网络测试工具与方法

根据指标找工具（网络性能）		
性能指标	**工具**	**说明**
吞吐量（BPS）	sar nethogs iftop	分别可以查看网络接口、进程以及IP地址的网络吞吐量
PPS	sar /proc/net/dev	查看网络接口的PPS
连接数	netstat ss	查看网络连接数
延迟	ping hping3	通过ICMP、TCP等测试网络延迟
连接跟踪数	conntrack	查看和管理连接跟踪状况
路由	mtr route traceroute	查看路由并测试链路信息
DNS	dig nslookup	排查DNS解析问题
防火墙和NAT	iptables	配置和管理防火墙及NAT规则
网卡功能	ethtool	查看和配置网络接口的功能
抓包	tcpdump Wireshark	抓包分析网络流量
内核协议栈跟踪	bcc systemtap	动态跟踪内核协议栈的行为

根据工具查指标（网络性能）	
性能工具	**主要功能**
ifconfig ip	配置和查看网络接口
ss	查看网络连接数
sar /proc/net/dev/sys/class/net/eth0/statistics/	查看网络接口的网络收发情况
nethogs	查看进程的网络收发情况
iftop	查看IP的网络收发情况
ethool	查看和配置网络接口
conntrack	查看和管理连接跟踪状况
nslookup dig	排查DNS解析问题
mtr route traceroute	查看路由并测试链路信息
ping hping3	测试网络延迟
tcpdump	网络抓包工具
Wireshark	网络抓包和图形界面分析工具
iptables	配置和管理防火墙及NAT规则
perf	剖析内核协议栈的性能
systemtap bcc	动态追踪内核协议栈的行为

网络性能优化和数据测试的方法与工具有很多，以上两个表格分别从**网络性能指标**和**性能工具**两个维度列出了网络测试的工具。以下就以其中常用工具为例说明使用方法。

1、ping

ping工具用来测试连通性和延时，基于ICMP协议。比如，执行下面的命令，就可以测试本虚拟机到 172.16.90.153 的物理机地址的连通性和延时：

```
jared@ubuntu:~$ ping -c5 172.16.90.153
PING 172.16.90.153 (172.16.90.153) 56(84) bytes of data.
64 bytes from 172.16.90.153: icmp_seq=1 ttl=128 time=1.10 ms
64 bytes from 172.16.90.153: icmp_seq=2 ttl=128 time=0.616 ms
64 bytes from 172.16.90.153: icmp_seq=3 ttl=128 time=0.557 ms
64 bytes from 172.16.90.153: icmp_seq=4 ttl=128 time=0.476 ms
64 bytes from 172.16.90.153: icmp_seq=5 ttl=128 time=0.421 ms

--- 172.16.90.153 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4057ms
rtt min/avg/max/mdev = 0.421/0.634/1.103/0.245 ms
```

ping的输出可以分为两部分： 第一部分，是每个ICMP请求的信息，包括 ICMP 序列号（icmp_seq）、TTL（生存时间，或者跳数）以及往返延时time。 第二部分，5次ICMP请求的总汇。

2、ifconfig

ifconfig工具用来查看和修改网络接口的的配置和状态。

```
jared@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:07:74:f2
          inet addr:192.168.45.135  Bcast:192.168.45.255  Mask:255.255.255.0
          inet6 addr: fe80::b73b:1cb4:79e9:9f92/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:280992 errors:0 dropped:0 overruns:0 frame:0
          TX packets:180078 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:231108102 (231.1 MB)  TX bytes:14332590 (14.3 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:320795 errors:0 dropped:0 overruns:0 frame:0
          TX packets:320795 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:75242665 (75.2 MB)  TX bytes:75242665 (75.2 MB)
```

