算符如下：

```
<= or le : 小于等于 >= or ge : 大于等于
== or eq : 等于
!= or ne : 不等于端口
< or lt : 小于这个端口 > or gt : 大于端口
```

OP实例

```
ss sport = :http 也可以是 ss sport = :80
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
```

为什么ss比netstat快：

netstat是遍历/proc下面每个PID目录，ss直接读/proc/net下面的统计信息。所以ss执行的时候消耗资源以及消耗的时间都比netstat少很多

ss命令帮助

```
# ss -h
Usage: ss [ OPTIONS ]
       ss [ OPTIONS ] [ FILTER ]
   -h, --help           this message
   -V, --version        output version information
   -n, --numeric        don't resolve service names
   -r, --resolve       resolve host names
   -a, --all            display all sockets
   -l, --listening      display listening sockets
   -o, --options       show timer information
   -e, --extended      show detailed socket information
   -m, --memory        show socket memory usage
   -p, --processes      show process using socket
   -i, --info           show internal TCP information
   -s, --summary        show socket usage summary
```

```
   -4, --ipv4          display only IP version 4 sockets
   -6, --ipv6          display only IP version 6 sockets
   -0, --packet display PACKET sockets
   -t, --tcp           display only TCP sockets
   -u, --udp           display only UDP sockets
   -d, --dccp          display only DCCP sockets
   -w, --raw           display only RAW sockets
   -x, --unix          display only Unix domain sockets
   -f, --family=FAMILY display sockets of type FAMILY

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

   -D, --diag=FILE     Dump raw information about TCP sockets to FILE
   -F, --filter=FILE   read filter information from FILE
       FILTER := [ state TCP-STATE ] [ EXPRESSION ]
```

参考：http://www.cyberciti.biz/tips/linux-investigate-sockets-network-connections.html

转摘请注明出处：Linux网络状态工具ss命令详解 http://www.ttlsa.com/html/2070.html