在shell中输入ifconfig可以看到系统中的网络设备有ens33以太网设备和lo回环设备，重点关注ens33。里面的信息可以分为如下四类： 第一，网络设备的状态标记： UP 和 RUNNING。UP表示网络设备的驱动使能，否则设备驱动关闭，无论网线是否接好都不能用，可以通过 `sudo ifconfig ens33 up` 和 `sudo ifconfig ens33 down` 分别打开和关闭设备。RUNNING表示物理网络是联通的，即网卡已经连接到了交换机或者路由器中。如果你看不到它们，通常表示网线被拔掉了。 第二，MTU。MTU 默认大小是 1500，根据网络架构的不同可能需要调大或者调小 MTU 的数值。设置方法通过如下命令: `sudo ifconfig ens33 mtu newval`。 第三，网络接口的 IP 地址、子网以及 MAC 地址。这些都是保障网络功能正常工作所必需的，你需要确保配置正确。 第四，网络收发的字节数、包数、错误数以及丢包情况，特别是 TX 和 RX 部分的 errors、dropped、overruns、carrier 以及 collisions 等指标不为 0 时，通常表示出现了网络 I/O 问题。其中：

- errors 表示发生错误的数据包数，比如校验错误、帧同步错误等；

- dropped 表示丢弃的数据包数，即数据包已经收到了Ring Buffer但因为内存不足等原因丢包；
- overruns 也是一种丢包，表示超限数据包数，即网络 I/O 速度过快，导致Ring Buffer中的数据包来不及处理（队列满）而导致的丢包；
- carrier 表示发生 carrirer 错误的数据包数，比如双工模式不匹配、物理电缆出现问题等；
- collisions 表示碰撞数据包数。

3、netstat、ss

ifconfig只显示了网络接口收发数据包的统计信息，没有网络协议栈中的统计信息，可以**通过netstat或者ss来查看套接字、网络栈、网络接口以及路由表的信息。**

- **套接字信息**

  可以通过执行下面的netstat命令查询套接字信息，ss用法一样。

```
rlk@ubuntu:socket$ ./tcp_server &
[1] 34947
rlk@ubuntu:socket$ ./udp_server &
[2] 34948
rlk@ubuntu:socket$ netstat -lpn | head -n 5
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:8078            0.0.0.0:*               LISTEN      34947/./tcp_server
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
rlk@ubuntu:socket$ netstat -lpnu | head -n 5
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
udp        0      0 0.0.0.0:8088            0.0.0.0:*                           34948/./udp_server
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
```

  其中-l表示只显示监听套接字，但是感觉非监听的也显示。-n表示显示数字地址和端口。-p表示显示进程信息。-t表示只显示tcp套接字。-u表示只显示udp套接字。上面的netstat命令显示了套接字的状态、接收队列、发送队列、本地地址、远端地址、进程 PID 和进程名称等。 其中，接收队列（Recv-Q）和发送队列（Send-Q）需要特别关注，它们通常应该是 0。当发现它们不是0时，说明有网络包的堆积发生。当然还要注意，在不同套接字状态下，它们的含义不同。 当套接字处于连接状态（Established）时，

  - Recv-Q表示套接字缓冲中还没有被应用程序取走的字节数（即接收队列长度）。
  - Send-Q 表示还没有被远端主机确认的字节数（即发送队列长度）。

  当套接字处于监听状态（Listening）时，

  - Recv-Q 表示 syn backlog 的当前值。
  - Send-Q 表示最大的 syn backlog 值。

  syn backlog 是 TCP 协议栈中的半连接队列长度，相应的也有一个全连接队列（accept queue），它们都是维护 TCP 状态的重要机制。 半连接，就是还没有完成 TCP 三次握手的连接，连接只进行了一半，而服务器收到了客户端的 SYN 包后，就会把这个连接放到半连接队列中，然后再向客户端发送 SYN+ACK 包。 全连接，则是指服务器收到了客户端的 ACK，完成了 TCP 三次握手，然后就会把这个连接挪到全连接队列中。这些全连接中的套接字，还需要再被 accept() 系统调用取走，这样，服务器就可以开始真正处理客户端的请求了。

- **协议栈统计信息**

```
netstat -s
.....
Tcp:
    1742 active connection openings
```

```
    3 passive connection openings
    917 failed connection attempts
    0 connection resets received
    4 connections established
    97127 segments received
    52841 segments sent out
    67 segments retransmitted
    0 bad segments received
    918 resets sent
Udp:
    98452 packets received
    8 packets to unknown port received
    1 packet receive errors
    98333 packets sent
    0 receive buffer errors
    0 send buffer errors
    InCsumErrors: 1
    IgnoredMulti: 905
.....
```

4、tcpdump、wireshark

tcpdump和wireshark用来抓取传输过程的网络包，tcpdump基于命令行，wireshark基于图形界面。

- tcpdump tcpdump**命令的基本格式**是tcpdump [选项] [过滤表达式] 。常用选项和过滤表达式如下所示。

tcpdump使用——选项类		
选项	**示例**	**说明**
-i	tcpdump -i eth0	指定网络接口，默认是0号接口（如eth0），any表示所有接口
-nn	tcpdump -nn	不解析IP地址和端口号的名称
-c	tcpdump -c5	限制要抓取网络包的个数
-A	tcpdump -A	以 ASCII 格式显示网络包内容（不指定时只显示头部信息）
-w	tcpdump -w file.pcap	保存到文件中，文件名通常以 .pcap 为后缀
-e	tcpdump -e	输出链路层的头部信息

tcpdump使用——过滤表达式类		
表达式	**示例**	**说明**
host、src host、dst host	tcpdump –nn host 35.190.27.188	主机过滤
net、src net、dst net	tcpdumpnet –nn 192.168.0.0	网络过滤
port、portrange、src port、dst port	tcpdump –nn dst port 80	端口过滤
ip、ip6、arp、tcp、udp、icmp	tcpdump –nn tcp	协议过滤
and、or、not	tcpdump –nn icmp or udp	逻辑表达式
tcp[tcpflags]	tcpdump –nn ”tcp[tcpflags] & tcp–syn != 0”	特定状态的TCP包

接下来举例看tcpdump的用法： 首先在一个终端输入如下命令。

```
$ sudo tcpdump -nn host 192.168.45.135
```

其中-nn表示不解析抓包中的域名（即不反向解析）、协议以及端口号。host 192.168.45.135表示只显示IP 地址（包括源地址和目的地址）为 35.190.27.188 的包。 然后在另一个终端输入如下命令。

```
$ ping ubuntu.gree.com
```

接着再回到终端1查看tcpdump的输出信息，tcpdump的**输出格式**的基本形式为：

```
时间戳 协议 源地址．源端口 > 目的地址．目的端口 网络包详细信息
```

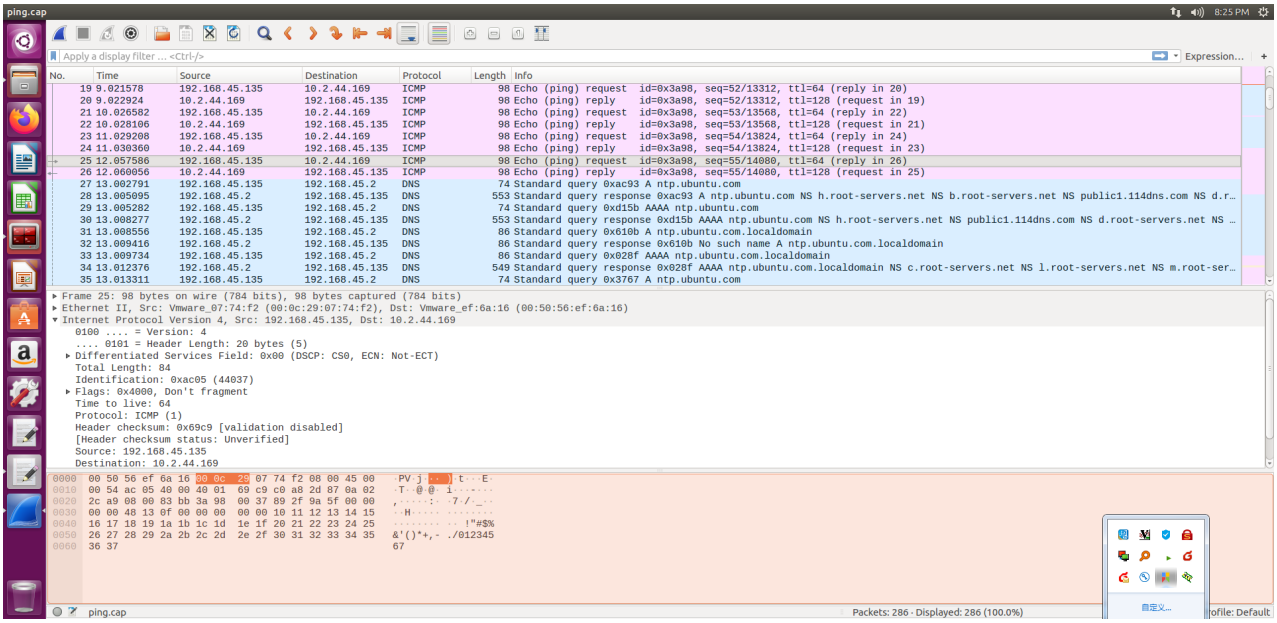
```
19:24:22.142154 IP 192.168.45.135.41349 > 192.168.45.2.53: 44984+ A? ntp.ubuntu.com.localdomain. (44)
19:24:22.142770 IP 192.168.45.2.53 > 192.168.45.135.41349: 44984 NXDomain*- 0/0/0 (44)
19:24:22.142983 IP 192.168.45.135.41349 > 192.168.45.2.53: 36487+ AAAA? ntp.ubuntu.com.localdomain. (44)
19:24:22.148452 IP 192.168.45.2.53 > 192.168.45.135.41349: 36487- 0/14/1 (507)
19:24:23.033947 IP 192.168.45.135 > 10.2.44.169: ICMP echo request, id 11209, seq 287, length 64
19:24:23.035629 IP 10.2.44.169 > 192.168.45.135: ICMP echo reply, id 11209, seq 287, length 64
19:24:24.037745 IP 192.168.45.135 > 10.2.44.169: ICMP echo request, id 11209, seq 288, length 64
19:24:24.039157 IP 10.2.44.169 > 192.168.45.135: ICMP echo reply, id 11209, seq 288, length 64
19:24:25.039540 IP 192.168.45.135 > 10.2.44.169: ICMP echo request, id 11209, seq 289, length 64
```

- wireshark

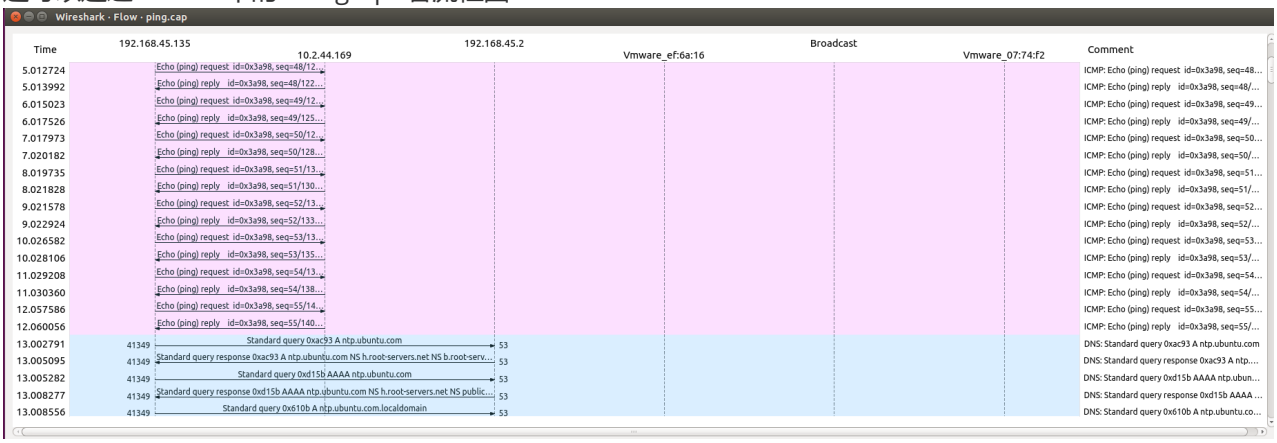
  首先和上面类似，先在一个终端输入如下命令。然后tcpdump输出的信息会保存到ping.cap中。

  ```
  $ sudo tcpdump -nn host 192.168.45.135 -w ping.cap
  ```

  然后在另一个终端ping ubuntu.gree.com。最后用wireshark打开ping.cap，得到如下的界面：



还可以通过statistic中的flow graph看流程图：



5、sar

sar是个综合性工具，这里主要讲**如何使用sar来查看系统当前的网络吞吐量和 PPS。** 给sar增加 -n 参数就可以查看网络的统计信息，比如网络接口（DEV）、网络接口错误（EDEV）、TCP、UDP、ICMP 等等。执行下面的命令，你就可以得到网络接口统计信息：

```
jared@ubuntu:~$ sar -n DEV 1
Linux 4.15.0-120-generic (ubuntu)       10/28/2020      _x86_64_        (8 CPU)

01:53:48 AM     IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s   rxcmp/s   txcmp/s  rxmcst/s   %ifutil
01:53:49 AM     ens33     40.00     40.00     17.00      3.12      0.00      0.00      0.00      0.01
01:53:49 AM        lo     80.00     80.00     19.03     19.03      0.00      0.00      0.00      0.00
```

- rxpck/s 和 txpck/s 分别是接收和发送的 PPS，单位为包 / 秒。
- rxkB/s 和 txkB/s 分别是接收和发送的吞吐量，单位是 KB/ 秒。
- rxcmp/s 和 txcmp/s 分别是接收和发送的压缩数据包数，单位是包 / 秒。
- %ifutil 是网络接口的使用率，即半双工模式下为 (rxkB/s+txkB/s)/Bandwidth，而全双工模式下为 max(rxkB/s, txkB/s)/Bandwidth。

6、ethtool、iperf3

- 带宽 在上一小节中计算网络接口的使用率需要带宽Bandwidth。可以通过ethtool来查询，它的单位通常是 Gb/s 或者 Mb/s，不过注意这里小写字母 b ，表示比特而不是字节。我们通常提到的千兆网卡、万兆网卡等，单位也都是比特。

```
jared@ubuntu:~$ sudo ethtool ens33 | grep Speed
    Speed: 1000Mb/s
```

通过以上命令可以看到这是个千兆网口。

- 吞吐量

  用iperf3来测试网络吞吐量。 首先，在shell(机器)上启动 iperf 服务端：

  ```
  # -s 表示启动服务端，-i 表示汇报间隔，-p 表示监听端口
  $ iperf3 -s -i 1 -p 10000
  ```

  然后，在另一个shell（机器）上运行iperf 客户端，运行测试：

  ```
  # -c 表示启动客户端，192.168.0.30 为目标服务器的 IP
  # -b 表示目标带宽（单位是 bits/s）
  # -t 表示测试时间
  # -P 表示同时启动的客户端数，-p 表示目标服务器监听端口
  $ iperf3 -c 192.168.45.135 -b 1G -t 10 -P 1 -p 10000
  ```

  10秒后回到iperfs服务器端的shell查看报告：

  ```
  - - - - - - - - - - - - - - - - - - - - - - - - -
  [ ID] Interval           Transfer     Bandwidth       Retr
  [  5]   0.00-10.05  sec  1.15 GBytes   986 Mbits/sec    0             sender
  [  5]   0.00-10.05  sec  1.15 GBytes   986 Mbits/sec                  receiver
  ```

  可以看到吞吐量为986Mb/s，稍微比带宽小一点。